technische universität
dortmund

fakultät
statistik

Comparing Simulation Strategies and Quantifying Similarity of Datasets

Dissertation
in Fulfillment of the Requirements for the Degree of
Doktor der Naturwissenschaften

Submitted to the
Department of Statistics
of the
TU Dortmund University

by
Marieke Stolte
on
April 9, 2025

Referees:
Prof. Dr. Jörg Rahnenführer
Prof. Dr. Andreas Groll
Prof. Dr. Jan G. Hengstler

Date of Oral Thesis Defense:
May 15, 2025

Abstract

Simulation studies are an essential tool for comparing and evaluating new and existing statistical methods. Generating realistic data is considered crucial for the reliability of the simulation results. There are various types of simulation studies that differ in the way in which data is generated. In parametric simulation studies, the data is generated using pseudo-random numbers according to a fully user-specified data-generating mechanism. The complete specification of the data-generating mechanism, i.e. of the data-generating process (DGP) for the covariates and the outcome-generating model (OGM) for generating observations of a target variable based on the generated covariates, might however result in oversimplification of complex real-world processes. An alternative to parametric simulation that is often claimed to produce more realistic data is statistical Plasmode simulation. For this, covariate data is generated by resampling from a real-world dataset. Observations of a target variable are then obtained by applying a user-specified OGM to that resampled data. The claim that Plasmode simulation leads to more realistic data and therefore better simulation results is, however, not proven by any empirical or theoretical results. Therefore, this thesis presents the first empirical comparison of parametric and Plasmode simulation studies. The estimation of the mean squared error (MSE) of the least squares (LS) estimator in linear regression, as well as the comparison of several binary classification methods, are considered as examples.

In the context of comparing different simulation strategies, the similarity of the simulated datasets to a real-world dataset is of interest. There are several methods for quantifying the similarity of two or more multivariate datasets proposed in the literature. Yet, there is no guidance available on which method to use when. Therefore, the remainder of the thesis is concerned with comparing methods for quantifying dataset similarity. First, a taxonomy of such methods based on their main ideas is provided together with a comparison based on 22 newly developed theoretical criteria for the applicability, interpretability, and theoretical properties of the methods. These can guide the choice of a suitable method for a given dataset comparison. To facilitate the choice in practice, an online tool is provided that allows for custom filtering of the theoretical criteria and sorting of the methods. To enable an empirical method comparison, an `R` package is provided that includes the most relevant dataset similarity methods implemented in a unified framework. Finally, a neutral comparison study of dataset similarity methods for categorical data is performed to provide insight into the performance of such methods in practice.

Acknowledgments

I would like to thank my supervisor Jörg for all his support during the development of this thesis. Thank you for your continuous and uncomplicated help whenever I needed it and at the same time for giving me the freedom to work on my own wherever possible.

I would also like to thank Prof. Hengstler for the interesting collaborations we had and Prof. Groll for taking the time to grade this thesis.

A special thanks goes to Andrea for her additional supervision of all projects included in this thesis. Your ideas and suggestions always improved my work remarkably. Thanks also to my co-authors Nicholas, Alla, Maral, and Axel for the many helpful discussions and suggestions. It was a pleasure working with you and the basis for this entire thesis.

In addition, I would like to thank my colleagues in my working group, and the RTG, with whom I have had a great time working together. In particular, I would like to thank Franziska for all her support, both professional and emotional, and for all the shared laughter. I would also like to thank the SBAZ team. It was a great learning experience and a lot of fun to work there. Thanks also to Uwe, who supported me throughout my studies and always had an open ear for any technical challenges that I encountered while working on this thesis.

Finally, I would like to thank my family and friends who have been there for me from the beginning. Thank you for always encouraging me.

This work was partly supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG), Collaborative Research Center SFB 876, project A3, and by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project P1) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation – Project Number 427806116).

I gratefully acknowledge the computing time provided on the Linux HPC cluster at TU Dortmund University (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

List of Publications

This cumulative thesis is based on the following five manuscripts:

Article 1: Stolte, M., Schreck, N., Slynko, A., Saadati, M., Benner, A., Rahnenführer, J., and Bommert, A. (2024c): "Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression", in: *PLOS ONE* 19 (5), e0299989, DOI: 10.1371/journal.pone.0299989

Contribution of the author:

The author of this thesis took part in the conceptualization of the simulation study, performed the implementation of the study as well as the analysis and visualization of the results, and wrote the first draft of the manuscript. All other authors discussed the procedure and results and revised the manuscript. Andrea Bommert and Jörg Rahnenführer were, in addition, more deeply involved in designing the simulation study and also supervised the project.

Article 2: Stolte, M., Schreck, N., Slynko, A., Saadati, M., Benner, A., Rahnenführer, J., Bommert, A., and for the topic group "High-dimensional data" (TG9) of the STRATOS Initiative (2025c): "Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in comparing classification methods on high-dimensional data", in: *PLOS ONE* 20 (6), pp. 1–36, DOI: 10.1371/journal.pone.0322887

Contribution of the author:

The authors' contributions are the same as for Article 1.

Article 3: Stolte, M., Kappenberg, F., Rahnenführer, J., and Bommert, A. (2024b): "Methods for quantifying dataset similarity: a review, taxonomy and comparison", in: *Statistics Surveys* 18, pp. 163–298, DOI: 10.1214/24-SS149

Contribution of the author:

The author of this thesis performed the review of all methods and the evaluation of all criteria and wrote the original draft of the manuscript except for the Introduction, which was written by Jörg Rahnenführer, and the first three methods in Section 3.3.1, for which the first draft was written by Andrea Bommert. In addition, the author of this thesis developed the criteria used for the method evaluation together with Andrea Bommert and implemented the interactive online table that supplements the manuscript. Andrea Bommert and Jörg Rahnenführer discussed and supervised the whole project and revised the manuscript. Revising was done together with Franziska Kappenberg.

Article 4: Stolte, M., Sauer, L., Rahnenführer, J., and Bommert, A. (2025b): "DataSimilarity: an R Package for Quantifying Similarity of Datasets and Multivariate Two- and k-Sample Testing", Unpublished

Contribution of the author:

The author of this thesis conceptualized and implemented the R package except for the functions `BMG()`, `Jeffreys()`, `LHZ()`, `BG()`, `Petrie()`, and `MMCM()` which were implemented by Luca Sauer. Luca Sauer also documented these functions, wrote the corresponding parts of the first draft of the manuscript, and helped with the application examples for the paper and the overall implementation and documentation of the package. The remaining documentation and original draft of the manuscript were written by the author of this thesis. Jörg Rahnenführer and Andrea Bommert supervised the process and revised the manuscript.

Article 5: Stolte, M., Rahnenführer, J., and Bommert, A. (2025a): "An Empirical Comparison of Methods for Quantifying the Similarity of Categorical Datasets", Unpublished

Contribution of the author:

The author of this thesis designed, implemented, and analyzed the simulation study and wrote the initial draft of the manuscript. The development of the evaluation criteria, discussions, and revising were done together with Andrea Bommert and Jörg Rahnenführer, who in addition supervised the project.

Further publications:

1. Stolte, M., Albrecht, W., Brecklinghaus, T., Gründler, L., Chen, P., Hengstler, J. G., Kappenberg, F., and Rahnenführer, J. (2023): "Classification of hepatotoxicity of compounds based on cytotoxicity assays is improved by additional interpretable summaries of high-dimensional gene expression data", in: *Computational Toxicology* 28, p. 100288, DOI: 10.1016/j.comtox.2023.100288

2. Stolte, M., Herbrandt, S., and Ligges, U. (2024a): "A comprehensive review of bias reduction methods for logistic regression", in: *Statistics Surveys* 18, pp. 139–162, DOI: 10.1214/24-SS148
   This article is based on the Master thesis "Bias in Logistic Regression [Verzerrung in der logistischen Regression]". It is, however, not part of this thesis.

3. Albrecht*, W., Brecklinghaus*, T., Stolte*, M., Kappenberg, F., Gründler, L., Chen, P., Cadenas, C., Damm, G., Edlund, K., Ghallab, A., Marchan, R., Nell, P., Reinders, J., Seehofer, D., Behr, A.-C., Braeuning, A., van Thriel, C., Gardner, I., Rahnenführer, J., and Hengstler, J. G. (2025): "Improved identification of human hepatotoxic potential by summary variables of gene expression", in: *ALTEX - Alternatives to animal experimentation*, DOI: 10.14573/altex.2403272

* Shared first authorship

Contents

Part I.

Introduction

1. Motivation

Simulation studies are a crucial tool in statistics as they enable researchers to evaluate and compare new or existing methods. They can, therefore, offer guidance for choosing appropriate methods in practice. Moreover, they can be used to check analytical results and code, assess the relevance of asymptotic approximations in finite samples, evaluate the performance of methods under the violation of assumptions, or for sample size and power calculations (Morris et al., 2019; Boulesteix et al., 2020; Friedrich and Friede, 2024). Typically, simulation studies are defined as computer experiments that involve generating data with some known truth and comparing the results of certain methods to that known truth (Burton et al., 2006; Morris et al., 2019; Boulesteix et al., 2020; Schreck et al., 2024).

A major advantage of simulation studies is that they can be used in situations where analytical results are hard or even impossible to obtain (Morris et al., 2019; Pawel et al., 2024). This has become particularly important nowadays since the more complex statistical modeling and machine learning methods cannot be evaluated mathematically as easily considering that mathematics often only covers simple cases of limited relevance and under certain assumptions about the data that are hard to verify in practice (Boulesteix et al., 2020). At the same time, simulations enable the assessment of method performance in situations where analyzing real-world data alone does not suffice since the assessment requires knowledge of some aspects of the true data-generating mechanism. For example, the true parameter values have to be known when analyzing bias, coverage, type I error rates, or power (Burton et al., 2006; Boulesteix et al., 2020). Assessments like this occur especially in situations with a focus on hypothesis testing, in regression tasks where the primary goal is explanation and not prediction, or in clustering. The control of the ground truth allows comparing the results for a method to this truth and enables investigating a large number of scenarios, including new and rare scenarios. Moreover, it allows for the systematic assessment of how the method performance depends on the assumptions and parameters by systematically varying these. Using a high number of repetitions in the simulation also makes it possible to average out random variation (Boulesteix et al., 2020). These points also distinguish simulation studies from *benchmark studies*, where several methods are compared on a set of so-called benchmark datasets (Friedrich and Friede, 2024).

Different types of simulation studies differ in the way in which data is generated within the study. For *parametric simulation*, data is generated by drawing pseudo-random numbers from a data-generating mechanism that is fully specified by a parametric stochastic model. The data is, therefore, fully artificial. Parametric simulation is most extensively studied and used (Schreck et al., 2024). In contrast, *nonparametric simulation* is based exclusively on resampling from a real-world dataset to generate data. This comes with the problem of not knowing any aspects of the data-generating mechanism (Boulesteix et al., 2020). It differs from the aforementioned benchmarking studies in the aspect that for a benchmarking study typically the whole dataset is used as it is, while for a nonparametric simulation study, many datasets are generated by resampling from the real-world dataset, which allows for an assessment of the variability of the results of the methods that are ap-

plied to the data. Depending on the definition, nonparametric simulation might not even be considered a simulation study. Burton et al. (2006) for example differentiate between simulation and "resampling studies". There are different ways to compromise between parametric and nonparametric simulation. One popular option is to base a parametric simulation study on a real-world dataset and, for example, estimate the parameters in the data-generating mechanism from this dataset (Burton et al., 2006; Friedrich and Friede, 2024; Schreck et al., 2024). Another approach that can be seen as semi-parametric simulation is the so-called *statistical Plasmode simulation*. For a Plasmode simulation study, covariate data is resampled from a real-world dataset, and observations for a target variable are generated artificially by a user-specified parametric model. This allows generating the covariate data based on a real-world dataset while still some truth is known through the application of the known outcome-generating model (Schreck et al., 2024).

The choice of how best to generate the data for a simulation study is hard. Schreck et al. (2024) stress that the quality of parametric simulation depends on how well the assumed data-generating mechanism reflects reality, which might limit the conclusions of the study. Boulesteix et al. (2020) differentiate between simulation studies that focus on a specific application and simulation studies that focus on the general behavior of methods. In the former case, the primary goal is to simulate data as similar as possible to a real-world dataset of interest. In the latter case, the primary goal is to cover a broad spectrum of plausible scenarios, which can also include unrealistic scenarios that help to understand the behavior of the method. However, as Burton et al. (2006) point out, arbitrary parameter choices in a simulation study can be criticized for not presenting realistic scenarios. Therefore, they claim that "[t]he simulated datasets should have some resemblance to reality" to ensure that the results generalize to real data and for the credibility of the simulation results. Boulesteix et al. (2020) also mention that the choice of the data-generating mechanism should reflect the distribution and relevant characteristics of some real-world data of interest. In the context of neutral comparison studies, Boulesteix et al. (2013) argue to choose datasets as representative as possible, possibly sampled from the domain of interest. So, overall, generating somewhat "realistic" data is of high relevance in simulation studies and might, therefore, be a main criterion in the choice of the data generation method.

According to Boulesteix et al. (2020) a main drawback of parametric simulation is that simplified scenarios do not reflect complex real-life data, which can result in a distorted impression of method performance. This risk of over-simplification in parametric simulation is commonly pointed out in the literature (Vaughan et al., 2009; Schreck et al., 2024). Since parametric simulations often make strong assumptions regarding the underlying distributions and dependence structure, they may be unable to capture complex real-world structures, which might result in misleading conclusions. This becomes even more severe with high-dimensional or otherwise complex data (Schreck et al., 2024). Burton et al. (2006) suggest using a real dataset as a motivating example and simulating data to "closely represent the structure of this real dataset" by, e.g., using the covariate data as it is and generating only artificial outcomes or by estimating, e.g., the correlation structure. Friedrich and Friede (2024) also suggest combining parametric simulation with the use of real-world data.

The ability to easily vary the assumptions on the data-generating mechanism is, at the same time, the main strength of parametric simulation. Combined with the possibility to generate many datasets, this grants high flexibility to the researcher performing the study (Schreck et al., 2024).

Plasmode simulations, on the other hand, are claimed to generate data that resembles reality in the closest way (Mehta et al., 2004) while still some truth is known. However, they still make several implicit assumptions. The generalizability of the results of a Plasmode study is limited by the representativity of the underlying data sample (Schreck et al., 2024).

Schreck et al. (2024) discuss the strengths and weaknesses of parametric in Plasmode simulation in detail. They conclude that "[a]ll in total, plasmode data sets may provide an attractive supplement to parametric simulations and can be applied in order to increase the reliability of the obtained research results." In their outlook, they point out the need for research on the impact of the specification of the outcome-generating model and the choice of the resampling scheme in Plasmode simulations, as these could potentially limit how realistic the simulated data in a Plasmode study is.

Therefore, one goal of this work is to perform an empirical comparison of parametric and Plasmode simulation that takes these aspects into account. The main idea in this comparison is that parametric simulation would be the obvious best choice if the true data-generating mechanism was known. However, the true data-generating mechanism is typically unknown in reality, so instead, as pointed out before, assumptions are made. Thus, the idea is to vary these assumptions and compare the results of the simulations based on these assumptions to the results when making the correct assumptions. This then enables an evaluation of how far the assumptions made in parametric simulation can differ from the truth until parametric simulation becomes worse than Plasmode as claimed in the literature. To begin with, the comparison is performed for the simple example of estimating the mean squared error of the least squares estimator in linear regression (Stolte et al., 2024c). In that study, the focus is on low-dimensional data consisting of up to 50 variables and the evaluation of a single, simple model whose properties are well-known. The next step is to perform the comparison for the more complicated but also more realistic example of comparing the classification performance of multiple classification methods (Stolte et al., 2025c). In that study, higher dimensional data with up to 150 variables is used, and a method comparison including black box models is performed. In both cases, it could be shown that the performance of the parametric simulation study critically depends on how well the assumptions made in the generation of the covariates resemble the truth. For settings that are far from the truth, Plasmode simulation outperformed parametric simulation. Wrong assumptions on the outcome-generating model affected parametric and Plasmode simulation equally. The choice of the resampling scheme had a notable effect on the results of the Plasmode simulation, but there was no clear conclusion regarding the best-performing resampling scheme.

In both studies, parameters of the data-generating mechanism are used to describe how much assumptions differ from the truth. These parameters of the true data-generating mechanism are typically unknown in a real study, so comparing the parameters that are used in the study to the true ones is not a practical option for quantifying how closely the chosen parameters resemble the truth. Since the results of the two studies suggest that simulation results are negatively impacted by generating data that is far from the truth, it might still be of interest to check that the assumptions used in the data generation are close to the truth. Burton et al. (2006) already pointed out that "[t]he generated data should be verified to ensure they resemble what is being simulated". They suggest comparing estimates on simulated and real data. Schreck et al. (2024) also suggest comparing the generated Plasmode data to the underlying real-world dataset as a quality check step in their step-by-step instruction on performing Plasmode simulations. In their outlook, they

state that the data generation method can be considered realistic if it reflects the real data structure and dependencies most accurately and conclude that, for verifying this, a distance measure between the generated Plasmode dataset and the real dataset is needed.

So, in the context of checking whether the assumptions made in the data generation of a simulation study are realistic, a method for quantifying how similar the generated datasets are to a dataset from the true but unknown data-generating mechanism would be of great help. Therefore, a literature search for such methods was performed. It revealed a plethora of methods as quantifying the similarity (or equivalently the distance) between two or more datasets has widespread applications in statistics and machine learning in addition to the aforementioned one. These applications include, for example, meta-learning or transfer learning in which the similarity of datasets is used to transfer results from one learning task to another. The largest application field is two- and k-sample testing, $k \in \mathbb{N}, k \geqslant 2$, where it is checked if two or more distributions coincide. The search was restricted to methods that are applicable to multivariate data, do not require parametric distributional assumptions, and do not focus on a particular property of the datasets or their underlying distributions. Therefore, the vast literature on univariate methods, methods devoted to certain distribution families like the normal distribution, and methods that only consider certain alternatives like shift or scale was excluded. Even with these restrictions, there were still plenty of methods left that could potentially be used to tackle the problem. Due to the lack of comparison studies, it was unclear which method to use. There are very limited method comparison studies of only a few of the available methods (Székely and Rizzo, 2004; Gretton et al., 2012; Biswas et al., 2014; Biswas and Ghosh, 2014; Petrie, 2016; Chen and Friedman, 2017; Lopez-Paz and Oquab, 2017; Chen et al., 2018; Pan et al., 2018; Mukhopadhyay and K. Wang, 2020; Hediger et al., 2021; Li et al., 2022; Mukherjee et al., 2022; Song and Chen, 2022; Zaremba, 2022; Huang and Sen, 2024; Song and Chen, 2023). However, none of them are neutral in the sense of Boulesteix et al. (2013) since they are all conducted in the context of presenting a new method and showing that it is competitive.

Therefore, another goal of this thesis is to provide a comprehensive comparison of methods for quantifying the similarity of datasets to finally provide guidance for choosing an appropriate method based on the datasets at hand and the goal of the dataset comparison. The procedure for this is as follows. First, the literature is reviewed, and a taxonomy of the methods based on their underlying ideas is provided to give a better overview of the methods. Then, a theoretical comparison with regard to applicability, interpretability, and theoretical properties like invariances, metric properties, and (if applicable) consistency of the respective test is performed, which enables narrowing the available methods down to a set of methods that are eligible for the data at hand and the goal of the dataset comparison (Stolte et al., 2024b). An online tool to interactively explore the results of this theoretical comparison is provided (`https://shiny.statistik.tu-dortmund.de/data-similarity/`). Based on the results of the theoretical comparison, an empirical comparison of the most promising methods is performed as there is a lack of neutral comparison studies for dataset similarity methods. The empirical comparison here includes the methods that are already implemented and, therefore, readily available for application by practitioners or that are promising since they are among the best-performing methods in the theoretical comparison either overall or within their class in the taxonomy. Implementing the empirical comparison requires implementing additional methods and unifying the in- and output formats of already implemented methods. These new and unified implementations are made available in the `R` package `DataSimilarity` (Stolte and Sauer, 2025; Stolte et al., 2025b). For categorical

datasets, no previous studies are present such that a comparison study is most needed there. In the empirical comparison (Stolte et al., 2025a), it is assessed how well methods detect certain differences between categorical datasets, i.e., differences in the class probabilities in the underlying distributions. The study conducted here also considers numerical aspects such as computing errors, memory required, and runtime. In the end, methods are clustered to find groups of methods that behave similarly. The aim is to be able to recommend methods for detecting certain differences in datasets and recommend a group of methods that together cover a broad range of alternatives, as it is to expect that no method fits all (Strobl and Leisch, 2024).

The remainder of this thesis is structured as follows. In Chapter 2, the statistical methods are presented. This includes an overview of simulation studies (Section 2.1) consisting of general considerations (Section 2.1.1), and more detailed descriptions of parametric (Section 2.1.2) and Plasmode (Section 2.1.3) simulation. Moreover, an overview of methods for quantifying the similarity of two or more datasets is given (Section 2.2) by describing the ideas of the classes defined in the taxonomy and briefly describing some example methods (Section 2.2.1). The criteria for the theoretical and empirical comparison of dataset similarity methods are also explained (Section 2.2.2). In Chapter 3, each of the five articles of which this thesis consists is summarized. Chapter 4 gives a summary of the thesis and an outlook to open research aspects. All full-length articles are attached thereafter.

2. Statistical Methods

2.1. Simulation Studies

In the following, simulation studies are introduced. First, general considerations for planning, conducting, and reporting simulation studies are presented. Then, the special cases of parametric and Plasmode simulations are described in more detail and compared.

2.1.1. General Considerations

Simulation studies are typically defined as computer experiments that involve generating (pseudo-random) data with some known truth and evaluating methods on the generated data by comparing their results to that known truth (Burton et al., 2006; Morris et al., 2019; Boulesteix et al., 2020; Schreck et al., 2024). The process can be seen as mimicking the process of repeatedly drawing samples from a large population by repeatedly generating synthetic data under pre-specified assumptions (Boulesteix et al., 2020). It can also be seen as a model-based approach since mathematical concepts and models need to be known (Friedrich and Friede, 2024). Through the control of the data generation, simulations allow for the systematic assessment of the influence of parameters and assumptions on method performance (Boulesteix et al., 2020).

In the following, the critical steps in planning, conducting, and reporting simulation studies are presented according to the *ADEMP* structure by Morris et al. (2019). Earlier, Burton et al. (2006) presented a protocol procedure for planning, analyzing, and reporting that includes similar points. Chipman and Bingham (2022) also give step-by-step instructions for planning, executing, and analyzing simulation studies with a focus on using design and analysis of experiments. Sigal and Chalmers (2016) give hands-on instructions and considerations for conducting simulation studies in the context of teaching statistics. Smith and Marshall (2011) give considerations for designing good quality simulation studies, in particular in the context of clinical research / clinical trial simulation. Pawel et al. (2024) show how bad quality of design, execution, and reporting of simulation studies can result in misleading conclusions by demonstrating how questionable research practices can make methods appear superior even when they are not. They demonstrate this for a made-up example method and make recommendations on how to prevent questionable research practices. These recommendations include pre-registration of simulation protocols, incentivizing neutral simulation studies, and transparency by code and data sharing. The ADEMP structure includes the points raised in the aforementioned publications and has been most widely adopted by the statistics community. Therefore, it is chosen to be presented here in more detail.

Morris et al. (2019) suggest considering five important steps when planning and reporting a simulation study. These include the *aims* of the study *(A)*, the *data-generating mechanisms (D)*, the *estimands* and other targets *(E)*, the *methods (M)*, and the *performance measures (P)*. These individual points in ADEMP will now be explained in more detail.

Aims The aims of a simulation study typically relate to some desirable properties of an estimator like consistency, unbiasedness, consistency of the corresponding variance estimator, coverage of confidence intervals (CIs), or efficiency. Morris et al. (2019) differentiate three situations with respect to the aims. Studies might aim to perform a proof-of-concept or aim to find situations in which the methods might break down, or they might compare methods that are known to work in principle but address slightly different problems in realistic scenarios.

Data-generating mechanisms The *data-generating mechanisms* determine how random numbers are used to generate a dataset. Datasets could be generated using parametric draws from a known model, which is referred to as parametric simulation. Alternatively, data can be generated by repeated resampling from a specific dataset, e.g. in a statistical Plasmode simulation. Experimental design methods can be used to systematically vary more than one factor in the data-generating mechanism, e.g. the sample size or some effect size (see also Chipman and Bingham, 2022). A factorial or fractional factorial design allows assessing interactions between the factors but might be infeasible due to computing time. Within this thesis, the focus is on simulation studies where the data-generating mechanisms consist of a *data-generating process (DGP)* for generating covariate data and an *outcome-generating model (OGM)* for generating observations of an outcome variable based on the generated covariate data.

Estimands and other targets In most simulation studies, methods for estimating some population quantity are evaluated. This population quantity of interest is called the *estimand* and is typically a parameter of the data-generating model. Other targets of a simulation study can include testing a null hypothesis, model selection, or prediction.

Methods The *methods* evaluated in a simulation study are mostly the models for analysis but might also be a design or, more generally, a procedure. When choosing the methods to include in a simulation study, the following criteria should be considered. For a method comparison, serious competitors need to be included. This requires knowledge of previous work in the area. Methods that are already known to be flawed can be excluded except when they are frequently used in practice. Methods that are not implemented can be excluded as it can be argued that they have low practical relevance.

Performance measures The *performance measures* are numerical quantities used to assess the performance of a method. The choice of the performance measures has to match the aims and targets. For example, if the unbiasedness of an estimator is assessed, bias or relative bias might be appropriate performance measures. Morris et al. (2019) highlight the importance of reporting estimates of the uncertainty in the form of Monte Carlo standard errors (MCSEs) along with the estimated performance measures. They provide formulas for estimating frequently used performance measures and corresponding MCSEs. The MCSE can also be used to determine the number of repetitions by performing a sample size calculation for upper-bounding the MCSE. Morris et al. (2019) stress that how the number of repetitions was determined should be reported transparently.

In addition to presenting the ADEMP scheme, Morris et al. (2019) give instructions for the analysis and reporting of simulation results. They highlight the importance of reporting the design clearly and of reporting estimates of Monte Carlo uncertainty along with estimates of the performance measures.

For the data-generating mechanisms, one of the main choices is whether data should be simulated fully artificially by drawing from a parametric model or if data should be resampled from a real-world dataset. The former is known as *parametric simulation*. It is most extensively studied and widely used (Schreck et al., 2024). For parametric simulation, it is assumed that the parametric stochastic model for data generation is realistic and representative (Schreck et al., 2024). Parametric simulation can explore many different data-generating mechanisms, but these might be unrealistic (Morris et al., 2019). One type of simulation that involves resampling from a real-world dataset combined with artificial outcome generation is *Plasmode simulation* (Schreck et al., 2024). Resampling typically explores only one mechanism that is relevant for at least that study (Morris et al., 2019).

In the following, parametric and Plasmode simulations are described and discussed in more detail.

2.1.2. Parametric Simulation

For parametric simulation, all parts of the data-generating mechanism have to be defined in closed form and can be represented by a parametric stochastic model. The parameters of the data-generating mechanism can be estimated from real data, derived from literature, or set by the user. The main advantage of parametric simulation is its flexibility. The assumptions on the data-generating mechanism can easily be varied, and many datasets can be generated from the chosen data-generating mechanisms (Schreck et al., 2024). However, this representation by a parametric stochastic model might oversimplify real-world data-generating mechanisms (Vaughan et al., 2009; Boulesteix et al., 2020) as strong assumptions regarding the distributions and dependence structure are often made, which may be unable to capture complex real-world structures. This might result in misleading conclusions. It is especially hard to preserve complex dependence structures. This becomes even harder in high-dimensional data (Schreck et al., 2024).

In the following, the focus is on simulation studies where the data consists of covariates and an outcome variable. Examples of this are method evaluation or comparison studies for predictive modeling. The schematic process for one scenario of such a parametric simulation study is shown in Figure 2.1. First, some *data-generating process (DGP)* for the covariates has to be defined. This typically consists of the marginal distributions and a dependence structure that have to be chosen. Pseudo-random numbers are generated from this DGP using a random number generator (RNG), which yields a covariate dataset $X \in \mathbb{R}^{n \times p}$. Next, an *outcome-generating model (OGM)* has to be chosen. This could, for example, be a generalized linear model (GLM) (McCullagh and Nelder, 1989). In that case, the distribution family, link function, and coefficients of the GLM have to be specified. Artificial outcome observations $Y \in \mathbb{R}^n$ are then generated by applying this chosen OGM to the generated covariate dataset X. In case of a GLM this would correspond to multiplying the model matrix $\tilde{X} = [\mathbf{1}\, X] \in \mathbb{R}^{n \times (p+1)}$ with the coefficient vector $\beta \in \mathbb{R}^{p+1}$ to calculate the linear predictor $\eta = \tilde{X}\beta$. Then, the response function h is applied to each element of η and Y_i is drawn from the respective distribution with mean $\mu_i = h(\eta_i), i = 1, \ldots, n$. With X and Y, one complete dataset for the simulation has been generated on which the methods chosen in the "M" step of ADEMP can be applied. These can then

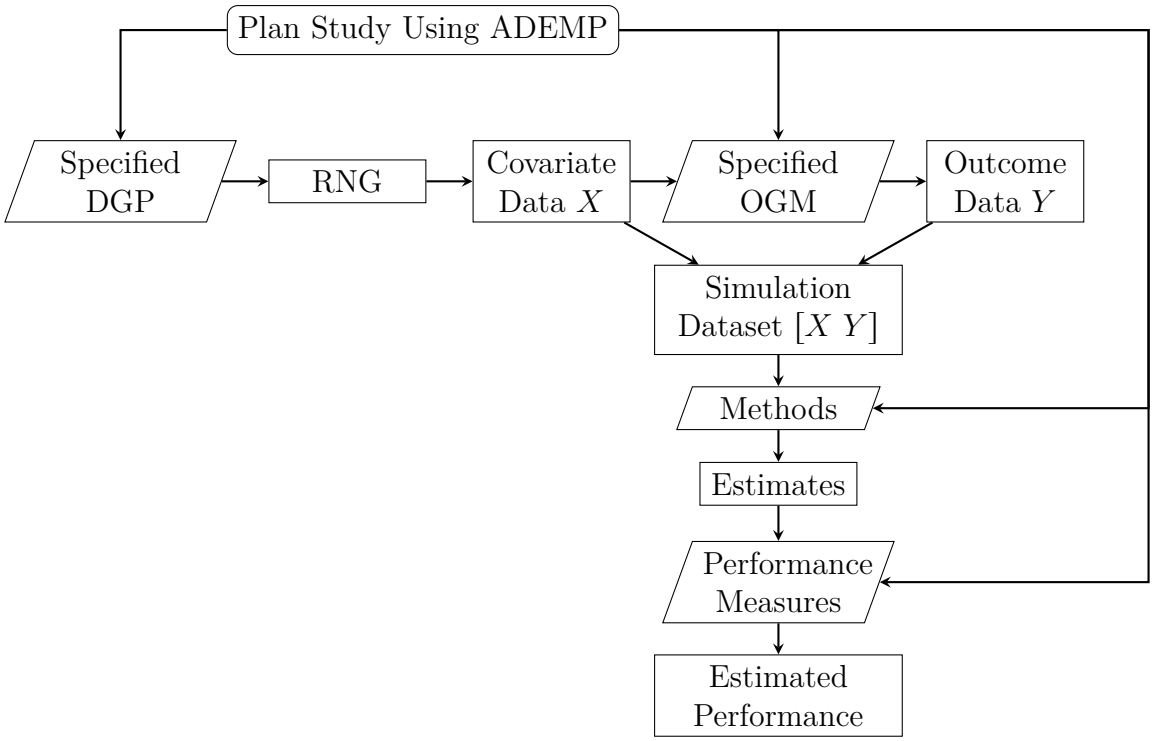


Figure 2.1.: Schematic process of performing a parametric simulation study.

be evaluated using the chosen performance measures. The process of generating data, applying the methods, and evaluating the performance measures is repeated according to the pre-specified number of repetitions B. This yields a set of estimates and performance measure values for this scenario consisting of the chosen DGP and OGM that can then be further analyzed to draw conclusions on the method performance.

2.1.3. Plasmode Simulation

The term Plasmode goes back to Cattell and Jaspers (1967). Schreck et al. (2024) differentiate between *biological Plasmodes* that are created in wet lab and *statistical Plasmodes* that use resampling from real-life datasets to generate covariate data and apply a user-specified outcome-generating model (OGM) to generate outcomes based on the resampled covariate data. An example of biological Plasmodes is spike-in experiments where, for example, a known quantity of RNA of a known sequence is mixed in with a probe to control that a method detects this increase in the corresponding gene expression (Mehta et al., 2006). Here, the focus is on statistical Plasmodes. The literature review of Schreck et al. (2024) shows that Plasmode simulation is often utilized for high-dimensional data, e.g., gene expression data.

A Plasmode simulation consists of two central steps. The generation of covariate data by resampling and potentially adding artificial covariates by a parametric model, and the outcome generation. The latter includes the choice of an appropriate OGM, the choice of covariate effects by individual specification or estimation based on original data, and the generation of new outcomes by drawing from the chosen OGM with specified effects applied to the generated covariate data. Due to the combination of resampling and the parametric OGM, Plasmodes can be seen as a semi-parametric simulation and, therefore, inherit the strengths and weaknesses of parametric simulation and resampling.

The main goal of Plasmode simulations is to preserve the covariate information. This requires the choice of an appropriate resampling scheme consisting of the number of generated datasets and the resampling technique. The aim is to make sure that the Bootstrap distribution of an estimator applied to the empirical distribution of the resampled data converges weakly to the theoretical distribution of that estimator. Otherwise, it is regarded as *Bootstrap failure*, and the results cannot be trusted. The estimator of interest here is typically some function of the covariance matrix of the covariates to ensure preserving the correlation structure (Schreck et al., 2024). There are no theoretical results for the complete Plasmode dataset consisting of the covariates and the outcome variable but only for the resampled covariates.

Several choices have to be made in the resampling scheme. The number of Plasmode datasets has to be chosen. This choice should be motivated. There are data-dependent procedures available in the literature (Andrews and Buchinsky, 2000; Davidson and MacKinnon, 2000). However, there are no general guidelines for choosing the number in a data-independent way such that asymptotic resampling results hold with sufficient accuracy. Moreover, the existing results might be invalid due to the additional application of the OGM (Schreck et al., 2024). For the resampling technique, there are several options. A discussion of drawing with vs. without replacement can be found in Schreck et al. (2024). In short, when drawing without replacement, one can choose to draw $m < n$ observations without replacement, which is called *subsampling* or *n-over-m Bootstrap*, or one can apply *sample-splitting Bootstrap* (cross-validation). When drawing with replacement, the options are drawing n observations with replacement, which is known as nonparametric or *n-out-of-n Bootstrap*, or drawing $m \leqslant n$ observations with replacement, which is called *m-out-of-n-Bootstrap*. Points to consider when choosing between the resampling techniques are that subsampling draws from the DGP of the original data and leads to consistent estimators under minimal conditions if m and n are appropriately specified. The Bootstrap draws from the empirical distribution derived from the underlying data, and additional assumptions (mainly: the influence of tied observations on the estimator should be small) are needed for consistent estimation. In the case that the Bootstrap is consistent, it is more efficient. The m-out-of-n-Bootstrap ($m \to \infty$, $\frac{m}{n} \to 0$) can prevent Bootstrap failure but is less efficient than the nonparametric Bootstrap if the latter is consistent. It is most often used in Plasmode studies. The m has to be chosen appropriately. An algorithm by Bickel and Sakov (2008) can be used to choose m in case of independent observations. There are theoretical results on guarantees for preserving the covariance structure by resampling but only for fixed p, $n \to \infty$, and using the nonparametric Bootstrap.

Irrespective of the resampling scheme, Plasmode simulation requires the representativity of the underlying data sample to achieve realistic simulations (Schreck et al., 2024). Like in the parametric simulation, the choice of artificial outcome generation leaves the choice of some aspects of the truth to the investigator. This choice determines the outcome type. Moreover, it might bias analysis results, e.g. by giving an advantage to certain models in model comparisons. The effects for the OGM can be chosen by sampling the coefficients from some distribution, estimating them on the original dataset, setting them manually, or by a mix of the aforementioned. Choosing the effects can be a strong intervention, e.g. by invalidating or nullifying existing associations, and can lead to unrealistic outcome generation (Schreck et al., 2024).

A schematic representation of the procedure of Plasmode simulation can be found in Figure 2.2. It is based on the step-by-step recommendations by Schreck et al. (2024). Beforehand, the research problem has to be formulated, and the study should be planned

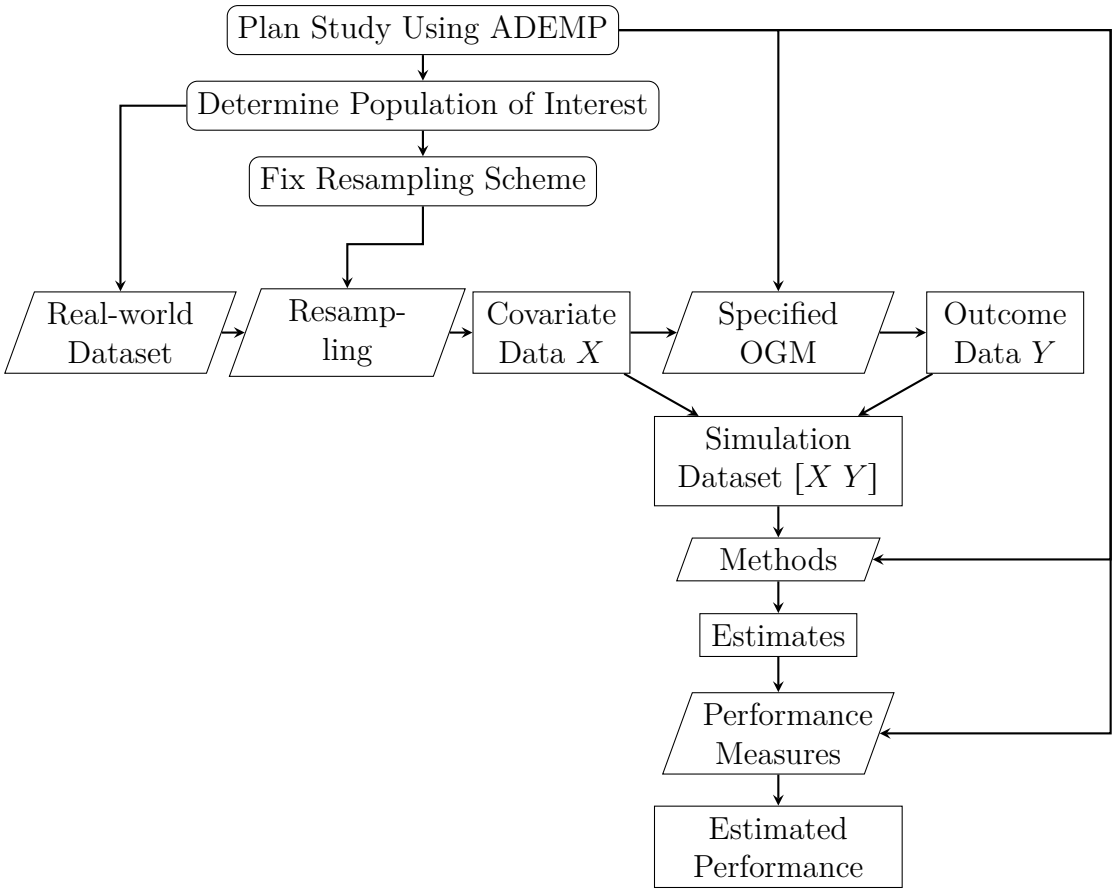


Figure 2.2.: Schematic process of performing a Plasmode simulation study. Adapted from Schreck et al. (2024).

using ADEMP. Moreover, the population of interest has to be determined, and a sample has to be taken from this population of interest. The representativeness of the sample and the sample size should be discussed and justified. Additionally, the resampling scheme has to be chosen, including the number of Plasmode datasets, the resampling technique, and the resampling size. All choices should be justified. Then the simulation proceeds as shown in Figure 2.2. A covariate dataset $X \in \mathbb{R}^{m \times p}$ is drawn from the sample of the population of interest by resampling according to the chosen resampling scheme. In this step, the preservation of the dependence structure should be considered. If applicable, an exposure-generating model can be chosen and applied to generate additional artificial covariates based on the resampled covariate data. From this step on, one proceeds as in parametric simulation.

An appropriate OGM has to be chosen and is then applied to the generated covariate data as in the parametric simulation. With this, new outcomes $Y \in \mathbb{R}^m$ are generated by applying the OGM to the resampled covariate data. The combination of the resampled covariate data and the generated outcomes then yields one Plasmode dataset. On this Plasmode dataset, again, the considered methods are applied and evaluated using the performance measures as for parametric simulation. The process is repeated according to the chosen number of Plasmode datasets to draw. Schreck et al. (2024) suggest comparing the distributions of the real and generated data as a quality check. Moreover, they emphasize that when reporting the study, each decision should be justified, and to enhance transparency and reproducibility, the code of the simulation study should be shared.

2.1.4. Similarities and Differences of Parametric and Plasmode Simulation

In the following, similarities and differences of parametric and Plasmode simulation according to Schreck et al. (2024) are discussed. Parametric and Plasmode simulation follow the same general structure but differ in the generation of covariate data. The DGP in parametric simulation has to be specified in advance, while no explicit specification is required for Plasmode. This enables parametric simulation to explore arbitrary scenarios, including extreme and rare scenarios, while Plasmode simulation is bound to the sample at hand. Moreover, Plasmode simulation relies heavily on the availability and representativeness of data, which is irrelevant for parametric simulation. On the other hand, parametric simulations may be unable to capture the complexity of real-world data, while Plasmodes are expected to resemble reality most accurately. For parametric simulation, complex dependencies become a challenge, while in Plasmode simulation, no modeling / estimation of the dependence structure is required. Parametric simulation for high-dimensional data usually becomes time- and cost-consuming, and latent dependencies can become problematic, while Plasmode simulation is mostly straightforward. Small sample size datasets are unproblematic with parametric simulations but might lead to difficulties in Plasmode simulations due to resampling.

The OGM needs to be specified in both cases. Parameters of the OGM can be estimated from data, derived from the literature, or set manually.

2.2. Quantifying Dataset Similarity

Methods for quantifying the similarity, or equivalently the distance, of two or more datasets can be used in numerous statistical applications. In the context of this thesis, the comparison of simulated and real data, as suggested by Burton et al. (2006) and Schreck et al. (2024), is the main application. Other applications involve *meta-learning* and *transfer learning*, which exploit the similarity between datasets to transfer knowledge between learning tasks for different datasets. Moreover, the generalizability of statistical models to new data relies on the similarity of the data used for fitting to the new datasets. In *two- or k-sample testing* the hypothesis of equal distributions

$$H_0 : F_1 = \cdots = F_k$$

is tested against the alternative

$$H_1 : \exists i \neq j \in \{1, \ldots, k\} : F_i \neq F_j$$

for $k = 2$ or $k \geqslant 2, k \in \mathbb{N}$, distributions $F_1, \ldots, F_k$, respectively. The test statistics of such tests can be seen as measures of similarity or distance of datasets that are drawn from the respective distributions $F_1, \ldots, F_k$. So, in all of these applications, a notion of how similar the datasets are to each other is required.

In the context of this thesis, methods for quantifying dataset similarity were selected from the literature according to the following three criteria:

1. The method is applicable to multivariate data.
2. The method does not require specific parametric distributional assumptions, e.g. a normal assumption.

3. The method does not focus on a specific characteristic of the distribution / dataset, like the mean or variance, but on the whole dataset / its whole distribution.

Methods that fulfill these criteria could, in general, be used to compare simulated to real-world datasets. The methods are grouped into classes based on their underlying ideas. Many of the methods compare the underlying distributions of the datasets rather than the datasets as a collection of points in space. This is also reflected in the classes. The general ideas of each class in the resulting taxonomy will be outlined in the following, accompanied by brief descriptions of example methods for demonstration. To further investigate the properties of the methods, several criteria for comparing the methods were developed in the scope of this thesis. These will be presented thereafter.

2.2.1. A Taxonomy of Methods for Quantifying Dataset Similarity

The methods for quantifying dataset similarity are divided into ten classes based on their underlying ideas. The resulting taxonomy of methods is not strict, as some methods fit into multiple classes. In that case, they are sorted into the class corresponding to their main idea. The full list of methods and method descriptions of all methods can be found in Stolte et al. (2024b). In the following, the main ideas of each class are sketched.

Comparison of Cumulative Distribution Functions, Density Functions or Characteristic Functions

Each of the *cumulative distribution function (CDF)*, *density function* (if it exists), and *characteristic function* fully characterizes a distribution. Thus, it is obvious to compare any of these when comparing distributions. In the univariate two-sample problem, methods comparing CDFs, like the Kolmogorov-Smirnov (KS) test for the equality of distributions, are popular. However, generalizing these methods to the multivariate case is not straightforward (Ramdas et al., 2017). Additionally, estimating the empirical CDFs, density functions, or characteristic functions gets harder with an increasing number of variables in the datasets. Nonetheless, there are some methods for multivariate data based on comparing the CDFs, density functions, or characteristic functions. For example, there are extensions of the KS test by permutation testing (Bickel, 1969) or by partitioning the multivariate sample space (Biau and Gyorfi, 2005). Another approach uses multivariate extensions of CDFs based on measure transportation (Boeckel et al., 2018). For comparing density functions, there are two main approaches. The first one is to partition the sample space to estimate the probability density function, similar to the idea of histograms for univariate data, e.g. by using classification trees (Ganti et al., 1999; Ntoutsi et al., 2008) or by probability binning (Roederer et al., 2001; H. Wang and Pei, 2005). The second approach is to compare kernel density estimates (Ahmad and Cerrito, 1993; Anderson et al., 1994; Cao and Keilegom, 2006). For comparing characteristic functions, there are different approaches for comparing empirical characteristic functions, e.g. using some metric or other notion of distance between the estimated functions (Alba-Fernández et al., 2004; Alba Fernández et al., 2008; Li et al., 2022).

Methods Based on Multivariate Ranks

Rank-based tests are popular for nonparametric univariate two-sample testing. There is no straightforward generalization to multivariate data possible since $\mathbb{R}^p$ has no natural

ordering. Therefore, defining multivariate ranks is not easy. One approach is projecting the data into $\mathbb{R}$, e.g. by using the direction vector of a linear classifier like a support vector machine (SVM) trained to distinguish between the two datasets and using univariate rank statistics on the projected data (Ghosh and Biswas, 2016). Other approaches generalize the concepts of ranks to multivariate data by using optimal transport (Ghosal and Sen, 2021; Deb et al., 2021) or graphs (Zhou and Chen, 2023).

Discrepancy Measures for Distributions

Discrepancy measures for distributions are often divided into *probability metrics* that fulfill all metric properties, i.e. they are positive definite, symmetric, and fulfill the triangle inequality, and *divergences* that fulfill some of the metric properties. Several subclasses of these two general classes are defined in the literature by certain properties, see e.g. Rachev (1991). Well-known examples of probability metrics are *Integral Probability Metrics (IPM)* as introduced by Müller (1997). These use the idea that if two distributions F_1 and F_2 are equal, the expectations of any function under both distributions are equal. Let $\mathcal{F}$ be a set of functions. Then the corresponding IPM is defined as the supremum distance of the expectations of functions from this class under both distributions

$$\mathrm{IPM}_{\mathcal{F}}(F_1, F_2) = \sup_{f \in \mathcal{F}} \left| \int f \, \mathrm{d}F_1 - \int f \, \mathrm{d}F_2 \right|.$$

The choice of the set of functions $\mathcal{F}$ determines the IPM.

Well-known examples of divergences are so-called *f-divergences* (Csiszár, 1964; Ali and Silvey, 1966), which use the idea that if two distributions F_1 and F_2 are equal, they assign the same likelihood to each point. Each f-divergence is determined by the choice of a convex, continuous function f that maps the likelihood ratio of one to the value of zero, $f(1) = 0$. The f-divergence is then defined as the expectation of this function f applied to the likelihood ratio under the first distribution F_1

$$D_f(F_1, F_2) = \int f\left(\frac{f_1}{f_2}\right) \mathrm{d}F_1.$$

One popular example for a f-divergence is the *Kullback-Leibler (KL) divergence*, which results from setting $f = \log$ (Kullback and Leibler, 1951).

Graph-Based Methods

Graph-based methods are popular in multivariate two- and k-sample testing. As pointed out by Arias-Castro and Pelletier (2016), many of these methods work by a similar procedure. First, a *similarity graph* is calculated on the pooled sample, e.g. the minimum spanning tree (MST) (Friedman and Rafsky, 1979; Chen and N. R. Zhang, 2013; Chen and Friedman, 2017; Chen et al., 2018; J. Zhang and Chen, 2022), the optimal non-bipartite matching (Rosenbaum, 2005; Petrie, 2016; Mukherjee et al., 2022), or the K-nearest neighbor (NN) graph (Schilling, 1986; Henze, 1988). Figure 2.3 shows the aforementioned examples of similarity graphs for two small example datasets. Next, for most graph-based methods, the number of edges connecting points from different samples, or the edges connecting points within each sample, is determined. The within-sample edges are indicated in gray in Figure 2.3, and the between-sample edges are indicated in green.

(a) Datasets drawn from the same distribution. (b) Datasets drawn from different distributions.

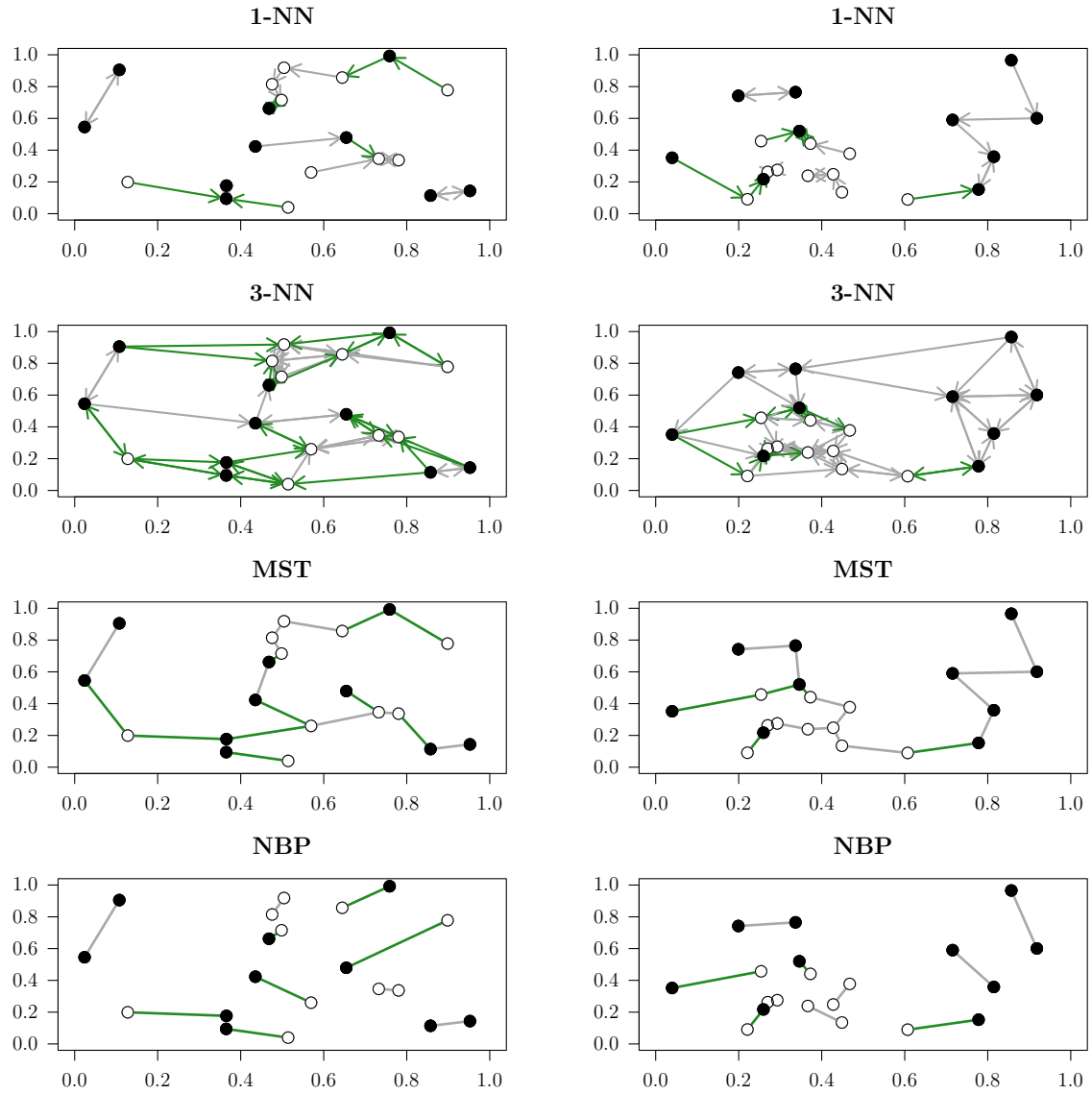


Figure 2.3.: Examples for similarity graphs calculated on the pooled sample for two datasets. White dots correspond to observations from the first dataset, and black dots to observations from the second dataset. The lines between points indicate undirected edges. Arrows indicate directed edges. Edges connecting points from the same sample are colored in gray, and edges connecting points from different samples are colored in green. 1-NN: nearest neighbor graph. 3-NN: 3-nearest neighbor graph. MST: minimum spanning tree. NBP: optimal non-bipartite pairing.

The idea is that if the underlying distributions coincide, the datasets are similar, and the number of between-sample edges is expected to be high. This is demonstrated in the left panel (Figure 2.3a), where many of the edges are colored green. Conversely, if the underlying distributions differ, the datasets are dissimilar, and the number of between-sample edges is expected to be low, as shown in the right panel (Figure 2.3b) with notably fewer green edges.

As already mentioned, the methods differ by the graph that is used. Moreover, they can differ in the calculation of the test statistic from the edge counts. For example, consider the following tree methods based on a minimum spanning tree. The method by Friedman

and Rafsky (1979) uses the between-sample edge count standardized with its mean and standard deviation under H_0. The method by Chen et al. (2018) uses the standardized weighted sum of the within-sample edge counts. Chen and Friedman (2017) use a squared Mahalanobis distance like transformation of the within-sample edge counts.

One particular problem for most methods in this class is that the similarity graph has to be unique for the test statistic to be well-defined. This might not be fulfilled when there are ties in the distance matrix of the pooled sample, which is especially problematic for categorical data. Solutions to this include using the average of the statistics on each of the optimal graphs or calculating the statistic on the union of all optimal graphs (Chen and N. R. Zhang, 2013; J. Zhang and Chen, 2022).

Methods Based on Inter-Point Distances

Under mild assumptions, the following two statements are equivalent (Maa et al., 1996):

1. $F_1 = F_2$.
2. The distributions of the *within-sample distances* $\left\|X^{(j)} - X'^{(j)}\right\|$, $j = 1, 2$, and the distribution of *between-sample distances* $\left\|X''^{(1)} - X''^{(2)}\right\|$ are equal, where $X^{(j)}$, $X'^{(j)}$, $X''^{(j)} \overset{\text{iid}}{\sim} F_j$, $j = 1, 2$, are independent copies.

Thus, some methods compare the three univariate distributions in 2. rather than directly the two multivariate distributions in 1. The idea is illustrated in Figure 2.4.

On the left (Figure 2.4a), two example datasets are drawn from the same distribution. As can be seen from the scatterplot at the top, the datasets are quite similar. Below, the distributions of the within- and between-sample distances are shown in histograms. The three distributions are very similar. On the right (Figure 2.4b), two example datasets are drawn from different distributions. As can be seen from the scatterplot at the top, lower values in both variables are frequently observed for the first dataset, while the points from the second dataset are more evenly distributed. Again, the distributions of the within- and between-sample distances are shown below. The distribution of within-sample distances for the first dataset is more right-skewed and more concentrated at low distance values than that of the within-sample distances for the second dataset. The distribution of between-sample distances is less skewed and has a higher range than that of the within-sample distances for the first dataset but is slightly more skewed than that of the within-sample distances for the second dataset. Therefore, the different distributions of the two datasets can easily be uncovered by comparing the distributions of the distances.

The best-known method based on this idea is the *energy statistic* (Székely and Rizzo, 2017) that compares twice the mean of the between-sample distances to the sum of the means of the within-sample distances

$$D_{\text{Energy}} = 2\,\mathbb{E}\left(\left\|X^{(1)} - X^{(2)}\right\|_2\right) - \mathbb{E}\left(\left\|X^{(1)} - X'^{(1)}\right\|_2\right) - \mathbb{E}\left(\left\|X^{(2)} - X'^{(2)}\right\|_2\right),$$

where $X^{(1)}, X'^{(1)} \overset{\text{iid}}{\sim} F_1, X^{(2)}$, $X'^{(2)} \overset{\text{iid}}{\sim} F_2$ and $\|\cdot\|_2$ denotes the Euclidean norm. It fulfills all metric properties.

Methods Based on Kernel (Mean) Embeddings

Kernel embeddings are commonly used in machine learning. For example, in support vector machines (Vapnik and Chervonenkis, 1974), they are employed to map data into a

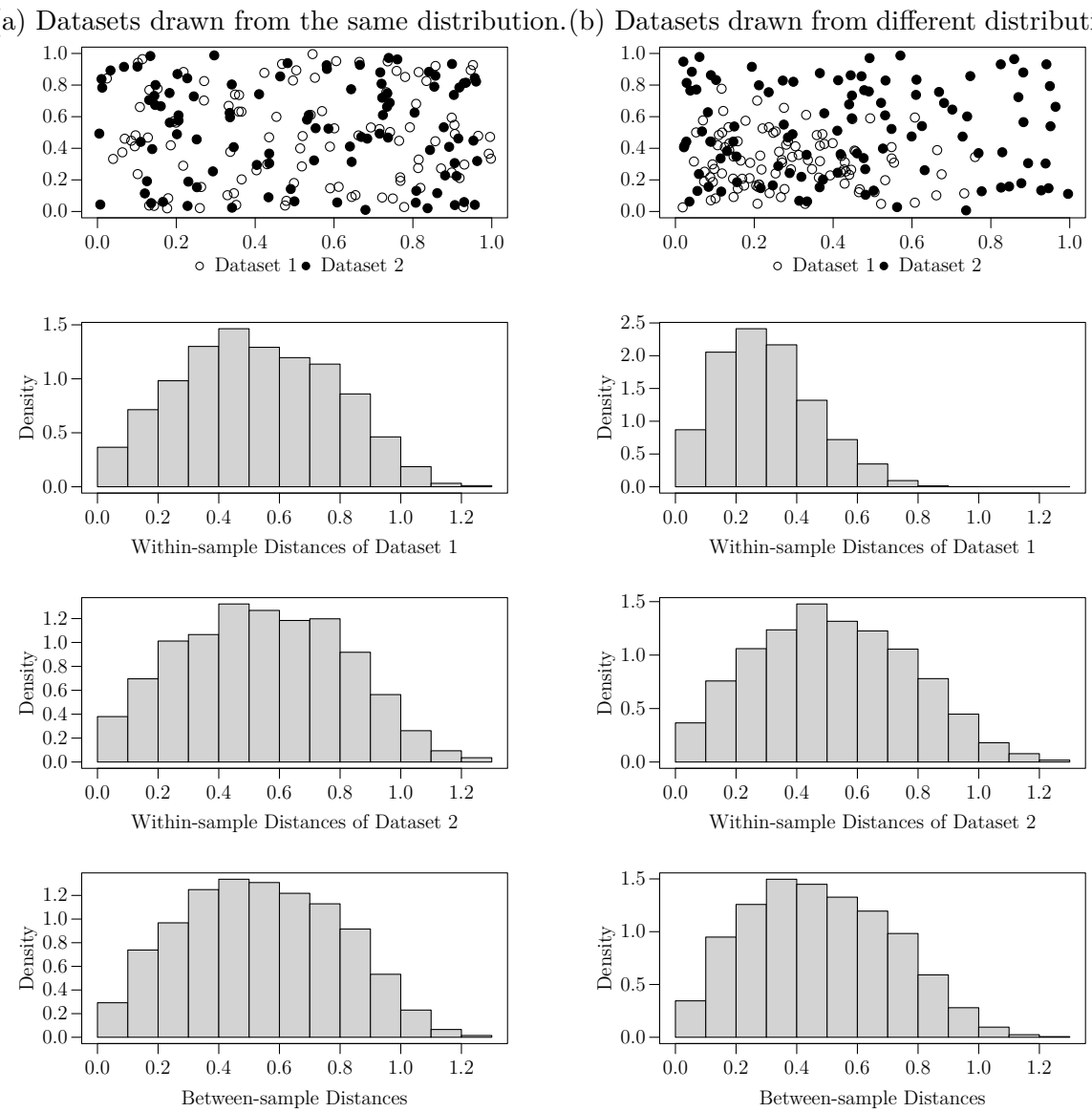


Figure 2.4.: Examples of distributions of between- and within-sample inter-point distances calculated on the pooled sample for two datasets. Scatterplots in the first row show the generated example datasets. Histograms in the other rows show the empirical distribution of the within- and between-sample distances for the example datasets.

higher-dimensional space using a *feature map* φ, which allows for better linear separation in that space if data is not linearly separable in the input space $\mathcal{X}$. Kernel mean embeddings extend this idea of feature maps to the space of probability distributions by representing each distribution F as a *mean function*

$$\varphi(F)(\cdot) = \mu_F(\cdot) := \int_{\mathcal{X}} K(x, \cdot)\, \mathrm{d}F(x) = \mathbb{E}_F(K(X, \cdot)), \tag{2.1}$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric and positive definite kernel function. A Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ that is endowed with a dot product $\langle \cdot, \cdot \rangle$ that satisfies the reproducing property

$$\langle f(\cdot), K(x, \cdot) \rangle = f(x) \;\Rightarrow\; \langle K(x, \cdot), K(x', \cdot) \rangle = K(x, x'),$$

is called a *reproducing kernel Hilbert space (RKHS)*. The reproducing property implies that the linear map from a function to its value at x can be viewed as an inner product. Under the assumption that the integral (2.1) exists, the kernel mean embedding as given above is essentially a transformation of the distribution F to an element in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ corresponding to the kernel K (Muandet et al., 2017). For characteristic kernels, the kernel mean representation captures all information about the distribution F, i.e. the map $F \mapsto \mu_F$ is injective, which implies

$$\|\mu_{F_1} - \mu_{F_2}\|_{\mathcal{H}} = 0 \Leftrightarrow F_1 = F_2$$

(Fukumizu et al., 2004; Sriperumbudur et al., 2008; Sriperumbudur et al., 2010). Consequently, kernel mean embeddings can be used for comparing distributions. This idea is depicted in Figure 2.5.

A metric for probability distributions based on this idea is the so-called *Maximum Mean Discrepancy (MMD)*

$$\mathrm{MMD}(\mathcal{H}, F_1, F_2) = \|\mu_{F_1} - \mu_{F_2}\|_{\mathcal{H}}. \tag{2.2}$$

It can be used in two-sample testing along with other applications (Gretton et al., 2006).

Methods Based on (Binary) Classification

The idea of the methods in this class is to use a classifier to distinguish between the datasets. For this, an augmented pooled dataset is created that includes a variable giving the dataset membership of each observation. Then, a classification method is trained using this dataset membership variable as the target variable. If the datasets differ, the classifier should be able to distinguish between them, while if the datasets are similar, the classifier should perform close to random guessing. Thus, a univariate two- or k-sample test can be applied to compare the scores output by the classifier, e.g., the predicted probabilities (Friedman, 2004), or the performance of the classifier, between the datasets. The univariate statistic applied to the scores can then be used as a test statistic (Yu et al., 2007; Lopez-Paz and Oquab, 2017; Hediger et al., 2021). Different classification methods can be used. Yu et al. (2007) use a decision tree, and Hediger et al. (2021) use a random forest. Lopez-Paz and Oquab (2017) leave the choice open and use a K-nearest neighbor classifier and neural nets in their applications. Typically, the classifier has to be trained on a separate dataset from the dataset on which the evaluation and test are performed. Thus, splitting the data into a training and test set is required. One exception is that when using a random forest classifier, the out-of-bag (OOB) data can be used for testing, whereby splitting the data can be prevented (Hediger et al., 2021).

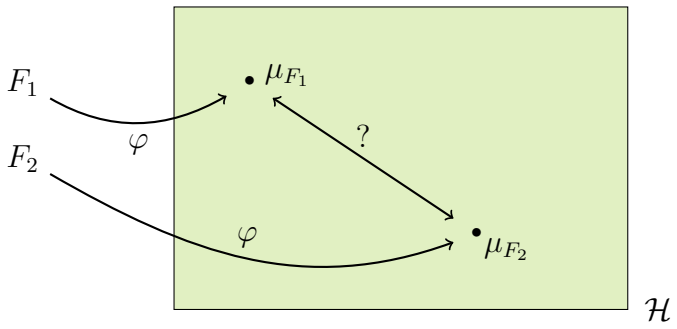


Figure 2.5.: Schematic idea of kernel mean embeddings. Distributions F_1, F_2 are mapped to mean functions μ_{F_1}, and μ_{F_2}, respectively, in the RKHS $\mathcal{H}$ using a feature map φ. The mean functions can then be compared in the RKHS.

Comparison Based on Summary Statistics

For the methods in this class, the datasets are reduced to summary statistics, and then these summary statistics are compared rather than the datasets directly. There are several proposals for summary statistics ranging from basic properties of the datasets, such as the number of variables, to complex statistics based on the distribution of the data. Often, the use of several statistics together is suggested (Johnson and Dasu, 1998; Tatti, 2007; Feurer et al., 2015).

Different Testing Approaches

Many of the methods in all classes are two- or k-sample tests. This class consists of additional two- or k-sample tests that do not fit into any of the other classes.

2.2.2. Criteria for Comparing Methods for Quantifying Dataset Similarity

For comparing the methods for quantifying dataset similarity, several criteria were developed in the scope of this thesis. These consist of desirable properties that such methods may have. Theoretical criteria are proposed that rate the applicability, interpretability, and theoretical properties of the method. Moreover, criteria to judge the performance of the methods in practice are defined. These aim to measure how sensitive the methods are to detecting certain differences between datasets, as well as the numeric stability and computational costs of the methods. The criteria will be briefly presented in the following.

Applicability

There are different aspects to the *applicability* of a method. First, a method should be applicable to datasets with different properties. The following aspects are considered here:

- Sensible inclusion of a target variable is possible as it is undesirable to treat a target variable (if present) like the covariates.
- Applicability to numeric data.
- Applicability to categorical data.
- Applicability to datasets of unequal sample sizes.
- Applicability to datasets with more variables than observations.
- Applicability to more than two datasets at the same time.

Further points that might enhance the applicability in practice but relate more to the nature of the method rather than to the data at hand are:

- No training data is required for applying the method, as data might be scarce in certain applications.
- No additional assumptions are required as these might be hard to check in practice.
- No parameters to choose / tune are required as the best parameter setting is often hard to determine.

- An implementation of the method is available as this enables direct use by practitioners.
- Low computational complexity.

The computational complexity is hard to rate in general and is therefore reported as it is. For all other criteria, it is determined for each method whether the respective criterion is fulfilled, fulfilled for certain parameter settings, unfulfilled, unknown, or inapplicable.

Interpretability

Interpretability of the method can be of great use for practitioners applying the method. Two aspects are considered here. The first aspect is whether a one-unit increase in the resulting value can be interpreted. This facilitates the general interpretation of the results. The second aspect is the boundedness of the output values. Lower and upper bounds to the values allow for assessing whether the observed similarity value can be considered as high or low similarity of the compared datasets.

Theoretical Properties

There are several theoretical properties deemed desirable in the literature on dataset similarity methods. First, there are *invariances* to certain transformations, i.e. the property that the dataset similarity does not change if both datasets are transformed accordingly. The invariances that are considered here include

- Rotation invariance,
- Location change invariance, and
- Homogeneous scale invariance.

Second, a lot of research regarding quantifying the similarity of distributions is concerned with developing (classes of) probability metrics or methods that at least fulfill some of the *metric properties*

- *Positive definiteness*, i.e. $d(F_1, F_2) = 0 \Leftrightarrow F_1 = F_2$,
- *Symmetry*, i.e. $d(F_1, F_2) = d(F_2, F_1)$, and
- *Triangle inequality*, i.e. $d(F1, F2) \leqslant d(F_1, F_3) + d(F_3, F_2)$,

where d denotes a dataset distance and F_1, F_2, F_3 denote probability distributions.

Lastly, many of the methods are proposed in the context of two- or k-sample testing. It is hard to operationalize test performance in a way that can be checked theoretically without performing extensive simulation studies, but one aspect that is often brought up in literature to promote newly proposed tests is *consistency*. Thus, additional criteria to rate methods that are proposed in the context of two- or k-sample testing are proofs of

- Consistency for $n_i \to \infty$, and
- Consistency for $p \to \infty$,

where $n_i, i = 1, \ldots, k$, denote the sample sizes and p denotes the number of variables that is assumed to be the same for all datasets as that is a requirement for almost all proposed methods.

Empirical Performance

For evaluating the empirical performance of dataset similarity methods, their ability to detect certain differences between datasets is of interest, e.g. differences in shift, scale, or correlation of the underlying distributions for numeric data or differences in the underlying class distributions for categorical data. As the ranges and distributions of the dataset similarity values for different methods differ drastically, the values cannot be compared directly. Typically, to this end, power comparisons are performed for two- or k-sample tests, i.e. the proportion of rejections of the null hypothesis of equal distributions is determined under certain alternatives in a simulation study.

The problem with this approach in the setting of this thesis is that not all methods define a test, and many methods that do define a test use a permutation test, which has a very high runtime and is, therefore, prohibitively expensive to run within a simulation study. The solution that was found for this challenge is based on the idea that tests typically reject the null hypothesis if the observed value of the test statistic is an extreme value with respect to the distribution of the test statistic under the null hypothesis and thus falls above (or below) a certain quantile of the null distribution corresponding to the test level. Therefore, typically, simulated power corresponds to the proportion of extreme values with respect to the null distribution.

Based on this observation, data is simulated under a specific setting in which the null hypothesis of equal distributions holds, e.g. both distributions are the multivariate standard normal distribution, to determine what are extreme values of the respective statistic under this null. Then, the proportion of values of the statistic that are more extreme than this threshold (*proportion of extreme simulation repetitions, PESR*) is determined in simulations under certain alternatives. As an example of one such alternative, one distribution could be the multivariate standard normal distribution, and the other could be a multivariate normal distribution with the same covariance matrix but a different mean vector. These proportions of extreme values can then be compared between the methods. Methods that result in high PESR values for small differences in the underlying distributions of the generated datasets show higher signals in detecting the difference between the datasets and can, therefore, be seen as preferable.

Computational Aspects

In addition to the ability to detect differences between datasets, numerical stability and low computational costs are desirable for methods to be used in practice. To rate these aspects, the following criteria are taken into account for the empirical comparison:

- Numerical stability, measured by the occurrence of missing or extreme values due to numerical problems.
- Computational resource consumption in terms of memory consumption and runtime.

3. Summary of the Articles

3.1. Article 1: Simulation Study to Evaluate when Plasmode Simulation is Superior to Parametric Simulation in Estimating the Mean Squared Error of the Least Squares Estimator in Linear Regression

As Schreck et al. (2024) pointed out, the impact of the choice of the *outcome-generating model (OGM)* and resampling scheme on Plasmode simulation has not been studied before. Moreover, the claim that Plasmode produces more realistic data than parametric simulation and therefore leads to better simulation results is not supported by any formal proof or empirical results. The aim here is to take a first step in closing this gap. The overall idea is that if the true data-generating mechanism, consisting of a *data-generating process (DGP)* for the covariates and an OGM, was known, parametric simulation using this would be the obvious choice. However, in reality, the true data-generating mechanism is unknown, so instead, assumptions are made. For parametric simulations, assumptions for both the DGP and the OGM have to be made. For Plasmode simulations, only assumptions for the OGM have to be made, but a representative sample from the true DGP of interest has to be available. The assumptions made on the data-generating mechanism in the simulation will most likely deviate from the truth. Thus, the goal is to evaluate how deviations of the assumptions from the truth affect the simulation results. More specifically, the goals are to find out the following.

1. How much can the DGP for a parametric simulation deviate from the truth before the performance of the parametric simulation study becomes worse than that of a corresponding Plasmode study for the same problem?
2. How do deviations of the chosen OGM from the true OGM affect both parametric and Plasmode simulations?
3. How does the choice of the resampling scheme affect the Plasmode simulation?

As the problem cannot be solved analytically, a simulation study is performed to evaluate these three points. In this first article (Stolte et al., 2024c), the aforementioned aspects are evaluated for the example of estimating the component-wise *mean squared error (MSE)*

$$\text{MSE}_j = \mathbb{E}\left[\left(\hat{\beta}_j - \beta_j\right)^2\right], j = 0, \ldots, p,$$

(Wackerly et al., 2014, p. 393) of the *least squares estimator* $\hat{\beta} \in \mathbb{R}^{p+1}$ in a *linear regression model*

$$Y = \tilde{X}\beta + \varepsilon,$$

where $Y \in \mathbb{R}^n$ denotes the vector of observations of an outcome variable, $\tilde{X} = [\mathbf{1}\ X] \in \mathbb{R}^{n\times(p+1)}$ denotes the *design matrix* with $X \in \mathbb{R}^{n\times p}$ denoting the covariate matrix. ε

with $\varepsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$, $i = 1, \ldots, n$, is the vector of error terms that are assumed to be homoscedastic and normally distributed with mean zero (Fahrmeir et al., 2013, p. 76). The analysis is restricted to lower-dimensional data with $n \in \{50, 100\}$ and $p \in \{2, 10, 50\}$, $p < n$, i.e. all combinations of these values for n and p except for $n = p = 50$ which is unidentifiable.

The overall idea of the simulation procedure is as follows. First, some true DGP and OGM are defined. Then, a parametric and Plasmode simulation study is performed as if this truth was unknown, i.e. with assumptions that might differ from the true DGP and OGM. Finally, the resulting estimated MSEs are compared to the MSEs under the true DGP and OGM to calculate the error made in the simulation studies with the deviating assumptions. This process is repeated to assess the variability that results from performing multiple simulation studies under the same assumptions. The resulting distributions of the errors in estimating the MSEs are compared between Plasmode and parametric simulations. In case of wrong assumptions made on the DGP in parametric simulation, this comparison is used to find out for which deviations parametric simulation becomes worse than Plasmode. The resulting distribution of errors in estimating the MSEs made in parametric or Plasmode simulation, respectively, with correct assumptions are compared to those made under deviating assumptions on the OGM to find out how wrongly specified OGMs affect the simulation results. Moreover, the resulting distributions of errors in estimating the MSEs are compared between different resampling schemes for Plasmode. The following resampling schemes are considered:

- *n-out-of-n Bootstrap* (Efron, 1979), i.e. drawing n observations with replacement,
- *m-out-of-n Bootstrap* (Götze, 1993; Bickel et al., 1997; Politis et al., 1999), i.e. drawing with replacement $m < n$ observations,
- *Subsampling*, i.e. drawing $m < n$ observations without replacement,
- *Smoothed bootstrap* (Efron, 1981; Silverman and Young, 1987; Hall et al., 1989; S. Wang, 1995), i.e. resampling from a smoothed empirical distribution,
- *Wild Bootstrap* (Wu, 1986), i.e. adding the standardized values of each variable scaled by a random number to the original variable, and
- No resampling, i.e. using the original dataset from the true DGP.

Due to the dependency of the MSE on the sample size, it is necessary to ensure that the simulation datasets after resampling have a fixed size n. For m-out-of-n Bootstrap and subsampling, different resampling rates are used as finding the optimal resampling rate using the algorithm of Bickel and Sakov (2008) is out of scope for the simulation study due to runtime. Different true data-generating mechanisms are used. For each of those, different misspecifications of the DGP and OGM are analyzed, e.g. differing with regard to the mean, variance, correlation, distribution family, or the coefficients of the linear model used as the OGM.

In short, the results of the study can be summarized as follows. When assuming the true DGP and OGM, as expected, parametric simulation outperforms Plasmode simulation in terms of median errors and variability of the simulated component-wise MSEs, especially for higher dimensional data and lower sample sizes. Subsampling with low resampling rates m/n can be recommended as the resampling scheme for Plasmodes. However, it requires more data to obtain a Plasmode dataset of fixed size. For a limited amount of available data, no resampling often outperforms Bootstrap with high resampling rates. Wild Bootstrap perform remarkably badly, such that it is excluded from most analyses.

For all tested types of misspecifications of the DGP in parametric simulation, e.g. mean, variance, correlation, or whole distribution family, at a certain severity of the misspecification, the errors made by parametric simulation exceed the ones for Plasmode simulation in which the DGP cannot be explicitly misspecified. Misspecifications of the OGM affect both parametric and Plasmode simulation studies equally. In case of misspecifications of the OGM, parametric therefore often slightly outperforms Plasmode due to its better baseline performance under the true OGM.

The study sets the groundwork for empirically comparing parametric and Plasmode simulation as there were no such studies available before. However, its scope is very limited as only one simplified example is considered.

3.2. Article 2: Simulation Study to Evaluate when Plasmode Simulation is Superior to Parametric Simulation in Comparing Classification Methods on High-Dimensional Data

The second article (Stolte et al., 2025c) builds on the first. It expands the empirical comparison of parametric and Plasmode simulation to the more realistic but also more complicated setting of comparing multiple binary classification methods, including Ridge (Schaefer et al., 1984) and LASSO (Tibshirani, 1996) logistic regression, support vector machines (SVM) (Vapnik and Chervonenkis, 1974), K-nearest neighbor (KNN) classifiers (Fix and Hodges, 1951; Cover and Hart, 1967), and random forests (Breiman, 2001). Moreover, higher-dimensional data ($p \in \{2, 10, 50, 150\}$, $n = 100$) and a more complicated true DGP with respect to the marginal distributions and the correlation structure are considered. The true OGM is a logistic model with coefficients that are sampled to also give a more complex model than, for example, setting all coefficients to a certain value. The classification performance is assessed by multiple performance measures, including accuracy, F_1-score, sensitivity, specificity, area under the receiver operating curve (AUC) (Olson and Delen, 2008, pp. 137, 144–146), and Brier score (Steyerberg, 2019, p. 279). The second study aims to:

1. Compare how well parametric and Plasmode simulation can estimate the classification performance and method ranking for the classification methods.
2. Evaluate how deviations from the true DGP and OGM affect parametric simulation in terms of estimating the classification performance and the ranking of the classification methods.
3. Evaluate how deviations from the true OGM and different resampling strategies affect Plasmode simulation in terms of estimating the classification performance and the ranking of the classification methods.
4. Find out how the aforementioned points are affected by the number of variables p.

Based on their bad performance in the preceding study, smoothed and wild Bootstrap are left out as resampling techniques for Plasmode. For the evaluation of simulation results, the relative error of one minus the measure value is used to penalize errors more strongly if the true performance is high, i.e. close to one. For the Brier score, usual relative errors are calculated since low values correspond to good performance. Based on these relative errors,

the number of acceptable simulation runs is evaluated. Since the results of a parametric simulation using the true DGP and OGM can be seen as the best that can be achieved, the range of acceptable errors is defined as the interval between the 2.5% and 97.5% quantile of these results for each true scenario. An acceptable simulation run is then defined as a run whose errors lie within this acceptable range. The proportion of acceptable simulation runs is used to summarize how well a simulation under certain assumptions performs. The simulated and true method rankings are compared using Kendall distances (Kendall, 1975), i.e. the number of required swaps of adjacent numbers in the ranking vector to transform the simulated into the true ranking.

The results of the second study mostly confirm those of the first study. However, there is no longer such a clear recommendation regarding the resampling scheme for Plasmode. One new aspect in the second study is the influence of the higher number of variables p. It is observed that the performance of Plasmode simulations decreases for an increasing number of variables p.

A lot of thought had to be put into designing the study, which was not as straightforward as for the first article. For example, it was originally planned to use each model estimated on a real dataset once as the true OGM to give no structural advantage to any of the models to ensure a fair comparison. However, this idea turned out to be impractical since the estimated models did not predict data that was well separated into two classes. Instead, most predicted probabilities were close to 0.5, so the separation of the target variable observations into 0 and 1 was very uncertain. This resulted in classification performances of the models trained on the generated data that were close to random guessing for all models, making the model comparison pointless. Most importantly, when all models perform equally badly, it is expected that a simulation study that generates data that is not well separated will always do a decent job at recreating the bad model performance. Therefore, a sensible comparison of different simulation studies requires the true classification performance to be reasonably good and ideally distinct between different classification models. Therefore, the logistic model, which allows to easily control the separation of the generated data was chosen instead. However, it was still not possible to keep the model coefficients constant for different numbers of variables p. This is a limitation of the study since it is unclear which effects can be attributed to increasing dimension and which to changes in the OGM. Nonetheless, the results for an increasing dimension are in line with the first study.

An additional challenge was to find a not-too-simple true DGP. For the first study, the true DGP was simple as the covariates were always generated from a multivariate normal with a diagonal or block diagonal correlation matrix. This presents a quite unrealistic example as, typically, the truth will be complex, and the simulation study will be a simplification that tries to approximate this truth, as pointed out before. So, for the second study, a more complex true DGP should be used. Therefore, different marginal distributions (normal, log-normal, bimodal, and contamination model) were used, and their parameters were drawn at random from certain distributions. The correlation matrix was also randomly generated. However, it turned out as a hard task to define the DGP this way such that it was still numerically possible to generate data from it. It required, for example, the use of `julia` (Bezanson et al., 2017) as a second programming language next to `R` (R Core Team, 2024) since no package available in `R` enables data generation in such a flexible way. Using only `julia` for the whole project was also infeasible due to prohibitively high runtimes for fitting and tuning the classification models. Therefore, this hybrid solution

with generating data in `julia` and performing the rest of the simulation study in `R` was chosen.

Overall, runtime was a limiting factor in the study as for a realistic comparison of the classification methods, in each iteration of the simulation studies, the hyperparameters of all classification methods were tuned. Due to the high runtime, some classification methods, such as boosting methods, were completely excluded. Moreover, the number of scenarios and repetitions per scenario had to be reduced, which presents another limitation to the study.

All in all, the second article still presents a valuable expansion of the groundwork on the comparison of parametric and Plasmode simulation studies and the influence of the OGM, resampling scheme, and dimensionality of the data on Plasmode simulation studies.

3.3. Article 3: Methods for Quantifying Dataset Similarity: A Review, Taxonomy and Comparison

As discussed in Chapter 2, quantifying the similarity of two or more datasets has many applications. Here, mainly the quantification of the similarity of simulated and real datasets is of interest. Other applications include the generalizability of statistical models to novel datasets, meta-learning, and transfer learning, as well as two- and k-sample testing. There are many methods for quantifying the similarity of two or more datasets proposed in the literature. In total, 118 methods were selected according to the inclusion criteria that they

- are applicable to multivariate data.
- do not require specific parametric or distributional assumptions.
- do not focus on a particular property of the data but on the entire dataset or its entire underlying distribution.

The third article (Stolte et al., 2024b) includes a review, taxonomy, and theoretical comparison of these methods. Its main goal is to give an overview of the available methods and to provide a comprehensive summary of the literature. The taxonomy divides the methods into ten classes based on their underlying principles. Short explanations of the classes and some example methods are given in Section 2.2.1 of this thesis. A theoretical comparison of the methods with regard to their applicability, interpretability, and theoretical properties is performed. For this, the corresponding 22 criteria presented in Section 2.1.4 that were developed specifically for this comparison within the scope of this thesis are evaluated for each method. Based on this evaluation, it is possible to narrow down the large pool of methods to a smaller set of suitable methods for the dataset comparison at hand. To facilitate this further, the result table of the evaluation of the criteria for each method is made available online in interactive form (`https://shiny.statistik.tu-dortmund.de/data-similarity/`). Users can select criteria that they need the method to fulfill by clicking. Moreover, the original article in which the method was presented, as well as the implementation, are linked (if available), and the methods can be sorted by the number of fulfilled criteria. The online table can serve as an extendable database for dataset similarity methods, as new methods can be added to the table upon request.

In the theoretical comparison, overall, graph-based methods perform particularly well, although there is a lot of heterogeneity within the classes from the taxonomy with respect

to method performance. The top-performing methods according to the highest number of fulfilled criteria are the following. The best-performing method fulfilling 16 of the 22 criteria is the *Kernel Measure of Multi-sample Dissimilarity (KMD)*, which is a kernel-based test using the association between the features and the sample membership to quantify the dissimilarity of multiple distributions (Huang and Sen, 2024). Next comes the *energy statistic* (Zech and Aslan, 2003), which was briefly introduced in Section 2.2 and fulfills 14 criteria and one conditionally. Then, there are three methods, each fulfilling 14 criteria and none conditionally. These are the graph-based k-sample test using non-bipartite optimal matchings by Mukherjee et al. (2022), the Rosenbaum *cross-match test* which is a two-sample test based on the optimal non-bipartite matching (Rosenbaum, 2005), and the *Friedman-Rafsky test* which is a two-sample test based on the minimum spanning tree (Friedman and Rafsky, 1979).

Overall, this article gives a comprehensive overview of the literature in the field and a first neutral comparison of methods for quantifying dataset similarity. Moreover, it contributes a taxonomy of dataset similarity methods and a set of theoretical criteria for rating such methods. The newly introduced criteria might also provide a starting point for improving existing methods or developing new methods to fulfill certain properties.

The main limitation is, however, that the comparison so far is only based on theoretical arguments and does not take into account the performance of the methods in practice. It can, therefore, answer the question of which methods are suitable, but the question of which of the suitable methods is the best remains open and is considered in the following.

3.4. Article 4: DataSimilarity: an R Package for Quantifying Similarity of Datasets and for Multivariate Two- and k-Sample Testing

Performing an empirical comparison of dataset similarity methods requires implementations of the relevant methods. Out of the 118 methods presented in the third article, only 34 were already implemented, and not all of these implementations could be used directly in practice. Ideally, one package containing all relevant methods, including promising methods that lacked an implementation before, would be used. Such a package should use the same input and output structure for all methods to facilitate the use of different methods. Moreover, it should include extensive documentation and should be maintained reliably such that occurring problems are fixed and the package remains available and usable over time. Unfortunately, no such package existed. Most available packages only include one or a few methods, which leads to strongly varying in- and output formats. The quality of the documentation and maintenance also varies heavily.

Thus, before performing an empirical comparison of the methods, implementations of new methods are required as well as unified in- and output formats for all available implementations. Since practitioners applying the dataset similarity methods can also profit from the additional method implementations, unified and simplified in- and output formats, and bundling of all available methods in one place, the implementations are published as an `R` (R Core Team, 2024) package named `DataSimilarity` (Stolte and Sauer, 2025) which is available on the Comprehensive R Archive Network (CRAN, R Foundation for Statistical Computing, 2025). It includes 14 implementations of methods that were not available in `R` before. In total, 36 methods are included. One aim of the

package was to provide extensive documentation of the implemented methods to facilitate the application for users who are not familiar with the methods. This includes, in addition to the necessary documentation of the in- and output parameters of each method, a short motivation and explanation of the method to give some background information, references to all relevant literature, and a clear description of the data types for which each method is intended.

The fourth article (Stolte et al., 2025b) is also part of that documentation as it contains an overview of all methods included in the package and their applicability, along with application examples for typical scenarios, method definitions, and implementation details for all methods. It also includes the general ideas of the implementation behind the package. A short version of the article is included in the package itself as a vignette. Briefly, all methods have two or more datasets as their first input arguments, followed by further method-specific inputs, e.g. the number of permutations for permutation tests. The output is an object class `htest`, which is commonly used for hypothesis testing in `R` and includes

- `statistic`: the test statistic
- `parameter` (optional): a parameter specifying the null distribution (e.g. degrees of freedom for a χ^2 distribution).
- `p.value`: the p value (if an asymptotic or permutation / Bootstrap test is performed, set to `NULL` otherwise).
- `estimate` (optional): the sample estimate(s) (e.g. the edge count for edge-count tests, `NULL` for many methods).
- `alternative`: the alternative hypothesis. For two datasets, this is $F_1 \neq F_2$, for k datasets it is $\exists i \neq j \in \{1, \dots, k\} : F_i \neq F_j$.
- `data.name`: names of the supplied datasets.
- further elements specific to the method (optional).

The use of the `htest` class ensures that the results are automatically printed in an appealing format. Moreover, this class is widely adopted for the output of hypothesis tests in `R` such that most users will already be used to working with objects of this class. Another general idea of the implementations is that the output consists of the results for exactly one test to incentivize that the test is chosen in advance rather than based on test results.

All in all, a well-documented `R` package containing all relevant methods in one place is provided that is implemented with unified and simple in- and output parameters. It is accompanied by an article giving an overview as well as the details of the implementation of the package.

The main limitation of the `R` package is that it is infeasible to implement all methods. Therefore, the most promising methods were chosen for the empirical comparison that follows in the next article. These are also the methods that are included in the package.

3.5. Article 5: An Empirical Comparison of Methods for Quantifying the Similarity of Categorical Datasets

Although there are many methods proposed in the literature for quantifying the similarity of two or more datasets, there are no neutral comparison studies in the sense of Boulesteix et al. (2013) for such methods. There are limited simulation studies (Székely and Rizzo, 2004; Gretton et al., 2012; Biswas et al., 2014; Biswas and Ghosh, 2014; Petrie, 2016; Chen and Friedman, 2017; Lopez-Paz and Oquab, 2017; Chen et al., 2018; Pan et al., 2018; Mukhopadhyay and K. Wang, 2020; Hediger et al., 2021; Li et al., 2022; Mukherjee et al., 2022; Song and Chen, 2022; Zaremba, 2022; Song and Chen, 2023; Huang and Sen, 2024) but all of these are performed in the context of proposing a new method and many are very limited in their scope both with regard to the included methods and with regard to the included data-generating mechanisms. The lack of neutral method comparisons is even worse for categorical datasets, as all of the studies mentioned above only included numeric data. Article 3 of this thesis can be seen as a neutral comparison of dataset similarity methods, but its main shortcoming is that it does not compare the methods' performances in practice. This article (Stolte et al., 2025a) starts to close this gap by performing an empirical comparison of dataset similarity methods for categorical data.

All methods that

- are implemented in `R`, or
- fulfill at least 11 of the criteria from Article 3 excluding consistency, or
- are the best within their subclass in Article 3, and no other method from that class was already selected by the first two criteria

are selected for comparison. These are exactly the methods included in the `DataSimilarity` package presented in Article 4. In Article 5, out of these methods, the ones applicable to two or more categorical datasets are compared since the comparison of methods for categorical data is most urgently needed, as explained above. The present article aims to:

1. Compare how well the methods can detect certain differences between datasets.
2. Compare how numerically stable and computationally demanding the methods are.
3. Find groups of methods that perform similarly across different scenarios and link these groups to the performance for certain differences between datasets.

The proportion of extreme simulation repetitions (PESR) introduced in Section 2.2.2 is used to investigate 1. The criteria introduced in Section 2.2.2 are used to examine 2. For 3., the PESR values are clustered. Six data-generating processes with two or four datasets are considered, consisting of binary or multinomial data. In the setting of equal distributions, uniform class probabilities are used for each variable in each dataset. Table 3.1 gives an overview of the considered alternative scenarios. For four samples, the number of differing datasets varies over all possibilities, i.e.:

a) One distribution differs from all others, which are equal, e.g. $F_1 = F_2 = F_3 \neq F_4$.

b) Two groups of two distributions each where the distributions within the groups are equal but the distributions between the groups differ, e.g. $F_1 = F_2 \neq F_3 = F_4$.

c) Two distributions are equal, and the other two distributions are different from these and each other, e.g. $F_1 = F_2 \neq F_3 \neq F_4, F_1 \neq F_4$.

d) All distributions differ, $F_i \neq F_j, i \neq j \in \{1, \ldots, 4\}$.

No. datasets	No. categories	Deviation
2	2	Unbalanced class probabilities in one dataset
2	5	Skewed class probability distribution in one dataset
2	5	Increase probability for one class and decrease probability of another ("1 up, 1 down") in one dataset
4	2	Unbalanced class probabilities in one to three datasets
4	5	Skewed class probability distribution in one to three datasets
4	5	Increase probability for one class and decrease probability of another ("1 up, 1 down") in one to three datasets

Table 3.1.: Overview of simulation scenarios.

In all cases, the number of variables p and overall sample size N are varied as well as the balance of the sample sizes of the individual datasets.

With the above-mentioned selection criteria, ten methods are selected for the two-sample case. Due to the lack of clear recommendations regarding parameter settings, most of these methods are used in multiple variants. In total, 56 variants of the ten methods are considered. In the multi-sample case, three methods in four variants are applied. Some variants are consistently inferior to others in the two-sample case such that those variants could be further restricted.

Overall, good performance is observed for the edge count tests (Friedman and Rafsky, 1979; Chen and Friedman, 2017; Chen et al., 2018; J. Zhang and Chen, 2022) in all scenarios. The constrained minimum (CM) distance (Tatti, 2007) performs even better in many scenarios except for the "1 up, 1 down" or for unbalanced sample sizes. The random forest-based test HMN (Hediger et al., 2021) is competitive for balanced sample sizes but does not work for unbalanced sample sizes. The tests based on the optimal non-bipartite matching (Petrie, 2016; Mukherjee et al., 2022) and the classifier-based tests (Yu et al., 2007; Lopez-Paz and Oquab, 2017) are typically inferior. Generally speaking, the CM distance and the optimal non-bipartite matching-based tests perform better for the "skewed" alternative than for the "1 up, 1 down" alternative in the multinomial case, and vice versa for the classifier-based tests.

For the multi-sample case, only the MMCM and Petrie's test based on the optimal non-bipartite matching and the classifier-based test C2ST (Lopez-Paz and Oquab, 2017) are applicable. The results, in that case, are similar to the ones for the two-sample case. The MMCM and Petrie's test perform better for binary data and the multinomial "skewed" alternative, while the classifier-based tests perform better for the multinomial "1 up, 1 down" alternative. A possible reason for this could be the choice of distance functions and coding. Moreover, some of the graph-based methods do not work for low numbers of variables ($p = 2$) due to the low number of possible observations. This issue is also discussed in the article.

All in all, the article presents helpful results that can guide the choice of a suitable dataset similarity method for categorical data. It contributes the first neutral empirical comparison of such methods. The cases of categorical datasets that include a target variable and numeric data are left for further research.

4. Discussion and Outlook

Simulation studies are a crucial tool in statistical method evaluation and comparison. The generation of realistic data is often seen as a prerequisite for the simulation study to produce reliable results (Burton et al., 2006; Boulesteix et al., 2013; Boulesteix et al., 2020; Schreck et al., 2024). This generation of realistic data is frequently doubted for parametric simulation, i.e. simulation studies in which the data is created entirely using pseudo-random number generation according to a user-specified data-generating process (DGP) for the covariates and outcome-generating model (OGM) for the target variable (Vaughan et al., 2009; Boulesteix et al., 2020; Schreck et al., 2024). The need to fully specify the DGP and OGM makes this type of simulation prone to oversimplification, especially for high-dimensional and complex data. An alternative that is often claimed to produce more realistic data are statistical Plasmode simulations (Cattell and Jaspers, 1967; Schreck et al., 2024), where the covariate data is generated by resampling from a real-world dataset. The target variable is generated according to a user-specified OGM as in parametric simulation, which enables assessing quantities like bias or power that require the knowledge of the true parameters of the OGM. A discussion of parametric and Plasmode simulation by Schreck et al. (2024), however, identified a lack of studies that verify the claim that Plasmode simulation is superior to parametric simulation due to the more realistic data generation. The authors pointed out that, in particular, the influence of the choices of the resampling scheme and OGM on the quality of Plasmode simulation studies is unclear. Moreover, they identify the need for methods to quantify the similarity of the Plasmode datasets and the real-world dataset used for resampling.

Motivated by this, Article 1 (Stolte et al., 2024c) and 2 (Stolte et al., 2025c) of this thesis deal with the empirical comparison of parametric and Plasmode simulation. The idea in these studies was that if the true DGP and OGM were known, parametric simulation using these as the data-generating mechanism would be the obvious best choice. However, since the true data-generating mechanisms are unknown in reality, researchers have to make assumptions instead, which will most likely deviate from the truth. Thus, a true DGP and OGM were chosen, and the results obtained by parametric and Plasmode simulations under different assumptions were compared to those given the true DGP and OGM. The influence of deviations of the assumptions from the truth on the difference between the simulation results under the respective assumptions and the results determined under the true DGP and OGM was then investigated. In Article 1, the estimation of the mean squared error (MSE) of the least squares (LS) estimator in linear regression was considered. In Article 2, the more complex comparison of multiple binary classification methods (random forest, support vector machine, LASSO / Ridge logistic regression, K-nearest neighbors) using different classification performance measures (accuracy, AUC, Brier score, sensitivity, specificity, F_1 score) was investigated. The results showed that for large enough misspecifications of the DGP in parametric simulation, the results of a Plasmode simulation study are closer to the truth than those of the parametric simulation. Moreover, it could be observed that deviations of the assumed OGM from the truth affect parametric and Plasmode simulation equally. With regard to the resampling scheme, an

impact of the resampling scheme on the performance of Plasmode simulation was visible in both studies. In the setting of the first study, subsampling with low resampling rates could be recommended. In the setting of the second study, the results were less clear.

Overall, the studies contributed an important first step in rigorously comparing parametric and Plasmode simulation and in investigating the influence of the choice of the OGM and resampling strategy on Plasmode simulation. It could be demonstrated that Plasmode simulation can be a preferable alternative to parametric simulation in cases where the specification of the DGP is uncertain and that the choices of the OGM and resampling strategy for Plasmode simulation do in fact influence the quality of the simulation results. The studies were however limited in the considered scenarios. Especially, the second study was limited with respect to the size of the datasets due to the high runtimes of the considered methods. Within the two studies, the deviations were quantified by differences in certain parameters of the distributions chosen for the DGP or of the OGM. As the true parameter values are unknown in practice, such a quantification is infeasible for real-world studies. A possible solution would be to instead use a dataset similarity measure within such studies to quantify how similar the simulated data under certain assumptions is to data generated according to the true DGP and OGM. Then, researchers performing simulation studies could use the same dataset similarity measure to compare their generated data to a real-world dataset from the data-generating mechanism of interest. By comparing their similarity value to the similarity values from the comparison of Plasmode and parametric simulation they could assess if their simulated data is close enough to the real-world data to expect meaningful simulation results. This evaluation of parametric and Plasmode simulation with regard to dataset similarity is left open for further research. It first requires the choice of a suitable dataset similarity method which was not clear at the time at which the studies in Article 1 and 2 were conducted. To find out which dataset similarity method to use, an extensive literature search was performed. This revealed many proposed methods but only a few comparison studies that are limited in their scope and not neutral in the sense of Boulesteix et al. (2020) since all of them are conducted in the context of proposing a new method. Therefore, the second goal of this thesis was to compare these methods to find out which method to use.

Article 3 (Stolte et al., 2024b) contributed a taxonomy of dataset similarity methods based on underlying ideas. Moreover, 22 theoretical criteria were introduced, which can be used to rate the applicability, interpretability, and theoretical properties. The results of this rating for 118 methods are presented in an interactive online table (`https://shiny.statistik.tu-dortmund.de/data-similarity/`). This enables practitioners to easily reduce the plethora of methods to a smaller set of methods that are suitable for the method comparison at hand.

Despite the high number of considered methods, the comparison was still limited with regard to the choice of methods. Univariate methods, methods for certain distribution families, and methods for certain parameters of the distributions, like the means, were excluded as these are not of general interest for the comparison of simulated and real-world datasets. Moreover, all methods that could be found in the literature are only applicable to datasets with the same number of variables which might limit the applicability. The most severe limitation of Article 3 is however that no performance comparison of the methods was performed, so only the question of which methods are suitable is answered but not the question of which of the suitable methods is the best, which is highly relevant in practice. Answering which dataset similarity method performs best, however, requires

extensive and neutral simulation studies. This was out of the scope of Article 3 but was tackled later in Article 5.

Performing a comparison study of dataset similarity methods heavily relies on the availability of unified implementations of all relevant methods. Many of these methods were not implemented yet, and the existing implementations were distributed over many different `R` (R Core Team, 2024) packages, resulting in widely differing in- and output formats. These issues were tackled by implementing the `DataSimilarity` `R` package (Stolte and Sauer, 2025) that includes new implementations of methods for which an implementation was missing before, as well as wrapper functions that unify and simplify the in- and output formats of the already existing implementations. Article 4 (Stolte et al., 2025b) describes this package and includes descriptions of all implemented methods, the main ideas behind the implementations, and illustrations of how to use the package. It is a part of the comprehensive documentation of the package.

The package is the first that includes many relevant methods in one place and allows for applying them in a unified framework. It was, however, infeasible to implement all 118 methods, so here the focus was on the ones that performed well with respect to the theoretical criteria in Article 3.

With the implementations of the `DataSimilarity` package, it was finally possible to empirically compare the dataset similarity methods presented in Article 3. Article 5 (Stolte et al., 2025a) includes a neutral comparison study of methods applicable to categorical datasets that do not include any target variable, which would need to be treated differently from the covariates. The ability of different methods to detect differences in the class probabilities of the categories between datasets for binary data and categorical data with five categories was investigated. Moreover, computational aspects like runtime and memory consumption were taken into account. Overall, 10 methods in more than 50 variants were considered for the two-sample case and three methods in four variants in the multi-sample case. Based on the simulation results, guidance on which methods can detect which kinds of deviations between datasets could be provided.

Article 5 is the first contribution to closing the gap in comparing dataset similarity methods. Clear tendencies for which methods work best for detecting certain differences between categorical datasets could be observed. However, the results of Article 5 might be explained in part by the different default choices of distance functions that are used in the calculation of the similarity or distance according to different methods. The implementations in the `DataSimilarity` package (Stolte and Sauer, 2025) were revised accordingly to allow for more flexible choices of distance functions. The method comparison of Article 5 was limited in the number of considered methods, especially with respect to the considered scenarios. The comparison of methods applicable to numeric data was left open for future research. Many methods are proposed as test statistics for two- or k-sample tests. In Article 5, no real power of these tests could be assessed as often a permutation / Bootstrap test is proposed that has a prohibitively high runtime to be performed within a simulation study. Moreover, not all considered methods define tests, but a unified performance measure that is also applicable to these methods was needed. The assessment of real power might, however, be of interest in the context of testing since the considered performance measure in Article 5 is often close to power but might indicate high performance even in cases when the empirical null distribution of the test statistic does not match the theoretical one.

In summary, the articles included in this thesis contributed a first empirical comparison of parametric and Plasmode simulation. In this context, methods for quantifying the sim-

ilarity of two or more datasets would be of great use. A literature search for such methods revealed a plethora of methods but a lack of guidance on which methods to prefer in which situations. Therefore, a taxonomy and theoretical comparison of such methods were provided, which can guide the choice of suitable methods for a given dataset comparison. To further advance in the comparison of dataset similarity methods, a neutral comparison study of methods for categorical data was performed. In the process of implementing this study, an `R` package of the most relevant dataset similarity methods in a unified framework was published.

Further research is needed to address some of the main limitations of the presented articles. The neutral comparison of dataset similarity methods applicable to categorical datasets that include a target variable is still pending, as well as that of methods applicable to numeric datasets. Moreover, the theoretical properties used for two- and k-sample testing like the agreement of the empirical and (asymptotic) theoretical null distribution as well as the asymptotic power could be investigated based on the available simulation results for methods that define an (asymptotic) test. When the comparison of dataset similarity methods is completed, the best-performing methods could finally be included in the comparison of parametric and Plasmode simulation. One particularly interesting special case would be the simulation of gene expression data which is especially challenging due to the high dimensionality and complex dependence structures. This would be of high relevance for the evaluation of methods in application areas like the prediction of hepatotoxicity (Stolte et al., 2023; Albrecht et al., 2025). Moreover, the similarity between datasets could be used in other applications like selecting virtual control groups to reduce the number of animals needed in animal experiments (Steger-Hartmann et al., 2025) or for assessing the quality of data imputation (Thurow et al., 2021).

So, all in all, this thesis presents a valuable contribution to the comparison of simulation strategies and the quantification of dataset similarity and provides the basis for tailored further research.

Bibliography

Ahmad, I. A. and Cerrito, P. B. (1993): "Goodness of fit tests based on the L_2-norm of multivariate probability density functions", in: *Journal of Nonparametric Statistics* 2 (2), pp. 169–181, DOI: 10.1080/10485259308832550.

Alba Fernández, V., Jiménez Gamero, M. D., and Muñoz García, J. (2008): "A test for the two-sample problem based on empirical characteristic functions", in: *Computational Statistics & Data Analysis* 52 (7), pp. 3730–3748, DOI: 10.1016/j.csda.2007.12.013.

Alba-Fernández, V., Ibáñez-Pérez, M. J., and Jiménez-Gamero, M. D. (2004): "A bootstrap algorithm for the two-sample problem using trigonometric Hermite spline interpolation", in: *Communications in Nonlinear Science and Numerical Simulation*, Recent Advances in Computational and Mathematical Methods for Science and Engineering 9 (2), pp. 275–286, DOI: 10.1016/S1007-5704(03)00117-5.

Albrecht, W., Brecklinghaus, T., Stolte, M., Kappenberg, F., Gründler, L., Chen, P., Cadenas, C., Damm, G., Edlund, K., Ghallab, A., Marchan, R., Nell, P., Reinders, J., Seehofer, D., Behr, A.-C., Braeuning, A., van Thriel, C., Gardner, I., Rahnenführer, J., and Hengstler, J. G. (2025): "Improved identification of human hepatotoxic potential by summary variables of gene expression", in: *ALTEX - Alternatives to animal experimentation*, DOI: 10.14573/altex.2403272.

Ali, S. M. and Silvey, S. D. (1966): "A General Class of Coefficients of Divergence of One Distribution from Another", in: *Journal of the Royal Statistical Society: Series B (Methodological)* 28 (1), pp. 131–142, DOI: 10.1111/j.2517-6161.1966.tb00626.x.

Anderson, N. H., Hall, P., and Titterington, D. M. (1994): "Two-Sample Test Statistics for Measuring Discrepancies Between Two Multivariate Probability Density Functions Using Kernel-Based Density Estimates", in: *Journal of Multivariate Analysis* 50 (1), pp. 41–54, DOI: 10.1006/jmva.1994.1033.

Andrews, D. W. K. and Buchinsky, M. (2000): "A Three-step Method for Choosing the Number of Bootstrap Repetitions", in: *Econometrica* 68 (1), pp. 23–51, DOI: 10.1111/1468-0262.00092.

Arias-Castro, E. and Pelletier, B. (2016): "On the consistency of the crossmatch test", in: *Journal of Statistical Planning and Inference* 171, pp. 184–190, DOI: 10.1016/j.jspi.2015.10.003.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017): "Julia: A fresh approach to numerical computing", in: *SIAM review* 59 (1), pp. 65–98.

Biau, G. and Gyorfi, L. (2005): "On the asymptotic properties of a nonparametric L_1-test statistic of homogeneity", in: *IEEE Transactions on Information Theory* 51 (11), pp. 3965–3973, DOI: 10.1109/TIT.2005.856979.

Bickel, P. J. (1969): "A Distribution Free Version of the Smirnov Two Sample Test in the p-Variate Case", in: *The Annals of Mathematical Statistics* 40 (1), pp. 1–23.

Bickel, P. J., Götze, F., and Zwet, W. R. van (1997): "Resampling Fewer Than n Observations: Gains, Losses, and Remedies for Losses", in: *Statistica Sinica* 7 (1), pp. 1–31.

Bickel, P. J. and Sakov, A. (2008): "On the Choice of m in the m Out of n Bootstrap and Confidence Bounds for Extrema", in: *Statistica Sinica* 18.

Biswas, M. and Ghosh, A. K. (2014): "A nonparametric two-sample test applicable to high dimensional data", in: *Journal of Multivariate Analysis* 123, pp. 160–171, DOI: 10.1016/j.jmva.2013.09.004.

Biswas, M., Mukhopadhyay, M., and Ghosh, A. K. (2014): "A distribution-free two-sample run test applicable to high-dimensional data", in: *Biometrika* 101 (4), pp. 913–926, DOI: 10.1093/biomet/asu045.

Boeckel, M., Spokoiny, V., and Suvorikova, A. (2018): *Multivariate Brenier cumulative distribution functions and their application to non-parametric testing*, arXiv:1809.04090 [math, stat], DOI: 10.48550/arXiv.1809.04090.

Boulesteix, A.-L., Groenwold, R. H., Abrahamowicz, M., Binder, H., Briel, M., Hornung, R., Morris, T. P., Rahnenführer, J., and Sauerbrei, W. (2020): "Introduction to statistical simulations in health research", in: *BMJ Open* 10 (12), e039921, DOI: 10.1136/bmjopen-2020-039921.

Boulesteix, A.-L., Lauer, S., and Eugster, M. J. A. (2013): "A Plea for Neutral Comparison Studies in Computational Sciences", in: *PLOS ONE* 8 (4), e61562, DOI: 10.1371/journal.pone.0061562.

Breiman, L. (2001): "Random Forests", in: *Machine Learning* 45 (1), pp. 5–32, DOI: 10.1023/A:1010933404324.

Burton, A., Altman, D. G., Royston, P., and Holder, R. L. (2006): "The design of simulation studies in medical statistics", in: *Statistics in Medicine* 25 (24), pp. 4279–4292, DOI: 10.1002/sim.2673.

Cao, R. and Keilegom, I. van (2006): "Empirical likelihood tests for two-sample problems via nonparametric density estimation", in: *Canadian Journal of Statistics* 34 (1), pp. 61–77, DOI: 10.1002/cjs.5550340106.

Cattell, R. B. and Jaspers, J. (1967): "A general plasmode (No. 30-10-5-2) for factor analytic exercises and research.", in: *Multivariate behavioral research monographs*, Publisher: Society of Multivariate Experimental Psychology.

Chen, H., Chen, X., and Su, Y. (2018): "A Weighted Edge-Count Two-Sample Test for Multivariate and Object Data", in: *Journal of the American Statistical Association* 113 (523), pp. 1146–1155, DOI: 10.1080/01621459.2017.1307757.

Chen, H. and Friedman, J. H. (2017): "A New Graph-Based Two-Sample Test for Multivariate and Object Data", in: *Journal of the American Statistical Association* 112 (517), pp. 397–409, DOI: 10.1080/01621459.2016.1147356.

Chen, H. and Zhang, N. R. (2013): "Graph-Based Tests for Two-Sample Comparisons of Categorical Data", in: *Statistica Sinica* 23 (4), pp. 1479–1503.

Chipman, H. and Bingham, D. (2022): "Let's practice what we preach: Planning and interpreting simulation studies with design and analysis of experiments", in: *Canadian Journal of Statistics* 50 (4), pp. 1228–1249, DOI: 10.1002/cjs.11719.

Cover, T. and Hart, P. (1967): "Nearest neighbor pattern classification", in: *IEEE Transactions on Information Theory* 13 (1), Conference Name: IEEE Transactions on Information Theory, pp. 21–27, DOI: 10.1109/TIT.1967.1053964.

Csiszár, I. (1964): "Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten", in: *Magyer Tud. Akad. Mat. Kutato Int. Koezl.* 8, pp. 85–108.

Davidson, R. and MacKinnon, J. G. (2000): "Bootstrap tests: how many bootstraps?", in: *Econometric Reviews* 19 (1), pp. 55–68, DOI: 10.1080/07474930008800459.

Deb, N., Bhattacharya, B. B., and Sen, B. (2021): *Efficiency Lower Bounds for Distribution-Free Hotelling-Type Two-Sample Tests Based on Optimal Transport*, arXiv:2104.01986 [math, stat], DOI: 10.48550/arXiv.2104.01986.

Efron, B. (1979): "Bootstrap Methods: Another Look at the Jackknife", in: *The Annals of Statistics* 7 (1), pp. 1–26, DOI: 10.1214/aos/1176344552.

Efron, B. (1981): "Nonparametric Estimates of Standard Error: The Jackknife, the Bootstrap and Other Methods", in: *Biometrika* 68 (3), pp. 589–599, DOI: 10.2307/2335441.

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. (2013): *Regression: Models, Methods and Applications*, Springer, Berlin, Heidelberg, DOI: 10.1007/978-3-642-34333-9_1.

Feurer, M., Springenberg, J., and Hutter, F. (2015): "Initializing Bayesian Hyperparameter Optimization via Meta-Learning", in: *Proceedings of the AAAI Conference on Artificial Intelligence* 29 (1), DOI: 10.1609/aaai.v29i1.9354.

Fix, E. and Hodges, J. L. (1951): *Discriminatory analysis: nonparametric discrimination, consistency properties*, tech. rep., USAF school of Aviation Medicine.

Friedman, J. H. (2004): *On Multivariate Goodness-of-Fit and Two-Sample Testing*, tech. rep., SLAC National Accelerator Lab., Menlo Park, CA (United States).

Friedman, J. H. and Rafsky, L. C. (1979): "Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests", in: *The Annals of Statistics* 7 (4), pp. 697–717.

Friedrich, S. and Friede, T. (2024): "On the role of benchmarking data sets and simulations in method comparison studies", in: *Biometrical Journal* 66 (1), p. 2200212, DOI: 10.1002/bimj.202200212.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2004): "Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces", in: *Journal of Machine Learning Research* 5 (Jan), pp. 73–99.

Ganti, V., Gehrke, J., Ramakrishnan, R., and Loh, W.-Y. (1999): "A Framework for Measuring Changes in Data Characteristics", in: *Proceedings of the 18th Symposium on Principles of Database Systems*, pp. 126–137.

Ghosal, P. and Sen, B. (2021): *Multivariate Ranks and Quantiles using Optimal Transport: Consistency, Rates, and Nonparametric Testing*, arXiv:1905.05340 [math, stat], DOI: 10.48550/arXiv.1905.05340.

Ghosh, A. K. and Biswas, M. (2016): "Distribution-free high-dimensional two-sample tests based on discriminating hyperplanes", in: *TEST* 25 (3), pp. 525–547, DOI: 10.1007/s11749-015-0467-x.

Götze, F. (1993): "Asymptotic approximation and the bootstrap", in: *IMS Bulletin*, p. 305.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2012): "A Kernel Two-Sample Test", in: *Journal of Machine Learning Research* 13, pp. 723–773.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2006): "A Kernel Method for the Two-Sample-Problem", in: *Advances in Neural Information Processing Systems*, vol. 19, MIT Press.

Hall, P., DiCiccio, T. J., and Romano, J. P. (1989): "On Smoothing and the Bootstrap", in: *The Annals of Statistics* 17 (2), pp. 692–704, DOI: 10.1214/aos/1176347135.

Hediger, S., Michel, L., and Näf, J. (2021): *On the Use of Random Forest for Two-Sample Testing*, arXiv:1903.06287 [stat], DOI: 10.48550/arXiv.1903.06287.

Henze, N. (1988): "A Multivariate Two-Sample Test Based on the Number of Nearest Neighbor Type Coincidences", in: *The Annals of Statistics* 16 (2), pp. 772–783.

Huang, Z. and Sen, B. (2024): "A Kernel Measure of Dissimilarity between M Distributions", in: *Journal of the American Statistical Association* 119 (548), pp. 3020–3032, DOI: 10.1080/01621459.2023.2298036.

Johnson, T. and Dasu, T. (1998): "Comparing Massive High-Dimensional Data Sets.", in: *KDD*, pp. 229–233.

Kendall, M. (1975): *Rank correlation methods*, 4th ed., Rank correlation methods, Griffin, Oxford, England.

Kullback, S. and Leibler, R. A. (1951): "On Information and Sufficiency", in: *The Annals of Mathematical Statistics* 22 (1), pp. 79–86, DOI: 10.1214/aoms/1177729694.

Li, X., Hu, W., and Zhang, B. (2022): "Measuring and testing homogeneity of distributions by characteristic distance", in: *Statistical Papers*, DOI: 10.1007/s00362-022-01327-7.

Lopez-Paz, D. and Oquab, M. (2017): "Revisiting Classifier Two-Sample Tests", in: *International Conference on Learning Representations*.

Maa, J.-F., Pearl, D. K., and Bartoszyński, R. (1996): "Reducing multidimensional two-sample data to one-dimensional interpoint comparisons", in: *The Annals of Statistics* 24 (3), pp. 1069–1074, DOI: 10.1214/aos/1032526956.

McCullagh, P. and Nelder, J. A. (1989): *Generalized Linear Models*, 2nd ed., Chapman & Hall, London.

Mehta, T. S., Tanik, M., and Allison, D. B. (2004): "Towards sound epistemological foundations of statistical methods for high-dimensional biology", in: *Nature Genetics* 36 (9), pp. 943–947, DOI: 10.1038/ng1422.

Mehta, T. S., Zakharkin, S. O., Gadbury, G. L., and Allison, D. B. (2006): "Epistemological issues in omics and high-dimensional biology: give the people what they want", in: *Physiological Genomics* 28 (1), pp. 24–32, DOI: 10.1152/physiolgenomics.00095.2006.

Morris, T. P., White, I. R., and Crowther, M. J. (2019): "Using simulation studies to evaluate statistical methods", in: *Statistics in Medicine* 38 (11), pp. 2074–2102, DOI: 10.1002/sim.8086.

Muandet, K., Fukumizu, K., Sriperumbudur, B., and Schölkopf, B. (2017): "Kernel Mean Embedding of Distributions: A Review and Beyond", in: *Foundations and Trends® in Machine Learning* 10 (1-2), pp. 1–141, DOI: 10.1561/2200000060.

Mukherjee, S., Agarwal, D., Zhang, N. R., and Bhattacharya, B. B. (2022): "Distribution-Free Multisample Tests Based on Optimal Matchings With Applications to Single Cell Genomics", in: *Journal of the American Statistical Association* 117 (538), pp. 627–638, DOI: 10.1080/01621459.2020.1791131.

Mukhopadhyay, S. and Wang, K. (2020): "A nonparametric approach to high-dimensional k-sample comparison problems", in: *Biometrika* 107 (3), pp. 555–572, DOI: 10.1093/biomet/asaa015.

Müller, A. (1997): "Integral Probability Metrics and Their Generating Classes of Functions", in: *Advances in Applied Probability* 29 (2), pp. 429–443, DOI: 10.2307/1428011.

Ntoutsi, I., Kalousis, A., and Theodoridis, Y. (2008): "A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees", in: *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)*, Proceedings, Society for Industrial and Applied Mathematics, pp. 810–821, DOI: 10.1137/1.9781611972788.73.

Olson, D. L. and Delen, D. (2008): *Advanced Data Mining Techniques*, Springer Berlin Heidelberg, Berlin, Heidelberg, DOI: 10.1007/978-3-540-76917-0_9.

Pan, W., Tian, Y., Wang, X., and Zhang, H. (2018): "Ball Divergence: Nonparametric Two Sample Test", in: *Annals of statistics* 46 (3), pp. 1109–1137, DOI: 10.1214/17-AOS1579.

Pawel, S., Kook, L., and Reeve, K. (2024): "Pitfalls and potentials in simulation studies: Questionable research practices in comparative simulation studies allow for spurious claims of superiority of any method", in: *Biometrical Journal* 66 (1), p. 2200091, DOI: 10.1002/bimj.202200091.

Petrie, A. (2016): "Graph-theoretic multisample tests of equality in distribution for high dimensional data", in: *Computational Statistics & Data Analysis* 96, pp. 145–158, DOI: 10.1016/j.csda.2015.11.003.

Politis, D. N., Romano, J. P., and Wolf, M. (1999): *Subsampling*, Springer Science & Business Media.

R Core Team (2024): *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.

R Foundation for Statistical Computing (2025): *The Comprehensive R Archive Network*, URL: https://cran.r-project.org/.

Rachev, S. T. (1991): *Probability metrics and the stability of stochastic models*, John Wiley & Sons, Chichester.

Ramdas, A., Trillos, N. G., and Cuturi, M. (2017): "On Wasserstein Two-Sample Testing and Related Families of Nonparametric Tests", in: *Entropy* 19 (2), p. 47, DOI: 10.3390/e19020047.

Roederer, M., Moore, W., Treister, A., Hardy, R. R., and Herzenberg, L. A. (2001): "Probability Binning Comparison: A Metric for Quantitating Multivariate Distribution Differences", in: *Cytometry* 45 (1), pp. 47–55.

Rosenbaum, P. R. (2005): "An Exact Distribution-Free Test Comparing Two Multivariate Distributions Based on Adjacency", in: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67 (4), pp. 515–530.

Schaefer, R. L., Roi, L., and Wolfe, R. (1984): "A Ridge Logistic Estimator", in: *Communications in Statistics - Theory and Methods* 13 (1), pp. 99–113, DOI: 10.1080/03610928408828664.

Schilling, M. F. (1986): "Multivariate Two-Sample Tests Based on Nearest Neighbors", in: *Journal of the American Statistical Association* 81 (395), pp. 799–806, DOI: 10.2307/2289012.

Schreck, N., Slynko, A., Saadati, M., and Benner, A. (2024): "Statistical plasmode simulations–Potentials, challenges and recommendations", in: *Statistics in Medicine* 43 (9), pp. 1804–1825, DOI: 10.1002/sim.10012.

Sigal, M. J. and Chalmers, R. P. (2016): "Play It Again: Teaching Statistics With Monte Carlo Simulation", in: *Journal of Statistics Education* 24 (3), pp. 136–156, DOI: 10.1080/10691898.2016.1246953.

Silverman, B. W. and Young, G. A. (1987): "The bootstrap: To smooth or not to smooth?", in: *Biometrika* 74 (3), pp. 469–479, DOI: 10.1093/biomet/74.3.469.

Smith, M. K. and Marshall, A. (2011): "Importance of protocols for simulation studies in clinical drug development", in: *Statistical Methods in Medical Research* 20 (6), pp. 613–622, DOI: 10.1177/0962280210378949.

Song, H. and Chen, H. (2022): *gTestsMulti: New Graph-Based Multi-Sample Tests*.

Song, H. and Chen, H. (2023): "Generalized kernel two-sample tests", in: *Biometrika*, asad068, DOI: 10.1093/biomet/asad068.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Lanckriet, G., and Schölkopf, B. (2008): "Injective Hilbert space embeddings of probability measures", in: *21st Annual Conference on Learning Theory (COLT 2008)*, Omnipress, pp. 111–122.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. G. (2010): "Hilbert Space Embeddings and Metrics on Probability Measures", in: *Journal of Machine Learning Research* 11 (50), pp. 1517–1561.

Steger-Hartmann, T., Sanz, F., Bringezu, F., and Soininen, I. (2025): "IHI VICT3R: Developing and Implementing Virtual Control Groups to Reduce Animal Use in Toxicology Research", in: *Toxicologic Pathology* 53 (2), pp. 230–233.

Steyerberg, E. W. (2019): *Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating*, Statistics for Biology and Health, Springer International Publishing, Cham, DOI: 10.1007/978-3-030-16399-0.

Stolte, M., Albrecht, W., Brecklinghaus, T., Gründler, L., Chen, P., Hengstler, J. G., Kappenberg, F., and Rahnenführer, J. (2023): "Classification of hepatotoxicity of compounds based on cytotoxicity assays is improved by additional interpretable summaries of high-dimensional gene expression data", in: *Computational Toxicology* 28, p. 100288, DOI: 10.1016/j.comtox.2023.100288.

Stolte, M., Herbrandt, S., and Ligges, U. (2024a): "A comprehensive review of bias reduction methods for logistic regression", in: *Statistics Surveys* 18, pp. 139–162, DOI: 10.1214/24-SS148.

Stolte, M., Kappenberg, F., Rahnenführer, J., and Bommert, A. (2024b): "Methods for quantifying dataset similarity: a review, taxonomy and comparison", in: *Statistics Surveys* 18, pp. 163–298, DOI: 10.1214/24-SS149.

Stolte, M., Rahnenführer, J., and Bommert, A. (2025a): "An Empirical Comparison of Methods for Quantifying the Similarity of Categorical Datasets", Unpublished.

Stolte, M. and Sauer, L. (2025): *DataSimilarity: Quantifying Similarity of Datasets and Multivariate Two- And k-Sample Testing*, R package version 0.1.1.

Stolte, M., Sauer, L., Rahnenführer, J., and Bommert, A. (2025b): "DataSimilarity: an R Package for Quantifying Similarity of Datasets and Multivariate Two- and k-Sample Testing", Unpublished.

Stolte, M., Schreck, N., Slynko, A., Saadati, M., Benner, A., Rahnenführer, J., and Bommert, A. (2024c): "Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression", in: *PLOS ONE* 19 (5), e0299989, DOI: 10.1371/journal.pone.0299989.

Stolte, M., Schreck, N., Slynko, A., Saadati, M., Benner, A., Rahnenführer, J., Bommert, A., and for the topic group "High-dimensional data" (TG9) of the STRATOS Initiative (2025c): "Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in comparing classification methods on high-dimensional data", in: *PLOS ONE* 20 (6), pp. 1–36, DOI: 10.1371/journal.pone.0322887.

Strobl, C. and Leisch, F. (2024): "Against the "one method fits all data sets" philosophy for comparison studies in methodological research", in: *Biometrical Journal* 66 (1), p. 2200104, DOI: 10.1002/bimj.202200104.

Székely, G. J. and Rizzo, M. L. (2004): "Testing for equal distributions in high dimension", in: *InterStat* 5 (16.10), pp. 1249–1272.

Székely, G. J. and Rizzo, M. L. (2017): "The Energy of Data", in: *Annual Review of Statistics and Its Application* 4 (1), pp. 447–479, DOI: 10.1146/annurev-statistics-060116-054026.

Tatti, N. (2007): "Distances between Data Sets Based on Summary Statistics.", in: *Journal of Machine Learning Research* 8 (1).

Thurow, M., Dumpert, F., Ramosaj, B., and Pauly, M. (2021): "Imputing missings in official statistics for general tasks – our vote for distributional accuracy", in: *Statistical Journal of the IAOS* 37 (4), Publisher: IOS Press, pp. 1379–1390, DOI: 10.3233/SJI-210798.

Tibshirani, R. (1996): "Regression Shrinkage and Selection via the Lasso", in: *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1), pp. 267–288.

Vapnik, V. and Chervonenkis, A. (1974): *Theory of pattern recognition*, Nauka, Moscow.

Vaughan, L. K., Divers, J., Padilla, M. A., Redden, D. T., Tiwari, H. K., Pomp, D., and Allison, D. B. (2009): "The use of plasmodes as a supplement to simulations: A simple example evaluating individual admixture estimation methodologies", in: *Computational Statistics & Data Analysis*, Statistical Genetics & Statistical Genomics: Where Biology, Epistemology, Statistics, and Computation Collide 53 (5), pp. 1755–1766, DOI: 10.1016/j.csda.2008.02.032.

Wackerly, D., Mendenhall, W., and Scheaffer, R. L. (2014): *Mathematical Statistics with Applications*, Cengage Learning.

Wang, H. and Pei, J. (2005): "A random method for quantifying changing distributions in data streams", in: *European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, pp. 684–691.

Wang, S. (1995): "Optimizing the smoothed bootstrap", in: *Annals of the Institute of Statistical Mathematics* 47 (1), pp. 65–80, DOI: 10.1007/BF00773412.

Wu, C. F. J. (1986): "Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis", in: *The Annals of Statistics* 14 (4), pp. 1261–1295, DOI: 10.1214/aos/1176350142.

Yu, K., Martin, R., Rothman, N., Zheng, T., and Lan, Q. (2007): "Two-sample Comparison Based on Prediction Error, with Applications to Candidate Gene Association Studies", in: *Annals of Human Genetics* 71 (1), pp. 107–118, DOI: 10.1111/j.1469-1809.2006.00306.x.

Zaremba, W. (2022): *B - test*.

Zech, G. and Aslan, B. (2003): *A new test for the multivariate two-sample problem based on the concept of minimum energy*, arXiv:math/0309164 version: 1, DOI: 10.48550/arXiv.math/0309164.

Zhang, J. and Chen, H. (2022): "Graph-Based Two-Sample Tests for Data with Repeated Observations", in: *Statistica Sinica* 32 (1), Publisher: Institute of Statistical Science, Academia Sinica, pp. 391–415.

Zhou, D. and Chen, H. (2023): "A new ranking scheme for modern data and its application to two-sample hypothesis testing", in: *Proceedings of Thirty Sixth Conference on Learning Theory*, ISSN: 2640-3498, PMLR, pp. 3615–3668.

Part II.

Publications

1. Article 1: Simulation Study to Evaluate When Plasmode Simulation is Superior to Parametric Simulation in Estimating the Mean Squared Error of the Least Squares Estimator in Linear Regression

PLOS ONE

RESEARCH ARTICLE

Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression

Marieke Stolte[1]*, **Nicholas Schreck**[2], **Alla Slynko**[3], **Maral Saadati**[2], **Axel Benner**[2], **Jörg Rahnenführer**[1], **Andrea Bommert**[1]

1 Department of Statistics, TU Dortmund University, Dortmund, North Rhine-Westphalia, Germany, **2** Division of Biostatistics, German Cancer Research Center, Heidelberg, Baden-Wuerttemberg, Germany, **3** Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada

* stolte@statistik.tu-dortmund.de

Citation: Stolte M, Schreck N, Slynko A, Saadati M, Benner A, Rahnenführer J, et al. (2024) Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression. PLoS ONE 19(5): e0299989. https://doi.org/10.1371/journal.pone.0299989

Editor: Mohamed R. Abonazel, Cairo University, EGYPT

Received: December 21, 2023

Accepted: February 20, 2024

Published: May 15, 2024

Data Availability Statement: The R code and results for the simulation are available on Zenodo (https://doi.org/10.5281/zenodo.10567144, https://doi.org/10.5281/zenodo.10567059).

Funding: MS has been supported (in part) by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project P1) funded by the Deutsche Forschungsgemeinschaft (DFG, https://gepris.dfg.de/gepris/projekt/427806116, German Research

Abstract

Simulation is a crucial tool for the evaluation and comparison of statistical methods. How to design fair and neutral simulation studies is therefore of great interest for both researchers developing new methods and practitioners confronted with the choice of the most suitable method. The term simulation usually refers to parametric simulation, that is, computer experiments using artificial data made up of pseudo-random numbers. Plasmode simulation, that is, computer experiments using the combination of resampling feature data from a real-life dataset and generating the target variable with a known user-selected outcome-generating model, is an alternative that is often claimed to produce more realistic data. We compare parametric and Plasmode simulation for the example of estimating the mean squared error (MSE) of the least squares estimator (LSE) in linear regression. If the true underlying data-generating process (DGP) and the outcome-generating model (OGM) were known, parametric simulation would obviously be the best choice in terms of estimating the MSE well. However, in reality, both are usually unknown, so researchers have to make assumptions: in Plasmode simulation studies for the OGM, in parametric simulation for both DGP and OGM. Most likely, these assumptions do not exactly reflect the truth. Here, we aim to find out how assumptions deviating from the true DGP and the true OGM affect the performance of parametric and Plasmode simulations in the context of MSE estimation for the LSE and in which situations which simulation type is preferable. Our results suggest that the preferable simulation method depends on many factors, including the number of features, and on how and to what extent the assumptions of a parametric simulation differ from the true DGP. Also, the resampling strategy used for Plasmode influences the results. In particular, subsampling with a small sampling proportion can be recommended.

Foundation - Project Number 427806116). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Introduction

Simulation studies are usually defined as computer experiments using artificial data generated by a pseudo-random number generator for which some truth about the data-generating process (DGP) and the outcome-generating model (OGM) is known, e.g., the true parameter values of the OGM or the distribution of the features. Well-designed, fair simulation studies are needed both for the evaluation of newly introduced methods and, in particular, for the neutral comparison of existing methods [1]. The DGP and OGM are usually chosen to either reflect realistic scenarios or edge cases for the application of the method of interest. We investigate the first case here.

We call the kind of simulation with artificial data, where the DGP and OGM are fully known, "parametric" simulations. Non-parametric simulation, where all data are real-life data, is not part of our analysis. Plasmode simulation is a special case of semi-parametric simulation, which is characterized by parts of the data being real-life data and parts of the DGP or OGM being specified. [2] give a technical introduction to (parametric) simulation studies and focus on guidance for best practices in performing and reporting simulation studies ("ADEMP" criteria). [3] give a general and more applied introduction to (parametric) simulation studies.

Parametric simulation studies are a crucial tool in the performance evaluation and comparison of statistical methods since they can offer insights beyond analytical results [2, 3] and can be used to evaluate criteria that cannot be assessed on real data where the DGP and OGM are unknown [3]. An example is the bias of an estimator, which can only be evaluated if the true parameter value can be controlled within the simulation. Therefore, one of the main advantages of parametric simulation studies is the full knowledge of the parameters of the DGP and OGM within the study. Another advantage is the possibility of investigating large numbers of different scenarios which permits analyzing how the performance of methods depends on the choice of the DGP and OGM. Moreover, it is possible to generate very large numbers of datasets. One of the main disadvantages is the simplification of real-life DGPs. Often very simple DGPs and OGMs are chosen arbitrarily which then do not reflect the often complex real-life processes. This may lead to wrong conclusions [3]. The over-simplification gets even worse for high-dimensional data, as it gets harder, for example, to specify realistic distributions and correlation structures for an increasing number of variables [4].

A different approach are so-called statistical Plasmodes as first introduced by [5]. [4] distinguish between statistical and biological Plasmodes depending on the procedure used for generating data. The motivation of statistical Plasmodes is to preserve a realistic data structure by resampling feature data from real-life datasets instead of using pseudo-random numbers as usually done in parametric simulation. At the same time, some control over the generated data is given by generating outcome variables for the given resampled feature data according to a known outcome-generating model like in parametric simulation. So, parametric and Plasmode simulations differ in the generation of features, while outcomes are generated in the same manner. For the feature resampling, different resampling approaches can be utilized [4]. Biological Plasmodes are generated by natural biological processes, for example in a wet lab by manipulating biological samples. In this paper, only statistical Plasmodes are considered.

The main advantage of Plasmode simulations is that the DGP does not have to be specified. The resampling is claimed to ensure the generation of realistic feature data. At the same time, quantities depending on the parameters of the OGM can still be assessed in contrast to fully non-parametric simulations. However, the resampling requires suitable datasets from the true DGP of interest with not too few observations. Depending on the application, this might be a major limitation. A more detailed discussion of the advantages and disadvantages of parametric vs. Plasmode simulation is given in [4]. The authors especially point out that evidence for

the often-made claim of Plasmode simulations producing more realistic data, i.e. data that is closer to the true DGP, is missing. Also, the authors emphasize the lack of studies on the effect of the often arbitrary choice of OGMs, which might affect both Plasmode and parametric simulation studies.

We aim to compare the ability of Plasmode and parametric simulation to assess the performance of statistical methods, especially concerning how misspecifications affect the results in both cases. We do this in a controlled simulation scenario so that we know both the true DGP and the true OGM, for evaluation purposes. In general, if we knew the truth, parametric simulation using this truth would be best. Since the truth is usually unknown in real-life applications, for parametric simulation researchers instead have to make assumptions about the DGP. These assumptions might deviate from the truth. Without deviations, the parametric simulation will always perform best since it accurately reflects the truth. On the other hand, when the parametric assumptions about the DGP are far from the truth, we expect Plasmode to be superior since the resampling is expected to give results that are rarely very far from the true DGP. Our goal is to determine the extent of deviation for which the parametric simulation gets worse than Plasmode. Therefore, we aim to find out

1. How much the DGP chosen in the parametric simulation can deviate from the truth before the parametric simulation becomes worse than Plasmode.
2. How deviations of the chosen OGM from the true OGM affect both parametric and Plasmode simulations.
3. How the choice of the resampling type affects the Plasmode simulation.

Based on the results, we are able to give guidance in which situations to choose parametric or Plasmode simulation and how to perform it.

We restrict our analysis to a simple scenario and focus on the estimation of the mean squared error (MSE) of the least squares estimator (LSE) in a linear regression model. Therefore, we focus on explanatory performance of the linear model and do not consider predictive performance. Moreover, we restrict to the low-dimensional setting, i.e., at most $p = 50$ features. We compare how well both parametric simulations with different assumptions about the DGP and the OGM and Plasmode simulations using different resampling strategies estimate the true MSE. So here, we check how well parametric and Plasmode simulation perform for one particular example of application. We investigate this for different true DGPs and OGMs. To compare different methods via simulation, this approach ensures that the simulation studies approximate well the performance of the methods for the true DGP and OGM.

The article is structured as follows. First, we describe parametric and Plasmode simulation in general and our specific simulation setup. Afterwards, we present the results of our simulations and provide recommendations for performing simulations based on our results. Finally, the results are summarized and discussed.

Methods

In the following, we briefly explain parametric and Plasmode simulation in general, pointing out, in particular, the different options for the resampling strategy in Plasmode simulations.

Parametric simulation

In parametric simulation studies, the whole data-generating process (DGP) and the outcome-generating model (OGM) have to be specified and are therefore known within the study. They are usually set up to either be as close as possible to a certain kind of data that the researcher is

interested in (e.g. gene expression data) or to cover as many situations as possible, possibly including extreme situations. We focus on the first case. Given the specified DGP, a large number of feature datasets is generated using pseudo-random number generators. In all cases that we are investigating, a target variable is then generated from these features by applying the OGM. This yields a large number of datasets which are then used for applying the methods of interest. This procedure allows the researcher to evaluate the performance of the methods with respect to a metric of interest. The process of generating the datasets can be seen as mimicking the repeated collection of samples from a large population. The results can provide insights into how the methods under study perform on average for datasets that are similar to the chosen DGPs and OGMs [3]. For more details on how to design, perform, analyze, and report parametric simulation studies, refer to [2].

Plasmode simulation

In Plasmode simulation studies, no assumptions on the DGP for the feature data are made. Instead, it is required to have a representative real-life dataset at hand that resulted from the true DGP [4]. If only real data was used, we would have no control over the DGP and OGM in our simulation. This means that we could not estimate certain quantities (e.g. the bias of an estimator) that directly depend on the true unknown parameters [3]. To enable us to estimate the quantities that directly depend on the true parameters of the OGM (which are most quantities of interest for performance evaluation of models), Plasmode simulation combines the use of real feature data with a known OGM. A Plasmode simulation study then works as follows. In each iteration, a Plasmode dataset is drawn from the real-life dataset at hand. The researcher has to decide on the resampling method. Possible methods include

- n out of n Bootstrap [6], i.e. drawing with replacement a dataset of the same size as the original dataset,
- m out of n Bootstrap [7–9], i.e. drawing with replacement $m < n$ observations of the original dataset,
- subsampling, i.e. drawing without replacement $m < n$ observations of the original dataset,

or other adaptations of Bootstrap like

- smoothed Bootstrap [10–13], i.e. applying kernel-estimation to the empirical distribution of the original dataset and resampling from this smoothed empirical distribution,
- wild Bootstrap [14], i.e. adding the standardized values of each variable scaled by a random number to the original variable, or
- no resampling, i.e. using the whole dataset as it is.

A discussion of the first three options in the context of Plasmode simulation can be found in [4]. In the case of m out of n Bootstrap and subsampling, the researcher also has to decide on the number of observations to draw. For m out of n Bootstrap, there exists a data-dependent algorithm to find the optimal value of m [15]. After resampling a number of Plasmode datasets, the OGM is applied to each of the datasets to generate the outcomes. The resulting datasets can then be used for computing the performance metrics of interest like in parametric simulation. For more details, see [4].

Setup of the comparison study

In the following, we first describe the general approach and then the detailed setup of our comparison study.

General approach

We conduct our comparisons with respect to different true DGPs and OGMs. For each scenario, we calculate the true MSE of the least squares estimator (LSE). We then perform a parametric and a Plasmode simulation for estimating the MSE. For these simulations, we choose different DGPs and OGMs. The estimated MSEs resulting from these simulations are then compared to the true MSEs to assess how well the parametric and Plasmode simulation approximate the true values.

The outcomes are in all cases generated according to a linear model

$$y = X\beta + \varepsilon \tag{1}$$

for the true scenarios as well as for the parametric and Plasmode simulations. Note that the intercept is included in this model. The true MSE of the LSE $\hat{\beta}$ depends on the number of observations n, the residual variance of ε, and the distribution of the features X. For fixed X, the LSE is unbiased and thus the MSE reduces to the variance, which is given by $\mathrm{Var}(\hat{\beta}|X) = \sigma^2(X^TX)^{-1}$.

To define the true DGP and OGM, we have to determine

- the true distribution of the features,
- the true parameter vector β, and
- the true distribution of the error term ε.

In the context of the parametric simulation, both DGP and OGM have to be chosen, so the distribution of the features, the coefficient vector, and the error distribution have to be specified. Inside the Plasmode simulation, only the OGM has to be chosen, so the coefficient vector and the error distribution have to be specified.

Simulation setup

In this section, we describe the true scenarios used in the simulation as well as the deviations from these true scenarios that are assumed for the parametric or Plasmode simulation.

True scenarios. We use different scenarios as our truth for the comparison. Table 1 gives an overview of these scenarios. The scenarios differ in the number of features (p) and observations (n) as well as the true correlation structure. In all scenarios, we assume that our features come from a multivariate normal distribution with mean zero and variances of one, that the true vector of coefficients (β) consists of all ones, and that the true error distribution is $N(0, 0.3^2)$.

We start with simple scenarios with only two features ($p = 2$), which are sampled from a bivariate Gaussian distribution with mean zero, variances of one, and a pairwise correlation of 0.2 or 0.5, and 50 or 100 observations. We use the same parameter settings for $p = 10$ except that we only look at pairwise correlations of 0.2. For $p = 50$, we always use 100 observations for identifiability reasons. Once again, we set all pairwise correlations to 0.2. Additionally, we use block diagonal correlation matrices with five blocks of ten features each. Within each block, the correlations are once set to $0.2^{|i-j|}$ and once to $0.5^{|i-j|}$ for all $i \neq j$. Features from different blocks are assigned a correlation of 0.

Table 1. Parameters for true data generating processes (DGP) and outcome generating models (OGM). In all scenarios, the true vector of coefficients is equal to $(1, \ldots, 1)^T \in \mathbb{R}^{p+1}$ and the error distribution is set to $\varepsilon \sim N(0, 0.3^2)$. $\mathbf{0}_p$ denotes the p-dimensional vector of zeros.

Name	p	n	Distribution of features
($p2n100\rho0.2$)	2	100	$(X_1, X_2)^T \sim N_2(\mathbf{0}_2, \Sigma)$ with $\Sigma_{i,j} = 0.2\ \forall i \neq j, \Sigma_{ii} = 1$
($p2n50\rho0.2$)	2	50	$(X_1, X_2)^T \sim N_2(\mathbf{0}_2, \Sigma)$ with $\Sigma_{i,j} = 0.2\ \forall i \neq j, \Sigma_{ii} = 1$
($p2n100\rho0.5$)	2	100	$(X_1, X_2)^T \sim N_2(\mathbf{0}_2, \Sigma)$ with $\Sigma_{i,j} = 0.5\ \forall i \neq j, \Sigma_{ii} = 1$
($p10n100\rho0.2$)	10	100	$(X_1, \ldots, X_{10})^T \sim N_{10}(\mathbf{0}_{10}, \Sigma)$ with $\Sigma_{i,j} = 0.2\ \forall i \neq j, \Sigma_{ii} = 1$
($p10n50\rho0.2$)	10	50	$(X_1, \ldots, X_{10})^T \sim N_{10}(\mathbf{0}_{10}, \Sigma)$ with $\Sigma_{i,j} = 0.2\ \forall i \neq j, \Sigma_{ii} = 1$
($p50n100\rho0.2$)	50	100	$(X_1, \ldots, X_{50})^T \sim N_{50}(\mathbf{0}_{50}, \Sigma)$ with $\Sigma_{i,j} = 0.2\ \forall i \neq j, \Sigma_{ii} = 1$
($p50n100\rho0.2^{\lvert i-j\rvert}$)	50	100	$(X_1, \ldots, X_{50})^T \sim N_{50}(\mathbf{0}_{50}, \Sigma)$ with covariance matrix Σ with blockdiagonal structure where within each of 5 blocks of 10 features the pairwise covariance/ correlation between the ith and jth feature of the block is given as $0.2^{\lvert i-j\rvert}$ and all variances are equal to 1
($p50n100\rho0.5^{\lvert i-j\rvert}$)	50	100	$(X_1, \ldots, X_{50})^T \sim N_{50}(\mathbf{0}_{50}, \Sigma)$ with covariance matrix Σ with block diagonal structure where within each of 5 blocks of 10 features the pairwise covariance/ correlation between the ith and jth feature of the block is given as $0.5^{\lvert i-j\rvert}$ and all variances are equal to 1
(quake)	3	100	$(X_1, \ldots, X_3)^T \sim N_3(\mathbf{0}_3, \Sigma)$ with covariance matrix Σ estimated from real dataset quake [16]
(wine_quality)	11	100	$(X_1, \ldots, X_{11})^T \sim N_{11}(\mathbf{0}_{11}, \Sigma)$ with covariance matrix Σ estimated from real dataset wine_quality [17]
(pol)	26	100	$(X_1, \ldots, X_{26})^T \sim N_{26}(\mathbf{0}_{26}, \Sigma)$ with covariance matrix Σ estimated from real dataset pol [18]
(Yolanda)	100	200	$(X_1, \ldots, X_{100})^T \sim N_{100}(\mathbf{0}_{100}, \Sigma)$ with covariance matrix Σ estimated from real dataset Yolanda [19]

https://doi.org/10.1371/journal.pone.0299989.t001

The scenarios are chosen to represent a low, a moderate, and a higher number of features for which the estimation process is still stable for 100 observations.

Moreover, we use covariance matrices estimated from real datasets to see how the simulations behave with more complicated correlation structures. We chose regression datasets that were available on OpenML [20], were used in the benchmark in [21], had at most 100 features, no constant features, no missing values and pairwise correlations with absolute values of at most 0.95. With these criteria, we ended up with four datasets: quake [16], wine_quality [17], pol [18], and Yolanda [19].

Deviations from true scenarios. We choose DGPs and OGMs for parametric and Plasmode simulation that present different kinds of deviations from the truth described in the previous section. The general structures of these deviations are listed in Table 2. A complete list of the specific parameter values that were chosen can be found in S1 Table. As a baseline, we assume the true scenario, which reflects the case that we—by chance—correctly specify all parameters in the simulations. Then we consider choices for each part of the DGP and OGM that reflect increasing deviations from the truth. For the coefficients, we use different values that are either wrong, but of the same order, or that even differ a large factor. We also included the case of assuming no effect ($\beta = 0$) which is an important special case that might be of interest in many studies. For the distribution of ε, we either only misspecify its standard deviation or misspecify the distribution as either more heavy-tailed (scaled t-distribution) or skewed (scaled and shifted χ^2-distribution). As deviations from the true feature distribution, we first still assume multivariate normal distribution but with wrong correlations, expectations, or variances. We then look at entirely wrong distributions, namely Gaussian mixture, log-normal, and Bernoulli distribution. The true correlation structure is preserved in those cases. We

Table 2. Deviations from true DGP and OGM for parametric and Plasmode simulation.

Scenario name	Description
True model	Assumptions coincide with truth
Coefficients misspecified I	Assumed β vector $(0, 1/p, 2/p, \ldots, 1)^T \in \mathbb{R}^{p+1}$ instead of $\mathbf{1}_{p+1}$
Coefficients misspecified II	Assumed β vector $\mathbf{0.05}_{p+1}$ instead of $\mathbf{1}_{p+1}$
Coefficients misspecified III	Assumed β vector $\mathbf{10}_{p+1}$ instead of $\mathbf{1}_{p+1}$
Coefficients misspecified IV	Assumed β vector $\mathbf{0}_{p+1}$ instead of $\mathbf{1}_{p+1}$
Error sd misspecified c	Assumed $\sigma = c$ instead of $\sigma = 0.3$ for $\varepsilon \sim N(0, \sigma^2)$
Correlation misspecified ρ	Assumed fixed pairwise correlation of ρ
Correlation misspecified $\rho^{\lvert i-j \rvert}$	Assumed pairwise correlation of $\rho^{\lvert i-j \rvert}$ for ith and jth feature for $p = 10$, or ith and jth feature within each of 5 blocks of 10 features for $p = 50$, respectively
Coefficients (I) and correlation (ρ) misspec.	$\mathbf{0.05}_{p+1}$ instead of $\mathbf{1}_{p+1}$ and fixed pairwise correlation of ρ instead of ρ_{true}
Coefficients (II) and correlation (ρ) misspec.	Assumed β vector $\mathbf{10}_{p+1}$ instead of $\mathbf{1}_{p+1}$ and fixed pairwise correlation of ρ instead of ρ_{true}
Error sd (0.4) and correlation (ρ) misspec.	Assumed $\sigma = 0.4$ instead of $\sigma = 0.3$ for $\varepsilon \sim N(0, \sigma^2)$ and fixed pairwise correlation of ρ instead of ρ_{true}
Feature distribution misspecified N (0,1), N(μ,1)	Assumed expectation of μ for second half of features
Feature distribution misspecified N (μ,1)	Assumed expectation of μ for all features
Feature distribution misspecified N (0,1), N(0,σ^2)	Assumed variance of σ^2 for second half of features
Feature distribution misspecified N(0, σ^2)	Assumed variance of σ^2 for all features
Feature distribution misspecified N (0,1), $(1-\alpha)$N(0,1)+αN(0,10)	Assumed marginal distribution of second half of features as Gaussian mixture with 100α% outliers sampled from $N(0, 10)$ and marginal distribution of first half of features misspecified as normal with mean 0 and variance that matches the variance σ^2 of the second half of features, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Feature distribution misspecified N(μ, σ^2), $(1-\alpha)$N(0,1)+αN(3,1)	Assumed marginal distribution of second half of features as Gaussian mixture with 100α% of the observations sampled from $N(3, 1)$ and marginal distribution of first half of features misspecified as normal with mean μ and variance σ^2 chosen such that they match mean and variance of the second half of features, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Feature distribution misspecified N (1.65,2.83), logN(0,1)	Assumed marginal distribution of second half of features misspecified as log-normal with parameters 0 and 1 and marginal distribution of first half of features misspecified as normal with matching mean and variance, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Feature distribution misspecified Bin (π)	Assumed marginal distribution of second feature misspecified as Bernoulli with a success probability of π, $Cor(X_i, X_j) = \rho_{true}$, $i \neq j$ still holds
Error distribution misspecified t(df) scaled	Assumed $\varepsilon \sim t_{\text{df}}$ and scaled ε to still have sd 0.3
Error distribution misspecified chisq (df) scaled	Assumed $\varepsilon \sim \chi^2_{\text{df}}$ and shifted and scaled ε to still have mean 0 and sd 0.3

https://doi.org/10.1371/journal.pone.0299989.t002

achieve this by generating Gaussian mixture, log-normal, and Bernoulli variables from multivariate normals and setting the covariance matrix of the underlying normals in a way such that the corresponding variables have the desired variances and covariances. For log-normals and Bernoulli variables with variables of the same distribution, the calculation can be found in [22, 23]. The calculation for log-normal and Bernoulli variables in combination with normal variables as well as all calculations for Gaussian mixture variables can be found in S1 Appendix.

Simulation procedure

The overall simulation structure is described in Algorithm 1. For each true scenario, we first approximate the true MSE by drawing 25 000 000 datasets of size n from the true distribution of X with the first column being a vector of ones, corresponding to the intercept of the model. We then calculate $X\beta$ and add random noise ε according to the true distribution of ε and define this as our outcome vector y belonging to the respective dataset. For each pair of data X and corresponding target y, we estimate $\hat{\beta}$ using least squares estimation. We then calculate the component-wise means over the replications of the simulation of $(\hat{\beta}_j - \beta_j)^2, j = 0, \ldots, p$, with p denoting the number of features, as estimates of the true component-wise MSEs. We refer to these quantities as the "true" component-wise MSEs. In each true scenario, we then perform parametric and Plasmode simulations for estimating the component-wise MSEs under the assumption that we do not know the respective true scenario.

Algorithm 1 Structure of simulation process

Require: $n > 0$ (number of observations), $0 < p < n$ (number of features), $n.mse > 0$ (number of MSE estimations), $n.mod > 0$ (number of LSEs, i.e. model estimates, used for estimation of one estimated MSE), true DGP (distribution of features), true OGM (β, distribution of ε), assumed DGP (assumed distribution of features), assumed OGM (β_a, assumed distribution of ε), *type* of Bootstrap, proportion π for resampling (= 1 for n out of n Bootstrap, Wild Bootstrap, and Smoothed Bootstrap)

Ensure: Error in estimated MSE for parametric simulation

$\text{MSE}_{true;j} \leftarrow \mathbb{E}[(\hat{\beta}_j - \beta_j)^2], j = 0, \ldots, p$, for the LSE $\hat{\beta}$ in the true model

for $k = 1, \ldots, n.mse$ **do**

  $X^{(k,i)} \leftarrow$ design matrix generated with Algorithm 2 or 3 for $i = 1, \ldots, n.mod$

  for $i = 1, \ldots, n.mod$ **do**

    $\varepsilon^{(k,i)} \leftarrow$ noise sampled from assumed distribution of ε

    $y^{(k,i)} \leftarrow X^{(k,i)} \beta_a + \varepsilon^{(k,i)}$

    $\hat{\beta}^{(k,i)} \leftarrow ((X^{(k,i)})^T X^{(k,i)})^{-1} (X^{(k,i)})^T y^{(k,i)}$ ▷ LSE

  end for

  $\text{MSE}_j^{(k)} \leftarrow \frac{1}{n.mod} \sum_{i=1}^{n.mod} (\hat{\beta}_j^{(k,i)} - \beta_{a;j})^2, j = 0, \ldots, p$

  $Err_j^{(k)} \leftarrow \text{MSE}_j^{(k)} - \text{MSE}_{true;j}, j = 0, \ldots, p$

end for

The process for data generation for parametric simulation is described in Algorithm 2. We make different assumptions on the distribution of the features (X), the values of the coefficients β, and the distribution of ε. We then generate $n.mod = 1000$ datasets according to these assumptions using pseudo-random numbers.

Algorithm 2 Structure of feature data generation for parametric simulation

Require: $n > 0$, $0 < p < n$, assumed DGP, k (iteration number of Algorithm 1)

Ensure: Generated datasets

for $i = 1, \ldots, n.mod$ **do** ▷ Inner Simulation

  $X^{(k,i)} \leftarrow$ design matrix drawn from assumed data generating process using a pseudo-random number generator

end for

In some scenarios, we use parametric simulation with estimation of mean and covariance. For this, at the beginning of each simulation, one dataset of size $n = 1000$ is sampled from the true DGP and the mean and covariance are estimated from this dataset and used as the assumed mean and covariance of the assumed DGP. This corresponds to the case that researchers might have data at hand from which they estimate some characteristics of the DGP

to incorporate them into a parametric simulation in order to perform a more realistic simulation.

The procedure for data generation for Plasmode simulation is described in Algorithm 3. Here, we have to specify the resampling method to use. As a first step for each Plasmode simulation, one dataset is drawn from the true DGP. Note that in each case, the number of observations *after* resampling has to match the number of observations used for parametric simulation and for the true scenario to ensure a fair comparison of methods. For a more detailed discussion of this issue, see below. We then draw $n.mod = 1000$ resampled datasets from our dataset according to the chosen resampling method.

Algorithm 3 Structure of feature data generation for Plasmode simulation

Require: $n > 0$, $0 < p < n$, true DGP, *type* of Bootstrap, proportion π for resampling (= 1 for n out of n Bootstrap, Wild Bootstrap and Smoothed Bootstrap), k (iteration number of Algorithm 1)
Ensure: Plasmode datasets
$X^{(k)}_{Plasm} \leftarrow$ design matrix $\in \mathbb{R}^{\lceil n/\pi \rceil \times (p+1)}$ drawn from true DGP
for $i = 1, \ldots, n.mod$ **do** ▷ Inner Simulation
  if *type* == "m out of n Bootstrap" **or** *type* == "n out of n Bootstrap" **then**
    $X^{(k,i)} \leftarrow n$ rows sampled from $X^{(k)}_{Plasm}$ with replacement
  else
    if *type* == "Subsampling" **then**
      $X^{(k,i)} \leftarrow n$ rows sampled from $X^{(k)}_{Plasm}$ without replacement
    else
      if *type* == "Wild Bootstrap" **then**
        $a \leftarrow$ vector of p numbers sampled from $N(0, 1)$
        $X_1^{(k,i)} \leftarrow \mathbf{1}_n$
        $X_j^{(k,i)} \leftarrow X_j^{(k,i)} + a_j \cdot (X_j^{(k,i)} - \bar{X}_j^{(k,i)})/SD(X_j^{(k,i)})$, $j = 2, \ldots, p + 1$
      else
        if *type* == "Smoothed Bootstrap" **then**
          $X^{(k,i)} \leftarrow n$ rows sampled from $X^{(k)}_{Plasm}$ with replacement + random noise from a multivariate normal distribution centered at the data points and parameterized by corresponding bandwidth matrix estimated by Silverman's rule [24];
        end if
      end if
    end if
  end if
end for

We utilize the following Bootstrap versions:

- m out of n Bootstrap [7–9] with resampling proportion $\pi \in \{0.01, 0.1, 0.5, 0.632, 0.8, 0.9\}$, i.e. drawing with replacement n observations out of $\lceil n/\pi \rceil$ observations,
- n out of n Bootstrap [6], i.e. drawing with replacement n observations out of n (special case of m out of n Bootstrap for $\pi = 1$),
- Smoothed Bootstrap [10–13], i.e. drawing with replacement n observations out of the smoothed empirical distribution of n observations,
- Wild Bootstrap [14], i.e. adding the standardized version of each observed feature vector scaled with a noise factor sampled from $N(0, 1)$ to the observed feature vectors, and
- subsampling with resampling proportion $\pi \in \{0.01, 0.1, 0.5, 0.632, 0.8, 0.9\}$, i.e. drawing without replacement n observations out of $\lceil n/\pi \rceil$ observations,

- no resampling, equivalent to subsampling with resampling proportion $\pi = 1$.

We do not determine an optimal resampling proportion, e.g. with the algorithm introduced in [15], since it takes too much time to repeat this for every dataset in the simulation. Instead, we try a range of resampling proportions.

On each dataset generated either according to the parametric or the Plasmode approach, the linear model (1) using the chosen parameters for the OGM is then applied to generate the outcome variable. From these, $\hat{\beta}$ is estimated for each dataset. The MSE is estimated as the average component-wise squared difference of the estimated and assumed coefficient vectors. One estimated MSE value corresponds to the result of one parametric or Plasmode simulation study. The whole process is repeated 100 times so we can see how much variation exists in the MSE estimation when repeating the parametric or Plasmode simulation study.

Performance evaluation

To compare the performance of parametric and Plasmode simulation, we look at their errors in MSE estimation. For each type of simulation (parametric, parametric with estimation of mean and variance, Plasmode with different resampling methods and proportions) we obtain 100 estimated MSEs for each deviation from the true DGP and OGM. We calculate the component-wise absolute errors as the differences between estimated MSEs and corresponding true MSEs in each case. Additionally, we calculate the relative errors by dividing the absolute errors by the corresponding true MSEs. We aggregate the absolute and relative errors per simulation over the coefficients by taking the arithmetic mean over the absolute component-wise values. We aggregate over the repetitions of the simulation studies by taking the median of the aggregated values. With this strategy, runs with large errors in single coefficients obtain large aggregated values, while the overall aggregated value across simulation repetitions is robust against single simulations with large aggregated errors.

We examine the errors graphically using boxplots. An example with a corresponding explanation will be shown later. Additionally, we analyze how much the assumptions in parametric simulation can deviate from the truth until the results are worse than with Plasmode simulation. Therefore, we sort the parametric deviations within each subgroup (e.g. deviation from the variance of the multivariate normal) in increasing order of the magnitude of the deviation (e.g. if the true variance is 1 and the tested values are 0.1, . . ., 0.99, these are ordered decreasingly) and identify the first value in this order for which the fully aggregated error for parametric simulation is larger than that for the considered type of Plasmode simulation. In this way, we can quantitatively compare parametric simulation to the different Plasmode variants. If the first value where parametric is worse than Plasmode is close to the true value, it follows that for deviations of this type, the parametric simulation is very sensitive to small deviations and we have to be very confident in our parameter settings for the DGP if we want to use parametric simulation. These are the cases where Plasmode might be superior to parametric simulation.

Software

All analyses are performed using R 4.2.2 [25]. We use the mvtnorm package [26, 27] to simulate data from multivariate normal distributions. For smoothed Bootstrap, the R package kernelboot [28] is used. For visualization of the results, we use the ggplot2 package [29] and ggh4x [30]. The R code and results for the simulation are available on Zenodo (https://doi.org/10.5281/zenodo.10567144, https://doi.org/10.5281/zenodo.10567059).

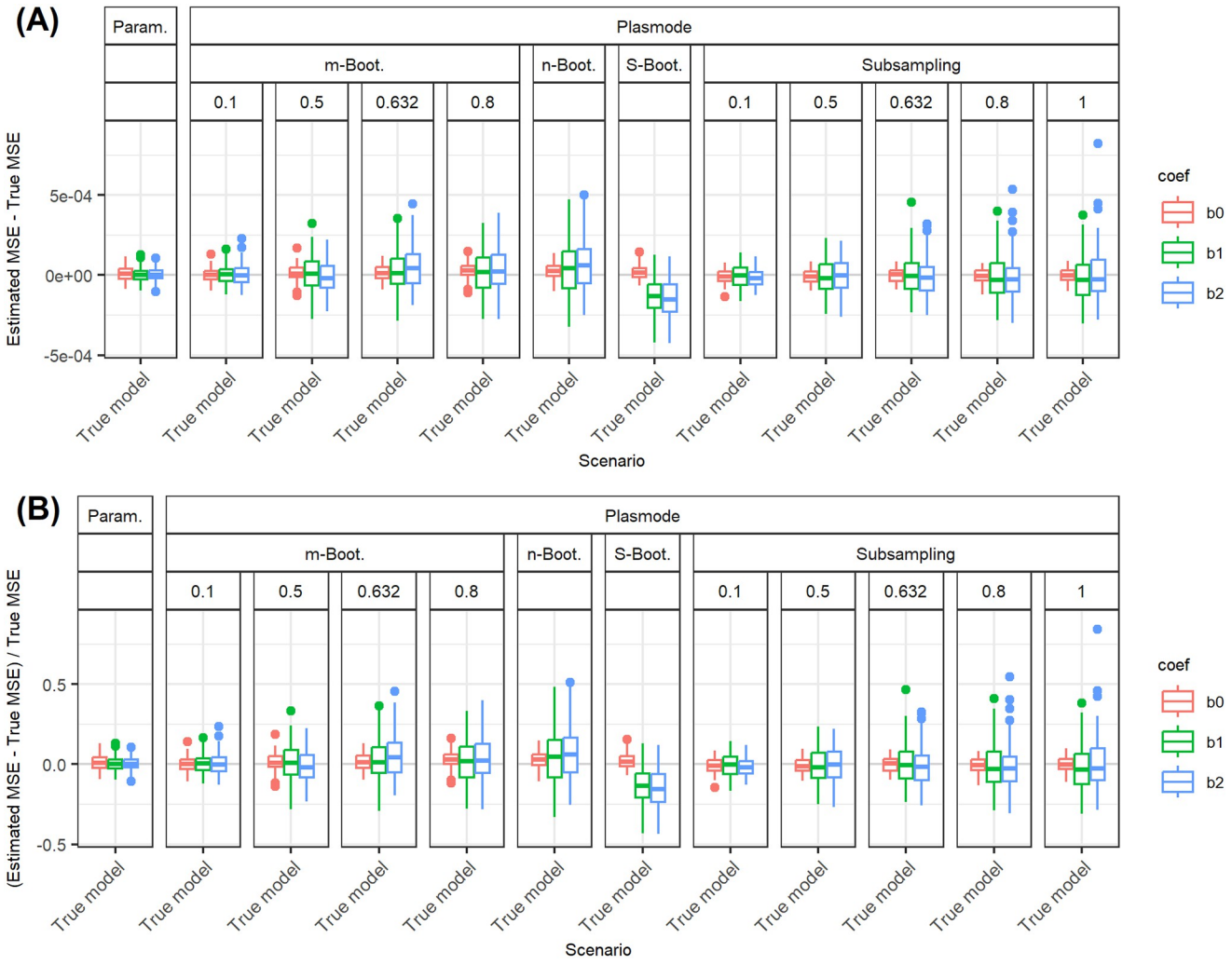


Fig 1. Relative error in MSE estimation for individual coefficients for different types of Plasmode simulation compared to parametric simulation under assumption of true DGP and OGM.

https://doi.org/10.1371/journal.pone.0299989.g001

Results

In this Section, we evaluate the results of the simulations. First, we explain the plots for one simple scenario and type of deviation. Then, the different resampling strategies for Plasmode are compared. Afterward, we discuss the results for the different types of deviations. Last, we consider the results for correlation structures estimated from real data as well as the effect of the size of the resampled dataset.

Example

In the following, we explain the displays that we use in the subsequent sections using one concrete example. We again consider the two scenarios with $p = 2$, $n = 100$, pairwise correlation of 0.2, $\beta = \mathbf{1}_3$, and $\varepsilon \sim N(0, 0.3^2)$ or $\varepsilon \sim N(0, 3^2)$. We calculated the errors in the MSE estimation using parametric and Plasmode simulation as described in the previous section. We display the errors in different ways using boxplots. We display the absolute or relative errors for each coefficient individually like in Figs 1 and 3, and S1 Fig, or aggregated over the coefficients like

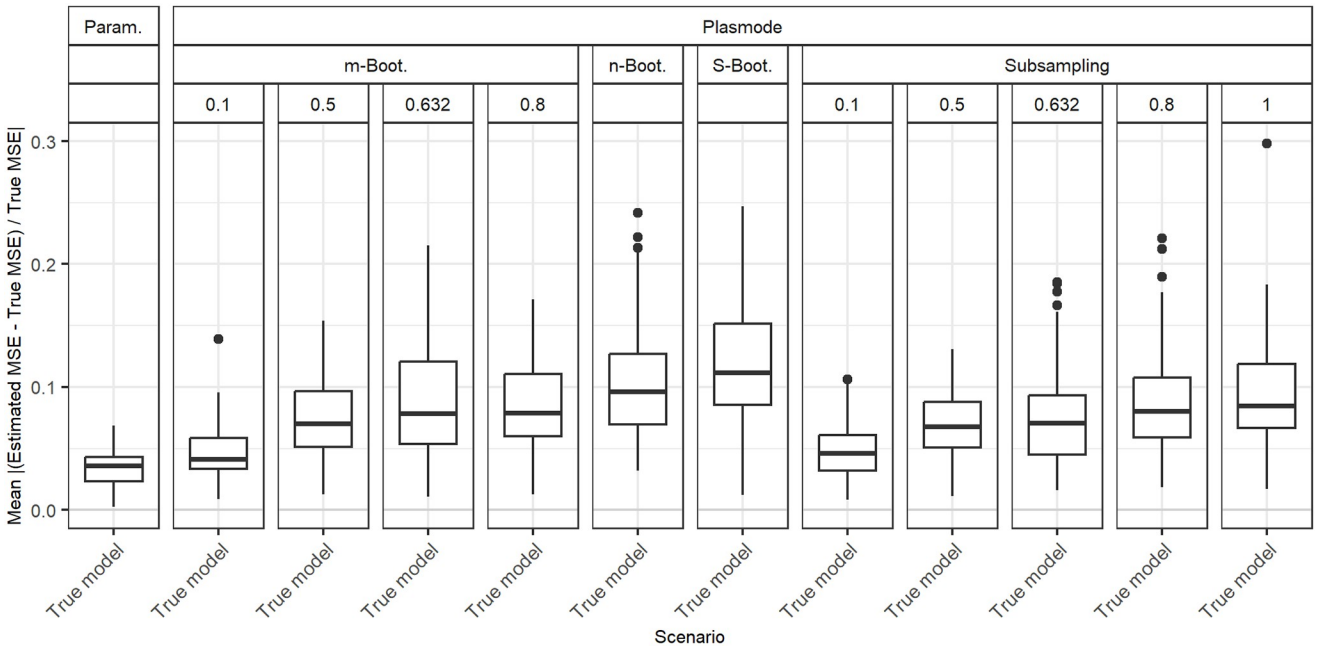


Fig 2. Absolute value of the relative error in MSE estimation averaged over individual coefficients, for different types of Plasmode simulation compared to parametric simulation under the assumption of the true DGP and OGM, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g002

in Fig 2. We use the unaggregated version in cases where the error for different coefficients might behave differently. This is for example the case for deviations in the feature distribution of the second half of features. Otherwise, if all coefficients behave similarly, we use the aggregated version.

For the individual coefficients, the absolute or relative errors of the 100 repetitions of each type of simulation (parametric, different types of Plasmode) are displayed in one box per coefficient. This is done separately for the true model and each deviation. The deviations are described on the x-axis and coefficients are distinguished by differently colored boxes. The headers give information about the type of simulation used. The first row is the distinction between parametric and Plasmode simulation. The second row gives the type of Plasmode simulation. The third row gives the resampling proportion. For example in Fig 1 in the third facet, the relative errors per coefficient for Plasmode using m out of n Bootstrap with a resampling proportion of 0.5 are displayed. This corresponds to sampling with replacement 100 observations from a dataset of 200 observations for each simulation. For parametric simulation, n out of n Bootstrap and Smoothed Bootstrap, there is no subsampling proportion so this field is left empty. We leave out Wild Bootstrap in the following analyses since it produces very large outliers and is consistently outperformed by all other Bootstrap types (see e.g. Fig 4). We abbreviate m out of n Bootstrap as m-Bootstrap, n out of n Bootstrap as n-Bootstrap and Smoothed Bootstrap as S-Bootstrap. If necessary we further abbreviate Bootstrap as Boot. or B., Parametric as Param. or Prm. and Subsampling as Sub.

For the aggregated errors, we display the mean over the absolute values of the errors of the individual coefficients per deviation, i.e. the mean error per coefficient of one simulation, for the 100 repetitions of each simulation type in one box. Apart from the aggregation, the figures are constructed in the same way as for the individual coefficients.

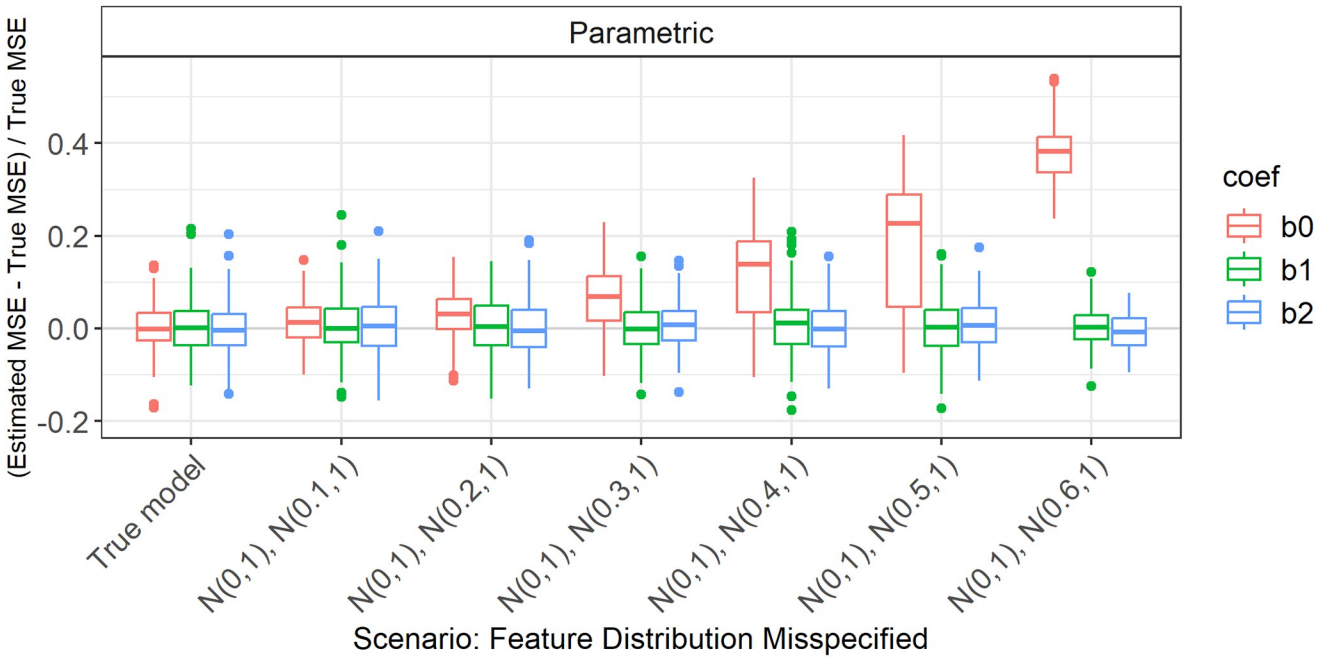


Fig 3. Absolute value of relative error in MSE estimation for individual coefficients when the assumed feature distribution in parametric simulation deviates from the true distribution, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g003

There are two types of comparisons: we can compare the performance of different types of simulation for the true model like in Figs 1 and 2, and S1 Fig to see how well each simulation type would perform if we knew the truth. Or we can compare the performance for differently strong deviations like in Fig 3. This allows us to assess the impact of different deviations on the performance. We can also combine both displays and show the performance for one kind of deviation for all those types of simulation that are affected by it and the performance for all other simulation types for the true model only. For example, in the case of deviation from the mean of the second feature distribution, we can show the errors for parametric simulation for different amounts of deviation as in Fig 3 along with the performance of the Plasmode types under the true model as in Fig 1 or S1 Fig. We cannot misspecify the feature distribution in Plasmode simulation, so only the true model is shown. This combined version is the display that we will use for the rest of our analysis.

In general, we might be interested in both absolute and relative errors. As can be seen in S1 Fig, the absolute errors for our specific problem are directly dependent on the chosen parameter for the error standard deviation: if the standard deviation changes by a factor of 10, e.g. here from $\sigma = 0.3$ to $\sigma = 3$, the errors change by a factor of approximately $10^2 = 100$, which can easily be checked by the theoretical relation $\mathrm{Var}(\hat{\beta}|X) = \sigma^2 X^T X$, using that the LSE is unbiased for fixed X. The relative errors, on the other hand, are independent of σ since the factor affects both the absolute error and the true MSE, by which the absolute error is divided, in the same way. This is for example demonstrated in Fig 1. Therefore, we will only display the relative version for the rest of this analysis since the absolute values could be scaled to be arbitrarily small or large by choosing the error variance accordingly. We will also restrict our analysis to the case $\sigma = 0.3$, since this leads to more stable simulations than $\sigma = 3$, as the latter corresponds to an extremely low signal-to-noise ratio.

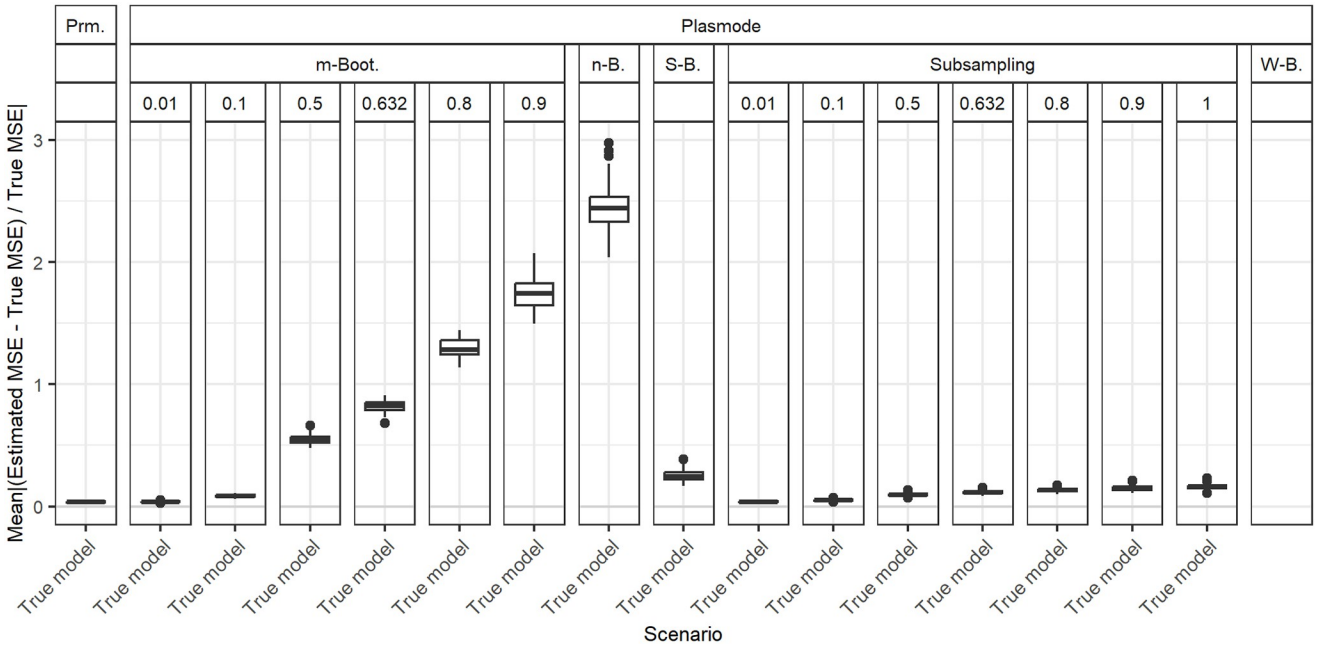


Fig 4. Absolute value of relative error in the MSE estimation averaged over individual coefficients for different types of Plasmode simulation compared to parametric simulation, under the assumption of the true data generating process and outcome generating model, for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g004

Comparison of different Plasmode types and resampling proportions

Fig 4 shows the aggregated relative errors for $p = 50$ with fixed pairwise correlations of 0.2 for the true model for all types of simulation. This example confirms that overall, Plasmode using Wild Bootstrap performs worst. All values for its relative errors lie outside the range of all other resampling types. This is similar for other scenarios, such that we do not show the results for Wild Bootstrap in any other plot. Within the other simulation types, Plasmode using the n out of n Bootstrap performs worst with relative mean errors of around 2.5 and also relatively high variation. m out of n Bootstrap and subsampling perform better both in terms of the median aggregated error and in terms of smaller variability with decreasing resampling proportion, i.e. the larger the dataset from which the 100 observations are sampled, the lower the variability. m out of n Bootstrap converges towards n out of n Bootstrap for increasing subsampling proportions. Except for very low subsampling proportions (0.1 and 0.01), Bootstrap performs worse than subsampling both with regard to median aggregated error and variability. It is interesting to note, that no resampling (i.e. subsampling with a subsampling rate of one), which means using the same feature data for the whole simulation and only sampling new observations of the target, still outperforms m out of n Bootstrap with subsampling proportions from 0.5 on as well as the smoothed, n out of n, and wild bootstrap. Smoothed Bootstrap performs worse than all subsampling versions, but better than the m out of n Bootstrap for subsampling proportions from 0.5 on. With Smoothed Bootstrap, the true MSE of the slope coefficients gets consistently underestimated under the true model (see e.g. Fig 1). Subsampling and m out of n Bootstrap are indistinguishable for very low subsampling proportions since the impact of duplicate observations decreases with increasing size of the dataset from which we resample. For a proportion of 0.01, both these approaches perform as well as the parametric simulation. These results reflect what we also have seen in all other scenarios,

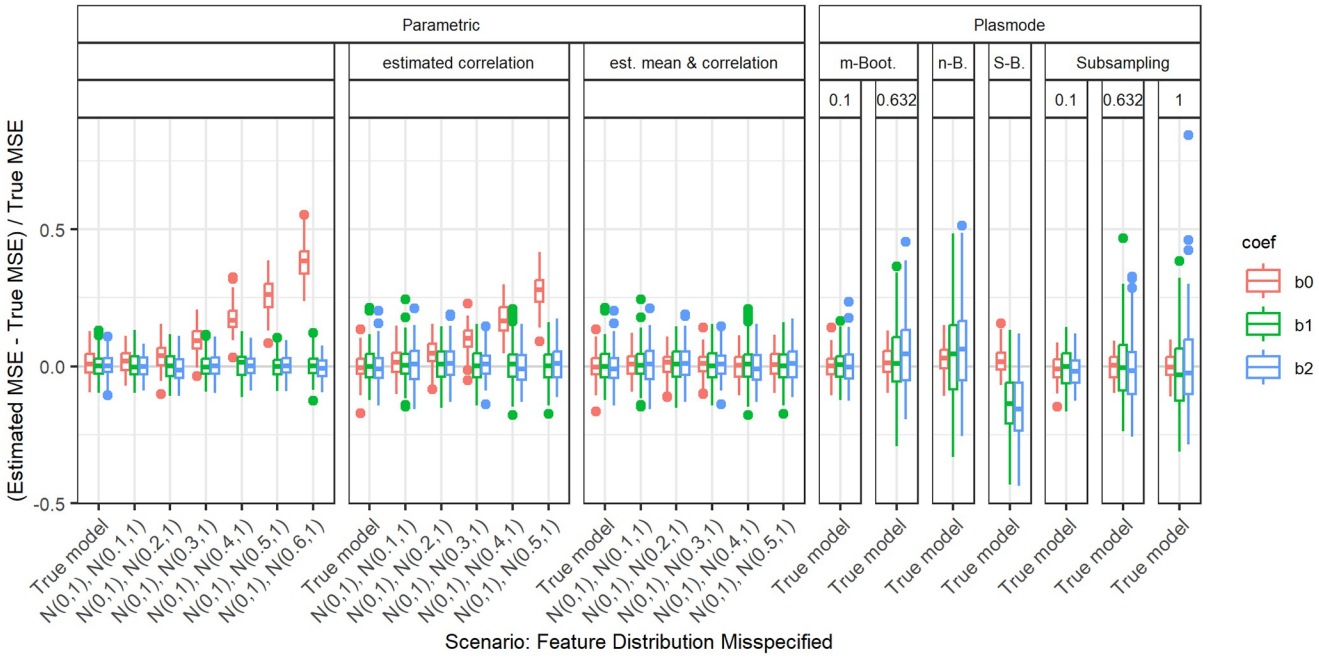


Fig 5. Relative error in MSE estimation for individual coefficients when the assumed mean of the marginal distribution of the second feature in parametric simulation deviates from the true mean, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. N(0,1), N($\mu$,1) denotes that the first feature is generated from a standard normal (truth), and the second feature is generated from a normal distribution with mean μ instead (deviation).

https://doi.org/10.1371/journal.pone.0299989.g005

although for lower p, the differences between the simulation types become very small. It should be noted that in these simulations, subsampling and m out of n Bootstrap require a larger dataset to resample from for lower resampling proportions. This might give them an advantage. Due to its very poor performance, we will exclude the wild Bootstrap from now on. We will also reduce the values of resampling proportions to 0.1 and 0.632 for m out of n Bootstrap and to 0.1, 0.632, and 1 for subsampling for more clarity. The numbers were chosen to represent a relatively low and a relatively high resampling proportion. Moreover, 0.632 has been used in Plasmode simulations, motivated by the expected proportion of non-duplicated observations for n out of n Bootstrap [31].

Deviations from true feature distribution

We will now take a look at the different deviations from the true feature distribution. These only affect the parametric simulation. Since in all cases, different coefficients are affected differently, we always show the individual errors per coefficient. We focus on the case $p = 2$ and $n = 100$ since for this we can still display the errors for individual coefficients in a clear manner. The results can be transferred to higher numbers of features or lower numbers of observations. As expected, the absolute values of the errors are larger for higher values of p or smaller values of n, but the qualitative results are the same. In all cases, we only display the range of deviations that is relevant to the comparison of parametric and Plasmode simulation.

Gaussian with wrong expectation. Fig 5 shows the relative errors in case of deviations from the expectation of the second feature (Feature distribution misspecified N(0,1), N(μ,1), cf. Table 2). We can observe that the second coefficient stays unaffected while the errors for the intercept increase with increasing deviations from the true mean. This result is to be expected, as can be seen by reparametrization. If the truth is $X_2 \sim N(0, 1)$ and we assume

$X_2^a \sim N(\mu, 1),\ \mu > 0$, we can rewrite the resulting linear model using X_2^a instead of X_2 as

$$\begin{aligned} Y &= \beta_0 + X_1\beta_1 + X_2^a\beta_2 + \boldsymbol{\varepsilon} \\ &= \beta_0 + X_1\beta_1 + (X_2 + \mu)\beta_2 + \boldsymbol{\varepsilon} \\ &= \underbrace{\beta_0 + \mu\beta_2}_{=:\beta_0^{new}} + X_1\beta_1 + X_2\beta_2 + \boldsymbol{\varepsilon}. \end{aligned}$$

As $\mu > 0$ and $\beta_2 > 0$ in our case, it holds $\beta_0^{new} > \beta_0$.

The errors in the intercept can be prevented by estimating the mean using a dataset sampled from the true DGP. This leads to slightly increased variance in the errors of the parametric simulation, but stable median errors that are close to zero for all coefficients.

Gaussian with wrong variance. Fig 6 shows the relative errors for deviations from the true variance of the second feature. We see that both slope coefficients are affected. The true MSE is underestimated by the simulation and this underestimation gets worse for increasing (misspecified) variance of the second feature.

Again, this behavior can be prevented by estimating the covariance matrix from a dataset from the true DGP at the cost of slightly increased variation. For estimation of the covariance matrix, the dataset from the true DGP must have sufficiently many observations. Here, we used 1000 observations which is sufficient for $p = 2$, as well as for $p = 50$. Smaller numbers of observations are insufficient for $p = 50$ as can be seen in S2 Fig. When increasing the variance of the second feature, the error in MSE converges to an upper bound corresponding to the true MSE, since the estimated MSE converges to zero for increasing variances. This can lead to problems later on, when we look for the first deviation where the aggregated error for

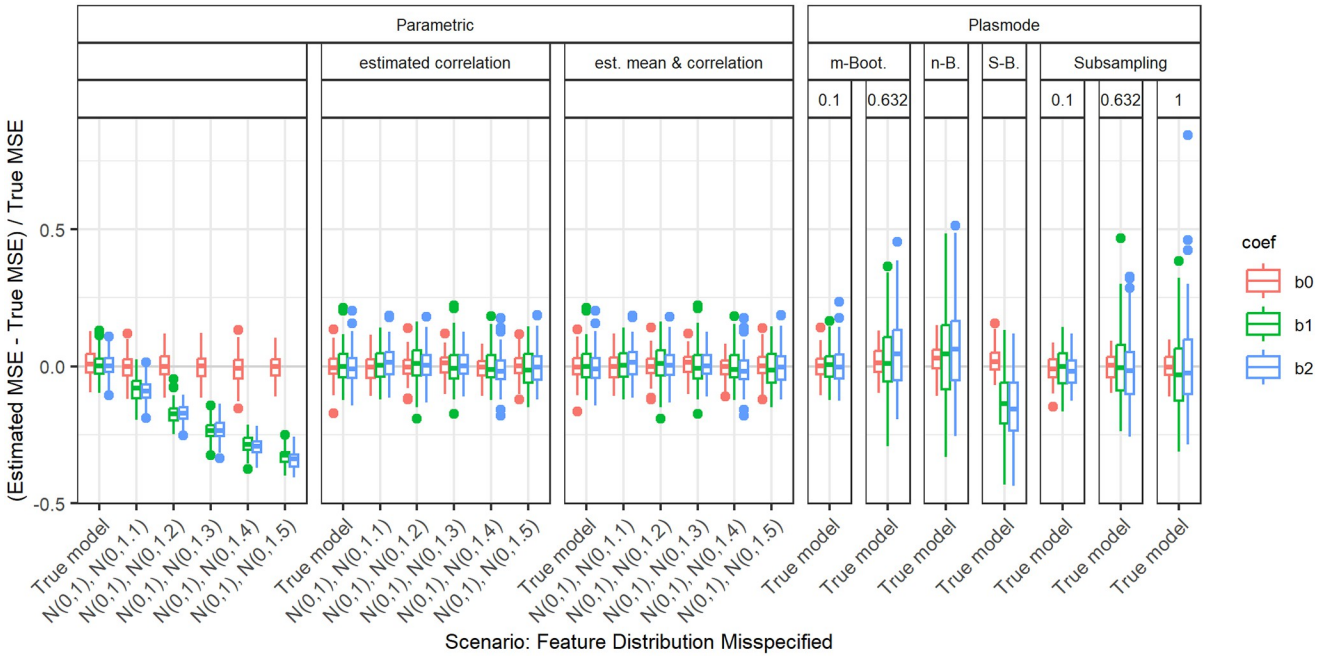


Fig 6. Relative error in MSE estimation for individual coefficients when the assumed variance of the marginal distribution of the second feature in parametric simulation deviates from the true variance, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. N(0,1), N(0,$\sigma^2$) denotes that the first feature is generated from a standard normal (truth), and the second feature is generated from a normal distribution with variance σ^2 instead (deviation).

https://doi.org/10.1371/journal.pone.0299989.g006

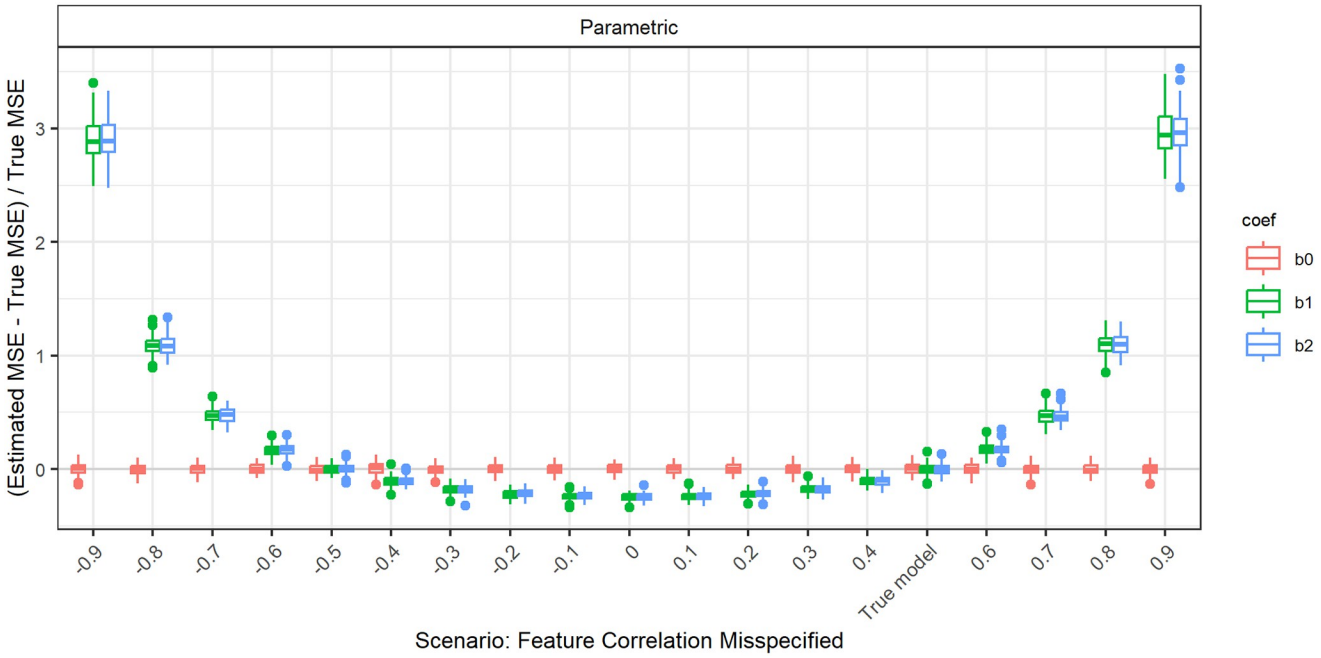


Fig 7. Relative error in MSE estimation for individual coefficients when the assumed correlation of the features in parametric simulation deviates from true correlation, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.5\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g007

parametric simulation exceeds the error for Plasmode simulation. For low p, the upper bound of the error for increasing the feature variance in parametric simulation is still larger than the errors obtained with Plasmode simulation. However, for large p, where Plasmode performs worse, the error reached even with very high values for the variance of the second half of features is smaller than that of some Plasmode types. This is demonstrated in S3 Fig for the case of $p = 50$. There, we show the mean of the relative errors of the coefficients per simulation run. In that case, we do not take the absolute values before averaging, to demonstrate the direction of the errors. In the present case, this is no problem since either the MSEs for all coefficients are overestimated or all are underestimated, so there is no risk of the errors of different coefficients cancelling out in the mean. Decreasing instead of increasing the variance of the second half of features leads to an overestimation of the true MSE and this overestimation is unbounded. Therefore, in settings where the upper bound does not exceed the errors of all Plasmode types, we use decreasing instead of increasing variances, see e.g. S4 Fig.

Gaussian with wrong correlations. The overall influence of misspecifying the pairwise correlations of the features is more easily demonstrated, when the true pairwise correlations are 0.5 instead of 0.2. The relative errors in this case for parametric simulation are shown in Fig 7.

The intercept is unaffected when misspecifying the correlation. For the errors in the slopes, we observe a parabolic shape that intersects with zero at the true correlation of 0.5 and at −0.5. For the MSE estimation, the sign of the correlation does not seem to have any influence, only the absolute value, as the parabolic shape is symmetrical around zero. When overestimating the absolute value of the true correlation, the true MSE is overestimated. For underestimating the absolute value of the true correlation, the true MSE is underestimated.

This pattern is also observed for a true correlation of 0.2 (Fig 8). For the comparison of parametric and Plasmode, we concentrate on assuming a correlation that is higher than the

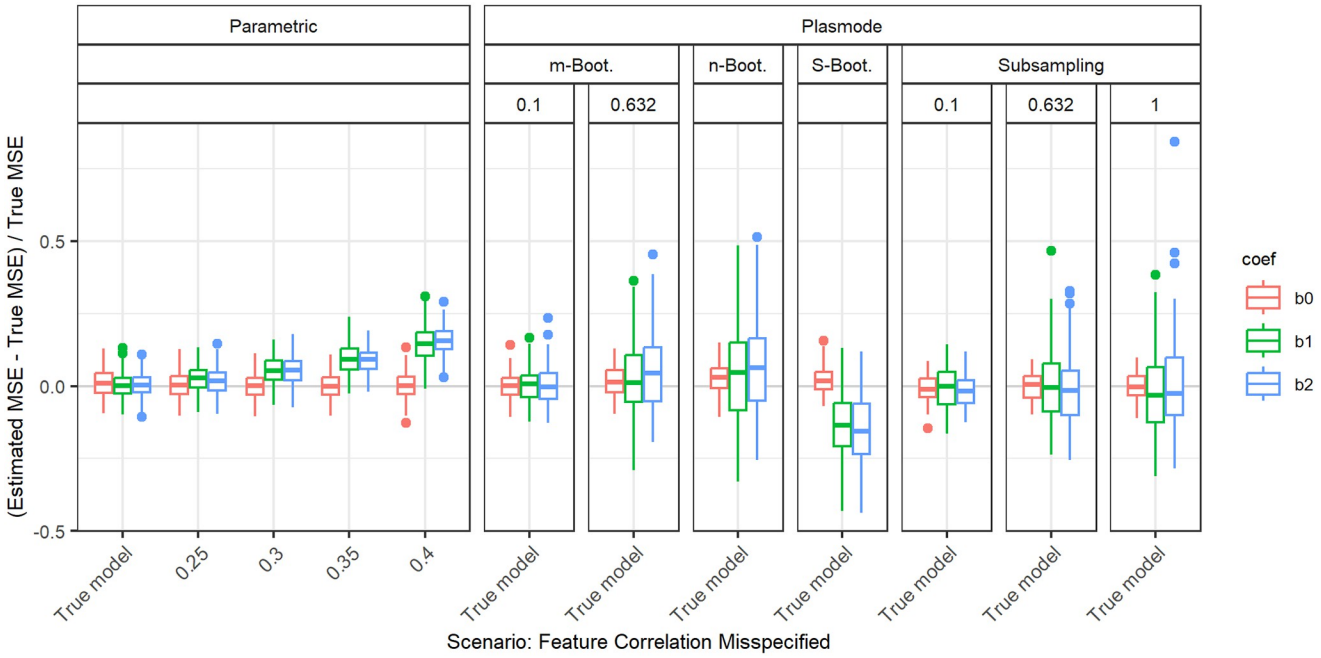


Fig 8. Relative error in MSE estimation for individual coefficients when the assumed correlation of the features in parametric simulation deviates from true correlation, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g008

true correlation, since for these deviations, the errors are monotonously increasing. This can for example be seen in the comparison for true fixed pairwise correlations of 0.2 and $p = 2$, $n = 100$ as shown in Fig 8.

The observed shape is plausible from a theoretical point of view. The MSE of the LSE given X is equal to its variance, as it is unbiased. This variance is given as the diagonal of $\sigma^2(X^TX)^{-1}$. For X drawn from a multivariate normal distribution, i.e. ignoring the intercept term, $(X^TX)^{-1}$ follows an inverse Wishart distribution. Its expectation is given by the inverse covariance matrix Σ^{-1} of this multivariate normal. When explicitly calculating the diagonal values of Σ^{-1} in case of pairwise fixed correlations of ρ, we can see that this expectation depends quadratically on ρ, which matches the observed form.

When the true correlation matrix has a block structure, we observe lower errors for the coefficients at the margins of the blocks if the value of the correlations but not their structure is misspecified (Fig 9). Again, this can be derived theoretically for the very simple case described above when inserting the block diagonal structure for Σ.

Gaussian mixture. Next, we use two different versions of Gaussian mixtures as feature distributions for the second half of the features. With this, not only the parameter but the whole shape of the distribution is altered. For the first type of Gaussian mixture, a proportion of α of the observations stems from a normal distribution with mean 3 and variance 1. This yields a bimodal distribution. For the second type of Gaussian mixture, a proportion of α of the observations stems from a normal distribution with mean 0 and variance 10. This represents a contamination model with outliers. In both cases, the remaining proportion of $1 - \alpha$ stems from the standard normal, in agreement with the true distribution. We always set the marginal distribution of the first feature to a normal that has the same mean and variance as the Gaussian mixture for the second marginal distribution and successively increase the

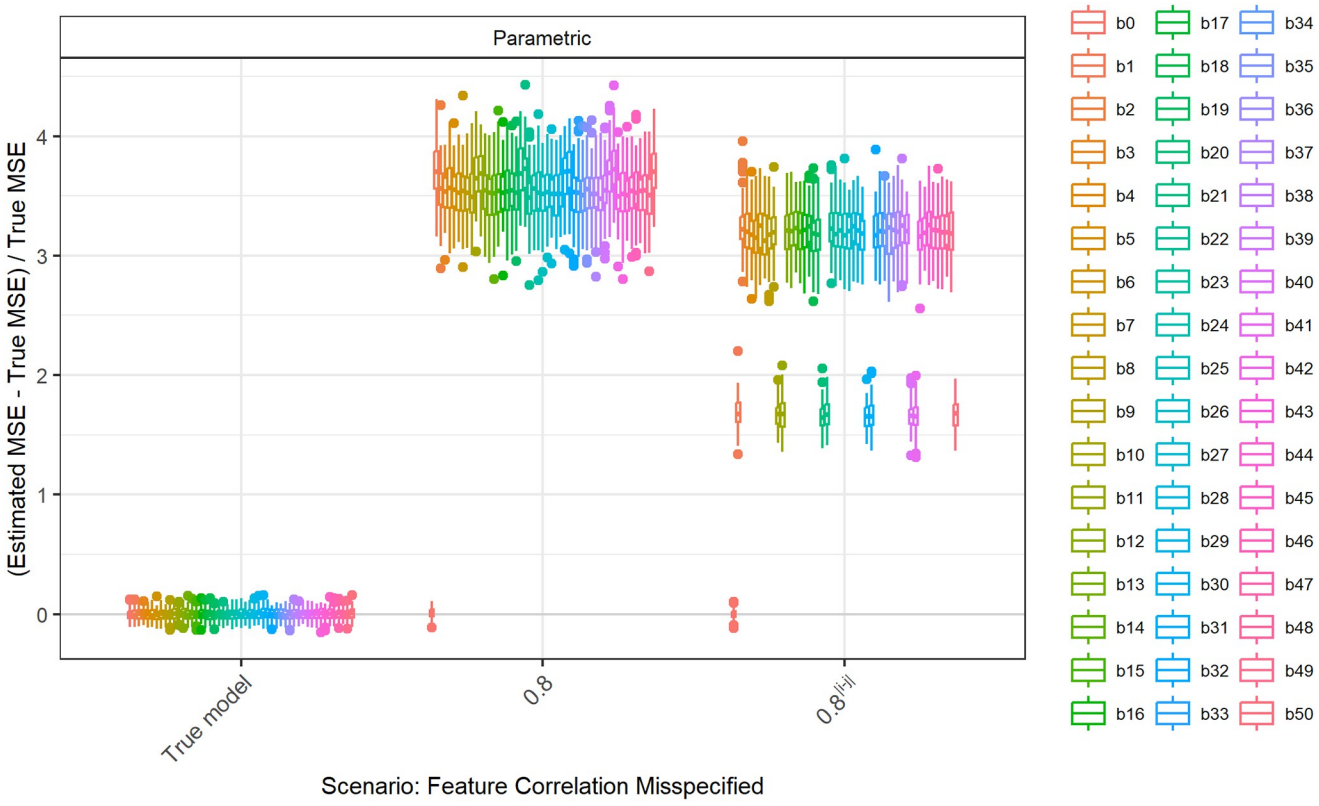


Fig 9. Relative error in MSE estimation for individual coefficients when the assumed correlation of the features in parametric simulation deviates from true correlation, for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2^{|i-j|}$ for ith and jth feature within each of the 5 blocks.

https://doi.org/10.1371/journal.pone.0299989.g009

proportion in the mixing distribution. This enables us to separate the influence of the change in expectation and variance of the distribution from the effect of the bimodality and outliers.

In the bimodal case (Fig 10) we see that with an increasing proportion of observations from the $N(3, 1)$ distribution, the underestimation of the MSE for the corresponding second coefficient also increases. It is still less pronounced than for the first coefficient which corresponds to the normal with wrong expectation and variance. This might be due to the fact that most of the observations in the mixture distribution belong to the true distribution. In the case of a normal with wrong expectation and variance, all observations come from a distribution that differs from the true one.

For the contamination model (Fig 11) we observe the same behavior, but the differences between the coefficients are smaller there.

Log-normal. Fig 12 shows the relative errors for the individual coefficients when the distribution of the second feature is misspecified as log-normal and the distribution of the first feature is misspecified as a normal with matching mean and variance. There is a large overestimation of the MSE for the intercept, while the MSEs for the other coefficients are underestimated. The underestimation is slightly worse for the second coefficient than for the first, so the additional skewness of the log-normal leads to worse MSE estimation compared to a normal with the same mean and variance. The errors in all coefficients for this deviation are considerably higher than the ones of any Plasmode variant that is compared here.

Bernoulli. Fig 13 shows the relative errors for the individual coefficients when the distribution of the second feature is misspecified as Bernoulli and the distribution of the first feature

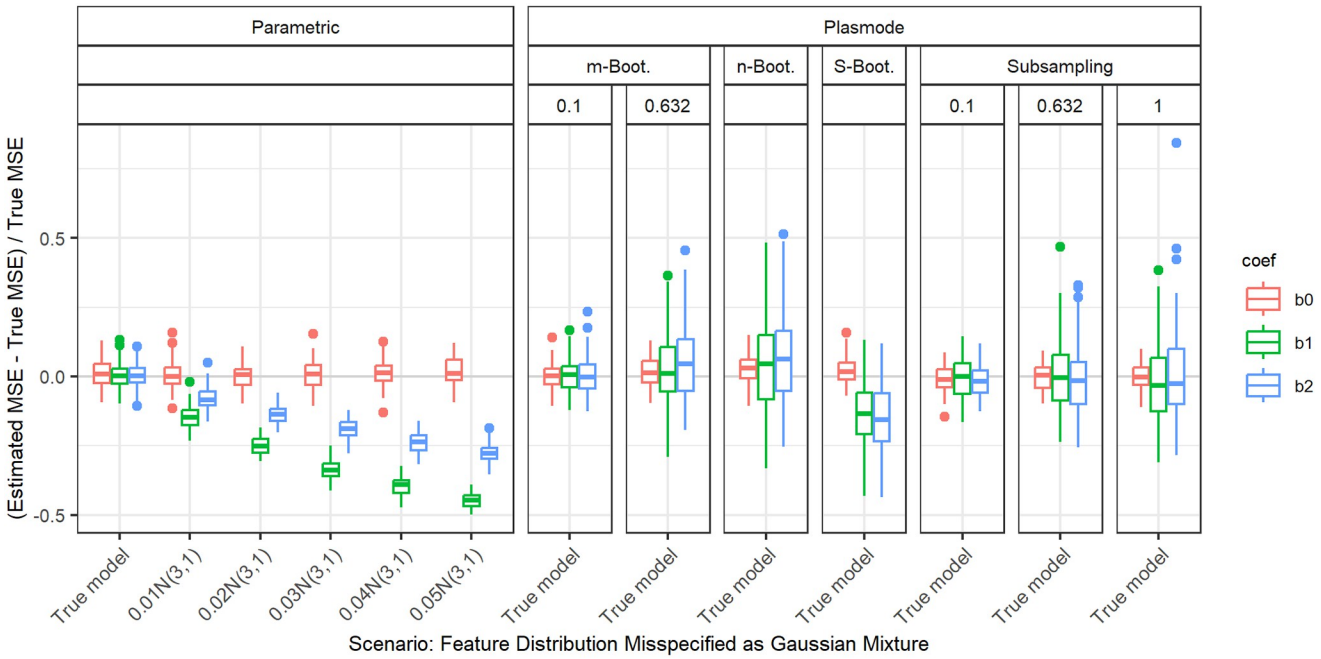


Fig 10. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as Gaussian mixture with increasing proportion of data drawn from Gaussian with different expectations (bimodal distribution). The mean and the variance of the marginal normal distribution of the first feature are set to match those of the second. The mixing proportion is given on the *x*-axis.

https://doi.org/10.1371/journal.pone.0299989.g010

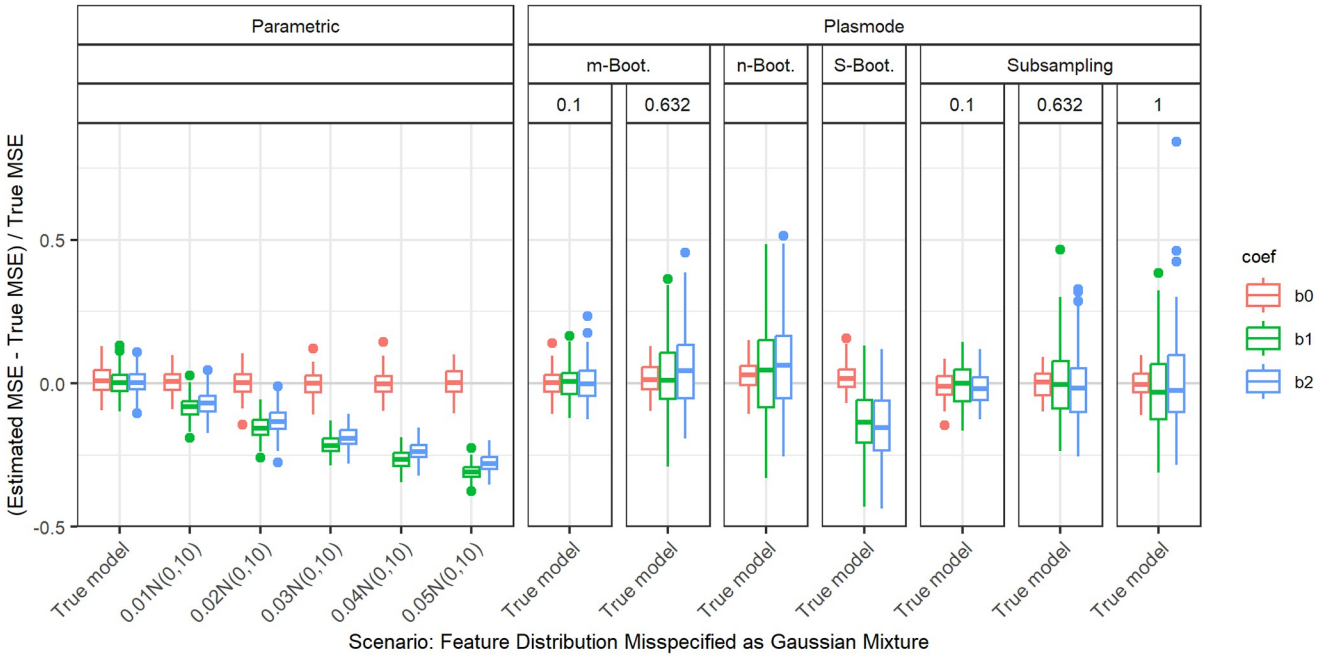


Fig 11. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as Gaussian mixture with increasing proportion of data drawn from Gaussian with different variance (contaminated distribution), for $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. The mean and the variance of the marginal normal distribution of the first feature are set to match those of the second. The mixing proportion is given on the *x*-axis.

https://doi.org/10.1371/journal.pone.0299989.g011

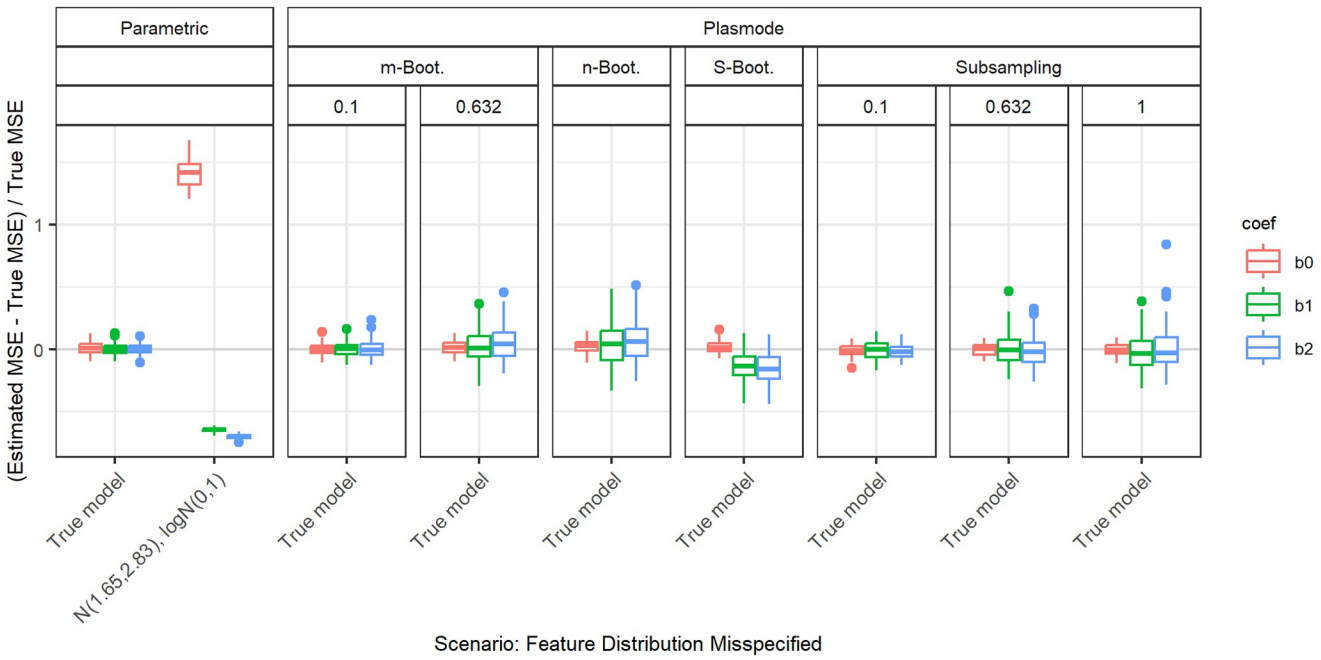


Fig 12. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as log-normal, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. The mean and the variance of the marginal normal distribution of the first feature are set to match those of the second.

https://doi.org/10.1371/journal.pone.0299989.g012

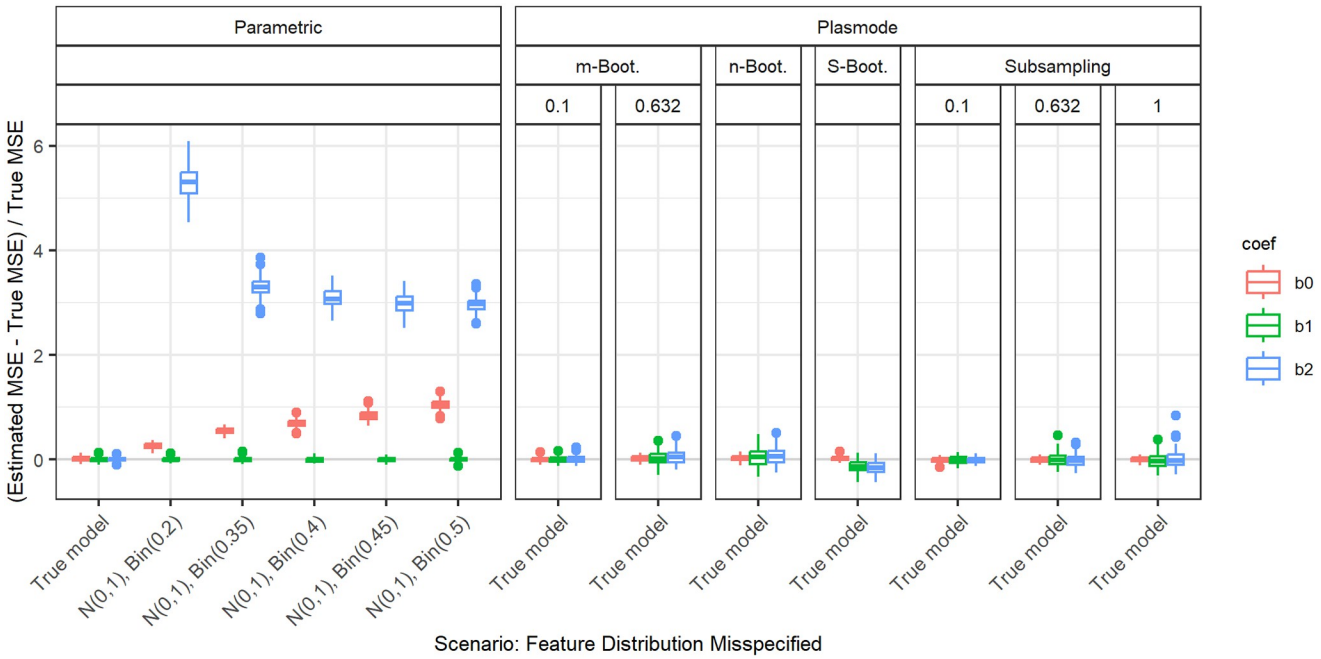


Fig 13. Relative error in MSE estimation for individual coefficients when the assumed marginal distribution of the second feature in parametric simulation is misspecified as Bernoulli with different success probabilities, for $p = 2$, $n = 100$, $\boldsymbol{\beta} = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g013

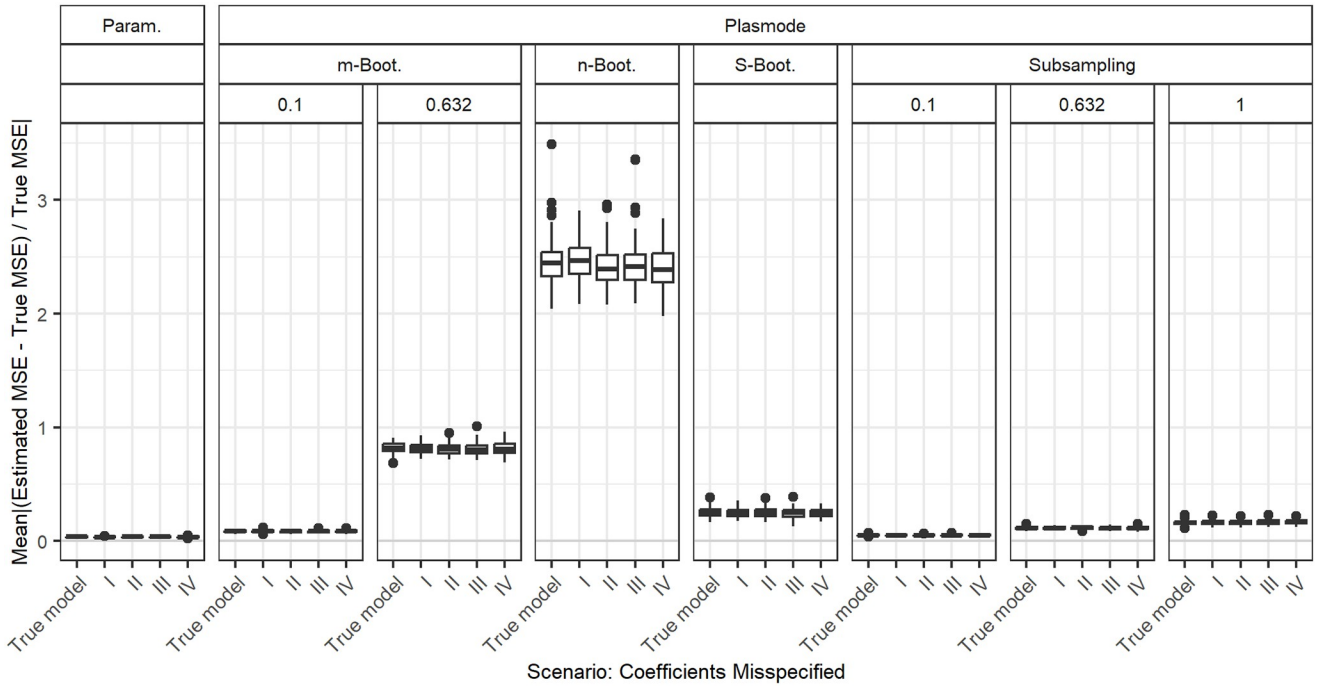


Fig 14. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed coefficients in parametric and Plasmode simulation are misspecified, for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$, $\beta_I = (0, 0.02, \ldots, 1)^T$, $\beta_{II} = 0.05_{51}$, $\beta_{III} = 10_{51}$, $\beta_{IV} = 0_{51}$. Large outliers for n out of n Bootstrap are not displayed.

https://doi.org/10.1371/journal.pone.0299989.g014

is correctly specified as a standard normal. We observe an increasing overestimation of the MSE for the intercept with increasing success probabilities. The MSE for the first coefficient is unaffected. The MSE for the coefficient belonging to the binary feature is also clearly overestimated where the overestimation decreases towards success probabilities of 0.5. The errors for the intercept and the second coefficient for this deviation are considerably higher than the ones of any Plasmode variant that is compared here.

Deviations from true coefficients

Fig 14 shows the aggregated relative errors in MSE estimation for $p = 50$ and fixed correlations of 0.2 for misspecifications of the coefficient vector β. Since the specification of the coefficient vector is part of the OGM, this concerns all types of simulations. For each simulation type, the errors for the misspecified coefficients do not differ from the errors for the true model. Therefore, we conclude that the assumed values for the coefficients do not affect the simulation results. The theoretical MSE formula for given X is also only dependent on σ and X, so independent of β.

Deviations from true error variance

In Fig 15, the aggregated relative errors in MSE estimation for $p = 50$ and fixed correlations of 0.2 for misspecifications of the standard deviation of the error term ε are shown. Here, we use the relative errors directly without taking the absolute value to demonstrate under- and overestimation. This again concerns all types of simulation. In general, for too small error standard deviations, the true MSE is underestimated, and for too large error standard deviations, the true MSE is overestimated. This pattern is visible for nearly all types of simulations. For m out

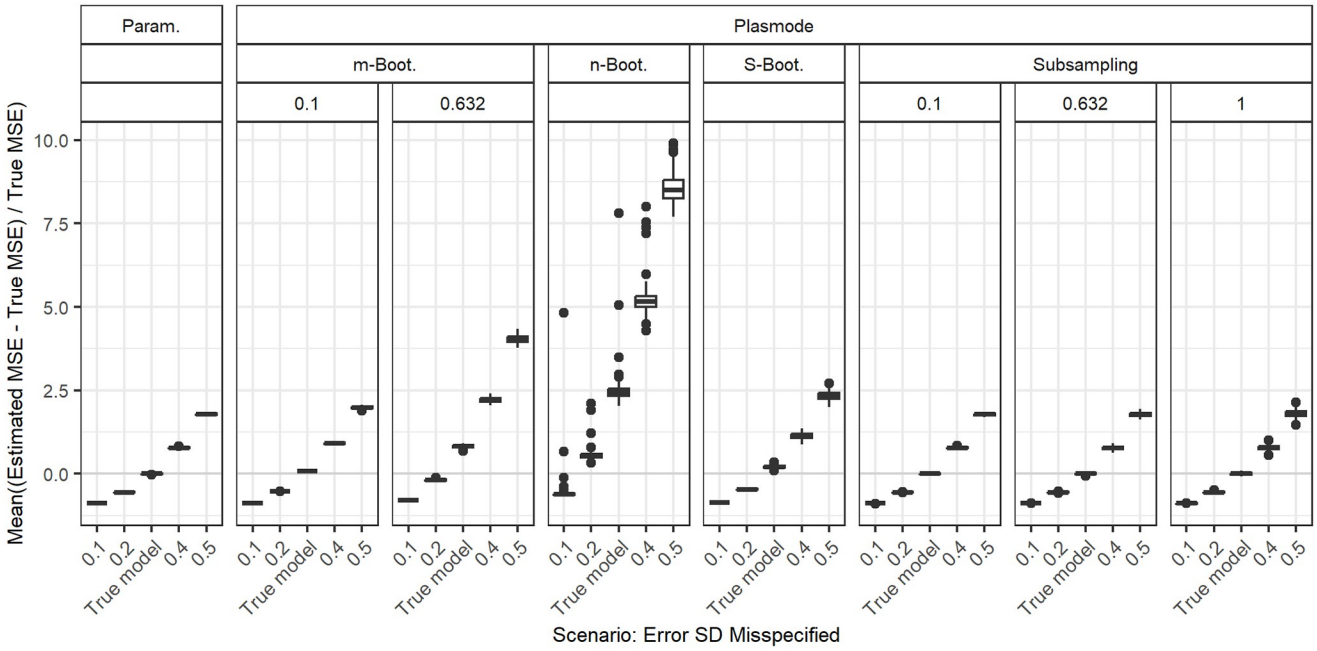


Fig 15. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed error variance in parametric and Plasmode simulation are misspecified for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. Large outliers for n out of n Bootstrap are not displayed.

https://doi.org/10.1371/journal.pone.0299989.g015

of n Bootstrap with large resampling proportions as well as for n out of n Bootstrap, the MSE is overestimated even for the true model, and the errors for other values of the error standard deviation are shifted up accordingly. This leads to values closest to zero for too small error standard deviations. In all cases, the variability of the errors increases with increasing error standard deviation. We observe the same ordering that has already resulted for the true model (see Fig 4) when comparing the errors from different simulation types for misspecified error standard deviations.

Deviations from true error distribution

In Fig 16, the aggregated relative errors in MSE estimation for $p = 50$ and fixed correlations of 0.2 for misspecifications of the distribution of the error term are shown. There are two types of misspecifications that we compare. We use t-distributed errors as an example of a heavier-tailed distribution and χ^2-distributed errors as an example of a skewed distribution. Both are scaled and shifted in a way that the errors still have zero expectation and a standard deviation of 0.3. Overall, the distribution of the errors does not seem to have any influence on the error in MSE estimation as long as the error standard deviation and zero mean are preserved.

True DGP: Correlation estimated from real data

We now analyze the results for the scenarios where the true correlation matrix is estimated from a real dataset. In the following, we only discuss the results that differ from those for the more simple correlation structures we looked at before. These are all deviations that do not alter the correlation matrix. For deviations from the true correlations, it gets more complicated. In the case of small correlations which differ little, the results are still similar to those that we saw before. For example, Fig 17 shows the results for the correlation estimated from

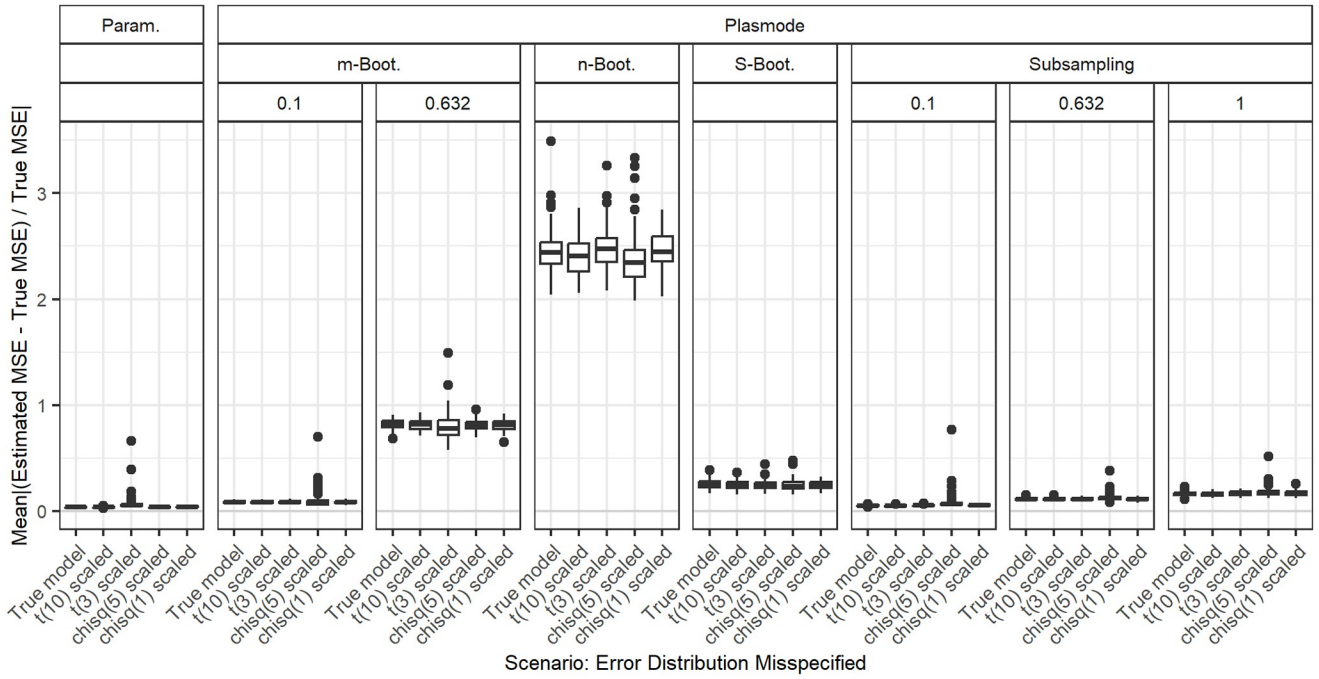


Fig 16. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed error distributions in parametric and Plasmode simulation are misspecified, for $p = 50$, $n = 100$, $\boldsymbol{\beta} = \mathbf{1}_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. Large outliers for n out of n Bootstrap are not displayed.

https://doi.org/10.1371/journal.pone.0299989.g016

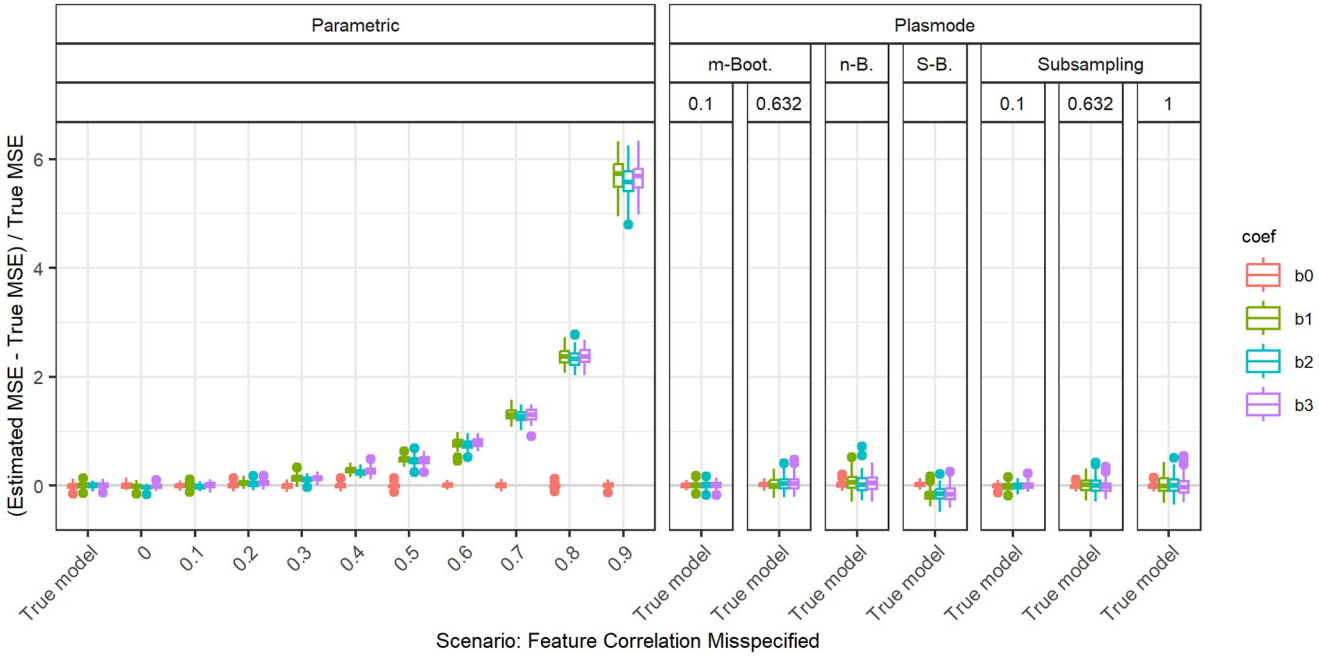


Fig 17. Absolute value of relative error in MSE estimation for individual coefficients when the assumed feature correlation matrix in parametric simulation is misspecified. True correlation matrix is estimated from the benchmark dataset quake ($p = 3$, $n = 100$, $\beta = \mathbf{1}_4$, $\sigma = 0.3$).

https://doi.org/10.1371/journal.pone.0299989.g017

the dataset quake. The true pairwise correlations are $\text{Cor}(X_1, X_2) = -0.1286$, $\text{Cor}(X_1, X_3) = -0.0151$, and $\text{Cor}(X_2, X_3) = 0.1353$. The results look similar to those we saw before for fixed correlations of 0.2. On the other hand, for the other datasets, the estimated pairwise correlations show higher variation, which means that no fixed value can be used to approximate all correlations simultaneously in a good way. This is for example clearly visible in Fig 18 for the correlation matrix estimated from the dataset wine_quality. For each choice of fixed pairwise correlation, there are some coefficients with very large relative errors. This can also lead to errors showing a pattern that differs from the parabolic shape we observed before (Fig 7), as can be seen in Fig 19 for the dataset Yolanda. For those cases where no constant correlation approximates all real correlations well, many of the Plasmode variants outperform parametric simulation for all assumed oversimplified correlation structures. A possible cure for parametric simulation would be to estimate the correlation structure from real data which—in this case—corresponds to the true model. Overall, assuming some simple correlation structure, like often done in parametric simulations, might lead to high errors in the estimation of the MSE in cases where the true correlation structure is more complicated. To correctly guess this correlation structure is highly unlikely, and it might even be impossible to specify complicated correlation structures in high-dimensional settings.

Size of resampled datasets

Until now, we have always compared simulations that use the same number of observations, which leads to differently sized datasets from which the Plasmode data is resampled. This might seem unintuitive, but is necessary to ensure a fairer comparison of the simulation methods since the true MSE that the estimations are compared to, is monotonously decreasing in the number of observations in the dataset. Therefore, if we set the size of the dataset that we

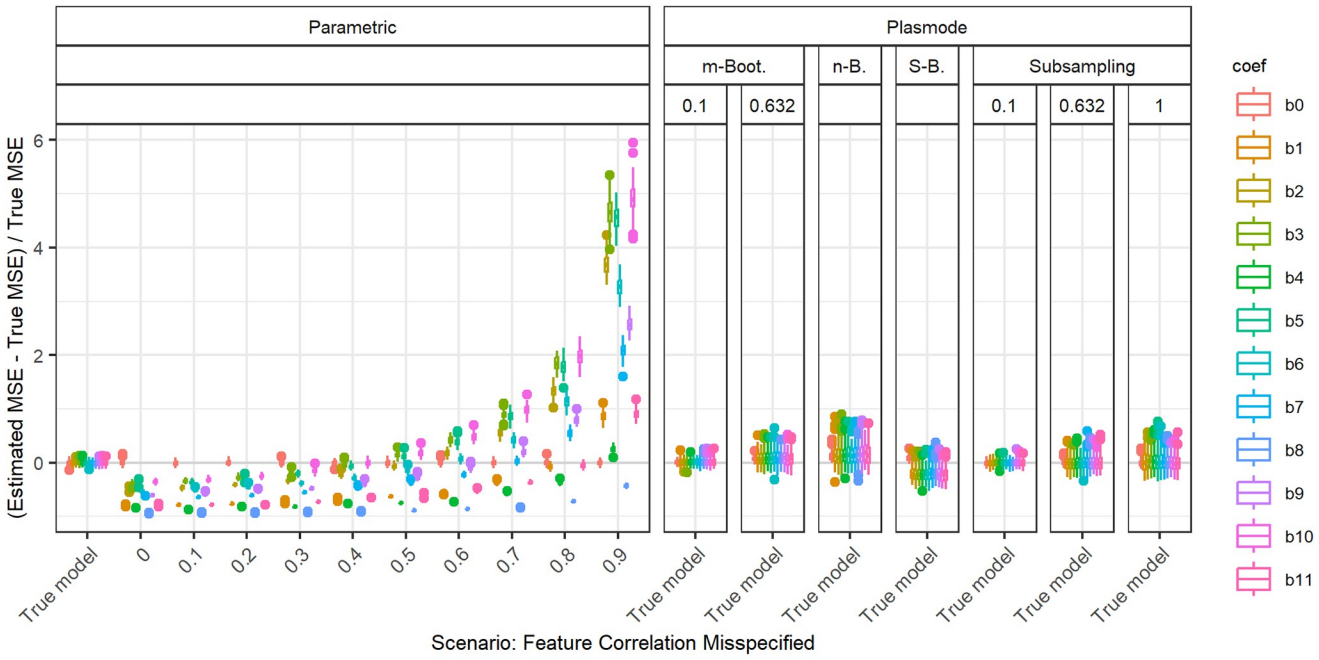


Fig 18. Absolute value of relative error in MSE estimation for individual coefficients when the assumed feature correlation matrix in parametric simulation is misspecified. True correlation matrix is estimated from benchmark dataset wine_quality ($p = 11$, $n = 100$, $\beta = \mathbf{1}_{12}$, $\sigma = 0.3$).

https://doi.org/10.1371/journal.pone.0299989.g018

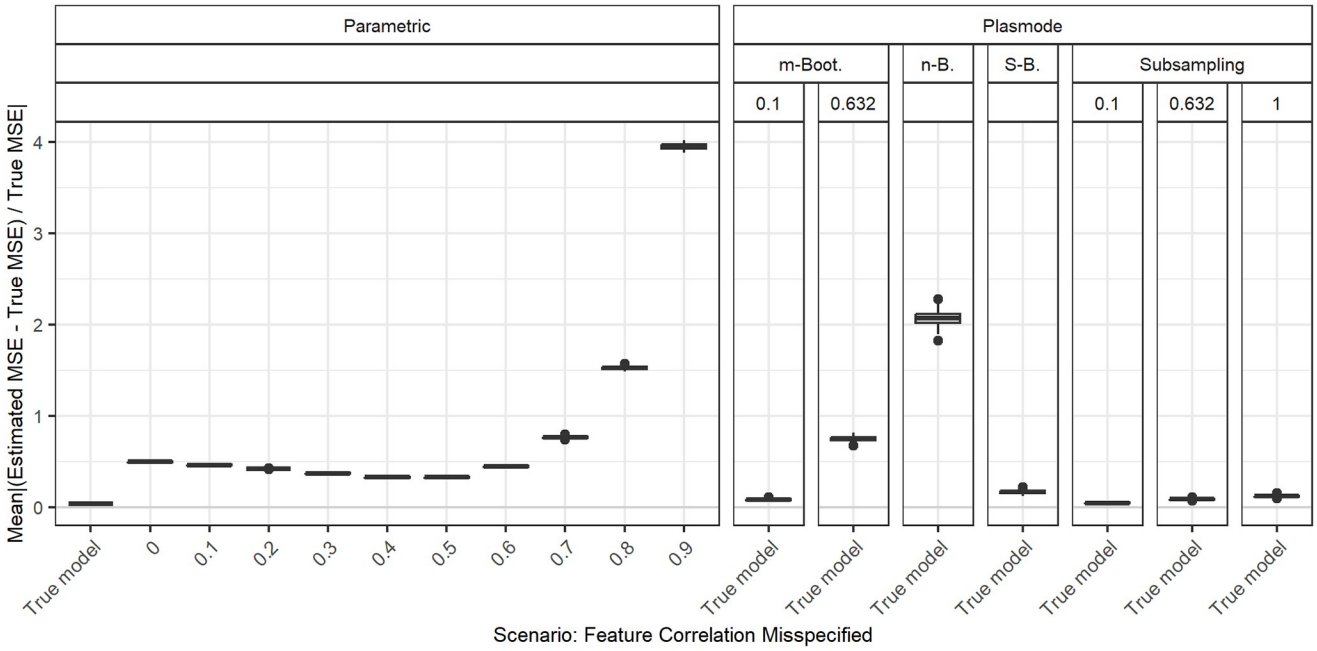


Fig 19. Absolute value of relative error in MSE estimation averaged over individual coefficients when the assumed feature correlation matrix in parametric simulation is misspecified. True correlation matrix is estimated from benchmark dataset Yolanda ($p = 100$, $n = 200$, $\beta = \mathbf{1}_{101}$, $\sigma = 0.3$).

https://doi.org/10.1371/journal.pone.0299989.g019

are resampling from to 100 and resample smaller datasets from this, the MSE will always be overestimated, even for the true model. This means that if we want to estimate the MSE for datasets of a certain size n, we have to use datasets of that exact size in our simulations. However, it might be unrealistic that we have a dataset of the correct size at hand to resample from for our simulation. For example, if we use simulation to estimate a quantity that cannot be estimated directly from the data since it depends on unknown parameters (e.g. the bias of an estimator), we might have a concrete dataset at hand for which we want to estimate this quantity. In this case, Plasmode would be a natural choice and since the number of observations is limited, we might use resampled datasets of smaller size to estimate the quantity for the whole dataset. We now discuss the results for this case for $p = 10$ for the true model. For $p = 2$, differences between the resampling methods are very small anyway. For $p = 50$, it will be hard to differentiate between the errors occurring due to the differently sized datasets and the errors caused by approaching the boundary of identifiability. Fig 20 shows the results for the different Bootstrap methods compared to parametric simulation for differing sizes of datasets resampled from a dataset of size 100. For comparison, the case of resampling 100 out of 158 observations that has been used in the analysis so far for a resampling proportion of 0.632 is also included. The estimated MSEs are compared to the true MSE for $n = 100$ in all cases. Higher errors are observed for smaller sizes of the resampled dataset. The smallest errors are observed for subsampling with the subsampling proportion approaching the number of observations in the dataset. So in the case where the number of observations is limited to the number of observations that we are interested in, it might even be the best choice to do no resampling at all and just generate different responses for the MSE estimation. It should be noted that when fixing the size of the dataset to resample from, the n out of n Bootstrap performs comparably well. A reason for this might be that it uses a dataset of size 100 for estimating the MSE. Therefore, no errors occur due to the dependency of the MSE on n. Moreover, the n out of n Bootstrap can

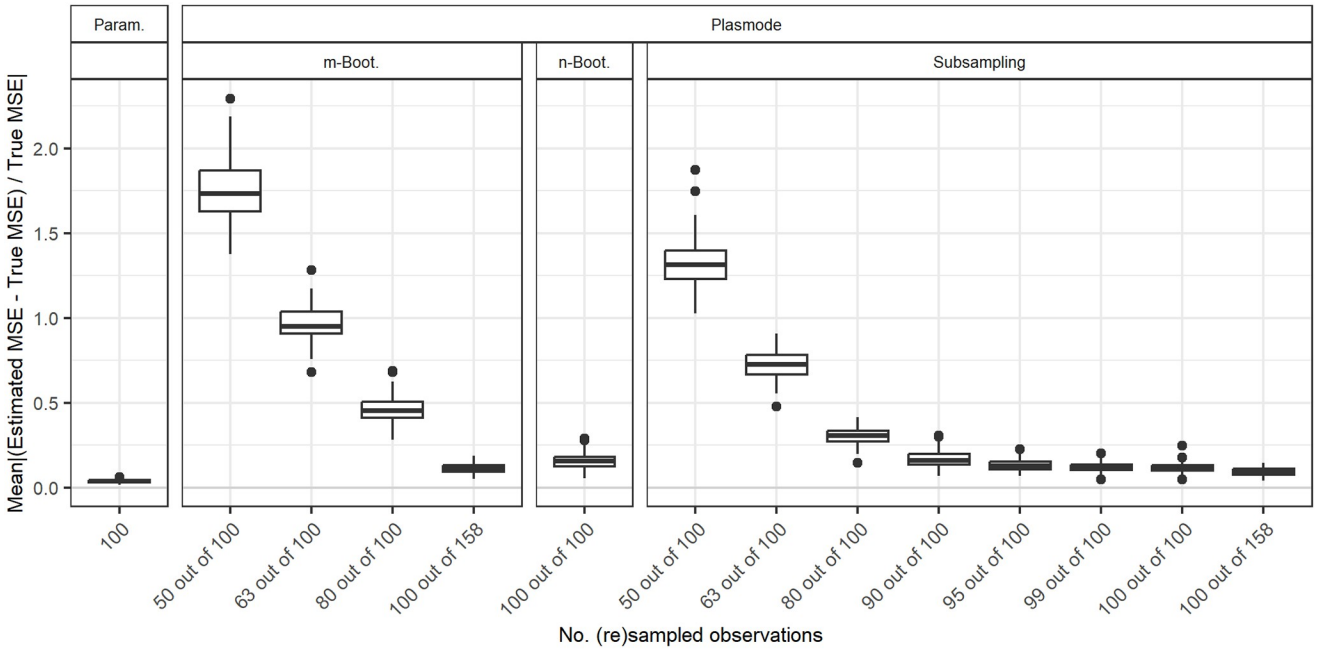


Fig 20. Comparison of different resampling types for different numbers of observations resampled from a dataset with 100 observations. Absolute value of relative error in MSE estimation averaged over individual coefficients when the true model is assumed in parametric and Plasmode simulation, for $p = 10$, $n = 100$, $\beta = \mathbf{1}_{11}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.

https://doi.org/10.1371/journal.pone.0299989.g020

use the dataset more efficiently since it uses more samples for the MSE estimation than subsampling or the *m* out of *n* Bootstrap with lower resampling proportions.

Conclusions and recommendations

In the following, we summarize what we have learned from the comparisons that we performed. First, we provide some general insights. Then, we present detailed comparisons, for which type of deviations from the data-generating process Plasmode was superior to parametric simulation in our analyses.

General insights

We looked at different true data-generating processes (DGP) and deviations from those for the estimation of the MSE of the least squares estimator (LSE) in linear regression to compare how well different simulation types perform in this case. Overall, we saw that if there is no deviation from the true scenario, parametric simulation outperforms all Plasmode simulations. The same holds for deviations that affect parametric as well as Plasmode, i.e. deviations from the outcome generating model (OGM), given that the DGP used for parametric simulation is close to the truth. We saw that the misspecification of the coefficients and of the error distribution (as long as expectation and variance are kept) does not have any effect on the quality of the MSE estimation while the misspecification of the error standard deviation does have an effect.

Misspecifications of the DGP only affect parametric simulation. For all kinds of misspecifications of the DGP in parametric simulation (misspecification of expectation, variance, correlation, whole distribution), parametric simulation can become worse than Plasmode. The degree of misspecification needed for Plasmode to be superior depends on the type of

misspecification, the resampling method used in the context of Plasmode that we compare with, and on the number of observations n and the number of features p. A detailed analysis of the degree of misspecification needed for Plasmode to be superior is given in Subsection 2.

Within the different resampling strategies for Plasmode simulations we observed that in general, Wild Bootstrap performed worst, followed by n out of n Bootstrap. m out of n Bootstrap performed better than n out of n and subsampling usually performed best. For both m out of n Bootstrap and subsampling, smaller resampling proportions are favorable. This means that for a fixed number of subsampled observations n of interest, larger datasets to resample from are required. Smoothed Bootstrap usually performs worse than subsampling and even than no resampling (subsampling proportion of one), but better than m out of n Bootstrap with moderate resampling rates, i.e. rates larger than 0.5. When the number of observations for resampling is limited to the number of observations that we are interested in, we are restricted to n out of n Bootstrap, Smoothed Bootstrap, Wild Bootstrap, no resampling at all (i.e. subsampling with the proportion of one), or resampling a dataset of smaller size for Plasmode. Our analyses suggest that no resampling at all or subsampling with a subsampling proportion very close to one might be the best choice in this case. This is due to the dependence of the MSE on the number of observations, which leads to biased estimates of the MSE if the number of observations used for the simulation differs from the number of observations of interest.

Detailed comparisons

Table 3 presents the values for each scenario and deviation at which certain types of Plasmode simulation are superior to parametric simulation. As discussed before, this is only applicable to deviations regarding the data-generating process (DGP). The numbers given in the Plasmode columns are calculated as follows. For the given scenario, deviation and Plasmode type, the deviations are ordered increasingly. Then, the first deviation for which the median aggregated relative error of parametric is higher than that for the Plasmode type is identified. These values correspond to the medians in the aggregated boxplots. For example in the first row, the case of $p = 2$, $n = 100$ and fixed pairwise correlations of 0.2 is analyzed for deviations of the assumed expected value for the second feature. The true expectation is 0. Plasmode with m out of n Bootstrap or subsampling with a resampling proportion of 0.1 is superior to parametric simulation for assumed expectations of 0.25 and higher. Plasmode with m out of n Bootstrap or subsampling with a resampling proportion of 0.632 is only superior for assumed expectations of 0.4 and higher, n out of n Bootstrap for values of 0.5 and higher, Smoothed Bootstrap for values of 0.55 and higher, and Plasmode without resampling (subsampling with proportion of 1) for values of 0.45 and higher.

When using correlation matrices estimated from real datasets, the order for the deviations in the correlations is unclear, as discussed before. Therefore, they are excluded from the comparison. Also, in all cases, assuming log-normal or binary data instead of normal data is worse than all Plasmode variants and therefore also excluded.

For these analyses, in the parametric simulations, the expectations and high variances were increased in steps of 0.05, and the low variances were decreased in steps of 0.1. The mixing proportion for Gaussian mixtures and the pairwise correlations were increased in steps of 0.01.

For $p = 50$ and assuming Gaussian mixtures, in some cases even a proportion of 100% data for the second half of features coming from the wrong distribution is not sufficient for Plasmode to be superior, as can be concluded from the values found for deviating expectations and variances. The corresponding entries in Table 3 are left empty in these cases.

Table 3. Smallest deviations in parametric simulations for which Plasmode simulation is superior to parametric simulation. p denotes the number of features, n the number of observations. True ρ gives the true correlation structure, scenario type the type of deviation and true value the true parameter value that the deviation refers to.

p	n	True ρ	Scenario type	True value	m-Bootstrap		n-Bootstrap	Smoothed Bootstrap	Subsampling		No resampling
					0.1	0.632			0.1	0.632	
2	100	0.2	Expectation of 2nd feature misspecified	0	0.25	0.4	0.5	0.55	0.25	0.4	0.45
2	100	0.2	Variance of 2nd feature misspecified	1	1.05	1.15	1.15	1.2	1.1	1.1	1.15
2	100	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
2	100	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.01	0.02	0.02	0.01	0.01	0.01
2	100	0.2	Feature correlation misspecified	N(0,1)	0.28	0.35	0.39	0.4	0.29	0.35	0.36
2	50	0.2	Expectation of 2nd feature misspecified	0	0.3	0.5	0.55	0.6	0.3	0.5	0.55
2	50	0.2	Variance of 2nd feature misspecified	1	1.1	1.2	1.25	1.3	1.1	1.2	1.2
2	50	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.03	0.03	0.04	0.01	0.03	0.03
2	50	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
2	50	0.2	Feature correlation misspecified	N(0,1)	0.33	0.41	0.41	0.41	0.29	0.39	0.41
2	50	0.5	Expectation of 2nd feature misspecified	0	0.25	0.35	0.4	0.5	0.25	0.35	0.4
2	50	0.5	Variance of 2nd feature misspecified	1	1.05	1.15	1.15	1.25	1.05	1.1	1.15
2	50	0.5	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
2	50	0.5	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.02	0.02	0.03	0.01	0.01	0.02
2	50	0.5	Feature correlation misspecified	N(0,1)	0.54	0.57	0.57	0.61	0.54	0.56	0.57
10	100	0.2	Expectation of 2nd half of features misspecified	0	0.25	0.45	0.6	0.7	0.25	0.4	0.5
10	100	0.2	Variance of 2nd half of features misspecified	1	1.1	1.25	1.4	1.55	1.1	1.2	1.25
10	100	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.03	0.04	0.01	0.02	0.02
10	100	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.02	0.02	0.03	0.01	0.01	0.02
10	100	0.2	Feature correlation misspecified	N(0,1)	0.24	0.29	0.33	0.36	0.24	0.28	0.3
10	100	0.2	Feature correlation misspecified $\rho^{\lvert i-j\rvert}$	N(0,1)	0.24	0.38	0.41	0.44	0.24	0.36	0.39
10	50	0.2	Expectation of 2nd half of features misspecified	0	0.3	0.7	0.9	0.7	0.3	0.55	0.65
10	50	0.2	Variance of 2nd half of features misspecified	1	1.1	1.55	2.45	1.55	1.1	1.3	1.45
10	50	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.05	0.08	0.04	0.01	0.03	0.04
10	50	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.03	0.06	0.03	0.01	0.02	0.03
10	50	0.2	Feature correlation misspecified	N(0,1)	0.26	0.36	0.43	0.36	0.25	0.31	0.34
10	50	0.2	Feature correlation misspecified $\rho^{\lvert i-j\rvert}$	N(0,1)	0.33	0.44	0.5	0.44	0.22	0.39	0.42
50	100	0.2	Expectation of 2nd half of features misspecified	0	0.4	1.55	2.7	0.8	0.25	0.5	0.65
50	100	0.2	Variance of 2nd half of features misspecified (too small)	1	0.88	0.38	0.17	0.69	0.94	0.84	0.77
50	100	0.2	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.02	0.57		0.05	0.01	0.02	0.03
50	100	0.2	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.98		0.04	0.01	0.02	0.02
50	100	0.2	Feature correlation misspecified	N(0,1)	0.27	0.57	0.78	0.37	0.24	0.29	0.32
50	100	0.2	Feature correlation misspecified $\rho^{\lvert i-j\rvert}$	N(0,1)	0.25	0.62	0.79	0.46	0.34	0.21	0.42
50	100	$0.2^{\lvert i-j\rvert}$ in 5 blocks	Expectation of 2nd half of features misspecified	0	0.4	2.05	2.6	0.8	0.25	0.5	0.6
50	100	$0.2^{\lvert i-j\rvert}$ in 5 blocks	Variance of 2nd half of features misspecified (too small)	1	0.88	0.39	0.17	0.68	0.94	0.84	0.77
50	100	$0.2^{\lvert i-j\rvert}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.02	0.51		0.05	0.01	0.02	0.03
50	100	$0.2^{\lvert i-j\rvert}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.26		0.03	0.01	0.02	0.02
50	100	$0.2^{\lvert i-j\rvert}$ in 5 blocks	Feature correlation misspecified	N(0,1)	0.2	0.5	0.74	0.28	0.2	0.2	0.22
50	100	$0.2^{\lvert i-j\rvert}$ in 5 blocks	Feature correlation misspecified $\rho^{\lvert i-j\rvert}$	$0.2^{\lvert i-j\rvert}$	0.3	0.59	0.78	0.41	0.25	0.32	0.36
50	100	$0.5^{\lvert i-j\rvert}$ in 5 blocks	Expectation of 2nd half of features misspecified	0	0.5	2.05	3.4	2.05	0.3	0.6	0.8
50	100	$0.5^{\lvert i-j\rvert}$ in 5 blocks	Variance of 2nd half of features misspecified (too small)	1	0.88	0.39	0.17	0.68	0.94	0.84	0.78
50	100	$0.5^{\lvert i-j\rvert}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.02	0.43		0.05	0.01	0.02	0.04
50	100	$0.5^{\lvert i-j\rvert}$ in 5 blocks	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.26		0.02	0.01	0.01	0.01
50	100	$0.5^{\lvert i-j\rvert}$ in 5 blocks	Feature correlation misspecified	N(0,1)	0.5	0.67	0.83	0.51	0.5	0.5	0.5
50	100	$0.5^{\lvert i-j\rvert}$ in 5 blocks	Feature correlation misspecified $\rho^{\lvert i-j\rvert}$	$0.5^{\lvert i-j\rvert}$	0.54	0.72	0.85	0.6	0.53	0.55	0.58
3	100	quake	Expectation of 2nd half of features misspecified	0	0.3	0.5	1	1	0.3	0.45	1
3	100	quake	Variance of 2nd half of features misspecified (too small)	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99
3	100	quake	Distribution misspecified: Gaussian mixture with N(0,10)	0	0.01	0.02	0.02	0.03	0.01	0.02	0.02
3	100	quake	Distribution misspecified: Gaussian mixture with N(3,1)	0	0.01	0.01	0.02	0.02	0.01	0.01	0.01

https://doi.org/10.1371/journal.pone.0299989.t003

Summary and discussion

We performed a simulation study to compare the performance of parametric and Plasmode simulation in the context of MSE estimation for the least squares estimator (LSE) in the linear regression model. For parametric simulation, artificial data is generated according to a fully user-specified data-generating process (DGP) for generating the feature data and an outcome-generating model (OGM) for generating the outcome variable. In contrast to that, in Plasmode simulation the feature data is generated by resampling from a real-life dataset and only the OGM has to be specified. For comparing the two approaches, we need control of the true underlying DGP and OGM. We used different true DGPs and OGMs. Since the true DGP and OGM are unknown in practice, they must be specified when conducting a simulation study. For Plasmode simulation the DGP is implicitly given by the chosen dataset. This specification is likely a deviation from the truth. Therefore, we examined the influence of different deviations on both types of simulation studies. Note that for Plasmode, there is no explicit deviation from the DGP. When resampling from a dataset, one samples from the empirical DGP which ideally converges to the true DGP.

Within Plasmode simulations, we compared different resampling strategies, namely n out of n Bootstrap, m out of n Bootstrap, subsampling, smoothed Bootstrap, and wild Bootstrap, and where applicable also different resampling proportions. Each simulation strategy was evaluated based on the differences between the MSEs estimated using the respective method and the true MSEs. If the true DGP and OGM are known, it is obvious that parametric simulation is the optimal choice as long as drawing from the true DGP and OGM is feasible. However, in reality, the true DGP and OGM are unknown and can at best be approximated using expert knowledge. In Plasmode simulations, as long as a dataset from the DGP of interest is given, only the OGM has to be specified. Therefore, our aim was to find out

1. How much the DGP chosen in the parametric simulation can deviate from the truth before the parametric simulation becomes worse than Plasmode simulation.
2. How deviations of the chosen OGM from the true OGM affect both parametric and Plasmode simulations.
3. How the choice of the resampling type affects the Plasmode simulation.

In general, we observed that parametric simulation is superior to Plasmode in all situations where the DGP is specified correctly, i.e. for the true situation or deviations from the true OGM only. For deviations from the DGP in parametric simulation, it depends on the kind of deviation, the degree of deviation, the number of observations, and especially on the number of features in the dataset and the type of resampling used for the Plasmode simulation. For very small deviations, parametric simulations usually remain superior. For low numbers of observations, or especially for higher numbers of features, the performance of Plasmode simulation decreases more drastically than that of parametric simulation both in terms of the median difference between the estimated and true MSE and the variation of the estimated MSE. This means that the deviations from the true DGP in the assumptions of parametric simulation have to be larger for Plasmode to be superior. The effect is more pronounced when using resampling strategies with replacement and a high resampling proportion. A reason for this might be that in these cases, the number of unique observations is lower. Therefore less information is contained in the data, so the variance and consequently the MSE of the estimator is inflated. On the other hand, there are certain settings where Plasmode was always superior to parametric simulation in our study, such as when the DGP was severely misspecified, e.g. when using binary instead of standard normal features.

The effect that Plasmode notably overestimates the true MSE for increasing p, especially for resampling with replacement and high resampling proportions, might be a property of the chosen simulation setup. It is known that using Bootstrap to estimate the variance of the LSE in linear regression models for $p/n \to \kappa \in (0, 1)$ can lead to severe overestimation of the true variance. For n out of n Bootstrap and features sampled from a multivariate standard Gaussian distribution, this property was formally shown and additionally demonstrated via simulation in [32]. Overestimation of the variance implies overestimation of the MSE, so the arguments made in [32] might in part explain the bad performance of Plasmode simulations that we observed. The authors also derived an overestimation of the variance by Jackknifing which is similar to subsampling with resampling proportions very close to one.

If the distribution class of the features was misspecified as log-normal or even Bernoulli instead of normal or if the true correlation matrix of the features is more complex and parametric simulation uses an oversimplified approximation for it, all types of resampling used for Plasmode simulations were superior.

Regarding the resampling strategy used for Plasmode simulations, we observed that wild Bootstrap performed by far the worst with respect to MSE estimation. For the remaining types, n out of n Bootstrap was usually inferior to the other resampling strategies. The performance of m out of n Bootstrap and subsampling depends on the chosen subsampling proportion. Generally, smaller proportions are beneficial. Note that the size of the resulting dataset after resampling has to be fixed, so a smaller proportion corresponds to a larger dataset from which to sample. For small resampling proportions, m out of n Bootstrap and subsampling behave very similarly, for larger proportions, subsampling performs better. Smoothed Bootstrap performs similarly to m out of n Bootstrap and subsampling with moderate resampling proportions. For the resampling proportion approaching one, m out of n Bootstrap converges to n out of n Bootstrap. No resampling (subsampling with a resampling proportion of one), i.e. using the whole dataset for the features and only generating new responses in each iteration of the simulation, performed better than n out of n Bootstrap. The differences between the resampling types except for wild Bootstrap are negligible for small numbers of features ($p = 2$). In that case, Plasmode using any resampling strategy might be a good option since even very small deviations from the DGP lead to parametric simulation being inferior. In general, we suggest using subsampling with a small resampling proportion if feasible.

For larger numbers of features, the performance of Plasmode simulations gets worse in general. Nevertheless, there might still be good reasons for Plasmode simulations in this case. For example, the specification of the DGP gets more and more complicated with an increasing number of features. Especially the specification of the correlation structure is non-trivial as the number of pairwise correlations increases quadratically with the number of features. This might lead to the choice of oversimplified correlation structures for which we observed a clearly inferior performance of parametric compared to Plasmode simulation. A remedy could be to at least estimate key parameters like mean and covariance matrix from a real dataset for the parametric simulation. We observed good results for that strategy at least as long as the dataset from which the parameters are estimated is big enough.

The availability of datasets might be a major limitation for the application of Plasmode simulations. In general, at least one suitable dataset from the DGP of interest is required, see Section 3.2 in [4] for a discussion. Ideally, this dataset is considerably larger than the sample size n of interest, to allow for a low resampling proportion. In practice, this might often not be given. If the dataset size is limited to the sample size one is interested in, our comparison suggests that no resampling (subsampling with a resampling proportion of one) might be a reasonable variant since the error made by using a dataset of the wrong sample size might outweigh the advantage of lower resampling proportions.

Overall, the choice of the simulation type should be carefully considered for each application. A combination of parametric and Plasmode simulation within a simulation study might be a solution to use both the flexibility of parametric simulation and the ability of Plasmode to preserve characteristics of real-life data.

So far, the comparison of parametric and Plasmode simulation is limited to the specific example of estimating the MSE of the LSE. Therefore, further studies on other endpoints as well as a comparison for high-dimensional data which brings additional challenges might be interesting extensions of the analysis at hand.

Supporting information

S1 Fig. Absolute error in MSE estimation for individual coefficients for different types of Plasmode simulation compared to parametric simulation under assumption of true data generating process and outcome generating model. (A) $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. (B) $p = 2$, $n = 100$, $\beta = (1, 1, 1)^T$, $\sigma = 3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$.
(TIF)

S2 Fig. Relative error in MSE estimation averaged over individual coefficients when the variance of the second half of features is misspecified for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. The first facet displays the errors in case the misspecified variances are used in the simulation. The remaining facets display the errors for using a variance that is estimated using datasets of different sizes from the true DGP for parametric simulation instead.
(TIF)

S3 Fig. Relative error in MSE estimation averaged over individual coefficients when the assumed variance of the second half of features exceeds the true variance for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. Large outliers for n out of n Bootstrap not displayed.
(TIF)

S4 Fig. Relative error in MSE estimation averaged over individual coefficients when the assumed variance of the second half of features underestimates the true variance for $p = 50$, $n = 100$, $\beta = 1_{51}$, $\sigma = 0.3$, $Cor(X_i, X_j) = 0.2\ \forall i \neq j$. Large outliers for n out of n Bootstrap not displayed.
(TIF)

S1 Table. Complete list of deviations from true scenarios.
(PDF)

S1 Appendix. Simulation of Bernoulli, log-normal, and Gaussian mixture variables with fixed correlations.
(PDF)

Acknowledgments

We thank Markus Pauly (TU Dortmund University & UA Ruhr, Research Center Trustworthy Data Science and Security) for helpful discussions.

Author Contributions

Conceptualization: Marieke Stolte, Nicholas Schreck, Alla Slynko, Maral Saadati, Axel Benner, Jörg Rahnenführer, Andrea Bommert.

Formal analysis: Marieke Stolte.

Methodology: Marieke Stolte, Jörg Rahnenführer, Andrea Bommert.

Software: Marieke Stolte.

Supervision: Jörg Rahnenführer, Andrea Bommert.

Visualization: Marieke Stolte.

Writing – original draft: Marieke Stolte.

Writing – review & editing: Nicholas Schreck, Alla Slynko, Maral Saadati, Axel Benner, Jörg Rahnenführer, Andrea Bommert.

References

1. Boulesteix AL, Lauer S, Eugster MJA. A Plea for Neutral Comparison Studies in Computational Sciences. PLOS ONE. 2013; 8(4):e61562. https://doi.org/10.1371/journal.pone.0061562 PMID: 23637855
2. Morris TP, White IR, Crowther MJ. Using simulation studies to evaluate statistical methods. Statistics in Medicine. 2019; 38(11):2074–2102. https://doi.org/10.1002/sim.8086 PMID: 30652356
3. Boulesteix AL, Groenwold RH, Abrahamowicz M, Binder H, Briel M, Hornung R, et al. Introduction to statistical simulations in health research. BMJ Open. 2020; 10(12):e039921. https://doi.org/10.1136/bmjopen-2020-039921 PMID: 33318113
4. Schreck N, Slynko A, Saadati M, Benner A. Statistical Plasmode Simulations—Potentials, Challenges and Recommendations; 2023. Available from: http://arxiv.org/abs/2305.06028.
5. Cattell RB. A General Plasmode (No. 30-10-5-2) for Factor Analytic Exercises and Research: By Raymond B. Cattell and Joseph Jaspers. Society of Multivariate Experimental Psychology; 1967.
6. Efron B. Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics. 1979; 7(1):1–26. https://doi.org/10.1214/aos/1176344552
7. Götze F. Asymptotic approximation and the bootstrap. IMS Bulletin. 1993; p. 305.
8. Bickel PJ, Götze F, van Zwet WR. Resampling Fewer Than *n* Observations: Gains, Losses, and Remedies for Losses. Statistica Sinica. 1997; 7(1):1–31.
9. Politis DN, Romano JP, Wolf M. Subsampling. Springer Science & Business Media; 1999.
10. Efron B. Nonparametric Estimates of Standard Error: The Jackknife, the Bootstrap and Other Methods. Biometrika. 1981; 68(3):589–599. https://doi.org/10.1093/biomet/68.3.589
11. Silverman BW, Young GA. The bootstrap: To smooth or not to smooth? Biometrika. 1987; 74(3):469–479. https://doi.org/10.1093/biomet/74.3.469
12. Hall P, DiCiccio TJ, Romano JP. On Smoothing and the Bootstrap. The Annals of Statistics. 1989; 17 (2):692–704. https://doi.org/10.1214/aos/1176347135
13. Wang S. Optimizing the smoothed bootstrap. Annals of the Institute of Statistical Mathematics. 1995; 47(1):65–80. https://doi.org/10.1007/BF00773412
14. Wu CFJ. Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis. The Annals of Statistics. 1986; 14(4):1261–1295. https://doi.org/10.1214/aos/1176350161
15. Bickel P, Sakov A. On the Choice of m in the m Out of n Bootstrap and Confidence Bounds for Extrema. Statistica Sinica. 2008; 18.
16. Simonoff JS. Smoothing Methods in Statistics. Springer Science & Business Media; 1996.
17. Cortez P, Cerdeira A, Almeida F, Matos T, Reis J. Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems. 2009; 47(4):547–553. https://doi.org/10.1016/j.dss.2009.05.016
18. Weiss SM, Indurkhya N. Rule-based Machine Learning Methods for Functional Prediction. Journal of Artificial Intelligence Research. 1995; 3:383–403. https://doi.org/10.1613/jair.199
19. Guyon I, Sun-Hosoya L, Boullé M, Escalante H, Escalera S, Liu Z, et al. In: Analysis of the AutoML Challenge series 2015-2018; 2017. Available from: https://www.automl.org/book/.
20. Vanschoren J, van Rijn JN, Bischl B, Torgo L. OpenML: networked science in machine learning. SIGKDD Explorations. 2013; 15(2):49–60. https://doi.org/10.1145/2641190.2641198
21. Gijsbers P, LeDell E, Thomas J, Poirier S, Bischl B, Vanschoren J. An Open Source AutoML Benchmark. CoRR. 2019;abs/1907.00909.

22. Astivia OLO, Zumbo BD. Population models and simulation methods: The case of the Spearman rank correlation. British Journal of Mathematical and Statistical Psychology. 2017; 70(3):347–367. https://doi.org/10.1111/bmsp.12085 PMID: 28140458

23. Emrich LJ, Piedmonte MR. A Method for Generating High-Dimensional Multivariate Binary Variates. The American Statistician. 1991; 45(4):302–304. https://doi.org/10.1080/00031305.1991.10475828

24. Silverman BW. Density estimation for statistics and data analysis. Chapman & Hall/CRC; 1986.

25. R Core Team. R: A Language and Environment for Statistical Computing; 2023. Available from: https://www.R-project.org/.

26. Genz A, Bretz F, Miwa T, Mi X, Leisch F, Scheipl F, et al. mvtnorm: Multivariate Normal and t Distributions; 2021. Available from: https://CRAN.R-project.org/package=mvtnorm.

27. Genz A, Bretz F. Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics. Heidelberg: Springer-Verlag; 2009.

28. Wolodzko T. kernelboot: Smoothed Bootstrap and Random Generation from Kernel Densities; 2023. Available from: https://CRAN.R-project.org/package=kernelboot.

29. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2016. Available from: https://ggplot2.tidyverse.org.

30. van den Brand T. ggh4x: Hacks for'ggplot2'; 2023. Available from: https://CRAN.R-project.org/package=ggh4x.

31. De Bin R, Janitza S, Sauerbrei W, Boulesteix AL. Subsampling versus Bootstrapping in Resampling-Based Model Selection for Multivariable Regression. Biometrics. 2016; 72(1):272–280. https://doi.org/10.1111/biom.12381 PMID: 26288150

32. Karoui NE, Purdom E. Can we trust the bootstrap in high-dimension?; 2016.

2. Article 2: Simulation Study to Evaluate When Plasmode Simulation is Superior to Parametric Simulation in Comparing Classification Methods on High-Dimensional Data

PLOS ONE

RESEARCH ARTICLE

Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in comparing classification methods on high-dimensional data

Marieke Stolte[1]*, Nicholas Schreck[2,3], Alla Slynko[4], Maral Saadati[2], Axel Benner[2], Jörg Rahnenführer[1], Andrea Bommert[1], for the topic group "High-dimensional data" (TG9) of the STRATOS Initiative‡

1 Department of Statistics, TU Dortmund University, Dortmund, North Rhine-Westphalia, Germany, **2** Division of Biostatistics, German Cancer Research Center, Heidelberg, Baden-Wuerttemberg, Germany, **3** Faculty of Liberal Arts and Sciences, Technical University of Applied Sciences Augsburg, Augsburg, Bavaria, Germany, **4** Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada

‡ Membership of the topic group "High-dimensional data" (TG9) of the STRATOS initiative is provided in the Acknowledgments.
* stolte@statistik.tu-dortmund.de

Citation: Stolte M, Schreck N, Slynko A, Saadati M, Benner A, Rahnenführer J, et al. (2025) Simulation study to evaluate when Plasmode simulation is superior to parametric simulation in comparing classification methods on high-dimensional data. PLoS One 20(6): e0322887. https://doi.org/10.1371/journal.pone.0322887.

Editor: Li-Pang Chen, National Chengchi University, TAIWAN

Received: November 12, 2024

Accepted: March 29, 2025

Published: June 2, 2025

Data availability statement: The full code and simulation results for the simulation study are available on Zenodo (https://doi.org/10.5281/zenodo.13707473).

Funding: MS has been supported (in part) by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project P1) funded by the Deutsche Forschungsgemeinschaft (DFG, https://gepris.dfg.de/gepris/projekt/427806116,

Abstract

Simulation studies, especially neutral comparison studies, are crucial for evaluating and comparing statistical methods as they investigate whether methods work as intended and can guide an appropriate method choice. Typically, the term simulation refers to parametric simulation, i.e. computer experiments using pseudo-random numbers. For these, the full data-generating process (DGP) and outcome-generating model (OGM) are known within the simulation. However, the specification of realistic DGPs might be difficult in practice leading to oversimplified assumptions. The problem is more severe for higher-dimensional data as the number of parameters to specify typically increases with the number of variables in the data. Plasmode simulation, which is a combination of resampling covariates from a real-life dataset from the DGP of interest together with a specified OGM is often claimed to solve this problem since no explicit specification of the DGP is necessary. However, this claim is not well supported by empirical results. Here, parametric and Plasmode simulations are compared in the context of a method comparison study for binary classification methods. We focus on studies conducted with some specific data type or application in mind whose true, unknown data-generating mechanism is mimicked. The performance of Plasmode and parametric comparison studies for estimating classifier performance is compared as well as their ability to reproduce the true method ranking. The influence of misspecifications of the DGP on the results of parametric simulation and of misspecifications of the OGM on the results of parametric and Plasmode simulation are investigated. Moreover, different resampling strategies are compared for Plasmode comparison studies. The study finds that misspecifications of the DGP and

German Research Foundation - Project Number 427806116). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

OGM negatively influence the ability of the comparison studies to estimate the classification performances and method rankings. The best choice of the resampling strategy in Plasmode simulation depends on the concrete scenario.

Introduction

Simulation studies are a crucial tool in evaluating and comparing the performance of statistical methods. They can provide useful insights into the behavior of the methods in certain situations. Neutral method comparison studies, i.e. comparison studies evaluating existing methods outside the context of proposing a new method, are particularly important to ensure that methods work as expected and for making an informed method choice for an analysis task at hand [1,2].

Most commonly, the term simulation is used to refer to parametric simulation. That is, computer experiments based solely on pseudo-random data generation according to data-generating processes (DGP) and outcome-generating models (OGM) specified by the researchers conducting the simulation study. Often, covariate data is generated from a specified distribution using a pseudo-random number generator. Then, the specified OGM is applied to the generated covariate data to generate observations of a target variable. This step might again include some pseudo-random number generation, e.g. to produce some noise in the target variable. This procedure has the advantage of full control over the data generation and full knowledge of all parameters within the simulation, which enables the calculation of performance measures that rely on knowledge of the true parameters like the bias of an estimator [3,4]. However, the specifications of the DGP and the OGM might be oversimplified and therefore unrealistic as the specification of complicated DGPs and OGMs is often hard in practice, especially for large numbers of variables. For example, the specification of a complex correlation structure becomes tedious for large numbers of variables [5].

Plasmode simulations [6] are often claimed as a solution to the problem of unrealistic assumptions made in parametric simulations. For statistical Plasmode simulation, the covariate data is generated by resampling from a real-world dataset that is drawn from the true DGP of interest. Therefore, no explicit DGP specification is needed. Moreover, the resampling from the real-world dataset is expected to accurately reflect the true DGP, assuming that the dataset is representative and possibly additional assumptions on the resampling scheme [5]. As for parametric simulation, the target observations are generated using an OGM specified by the researchers. Therefore, some truth is still known in the data generation and all performance measures, such as the bias, that need knowledge of parameters in the OGM can still be calculated. Thus, Plasmode simulation seems like a good alternative to parametric simulation for investigating complex DGPs while still being able to evaluate the performance of statistical methods of interest [5].

However, [5] noted that this often-made claim of Plasmode simulation producing more realistic data is not well supported by any empirical results. Moreover, they point out potential pitfalls when conducting Plasmode simulation studies. For example, they mention the importance of choosing an appropriate dataset to resample from, the difficulties of small sample sizes for resampling, and the choice of the resampling strategy itself. In addition, they highlight the importance of the choice of an appropriate OGM and question, for example, the practice of nullifying existing associations between covariates and the target variable. Therefore, a comparison of parametric and Plasmode simulation is required to find out in which situations Plasmode simulation is actually preferable.

As a first step to close this gap, [7] empirically compared parametric and Plasmode simulations for the example of estimating the mean squared error of the least squares estimator in linear regression. They found that, as expected, parametric simulation performs best if the DGP and OGM are specified correctly, but it quickly gets worse when some aspects of the DGP or OGM are misspecified. The performance of the Plasmode simulation also deteriorated in case of misspecifications of the OGM. Moreover, the performance of the simulations, especially for Plasmode, got worse when increasing the number of variables or decreasing the number of observations in the generated datasets. Regarding the resampling step in Plasmode simulations, often subsampling with low resampling proportions outperformed the other options in the comparison, but this required a larger dataset to resample from. However, that study was limited to only one specific example case of a method evaluation study.

Here, we want to expand on this by comparing parametric and Plasmode simulation in the context of method comparison studies, using the example of comparing multiple binary classification methods. The comparison of multiple methods is more complex as not only the performance of each method but also their ranking with respect to the performance is of interest. We focus on the case where researchers designing a simulation study have a certain type of data or a certain application in mind as in this case, Plasmode is a reasonable alternative to parametric simulation. Therefore, we assume there is some true but typically unknown data-generating mechanism that researchers try to mimic through their simulation. Here, we compare how well the true classification performance and method ranking can be recovered for parametric simulation studies and for Plasmode simulation studies with different resampling strategies. Under the true scenario, it is expected that parametric simulation performs best. However, the truth is typically unknown to researchers conducting simulation studies and instead, they have to make assumptions trying to approximate this truth. These assumptions are likely to deviate from the truth. Therefore, we analyze how performance estimation and method ranking are affected by misspecifications of the DGP (for parametric comparison studies) and of the OGM (for parametric and Plasmode comparison studies). In comparison to the previous study, we use a higher-dimensional setup and additional deviations. Note that we do not aim to perform a neutral comparison study of classification methods but to compare how well such a study would perform using different simulation approaches and assumptions within the comparison study.

The remaining article is structured as follows. In the Simulation setup section, the setup of the simulation study is explained. In the Results section, the results for the comparison of parametric and Plasmode, and the influence of misspecifications of the DGP and OGM are described. First, results regarding the estimation of the classification performance measures are shown. Then, results regarding the method ranking are presented. Last, an overall comparison based on the proportion of acceptable simulation results is performed. Precisely, this is the proportion of simulation results whose relative errors in estimating the classification performance fall into the 2.5% to 97.5%-quantile interval of the relative errors for parametric simulation using the true DGP and OGM. In the Discussion section, the results are summarized and discussed.

Simulation setup

In the following, we describe the simulation setup following the ADEMP (Aims, Data-generating mechanism, Estimands, Methods, Performance measures) structure [3]. The overall procedure for the simulation is visualized in Fig 1.

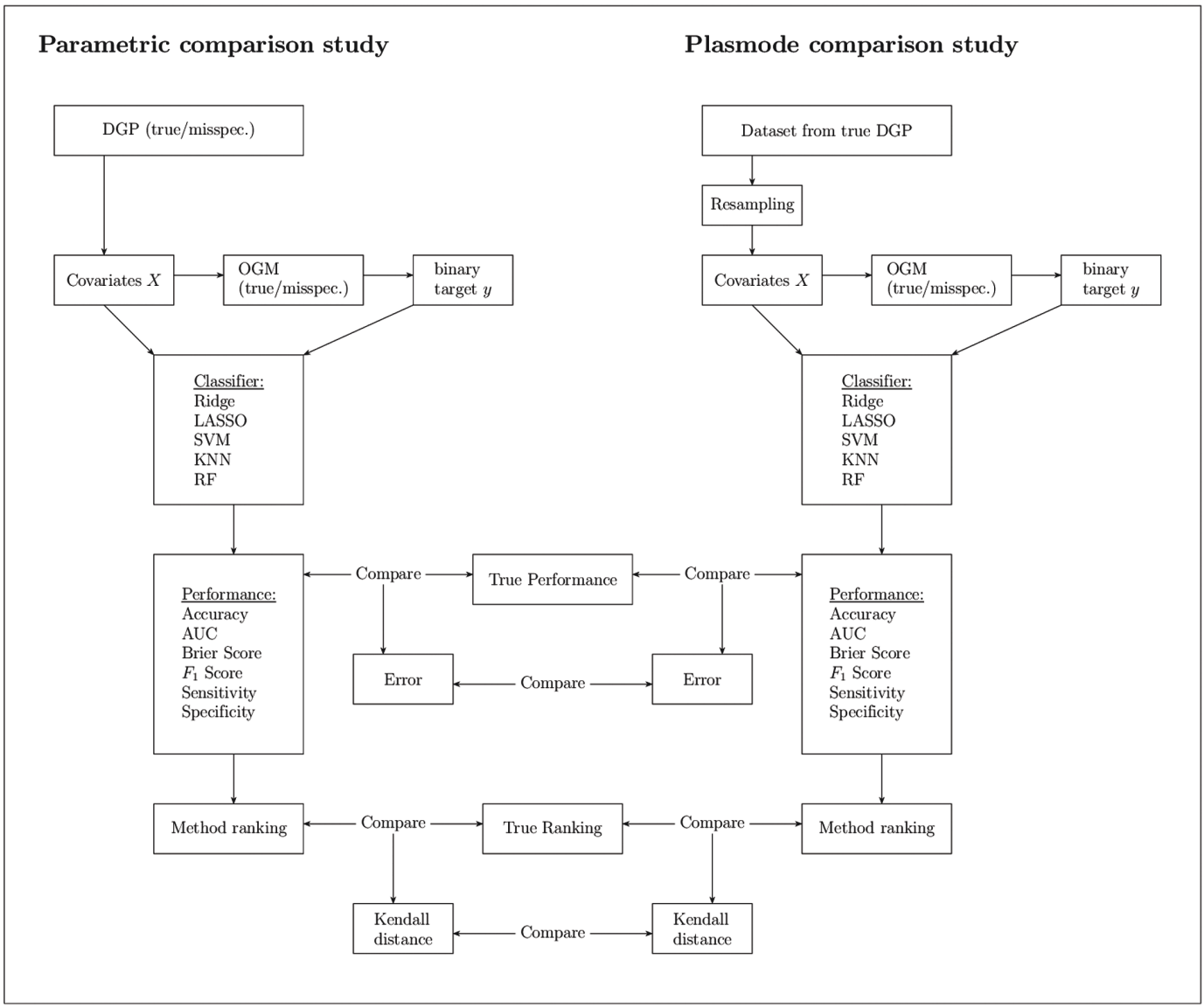


Fig 1. Schematic process of the simulation study.

https://doi.org/10.1371/journal.pone.0322887.g001

Aims

The aims of our simulation study are:

1. Compare how well parametric and Plasmode simulation can estimate the performance and method ranking for several classification methods.
2. Find out how deviations from the true DGP and OGM affect parametric simulation in terms of estimating the performance and ranking classification methods.
3. Find out how deviations from the true OGM and different resampling strategies affect Plasmode simulation in terms of estimating the performance and ranking of classification methods.
4. Find out how the dimension of the datasets, i.e. the number of covariates, affects 1. to 3.

Note that the study does not aim to perform a neutral method comparison of classification methods. Instead, it is of interest how well such a method comparison study can recover the true method performances under certain simulation approaches and possibly misspecified assumptions in the comparison study.

Data-generating mechanism

Since we want to compare how well comparison studies can recover the true method performances, we first have to define a DGP and OGM that are considered the truth for our study. Additionally, we have to specify the assumptions on the DGP and OGM within the comparison study. These do not have to coincide with the truth as the truth is typically unknown to the researchers performing such comparison studies. However, the assumptions on the DGP and OGM made within the comparison studies are chosen fairly close to the truth based on the assumption that researchers conducting the study would try to mimic the truth as well as possible.

True scenarios. The true scenarios consist of a true data-generating process (DGP) and a true outcome-generating model (OGM). We must have full knowledge of both. At the same time, in practice, the true DGP and OGM are typically complicated, which we try to reflect here as well. We fix the sample sizes for all generated datasets at $n = 100$. For larger n the classification problem becomes easier. For smaller n the training datasets become very small. The number of variables for each sample is varied as $p = 2, 10, 50, 150$. This means that we have one true scenario for each p. However, we try to keep the true scenarios for different ps as comparable as possible. Larger values of p quickly result in infeasibly long runtimes of the classification models. Smaller n leads to deficient true classification performances of the classifiers, which makes the comparison of different simulation strategies pointless. In this case, often the model is random guessing under the true scenario, and the model in the comparison study is also random guessing. Consequently, the simulated performances are close to the classification performances under the true scenario by chance.

True DGPs. We specify the distribution of 150 variables. For the other values of p, subsets of the marginal distributions will be chosen as described below and the correlation matrix is reduced to the corresponding entries. This ensures that the DGPs for different p are comparable.

Here, for the true DGP, the marginal distributions and the correlation structure of the 150 variables have to be specified. For the marginal distributions, different distribution families are chosen including normal distribution, log-normal distribution representing a skewed distribution, Gaussian mixture distributions representing bimodal distributions, and a contamination model for outliers, respectively. Table 1 gives an overview of the numbers of variables per distribution class for each value of p.

For $p = 150$, for normal distributions, we generate 50 variables for which the means and variances are randomly sampled such that the expected parameter for the mean is zero and

Table 1. Number of variables generated from each distribution class per number of variables *p*.

p	Normal	Log-normal	Bimodal	Outlier
150	50	50	25	25
50	15	15	10	10
10	3	3	2	2
2	1	0	1	0

https://doi.org/10.1371/journal.pone.0322887.t001

the expected parameter for the variance is one (see Section A of S1 Appendix). For log-normal distributions, 50 variables are generated and the parameters μ and σ are randomly sampled in the same way as for the normal variables. For Gaussian mixture distributions, also 50 variables are generated. Half of these are generated from bimodal distributions and the other half is generated from a contamination model. The parameters are sampled as follows. For the first component of these variables, parameters are drawn such that on average standard normal parameters are achieved. For the bimodal distributions, the second component has an expected μ of 4. For the outlier distributions, the second component has an expected variance of 10. For details see Section A of S1 Appendix. The distribution of the first few variables of each type is visualized in Fig A.2 in Section A of S1 Appendix. Drawing the parameters produces more diverse marginal distributions than specifying the values by hand.

The correlation matrix is also generated randomly. Fig A.3 in Section A of S1 Appendix shows the distribution of pairwise correlations. For details on the random generation, see Section A of S1 Appendix. All marginal distributions with generated parameters can be found in S3 Appendix. The full correlation matrix is given in S4 Appendix.

The parameters for all distributions and the correlations are drawn only once and set as the true parameters for these true distributions for the whole simulation.

For $p = 50$, we select the first 15 of the normal distributions, the first 15 of the log-normal distributions, and the first 10 for each of the bimodal and outlier Gaussian mixture distributions and the corresponding entries from the true correlation matrix.

For $p = 10$, we select the first three of the normal distributions, the first three of the log-normal distributions, and the first two for each of the bimodal and outlier Gaussian mixture distributions and the corresponding entries from the true correlation matrix.

For $p = 2$, we select the first of the normal distributions, the first of the bimodal Gaussian mixture distributions, and the corresponding entries from the true correlation matrix.

Note that in these cases we are not selecting parts from the same dataset with $p = 150$ variables but instead, we are drawing data from the respective subsets of the 150 distributions.

Since it helps with constructing the deviation scenarios, we rescale all variables in the generated datasets from the true DGP to [0,1] using a min-max transformation

$$x_{i,\text{rescaled}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}},$$

where x_i denotes the ith observation of variable x and $x_{\min}$ and $x_{\max}$ denote the minimum and maximum of x, respectively.

True OGMs. We use a logistic model as true OGM since it allows us to control the true separation of the two classes most efficiently. Note that this choice can give an unfair advantage to linear classification methods like Ridge and LASSO logistic regression. Since we are not inherently interested in the method comparison of the classification methods but in how well the simulation studies reconstruct the true comparison, we can give up the fairness in comparing classification methods to some extent. It might even be advantageous here to have a slightly unfair classification method comparison since then the differences between the classifiers are expected to be more distinct and therefore the method order is clearer and easier to reconstruct in the simulations. If the true order is ambiguous since all methods perform equally well, it is expected that the simulation studies cannot reconstruct this order well. For a discussion of an alternative approach and its disadvantages that made us not consider it and instead led to our choice, see Section B in S1 Appendix.

The coefficients for the logistic model for $p = 150$ are chosen as follows. The 100 coefficients for the normal, log-normal, and outlier variables are drawn at random either from a $U(-8, -3)$ or from a $U(3, 8)$ distribution. The remaining 50 coefficients for the bimodal variables are drawn at random either from a $U(-15, -10)$ or from a $U(10, 15)$ distribution. The choice of larger absolute coefficients for the bimodal distributions ensures a clear separation of the data into the two classes that is necessary to achieve reasonable performances of the classification methods for larger p. The intercept is set to adjust the predicted probabilities such that the target variable is nearly balanced. Own analyses showed that extreme unbalance results in many generated datasets with either no generated zero responses or no generated responses of one, which makes the classification unnecessary. Note that the coefficients are seemingly very large but the data is rescaled to [0,1] before applying the OGM. Therefore, odds ratios (OR) for a variable increase of 0.1 are more realistic than the typical increase of 1. The ORs for an increase of 0.1 and the positive coefficients of normal, log-normal, and outlier variables are between 1.35 and 2.23, and for the coefficients of bimodal variables between 2.72 and 4.48. The resulting distribution of predicted probabilities (Fig D.1 in Section D of S1 Appendix) shows a clear separation between the two classes and is approximately symmetric, resulting in an approximately balanced binary target variable. The exact coefficients can be retrieved from the R code available on Zenodo (https://doi.org/10.5281/zenodo.13707473). Fig C.1 in Section C of S1 Appendix shows the distribution of coefficients.

For $p < 150$, the coefficients corresponding to the respective variables chosen from the true DGP are used and modified slightly if necessary to achieve good separation. For details see Section D of S1 Appendix.

As with the true DGP, the coefficients for $p = 150$ are drawn exactly once in the beginning and then kept constant during the whole simulation process.

The ith target observation for a given simulated covariate dataset is generated by drawing from a Bernoulli distribution with the success probability set to the probability predicted by the true OGM as

$$\hat{\pi}_i = \frac{1}{1 + \exp(-x_i^T \beta)},$$

where x_i^T is the ith row of the simulated dataset supplemented by a leading one for the intercept, $i = 1, \ldots, 100$, and β is the coefficient vector generated as described above.

Note that the true OGMs are constructed such that each feature influences the outcome.

Deviations. In the following, it is described how the DGP and the OGM are misspecified within the comparison studies. In addition to the misspecifications described below, the true DGP and OGM are always used once for a parametric and for a Plasmode comparison study, respectively. Table 2 gives an overview of all applied misspecifications.

Misspecifications of the DGP. The DGP in parametric simulations can be misspecified by changing some characteristics of the distribution. Shift, scale, correlation, and the whole distribution are misspecified as follows one at a time. The concrete parameter values are given in Table 2. Note that the generated data from the true DGP is first rescaled to [0,1] and then for shift the value of δ is added to all observations and for scale the data is multiplied by s.

The parameter settings were chosen as follows. As all data is scaled to [0,1], a shift of ± 0.5 is already extreme. Too extreme values of the shift might result in the generation of extremely imbalanced classes which in the extreme case makes the application of the classification models or the performance estimation impossible. Therefore, the extreme shift values were considered together with less extreme values. For scale, the most extreme values of $s = 0.25$ and

Table 2. Misspecifications of the DGP in parametric and of the OGM in parametric and Plasmode simulation.

Type of misspecification		Values
DGP	Shift	$\delta \in \{-0.5, -0.25, -0.125, 0.125, 0.25, 0.5\}$
	Scale	$s \in \{0.25, 0.5, 0.75, 1.33, 2, 4\}$
	Correlation	$\rho \in \{-0.2, -0.1, 0, 0.1, 0.2\}$
	Distribution	$N(0, I)$
OGM	Scaled	$c \in \{0.5, 2, 0\}$

https://doi.org/10.1371/journal.pone.0322887.t002

$s = 4$ were chosen such that still reasonable proportions of zeroes and ones are generated. As especially $s = 0.25$ turned out as too extreme in certain scenarios, the less extreme values of 3/4 and 4/3 were added. For misspecifying the correlation, all pairwise correlations are fixed as ρ. Larger absolute correlations are infeasible for many pairs of marginal distributions, see the discussion in Section A of S1 Appendix.

Lastly, the distribution is completely misspecified as standard normal. This case is included because researchers with no prior knowledge about the true DGP often use standard normal data in their comparison study by default.

Misspecification of the OGM. A very general approach to modify classification models applicable to all models that output predicted probabilities is described in [8]. The predicted probabilities $\hat{\pi}$ of the model are transformed into log-odds $\log(\hat{\pi}/(1-\hat{\pi}))$. These log-odds are multiplied by a constant c to get stronger or weaker associations. The new log-odds are then transformed back to the probability scale

$$\hat{\pi}_{\text{new}} = \frac{1}{1 + \exp(c \cdot \log(\hat{\pi}/(1-\hat{\pi})))}.$$

These new probabilities are used to generate observations of the target variable. We adopt this approach here with a factor of

$$c \in \{0.5, 2\}$$

to get models with weaker and stronger associations, respectively. Values of $|c| > 1$ correspond to stronger associations and lead to better-separated classes while values of $|c| < 1$ lead to weaker associations and less separated classes in the simulated responses. For the special case of the logistic model as the true OGM, this is equivalent to multiplying each coefficient by c. Note that for $c < 0$ it holds:

$$c \log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = \log\left(\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right)^{c}\right) = \log\left(\left(\frac{1-\hat{\pi}}{\hat{\pi}}\right)^{|c|}\right) = |c| \log\left(\frac{1-\hat{\pi}}{\hat{\pi}}\right).$$

Therefore, using negative factors is equivalent to changing the roles of zeroes and ones and using the absolute value of the factor. Changing the roles of zeroes and ones does not affect the classification performance measured by accuracy, AUC, and the Brier score but changes the roles of sensitivity and specificity (see Section on performance measures). For the F_1-score, it is unclear how the performance changes. As it is clear how the use of negative factors affects most of the performance measures used, only positive values are used. In addition, we use a logistic model with constant coefficients of 0, i.e. no effect of the covariates on the response, as this might be done in many simulations to illustrate a null situation. This is included as [5] pointed out that nullifying true existing effects is potentially problematic. Overall we misspecify the OGM once by scaling by 0.5 to achieve weaker associations,

once by scaling by 2 to achieve stronger associations, and once by setting all coefficients to 0 as discussed above.

Targets

The targets of the study are the classification performance on the simulated data and performance rankings for comparing classification methods obtained by parametric and Plasmode simulation.

Methods

In the following, parametric and Plasmode simulation and the classification methods for the method comparison studies are briefly explained.

Parametric simulation. Parametric simulation refers to simulations where the whole data consists of pseudo-random numbers drawn from a data-generating process (DGP) and an outcome-generating model (OGM) specified by the researcher. Therefore, both the DGP and OGM are fully known. The choice of these can be hard in practice. Researchers might try to set up their parametric simulation to be as close as possible to certain data of interest. Alternatively, researchers might want to cover as many situations as possible including extreme scenarios. The first case is the one where Plasmode simulations might be a reasonable alternative. In the latter case, parametric simulation would be the obvious choice as it allows specification of all aspects of the DGP and OGM. Therefore, we focus on the first case here. When the DGP and OGM are specified, a large number of covariate datasets can be generated using a pseudo-random number generator to draw observations from the DGP. Then, the OGM is applied to this generated covariate data to generate corresponding observations of the target variable. This process mimics repeatedly drawing samples from a large population with the specified DGP and OGM. For method comparison, the methods are then applied to the generated datasets, and their performance is evaluated with regard to performance metrics of interest. Since all aspects of the true DGP and OGM are known, performance metrics depending on these (e.g. bias) can be assessed. The results can help to understand how the methods perform for datasets similar to the chosen DGPs and OGMs and which method to prefer in which situations. This is of great use for an adequate method choice in practice [1,4]. For more details on how to design, perform, analyze, and report parametric simulation studies, refer to [3]. For method comparison studies, see also [1].

Here, we perform the parametric simulation studies as follows. In each of the 100 iterations for a scenario consisting of a combination of p, the choice of the DGP, and the choice of the OGM, we draw 100 observations from the chosen DGP. Then, the chosen OGM is applied to this generated covariate data to generate observations of the binary target variable. Subsequently, all classification methods are applied using 5-fold nested cross-validation for hyperparameter tuning and performance estimation. Last, the methods are ranked according to their performance with regard to each performance measure.

Plasmode simulation. The main difference between Plasmode simulation and parametric simulation is the generation of the covariate datasets. In Plasmode simulation studies, instead of specifying the DGP like in the parametric case, data is resampled from a real-life dataset from the true DGP of interest. Therefore, no explicit assumptions on the DGP are made. However, it is required to have a representative real-life dataset from the true DGP at hand. The OGM is then applied to the resampled covariate datasets and the method comparison is performed analogously to the parametric simulation. Plasmode can be seen as a semi-parametric approach as it combines the resampling from a real-life dataset in non-parametric simulation with the use of a specified OGM in parametric simulation. This has the advantage

that some control over the data generation is given and some aspect of the truth within the simulation is known while at the same time, the problem of unrealistic specifications of the DGP in parametric simulation is avoided [5]. When using only real data, certain quantities depending on unknown parameters (e.g. bias) cannot be assessed [4]. For a more detailed discussion of the advantages and disadvantages of Plasmode simulations as well as guidance on how to perform them refer to [5].

There are multiple options for the resampling step. Here we use all resampling techniques that are commonly used according to [5]:

- m out of n Bootstrap [9–12] with resampling proportions 0.632 and 1, i.e. drawing with replacement $m \leq n$ observations of the original dataset.
- Subsampling with resampling proportions 0.632 and 1, i.e. drawing without replacement $m < n$ observations of the original dataset or using the whole dataset.

These values for the resampling proportions were chosen for comparability with the previous study [7] where the values of 0.632 and 1 were used as they were identified as relevant special cases from the literature. Additionally, smaller resampling proportions like 0.1 were previously used and showed good performance. Here, nested 5-fold cross-validation will be applied to the datasets later on (see Subsection Performance measures). For $n = 100$, the training datasets have size $100 \cdot 4/5 \cdot 4/5 = 64$ in the inner cross-validation loop. If we apply subsampling or Bootstrapping this number of training datapoints reduces accordingly. For a resampling proportion of 0.632, there are about 40 training points left which is already few. Therefore no smaller resampling proportions are used. Another solution would be to increase the number of folds in the cross-validation, but the runtime increases roughly quadratically in the number of folds. Therefore, the number of folds is kept low and the resampling proportions higher.

For each specific scenario, consisting of the number of variables p, a chosen resampling strategy, and a chosen OGM, a dataset of size 100 is generated from the true DGP. This dataset is then used to resample from it, for the 100 iterations of the Plasmode simulation. After resampling from this dataset from the true DGP, the next steps are the same as for the parametric simulation, applying the OGM and analyzing the generated data.

Classification methods. Within our parametric or Plasmode method comparison studies, we compare several methods for binary classification including

- Ridge logistic regression [13],
- LASSO logistic regression [14],
- Support vector machine (SVM) [15],
- k-nearest neighbors (KNN) [16,17], and
- random forest (RF) [18].

As we are not primarily interested in the method comparison itself we do not include boosting or neural nets due to their high runtimes and sensitivity to tuning. We concentrate on commonly used classification methods for the low to high-dimensional regime that we investigate here. For even higher-dimensional data, specialized classification methods might be needed [19,20]. We use 5-fold nested, stratified cross-validation (see Subsection) and random search with a budget of 100 evaluations for hyperparameter tuning of each method. We tune with respect to classification accuracy. The low budget is chosen as we are not primarily interested in the method comparison itself and runtime is an issue in this study. The hyperparameter spaces are chosen as suggested in [21].

Performance measures

For judging the performance of a binary classification method, typically its predicted outcome values are compared to the true outcome values. These can be summarized in a confusion matrix counting the numbers of observations for all possible combinations of true and predicted outcomes (see Table 3).

For the method comparison within each simulated simulation study,

- Accuracy = $\frac{\text{TN+TP}}{N}$,
- F_1-Score = $\frac{2\text{TP}}{2\text{TP+FP+FN}}$,
- Sensitivity = $\frac{\text{TP}}{\text{TP+FN}}$,
- Specificity = $\frac{\text{TN}}{\text{FP+TN}}$
- AUC (Area under the Receiver Operating Curve that is the diagram of Sensitivity against 1–Specificity for different cutoff values for the predicted probabilities corresponding to a prediction of a 1), and
- Brier score = $\frac{1}{N}\sum_{i=1}^{N}(\hat{\pi}_i - y_i)^2$, where $\hat{\pi}_i$ is the predicted probability for a 1 for the ith observation and $y_i \in \{0, 1\}$ the corresponding true outcome value,

are used to judge the performance of the classification methods. Subsequently, the methods are ranked according to each measure. All measures return values in [0,1]. For all except the Brier score, high values indicate good performance. For the Brier score, low values indicate good performance [22,23]. 5-fold nested cross-validation [24] is applied for performance estimation and hyperparameter tuning. Note that the performance measures are chosen because they are commonly used performance measures for binary classification methods rather than recommendations. For instance, only the Brier score is a proper measure, AUC is semi-proper and all other measures are improper measures.

We calculate performance measures, based on scoring rules to assess the quality of probabilistic predictions by assigning a numerical score to compare predictions and the occurring event. A scoring rule is proper if the best predictor is the true probability of the event. A strictly proper scoring rule such as the Brier score guarantees that the best value is only achieved when we get as close as possible to the true probability. A semi-proper measure not only does not guarantee that the best performance is achieved by a predictor whose predictions are closest to the true probabilities, but it is also possible to improve the values of the measure by moving the predicted probabilities away from their true values. An improper scoring rule, such as 'Accuracy', does not predict probabilities as close as possible to the true probabilities [25].

If a method fails and an error is thrown, a fallback learner that always predicts the majority class is used instead to calculate the performance. Using a fallback learner is recommended over excluding the iterations with method failure or penalizing method failure by imputing the worst possible score [26].

Table 3. Confusion matrix for binary classification methods. y, true outcome; $\hat{y}$, predicted outcome; TN, number of true negatives; FN, number of false negatives; FP, number of false positives; TP, number of true positives; N, the number of observations.

	$y = 0$	$y = 1$	$\sum$
$\hat{y} = 0$	TN	FN	TN + FN
$\hat{y} = 1$	FP	TP	FP + TP
$\sum$	TN + FP	FN + TP	N

https://doi.org/10.1371/journal.pone.0322887.t003

For the parametric and Plasmode simulation, 100 datasets are generated per scenario on which the method comparison is performed. This number is mainly motivated by runtime. If only ones or only zeros are generated in an iteration, the whole data including the covariates is redrawn up to 50 times. It might still happen that during cross-validation some of the folds have only ones or only zeroes as response values. Then, the sensitivity or specificity cannot be calculated and consequently also the AUC cannot be calculated for this fold. In this case, the values of the measure in the remaining folds are averaged and the fold with only ones or only zeroes as response values is left out (for the affected measures only). In the case of sensitivity and specificity, this procedure gives similar results to calculating the measure on all predicted responses across the folds as the proportions of ones and zeroes are similar in all folds since we use stratified cross-validation. For the AUC, the results when first pooling the predictions over the folds could differ notably if the classifiers in the different folds are calibrated differently. Therefore, pooling would not be a good idea and we choose the approach of averaging over the remaining folds. If there are no true or predicted ones for a certain fold, the F_1-score cannot be calculated. In case of no true ones, the same approach as for sensitivity and AUC is chosen. In case of no predicted ones, a value of zero is assigned as the F_1-score for that fold which corresponds to the worst possible value. If there are ones, but the classifier does not predict any, then its performance regarding predicting ones is as bad as possible.

To judge the performance of the simulation studies themselves we calculated the differences between the estimated performance values and their true values for each measure. The true performances and rankings are approximated using datasets drawn from the true DGP and responses generated by the true OGM and benchmarking all five classification methods with regard to all performance measures on these simulated datasets, as described before. This is done 500 times for the true model for each value of p. The mean performance of each classification method is calculated as its true performance for each measure. The method ranking based on these mean performances is used as the true ranking. Ranks are always assigned such that lower ranks indicate better performance regardless of whether high or low values of the corresponding performance measure indicate good performance. Moreover, the Kendall distance [27] of the simulated and the true ranking according to each measure is calculated. It is a standard metric for comparing permutations [28]. The Kendall distance is defined as the number of swaps of neighboring values required to transform the simulated ranking into the true one. Kendall distance values are normalized to [0,1] where 0 corresponds to equal rankings (best possible value) and 1 corresponds to reversed rankings (worst possible value). Ties in the method rankings are broken at random as average ranks are not permitted for the calculation of Kendall distance. Since the true ranks are in $\{1, 2, 3, 4, 5\}$ for example the ranking 1, 2.5, 4, 2.5, 5 cannot be transformed to the true ranks via permuting adjacent numbers in the ranking. Ideally, the estimated method ranking should be similar to the true ranking as method rankings established by simulation studies should be used as guidance for choosing a suitable method in practice [1]. Therefore, a wrong method ranking in a simulation study can result in non-optimal method choices in practice.

Even if the classification performance measure values for the classifiers are estimated precisely, still the method ranking might differ from the true ranking as often already small differences in the classification performance can change the rank of a method. Conversely, the estimation of the method ranking can still be good if all estimates of the classification performance measures are biased in the same direction and by roughly the same amount. A good simulation study should recover the true method ranking without under- or overestimating the true classification performances. Therefore, both the errors in estimating the classification performance as well as the Kendall distances of the method rankings are taken into account.

To summarize the results of the comparison studies, the proportions of acceptable simulation results are calculated as follows. First, relative errors of one minus the respective measure with respect to one minus the true measure are calculated for each iteration as

$$\text{Relative Error}_i = \frac{(1 - \widehat{M}_i) - (1 - M)}{1 - M},$$

where M denotes the true measure value and $\widehat{M}_i$ the simulated measure value in the ith iteration. This weighs errors for high true performance values higher than for moderate performance which is how we would judge the performance intuitively. For the Brier score, where low values correspond to good performance, the usual relative errors

$$\text{Relative Error}_{\text{Brier Score},i} = \frac{\text{Simulated Brier Score}_i - \text{True Brier Score}}{\text{True Brier Score}},$$

are calculated. Then, the proportion of "acceptable" simulation results is calculated per simulation type, measure, classifier, and scenario. For a simulation result to be called acceptable, here, its relative error must lie within the 2.5% and 97.5% quantile interval of the relative errors for the parametric simulation for the true scenario for the corresponding measure, classifier, and p. Therefore, for the true scenario and parametric simulation, the proportion of acceptable iterations is 95% by design. The relative errors for the parametric simulation for the true scenario for a measure and classifier can be seen as the best results possible in a comparison study. The proportions are compared across the other simulation types and scenarios for each classifier, measure, and number of variables p. High proportions of acceptable estimates mean that comparison studies with the respective assumptions will perform comparably well as comparison studies under the true scenario which yield the best result we can achieve.

Software

The true DGP was set up in `julia 1.10.2` [29] using the packages `Bigsimr.jl` [30] and `Distributions.jl` [31,32]. All other calculations were performed using `R 4.3.3` [33]. For data generation, the R package `bigsimr` [34] was used which is built on the `Bigsimr.jl`-package. For benchmarking the classifiers, the R package `mlr3 0.18.0` [35] together with `mlr3measures` [36] was used. This uses the LASSO and Ridge implementation from `glmnet` [37], the SVM implementation of `e1071` [38], the KNN implementation of `kknn` [39] and the random forest implementation of `ranger` [40]. `Rankcluster` [41] is used to calculate the Kendall distances. For visualization, `ggplot2 3.5.0`, `ggh4x` and `ggmosaic` [42–44] are used. The simulations were conducted on the local compute cluster of the Department of Statistics at TU Dortmund University. The `batchtools 0.9.17` package [45] was used for distributed computing.

The full code and simulation results for the simulation study are available on Zenodo (https://doi.org/10.5281/zenodo.13707473).

Results

In the following, the results are presented. First, the true values for the performance measures are presented. Then, the errors of the comparison studies in estimating the performance of the classifiers are discussed. Afterward, the errors of the comparison studies in estimating the

method ranking are presented. Finally, the results are summarized by analyzing the proportion of acceptable estimates. This analysis of the proportion of acceptable estimates summarizes the results regarding the errors of the comparison studies in estimating the performance of the classifiers concisely and can be understood without the more detailed discussion of the results before.

True performances and method rankings

In Tables 4 to 7, the true performance measures and rankings for the classifiers are presented for each value of $p = 2, 10, 50, 150$. For low p, all classifiers achieve high performance according to all measures. For the highest $p = 150$, the performance decreased to moderate performance measure values. In general, the differences between the methods are not large. The ranks are obtained based on the true performance measure values such that rank one always corresponds to the best performance regarding the respective measure.

For $p = 2$, LASSO performs best with regard to many performance measures, followed by SVM, Ridge, KNN, and RF. For $p = 10$, the method order is Ridge, LASSO, SVM, KNN, and RF except for the Brier score for which SVM and Ridge are swapped and for the sensitivity, for which LASSO, KNN, Ridge, and SVM are swapped. For $p = 50$, Ridge and SVM are performing best, followed by LASSO, RF, and KNN. For $p = 150$, SVM performs best with respect to all measures, and LASSO performs worst according to all except for the Brier score. The remaining ranking differs more between the performance measures.

Method failure

Within the comparison studies, fitting the classification methods to the simulated data may fail, which typically results in a warning message in case of non-convergence or in an error message in case no fit could be obtained at all. For Ridge and LASSO, non-convergence is an issue. Moreover, both models can not be fit if the data does not contain observations of both classes. In that case, SVM also outputs an error message. The random forest is still fit but outputs a warning message. KNN did not encounter any errors or warnings. In case of an error message, the fallback learner that always predicts the majority class is used. The numbers of iterations out of the total 100 iterations in which any warning or error message per scenario and classifier are given in Table A.1 to Table A.7 in Section A of S2 Appendix. Note that not necessarily all folds are affected.

Table 4. True values for performance measures for the five classifiers averaged over 500 runs under the true scenario for $p = 2$ and $n = 100$.

	Ridge	LASSO	SVM	KNN	Random Forest
True Accuracy	0.9421	0.9512	0.9431	0.9345	0.9231
True AUC	0.9910	0.9924	0.9854	0.9726	0.9804
True Brier score	0.0625	0.0397	0.0446	0.0503	0.0594
True F_1-score	0.9243	0.9383	0.9280	0.9179	0.9032
True Sensitivity	0.9048	0.9326	0.9245	0.9184	0.9033
True Specificity	0.9652	0.9627	0.9539	0.9432	0.9330
True Rank Accuracy	3	1	2	4	5
True Rank AUC	2	1	3	5	4
True Rank Brier score	5	1	2	3	4
True Rank F_1-score	3	1	2	4	5
True Rank Sensitivity	4	1	2	3	5
True Rank Specificity	1	2	3	4	5

https://doi.org/10.1371/journal.pone.0322887.t004

Table 5. True values for performance measures for the five classifiers averaged over 500 runs under the true scenario for $p = 10$ and $n = 100$.

	Ridge	LASSO	SVM	KNN	Random Forest
True Accuracy	0.8982	0.8885	0.8832	0.8588	0.8565
True AUC	0.9628	0.9533	0.9510	0.9300	0.9286
True Brier score	0.0909	0.0897	0.0861	0.1134	0.1165
True F_1-score	0.8693	0.8642	0.8565	0.8355	0.8112
True Sensitivity	0.8641	0.8776	0.8624	0.8694	0.7920
True Specificity	0.9136	0.8913	0.8944	0.8433	0.8934
True Rank Accuracy	1	2	3	4	5
True Rank AUC	1	2	3	4	5
True Rank Brier score	3	2	1	4	5
True Rank F_1-score	1	2	3	4	5
True Rank Sensitivity	3	1	4	2	5
True Rank Specificity	1	4	2	5	3

https://doi.org/10.1371/journal.pone.0322887.t005

Table 6. True values for performance measures for the five classifiers averaged over 500 runs under the true scenario for $p = 50$ and $n = 100$.

	Ridge	LASSO	SVM	KNN	Random Forest
True Accuracy	0.7968	0.7391	0.7944	0.7099	0.7213
True AUC	0.8889	0.7954	0.8698	0.7624	0.7888
True Brier score	0.1489	0.1756	0.1418	0.2025	0.1938
True F_1-score	0.6744	0.5858	0.7271	0.5822	0.5719
True Sensitivity	0.6408	0.5610	0.7161	0.5626	0.5334
True Specificity	0.8636	0.8190	0.8283	0.7692	0.8066
True Rank Accuracy	1	3	2	5	4
True Rank AUC	1	3	2	5	4
True Rank Brier score	2	3	1	5	4
True Rank F_1-score	2	3	1	4	5
True Rank Sensitivity	2	4	1	3	5
True Rank Specificity	1	3	2	5	4

https://doi.org/10.1371/journal.pone.0322887.t006

Table 7. True values for performance measures for the five classifiers averaged over 500 runs under the true scenario for $p = 150$ and $n = 100$.

	Ridge	LASSO	SVM	KNN	Random Forest
True Accuracy	0.6628	0.6323	0.7103	0.6327	0.6617
True AUC	0.7378	0.6263	0.7676	0.6552	0.7072
True Brier score	0.2181	0.2237	0.1914	0.2439	0.2169
True F_1-score	0.4809	0.4454	0.6467	0.5175	0.5588
True Sensitivity	0.5061	0.4813	0.6448	0.5031	0.5562
True Specificity	0.6839	0.6602	0.7167	0.6788	0.6755
True Rank Accuracy	2	5	1	4	3
True Rank AUC	2	5	1	4	3
True Rank Brier score	3	4	1	5	2
True Rank F_1-score	4	5	1	3	2
True Rank Sensitivity	3	5	1	4	2
True Rank Specificity	2	5	1	3	4

https://doi.org/10.1371/journal.pone.0322887.t007

For $p = 2$, no error messages are encountered in any scenario. There are only a few warning messages for Ridge and LASSO (Table A.1 in Section A of S2 Appendix). For $p = 10$, LASSO encountered many warnings (some in every scenario) which might indicate convergence

issues. Ridge also encountered many warnings in all scenarios except for correlation and shift alternatives. For data generated from a standard normal distribution or with a scale of 0.25 or 0.5 it happened that no zeroes or no ones were generated. These are the cases where SVM encounters an error and RF a warning (Table A.2, A.3 in Section A of S2 Appendix). For the scale of 0.25 these are all iterations. For $p = 50$, it looks similar, but there are more warnings for RF and Ridge. For the scale of 0.25 and 0.5, and for a shift of ± 0.5, in almost all iterations either no ones or no zeroes are generated (Table A.4, A.5 in Section A of S2 Appendix). For the scale of 4, also many iterations are affected. For $p = 150$, again many warnings are encountered for Ridge and LASSO. For the scale of 0.25, again no zeroes or ones are generated in any iteration. For the scale of 0.5 and for a shift of ± 0.5 this happens in around one-third of the iterations (Table A.6, A.7 in Section A of S2 Appendix).

Errors in estimation of performance measures

In the following, the performance of parametric and Plasmode simulation regarding the estimation of the classification performance measures for all classifiers are compared under different scenarios and using different resampling types for Plasmode. First, the resampling types for the Plasmode simulation are compared. Afterwards, the influence of different misspecifications of the data-generating process (DGP) and outcome-generating model (OGM) are discussed. The results are always presented according to the number of variables p and according to the classification performance measure. For each combination of p and each classification performance measure, the errors for estimating this measure are visualized using boxplots stratified by the simulation type, classifier, and potentially by the type of misspecification. This section gives a detailed discussion of the results. A summary of the results can be found afterward in the presentation of the proportions of acceptable estimates. That section can also be understood independently of the following detailed analysis of the results.

Comparison of resampling strategies for Plasmode. In the following, under the true scenario, the errors in estimating the classification performances are compared between parametric simulation and Plasmode simulation with different resampling types. This comparison shows how well each simulation approach can perform at best when no wrong assumptions are made in the comparison study. The differences between the simulated accuracy and the true accuracy are displayed in a boxplot over 100 iterations. Ideally, the errors should all be close to zero since the assumptions in the simulations coincide with the truth. Positive values correspond to overestimation, and negative values to underestimation. The columns in each plot correspond to the type of simulation. The rows correspond to the classifier.

Overall, it can be seen that parametric simulation is often superior to Plasmode simulation as the median relative errors are typically closer to zero and the boxes are narrower indicating more stable estimation. In general, Plasmode seems to perform worse compared to parametric for increasing p. Compare for example the estimation errors for accuracy for $p = 2$ for the two Bootstrap types in Fig 2 to the corresponding ones for $p = 150$ in Fig 3.

In many cases, some Plasmode variant performs similarly well but no variant is consistently as good as parametric across all classifiers, measures, and values of p. There is no clear structure when which Plasmode variant performs well but often 0.632-subsampling performs worst for $p = 10$ (Figs B.6 to B.11 in Section B of S2 Appendix) and one of the Bootstrap types performs worst for $p = 50$ (Figs B.12 to B.22 in Section B of S2 Appendix). No resampling performs satisfactorily in most cases with few exceptions (e.g. for $p = 150$ and the F_1-score, see Fig B.20). Which of the Bootstrap types to prefer depends on the concrete situation but often 0.632-Bootstrap is preferable over the ordinary Bootstrap. For example, 0.632-Bootstrap performs often well for $p = 10$, and often worse but still better than ordinary Bootstrap for $p = 50$,

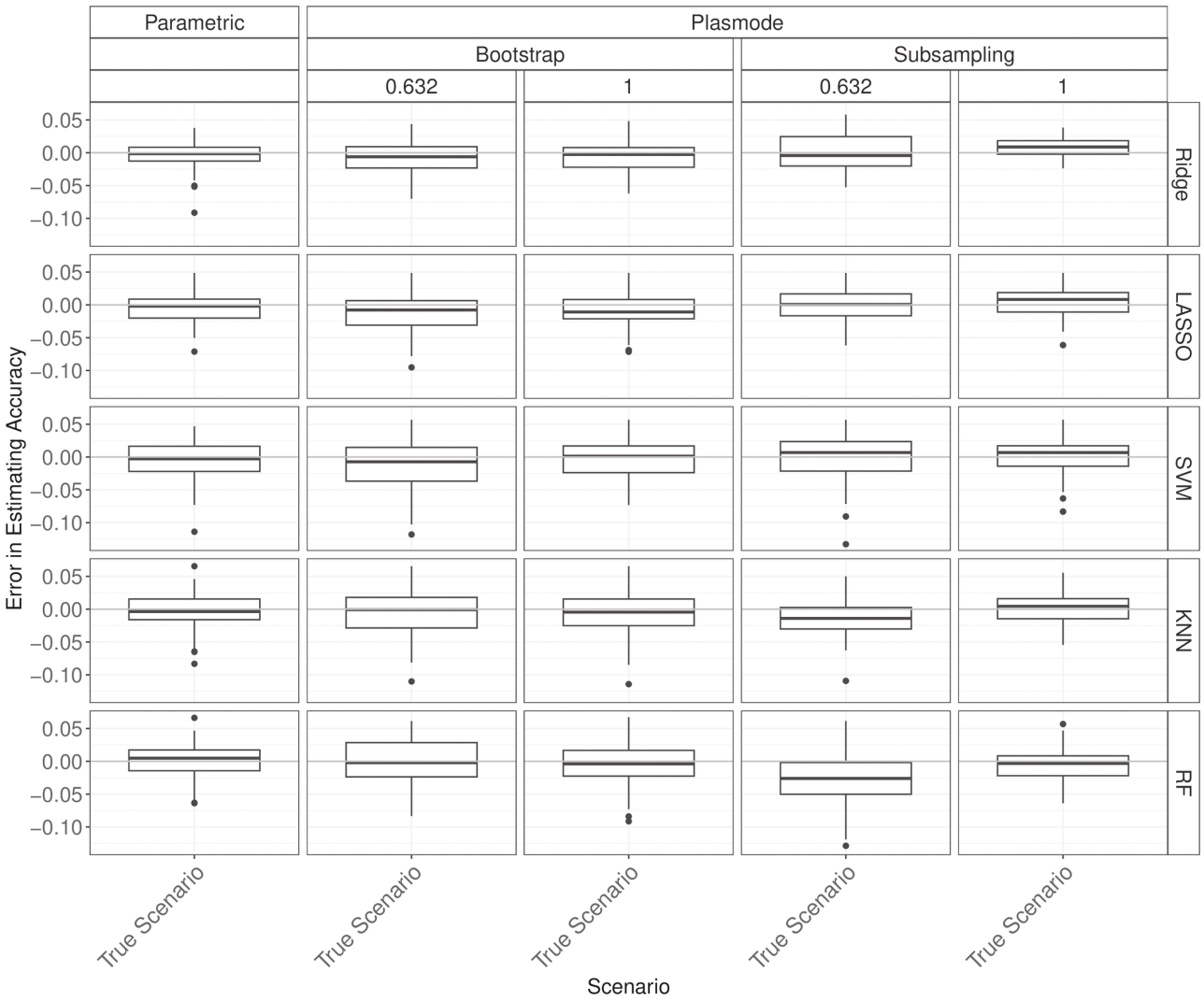


Fig 2. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches under the true scenario for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g002

see Figs B.6 to B.11 and B.12 to B.17. Moreover, there is a tendency towards larger errors for all simulation types for the F_1-score, specificity, and sensitivity, especially for high p and especially for Ridge and LASSO as classifiers. A reason for this might be that we observed that especially Ridge and LASSO tend to predict only ones or only zeroes when p gets larger and n is kept constant.

The plots for all combinations of p and classification performance measures can be found in Section B of S2 Appendix.

Shift. In the following, it is discussed how misspecifying the shift in the DGP affects the ability of the parametric simulations to estimate the classification performances. For $p = 2$ and $p = 10$, almost no differences are visible between the errors for shifted data and the errors under the true scenario (see Figs C.1 to C.12 in Section C of S2 Appendix). For $p = 2$, one of the coefficients is positive and the other is negative with similar absolute values. Therefore, the effects of shifting both variables by the same amount might cancel out to some extent.

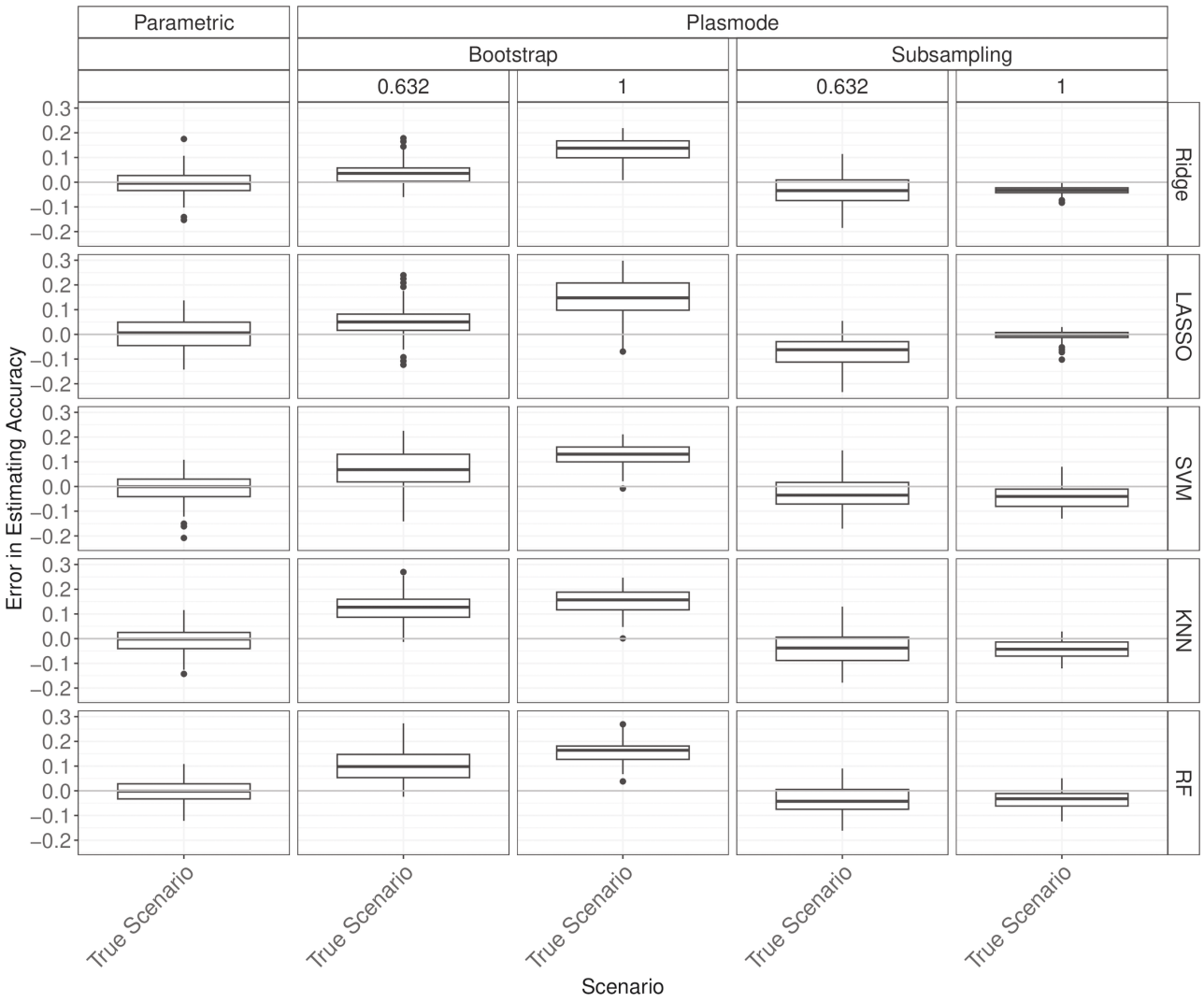


Fig 3. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches under the true scenario for $p = 150$.

https://doi.org/10.1371/journal.pone.0322887.g003

Fig 4 shows the errors in estimating the accuracy for $p = 50$ and misspecifications for shift for the parametric simulation. Note that for shifts of ±0.5, Ridge, LASSO, and SVM failed in almost all iterations and the fallback learner was used instead (see Table A.5 in Section A of S2 Appendix). An overestimation of the true accuracy can be observed for all shifts. The errors for parametric simulation based on shifted data quickly get worse than the well-performing Plasmode variants with resampling proportions of one.

For AUC and the Brier score, an inverted pattern can be observed (see Figs C.13 and C.14 in Section C of S2 Appendix). The errors for the F_1-score and sensitivity estimation increase with increasing shifts while the errors for specificity decrease with increasing shifts (see Fig 5, and Figs C.15 to C.16 in Section C of S2 Appendix). A possible explanation for these observations is that shifting the data seems to result in higher predicted probabilities and therefore more generated ones for positive shifts and lower predicted probabilities and therefore a higher proportion of zeroes. If a method then, e.g. for a positive shift, predicts mostly high

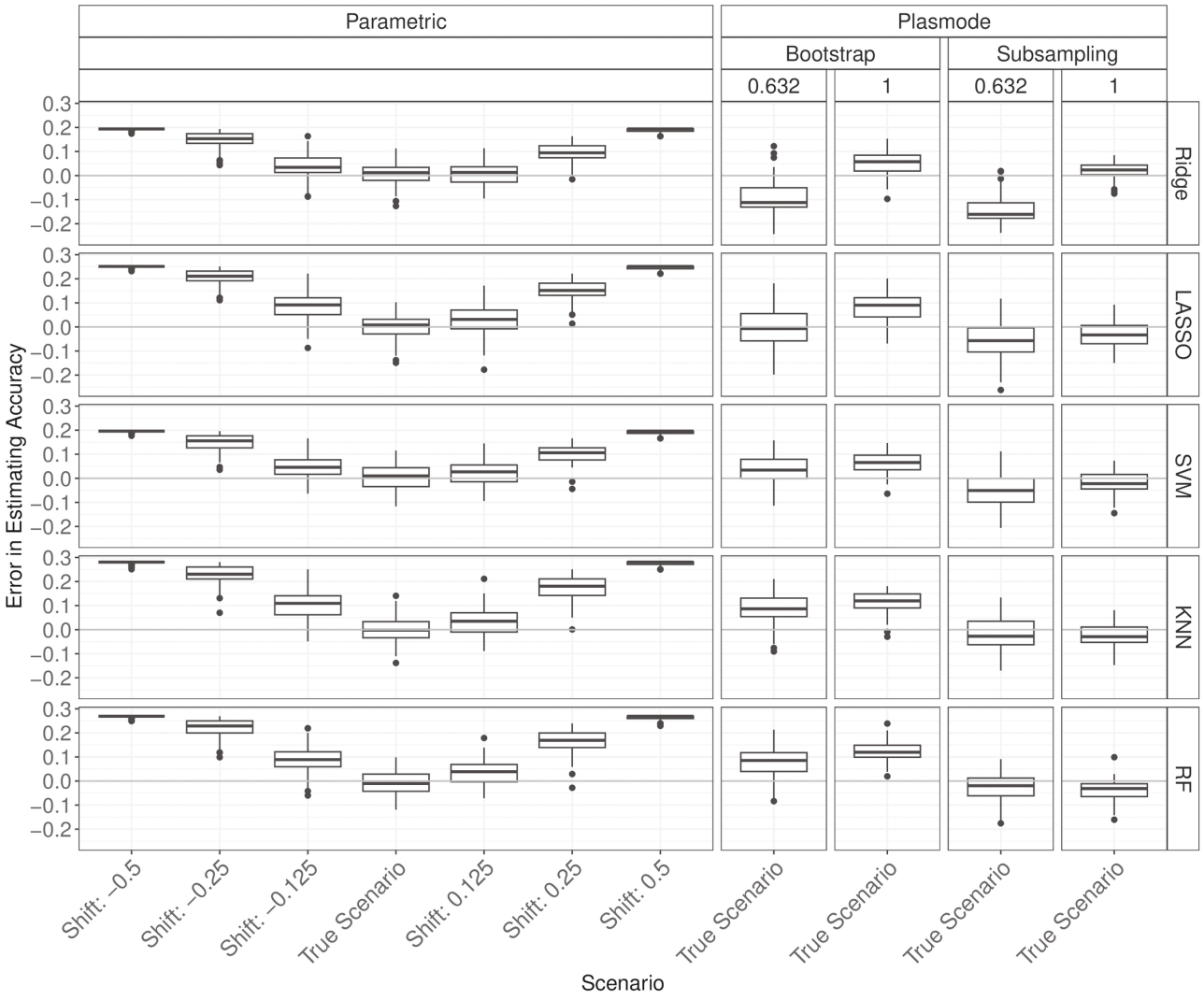


Fig 4. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecified shift for parametric simulation for $p = 50$.

https://doi.org/10.1371/journal.pone.0322887.g004

probabilities it will achieve high performance with respect to accuracy, sensitivity, and F_1 score as it correctly predicts the ones. On the other hand, the specificity decreases as true zeroes are often also predicted as one. For the extreme shifts and Ridge, LASSO, and SVM this happens since the fallback learner that always predicts the majority class is used in almost all iterations.

The results for $p = 150$ are similar to those for $p = 50$ (see Figs C.17 to C.22 in Section C of S2 Appendix).

Scale. Fig 6 shows the errors in the estimation of accuracy when the scale in the parametric simulation is misspecified for $p = 2$. The true accuracy is underestimated for scales smaller than one and overestimated for scales larger than one.

For the AUC, the underestimation for small scales remains, but only slight differences can be observed for scales larger than one (see Fig D.1 in Section D of S2 Appendix), possibly because overestimation is almost impossible due to the true high AUCs. For the Brier score,

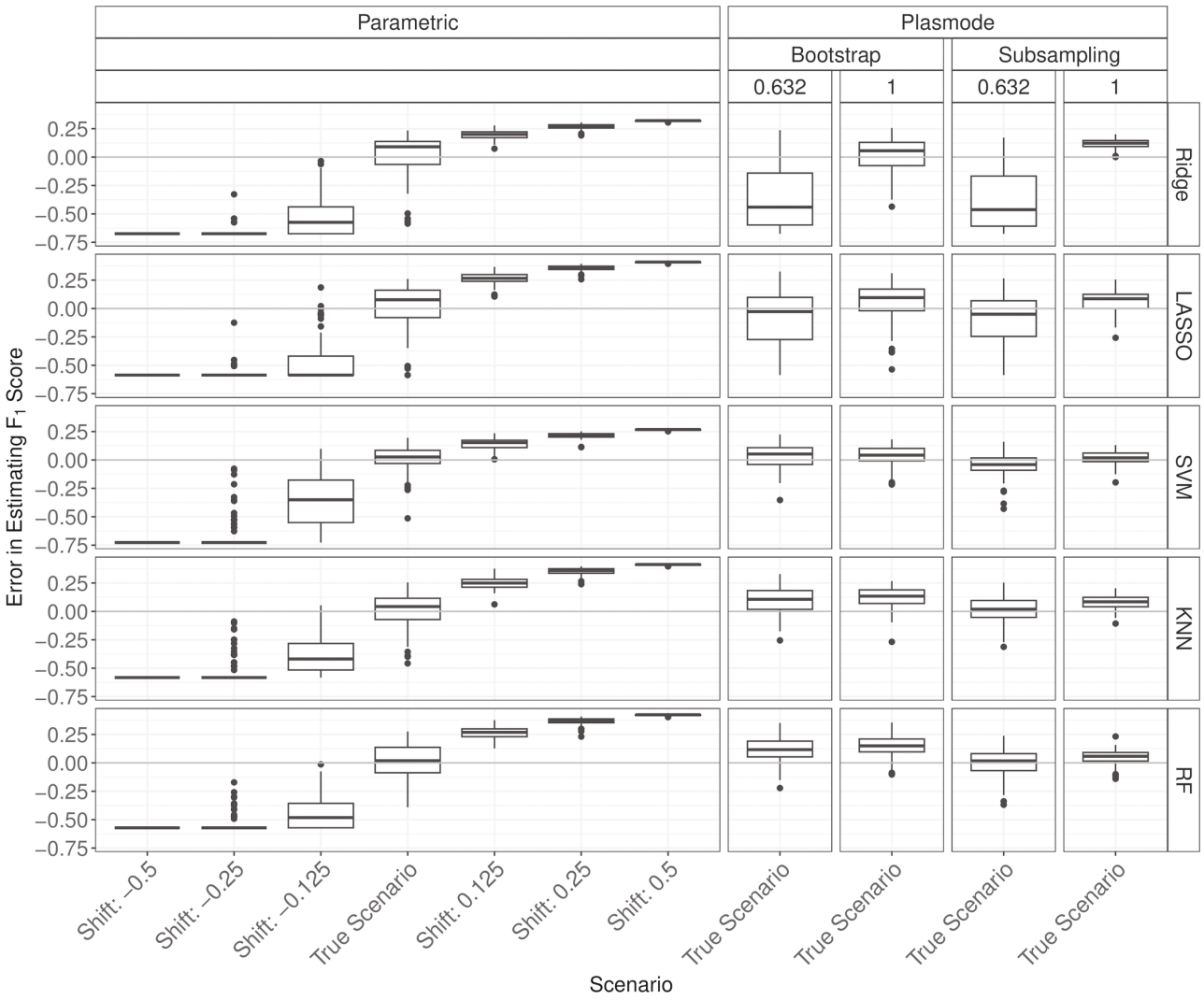


Fig 5. Errors in the estimation of the F_1 score in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecified shift for parametric simulation for $p = 50$.

https://doi.org/10.1371/journal.pone.0322887.g005

the results are similar to the ones for accuracy but with inverted signs (Fig D.2 in Section D of S2 Appendix). The results for F_1-score and sensitivity are similar to those for the AUC (Figs D.3 and D.4 in Section D of S2 Appendix). For specificity, an increase of the errors with increasing scale can be observed (Fig D.5 in Section D of S2 Appendix).

For $p = 10$, more extreme estimation errors compared to $p = 2$ can be observed for all measures, especially for the F_1-score and sensitivity. The direction of over- and underestimation is not always consistent with that observed for $p = 2$ (see Figs D.6 to D.11 in Section D of S2 Appendix). Note that for small scales, especially for the scale of 0.25, Ridge, LASSO, and SVM failed in many up to all iterations, see Table A.3 in Section A of S2 Appendix.

For $p = 50$, and for a scale of 0.25, no AUCs and sensitivities could be estimated in any iteration, i.e. no ones were generated even after redrawing the data 50 times.

The true accuracy is overestimated for all scales, while the AUC and Brier score are underestimated (Figs D.13 to D.15 in Section D of S2 Appendix). For the F_1-score and sensitivity,

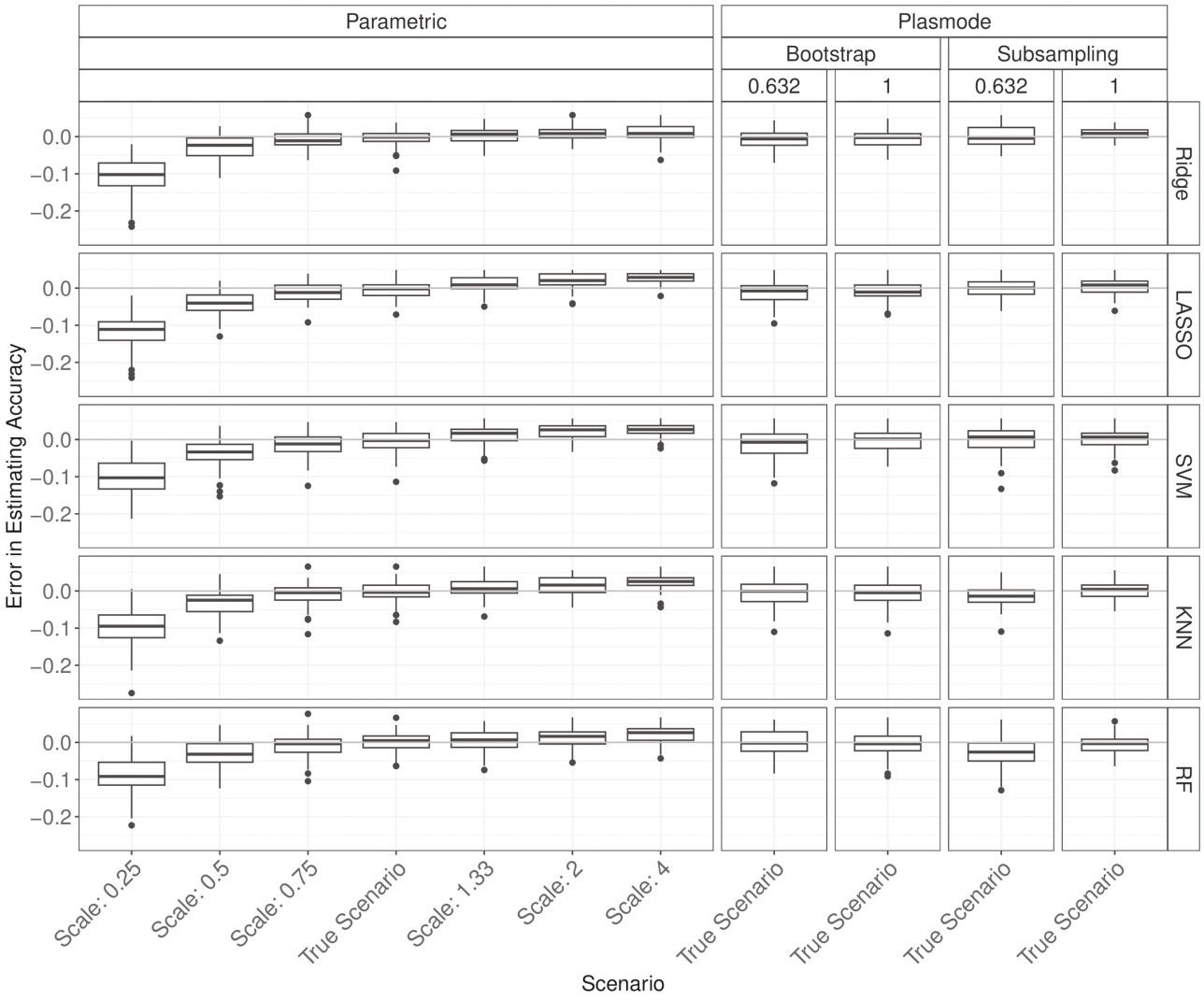


Fig 6. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecified scale for parametric simulation for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g006

severe underestimation can be observed for small scales and overestimation for large scales (Figs D.15, D.16 in Section D of S2 Appendix). In the extreme case of small scales, the estimated measure is always close to zero, and for large scales always close to one. For specificity, the pattern observed for the F_1-score and sensitivity is inverted (Fig D.17 in Section D of S2 Appendix).

The results regarding the estimation errors are very similar to those for $p = 50$ (see Figs D.18 to D.23 in Section D of S2 Appendix).

Overall, we observe that misspecifying the scale in the DGP for parametric simulation often results in errors that are larger than the errors for Plasmode simulation for which we cannot misspecify the DGP directly.

Correlation. In the following, the effect of changing the correlation in the DGP for parametric simulation is discussed. For $p = 2$, there are almost no differences visible when changing the correlation structure except for slight deviations for fixed pairwise correlations of 0.2 (see Figs E.1 to E.6 in Section E of S2 Appendix).

For $p = 10$, almost no differences are visible (see Figs E.7 to E.12 in Section E of S2 Appendix).

For $p = 50$, there are slight differences visible except for specificity. The true measures are overestimated for a correlation of 0.2 and slightly underestimated for the other correlation values (vice versa for the Brier score). For accuracy, this is shown in Fig 7. The results for all other measures can be found in Figs E.13 to E.17 in Section E of S2 Appendix.

For $p = 150$, there are also slight errors in almost all cases (Figs E.18 to E.23 in Section E of S2 Appendix).

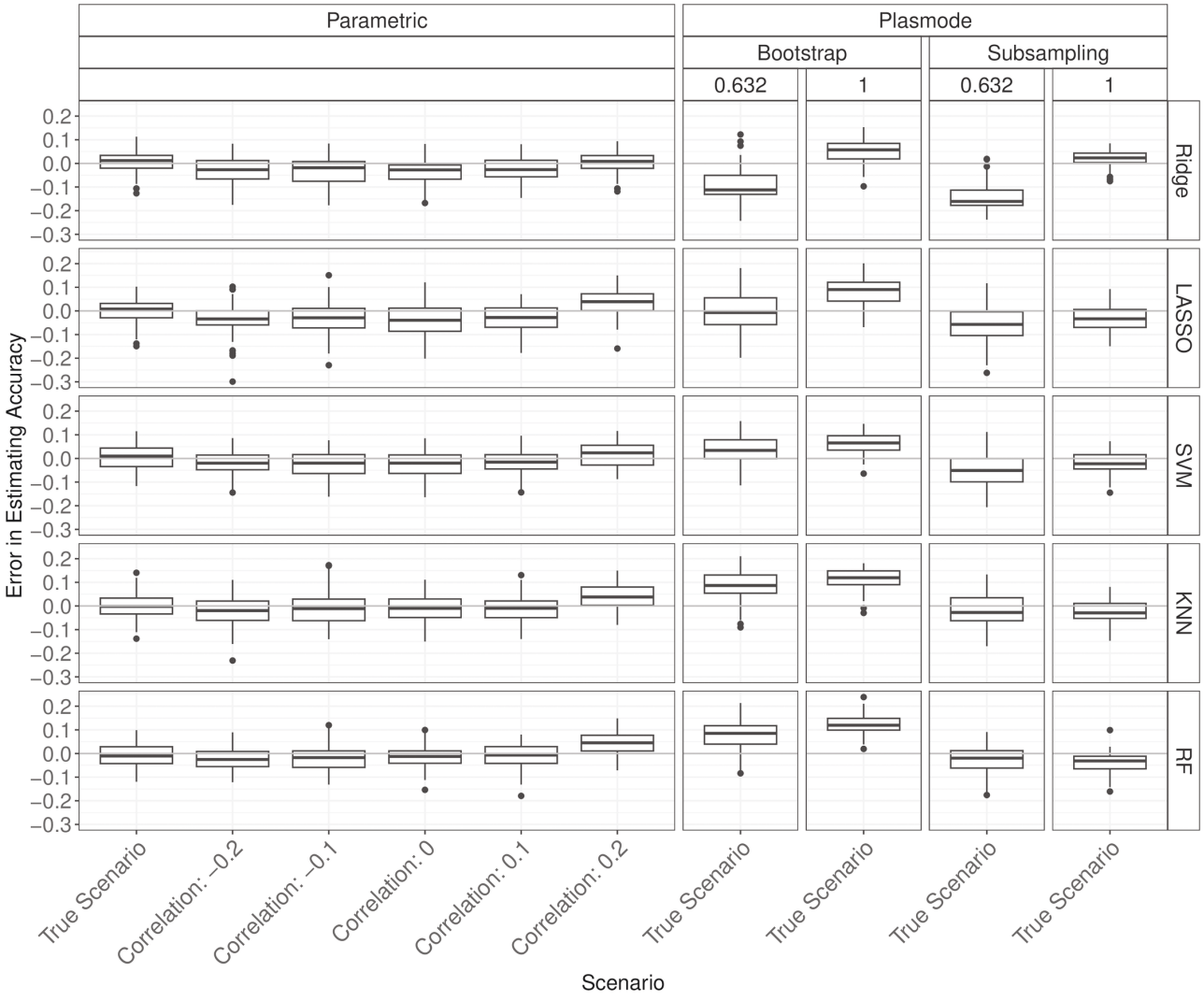


Fig 7. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecified correlation for parametric simulation for $p = 50$.

https://doi.org/10.1371/journal.pone.0322887.g007

Overall, changing the correlation does not seem to affect the simulation results for parametric simulation in many cases. Often the errors made by misspecifying the correlation are still smaller than those of Plasmode simulation under the true scenario. However, it should be noted that the misspecifications here are not very large as the true correlations are mostly scattered around zero. Using correlations further away from the truth might lead to larger errors for parametric simulation but could not be investigated because of numerical problems as discussed in Section Deviations.

Complete misspecification as standard normal. For $p = 2$, when misspecifying the distribution as a standard normal there are some differences visible for almost all measures and classifiers. The errors for accuracy, Brier score, and specificity are notable (Figs 8, and F.2, F.5 in Section F of S2 Appendix). For AUC, F_1-score, and sensitivity only slight over- or underestimation occurs (Figs F.1, F.3, F.4 in Section F of S2 Appendix).

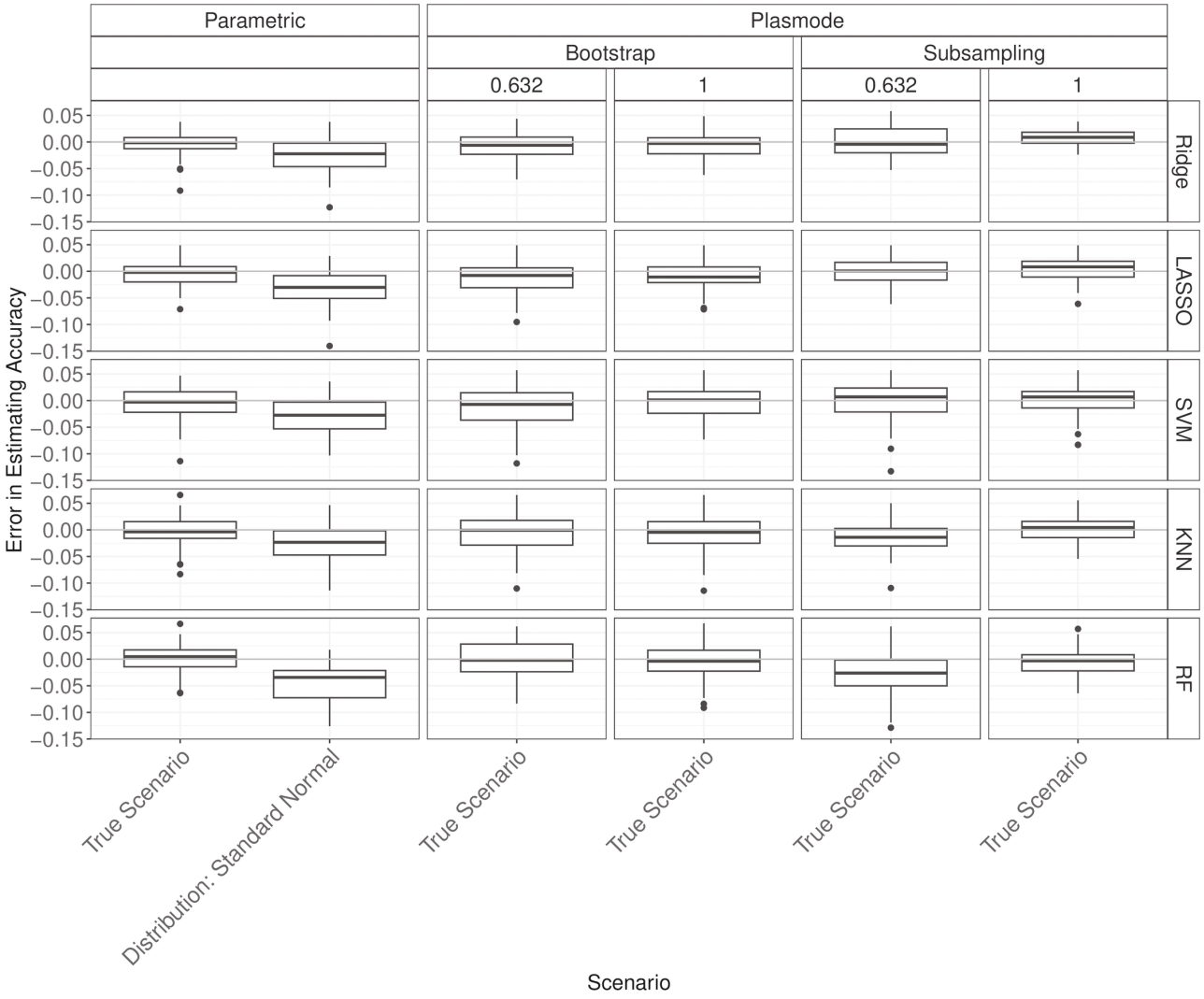


Fig 8. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with distribution misspecified as standard normal for parametric simulation for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g008

The true accuracies and specificities are overestimated notably while the AUCs, Brier scores, F_1-scores, and sensitivities are underestimated notably (see Figs F.6 to F.11 in Section F of S2 Appendix).

For $p = 50$, slight errors are observed for accuracy, AUC, and Brier score (see Figs F.12 to F.14 in Section F of S2 Appendix). For F_1-score and sensitivity, underestimation can be seen and for specificity, overestimation can be seen (see Figs F.15 to F.17 in Section F of S2 Appendix).

At least slight over- or underestimation can be observed in almost all cases for $p = 150$ (Figs F.18 to F.23 in Section F of S2 Appendix). For KNN, the median errors in estimating accuracy, AUC, and Brier score are almost zero. Note that the standard normal distribution approximates the distribution of many variables in the true DGP reasonably. This might explain why the resulting errors for parametric simulation are often not very large.

OGM. In the following, the effect of misspecifying the OGM on the results of both parametric and Plasmode are presented. For $p = 2$, an increase in the errors from underestimation for OGMs scaled by 0.5 to (slight/very slight) overestimation for scaled by 2 and considerable underestimation for setting all coefficients to 0 can be observed for all measures except the Brier score for parametric as well as Plasmode simulation (Fig 9 below, and Figs G.1, G.3 to G.5 in Section G of S2 Appendix). For the Brier score, the pattern is inverted (Fig G.2 in Section G of S2 Appendix).

The results for $p = 10$ are similar to those for $p = 2$ (see Figs G.6 to G.11 in Section G of S2 Appendix).

For $p = 50$, the results are mostly similar to those for $p = 2$ and 10 (see Figs G.12 to G.17 in Section G of S2 Appendix). In part, overestimation for sensitivity and F_1-score for the model with all coefficients set to zero can be seen, and the increase for 0.5 and 2 is less clear, in part even with a decrease for 2 again.

For $p = 150$, results are again similar to those for $p = 2, 10$ but the pattern for scaled OGMs is less consistent (see Figs 10, and G.18 to G.22 in Section G of S2 Appendix). In the parametric case often no difference between these models scaled by 0.5 and 2 and the true OGM is visible. For Plasmode, typically some difference can be observed, but not necessarily an increase (see e.g. Fig 10).

Errors in estimation of method ranking

In the following, the errors in estimating the true method ranking in the parametric or Plasmode comparison studies are discussed. Low ranks always correspond to good performance, independent of the measure. The Kendall distances of the simulated and true method rankings are displayed in boxplots analogously to the errors in the previous section. Fig H.1 in Section H of S2 Appendix shows the Kendall distances for 10000 pairs of randomly drawn rankings of $1, \dots, 5$ for comparison. The median Kendall distance is at 0.5 and the distribution is approximately symmetric around that value. Ideally, the simulated and true method rankings should show lower Kendall distances than these randomly drawn rankings.

When comparing the Kendall distances for $p = 2$, it can be seen that differences in the Kendall distance of the simulated and true rankings occur where notable errors in estimating the classification performance due to misspecifications of the DGP or OGM were observed previously. For example, the Kendall distance is only slightly influenced by changing the correlation as shown in Fig 11, but more heavily influenced by changing the OGM (Fig 12) or scale (Fig 13). Interestingly, the Kendall distance sometimes gets smaller for misspecifications than for the true scenario, see e.g. Fig 13. Also, the median Kendall distance of Plasmode simulation is often smaller than for parametric, especially for no resampling except for specificity

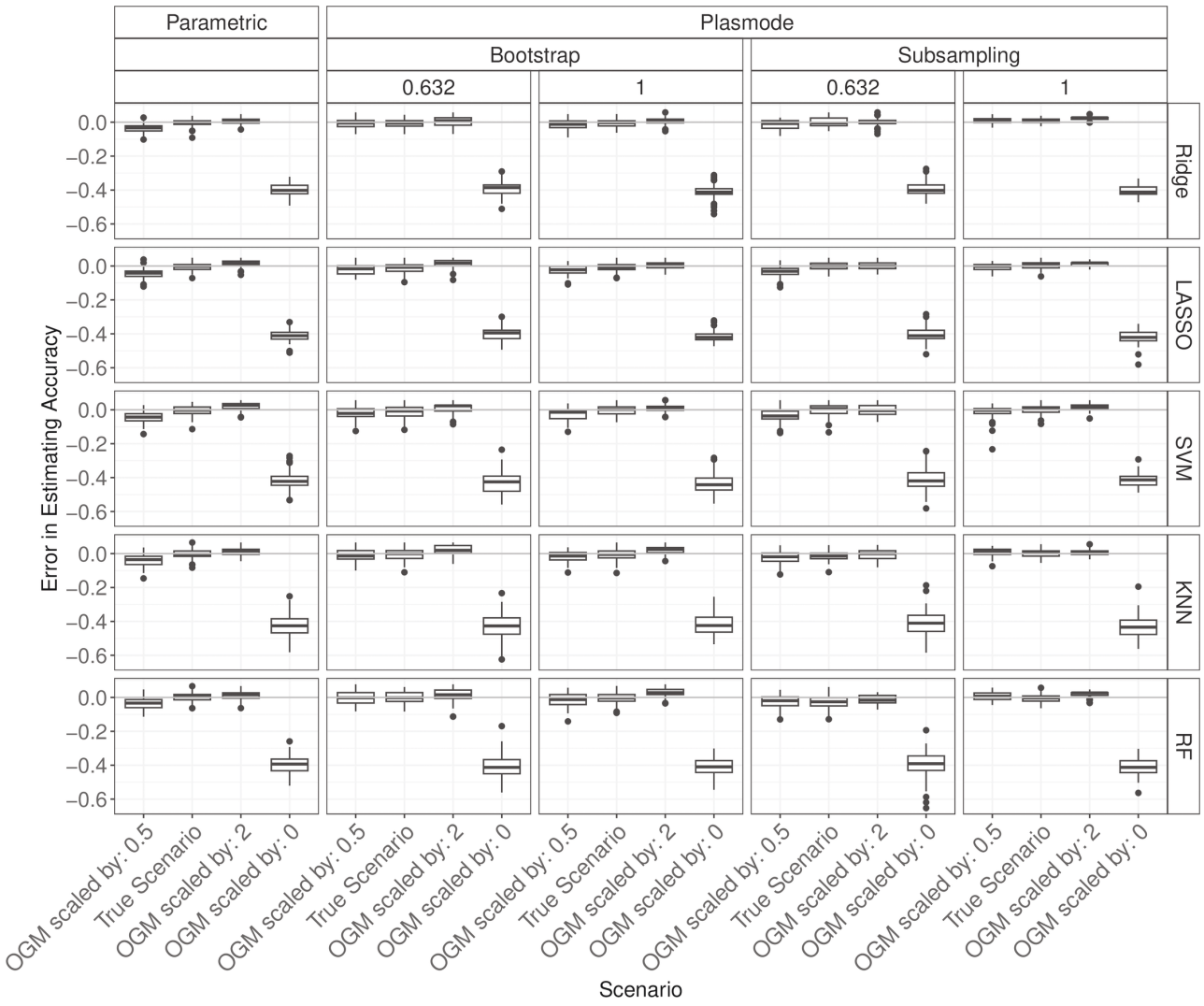


Fig 9. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecifications of the OGM for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g009

(see Figs H.2 to H.16 in Section H of S2 Appendix). Overall, the results are more volatile than those for the estimation errors.

For $p > 2$, the results are similar to those for $p = 2$ but Plasmode is usually performing worse than parametric again for increasing p under the true scenario (see Fig in Section H in S2 Appendix).

Proportion of acceptable simulation results

To summarize the above findings, the proportions of acceptable estimates are discussed next. Therefore, the proportion of iterations with relative errors within the 2.5% to 97.5%-quantile interval of the relative errors for the parametric comparison studies under the true scenario are calculated, for each combination of p, simulation type, classifier, and measure for each

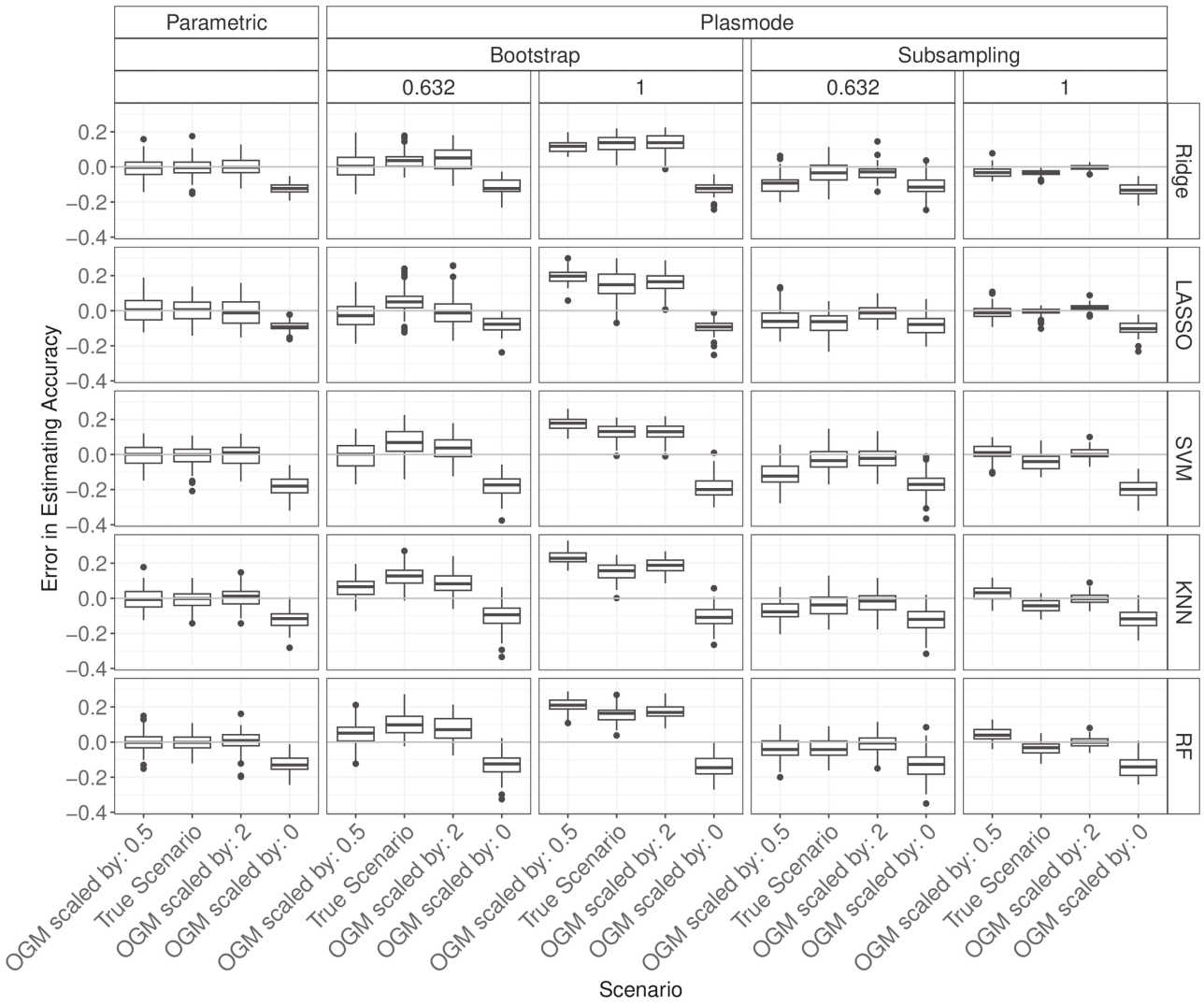


Fig 10. Errors in the estimation of accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecifications of the OGM for $p = 150$.

https://doi.org/10.1371/journal.pone.0322887.g010

deviation. The resulting proportions are displayed in heatmaps for all scenarios. Each cell corresponds to one combination of simulation type, measure, classifier, and scenario, as indicated by the facet and axis labels. Violet-colored cells indicate high proportions and thus a good performance. Pink, red, and orange colors already indicate increasingly worse performance, and yellow indicates that almost all iterations yielded unacceptable results.

The results for $p = 2$ are shown in Fig 14. It can be seen that in many cases the proportion of acceptable iterations is high. For all simulation types, the proportion of acceptable iterations is very low when all coefficients of the OGM are set to zero. Moreover, for all measures except for the specificity, the proportion is low for a scale of 0.25. Depending on the measure and classifier there are also moderate proportions for the higher values of scale, for setting the distribution to standard normal, and for the other modifications of the OGM. For Plasmode and the true scenario, we also observe some moderate values, especially for the 0.632 resamplings. No resampling performs very well here, except for the Brier score.

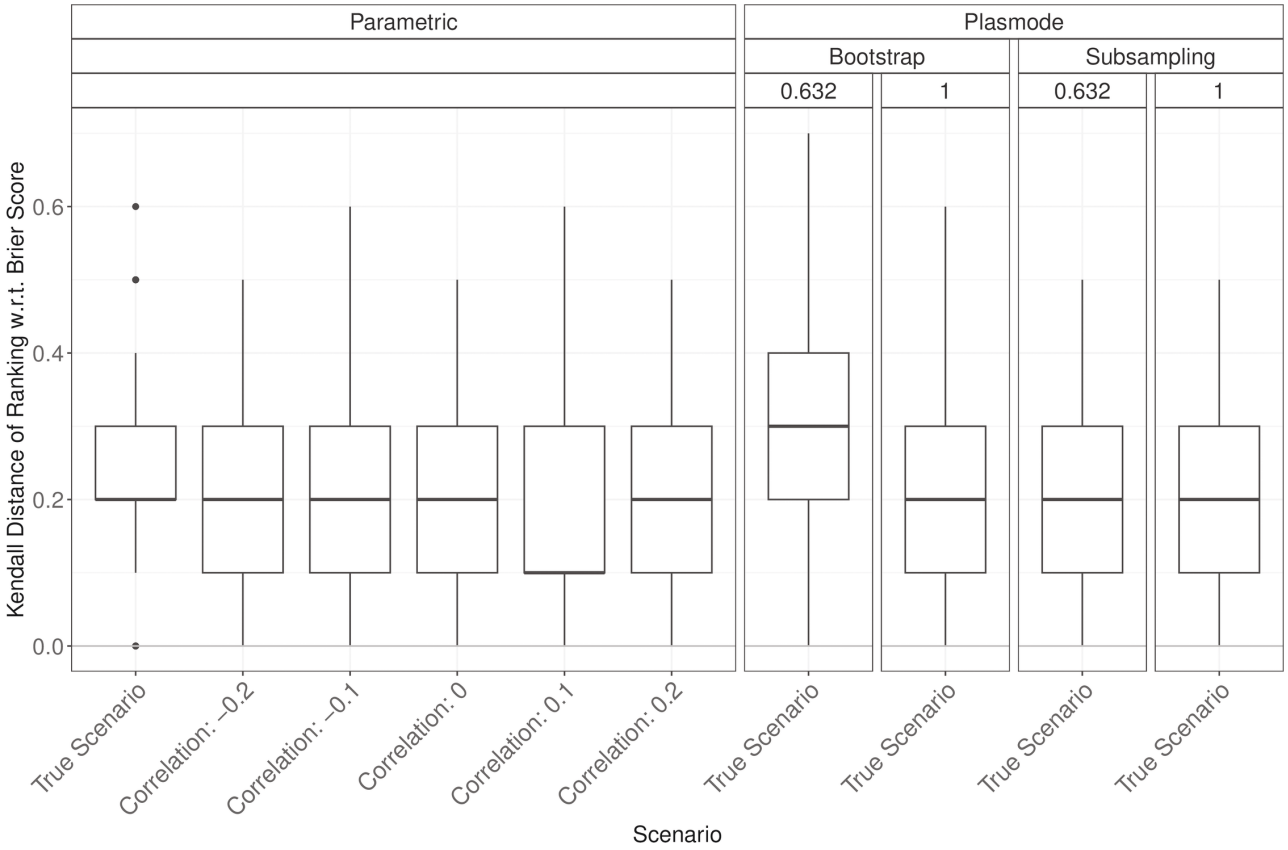


Fig 11. Kendall distance of the simulated and true method ranking based on the Brier score in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecifications of the correlation for parametric simulation for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g011

Fig 15 shows the results for $p = 10$. Compared to $p = 2$, many proportions of acceptable iterations decrease. Especially for scale alternatives and for setting the distribution to standard normal, smaller proportions are observed. Most Plasmode types also perform worse, except for no resampling for the true scenario.

The results for $p = 50$ as shown in Fig 16 are even worse. For shift and scale alternatives, almost all results are unacceptable. For standard normal and for all coefficients of the OGM set to zero, the proportions of acceptable estimates are also often (very) low. The performance of Plasmode except for no resampling gets worse again, especially for the Ridge model and for some measures also for RF.

Fig 17 shows the results for $p = 150$. The results are similar to those for $p = 50$. The proportions for moderate shift and scale are not as low. The proportions of acceptable iterations for the ordinary Bootstrap and accuracy, AUC, and Brier score are now very low. The performance of the 0.632-Bootstrap and no resampling also drop, but not as much. 0.632-subsampling now performs comparably well. The results for Ridge and LASSO often show high proportions of acceptable simulation results.

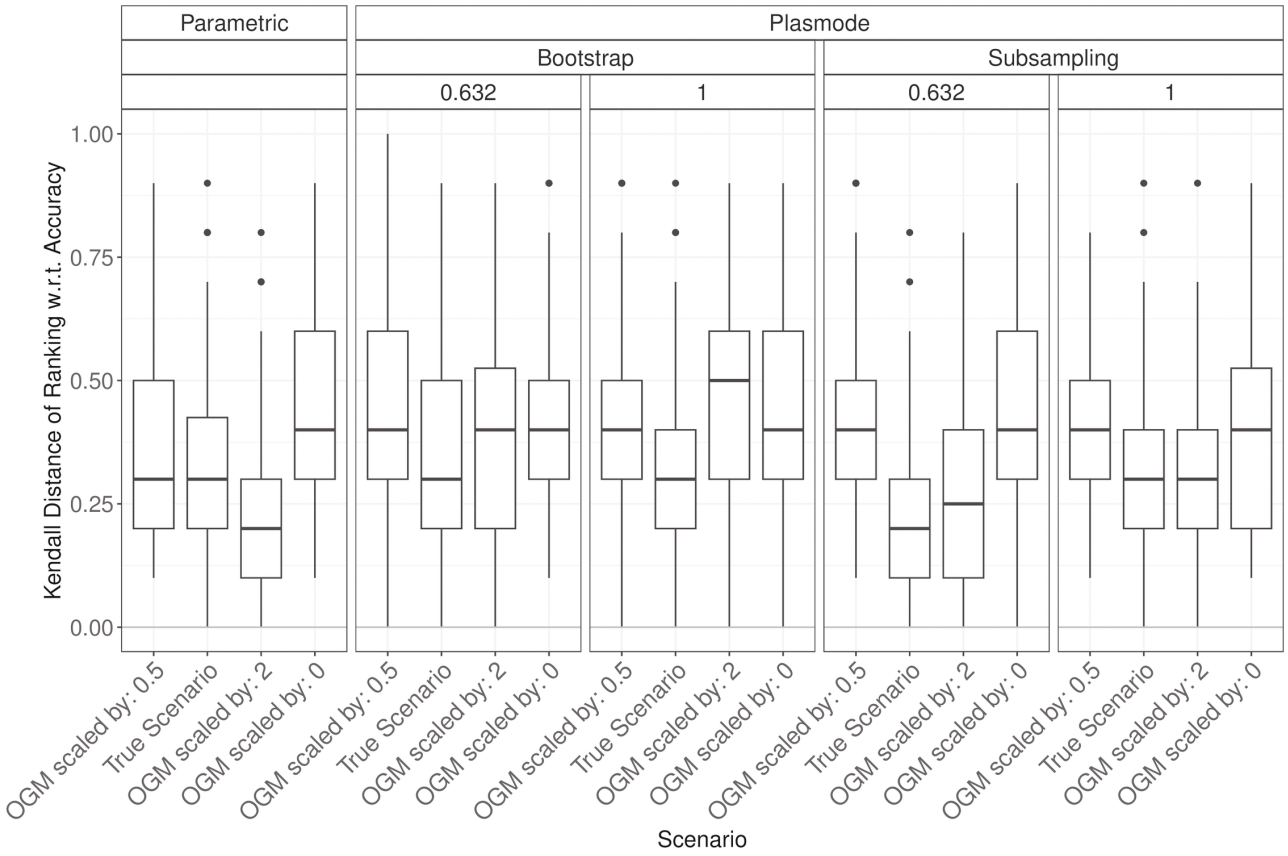


Fig 12. Kendall distance of the simulated and true method ranking based on accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecifications of the OGM for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g012

Discussion

We conducted a simulation study with the following tasks:

1. Compare how well parametric and Plasmode simulation can estimate the performance and method order for several classification methods.
2. Find out how misspecifications of the data-generating process (DGP) and outcome-generating model (OGM) affect parametric simulation in terms of estimating the performance and ranking of classification methods.
3. Find out how misspecifications of the OGM and different resampling strategies affect Plasmode simulation in terms of estimating the performance and order of classification methods.
4. Find out how the number of covariates affects the above.

Errors in the estimation of classification performance measured by accuracy, AUC, Brier score, F_1-score, sensitivity, and specificity were compared as well as errors in the estimation of the resulting method ranking of five binary classification methods including Ridge and LASSO logistic regression, support vector machine (SVM), K-nearest neighbors (KNN), and random forest (RF). Additionally, the proportion of acceptable estimates was analyzed. An iteration was defined to be acceptable if its relative errors lie within the 2.5% and 97.5%

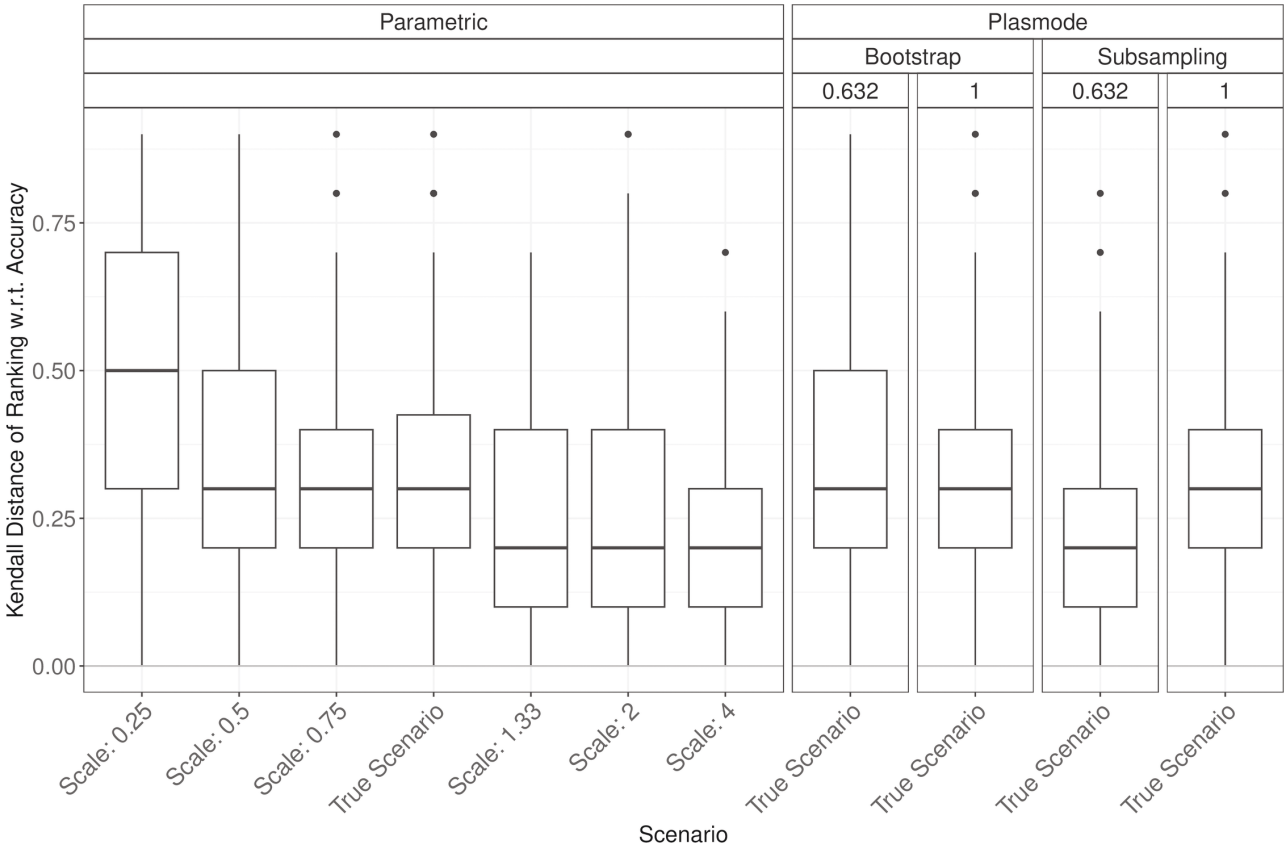


Fig 13. Kendall distance of the simulated and true method ranking based on accuracy in 100 iterations of a classification method comparison study per classifier for different simulation approaches with misspecifications of the scale for parametric simulation for $p = 2$.

https://doi.org/10.1371/journal.pone.0322887.g013

quantile interval of the errors for parametric simulation assuming the true DGP and OGM. The analyses were each conducted for sample sizes $n = 100$ and numbers of variables $p = 2, 10, 50, 150$.

For all misspecifications, some errors could be observed for at least some combination of classification performance measure, classifier, and number of variables p. The magnitude and sign of the errors depended on the exact settings. Often, the errors observed for the estimation of the Brier score were similar to those for accuracy but with inverted signs. Errors for estimating F_1-scores were similar to those for the sensitivity. A reason for this might be that the F_1-score is the harmonic mean of precision and recall, where recall equals sensitivity. Errors in estimating specificities often showed inverted patterns to those for the F_1-scores and sensitivities, probably since improved classification of true ones typically leads to a worse classification of true zeroes. In general, often, more extreme errors were observed for estimating the F_1-score, sensitivity, and specificity, while only smaller errors were observed for estimating the AUC. Misspecifications of the OGM affect both parametric and Plasmode simulations similarly. For misspecifications of the DGP, only the parametric simulation is affected. Such misspecifications of the DGP can lead to severe errors in the parametric method comparison study. In those cases, Plasmode is the more robust choice. Overall, the observed errors were more severe for larger values of p in most cases.

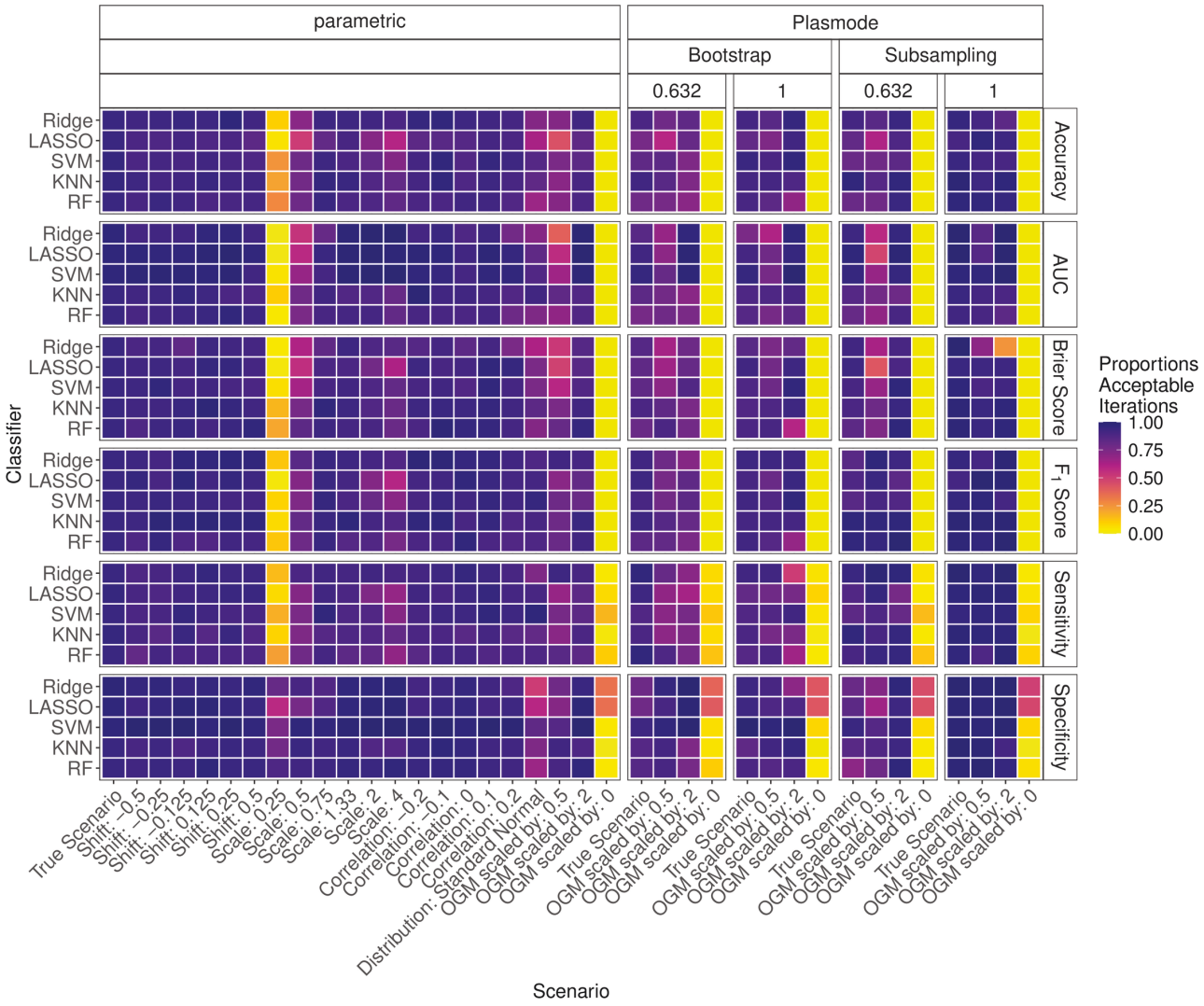


Fig 14. Proportion of acceptable iterations for 100 iterations of parametric and Plasmode method comparison studies under different scenarios for different classifiers and classification performance measures for $p = 2$. An iteration is defined as acceptable if its relative error for the respective measure lies within the 2.5% and 97.5% quantile of the parametric simulation error for the true scenario for that measure and classifier.

https://doi.org/10.1371/journal.pone.0322887.g014

With regard to the resampling strategies for Plasmode simulation, no clear conclusion could be drawn. Often, 0.632-subsampling led to comparably large errors and no resampling to comparably small errors, but this was not consistent across all classification performance measures and all values of p. Which of the Bootstrap types performed better was different depending on the specific scenario. The performance of the Plasmode simulations decreased for larger values of p.

It should be noted that the smaller the resampling proportion and the higher the number of duplicate observations in the Plasmode dataset, the less information is available for the classifier during training, which could affect its performance systematically for all resampling types except no resampling.

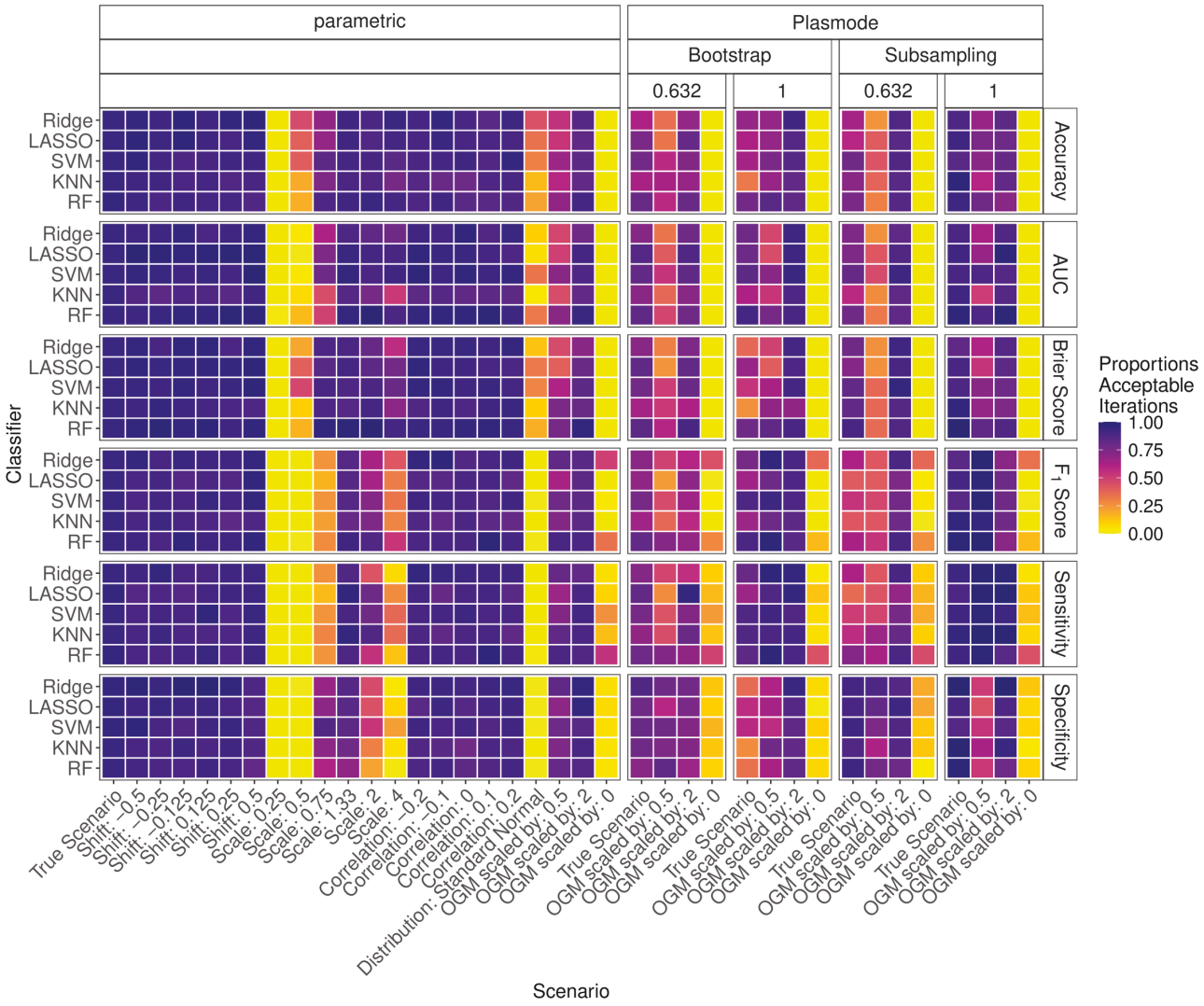


Fig 15. Proportion of acceptable iterations for 100 iterations of parametric and Plasmode method comparison studies under different scenarios for different classifiers and classification performance measures for $p = 10$. An iteration is defined as acceptable if its relative errors for the respective measure lie within the 2.5% and 97.5% quantile of the parametric simulation errors for the true scenario for that measure and classifier.

https://doi.org/10.1371/journal.pone.0322887.g015

In summary, we observed:

1. As expected, under the true scenario parametric simulation performs better than Plasmode with regard to estimating the classification performance.
2. Misspecifications of the DGP lead to errors in parametric simulation that quickly get larger than the errors for Plasmode, for which we cannot misspecify the DGP directly.
3. Misspecifications of the OGM affect parametric simulation and Plasmode simulation equally in terms of estimating the classification performance.
4. With regard to the resampling used for Plasmode, no resampling type consistently outperformed the others. However, often no resampling at all performed well and subsampling with a resampling proportion of 0.632 performed badly.
5. An increase in the number of variables decreases the ability to estimate the classification performance, especially for Plasmode simulations.

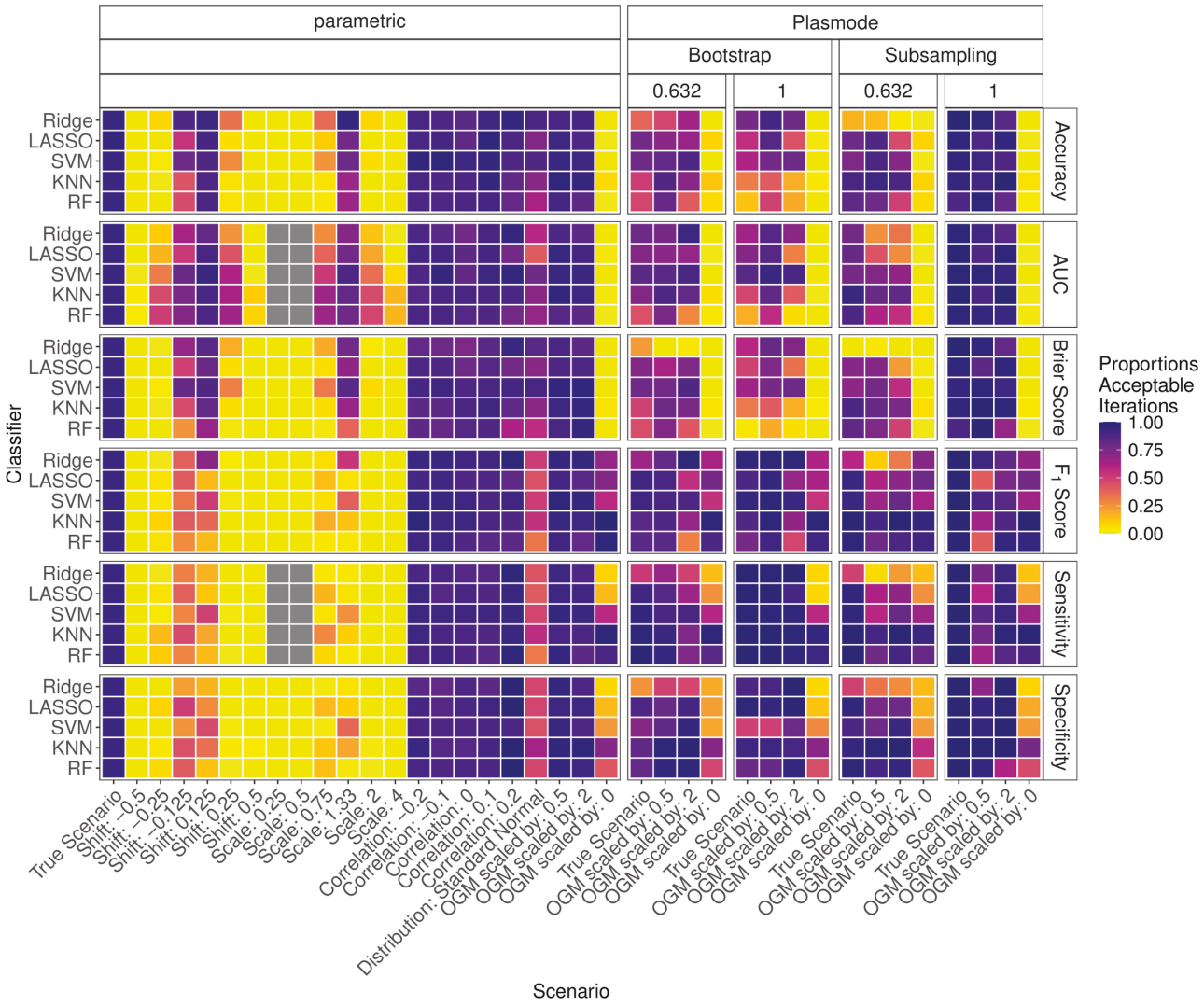


Fig 16. Proportion of acceptable iterations for 100 iterations of parametric and Plasmode method comparison studies under different scenarios for different classifiers and classification performance measures for $p = 50$. An iteration is defined as acceptable if its relative errors for the respective measure lie within the 2.5% and 97.5% quantile of the parametric simulation errors for the true scenario for that measure and classifier.

https://doi.org/10.1371/journal.pone.0322887.g016

One limitation of the study conducted here is that it was infeasible to keep the true OGMs constant for different values of p and at the same time have reasonable true classification performances of the classifiers. Therefore, the true OGMs depend on p and the effects due to the OGM and due to the dimension cannot be separated. Nonetheless, the observed performance decrease of simulations, especially for Plasmode, is in line with previous results of a study for lower numbers of variables and the estimation of the MSE of the least squares estimator in linear regression [7]. In that study, the true models were kept constant across different values of p. Therefore, it seems reasonable that this effect can mainly be attributed to the value of p rather than the subtle differences in the OGMs. In contrast, the shift of variables had almost no effect for $p = 2$ and $p = 10$, but clear effects for larger p. This could be explained by the concrete coefficients of the models for the lower ps and therefore can probably be attributed to the differences in the models rather than to the dimension of the data. Overall, the scalability of

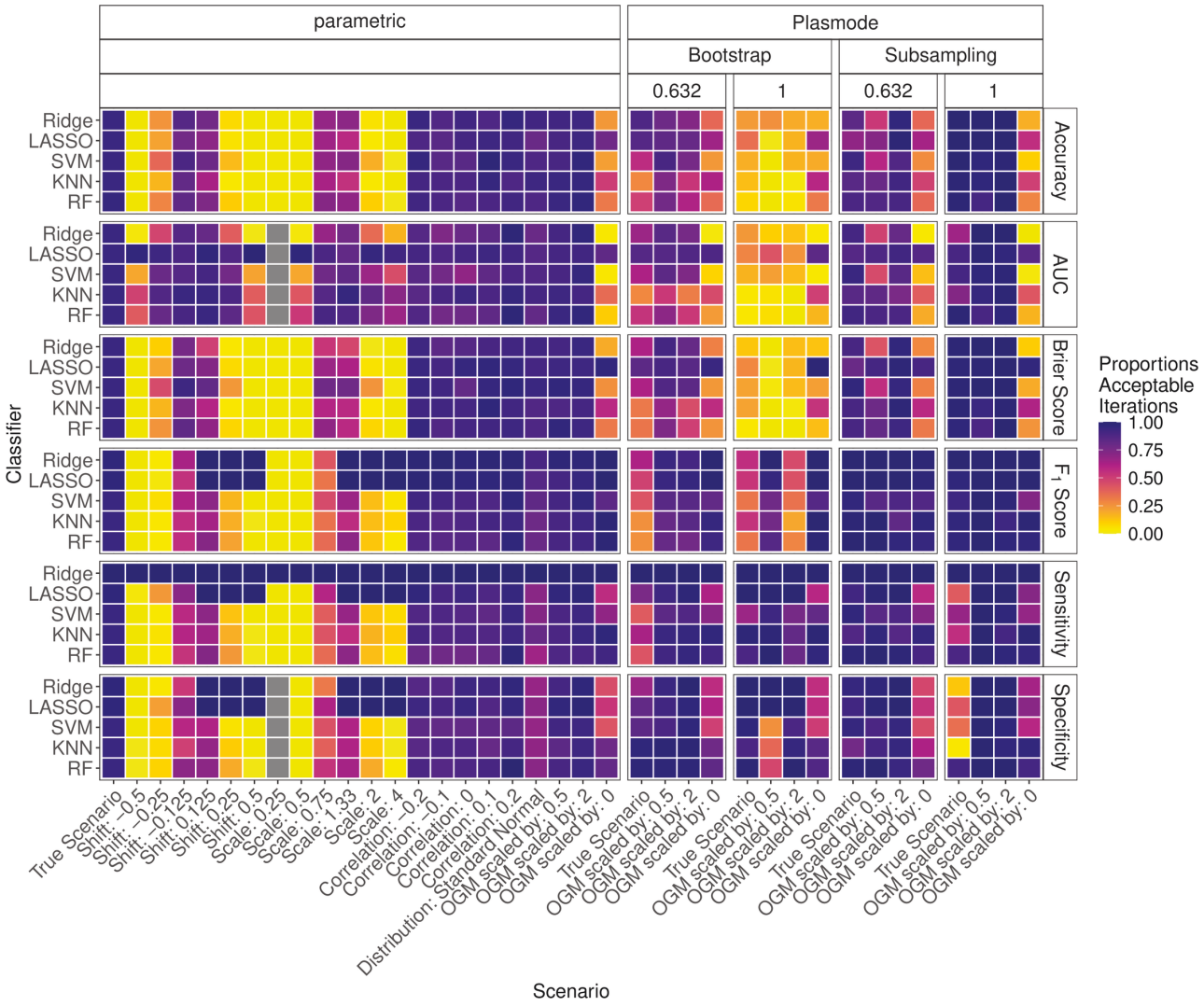


Fig 17. Proportion of acceptable iterations for 100 iterations of parametric and Plasmode method comparison studies under different scenarios for different classifiers and classification performance measures for $p = 150$. An iteration is defined as acceptable if its relative errors for the respective measure lie within the 2.5% and 97.5% quantile of the parametric simulation errors for the true scenario for that measure and classifier.

https://doi.org/10.1371/journal.pone.0322887.g017

Plasmode simulations seems questionable since we observed increasing errors for increasing numbers of variables in both studies. More research regarding this aspect is needed. On the other hand, in practical applications, the use of parametric simulation for high-dimensional data is hard as the number of marginal distributions and especially the number of pairwise correlations to be specified increases with the number of variables. Thus, it is reasonable to assume that the problem of over-simplification and therefore misspecifications of the DGP in parametric simulation also increases with the number of variables. The chosen numbers of variables p in this study represent the range of common numbers of variables of real-world datasets of low to moderately high dimensions. They are however not representative of ultra-high-dimensional data. Higher numbers of variables were infeasible to use within the simulation study due to runtime.

Another limitation of this study is that the number of samples and the true OGM and DGP were not varied which restricts the scope of this study. This limited number of scenarios, the comparably low number of iterations per scenario, and the exclusion of some classifiers are due to the high runtime and limited computing capacity. The effect of changing the number of samples and the true OGM and DGP are left open for further research.

Supporting information

S1 Appendix. Additional information on simulation setup. Detailed description of the true DGP, discussion of the use of fitted models as true models, coefficients for true OGM, and predicted probabilities for true OGMs.
(PDF)

S2 Appendix. Additional Result Figures and Tables. Tables containing numbers of error and warning messages, additional figures for errors in performance estimation, and Kendall distances of true and simulated method rankings.
(PDF)

S3 File. Marginal distributions of true DGP as CSV-file.
(CSV)

S4 File. Correlation matrix of true DGP as CSV-file.
(CSV)

Acknowledgments

We would like to thank the members of Topic Group (TG) 9 and the Publications Panel of the STRengthening Analytical Thinking for Observational Studies (STRATOS) initiative for their helpful comments.
At the time of submission, STRATOS TG9 consisted of the following members (in alphabetical order): Axel Benner (DKFZ Heidelberg, Germany), Harald Binder (University of Freiburg, Germany), Anne-Laure Boulesteix (Ludwig-Maximilians-University München, Germany), Kevin Dobbin (Augusta University, USA), Roman Hornung (Ludwig-Maximilians-University München, Germany), Lara Lusa (University of Primorska, University of Ljubljana, Slovenia), Stefan Michiels (Univ. Paris Saclay, France), Eugenia Migliavacca (Nestlé Research Center, Switzerland), Jörg Rahnenführer (TU Dortmund University, Germany), and Willi Sauerbrei (University of Freiburg, Germany). The group is co-chaired by Federico Ambrogi (University of Milan, Italy), Riccardo De Bin (University of Oslo, Norway), and Lisa McShane (National Cancer Institute, Bethesda, USA, mcshaneL@ctep.nci.nih.gov).

Author contributions

Conceptualization: Marieke Stolte, Nicholas Schreck, Alla Slynko, Maral Saadati, Axel Benner, Jörg Rahnenführer, Andrea Bommert.

Formal analysis: Marieke Stolte.

Project administration: Jörg Rahnenführer, Andrea Bommert.

Resources: Jörg Rahnenführer.

Software: Marieke Stolte.

Supervision: Jörg Rahnenführer, Andrea Bommert.

Visualization: Marieke Stolte.

Writing – original draft: Marieke Stolte.

Writing – review & editing: Nicholas Schreck, Alla Slynko, Maral Saadati, Axel Benner, Jörg Rahnenführer, Andrea Bommert.

References

1. Boulesteix A-L, Lauer S, Eugster MJA. A plea for neutral comparison studies in computational sciences. PLoS One. 2013;8(4):e61562. https://doi.org/10.1371/journal.pone.0061562 PMID: 23637855
2. Boulesteix A-L, Binder H, Abrahamowicz M, Sauerbrei W, Simulation Panel of the STRATOS Initiative. On the necessity and design of studies comparing statistical methods. Biom J. 2018;60(1):216–8. https://doi.org/10.1002/bimj.201700129 PMID: 29193206
3. Morris TP, White IR, Crowther MJ. Using simulation studies to evaluate statistical methods. Stat Med. 2019;38(11):2074–102. https://doi.org/10.1002/sim.8086 PMID: 30652356
4. Boulesteix A-L, Groenwold RH, Abrahamowicz M, Binder H, Briel M, Hornung R, et al. Introduction to statistical simulations in health research. BMJ Open. 2020;10(12):e039921. https://doi.org/10.1136/bmjopen-2020-039921 PMID: 33318113
5. Schreck N, Slynko A, Saadati M, Benner A. Statistical plasmode simulations—potentials, challenges and recommendations. Stat Med. 2024;43(9):1804–25. https://doi.org/10.1002/sim.10012 PMID: 38356231
6. Cattell R, Jaspers J. A general plasmode (no. 30-10-5-2) for factor analytic exercises and research. Society of Multivariate Experimental Psychology; 1967.
7. Stolte M, Schreck N, Slynko A, Saadati M, Benner A, Rahnenführer J, et al. Simulation study to evaluate when plasmode simulation is superior to parametric simulation in estimating the mean squared error of the least squares estimator in linear regression. PLoS One. 2024;19(5):e0299989. https://doi.org/10.1371/journal.pone.0299989 PMID: 38748677
8. Hafermann L, Klein N, Rauch G, Kammer M, Heinze G. Using background knowledge from preceding studies for building a random forest prediction model: a plasmode simulation study. Entropy (Basel). 2022;24(6):847. https://doi.org/10.3390/e24060847 PMID: 35741566
9. Efron B. Bootstrap methods: another look at the jackknife. Ann Statist. 1979;7(1):1–26. https://doi.org/10.1214/aos/1176344552
10. Götze F. Asymptotic approximation and the bootstrap. IMS Bull. 1993;305.
11. Bickel P, Götze F, van Zwet W. Resampling fewer than n observations: gains, losses, and remedies for losses. Stat Sin. 1997;7(1):1–31.
12. Politis DN, Romano JP, Wolf M. Subsampling. Springer Science & Business Media. 1999.
13. Schaefer RL, Roi LD, Wolfe RA. A ridge logistic estimator. Commun Stat Theory Methods. 1984;13(1):99–113. https://doi.org/10.1080/03610928408828664
14. Tibshirani R. Regression shrinkage and selection via the lasso. J R Stat Soc Series B Stat Methodol. 1996;58(1):267–88. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x
15. Vapnik V, Chervonenkis A. Theory of pattern recognition. Nauka; 1974.
16. Fix E, Hodges J. Discriminatory analysis: nonparametric discrimination, consistency properties. USAF School of Aviation Medicine; 1951.
17. Cover T, Hart P. Nearest neighbor pattern classification. IEEE Trans Inform Theory. 1967;13(1):21–7. https://doi.org/10.1109/tit.1967.1053964
18. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32. https://doi.org/10.1023/a:1010933404324
19. Fan J, Fan Y. High dimensional classification using features annealed independence rules. Ann Stat. 2008;36(6):2605–37. https://doi.org/10.1214/07-AOS504 PMID: 19169416
20. Chen L-P. Classification and prediction for multi-cancer data with ultrahigh-dimensional gene expressions. PLoS One. 2022;17(9):e0274440. https://doi.org/10.1371/journal.pone.0274440 PMID: 36107929
21. Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S. Hyperparameter optimization: foundations, algorithms, best practices and open challenges. arXiv, preprint, 2021.
22. Olson DL, Delen D. Advanced data mining techniques. Berlin, Heidelberg: Springer; 2008.
23. Steyerberg EW. Clinical Prediction Models. Springer International Publishing; 2019. https://doi.org/10.1007/978-3-030-16399-0

24. Simon R. Resampling strategies for model assessment and selection. In: Dubitzky W, Granzow M, Berrar D. (eds.) Fundamentals of data mining in genomics and proteomics. Boston, MA: Springer; 2007. https://doi.org/10.1007/978-0-387-47509-7_8
25. Gneiting T, Raftery AE. Strictly proper scoring rules, prediction, and estimation. J Am Statist Assoc. 2007;102(477):359–78. https://doi.org/10.1198/016214506000001437
26. Bischl B, Sonabend R, Kotthoff L, Lang M, editors. Applied machine learning using mlr3 in R. New York: Chapman and Hall/CRC; 2024.
27. Moran PAP, Kendall MG. Rank correlation methods. Int Stat Rev. 1973;41(3):399. https://doi.org/10.2307/1402637
28. Diaconis P. Group representations in probability and statistics. Lecture Notes-Monograph Series 11. Hayward, CA: Institute of Mathematical Statistics; 1988, pp. i–192.
29. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev. 2017;59(1):65–98. https://doi.org/10.1137/141000671
30. Knudson A, Schissler G. Bigsimr.jl: simulate multivariate distributions with arbitrary marginals. https://github.com/SchisslerGroup/Bigsimr.jl. 2024.
31. Lin D, White J, Byrne S, Bates D, Noack A, Pearson J. JuliaStats/Distributions.jl: a Julia package for probability distributions and associated functions. 2019.
32. Besançon M, Papamarkou T, Anthoff D, Arslan A, Byrne S, Lin D, et al. Distributions.jl: definition and modeling of probability distributions in the JuliaStats ecosystem. J Stat Soft. 2021;98(16). https://doi.org/10.18637/jss.v098.i16
33. R Core Team. R: A language and environment for statistical computing; 2024. Available from: https://www.R-project.org/.
34. Knudson A, Schissler G. Bigsimr: fast generation of high-dimensional random vectors. 2024. https://CRAN.R-project.org/package=bigsimr
35. Lang M, Binder M, Richter J, Schratz P, Pfisterer F, Coors S, et al. mlr3: a modern object-oriented machine learning framework in R. J Open Source Softw. 2019;4(44):1903. https://doi.org/10.21105/joss.01903
36. Lang M. mlr3measures: Performance Measures for 'mlr3'. 2022. Available from: https://CRAN.R-project.org/package=mlr3measures
37. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw. 2010;33(1):1–22. https://doi.org/10.18637/jss.v033.i01 PMID: 20808728
38. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F. e1071: misc functions of the department of statistics, probability theory group (formerly: E1071). TU Wien; 2021. Available from: https://CRAN.R-project.org/package=e1071.
39. Schliep K, Hechenbichler K. Kknn: weighted k-nearest neighbors. 2016. Available from: https://CRAN.R-project.org/package=kknn
40. Wright MN, Ziegler A. ranger: a fast implementation of random forests for high dimensional data in C++ and R. J Stat Soft. 2017;77(1). https://doi.org/10.18637/jss.v077.i01
41. Jacques J, Grimonprez Q, Biernacki C. Rankcluster: an R package for clustering multivariate partial rankings. R J. 2014;6(1):101. https://doi.org/10.32614/rj-2014-010
42. Wickham H. ggplot2: Elegant Graphics for Data Analysis. New York: Springer; 2016. Available from: https://ggplot2.tidyverse.org.
43. van den Brand T. Ggh4x: hacks for 'ggplot2'. 2024. https://CRAN.R-project.org/package=ggh4x
44. Jeppson H, Hofmann H, Cook D. ggmosaic: mosaic plots in the 'ggplot2' framework. 2021. Available from: https://CRAN.R-project.org/package=ggmosaic.
45. Lang M, Bischl B, Surmann D. batchtools: tools for R to work on batch systems. J Open Source Softw. 2017;2(10):135. https://doi.org/10.21105/joss.00135

3. Article 3: Methods for Quantifying Dataset Similarity: A Review, Taxonomy and Comparison

Statistics Surveys
Vol. 18 (2024) 163–298
ISSN: 1935-7516
https://doi.org/10.1214/24-SS149

Methods for quantifying dataset similarity: a review, taxonomy and comparison

Marieke Stolte[1], Franziska Kappenberg[1], Jörg Rahnenführer[1] and Andrea Bommert[1]

[1] *Department of Statistics, TU Dortmund University,*
e-mail: stolte@statisitk.tu-dortmund.de; kappenberg@statistik.tu-dortmund.de; rahnenfuehrer@statistik.tu-dortmund.de; bommert@statistik.tu-dortmund.de

Abstract: Quantifying the similarity between datasets has widespread applications in statistics and machine learning. The performance of a predictive model on novel datasets, referred to as generalizability, depends on how similar the training and evaluation datasets are. Exploiting or transferring insights between similar datasets is a key aspect of meta-learning and transfer-learning. In simulation studies, the similarity between distributions of simulated datasets and real datasets, for which the performance of methods is assessed, is crucial. In two- or k-sample testing, it is checked, whether the underlying distributions of two or more datasets coincide.

Extremely many approaches for quantifying dataset similarity have been proposed in the literature. We examine more than 100 methods and provide a taxonomy, classifying them into ten classes. In an extensive review of these methods the main underlying ideas, formal definitions, and important properties are introduced.

We compare the 118 methods in terms of their applicability, interpretability, and theoretical properties, in order to provide recommendations for selecting an appropriate dataset similarity measure based on the specific goal of the dataset comparison and on the properties of the datasets at hand. An online tool facilitates the choice of the appropriate dataset similarity measure.

MSC2020 subject classifications: 62E99, 62G10, 62H15, 62H30, 05C90.
Keywords and phrases: Dataset similarity, similarity measure, distance measure, graph-based distance, probability metric, divergence, inter-point distance, energy statistic, kernel mean embedding, maximum mean discrepancy, binary classification, optimal transport, meta-learning, two-sample test, permutation test, theoretical properties.

Received February 2024.

Contents

arXiv: 2312.04078

1. Introduction

Quantifying how similar or different two or more datasets are is a crucial subtask in various applications of statistics and machine learning. Examples of applications include but are not limited to (i) the assessment of the generalizability of a predictive model to a broader context, (ii) the transfer of knowledge from one task to another task in transfer-learning or meta-learning, (iii) the comparison of distributions of simulated data and data from the true data-generating process when planning and implementing simulation studies, and (iv) checking whether the underlying distributions of two or more datasets coincide via two- or k-sample testing.

In statistics and machine learning, generalizability is a measure of a model's performance for a broader context, compared to the data on which it was fitted. Generalizability is a useful property since conclusions drawn from the present study can be transferred to a more general set of study objects. The performance of the model on a new or unseen dataset depends on the similarity between the dataset that was used for fitting the model and the new dataset. Quantifying this similarity with a univariate measure thus can help to assess whether generalizability is given without fitting the model on the new dataset.

In meta-learning and transfer-learning, a central component is to exploit or transfer insights between different datasets. For example, some meta-learning models try to find the most suitable datasets to train specific models. Likewise, in transfer-learning, a common approach is to pre-train a model on a large (source) dataset and then fine-tune the model on the (target) dataset of interest. Also in this situation, it is crucial to understand and measure similarities between datasets in order to select appropriate source datasets for the first step of the process.

Further, when transferring insights of simulation studies to given data, the similarity between distributions of simulated datasets and the distribution of a (target) dataset, for which the performance of methods is assessed, is critical. If assumptions are made about the underlying distribution, such as that it is a normal distribution, and if these assumptions are not met, the conclusions from the simulation study for the target dataset may be fundamentally flawed.

In the statistical learning and machine learning literature, a vast number of approaches for quantifying dataset similarity have been proposed. However, to the best of our knowledge, there is no comprehensive comparison between many of these different approaches. In the following, we refer to some publications in which at least comparisons of subsets of the methods have been carried out.

In Thas (2010), many methods for comparing distributions for univariate data, including graphical methods as well as hypothesis tests, are explained and discussed, but the multivariate case is not covered. In Rachev (1991) and in Liese and Vajda (1987), properties of several probability metrics and divergences, respectively, are discussed, i.e. distance measures between probability

distributions.

In general, many articles that present new methods for measuring dataset similarity include brief summaries of competing methods (e.g. Rosenbaum, 2005; Biswas and Ghosh, 2014; Chen and Friedman, 2017; Sarkar, Biswas and Ghosh, 2020; Deb and Sen, 2021; Kim et al., 2021; Li, Hu and Zhang, 2022; Huang and Sen, 2023). Most of these include only a small number of competing methods and, in most cases, only methods based on the same principle. Further, some articles provide comprehensive reviews of single methodological classes (e.g. Muandet et al. (2017) for kernel mean embeddings or Székely and Rizzo (2017) for the energy distance).

Simulation studies comparing the new method with some previous methods are often presented additionally (e.g. Biswas and Ghosh, 2014; Chwialkowski et al., 2015; Mondal, Biswas and Ghosh, 2015; Jitkrittum et al., 2016; Petrie, 2016; Chen and Friedman, 2017; Lopez-Paz and Oquab, 2017; Liu, Li and Póczos, 2018; Liu et al., 2020; Sarkar, Biswas and Ghosh, 2020).

We do not know of any comparison of methods belonging to many of the different approaches, in particular methods based on different principles. In this paper, we give an extensive review providing characterization and classification of dataset similarity methods and their properties. In total, we examine more than 100 methods for quantifying dataset similarity and provide a taxonomy dividing the methods into ten classes, based on the underlying principles we identified. The methods were selected from an extensive literature search by using the following criteria:

- *The method is applicable for multivariate data.* This excludes the vast literature on methods for comparing univariate distributions. For example, a comprehensive overview of methods for one-dimensional data can be found in Thas (2010).
- *The method requires no specific parametric or distributional assumptions on the underlying distributions of the datasets* (e.g. normal distribution). The general assumptions of discrete or continuous data are allowed since they can be easily verified in practice.
- *The method does not focus on a particular property of the data* (e.g. means), but on the entire dataset or its entire distribution. This particularly excludes tests based solely on location or scale differences.

The classes into which the methods are divided are (i) comparison of cumulative distribution functions, density functions, or characteristic functions; (ii) methods based on multivariate ranks; (iii) discrepancy measures for distributions; (iv) graph-based methods; (v) methods based on inter-point distances; (vi) kernel-based methods; (vii) methods based on binary classification; (viii) distance and similarity measures for datasets; (ix) comparison based on summary statistics; and (x) testing approaches. The division is based on the fundamental underlying ideas that we identified in the set of analyzed dataset similarity methods. This taxonomy is not strict, but helpful for structuring the set of methods. Some methods could be classified into several classes. In those cases, we put them into the class matching their main idea.

Moreover, we present a comprehensive comparison of the methods. We introduce 22 criteria to judge the applicability, interpretability, and theoretical properties of dataset similarity measures. For each method, we check which of these criteria are fulfilled to provide guidance for the choice of a suitable method for quantifying the similarity of given datasets. To further facilitate the comparison of methods we implement an online tool that allows for interactive filtering and sorting of the methods (https://shiny.statistik.tu-dortmund.de/data-similarity).

In Section 2, we present the notation and assumptions that are used throughout the article. In Section 3, a detailed description of all methods can be found. The methods of each class are presented in a separate subsection. Within each class, methods are ordered chronologically. An overview of all methods is given in Section 3.1. In Section 4, we present a summary of the methods in the ten classes. For each group of methods, we describe its general concept and explain some prototypical example methods. This summary points out the main ideas of each class and can be understood without reading the detailed method descriptions. In Section 5, we introduce the list of criteria for rating the dataset similarity measures. These criteria are organized according to three main categories: applicability, interpretability, and theoretical properties. In Section 6, we provide the results of the method comparison based on the criteria presented before. In Section 7, a brief summary of the review and comparison and an outlook are given.

2. Notation and general assumptions

In general, a dataset can be viewed in two different ways, which impacts the approach to measure similarity. First, a dataset can be viewed simply as a collection of points in space. Second, and this is the more common view in the methods we examined, a dataset can be viewed as a sample of random variables that follow a true underlying distribution. In the second case, it is often of interest to estimate the similarity of these underlying distributions rather than the similarity of the datasets themselves. Therefore, many of the methods presented below focus on comparing multivariate distributions rather than directly comparing datasets.

In the following, we assume at least two different datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ consisting of n_1 and n_2, respectively, samples $X_1, \ldots, X_{n_1} \sim F_1$ and $Y_1, \ldots, Y_{n_2} \sim F_2$. We assume $X_i, Y_j \in \mathbb{R}^p \, \forall i \in \{1, \ldots, n_1\}, j \in \{1, \ldots, n_2\}$ and call the p components of each sample *features*. We denote the pooled sample as $\{Z_1, \ldots, Z_N\} = \{X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}\}$, where $N = n_1 + n_2$ is the total sample size. For most of the methods, we assume that all Z_i are distributed independently. For asymptotics, it is assumed that $N \to \infty$ such that $\frac{n_1}{N} \to const \in (0, 1)$ if not explicitly stated otherwise. We define the two-sample problem as the testing problem

$$H_0 : F_1 = F_2 \text{ vs. } H_1 : F_1 \neq F_2. \tag{1}$$

This testing problem is sometimes also called testing for homogeneity of the two distributions.

In some cases, we also assume that there are n_i observations of a target variable in each dataset, but most methods only require the feature variables and cannot deal with a target variable in a meaningful way. Analogously to the two-sample problem, we define the k-sample problem for $k \geqslant 2$ datasets $\mathcal{D}_1, \ldots, \mathcal{D}_k$ with sample sizes $n_i, i = 1, \ldots, k$ as

$$H_0 : F_1 = F_2 = \cdots = F_k \text{ vs. } H_1 : \exists i \neq j \in \{1, \ldots, k\} : F_i \neq F_j,$$

where F_i denotes the distribution of each sample in the ith dataset. We use the notation F_i to denote the distribution as well as its cumulative distribution function. By f_i we denote the corresponding density functions if they exist. If not explicitly stated otherwise we refer to the special case of the two-sample problem (1).

In general, we denote random variables in uppercase letters and the corresponding observations in lowercase letters. We use the hat symbol to denote estimators. $\hat{F}_i$ and $\hat{f}_i$ denote the empirical distribution and density functions, respectively. We use T as the symbol for (test) statistics and $d(\cdot, \cdot)$ to denote distance measures.

3. Detailed description of data similarity methods

In the following, we describe all selected dataset similarity methods. The methods are sorted according to the classes we identified. The first subsection gives an overview of all methods. In Subsections 3.2 to 3.13, the methods belonging to each class are described.

3.1. Overview of all methods

Table 1 gives an overview of the dataset similarity methods included in this review. The first column gives the name of the method, if available, and otherwise, the reference where the method is defined. In all cases, the reference is linked. If the article defining the method is published online, the online publication can be accessed by clicking on the link that is given in parentheses after the method name. The classes to which the methods are assigned are given in subheadings within the body of the table. The subclasses are listed in the second column. The third column shows the section and page where the method is described within this article.

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Comparison of CDFs, density or characteristic functions		
Bickel (1969) (link)	Comparison of CDFs	3.2, p. 174

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Biau and Gyorfi (2005) (link)	Comparison of CDFs	3.2, p. 175
Boeckel, Spokoiny and Suvorikova (2018) (link)	Comparison of CDFs	3.2, p. 175
Ntoutsi, Kalousis and Theodoridis (2008) (link)	Comparison of density functions	3.3.1, p. 179
Ganti et al. (1999) (link)	Comparison of density functions	3.3.1, p. 176
Roederer et al. (2001) (link)	Comparison of density functions	3.3.1, p. 178
Wang and Pei (2005) (link)	Comparison of density functions	3.3.1, p. 178
Ahmad and Cerrito (1993) (link)	Comparison of density functions	3.3.2, p. 180
Anderson, Hall and Titterington (1994) (link)	Comparison of density functions	3.3.2, p. 180
Cao and van Keilegom (2006) (link)	Comparison of density functions	3.3.2, p. 180
Alba-Fernández, Ibáñez-Pérez and Jiménez-Gamero (2004) (link)	Comparison of characteristic functions	3.4, p. 181
Alba Fernández, Jiménez Gamero and Muñoz García (2008) (link)	Comparison of characteristic functions	3.4, p. 181
Liu, Xia and Zhou (2015) (link)	Comparison of characteristic functions	3.4, p. 181
Li, Hu and Zhang (2022) (link)	Comparison of characteristic functions	3.4, p. 182
Rank-based methods		
Ghosh and Biswas (2016) (link)	Rank-based	3.5, p. 183
Ghosal and Sen (2021) (link)	Rank-based	3.5, p. 184
Deb, Bhattacharya and Sen (2021) (link)	Rank-based	3.5, p. 184
Zhou and Chen (2023) (link)	Rank-based	3.5, p. 185
Discrepancy measure for distributions		
Engineer metric (Rachev, 1991)	Probability metric	3.6.1, p. 187
Zolotarev's semimetric (Rachev, 1991)	Probability metric	3.6.1, p. 187
Ky Fan metric (Rachev, 1991)	Probability metric	3.6.1, p. 187

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Prokhorov metric (Rachev, 1991)	Probability metric	3.6.1, p. 187
Dudley metric	Probability metric	3.6.1, p. 187
Total variation metric (Zolotarev, 1984)	Probability metric	3.6.1, p. 187
Kantorovich-Rubinstein metric (Zolotarev, 1984; Dudley, 1989) (link)	Probability metric	3.6.1, p. 187
L^q metrics	Probability metric	3.6.1, p. 187
Wasserstein metrics	Probability metric	3.11, p. 245
(Squared) Hellinger distance	Divergence	3.6.2, p. 193
Vincze Le Cam distance (Vincze, 1981; Le Cam, 1986)	Divergence	3.6.2, p. 193
KL divergence (Kullback and Leibler, 1951) (link)	Divergence	3.6.2, p. 193
Jeffrey's divergence	Divergence	3.6.2, p. 193
Extended ϕ_α divergence	Divergence	3.6.2, p. 193
Jensen Shannon divergence	Divergence	3.6.2, p. 193
Pearson divergence (Pearson, 1900) (link)	Divergence	3.6.2, p. 193
Relative Pearson divergence (Yamada et al., 2013) (link)	Divergence	3.6.2, p. 193
f-dissimilarity (Györfi and Nemetz, 1975; García-García and Williamson, 2012) (link)	Divergence	3.6.2, p. 193
Rényi divergence (Rényi, 1961) (link)	Divergence	3.6.2, p. 199
Relative information of type s (Taneja and Kumar, 2004) (link)	Divergence	3.6.2, p. 200
H-divergence (Zhao et al., 2021) (link)	Divergence	3.6.2, p. 200
Muñoz et al. (2012) (link)	Divergence	3.6.2, p. 202
Muñoz, Martos and González (2013) (link)	Divergence	3.6.2, p. 203
Graph-based methods		
Friedman and Rafsky (1979) (link)	Graph-based	3.7.1, p. 204
Rosenbaum (2005) (link)	Graph-based	3.7.1, p. 205
Chen and Zhang (2013) (link)	Graph-based	3.7.1, p. 206

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Biswas, Mukhopadhyay and Ghosh (2014) (link)	Graph-based	3.7.1, p. 206
Petrie (2016) (link)	Graph-based	3.7.1, p. 207
Chen and Friedman (2017) (link)	Graph-based	3.7.1, p. 207
Chen, Chen and Su (2018) (link)	Graph-based	3.7.1, p. 208
Zhang and Chen (2019) (link)	Graph-based	3.7.1, p. 208
Sarkar, Biswas and Ghosh (2020) (link)	Graph-based	3.7.1, p. 208
Mukhopadhyay and Wang (2020a) (link)	Graph-based	3.7.1, p. 209
Mukherjee et al. (2022) (link)	Graph-based	3.7.1, p. 209
Weiss (1960) (link)	Nearest Neighbor	3.7.2, p. 211
Friedman and Steppel (1973)	Nearest Neighbor	3.7.2, p. 211
Schilling (1986); Henze (1988) (link)	Nearest Neighbor	3.7.2, p. 212
Barakat, Quade and Salama (1996) (link)	Nearest Neighbor	3.7.2, p. 213
Nettleton and Banerjee (2001) (link)	Nearest Neighbor	3.7.2, p. 213
Hall and Tajvidi (2002) (link)	Nearest Neighbor	3.7.2, p. 213
Chen, Dou and Qiao (2013) (link)	Nearest Neighbor	3.7.2, p. 214
Mondal, Biswas and Ghosh (2015) (link)	Nearest Neighbor	3.7.2, p. 214
Comparison based on inter-point distances		
Energy statistic (Zech and Aslan, 2003) (link)	Comparison based on inter-point distances	3.8.1, p. 215
Generalized energy statistic (Sejdinovic et al., 2013) (link)	Comparison based on inter-point distances	3.8.1, p. 215
Chakraborty and Zhang (2021) (link)	Comparison based on inter-point distances	3.8.1, p. 215
DISCO (Rizzo and Székely, 2010) (link)	Comparison based on inter-point distances	3.8.1, p. 215
Huang and Huo (2017) (link)	Comparison based on inter-point distances	3.8.1, p. 215
Deb and Sen (2021) (link)	Comparison based on inter-point distances	3.8.1, p. 215

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Al-Labadi, Asl and Saberi (2022) (link)	Comparison based on inter-point distances	3.8.1, p. 215
Baringhaus and Franz (2010) (link)	Comparison based on inter-point distances	3.8.2, p. 219
Liu and Modarres (2011) (link)	Comparison based on inter-point distances	3.8.2, p. 220
Biswas and Ghosh (2014) (link)	Comparison based on inter-point distances	3.8.2, p. 220
Sarkar and Ghosh (2018) (link)	Comparison based on inter-point distances	3.8.2, p. 220
Montero-Manso and Vilar (2019) (link)	Comparison based on inter-point distances	3.8.2, p. 221
Tsukada (2019) (link)	Comparison based on inter-point distances	3.8.2, p. 221
Kernel-based methods		
(Linear) MMD^2 (Gretton et al., 2009; Muandet et al., 2017; Gretton et al., 2012a) (link)	Maximum Mean Discrepancy	3.9.1, p. 222
Block MMD (Zaremba, Gretton and Blaschko, 2013) (link)	Maximum Mean Discrepancy	3.9.1, p. 225
fastMMD (Zhao and Meng, 2015) (link)	Maximum Mean Discrepancy	3.9.1, p. 225
ME (Chwialkowski et al., 2015; Jitkrittum et al., 2016) (link)	Maximum Mean Discrepancy	3.9.1, p. 226
SCF (Chwialkowski et al., 2015; Jitkrittum et al., 2016) (link)	Maximum Mean Discrepancy	3.9.1, p. 226
Regularized MMD (Danafar et al., 2014) (link)	Maximum Mean Discrepancy	3.9.1, p. 228
Anisotropic kernel MMD (Cheng, Cloninger and Coifman, 2020) (link)	Maximum Mean Discrepancy	3.9.1, p. 228
DMMD/ DFDA (Kirchler et al., 2020) (link)	Maximum Mean Discrepancy	3.9.1, p. 228
GPK (Song and Chen, 2023) (link)	Maximum Mean Discrepancy	3.9.1, p. 229
Kernel FDA (Moulines, Bach and Harchaoui, 2007) (link)	Kernel-based	3.9.2, p. 232
Fromont et al. (2012) (link)	Kernel-based	3.9.2, p. 232

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Scetbon and Varoquaux (2019) (link)	Kernel-based	3.9.2, p. 232
Kernel-based quadratic distance (Chen and Markatou, 2020) (link)	Kernel-based	3.9.2, p. 233
Bayesian kernel test (Zhang et al., 2022) (link)	Kernel-based	3.9.2, p. 234
Kernel Measure of Multi-Sample Dissimilarity (Huang and Sen, 2023) (link)	Kernel-based	3.9.2, p. 234
Methods based on binary classification		
Friedman (2004)	Method based on binary classification	3.10, p. 235
C2ST Lopez-Paz and Oquab (2017) (link)	Method based on binary classification	3.10, p. 236
Regression based test (Kim, Lee and Lei, 2019) (link)	Method based on binary classification	3.10, p. 237
Cheng and Cloninger (2022) (link)	Method based on binary classification	3.10, p. 238
Yu et al. (2007) (link)	Method based on binary classification	3.10, p. 239
DiProPerm test (Wei et al., 2016) (link)	Method based on binary classification	3.10, p. 239
Classifier Probability Test (Cai, Goggin and Jiang, 2020) (link)	Method based on binary classification	3.10, p. 240
Kim et al. (2021) (link)	Method based on binary classification	3.10, p. 240
Hediger, Michel and Näf (2022) (link)	Method based on binary classification	3.10, p. 241
Distance/ similarity measure for datasets		
Feurer, Springenberg and Hutter (2015) (link)	Distance measure for datasets	3.11, p. 241
Gromov-Hausdoff distance (Mémoli, 2017) (link)	Distance measure for datasets	3.11, p. 242
Leite, Brazdil and Vanschoren (2012); Leite and Brazdil (2021) (link)	Similarity measure for datasets	3.11, p. 243
DeDiMs (Calderon Ramirez et al., 2022) (link)	Distance measure for datasets	3.11, p. 244

Table 1: List of all dataset similarity methods

Method/ Article	Subclass	Section, Page
Alvarez-Melis and Fusi (2020) (link)	Distance measure for datasets	3.11, p. 245
Comparison based on summary statistics		
DataSpheres (Johnson and Dasu, 1998) (link)	Comparison based on summary statistics	3.12, p. 247
Constrained minimum distance (Tatti, 2007)	Comparison based on summary statistics	3.12, p. 248
Testing approaches		
Romano (1989) (link)	Testing approach	3.13, p. 250
Burke (2000) (link)	Testing approach	3.13, p. 250
Ping (2000) (link)	Testing approach	3.13, p. 250
Chen and Hanson (2014) (link)	Testing approach	3.13, p. 251
Zhou, Zheng and Zhang (2017) (link)	Testing approach	3.13, p. 251
Pan et al. (2018) (link)	Testing approach	3.13, p. 252
Wan, Liu and Deng (2018) (link)	Testing approach	3.13, p. 253
Kim, Balakrishnan and Wasserman (2020) (link)	Testing approach	3.13, p. 253
Li and Zhang (2020) (link)	Testing approach	3.13, p. 255
Liu et al. (2022) (link)	Testing approach	3.13, p. 255
Paul, De and Ghosh (2022a) (link)	Testing approach	3.13, p. 255

3.2. Comparison of cumulative distribution functions

In this subsection, methods based on the comparison of cumulative distribution functions (cdf) will be presented. Comparing distributions by their cumulative distribution functions is an intuitive approach since a distribution is fully characterized by its cumulative distribution function. In the one-dimensional case, methods of the Kolmogorov-Smirnov (KS) type that compare the maximal absolute difference of the (empirical) cumulative distribution functions are particularly popular. Still, their extension to the multivariate case is not straightforward (Ramdas, Trillos and Cuturi, 2017).

Extension of the Kolmogorov-Smirnov test via permutation Bickel (1969) gives a generalization of the Kolmogorov-Smirnov test to multivariate data based on applying a permutation procedure to the classical Kolmogorov-Smirnov test. It is distribution-free for continuous distributions and consistent

against all alternatives. Bickel (1969) states that the asymptotic value of the cut-off point depends on the distribution F if F is not continuous. No details are given on the practical implementation of the test. Chen and Friedman (2017) claim that the required sample size is exponential in the dimension p and Mondal, Biswas and Ghosh (2015) note that the test cannot be used for $p > n_i$.

Extension of the Kolmogorov-Smirnov test via partitioning Biau and Gyorfi (2005) design a test using the L^1 distance between empirical distributions restricted to a finite partition of the support of the two distributions. For this test, $n_1 = n_2$ is required and a finite partition of $\mathbb{R}^p$ is needed. The authors themselves state that the "choice of the partition in [the test statistic] is a difficult one". Biau and Gyorfi (2005) assume for this partition that for $N \to \infty$ the maximum of measures over each part goes to zero. A rectangle partition is said to be a good choice if cell probabilities are approximately equal. The resulting test is distribution-free and strongly consistent. In addition, an asymptotic version of the test is given, which is not distribution-free but is consistent. According to Gretton et al. (2006) performing the test becomes difficult or impossible for high-dimensional problems due to the partitioning that becomes increasingly difficult in higher dimensions.

Multivariate distribution functions based on measure transportation Boeckel, Spokoiny and Suvorikova (2018) define a new generalization of distribution functions to the multivariate case, called ν-Brenier Distribution Functions (BDF), and their empirical counterparts. The ν-BDF of a distribution is defined as the push-forward (measure-preserving transformation) of a continuous measure to a reference measure ν that has compact, convex support. To be more precise, the push forward of the mixture $t\mu_X + (1-t)\mu_Y, t \in [0,1]$ to a uniform distribution in the unit ball, where t is the asymptotic ratio of sample sizes n_1/N. Boeckel, Spokoiny and Suvorikova (2018) assume that both measures belong to the family of absolutely continuous measures with finite second moments and compact support. They show that an analog of the Glivenko-Cantelli theorem holds for the empirical ν-BDF. For their testing procedure, Boeckel, Spokoiny and Suvorikova (2018) choose ν as the uniform measure on the unit sphere in $\mathbb{R}^p$. The test statistic is the 2-Wasserstein distance (11) between image measures of the distributions of X and Y generated by the push-forward of the mixture distribution of X and Y to ν. In practice, the empirical counterparts are used. The procedure works by generating a uniform partition of the unit ball into N parts, then calculating the optimal transport of both samples to this partition and taking the 2-Wasserstein distance of the empirical distributions of these optimal transports. The critical value is obtained by using the (1-α)-quantile of the empirical distribution of 2-Wasserstein distances of empirical distributions of M random permutations of a uniform partition of the unit ball into N parts. An asymptotic upper bound for the type II error is derived. According to Deb and Sen (2021), this test is one of two tests for the multivariate two-sample problem that is exactly distribution-free, computationally feasible, and consistent against all alternatives. However, they criticize that the test statistic is

random given the data due to external randomization in the construction of the test statistic, that strong assumptions on underlying distributions are needed, such as that the data generating distribution is compactly supported and absolutely continuous, and that there is no asymptotic null distribution theory. The method of Boeckel, Spokoiny and Suvorikova (2018) could also be seen as a method based on multivariate ranks or as a test based on the Wasserstein distance.

3.3. Comparison of density functions

The comparison of density functions follows a similar idea as the comparison of cumulative distribution functions. Different approaches are presented below.

3.3.1. Comparison of probability densities based on partitions

Partitions based on decision trees I Ganti et al. (1999) propose measuring the deviation between datasets based on criteria derived from decision tree models. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ denote two datasets that include a categorical target variable each.

Ganti et al. (1999) calculate a decision tree model for each dataset $\mathcal{D}_1$ and $\mathcal{D}_2$ and calculate the *greatest common refinement (GCR)* induced by these trees. That is the intersection of the partitions of the sample space induced by each tree. They then compare the distribution of both datasets over this GCR. Let n_r denote the number of segments of the GCR, p_i the proportion of observations of $\mathcal{D}_1$ that map to the i-th segment, and q_i the respective proportion of observations of $\mathcal{D}_2$ mapping to the i-th segment. Then Ganti et al. (1999) compare the vector p and q by a difference function $f : \mathbb{R}^{n_r} \to \mathbb{R}^{n_r}$ and aggregate the results from that by an aggregate function $g : \mathbb{R}^{n_r} \to \mathbb{R}$ to obtain a measure of distance between the two datasets

$$\text{GAN1} = g(f(p, q)).$$

Large values then indicate differences between the datasets. They propose the absolute difference function

$$f_a(p, q)_i = |p_i - q_i|,$$

and the scaled difference function

$$f_s(p, q)_i = \begin{cases} \frac{|p_i - q_i|}{(p_i + q_i)/2}, & \text{if } (p_i + q_i) > 0 \\ 0, \text{otherwise} \end{cases}.$$

For the aggregate function, they propose the sum or maximum of the values from the difference function. For using the sum as the aggregate function together with either f_a or f_s, it can be shown that the GCR is optimal in the sense that it gives the lowest value over all common refinements. For using the maximum, this

property is not fulfilled. As in general, for different combinations of difference and aggregate functions, there might not be an upper bound for the difference given by the proposed measure GAN1, Ganti et al. (1999) propose using a Bootstrap test procedure for assessing whether or not the two datasets are generated by the same data-generating process. The lower bound for GAN1 for all proposed difference and aggregate functions is 0.

Typical measures for monitoring change and assessing how much a new tree model differs can be seen as a special case of the measure proposed by Ganti et al. (1999). For these, they calculate a decision tree model for dataset $\mathcal{D}_1$.

Then, for the first measure of change between models, they use this model fit on the first dataset to make predictions $\widehat{Y}$ for the target variable in dataset $\mathcal{D}_2$. Finally, they calculate the misclassification rate

$$\text{GAN2} = \frac{|\{i : \widehat{Y}_i \neq Y_i\}|}{|\mathcal{D}_2|},$$

which is the proportion of observations in $\mathcal{D}_2$ whose target value is predicted incorrectly by the model fitted on $\mathcal{D}_1$.

For the second measure of change between models, they consider the partition of the feature space induced by the decision tree calculated on $\mathcal{D}_1$. Let n_r again denote the number of segments of this partition, p_i the proportion of observations of $\mathcal{D}_1$ that map to the i-th segment, and q_i the respective proportion of observations of $\mathcal{D}_2$ mapping to the i-th segment. If the datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ come from the same data generating process, the number of observations of $\mathcal{D}_2$ that are expected to map to the i-th segment can be estimated by $|\mathcal{D}_2| \cdot p_i$. The number of observations mapping to the i-th segment equals $|\mathcal{D}_2| \cdot q_i$. Let $c \in \mathbb{R}$ denote a small positive constant, for example $c = 0.5$. Ganti et al. (1999) propose calculating the χ^2-statistic

$$\text{GAN3} = \sum_{i=1}^{n_r} \chi(p_i, q_i)$$

with

$$\chi(p_i, q_i) = \begin{cases} \frac{(|\mathcal{D}_2| \cdot p_i - |\mathcal{D}_2| \cdot q_i)^2}{|\mathcal{D}_2| \cdot p_i}, & \text{if } p_i > 0, \\ c, & \text{otherwise.} \end{cases}$$

For both measures, low values indicate similar datasets. With respect to bounds, $0 \leqslant \text{GAN1} \leqslant 1$ holds because GAN1 is a proportion. Also, $0 \leqslant \text{GAN2}$ holds because it is a sum of squared values. A dataset-independent upper bound cannot be specified for GAN2 because χ can attain higher values with more observations in $\mathcal{D}_2$. Again, a Bootstrap test is proposed to assess the significance of the χ^2 value since usual asymptotics for the classical χ^2 test typically will not apply here due to low expected counts.

Ganti et al. (1999) do not address the choice of hyperparameters for creating the decision tree model. As discussed in the previous paragraph, this can have a non-negligible impact on the resulting data similarity measures.

Partitions based on probability binning I Roederer et al. (2001) suggest probability binning for comparing the multivariate distributions of two datasets. Their method only considers the feature space $\mathcal{X}$ and is only applicable to numeric features. First, one dataset is chosen to define a partition of the feature space. To do so, for each feature, the median value and the variance are computed. Two bins (segments) are created by splitting the feature space at the median value of the feature with the largest variance. Then, the calculation of median and variance as well as the splitting is continued recursively for both subspaces, until a predefined minimum number of observations per bin is reached.

Having obtained the partition, the proportions of observations falling into each bin are calculated for both datasets. Let $p_{1,i}$ denote the proportion of observations of the first dataset that fall into the i-th bin, and let $p_{2,i}$ denote the respective proportion for the second dataset. Then, Roederer et al. (2001) propose the measure

$$\text{ROE} = \sum_{i=1}^{n_b} \frac{(p_{1,i} - p_{2,i})^2}{p_{1,i} + p_{2,i}}$$

with n_b denoting the number of bins to quantify the difference between the two datasets. As stated in Roederer et al. (2001), the measure is bounded by $0 \leqslant \text{ROE} \leqslant 2$ with low values corresponding to high similarity.

Roederer et al. (2001) explain that the minimum number of observations per bin should not be smaller than 10. Also, it should be chosen appropriately by the user, such that good coverage of a potentially high-dimensional space can be achieved. With a predefined minimum number of observations per bin, the number of observations in the first dataset determines the number of bins. This might lead to problems when the number of observations is very different for the two datasets.

Partitions based on probability binning II Wang and Pei (2005) propose a measure to quantify changes between two datasets with class labels for the application of fraud and intrusion detection. Their dataset distance measure quantifies concept drifts. It uses a universal model that has minimal learning cost. Wang and Pei (2005) aim to use a quantification of change to improve the current prediction model directly instead of refitting the model after change gets detected. They note that their approach of using arbitrary partition structures instead of learning a tree structure on one of the datasets as a simple alternative serves the same purpose but eliminates the cost of learning the structure. They propose to calculate the class distribution for each dataset for a certain partition of the feature space, which they call the *signature* of the data, by randomly partitioning the multi-dimensional space into a set of bins. This partition can be achieved either by recursive partitioning (random decision tree) or by iteratively creating new bins (random histogram). Wang and Pei (2005) give explicit recommendations on how to create the partition. They recommend the use of random histograms over random decision trees. If $n_{x,j,K}$ and $n_{y,j,K}$ denote the number of observations of the jth part for the Kth class for the first and sec-

ond dataset, respectively, and n_c is the total number of classes and n_r the total number of parts in the partition, the distance function between the signatures of both datasets is defined as

$$\text{Dist}_s(X, Y) = \frac{1}{2} \sum_{j=1}^{n_r} \sum_{K=1}^{n_c} \left| \frac{n_{x,j,K}}{n_1} - \frac{n_{y,j,K}}{n_2} \right| \in [0, 1].$$

The distance can be calculated efficiently and can be used to make predictions. The calculation is repeated to create B random structures resulting in B random signatures. Then, the distance measure as above is calculated for each random structure and the mean over the values is taken as the final distance between the two datasets. The random signatures can be used for classification as well.

Partitions based on decision trees II Ntoutsi, Kalousis and Theodoridis (2008) propose measuring dataset similarity based on probability density estimates derived from decision trees. Consider two classification datasets $\mathcal{D}_1$ and $\mathcal{D}_2$. For each of them, construct a decision tree for the target variable Y. Then, derive a partition of the feature space $\mathcal{X}$ based on the split rules such that each leaf node corresponds to one segment in the partition. Next, overlay the two partitions resulting in smaller hyper rectangles.

Based on the joint partition, the probability densities $P_D(\mathcal{X})$ and $P_D(Y, \mathcal{X})$ are estimated for $D \in \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_1 \cup \mathcal{D}_2\}$. Let n_r denote the number of segments in the joint partition and n_c the number of classes in $\mathcal{D}_1$ and $\mathcal{D}_2$. To estimate $P_D(\mathcal{X})$, assess the proportion of observations in D that fall into each segment of the joint partition, $\hat{P}_D(\mathcal{X}) \in \mathbb{R}^{n_r}$. For the estimation of the joint density $P_D(Y, \mathcal{X})$, determine the proportion of observations that fall into each segment of the joint partition and belong to each class, $\hat{P}_D(Y, \mathcal{X}) \in \mathbb{R}^{n_r \times n_c}$. Estimate the conditional density $P_D(Y|\mathcal{X})$ by calculating the proportion of observations belonging to each class separately for each segment, $\hat{P}_D(Y|\mathcal{X}) \in \mathbb{R}^{n_r \times n_c}$.

Ntoutsi, Kalousis and Theodoridis (2008) consider the similarity index

$$s(p, q) = \sum_i \sqrt{p_i \cdot q_i}$$

for vectors p and q. If p and q are $(n_r \times n_c)$-matrices, they are interpreted as $(n_r \cdot n_c)$-dimensional vectors. For the conditional distribution, the similarity vector $S(Y|\mathcal{X}) \in \mathbb{R}^{n_r}$ is computed with $S(Y|\mathcal{X})_i = s(\hat{P}_{\mathcal{D}_1}(Y|\mathcal{X})_{i\bullet}, \hat{P}_{\mathcal{D}_2}(Y|\mathcal{X})_{i\bullet})$ and index $i\bullet$ denoting the i-th row. Three similarity measures for datasets are suggested:

1. NTO1 $= s(\hat{P}_{\mathcal{D}_1}(\mathcal{X}), \hat{P}_{\mathcal{D}_2}(\mathcal{X}))$
2. NTO2 $= s(\hat{P}_{\mathcal{D}_1}(Y, \mathcal{X}), \hat{P}_{\mathcal{D}_2}(Y, \mathcal{X}))$
3. NTO3 $= S(Y|\mathcal{X})^T \hat{P}_{\mathcal{D}_1 \cup \mathcal{D}_2}(\mathcal{X})$.

For probability estimates p and q, $s(p, q)$ is bounded by $0 \leqslant s(p, q) \leqslant 1$. Therefore, measures NTO1 and NTO2 are bounded by $0 \leqslant$ NTO1, NTO2 $\leqslant 1$. Measure NTO3 is bounded by $0 \leqslant$ NTO3 $\leqslant 1$ because $S(Y|\mathcal{X})_i \leqslant 1$ for all

$i \in \{1, \ldots, n_r\}$ and $\sum_{i=1}^{n_r} (\hat{P}_{\mathcal{D}_1 \cup \mathcal{D}_2}(\mathcal{X}))_i = 1$. For the three measures, high values correspond to high similarity.

Ntoutsi, Kalousis and Theodoridis (2008) do not specify how to choose the hyperparameters for the decision tree computation. Especially for decision trees with many leaf nodes, it is likely that the joint partition contains empty or very sparse segments.

3.3.2. Comparison of probability densities based on kernel density estimation

Comparison of kernel density estimates in L^2-norm I Ahmad and Cerrito (1993) define a test statistic based on the L^2-norm of the difference between kernel density estimates. It has to be assumed that densities exist and are differentiable up to second order with bounded derivatives. Also, assumptions on the kernel function are needed. Under these assumptions, an asymptotic test is proposed. The test statistic has an asymptotic normal distribution under both null and alternative hypothesis, with less restrictive assumptions than in related articles. Different sequences of weights have to be chosen to calculate the estimators of the test statistic and its variance.

Comparison of kernel density estimates in L^2-norm II Anderson, Hall and Titterington (1994) present a test based on the integrated square distance between kernel-based density estimates as well as asymptotic distributional results and power calculations for this test. For the calculation, a kernel and the bandwidth for kernel density estimation must be chosen. Anderson, Hall and Titterington (1994) use a bandwidth of $h = 1$ in their derivation of theoretical results. They show that the minimum distance at which the statistic can discriminate between f_1 and f_2 can be expressed by $f_2 = f_1 + N^{-1/2} h^{-p/2} \cdot g$ under the condition that $h \to 0$ as $n_1, n_2 \to \infty$, where n_1 and n_2 are assumed to be of the same order of magnitude. For this, they use the assumption that $f_1 = f_2$ has two continuous, square-integrable derivatives. Additionally, regularity conditions on the kernel are made: it has to be bounded, absolutely integrable, and its Fourier transform must not vanish on any interval. These conditions are e.g. fulfilled for p-variate uniform, standard normal densities, and p-variate forms of Epanechnikovs kernel. The test statistic is not asymptotically normally distributed. The resulting test is consistent.

Comparison of kernel density estimates based on empirical likelihood Cao and van Keilegom (2006) propose a test based on comparing kernel estimators of the two density functions for continuous distributions based on an empirical likelihood criterion. They state that "for high-dimensional distributions, the curse of dimensionality implies that the method will not be applicable in practice". An alternative presented is to use a model-based approach that only concentrates on elliptically contoured distributions.

3.4. Comparison of characteristic functions

The idea of comparing characteristic functions is that they fully characterize the distribution. Meintanis (2016) reviews tests based on empirical characteristic functions, including tests for the two- and k-sample problem and an interpretation in terms of moments of a general test statistic, which is based on integrating over the weighted squared difference of empirical characteristic functions. Different tests can be derived from this general test statistic via different weight functions (e.g. Alba Fernández, Jiménez Gamero and Muñoz García (2008), Lindsay, Markatou and Ray (2014), Hušková and Meintanis (2008)).

Comparison of empirical characteristic functions in L^2-norm Alba-Fernández, Ibáñez-Pérez and Jiménez-Gamero (2004) consider the L^2-norm of the difference between empirical characteristic functions (ecf). Their test statistic is the integral of the weighted absolute difference between the ecfs of the two samples. They use a trigonometric Hermite interpolant to obtain a numerical integration formula to approximate the test statistic. The p-value of the test is estimated by a Bootstrap algorithm similar to Alba, Barrera and Jiménez (2001). The test can be applied to continuous and discrete data. The assumption of existing second moments is made for all theoretic results and additional assumptions on the approximation of the test statistic are required to show consistency against a wide range of fixed alternatives.

Comparison of empirical characteristic functions based on weighted integrals Alba Fernández, Jiménez Gamero and Muñoz García (2008) propose a class of tests based on the weighted integral of empirical characteristic functions. The tests are not asymptotically distribution-free and neither the null distribution nor the asymptotic null distribution of the test statistic are known. Instead, a permutation or Bootstrap procedure yields asymptotically consistent approximations of the null distribution. The weight function must be chosen, but a choice of the weight function that is not very restrictive already yields consistency against any fixed alternative. Specifically, if the weight function is chosen such that the distance between populations is larger than zero for any $F_1 \neq F_2$, the test is consistent against any fixed alternative. This is for example fulfilled for weight functions with positive density for almost all points in $\mathbb{R}^p$. The weight function also influences the computing time. There are no conditions assumed on the populations. The method can be applied to continuous as well as discrete data of any arbitrary fixed dimension. The test is a generalization of tests in Alba, Barrera and Jiménez (2001) and Alba-Fernández, Ibáñez-Pérez and Jiménez-Gamero (2004). According to Li, Hu and Zhang (2022), the choice of the weight function is a difficult problem.

Comparison of empirical characteristic functions based on jackknife empirical likelihood Liu, Xia and Zhou (2015) develop a jackknife empirical likelihood (JEL) test by incorporating characteristic functions. The test statistic reduces to a two-sample U-statistic, which simplifies the estimation. For fixed

dimension, the authors derive a nonparametric Wilks's theorem. For $p \to \infty$, $p = o(N^{1/3})$, under some mild conditions the normalized JEL ratio statistic has a standard normal limit. For $p > N$ an alternative version of the JEL test is proposed that has an asymptotical χ^2_2 distribution under the null. For computing the test statistic, a range over which the statistic is calculated has to be chosen. According to Li, Hu and Zhang (2022), the choice of the weight function is a difficult problem. Liu, Liu and Zhou (2019) claim that the test works well in the case of small samples and also for asymmetric data.

Comparison of empirical characteristic functions based on characteristic distance Li, Hu and Zhang (2022) introduce the characteristic distance, which does not need any assumptions on moments and parameters and fully characterizes the homogeneity of two distributions since it is nonnegative and equal to zero if and only if the distributions are equal. The characteristic distance relies on the equivalence of almost surely equal characteristic functions and the equality

$$\mathbb{E}\left(\exp\left(i\langle X, X'\rangle\right)|X'\right) = \mathbb{E}\left(\exp\left(i\langle Y, X'\rangle\right)|X'\right), \ a.s.,$$

where i denotes the imaginary unit. Let X', X'' and Y', Y'' denote independent copies of X and Y, respectively. Then the characteristic distance is defined as

$$\begin{aligned}\mathrm{CD}(X,Y) = \mathbb{E}\big[\|\,\mathbb{E}\left(\exp\left(i\langle X'', X - X'\rangle\right)|X - X'\right) \\ - \mathbb{E}\left(\exp\left(i\langle Y, X - X'\rangle\right)|X - X'\right)\|^2\big] \\ + \mathbb{E}\big[\|\,\mathbb{E}\left(\exp\left(i\langle X, Y - Y'\rangle\right)|Y - Y'\right) \\ - \mathbb{E}\left(\exp\left(i\langle Y'', Y - Y'\rangle\right)|Y - Y'\right)\|^2\big].\end{aligned}$$

An empirical version is obtained by replacing the conditional expectations with empirical means. Li, Hu and Zhang (2022) derive the distribution of the characteristic distance under the null and alternative hypotheses. They call the resulting test distribution-free, but the asymptotic distribution depends on an unknown true distribution, so a permutation test is used instead. According to Li, Hu and Zhang (2022), the test has a clear and intuitive probabilistic interpretation and its estimator is easy to calculate. They derive the asymptotic distribution and show that the test is consistent against any generic alternative. The test is robust since no moment assumptions are needed and it is free of any tuning parameters.

3.5. Methods based on multivariate ranks

For the univariate two-sample problem, tests based on ranks are popular methods. Since $\mathbb{R}^p$ has no natural ordering, the generalization of these methods to the multivariate case is not straightforward. Different approaches to multivariate rank procedures are presented below.

Ranks based on projections obtained by binary classification The first procedures based on multivariate ranks, which are often only applicable to the location or scale problem instead of the general two-sample problem, were proposed by Puri et al. (1971), Randles and Peters (1990) (only location problem), Hettmansperger and Oja (1994) (also only for location problem), Choi and Marden (1997) (theory only for location problem), Hettmansperger, Möttönen and Oja (1998) (also only for location problem). According to Ghosh and Biswas (2016), these procedures usually yield poor results for high-dimensional data, none of them can be used for $p > N$, and none is distribution-free in finite sample situations, and although some of them are asymptotically distribution-free and for some one can implement conditional versions using permutation type techniques. The test from Liu and Singh (1993) (can only detect location and/or additional dispersion differences) is based on a quality index based on ranks. The test is distribution-free but computationally infeasible in high dimensions and also cannot be used when $p > N$. Ghosh and Biswas (2016) instead present a general procedure for multivariate generalizations of univariate distribution-free tests based on ranks of real-valued linear functions of multivariate observations. The linear function is obtained by solving a classification problem between the two distributions. The procedure is exactly distribution-free in finite samples under very general conditions and applicable even when the dimension exceeds the sample size.

The idea behind the test procedure is that $H_0 : F_1 = F_2$ can be expressed as $F_{\beta,1} = F_{\beta,2} \forall \beta \in \mathbb{R}^p$, where $\beta^T X \sim F_{\beta,i}$ if $X \sim F_i$. This means, that if the distributions differ, it is expected that for some β values $F_{\beta,1}$ differs from $F_{\beta,2}$. Ghosh and Biswas (2016) choose a projection such that the separation between observations of different samples is maximized by using the direction vector of a linear classifier that discriminates between the samples. They train a support vector machine (SVM) or distance-weighted discrimination (Marron, Todd and Ahn, 2007) on a training set. Then projections are calculated on a test set and a one-sided KS test or Wilcoxon test is performed on these projections. This procedure is repeated for several train/test splits and the test statistics are averaged. Then either a randomized test is used or tests with Bonferroni correction/control of FDR for each split are performed, and H_0 is rejected if any of the tests reject. Ghosh and Biswas (2016) derive asymptotic results for $p \to \infty$ under the assumption of uniformly bounded fourth moments, weak dependence among component variables, and convergence of variances and the squared differences between expectations (the last two hold automatically for i.i.d. components with bounded second moments). These assumptions ensure that under H_1 the amount of information for discrimination grows to infinity as the dimension increases. Under these assumptions, consistency is shown. The same holds even if $p \to \infty$ and $N \to \infty$ such that $N/p^2 \to 0$ and also for general one-sided linear rank statistics that are linear combinations of a monotonically increasing function applied to the ranks. The test remains distribution-free if data is transformed by a real-valued measurable function chosen on the training set (instead of linear transformation as before). Power is maximized if this transformation is chosen as the likelihood ratio (LR), but this is hard to esti-

mate due to the curse of dimensionality. An alternative is to use a nonlinear SVM with a suitable kernel choice. The method can also be seen as a method based on binary classification.

Ranks based on optimal transport I Ghosal and Sen (2021) propose multivariate nonparametric tests free of tuning parameters based on a new notion of multivariate quantiles/ranks, which were introduced by Chernozhukov et al. (2017) and use optimal transport theory (see 3.11). The idea behind this approach is based on the insight that in the univariate case, ranks can be understood as the solution of transporting the data distribution to the uniform distribution. Therefore, the multivariate ranks are defined as the solution to the corresponding multivariate optimal transport problem. The test statistic then consists of the integral (w.r.t. a reference distribution) over the squared distance of the rank map of the pooled sample applied to the quantile maps of the individual samples. The authors show asymptotic consistency under fixed alternatives and derive rates of convergence of the test statistics under null and alternative hypotheses. Ghosal and Sen (2021) make the assumption of absolutely continuous distributions. It is not known if the test is (asymptotically) distribution-free. Instead, a permutation test is performed. The test statistic tends to zero under H_0 for $n_1, n_2 \to \infty$ such that $n_1/N \to$ const. The test is implemented in the `R` (R Core Team, 2021) package `testOTM` (Xu, 2019).

Ranks based on optimal transport II Deb, Bhattacharya and Sen (2021) propose distribution-free analogs of Hotelling's T^2 test based on optimal transport. The test statistic is the squared difference of the mean of multivariate ranks between two samples. Here, ranks are assigned for the pooled sample based on optimal transport. They claim consistency for general alternatives and efficiency under location shift alternatives. Deb, Bhattacharya and Sen (2021) aim to design multivariate nonparametric distribution-free tests that attain similar asymptotic relative efficiency (ARE) values compared to Hotelling's T^2 test. The resulting test statistic follows a limiting χ^2_p distribution under the null and a non-central χ^2 distribution under contiguous alternatives. The test is consistent against large classes of natural alternatives including a location shift model and a contamination model. Deb, Bhattacharya and Sen (2021) present numerous lower bounds on the ARE for multiple subfamilies of multivariate probability distributions, e.g. distributions with independent components (for multiple subfamilies there is no loss of efficiency). Their test is exactly distribution-free and generalizes the two-sided Wilcoxon rank-sum test and the van der Waerden score test for $p > 1$. The test can be further generalized by calculating scores from ranks. Then assumptions on the score function and its covariance matrix are required. A reference distribution is needed to define the ranks. The choice of this reference distribution affects the ARE of the resulting test. Throughout, the assumption of Lebesgue absolutely continuous probability measures is used. Under this assumption, weak convergence of the rank distribution to a reference distribution is shown. No moment assumptions are made for the distributions that are compared, only for the reference distribution and the score function.

The moment assumptions are for example always satisfied for the identity as score function and $U[0,1]^p$ as reference distribution. A basic version of the statistic is presented in Hallin, Hlubinka and Hudecová (2022) for the special case that the reference distribution is spherical uniform and for a specific choice of rank set but the authors did not study theoretical properties as consistency and asymptotic efficiency of the resulting test. The results of Deb, Bhattacharya and Sen (2021) prove consistency and can be used to derive ARE for this special case as well.

Ranks based on similarity graphs Zhou and Chen (2023) define ranks based on similarity graphs and use these for constructing a two-sample test. They define a sequence of simple similarity graphs $\{G_l\}_{l=0}^K$ on the pooled sample via

$$G_{l+1} = G_l \cup G_{l+1}^*$$

with

$$G_{l+1}^* = \arg \max_{G' \in \mathcal{G}_{l+1}} \sum_{(i,j) \in G'} S(Z_i, Z_j),$$

where G_0 has no edges, $\mathcal{G}_{l+1} = \{G' \in \mathcal{G} : G' \cap G_l = \varnothing\}$ with $\mathcal{G}$ the set of graphs that fulfill specific user-defined constraints, and $S(\cdot, \cdot)$ a similarity measure, e.g. the negative Euclidean distance for Euclidean data, and $Z_1, \ldots, Z_N$ denoting the pooled sample. This construction scheme includes as special cases the K-nearest neighbor graph, the K-minimum spanning tree, the K-minimum distance non-bipartite pairing, and the K-shortest Hamiltonian path. Based on the sequence of similarity graphs, Zhou and Chen (2023) define the following two graph-based rank matrices $R = (R_{ij})_{i,j=1}^N$. The *graph-induced ranks* are defined as

$$R_{ij} = \sum_{l=1}^{K} \mathbb{1}\left((i,j) \in G_l\right)$$

and the *overall ranks* are defined as

$$R_{ij} = \operatorname{rank}\left(S\left(Z_i, Z_j\right), G_K\right),$$

where $\operatorname{rank}\left(S\left(Z_i, Z_j\right), G_K\right)$ denotes the rank of $S\left(Z_i, Z_j\right)$ among the values $\{S\left(Z_u, Z_v\right)\}_{(u,v) \in G_K}$ if $(i,j) \in G_k$ and zero otherwise. The graph-induced rank R_{ij} can be interpreted as the number of graphs that contain the edge (i,j) in the sequence of graphs. The overall rank can be interpreted as the rank of the similarity of edges in the graph G_K. Both depend on the choice of K. For the test, the symmetrized rank matrix $1/2(R + R^T)$ is used. For convenience, it is also denoted by R.

For the test statistic, the within-sample rank sums of the first and second samples are defined as

$$U_x = \sum_{i,j=1}^{n_1} R_{ij}, U_y = \sum_{i,j=n_1+1}^{N} R_{ij}.$$

Using these, the *Rank In Similarity graph Edge-count two-sample test (RISE)* statistic is defined as

$$T_R = (U_x - \mu_x, U_y - \mu_y)\Sigma^{-1}(U_x - \mu_x, U_y - \mu_y)^T,$$

where $\mu_x = \mathbb{E}(U_x)$, $\mu_y = \mathbb{E}(U_y)$, and $\Sigma = \mathbb{C}\text{ov}((U_x, U_y)^T)$. The quantities μ_x, μ_y, and Σ are explicitly calculated under the permutation null hypothesis, and sufficient conditions under which Σ is invertible and therefore T_R is well-defined are given. The test statistic can be decomposed into two quantities Z_w and Z_{diff}, which can be related to the graph-based tests of Chen and Friedman (2017), Chen, Chen and Su (2018), and Zhang and Chen (2019). For small samples, the exact permutation null distribution can be used for testing. For large samples and under several assumptions on the similarity graphs, the asymptotic χ^2_2-distribution of T_R can be used for testing. For continuous distributions and the K-MST or K-NN graph based on the Euclidean distance and with $K = \mathcal{O}(1)$, the test is consistent for $n_1, n_2 \to \infty$ and $n_1/N \to \pi \in (0, 1)$. Under several other assumptions, consistency of the test using the graph-induced ranks for the K-NN graph or for using the overall ranks for the K-minimum distance non-bipartite pairing is shown. Extensions of the RISE test using kernel functions for the similarity measure and using another graph-based rank definition are briefly presented. The RISE test can also be classified as a graph-based approach.

3.6. Discrepancy measures for distributions

There are two main classes of discrepancy measures for distributions: probability metrics and divergences. The best-known subclasses are *Integral Probability Metrics* (IPM, also called probability metrics with a ξ-structure (Zolotarev, 1976, 1984)) as introduced by Müller (1997) and *f-Divergences* (sometimes also called Ali-Silvey distances, going back to Ali and Silvey (1966), or Csiszár's Φ-divergences, going back to Csiszár (1963)). The latter were introduced in the two aforementioned articles. These two classes of Integral Probability Metrics and f-divergences only intersect at the total variation distance as shown by Sriperumbudur et al. (2012).

For probability metrics, Zolotarev (1984) distinguishes between probability metrics with a Λ-structure, probability metrics with a ξ-structure (= IPMs), metrics with a Hausdorff structure, and metrics with an Integral structure. He also notes that there are other metrics that do not have any of these structures, e.g. the Hellinger metric. Other classes of divergences are Bregman-divergences (Bregman, 1967) and the Burbea-Rao divergences (Burbea and Rao, 1982), which will not be discussed here since they require a parametric model of the distributions. Another class is formed by H-divergences, which overlap with both f-divergences and IPMs and were recently introduced by Zhao et al. (2021).

A detailed overview of the theory on probability metrics as well as a comprehensive list of examples can be found in Rachev (1991). A more applied description is given in Rachev, Stoyanov and Fabozzi (2008, 2011). An overview of inequalities specifying the relationships between different f-divergences is given in

Sason and Verdú (2016). In the following, we will focus on the main ideas and important examples. We will again only look at discrepancy measures for comparing two distributions. There are often versions of the measures discussed here for the case of comparing an empirical distribution to a known distribution in a goodness-of-fit context. This case is for example discussed in Basu, Shioya and Park (2011) in more detail.
Note that many of the methods presented below can also be seen as methods based on inter-point distances or methods based on cumulative distribution functions or density functions, e.g. all f-divergences are based on density functions.

3.6.1. Probability (semi-)metrics

Zolotarev (1984) reviews probability metrics and identifies four classes, which are defined via a functional $\mu(X, Y)$ on the space of bivariate distributions. This functional takes values on $[0, \infty]$, with the following three properties:

1. $\mathbb{P}(X = Y) = 1 \Rightarrow \mu(X, Y) = 0$
2. $\mu(X, Y) = \mu(Y, X)$
3. $\mu(X, Y) \leqslant \mu(X, Z) + \mu(Z, Y)$

Note that Zolotarev (1984) argues to use the term probability *metric* although the first condition is only analogous to conditions from functional analysis that characterizes a semimetric and not a metric.

In contrast, Müller (1997) defines a probability metric d with the following properties:

1. $d(F_1, F_2) = 0 \Leftrightarrow F_1 = F_2$ (positive definite)
2. $d(F_1, F_2) = d(F_2, F_2)$ (symmetry)
3. $d(F_1, F_3) \leqslant d(F_1, F_2) + d(F_2, F_3)$ (triangle inequality).

If the first property is replaced by

$$d(F, F) = 0 \text{ for all distributions } F,$$

d is called a probability semimetric.
The latter way to distinguish between probability metric and semimetric is more common in the literature (e.g. Rachev, 1991) and more precise and is therefore used in the following.
Zolotarev (1984) propose a distinction between *simple* and *compound* metrics, Rachev (1991) also distinguishes *primary* metrics from the former. He defines three types as follows: Let P be the joint distribution of F_1 and F_2 and let Q be the joint distribution of two distributions G_1, G_2 defined on the same sample spaces as F_1 and F_2. Denote the space of joint distributions to which P and Q belong by $\mathcal{P}$. A (semi)metric $d : \mathcal{P} \to [0, \infty)$ is called *primary (semi)metric*, if it is a probability (semi)metric and there exists a function $h : \mathcal{P}_1 \to \mathbb{R}^J, J \in \mathbb{N}$, such that

$$(h(F_1) = h(G_1) \wedge h(F_2) = h(G_2)) \Leftrightarrow d(P) = d(Q).$$

$\mathcal{P}_1$ denotes the set of Borel probability measures for some separable metric space. Examples of primary metrics are distances between moments of distributions as well as the L_q-*Engineer metric*

$$\mathrm{EN}(X, Y; q) = \left[\sum_{i=1}^{p} |\mathbb{E}(X_i) - \mathbb{E}(Y_i)|^q\right]^{\min(q,1/q)} \quad \text{with } q > 0,$$

where X_i, Y_i denote the ith component of the p-dimensional random vectors $X \sim F_1$ and $Y \sim F_2$.
A probability (semi)metric $d : \mathcal{P} \to [0, \infty)$ is called a *simple semimetric*, if for each $P \in \mathcal{P}$ with marginals F_1, F_2:

$$F_1 = F_2 \Rightarrow d(P) = 0$$

and a *simple metric* if the converse implication also holds. Simple metrics as well as primary metrics only depend on the marginals F_1, F_2 instead of their joint distribution and can therefore also be denoted by $d(F_1, F_2)$ instead of $d(P)$. Examples of simple (semi)metrics are the Kantorovich metric (Kantorovich, 1960), Prokhorov metric (Prokhorov, 1956), Birnbaum-Orlicz metric (Birnbaum and Orlicz, 1931), and Zolotarev's semimetric (Zolotarev, 1984).
In the sense of Rachev (1991), every probability metric is a *compound metric*. In other papers, the term compound metric is used only for metrics that are not simple (Rachev, 1991). Examples of compound metrics are the Ky Fan metrics Fan, 1943.

Probability metrics can also be classified based on their structure, according to Zolotarev (1984). The different structures are presented below, mainly following the overview of Rachev (1991). We will denote the sample space by S and assume that it is a metric space with a corresponding metric that we denote by d' to avoid confusion with the probability metrics denoted by d.

Hausdorff structure Following Rachev (1991), a probability semimetric d is said to have a *Hausdoff structure* or a *h-structure* if it can be represented in the following form:

$$d(X, Y) = h_{\lambda,\phi,\mathfrak{B}_0}(X, Y) := \max\left\{h'_{\lambda,\phi,\mathfrak{B}_0}(X, Y), h'_{\lambda,\phi,\mathfrak{B}_0}(Y, X)\right\},$$

where

$$h'_{\lambda,\phi,\mathfrak{B}_0}(X, Y) = \sup_{A\in\mathfrak{B}_0} \inf_{B\in\mathfrak{B}_0} \max\left\{\frac{1}{\lambda} r(A, B), \phi(X, Y; A, B)\right\}$$

and

$$r(A, B) = \inf\{\varepsilon > 0 : A^\varepsilon \supseteq B, B^\varepsilon \supseteq A\}$$

is the Hausdorff semimetric in the Borel σ-algebra $\mathfrak{B}(S)$ with A^ε denoting the open ε-neighborhood of A, $\lambda > 0$, $\mathfrak{B}_0 \subseteq \mathfrak{B}(S)$ and ϕ such that

1. $\mathbb{P}(X = Y) = 1 \Rightarrow \phi(X, Y; A, B) = 0 \,\forall A, B \in \mathfrak{B}_0$, and

2. $\exists$ constant $K_\phi \geqslant 1 : \forall A, B, C \in \mathfrak{B}_0$ and random variables X, Y, Z

$$\phi(X, Z; A, B) \leqslant K_\phi[\phi(X, Y; A, C) + \phi(Y, Z; C, B)].$$

Examples are the Lévy metric for univariate distributions

$$L(X, Y) = L(F_1, F_2) = \inf\{\varepsilon > 0 : F_1(x-\varepsilon) - \varepsilon \leqslant F_2(x) \leqslant F_1(x+\varepsilon) + \varepsilon, x \in \mathbb{R}\}$$

and the Prokhorov metric $\pi_\lambda, \lambda > 0$

$$\pi_\lambda(F_1, F_2) := \inf\{\varepsilon > 0 : \mathbb{P}_1(C) \leqslant \mathbb{P}_2(C^{\lambda\varepsilon}) + \varepsilon \text{ for any } C \in \mathcal{C}\},$$

where $\mathcal{C}$ denotes the set of all nonempty closed subsets of S. Every semimetric has the trivial Hausdorff representation $h_{\lambda,\phi,\mathfrak{B}_0} = \mu$ with $\mathfrak{B}_0$ a singleton, e.g. $\mathfrak{B}_0 \equiv A_0$ for some set A_0, and $\phi(X, Y; A_0, A_0) = d(X, Y)$ (Rachev, 1991, pp. 51-68). The original definition of Zolotarev (1984) explicitly excludes the trivial representation.

Λ-structure A semimetric d has a Λ-structure if there exists a non-negative function ν that satisfies the conditions

1. $\mathbb{P}(X = Y) = 1 \Rightarrow \nu(X, Y; t) = 0 \; \forall t \geqslant 0$
2. $\nu(X, Y; t) = \nu(Y, X; t) \; \forall t \geqslant 0$
3. $0 \leqslant t' < t'' \Rightarrow \nu(X, Y; t') \geqslant \nu(X, Y; t'')$
4. $\nu(X, Z; t' + t'') \leqslant \nu(X, Y; t') + \nu(Y, Z; t'') \; \forall t', t'' \geqslant 0$

such that d can be represented as

$$d(X, Y) = \Lambda_{\lambda,\nu}(X, Y) := \inf\{\varepsilon > 0 : \nu(X, Y; \lambda\varepsilon) < \varepsilon\}$$

for some $\lambda > 0$.
It can be shown that each semimetric has a trivial Λ-structure (Rachev, 1991, pp. 68-69). Again, this trivial representation is excluded in the original definition by Zolotarev (1984). Examples of (semi-)metrics with a Λ-structure in the strong sense are the Ky Fan metric and the generalized Lévy-Prokhorov metrics. In general, each semimetric with a Hausdoff structure also has a Λ-structure (Rachev, 1991, p. 69).

Integral structure Zolotarev (1984) additionally defines (semi)metrics with an integral structure that comprise those (semi)metrics that can be represented as

$$d(X, Y) = \psi(\mathbb{E}(\phi(h(X, Y)))),$$

where h is a metric in (S, d') which is a measurable function, ϕ is a strictly increasing convex function on $(0, \infty)$ vanishing at zero and ψ is the superposition of a non-decreasing concave function vanishing at zero and the inverse function of ϕ. All metrics with an integral structure are compound metrics. An example are metrics of the form

$$\gamma_q(X, Y) = \left[\mathbb{E}\left((d')^q(X, Y)\right)\right]^{\min(1, 1/q)}, q > 0.$$

Integral probability metrics / probability metrics with a ξ-structure
Probability Metrics with a ξ-structure are better known as Integral probability metrics, going back to Müller (1997). They are based on the idea that if two distributions are identical, any function should have the same expectation under both distributions. Let $\mathcal{F}$ be a set of functions $f : \mathcal{X} \to \mathbb{R}$. Then an integral probability metric is given by

$$\mathrm{IPM}_{\mathcal{F}}(F_1, F_2) = \sup_{f \in \mathcal{F}} \left| \int f \, \mathrm{d}F_1 - \int f \, \mathrm{d}F_2 \right|.$$

All IPMs are probability metrics.
An example for an IPM is the *Dudley metric* β, which is generated by $\mathcal{F}_\beta = \{f : \|f\|_\infty \leqslant 1, \|f\|_L \leqslant 1\}$, where $\|\cdot\|_L$ denotes the Lipschitz-norm defined on a metric space (S, d') as

$$\|f\|_L := \sup_{x \neq y \in S} \frac{|f(x) - f(y)|}{d'(x, y)}$$

and $\|\cdot\|_\infty$ denotes the supremum norm. Another special case is the *Total Variation Metric* (Zolotarev, 1984)

$$\sigma(F_1, F_2) := |F_1 - F_2|(S),$$

where

$$\|\mu\| := |\mu|(S) \tag{2}$$

denotes the total variation norm on the set of all signed measures on an arbitrary measure space $(S, \mathcal{S})$ with total variation

$$|\mu| = \mu^- + \mu^+.$$

Here μ^- and μ^+ denote the negative and positive parts of μ, respectively. The total variation metric has the generator $\mathcal{F}_\sigma := \{2 \cdot \mathbb{1}_B : B \in \mathcal{S}\}$ since

$$\|\mu\| = 2 \sup_{A \in \mathcal{S}} |\mu(A)| \text{ for all signed measures } \mu \text{ with } \mu(S) = 0.$$

It also fulfills the property

$$d(F_1 * G, F_2 * G) \leqslant d(F_1, F_2) \text{ for all probability measures } G. \tag{3}$$

The *stop-loss metric*

$$d_{\mathrm{SL}}(F_1, F_2) = \sup_{t \in \mathbb{R}} |\, \mathbb{E}_{F_1}(X - t)^+ - \mathbb{E}_{F_2}(X - t)^+ |$$

is motivated by risk-theoretic considerations (Gerber, 1979; Rachev and Rüschendorf, 1990) and has the generator $\mathcal{F}_{\mathrm{SL}} = \{s \to \Phi_t(s) = (s - t)^+, t \in \mathbb{R}\}$. It fulfills the condition (3) and additionally, it holds

$$d_{\mathcal{F}}(\delta_a, \delta_b) = d'(a, b), \tag{4}$$

where δ_x denotes the Dirac measure on x. More properties for the univariate case are given in Rachev and Rüschendorf (1990). The last example introduced by Müller (1997) is the *Kantorovich-Rubinstein metric* ξ_1 (Zolotarev, 1984; Dudley, 1989), which is generated by the set of Lipschitz functions $\mathcal{L}_1 = \{\text{Lipschitz functions } f : \|f\|_L \leqslant 1\}$. For $S = \mathbb{R}$ it reduces to

$$\xi_1(F_1, F_2) = \ell_1(F_1, F_2) := \int |F_1(t) - F_2(t)| \, \mathrm{d}t.$$

Additional to the conditions (3) and (4), it fulfills the condition

$$d_{\mathcal{F}}(aF_1, aF_2) = a \, d_{\mathcal{F}}(F_1, F_2). \tag{5}$$

If S is separable, the Kantorovich-Rubinstein metric is the dual representation of the L_1-*Wasserstein distance*

$$W_1(F_1, F_2) = \inf_{\pi \in \Pi(F_1, F_2)} \int d(x, y) \, \mathrm{d}\pi(x, y),$$

where $\Pi(F_1, F_2)$ is the set of joint distributions with marginal distributions F_1 and F_2 (for general Wasserstein distance, see (11)). There is a connection between the Kantorovich-Rubinstein metric and optimal transport (see 3.5, 3.11) via the Kantorovich–Rubinstein duality. For details see, e.g. Rachev and Rüschendorf (1998). Other examples are all L^q-metrics

$$\theta_p(F_1, F_2) := \|F_1 - F_2\|_q,$$

where $q \in [1, \infty]$, as well as the engineer metric (Rachev, 1991, p. 73).

Sriperumbudur et al. (2012) define empirical estimates for the Kantorovich metric, Fortet-Mourier metric, dual-bounded Lipschitz distance (Dudley metric), total-variation distance and kernel distance (Mean Maximum Discrepancy, MMD, see Section 3.9.1) that are easily computable and strongly consistent, except for the total variation distance. They are motivated by their observation that homogeneity tests as an important application are often based on estimates of distances as test statistics. Further, while the MMD and the total variation metric are already successfully applied in this context, most other IPMs are not, due to the lack of good estimates for continuous random variables, especially in the multivariate case. For the application, it is crucial that the statistics have a consistent estimator exhibiting fast convergence behavior and low computational complexity. They show that the estimate for the kernel distance (MMD) in comparison is computationally cheaper, converges at a faster rate to the population value, and its rate of convergence is independent of the dimension p of the space. The additional IPMs considered by Sriperumbudur et al. (2012) are defined as follows. The *Fortet-Mourier metric* is a generalization of the Kantorovich metric with $\mathcal{F} = \|f\|_c \leqslant 1$, where

$$\|f\|_c := \sup\{\frac{|f(x) - f(y)|}{c(x, y)} : x \neq y \in S\}$$

and $c(x,y) = d'(x,y)\max(1, d'(x,a)^{q-1}, d'(y,a)^{q-1})$ for $q \geqslant 1$ and for some $a \in S$. For $q = 1$, this yields the definition of the Kantorovich metric. The kernel (MMD) distance $\gamma_{\mathcal{F}}$ is defined by setting $\mathcal{F} = \{f : \|f\|_{\mathcal{H}} \leqslant 1\}$, where $\|\cdot\|_{\mathcal{H}}$ denotes the norm on the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ induced by the kernel function. For details see Section 3.9.1.
The general empirical estimator for an IPM given samples

$$\{X_1, \dots X_{n_1}\} \sim F_1 \text{ and } \{Y_1, \dots Y_{n_2}\} \sim F_2$$

is defined as

$$\gamma_{\mathcal{F}}(F_{1,n_1}, F_{2,n_2}) = \sup_{f \in \mathcal{F}} \left| \sum_{i=1}^{N} \tilde{Z}_i f(Z_i) \right|, \tag{6}$$

where $F_{1,n_1} := \frac{1}{n_1}\sum_{i=1}^{n_1} \delta_{X_i}$ and $F_{2,n_2} := \frac{1}{n_2}\sum_{j=1}^{n_2} \delta_{Y_j}$ denote the empirical distributions of F_1 and F_2, $N = n_1 + n_2$, with

$$Z = \{Z_1, \dots, Z_N\} = \{X_1 \dots, X_{n_1}, Y_1, \dots Y_{n_2}\}$$

denoting the pooled sample and $\tilde{Z}_i = \frac{1}{n_1}$ when $Z_i = X_i$ for $i = 1, \dots, n_1$ and $\tilde{Z}_i = -\frac{1}{n_2}$ when $Z_i = Y_{i-n_1+1}$ for $i = n_1 + 1, \dots, N$. The computation is not straightforward for arbitrary $\mathcal{F}$, so Sriperumbudur et al. (2012) define the empirical estimators for special cases and show how to calculate them by solving linear programs or using a closed-form expression.

The resulting estimator of the kernel distance can be given as a closed-form expression and therefore is easy to implement compared to the other estimators. Moreover, it is the only presented estimator for which (6) has a unique solution. For the estimators for the Kantorovich and total variation metric, strong consistency is shown under the assumption that (S, d') is a totally bounded metric space. To show strong consistency of the estimator for the kernel distance, the following assumptions are made: For any $r \geqslant 1$ and probability measure F, define the L^r-norm $\|f\|_{F,r} := \left(\int |f|^r \, \mathrm{d}F\right)^{1/r}$ and let $L^r(F)$ denote the metric space induced by this norm. The covering number $\mathcal{N}(\varepsilon, \mathcal{F}, L^r(F))$ is the minimal number of $L^r(F)$ balls of radius ε needed to cover $\mathcal{F}$. $\mathcal{H}(\varepsilon, \mathcal{F}, L^r(F)) := \log \mathcal{N}(\varepsilon, \mathcal{F}, L^r(F))$ is called the entropy of $\mathcal{F}$ using the $L^r(F)$ metric. Define the minimal envelope function as $G(x) := \sup_{f \in \mathcal{F}} |f(x)|$. Under the assumptions

1. $\int_S G \, \mathrm{d}F_1 < \infty$,
2. $\int_S G \, \mathrm{d}F_2 < \infty$,
3. $\forall \varepsilon > 0, \frac{1}{n_1}\mathcal{H}(\varepsilon, \mathcal{F}, L^r(F)) \xrightarrow{F_1} 0$ as $n_1 \to \infty$,
4. $\forall \varepsilon > 0, \frac{1}{n_2}\mathcal{H}(\varepsilon, \mathcal{F}, L^r(F)) \xrightarrow{F_2} 0$ as $n_2 \to \infty$,

strong consistency can be shown for the kernel distance estimator.
Sriperumbudur et al. (2012) also derive convergence rates of the estimators to their population values. For the estimators for the Kantorovich and total variation metric, these convergence rates depend on the dimension p, and thus in large dimensions, more samples are needed to obtain useful estimates. The

rate for the kernel distance is independent of the dimension p. The authors also show how these convergence rates can be used to derive critical values for tests for $H_0 : F_1 = F_2$ vs. $H_1 : F_1 \neq F_2$. Moreover, the theoretical results on convergence and dependence on p are confirmed by simulations for cases in which the measures can be computed exactly.
For the total variation distance, it is shown that the estimator resulting from (6) is not consistent.

Tests based on Wasserstein distances Ramdas, Trillos and Cuturi (2017) present an overview of tests based on Wasserstein distance (11) and their relationships with each other, as well as with the energy and MMD test (see Section 3.9.1) in the multivariate case and the Kolmogorov-Smirnov (KS) test, probability-probability (PP) and quantile-quantile (QQ) plots, and receiver operating (ROC) or ordinal dominance (ODC) curves in the univariate case. They show a connection between the Wasserstein distance and the energy distance via entropic smoothing and then use the connection between energy distance and MMD noted by Sejdinovic et al. (2013). The tests presented are derived only for one-dimensional data and are therefore not discussed further here.

Wang, Gao and Xie (2021) propose a test based on the Wasserstein distance (11) for an optimal linear projection of the data. The idea behind this is to circumvent the curse of dimensionality for the Wasserstein distance through a projection of data into a lower-dimensional space. Wang, Gao and Xie (2022) build on this idea but try to improve the test by using non-linear projections, based on their observation that the original test of Wang, Gao and Xie (2021) cannot efficiently capture features from data with non-linear patterns. They compare the new test to the MMD (Gretton et al. (2006); Section 3.9.1) and ME (Chwialkowski et al. (2015); Section 3.9) test as well as to the old version. Both tests of Wang, Gao and Xie (2021) and Wang, Gao and Xie (2022) rely on a train/test split of the data since an optimal projection of the data needs to be learned before performing the test. The projection of Wang, Gao and Xie (2021) also relies on a kernel and the choice of the kernel is crucial for the test to perform well. No explicit guidelines for the choice of the kernel are given. Moreover, there is no theoretical guarantee of finding the global optimum for the projection in the nonlinear case.

3.6.2. Divergences

There are different definitions of divergences in the literature. Most have in common that a divergence is a discrepancy measure that does not fulfill all criteria for distances or metrics. It is usually required to be a non-negative function. E.g. Sugiyama et al. (2013a) define a divergence d as a pseudo-distance, i.e. it acts like a distance but may violate some of the conditions

1. Non-negativity: $\forall X, Y : d(X, Y) \geqslant 0$
2. Non-degeneracy: $d(X, Y) = 0 \Leftrightarrow X = Y$
3. Symmetry: $d(X, Y) = d(Y, X)$

4. Triangle inequality: $\forall X, Y, Z : d(X, Z) \leqslant d(X, Y) + d(Y, Z)$.

In Zhao et al. (2021), given a finite set or finite-dimensional vector space $\mathcal{X}$ and a set $\mathcal{P}(\mathcal{X})$ of probability distributions on $\mathcal{X}$ with density, a probability divergence is defined as a function $D : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ that satisfies

$$D(F_1, F_2) \geqslant 0$$
$$D(F_1, F_1) = 0 \, \forall F_1, F_2 \in \mathcal{P}(\mathcal{X}).$$

D is called *strict* if $D(F_1, F_2) > 0 \, \forall F_1 \neq F_2$ and *non-strict* otherwise.

f-Divergences f-divergences use the idea that two identical distributions assign the same likelihood to every point and thus measure how far the likelihood ratio is from one (Zhao et al., 2021). Given a convex continuous function $f : \mathbb{R}_+ \to \mathbb{R}$ such that $f(1) = 0$, an f-divergence is defined as

$$D_f(F_1, F_2) = \mathbb{E}_{F_1} \left[f(f_1(X)/f_2(X)) \right],$$

where f_1, f_2 are the density functions of distributions F_1 and F_2 (Csiszár, 1963; Ali and Silvey, 1966).
In the following, we will discuss the most important properties and examples of f-divergences. There is extensive literature on f-divergences in general and also on more details for special f-divergences. For a general discussion of f-divergences see e.g. Liese and Vajda (1987). For a discussion of special f-divergences see the literature cited below and the references therein.

Vajda (2009) discusses metric properties of f-divergences. In general, f-divergences do not fulfill the conditions of a metric. Especially the triangle inequality is violated for all f-divergences except for the total variation metric (or multiples of it). On the other hand, positive powers of f-divergences are probability metrics, if the f-divergence itself is symmetric and bounded. For a general f-divergence D_f, it holds

$$0 \leqslant D_f(F_1, F_2) \leqslant f(0) + f^*(0),$$

where

$$f(0) = \lim_{t \downarrow 0} f(t),$$
$$f^*(0) = \lim_{t \downarrow 0}, f^*(t)$$
$$f^*(t) = t f\left(\frac{1}{t}\right), t > 0.$$

It holds $D_f(F_1, F_2) = 0$ if and only if $F_1 = F_2$. If $F_1 \perp F_2$ (orthogonal, i.e. disjoint supports), then it holds $D_f(F_1, F_2) = f(0) + f^*(0)$. The reverse conclusion holds if the right-hand is finite. An f-divergence is symmetric if and only if

$$\exists c \in \mathbb{R} : f^*(t) = f(t) + c(t - 1).$$

For symmetric f-divergences, the finiteness of f implies the finiteness of f^*. For any f-divergence, the relation

$$D_f(F_1, F_2) = D_{f^*}(F_2, F_1)$$

holds for any two distributions F_1, F_2. The f-divergence between the restrictions of two distributions to a sub-σ-algebra is always smaller or equal to the f-divergence of the unrestricted distributions, with equality if this sub-σ-algebra is sufficient. In the case of a strictly convex function f and a corresponding finite f-divergence, the equality is equivalent to sufficiency of the sub-σ-algebra. There exist representations of each f-divergence based on finite, measurable partitions of the sample space for the general case, and in case of a σ-algebra that is generated by an at most countable partition, it exists a representation in terms of a measurable partition of the sample space.

Vajda (2009) also gives examples of f-divergences and checks if they fulfill the above properties.

The *squared Hellinger distance* is defined as

$$H^2(F_1, F_2) = 2 \int \left(\sqrt{f_1} - \sqrt{f_2} \right)^2 \mathrm{d}\mu,$$

where μ is a σ-finite measure dominating F_1 and F_2 w.r.t which the densities f_1 and f_2 exist. It satisfies all metric conditions in the power $1/2$. The same holds for the *squared Le Cam distance (Vincze-Le Cam distance)* (Vincze, 1981; Le Cam, 1986)

$$\mathrm{LC}^2(F_1, F_2) = \frac{1}{2} \int \frac{(f_1 - f_2)^2}{f_1 + f_2} \, \mathrm{d}\mu.$$

In contrast, no power of the *Kullback-Leibler divergence* (*information divergence*) (Kullback and Leibler, 1951)

$$\mathrm{KL}(F_1, F_2) := \int \log \left(\frac{f_1(x)}{f_2(x)} \right) f_1(x) \, \mathrm{d}x,$$

where f_1, f_2 denote the density functions of F_1 and F_2, is a metric since symmetry is never fulfilled. In addition, powers of the *symmetrized Kullback-Leibler divergence*, which is also known as *Jeffrey's divergence*

$$J(F_1, F_2) = \mathrm{KL}(F_1, F_2) + \mathrm{KL}(F_2, F_1),$$

also do not fulfill the triangle inequality.

Vajda (2009) introduces the *extended ϕ_α-divergences* $D_{\phi_\alpha}(F_1, F_2)$ with

$$\phi_\alpha(t) = \begin{cases} \frac{\alpha}{|\alpha|(1-\alpha)} \left[\left(t^{1/\alpha} + 1 \right)^\alpha - 2^{\alpha-1} (t+1) \right] & \text{if } \alpha(1-\alpha) \neq 0, \\ t \log(t) + (t+1) \log \left(\frac{2}{t+1} \right) & \text{if } \alpha = 1, \\ \frac{|t-1|}{2} & \text{if } \alpha = 0. \end{cases}$$

These are symmetric f-divergences with $f = \phi_\alpha$ strict convex on $(0, \infty)$ unless $\alpha = 0$. Powers $D(F_1, F_2) = D_{\phi_\alpha}(F_1, F_2)^{\pi(\alpha)}$ for

$$\pi(\alpha) = \begin{cases} \frac{1}{2} & \text{if } -\infty < \alpha \leqslant 2, \\ \frac{1}{\alpha} & \text{if } \alpha > 2, \end{cases}$$

fulfill all metric properties. Special cases include the total variation, the Hellinger distance, the Le Cam distance, and the Jensen-Shannon divergence (Lin, 1991) (or scaled versions of them).

Liese and Vajda (1987) define a different class of f-divergences, called I_α-*divergences*. They are generated by the functions

$$I_\alpha(x) = \begin{cases} -\log(x) + x - 1 & \text{if } \alpha = 0, \\ \frac{x^\alpha - \alpha x + \alpha - 1}{\alpha(\alpha - 1)} & \text{if } \alpha \neq 0, \alpha \neq 1, \\ x \log(x) - x + 1, & \text{if } \alpha = 1, \end{cases}$$

with $-\log(0) := \infty$ and $0 \log(0) := 0$. A special case is the KL-divergence for $\alpha = 1$. Moreover, the I_α-divergence is equal to the Rényi divergence of order α for $\alpha \in \{0, 1\}$ and $D_\alpha(F_1, F_2) = \frac{1}{\alpha(\alpha-1)} \log\left(1 + \alpha(\alpha - 1) D_{I_\alpha}(F_1, F_2)\right)$ for $\alpha > 0, \alpha \neq 1$, where $D_\alpha(F_1, F_2)$ denotes the Rényi divergence of order α (see definition below).

Sugiyama et al. (2013a) provide a review of recent advances in direct divergence approximation for some f-divergences. They define a divergence d as a pseudo-distance, i.e. it acts like a distance but may violate some of the conditions.

The first divergence considered is the Kullback-Leibler (KL) divergence. It is almost positive definite and is additive for independent random events, i.e. the divergence for the joint distributions equals the sum of the divergences for the marginal distributions of both variables. Moreover, the KL divergence is invariant for non-singular transformations (Kullback and Leibler, 1951). Advantages of the KL divergence according to Sugiyama et al. (2013a) are that it is compatible with Maximum Likelihood (ML) estimation, invariant under input metric change, that its Riemannian geometric structure is well studied, and that it can be approximated accurately via direct density ratio estimation. However, it is not symmetric, does not fulfill the triangle inequality, its approximation is computationally expensive due to the log function, it is sensitive to outliers, and it is numerically unstable because of the strong non-linearity of the log function and the possible unboundedness of the density-ratio function.

The *Pearson (PE) divergence* (Pearson, 1900), also known as χ^2 *divergence*

$$\mathrm{PE}(F_1, F_2) := \int f_2(x) \left(\frac{f_1(x)}{f_2(x)} - 1 \right)^2 \mathrm{d}x$$

is a squared-loss variant of the KL divergence. Since it is also an f-divergence like the KL divergence, both share similar theoretical properties. Advantages of the PE divergence according to Sugiyama et al. (2013a) are again invariance under input metric change, that it can be accurately estimated via direct

density-ratio estimation, that its estimator can be obtained analytically, so it is computationally much more efficient due to the compatibility of the quadratic function with least squares (LS) estimation. Also, it is more robust against outliers. But it is still not symmetric, also violates the triangle inequality, and the density ratio is possibly unbounded.

One way to overcome the possible unboundedness is to use the *Relative Pearson (rPE) divergence* (Yamada et al., 2013)

$$\mathrm{rPE}(F_1, F_2) := \mathrm{PE}(F_1, F_\alpha) = \int f_\alpha(x) \left(\frac{f(x)}{f_\alpha(x)} - 1 \right)^2 \mathrm{d}x,$$

where for $\alpha \in [0, 1)$, f_α is defined as the density function of the α-mixture

$$F_\alpha = \alpha F_1 + (1 - \alpha) F_2$$

of F_1 and F_2. For $\alpha = 0$ this yields the Pearson divergence. The ratio $\frac{f(x)}{f_\alpha(x)}$ is called the *relative density-ratio* and is always upper-bounded by $\frac{1}{\alpha}$ for $\alpha > 0$. The advantages of the relative Pearson divergence are that it overcomes the unboundedness, is still compatible with LS estimation, and can be approximated in almost the same way as the PE divergence via direct relative density-ratio estimation. Its approximation can still be obtained analytically in an accurate and computationally efficient manner and the rPE is still invariant under input metric change. Its disadvantages are that it violates symmetry and the triangle inequality and that the choice of α may not be straightforward.

Lastly, a divergence presented by Sugiyama et al. (2013a) is the L^2*-distance*

$$L^2(F_1, F_2) := \int (f_1(x) - f_2(x))^2 \, \mathrm{d}x$$

which is a standard distance measure between probability measures. It does not belong to the class of f-divergences but rather to the class of IPMs as it is the special case of the L^q-distances presented above (Section 3.6.1) for $q = 2$. Advantages according to Sugiyama et al. (2013a) are that it is a proper distance measure that the density difference is always bounded as long as each density is bounded. Therefore the L^2-distance is stable without the need for tuning any control parameter. It is also compatible with LS estimation and can be accurately and analytically approximated in a computationally efficient and numerically stable way via direct density-difference estimation (Sugiyama et al., 2013b). In contrast to the aforementioned divergences, the L^2-distance is not invariant under input metric changes.

Due to the advantages listed before, Sugiyama et al. (2013a) argue that Pearson divergence, relative Pearson divergence, and L^2-distance are more useful in practice than the "overwhelmingly popular" KL divergence.

A naive way to approximate the divergences, given samples $X := \{X_i\}_{i=1}^{n_1} \sim F_1$ and $Y := \{Y_j\}_{j=1}^{n_2} \sim F_2$, would be to first obtain estimators for the densities f_1, f_2 and then compute a plug-in approximator similar to the methods for comparing density functions. This violates Vapnik's principle of never trying

to solve a more general problem as an intermediate step when having a restricted amount of information, so Sugiyama et al. (2013a) argue for the use of direct density-ratio or direct density-difference estimation as an alternative. "Direct divergence approximators theoretically achieve optimal convergence rates [...] and compare favorably with the naive density-estimation counterparts" (Sugiyama et al., 2013a). They still suffer from the curse of dimensionality. The key idea behind these techniques is to estimate $\frac{f_1}{f_2}$ or $f_1 - f_2$ without explicitly estimating f_1 and f_2. Therefore, a density-ratio or density-difference model is used. Sugiyama et al. (2013a) make use of the Gaussian density-ratio model

$$r(x) = \sum_{l=1}^{n} \theta_l \exp\left(-\frac{\|x - X_l\|^2}{2\sigma^2}\right)$$

with parameters $\theta_1, \ldots, \theta_n$, or of the Gaussian density-difference model

$$f(x) = \sum_{l=1}^{N} \xi_l \exp\left(-\frac{\|x - c_l\|^2}{2\sigma^2}\right),$$

where $(c_1, \ldots, c_n, c_{n+1}, \ldots, c_N) = (X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2})$ are Gaussian centers and $\xi_1, \ldots, \xi_N$ parameters of the model.

There have been different proposals for f-divergence estimation before. Wang, Kulkarni and Verdu (2005) give an estimator for f-divergences for continuous distributions under certain regularity conditions that is based on estimating the density functions using a data-dependent partition of the observation space. Later, Wang, Kulkarni and Verdu (2006) improved on this method by using nearest neighbor distances instead. This method again only works for continuous distributions. It is shown that the bias and variance of the estimator converge to zero for $n_1, n_2 \to \infty$.

Nguyen, Wainwright and Jordan (2010) define M-estimators for f-divergences and likelihood ratios. They make use of an equivalent reformulation of f-divergences, where an f-divergence can be seen as the solution to a Bayes decision problem which is a convex optimization problem. Nguyen, Wainwright and Jordan (2010) propose a kernel-based implementation for estimation. They assume equal sample sizes and the concrete form of the estimator is only given for the KL-divergence, although the ideas for generalization to general f-divergences with differentiable and strictly convex f are later also presented. Under several assumptions, mainly on the density ratio, consistency and convergence rates can be shown for the estimators.

Kanamori, Suzuki and Sugiyama (2012) define tests based on f-divergences using density-ratio models for estimation. They need many assumptions for their theory. Kanamori, Suzuki and Sugiyama (2012) derive an optimal estimator for f-divergences (regarding asymptotic variance) based on a semiparametric density-ratio model. They use this estimator as a test statistic. The critical value is calculated based on the asymptotic χ^2 distribution of the test statistic. The choice of a specific f-divergence is left open, but up to first order, the local power does not depend on the chosen f-divergence. The choice of the parametric

model for density ratio is left open as well. Conditions on the model to obtain optimality of the f-divergence estimator are given. Moreover, different examples for choosing the model and f-divergence are presented that fulfill the conditions for optimality. Under several additional assumptions, it is shown that the local asymptotic power of the test is equal to that of the empirical likelihood score test of Fokianos et al. (2001) (not described here due to restrictive assumptions on distributions) if the density ratio model is correctly specified and that the power is larger or equal under certain misspecifications of the density ratio model. To calculate ratios of densities, implicit assumptions are required.

A generalization of f-divergences also known as *f-dissimilarity* for simultaneously comparing multiple distributions can be found in Györfi and Nemetz (1975). This extension to the k-sample case is discussed in more detail by García-García and Williamson (2012). It is shown to fulfill all properties as the two-sample version, i.e. the information processing property, reflexivity, invariance to affine terms, uniqueness, change of order, and bounds hold as for the two-sample case. Other extensions did not keep these properties. Define $\boldsymbol{P}_{[k]} = (P, \dots, P_k)^T$, $t^j = \frac{1}{\mathrm{d}P_j}\boldsymbol{P}_{[k]}$ and $\tilde{t}^j = \left(\frac{\mathrm{d}P_1}{\mathrm{d}P_j}, \dots, \frac{\mathrm{d}P_{j-1}}{\mathrm{d}P_j}, \frac{\mathrm{d}P_{j+1}}{\mathrm{d}P_j}, \dots, \frac{\mathrm{d}P_k}{\mathrm{d}P_j}\right)^T$. Then the multi-distribution f-divergence or f-dissimilarity is defined as

$$\mathbb{I}_{\phi,j}(\boldsymbol{P}_{[k]}) = \mathbb{E}_{P_j}[\phi(t^j)] = \mathbb{E}_{P_j}[f_j(\tilde{t}^j)],$$

where the jth distribution is chosen as a reference measure and $f_j \in \mathcal{C}_1^{k-1}$ is a convex function, $\mathcal{C}_1^k := \{\phi : [0,\infty)^k \to \mathbb{R}, \phi \text{ convex}, \phi(\mathbf{1}_k) = 0\}$ such that $f_j(\tilde{t}^j) = \phi(t^j)$. The two expressions are equivalent. The second notation with $k-1$ terms matches the usual f-divergence definition for two distributions. The multi-distribution f-divergence can be seen as a two-step procedure. First, the probability distributions are relativized by taking Radon-Nikodym derivatives with respect to the chosen reference distribution. Second, the dispersion of the resulting likelihood ratio is measured using the convex function.

Rényi divergence Properties of the *Rényi divergence (of order α)* (Rényi, 1961)

$$D_\alpha(F_1, F_2) = \begin{cases} \frac{1}{\alpha-1}\log\left(\int f_1^\alpha f_2^{1-\alpha}\,\mathrm{d}\mu\right) & \alpha < 1 \\ \frac{1}{\alpha-1}\log\left(\int f_1^\alpha / f_2^{\alpha-1}\,\mathrm{d}\mu\right) & \alpha > 1, \end{cases}$$

are described in van Erven and Harremoës (2014). Here, the conventions $0/0 := 0$ and $x/0 := \infty$, $x > 0$ are applied. Like the Kullback-Leibler divergence, the Rényi divergence is particularly popular in information theory since it can be motivated by coding. The KL divergence is a special case of Rényi divergence for order $\alpha = 1$. The Rényi divergence is only symmetric for order $\alpha = 1/2$, and in that case, it is connected to the squared Hellinger distance. In general, it fulfills the so-called skew symmetry: $D_\alpha(F_1, F_2) = \frac{\alpha}{1-\alpha} D_{1-\alpha}(F_2, F_1)$ for $\alpha \in (0,1)$. For $\alpha = 2$, it is a function of the χ^2-divergence. The Rényi divergence is nondecreasing in its order and it is an upper bound for $\alpha/2$ times the total variation for $\alpha \in (0,1]$. For $F_{1;1}, F_{1;2}, \dots$ and $F_{2;1}, F_{2;2}, \dots$ and $F_1^N = \times_{i=1}^N F_{1;i}$,

$F_2^N = \times_{i=1}^N F_{2;i}$ it holds

$$\sum_{i=1}^{N} D_\alpha(F_{1;i}, F_{2;i}) = D_\alpha(F_1^N, F_2^N)$$

for any $\alpha \in [0, \infty]$ and any $N \in \mathbb{N}$ as well as for any $\alpha \in (0, \infty]$ and $N \in \mathbb{N} \cup \{\infty\}$ (additivity). Furthermore, the dominance of measures can be characterized by $F_1 \ll F_2$ if and only if $D_0(F_1, F_2) = 0$. On the other hand it holds that $F_1 \perp F_2$ if and only if $D_\alpha(F_1, F_2) = \infty$ for some $\alpha \in [0, 1)$ or (equivalently) all $\alpha \in [0, \infty]$. Rényi divergence is nonnegative for all $\alpha \in [0, \infty]$, but D_α is neither a metric nor the square of a metric for any order. Similar to f-divergences, the Rényi divergence between the restrictions of two distributions to a sub-σ-algebra is always smaller or equal to the Rényi divergence of the unrestricted distributions. For $\alpha > 0$, the Rényi divergence is equal to 0 if and only if the distributions are the same. For $\alpha = 0$, $D_\alpha(F_1, F_2) = 0$ if and only if $F_1 \ll F_2$. Van Erven and Harremoës (2014) extend the Rényi divergence to negative orders. The results for positive orders carry over to negative orders with reversed properties in most cases.

Relative information of type s Taneja and Kumar (2004) give a generalization of the KL-divergence similar to the Rényi divergence that is defined only for discrete distributions. In addition, they present an overview of inequalities between f-divergences as well as Rényi divergences and their class of *relative information of type s*.

H-divergence In Zhao et al. (2021), given a finite set or finite-dimensional vector space $\mathcal{X}$ and a set $\mathcal{P}(\mathcal{X})$ of probability distributions on $\mathcal{X}$ that have a density, a probability divergence is defined as a function $D : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to \mathbb{R}$ that satisfies

$$D(F_1, F_2) \geqslant 0$$
$$D(F_1, F_1) = 0 \; \forall F_1, F_2 \in \mathcal{P}(\mathcal{X}).$$

D is called *strict* if $D(F_1, F_2) > 0 \; \forall F_1 \neq F_2$ and *non-strict* otherwise. Different probability divergences are presented. The class of H-divergences is introduced by Zhao et al. (2021). It makes use of H-entropies. The idea is that distributions are different if the optimal decision loss is higher on their mixture than on each individual distribution, so the generalized entropy of the mixture distribution $(F_1 + F_2)/2$ is compared to the generalized entropy of F_1 and F_2. If F_1 and F_2 are different, it is more difficult to minimize the expected loss under the mixture, hence it should have higher generalized entropy. If the distributions are identical, the mixture is identical to F_1 and to F_2 and they all have the same generalized entropy.
For an action space $\mathcal{A}$ and loss function $\ell : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ a corresponding H-entropy

$$H_\ell(F) = \inf_{a \in \mathcal{A}} \mathbb{E}_F[\ell(X, a)]$$

is the Bayes optimal loss of a decision maker who must select some action a not for a particular x, but in expectation for a random X drawn from F. Examples are the Shannon Entropy, where $\mathcal{A} = \mathcal{P}(\mathcal{X})$ is the set of probabilities and $\ell(x, a) = -\log a(x)$, the variance with $\mathcal{A} = \mathcal{X}$ and $\ell(x, a) = \|x - a\|_2^2$, and the predictive V-entropy with $\mathcal{A} \subset \mathcal{P}(\mathcal{X})$ some subset of distributions and $\ell(x, a) = -\log a(x)$.

Using H-entropies, a new class of discrepancies based on optimal loss for decision tasks can be defined. For two distributions F_1 and F_2 on $\mathcal{X}$ and a continuous function $\phi : \mathbb{R}^2 \to \mathbb{R}$ such that $\phi(\theta, \lambda) > 0$ whenever $\theta + \lambda > 0$ and $\phi(0, 0) = 0$

$$D_\ell^\phi(F_1, F_2) = \phi\left(H_\ell\left(\frac{F_1 + F_2}{2}\right) - H_\ell(F_1), H_\ell\left(\frac{F_1 + F_2}{2}\right) - H_\ell(F_2)\right)$$

is a H-divergence. The term $H_\ell\left(\frac{F_1+F_2}{2}\right) - H_\ell(F_i)$ measures how much more difficult it is to minimize loss on the mixture distribution than on F_i, and ϕ maps the differences to a scalar divergence. Special cases of the general definition corresponding to particular H-entropies are the H-Jensen Shannon divergence, where $\phi(\theta, \lambda) = \frac{\theta+\lambda}{2}$ such that

$$D_\ell^{\mathrm{JS}}(F_1, F_2) = H_\ell\left(\frac{F_1 + F_2}{2}\right) - \frac{1}{2}\left(H_\ell(F_1) + H_\ell(F_2)\right),$$

and the H-Min divergence, where $\phi(\theta, \lambda) = \max(\theta, \lambda)$ such that

$$D_\ell^{\mathrm{Min}} = H_\ell\left(\frac{F_1 + F_2}{2}\right) - \min\left(H_\ell(F_1), H_\ell(F_2)\right).$$

Also, all squared MMD distances (Gretton et al. (2006); see Section 3.9.1) are H-divergences.

Each H-divergence is a probability divergence. Whether the H-divergence is strict depends on the choice of ℓ.

Given n i.i.d. samples $\{X_1, \ldots, X_n\}$ drawn from F_1 and $\{Y_1, \ldots, Y_n\}$ drawn from F_2, an empirical estimator is given by

$$\hat{D}_\ell^\phi(\hat{F}_1, \hat{F}_2) = \phi\left(\inf_a \frac{1}{n}\sum_{i=1}^n \ell\left(\tilde{Z}_i, a\right) - \inf_a \frac{1}{n}\sum_{i=1}^n \ell\left(X_i, a\right), \right.$$
$$\left. \inf_a \frac{1}{n}\sum_{i=1}^n \ell\left(\tilde{Z}_i, a\right) - \inf_a \frac{1}{n}\sum_{i=1}^n \ell\left(Y_i, a\right)\right),$$

with $\tilde{Z}_i = X_i b_i + Y_i(1 - b_i)$ and b_i i.i.d. uniformly sampled from $\{0, 1\}$ such that $\tilde{Z}_i$ is a sample from the mixture $(F_1 + F_2)/2$. This estimator is consistent under certain regularity assumptions.

The H-divergence can be used in a permutation test for $H_0 : F_1 = F_2$. The same holds for other divergences. A simulation study is performed by Zhao et al. (2021) with $\phi(\theta, \lambda) = \left(\frac{\theta^s+\lambda^s}{2}\right)^{1/s}$, $s > 1$ and $\ell(x, a)$ the negative log-likelihood of x under distribution a, $a \in \mathcal{A}$ with $\mathcal{A}$ a certain model family (mixture of

Gaussian distributions, Parzen density estimator, Variational Autoencoder). s and $\mathcal{A}$ are tuned on a training dataset and power is evaluated on test data. The test is compared to the ones based on the deep kernel MMD (Liu et al., 2020), the optimized kernel MMD (Gretton et al., 2012b) as well as the ME and SCF test (Chwialkowski et al., 2015; Jitkrittum et al., 2016) and the tests using optimized frequencies (Lopez-Paz and Oquab, 2017; Cheng and Cloninger, 2022). The test based on H-divergence shows the highest power under the same experimental setup that was used by Liu et al. (2020) in their experiments.

Distance for probability measures based on level sets Muñoz et al. (2012) consider a vector space of test functions where each distribution is seen as a continuous linear functional on this space $\mathcal{D}$. In this setting, they view a probability measure as a Schwartz distribution (generalized function) $F : \mathcal{D} \to \mathbb{R}$ by setting $F(\phi) = \langle F, \phi \rangle = \int \phi \, \mathrm{d}F = \int \phi(x) f(x) \, \mathrm{d}\mu(x) = \langle \phi, f \rangle$, where f is the density function w.r.t. the ambient measure μ. Then, two probability distributions viewed as linear functionals are the same (similar) if they behave identically (similarly) on all $\phi \in \mathcal{D}$. Thus, the distance between two distributions can be measured as the differences between functional evaluations for an appropriately chosen set of test functions. Muñoz et al. (2012) use indicator functions of α-level sets and define a distance of distributions by weighting the distances between the integrals of these functions w.r.t. the distributions. An α-level set is defined as $S_\alpha(f) = \{x \in \mathcal{X} | f(x) \geqslant \alpha\}$ such that $\mathbb{P}(S_\alpha(f)) = 1 - \nu$ for $\nu \in (0, 1)$. Consider sets of the type $A_i(F) = S_{\alpha_i}(f) \backslash S_{\alpha_{i+1}}(f), i = 1, \ldots, n-1$, for a sequence $0 < \alpha_1 < \cdots < \alpha_n < 1$. Then it holds for $n \to \infty$ that $A_i(F_1) = A_i(F_2)\ \forall i \Rightarrow F_1 = F_2$. Muñoz et al. (2012) consider $\phi_{1i} = \mathbb{1}_{A_i(F_1) \backslash A_i(F_2)}$ and $\phi_{2i} = \mathbb{1}_{A_i(F_2) \backslash A_i(F_1)}$ and $d_i(F_1, F_2) = |\langle F_1, \phi_{1i} \rangle - \langle F_2, \phi_{1i} \rangle| + |\langle F_1, \phi_{2i} \rangle - \langle F_2, \phi_{2i} \rangle|$ which is approximately equal to $\mu(A_i(F_1) \triangle A_i(F_2))$, where $A \triangle B = (A \backslash B) \cup (B \backslash A)$ denotes the symmetric difference of two sets. Then, the *weighted level-set distance* is defined as

$$d_{\boldsymbol{\alpha}}(F_1, F_2) = \sum_{i=1}^{n-1} \alpha_i \frac{\mu(A_i(F_1) \triangle A_i(F_2))}{\mu(A_i(F_1) \cup A_i(F_2))},$$

where $\boldsymbol{\alpha} = \{\alpha_{(i)}\}_1^n$ and μ is the ambient measure. An estimator for the weighted level-set distance is given by

$$\hat{d}_{\boldsymbol{\alpha}}(F_1, F_2) = \sum_{i=1}^{n-1} \alpha_i \frac{\# \left(\hat{A}_i(F_1) \triangle^S \hat{A}_i(F_2) \right)}{\# \left(\hat{A}_i(F_1) \cup \hat{A}_i(F_2) \right)},$$

where $\hat{A}_i(F) = \hat{S}_{\alpha_i}(f) \backslash \hat{S}_{\alpha_{i+1}}(f)$ are estimators of the sets A_i, $\# A$ denotes the number of points in A and $\triangle^S$ denotes the set estimate of the symmetric difference. $d_{\boldsymbol{\alpha}}(F_1, F_2)$ and its estimator $\hat{d}_{\boldsymbol{\alpha}}(F_1, F_2)$ are both semimetrics. Estimation of level sets by using a Support Neighbor Machine (Munoz and Moguerza, 2006) and estimation of the symmetric difference between sets by using a covering of the points with closed balls to circumvent that the intersection of the observed sets is empty are described in Muñoz et al. (2012).

Building up on this, Muñoz, Martos and González (2013) propose another weighting in the weighted level-set distance.

Dataset distance based on reproducing kernel Hilbert spaces (RKHS) Muñoz, Martos and González (2013) make use of a kernel to define a dataset distance. Consider a set A of points generated from a distribution F with sample space $\mathcal{X}$. Then a point $y \in 2^{\mathcal{X}}$ (power set of $\mathcal{X}$) is called *indistinguishable* from $x \in A$ with respect to F in the set A, $y \overset{A(F)}{=} x$, if $d(x,y) \leqslant r_A$, where $r_A = \min d(x_l, x_s), x_l, x_s \in A$, denotes the minimum resolution for the dataset A. The idea is to build kernel functions for two datasets A and B from distributions F_1 and F_2 such that the kernel takes the value one for points that are indistinguishable w.r.t. F_1 in A or w.r.t. F_2 in B, and zero otherwise. This is fulfilled for the *distributional indicator kernel*. Given datasets A sampled from F_1 and B sampled from F_2, define $K_{A,B} : \mathcal{X} \times \mathcal{X} \to [0,1]$ as

$$K_{A,B}(x,y) = f_{x,r_A,\gamma}(y) + f_{y,r_B,\gamma}(x) - f_{x,r_A,\gamma}(y) f_{y,r_B,\gamma}(x),$$

where the *smooth indicator functions* with center x for $r > 0$ and $\gamma > 0$ is defined as

$$f_{x,r,\gamma}(y) = \begin{cases} \exp\left(-\frac{1}{(\|x-y\|^\gamma - r^\gamma)^2} + \frac{1}{r^{2\gamma 2}}\right) & \text{if } \|x-y\| < r \\ 0 & \text{otherwise} \end{cases},$$

$r_A = \min d(x_l, x_s), x_l, x_s \in A$, $r_B = \min d(y_l, y_s), y_l, y_s \in B$, and γ is a shape parameter. With this, a kernel for datasets C and D in $2^{\mathcal{X}}$ can be defined as

$$K(C,D) = \sum_{x \in C} \sum_{y \in D} K_{A,B}(x,y).$$

For $C = A$ and $D = B$, $\mu_{K_{A,B}}(A \cap B) = K(A,B)$ can be interpreted as a measure for $A \cap B$ by counting the common points. With $\mu_{K_{A,B}}(A \cup B) = N = \#(A \cup B)$, it follows that $\mu_{K_{A,B}}(A \triangle B) = N - \mu_{K_{A,B}}(A \cap B)$. In general, $\mu_{K_{C,D}}(C \cap D) = K(C,D)$ can be interpreted as a measure for $C \cap D$ by counting the common points using $\overset{A(F_1)}{=}$ and $\overset{B(F_2)}{=}$ as equality operators, so taking the distance between C and D is conditioned to a resolution level (r_A and r_B) determined by A and B. Therefore the kernel distance between datasets is defined as

$$d_K(C,D) = 1 - \frac{K(C,D)}{N},$$

where $N = \#(C \cup D)$. This is a semimetric. For $C = A$ and $D = B$ and the sizes for both sets increasing, it holds $\mu_{K_{A,B}}(A \cap B) \overset{n_1,n_2 \to \infty}{\to} \mu(A \cap B)$ and $\mu_{K_{A,B}}(A \cup B) \overset{n_1,n_2 \to \infty}{\to} \mu(A \cup B)$, so the limit of the kernel distance is the Jaccard distance for datasets, that is $1 - \frac{\mu(A \cap B)}{\mu(A \cup B)}$. This divergence can also be interpreted as a kernel-based method.

3.7. Graph-based methods

In the following, methods using similarity graphs are presented. First, graph-based methods based on different similarity graphs are discussed. Then methods based on the important case of the nearest neighbor graph are shown.

3.7.1. General graph-based methods

Graph-based methods to compare distributions are especially popular in testing. Arias-Castro and Pelletier (2016) present a general framework of graph-based tests: recall that the pooled sample is defined as

$$\{Z_1, \ldots, Z_N\} = \{X_1, \ldots, X_{n_1}, Y_1, \ldots, Y_{n_2}\}.$$

Let $\mathcal{G}$ be a directed graph with this pooled sample as the node set and write $Z_i \to Z_j$ if there is an edge from Z_i to Z_j in $\mathcal{G}$. Reject H_0 for small values of

$$T_{\mathcal{G}}(Z) = \#\{i \leqslant n_1, j > n_1 : Z_i \to Z_j\} + \#\{i \leqslant n_1, j > n_1 : Z_j \to Z_i\},$$

that is the number of neighbors in the graph from different samples. Many of the methods presented below fall within this framework. For the K-nearest neighbor graph as $\mathcal{G}$ under the assumption of distinct values in the pooled sample, the test by Schilling (1986) is given which is a special case of the general approach of Friedman and Steppel (1973). The minimum spanning tree starting with the complete graph weighted by Euclidean distances on the other hand results in the multivariate runs test of Friedman and Rafsky (1979). A minimum distance matching gives the test by Rosenbaum (2005).

Mukhopadhyay and Wang (2020a) also try to generalize different graph-based tests into a single framework. They note that the tests by Weiss (1960), Friedman and Rafsky (1979), Chen and Friedman (2017), Chen, Chen and Su (2018), Rosenbaum (2005), and Biswas, Mukhopadhyay and Ghosh (2014) have the following steps in common:

1. Construct a weighted undirected graph $\mathcal{G}$ based on pairwise Euclidean distances on the pooled sample.
2. Compute a subgraph $\mathcal{G}^*$ that contains a certain optimal subset of edges (e.g. shortest Hamiltonian path).
3. Compute cross-match statistics by counting the number of edges between samples from two different populations.

All of the tests mentioned will be described in more detail below.

Tests based on minimal spanning trees (Friedman-Rafsky test) One of the first and best-known graph-based tests is the multivariate runs test by Friedman and Rafsky (1979). It generalizes the Wald-Wolfowitz runs test to the multivariate domain based on a minimal spanning tree of pooled sample points. Henze and Penrose (1999) proved later that the test is asymptotically

distribution-free and universally consistent. Chen, Dou and Qiao (2013) on the other hand observe that power decreases with the imbalance of sample sizes in their simulations. Chen, Chen and Su (2018) investigate this problem in more detail. Biswas and Ghosh (2014) highlight that the test is rotation invariant and invariant under location change and homogeneous scale transformation and that it can be used even when the dimension of data is larger than the sample size, but they also derive sufficient conditions for failure for $p \to \infty$. Biswas, Mukhopadhyay and Ghosh (2014) give sufficient conditions for failure where power converges to 0 as $p \to \infty$ and criticize that the test is not distribution-free for finite samples. Sarkar, Biswas and Ghosh (2020) again show situations where power is very low and formal conditions under which power decreases to 0 for increasing p. Chen and Friedman (2017) observe that in practice (simulations), the test has low or even no power for scale alternatives when the dimension is moderate to high unless the sample size is "astronomical" due to the curse of dimensionality.

Friedman and Rafsky (1979) propose a second test in their paper that is a generalization of the KS test to the multivariate domain based on a minimal spanning tree of pooled sample points. The test needs a reasonable distance measure between points. An approximation of the null distribution is used for testing. Friedman and Rafsky (1979) themselves show that the test has either no power for scale-only or no power for location-only alternatives. In addition to that, Chen and Zhang (2013) demonstrate that the test does not work well on categorical data due to ties.

Both tests of Friedman and Rafsky (1979) are implemented in the `R` package `GSAR` (Rahmatallah et al., 2017) and `gTests` (Chen and Zhang, 2017). Note that in the `GSAR` implementation the test statistic is standardized by empirical mean and standard deviation instead of the theoretical values under H_0 as in the original definition.

Tests based on optimal non-bipartite matching (Rosenbaum's cross-match test) Another well-known test is the cross-match test by Rosenbaum (2005). Here, an optimal non-bipartite matching is formed in the pooled sample based on the inter-point distances. The number of pairs containing one observation from the first distribution and one from the second is considered as a test statistic. Consistency and the asymptotic distribution of the test statistic are shown under the assumption of discrete distributions with finite support. The computational cost for finding an optimal non-bipartite matching of N subjects is $\mathcal{O}(N^3)$. In case of an odd pooled sample size N, one observation needs to be discarded. The test is not applicable for partially ordered responses. Chen and Zhang (2013) note that the test does not work well on categorical data due to ties and show via simulation that power decreases with the imbalance of sample sizes. In general, different distance measures can be used for the optimal non-bipartite matching. Biswas and Ghosh (2014) note that the test can be used even when the dimension of data is larger than the sample size if the Euclidean distance is used. Due to the high computational cost, a greedy algorithm that reduces the cost to $\mathcal{O}(N^2)$ might be employed, but Huang and Huo (2017) point

out that solving with the greedy heuristic does not guarantee finding the optimum. Furthermore, Sarkar, Biswas and Ghosh (2020) show situations where power is very low. On the other hand, Deb and Sen (2021) state that the test by Rosenbaum (2005) is one of two tests for the multivariate two-sample problem that is exactly distribution-free, computationally feasible, and consistent against all alternatives. Consistency against all fixed alternatives is shown by Arias-Castro and Pelletier (2016). For the consistency of the general cross-match statistic, they need the assumptions that densities w.r.t. Lebesgue measure of both distributions exist, that the sample sizes are comparable in the sense that their ratio converges to a fixed constant in $(0, 1)$, the assumption of bounded out- and in-degree in the graph as well as that the out-degree is essentially constant and long edges are essentially absent and that the dimension is constant. They note that the graph-based setting exhibits a typical curse of dimensionality although literature is silent on that topic. The required conditions cover the minimum spanning tree used by Friedman and Rafsky (1979) (Henze and Penrose, 1999), nearest-neighbor graphs (Schilling, 1986) and general matchings (shown here). Additionally, Arias-Castro and Pelletier (2016) show that the null distribution as studied by Rosenbaum (2005) and Heller et al. (2010) is available in closed form and coincides with the permutation distribution. The test is implemented in the `R` package `crossmatch` (Heller, Small and Rosenbaum, 2012).

Extensions of Friedman-Rafsky and Rosenbaum tests Chen and Zhang (2013) propose a graph-based test for categorical data with a large number of categories and a sparsely populated contingency table that extends the Friedman-Rafsky and Rosenbaum tests to categorical data. They assume that a distance matrix is given on the set of categories and that there are not many ties in these distances. In the presence of ties, the resulting graphs are not unique anymore and the number of possible graphs grows fast with the number of ties. Their tests work by either averaging over all optimal graphs for a certain graph type (e.g. MST) or by taking the union of all optimal graphs. An analytic form and an asymptotic form for both types of tests are proposed. The analytic form of their first proposed test requires the enumeration of all MSTs on categories which might not be computationally feasible but can be bypassed by assuming that instead of the distance matrix, the similarity is directly represented by a graph with the categories as nodes. For the asymptotic normality of the test statistics, assumptions on cell counts and graph structure are required and the number of categories has to go to infinity. The computational cost of the test is $\mathcal{O}(K^2)$ with K number of categories for the Rosenbaum version resp. $\mathcal{O}(M)$ with M number of minimum spanning trees on categories for the Friedman-Rafsky version or $\mathcal{O}(K^2)$ for the bypassed version. The tests are implemented in the `R` package `gTests` (Chen and Zhang, 2017).

Tests based on the shortest Hamilton path Biswas, Mukhopadhyay and Ghosh (2014) present a multivariate generalization of two-sample run tests based on the shortest Hamilton path using Euclidean distances that is applicable to

high-dimensional data and small sample sizes. It is also invariant under location change, rotation, and homogeneous scale transformations and distribution-free in finite-sample situations. Finding the shortest Hamilton path is NP-complete for complete graphs so instead, a heuristic search algorithm is used that does not always yield the optimum. Under several assumptions, consistency for $p \to \infty$ is shown for the test, but Sarkar, Biswas and Ghosh (2020) show situations where power is very low. They also note that the computational complexity is still $\mathcal{O}(N^2 \log N)$ with the heuristic method based on Kruskal's algorithm. Deb and Sen (2021) therefore conclude that the test is extremely expensive to compute and possibly not applicable even for moderate sample sizes.

Tests based on orthogonal perfect matchings, minimum spanning tree or nearest neighbors Petrie (2016) presents tests based on new graphs using orthogonal perfect matchings as well as on minimum spanning trees or nearest neighbors. The construction for the new graph type works by first finding the optimal perfect matching on the data, then finding the optimal perfect matching without the edges from the first matching, and so on until the Kth matching is reached. The graph is then given as the union of these matchings. $K \approxeq 0.15N$ is suggested as a heuristic choice. The test is intended for continuous data. It can be used to compare multiple samples. An asymptotic normal test is proposed that is claimed to have good HDLSS performance. Mukherjee et al. (2022) on the other hand observe that it tends to have low power as the dimension and/ or number of samples to be compared increase and point out that its mathematical properties have not been investigated. The test for Euclidean data and using the optimal non-bipartite matching as a graph is implemented in the `R` package `multicross` (Agarwal, Bhattacharya and Zhang, 2020).

Test based on similarity graphs Chen and Friedman (2017) propose a new test based on a similarity graph constructed over the pooled sample that has higher power for differences in location as well as in scale in contrast to former tests that often only have high power for one of those. Consistency is only shown for continuous distributions that differ on a set of positive measures. Additionally, certain conditions for the similarity graph are required that are fulfilled by a K-MST with $K = \mathcal{O}(1)$. The new test statistic is given as the quadratic form of the vector of the numbers of edges connecting observations within the same sample for both samples centered with its expectation under the permutation null distribution and the inverse covariance matrix of these numbers under the permutation null distribution. The test statistic cannot be determined if all nodes in the chosen graph have the same degree or if the graph is star-shaped since in these cases the covariance matrix is singular. Chen and Friedman (2017) recommend not to use the test if the graph is very close to one of these cases since then the inversion of the covariance matrix is already ill-conditioned. The test is implemented in the package `gTests` (Chen and Zhang, 2017)

Extension of edge-count tests for imbalanced data Chen, Chen and Su (2018) aim to improve edge-count tests for unequal sample sizes by weighting. They show consistency only for continuous distributions and if the graph is the MST based on Euclidean distance. The test is also only more powerful than former edge-count tests under locational alternatives, not under general alternatives. The test is implemented in the `R` package `gTests` (Chen and Zhang, 2017). Pan et al. (2018) observes that the efficiency of the test is limited by the choice of the number of neighbors.

Extension of edge-count tests for categorical data Zhang and Chen (2019) extend the generalized edge-count test of Chen and Friedman (2017) and the weighted edge-count test of Chen, Chen and Su (2018) for categorical data following the approach of Chen and Zhang (2013), i.e. using either the union of all optimal graphs or averaging over the edge-counts of all optimal graphs. Additionally, they propose a new group of graph-based tests for categorical data, the *extended max-type edge-count tests*. These consist of two components. First, the weighted edge-count statistic of the extension of the test of Chen, Chen and Su (2018) standardized by its mean and variance under H_0 and multiplied with a factor κ is considered. Second, the absolute difference of the edge counts of points within the first sample that are connected by an edge and points in the second sample that are connected by an edge, again standardized by its mean and variance under H_0, is taken into account. The test statistic of the extended max-type edge-count test is then given by the maximum of these two. Again, two versions are proposed. The first version is based on the union of all optimal graphs and the second version is based on averaging over all optimal graphs. The resulting tests are claimed to be effective for both location and scale alternatives. Without prior knowledge about the difference between the distributions, a choice of $\kappa \in \{1.31, 1.14, 1\}$ is recommended based on a small simulation study. Asymptotic null distributions are derived for all proposed statistics under several conditions on the similarity graph and the class distributions. All asymptotic as well as permutation versions of the tests are implemented in the `R` package `gTests` (Chen and Zhang, 2017). For numerical data, the newly proposed max-type test is also implemented in this package.

Extensions of nearest neighbor, Rosenbaum, and Friedman-Rafsky test Sarkar, Biswas and Ghosh (2020) modify graph-based tests to overcome weak performance caused by distance concentration to achieve higher power for high-dimensional data with low sample sizes (HDLSS). Under certain assumptions including uniformly bounded fourth moments, the order of correlations between inter-point distances and the convergence of mean distances and traces of covariance matrices for $p \to \infty$, consistency is shown under $p \to \infty$ for fixed sample size for the modified tests of Biswas, Mukhopadhyay and Ghosh (2014) and Rosenbaum (2005) and under additional assumptions on sample size also for the nearest neighbor test of Henze (1988) and Schilling (1986) as well as the MST-run test of Friedman and Rafsky (1979). Depending on the choice of distance, the computation of distances between two points has a cost of $\mathcal{O}(pN)$

compared to $\mathcal{O}(p)$ for Euclidean distance or other distances.

Power of tests Bhattacharya (2020) presents results regarding the limiting distribution under general alternatives as well as power and consistency for Friedman-Rafsky and nearest neighbor tests. The results are based on several different assumptions.

Test based on orthonormal polynomials A new class of distribution-free tests for the high-dimensional k-sample problem based on new nonparametric tools and connections with spectral graph theory is proposed by Mukhopadhyay and Wang (2020a). Their motivation is that classical multivariate rank-based k-sample tests like Puri et al. (1971) or Oja and Randles (2004) are not applicable for $p > N$. They demand several desirable properties of tests:

1. should be robust and not unduly influenced by outliers (current methods even perform poorly for datasets that are contaminated by only a small percentage of outliers)
2. should allow testing beyond location-scale alternatives
3. valid for a combination of discrete and continuous covariates
4. should provide insight into why the hypothesis was rejected
5. should work for k-sample problem (all former tests only work for two-sample).

For their test, a nonparametrically designed set of orthogonal functions (LP polynomials) is obtained by orthonormalizing a set of functions constructed as orthonormal polynomials of mid-distribution transforms. These are used for the construction of a polynomial kernel of degree 2 that encodes the similarity between two p-dimensional data points in the LP-transformed domain. The values of the kernel Gram matrix are then used as weights on a graph with the pooled sample as vertices. The idea is to cluster points for the graph into k groups that have higher connectivity and compare how closely related this clustering is to the true memberships to the k distributions. Then testing for homogeneity becomes a problem of testing independence which can be accomplished by determining whether all of the LP comeans are zero. The test statistic has an asymptotic $\chi^2_{(k-1)^2}$ distribution. It has to be explicitly chosen for which moments to test for equality. Implicitly this requires the corresponding moments of the distributions to exist. The test is implemented in the `LPKsample` package (Mukhopadhyay and Wang, 2020b) in `R`.

Extension of Rosenbaum test for k-sample problem Mukherjee et al. (2022) propose a generalization of the test by Rosenbaum (2005) to the k-sample problem that is exactly distribution-free. It can be applied if inter-point distances are well-defined. If distributions have densities w.r.t. Lebesgue measure on $\mathbb{R}^p$, the test is universally consistent. As for the original test by Rosenbaum (2005), the pooled sample size N has to be even, or one observation has to be discarded. For the test, the optimal non-bipartite matching on the pooled sample is calculated. Then a matrix of cross-match counts is constructed whose

entries are given by the number of matches with one observation coming from one sample and the other from another sample for each pair of samples. The test statistic is given as the Mahalanobis distance of the observed cross-counts under the null hypothesis. The test was implemented in the `R` package `multicross` (Agarwal, Bhattacharya and Zhang, 2020).

Tests for the k-sample problem for high-dimensional and non-Euclidean data Song and Chen (2022a) propose three new tests for the k-sample problem, especially for high-dimensional and non-Euclidean data. Their main idea is to use not only the between-sample edges of a similarity graph on the pooled sample, i.e. the edges connecting points from different samples, but also the within-sample edges, i.e. the edges connecting points from the same sample, to use as much information as possible. Let R^W denote the vector containing the numbers of within-sample edges for each of the k samples and R^B denote the vector containing the numbers of between-sample edges for all $k(k-1)$ pairs of different samples. Then the first test statistic is given by

$$\begin{aligned} S &= S^W + S^B \\ S^W &= \left(R^W - \mathbb{E}(R^W)\right)^T \Sigma_W^{-1} \left(R^W - \mathbb{E}(R^W)\right) \\ S^B &= \left(R^B - \mathbb{E}(R^B)\right)^T \Sigma_B^{-1} \left(R^B - \mathbb{E}(R^B)\right), \end{aligned}$$

where $\mathbb{E}$ and Σ denote the expectation and covariance matrix under the permutation null hypothesis. The second test statistic is based on the vector R^A of all linearly independent numbers of edges between and within samples, i.e. all numbers of edges between all pairs of samples including the pairs of a sample with itself except for the pair of the sample $(k-1)$ with the kth sample. The test statistic is then defined as

$$S^A = \left(R^A - \mathbb{E}(R^A)\right)^T \Sigma_A^{-1} \left(R^A - \mathbb{E}(R^A)\right),$$

where again $\mathbb{E}$ and Σ denote the expectation and covariance matrix under the permutation null hypothesis. While Σ_W is shown to be always invertible, no such proof exists for Σ_B and Σ_A. Therefore, Song and Chen (2022a) suggest checking the invertibility numerically before applying the test and using a generalized inverse if necessary. Formulas for the expectations and covariance matrices under the permutation null are given in Theorem 2.1 of Song and Chen (2022a). Moreover, Song and Chen (2022a) show that under some assumptions on the similarity graph that are fulfilled by a K-MST with $K = \mathcal{O}(1)$, $S^W \to \chi^2_k$, $S^B \to \chi^2_b$, $S^A \to \chi^2_a$ asymptotically, where $b = \text{rank}(\Sigma_B)$ and $a = \text{rank}(\Sigma_A)$. The asymptotic distribution of S is more complicated and hard to compute in practice, therefore it is suggested to use a fast test instead. This fast test combines tests using S^W and S^B and takes the Bonferroni-adjusted p-value of both these tests. Alternatively, a permutation test can be performed. Consistency of the asymptotic tests based on S and S^A against all alternatives is shown in the multivariate setting for the K-MST and under the condition that Σ_A is invertible. The tests are implemented in the `R` package `gTestsMulti` (Song and Chen, 2022b).

Graph-based tests based on denser graphs Zhu and Chen (2024) present a review of graph-based tests as well as new theoretical results based on less strict assumptions. Their motivation is that the assumptions made on the graphs e.g. by Friedman and Rafsky (1979), Chen and Friedman (2017), and Chen, Chen and Su (2018) to show asymptotic distributions of the test statistics are quite strict and often not fulfilled in practice, especially for denser graphs. However, empirically an improved performance of tests based on denser graphs, e.g. the 5-MST instead of the MST, was observed. Therefore, constructing tests based on denser graphs is promising for improving the power. Zhu and Chen (2024) derive less strict sufficient conditions under which the asymptotic null distributions of the original, weighted, generalized, and max-type edge count statistic hold. These allow for using much denser graphs than the conditions derived before. Additionally, simulations on the newly derived assumptions are presented.

3.7.2. Methods based on nearest neighbors

An important subgroup of graph-based tests are nearest-neighbor type tests. Chen and Zhang (2013) claim that they do not work well for categorical data in general.

Weiss test based on spheres One of the first approaches for the multivariate two-sample problem goes back to Weiss (1960). The procedure presented there needs the assumption that for both distributions, piecewise continuous and bounded densities exist. The test statistic is given by the proportion of points from the first sample where no point of the second sample is contained in the sphere around the respective point from the first sample with radius 1/2 of the distance to its nearest neighbor from the first sample. This yields a multivariate analog of the Wald-Wolfowitz run test. The test is not distribution-free, but the calculation of its critical value is possible under assumptions on densities under H_0. The test statistic is invariant under translations and rotations of space or under linear stretching of each of the axes by the same factor. Disadvantages of the procedure are that the test statistic lacks symmetry (roles of the first and second sample not interchangeable) and as pointed out by Henze (1988) the test lacks proof of consistency.

Nearest neighbor test of Friedman and Steppel The idea of nearest-neighbor type tests dates back to Friedman and Steppel (1973) and was motivated by assessing the influence of different features on a target variable by splitting the feature dataset according to values of the target and comparing if the subsets differ in their distributions. It is assumed that the distributions have existing density functions. The K nearest neighbors in the pooled sample that originate from the first sample are counted and the distribution of these frequencies is compared to that expected under the null hypothesis by a permutation procedure. One way to do this is to compare the frequency distribution of K nearest neighbors that originate from the first sample in the

first sample to that in the second sample. Under the null, these are expected to be equal. Friedman and Steppel (1973) suggest performing a test based on a t-statistic that compares the mean frequencies from both samples or alternatively use another test for comparison of univariate distributions. This test may be asymptotically optimal, but in finite samples is relatively insensitive to differences in scale between the two multivariate samples. An alternative to fix this is to compare frequency distributions (or the distribution of summed frequencies from both samples) directly with that expected under the null via a χ^2 goodness-of-fit test. The null distribution is in general hard to derive but can be approximated by binomial distribution $\text{Bin}(K, n_1/N)$. The choice of K and of the metric to determine the nearest neighbors is left open. For consistency, K should be a function of the total sample size N that goes to ∞ for $N \to \infty$ such that $K(N)/N \to 0$ for $N \to \infty$. More important than the choice of K is the choice of the metric, the features to use, and their scaling. Friedman and Steppel (1973) recommend scaling the data by the inverse covariance matrix if no prior knowledge is given. More features only improve the power of the test if they contain information concerning the hypothesis under test, otherwise adding features decreases power. Friedman and Steppel (1973) give no general recommendation for the choice of metric (Minkowski q-metrics for $q = 1, 2, \infty$ are considered) and list different algorithms for the nearest neighbor calculation giving $\mathcal{O}(2^p N \log_2 N)$, $\mathcal{O}(p[Kp\Gamma(p/2)/2]^{1/p} N^{2-1/p})$ or for brute force $\mathcal{O}(pN^2)$ cost for the computation.

Nearest neighbor test of Schilling and Henze Schilling (1986) developed a two-sample test based on nearest neighbor type coincidences and Henze (1988) proved its asymptotic properties. Their test is probably the best-known nearest-neighbor test. It is based on the ideas of Friedman and Steppel (1973) and Rogers (1978). The test statistic is the proportion of all K nearest neighbor comparisons based on the (Euclidean) distance in which a point and its neighbor belong to the same sample. A scaled version of this test statistic is shown to be asymptotically normal, which motivates an asymptotic test. The method is in general only applicable for continuous distributions since then the probability of ties in the calculation of the nearest neighbors is zero. Schilling (1986) originally proposed to use randomization of the ranks if distances are tied. The test is shown to be consistent against all alternatives. Weighted versions of the test statistic are given by Schilling (1986). The generalization to the multisample problem is presented by Henze (1988). The determination of the number K of nearest neighbors to consider is left open. Biswas, Mukhopadhyay and Ghosh (2014) state that the test is not exactly distribution-free and Sarkar, Biswas and Ghosh (2020) show situations where its power is very low and formal conditions under which power decreases to zero for increasing p. Chen, Dou and Qiao (2013) show via simulation that the test's power decreases with the imbalance of the sample sizes. Aslan and Zech (2005a) reported that according to a private communication with Henze, there is no recipe for how to choose the number of neighbors. Biswas and Ghosh (2014) mention that the test statistic is rotation invariant and invariant under location change and under homogeneous scale

transformation and can be used even when the dimension of the data is larger than the sample size. They also derive sufficient conditions when power decreases to zero for $p \to \infty$.
Henze and Voigt (1992) derive sufficient conditions for almost sure convergence of a class of sequences of symmetric test statistics for the k-sample problem that includes, e.g. the test statistics of Schilling (1986) and Henze (1988). They assume absolutely continuous Lebesgue densities.

Barakat, Quade and Salama (1996) further generalize Schilling's nearest neighbor test to circumvent choosing the number of nearest neighbors. Their test statistic is the sum of edge counts for all values of K for the K-nearest neighbor graph. Alternatively, it can be seen as a term relying on the sample sizes and a quantity that can be interpreted as follows. First, randomly an observation x is selected from the pooled sample, then one observation x_1 is randomly selected from the sample to which the first selected observation belongs, and one observation x_2 from the other sample. There are $n_1 n_2 (N-2)$ such choices, i.e. sets of such three observations. Then the number of cases for which the first selected observation x is closer to the observation x_1 from the same sample than to the observation x_2 from the other sample is calculated, and a correction term depending on the sample sizes is added. The resulting test is equivalent to a sum of Wilcoxon rank sums. It requires samples in the Euclidean space $\mathbb{R}^p$ and it is assumed that there are no ties in ranking w.r.t. to nearness.

Nearest neighbor test for categorical data Nettleton and Banerjee (2001) propose a test for the two or k-sample problem with categorical components. A function that gives the distance between any two data vectors is defined and the number of edges in a nearest-neighbor graph that connect observations from different samples is counted. The test works by adding up values of distance functions over dimensions and calculating the number of edges that link data points from different groups based on a nearest-neighbor graph of the pooled sample. The p-value of the test can be determined both by permutation testing or by an asymptotic test. The distance function that maps $\{0, \ldots, K-1\}^2$ with K denoting the number of classes to $\mathbb{R}$ is chosen depending on the application (e.g. Hamming distance for binary data) and there are no clear general recommendations for this choice. According to Nettleton and Banerjee (2001), the procedure needs a "few minutes using a personal computer" to calculate an estimate of the conditional p-value. It might therefore not be feasible if a large number of tests needs to be performed.

Nearest neighbor test for continuous data (Hall and Tajvidi test) Hall and Tajvidi (2002) propose a permutation test based on ranking the pairwise distances between data points. Their test is only applicable for continuous distributions with identical support. The distance measure should be symmetric and does not have to satisfy the triangle inequality. Similar to nearest neighbor tests, the number of j nearest neighbors in the pooled sample that belong to the same sample as the point under consideration are determined for both samples.

The test statistic then is a weighted sum of powers of the absolute deviations of these numbers from their expectations under H_0 over all sample points and all possible values of j. The choice of the power and the weight functions is left open. Weight functions of the form $w_i(j) = 1$, $w_i(j) = j$, and $w_i(j) = n_i + 1 - j$ are proposed. For theoretic results, a weight function is required that converges to a non-degenerate function when viewed as a function of j/n_2. The test can distinguish between local alternatives that are distant $n_2^{-1/2}$ from the null hypothesis if the distance is chosen as the Euclidean distance and both distributions have continuous densities. According to Biswas, Mukhopadhyay and Ghosh (2014), the test is not distribution-free. Biswas and Ghosh (2014) mention that the test statistic is rotation invariant and can be used even when the dimension of data is larger than the sample size. Montero-Manso and Vilar (2019) claim that the test is valid for infinite dimensional Euclidean spaces since it can take any dissimilarity function as distance.

Extension of Schilling-Henze test for unbalanced sample sizes Chen, Dou and Qiao (2013) use a test based on the nearest neighbor method of Schilling (1986) and subsampling to improve the unsatisfactory performance of two-sample tests when sample sizes are unbalanced. The finite sample distribution of the resulting test statistic is unknown but asymptotic and permutation approaches are presented. Consistency is shown when the ratio of sample sizes either goes to a finite limit or tends to infinity. For this, it is assumed that the distributions are absolutely continuous with respect to the Lebesgue measure and that there are no ties for the identification of nearest neighbors. The size of the subsample from the larger of the two samples needs to be chosen to calculate the test statistic. Based on simulations, a subsample size equal to the size of the smaller sample is recommended.

Nearest neighbor test for high dimension low sample size setting Mondal, Biswas and Ghosh (2015) propose a new multivariate two-sample test based on nearest neighbor type coincidences suitable also for the high dimension low sample size (HDLSS) regime that has higher power than the test of Schilling (1986) and Henze (1988) in certain situations. The test statistic modifies the one of Schilling (1986) and Henze (1988) by subtracting the expected values under H_0 from both proportions of nearest neighbors from the same sample and taking either the absolute value of this difference or squaring it. Under similar conditions as in other papers for the HDLSS regime, consistency for fixed sample size and $p \to \infty$ is shown for both variants. Also, conditions are derived under which the tests are not consistent for increasing dimensionality. The tests are shown to be asymptotically distribution-free for $N \to \infty$, but a permutation procedure is used in practice where H_0 is rejected for large values of the test statistics. Moreover, consistency for fixed p and $N \to \infty$ is shown for distributions with continuous densities. The choice of the number of nearest neighbors to consider is left open. Mondal, Biswas and Ghosh (2015) consider $K = 3$ neighbors in all examples and applications.

3.8. Methods based on inter-point distances

Many methods are based on analyzing the distributions of inter-point distances in and between the samples. A theoretical justification for methods based on inter-point comparisons based on a univariate function (e.g. a distance) is given by Maa, Pearl and Bartoszyński (1996). They show that equality of distributions of in-sample comparisons (i.e. $\|X - X'\|$ and $\|Y - Y'\|$) together with equality of distributions of between-sample comparisons (i.e. $\|X - Y\|$) between points is equivalent to the equality of distributions of the samples. This holds in general for discrete distributions. For the continuous case, some restrictions on the density function are needed, including the existence of expectations and a second condition that is for example fulfilled if one of the densities is bounded or continuous.

The advantages of using tests based on inter-point distances according to Montero-Manso and Vilar (2019) are that

- it reduces the dimension of the problem,
- the use is not limited to dealing with continuous data,
- tests can be conducted whenever distances are available even though the original observations are not accessible,
- the versatility to choose a proper distance facilitates the introduction of prior domain knowledge,
- it is intuitively expected that employing a suitable distance should increase the test power.

3.8.1. Energy statistic

The most popular statistic based on inter-point distances is the so-called energy statistic. It was proposed by Zech and Aslan (2003) and by Aslan and Zech (2005b), where the concept of statistical energy of statistical distributions similar to electric charge distributions is introduced, which was later on also proposed by Székely and Rizzo (2004). Independent from that, Baringhaus and Franz (2004) introduced a test based on the difference of the sum of all Euclidean distances between random vectors belonging to different samples and 1/2 of both sums of distances between random vectors belonging to the same sample, which they call the Cramér test. Its test statistic is equal to the energy statistic. The Cramér test is not distribution-free and needs the assumption that expectations of both distributions exist. It is shown to be consistent against any fixed alternative $F_1 \neq F_2$ with finite expectations. Convergence of a Bootstrap version of the test is shown as well. The test is invariant w.r.t. orthogonal linear transformations. It is implemented in the `R` package `cramer` (Franz, 2019). According to Sarkar and Ghosh (2018), the Cramér test needs the two distributions to differ in their locations or average variances to perform well in the HDLSS setup. Biswas and Ghosh (2014) note that the test is rotation invariant and invariant under location changes and homogeneous scale transformations, it can be used even when the dimension of data is larger than the sample size, and

under conditions similar to those made in Biswas and Ghosh (2014), a similar consistency result can be shown for the Cramér test as for the test introduced by Biswas and Ghosh (2014). On the other hand, they demonstrate situations in which the test fails to detect differences in distributions.

A comprehensive review of the literature on the energy statistic and its applications is given in Székely and Rizzo (2017). We focus on the results for the two-sample situation here, although applications of the energy statistic also include for example one-sample goodness-of-fit tests, clustering, or testing for independence (Székely and Rizzo, 2017). We extend the presentation of the two-sample energy statistic by Székely and Rizzo (2017) using the references therein as well as additional literature.

In the following, we present the energy statistic and its application in two-sample testing according to Székely and Rizzo (2004). Aslan and Zech (2005b) give a slightly more general form of the statistic since they leave the choice of the distance function between points open (for discussion of properties, see below) while Székely and Rizzo (2004) define the statistic using the Euclidean distance. They propose a distribution-free test for the equality of two or more multivariate distributions. The approximate permutation test uses Euclidean distances between elements of the samples. Its computational complexity is independent of the dimension and the number of datasets. The test is motivated by the lack of distribution-free extensions of approaches for the two-sample problem based on comparing EDFs (e.g. Kolmogorov-Smirnov and Cramér-von-Mises test) to the multivariate case as well as the lack of extensions of tests for the multivariate problem relying on ML to the general k-sample problem due to the distributional assumptions.

The test statistic of Székely and Rizzo (2004) relies on the e-distance between finite sets: The e-distance $e(\mathcal{X}, \mathcal{Y})$ between disjoint nonempty subsets $\mathcal{X} = \{X_1, \ldots, X_{n_1}\}$ and $\mathcal{Y} = \{Y_1, \ldots, Y_{n_2}\}$ of $\mathbb{R}^p$ is defined as

$$\begin{aligned} e(\mathcal{X}, \mathcal{Y}) = & \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \|X_i - Y_j\|_2 \\ & - \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} \|X_i - X_j\|_2 - \frac{1}{n_2^2} \sum_{i=1}^{n_2} \sum_{j=1}^{n_2} \|Y_i - Y_j\|_2, \end{aligned}$$

where $\|\cdot\|_2$ is the Euclidean norm. Its population equivalent is given by

$$\mathcal{E}(X, Y) = 2\,\mathbb{E}[\|X - Y\|_2] - \mathbb{E}[\|X - X'\|_2] - \mathbb{E}[\|Y - Y'\|_2],$$

where X' and Y' denote independent copies of X and Y, respectively.

Given $\mathcal{X}_1, \ldots, \mathcal{X}_k$, $k \geqslant 2$ independent random samples of random vectors in $\mathbb{R}^p$ with sizes $n_1, \ldots, n_k$, let $N = \sum_{i=1}^{k} n_i$. Denote the e-distance of each pair of samples $(\mathcal{X}_i, \mathcal{X}_j)$, $i \neq j$, by $\mathcal{E}_{n_i, n_j}(\mathcal{X}_i, \mathcal{X}_j) = e(\mathcal{X}_i, \mathcal{X}_j)$. Then the k-sample test statistic is given by the sum of the e-distances for all $k(k-1)/2$ pairs of samples:

$$T_{\text{Energy}} = \sum_{1 \leqslant i < j \leqslant k} \frac{n_1 n_2}{n_1 + n_2} \mathcal{E}_{n_i, n_j}(\mathcal{X}_i, \mathcal{X}_j) = \sum_{1 \leqslant i < j \leqslant k} \frac{n_1 n_2}{n_1 + n_2} e(\mathcal{X}_i, \mathcal{X}_j).$$

Large values of the test statistic lead to rejection of the null hypothesis. To obtain the (approximate) null distribution of the test statistic a permutation test is performed by drawing B Bootstrap samples of the pooled sample and partitioning each Bootstrap sample into sets of the same sizes as $n_i, i = 1, \ldots, k$. For each Bootstrap sample, the test statistic for these new sets is calculated and the null hypothesis is rejected if the observed value of the test statistic is larger than the $(1 - \alpha)$-quantile of the empirical distribution of test statistics from the Bootstrap samples. The test is implemented in the `R` (R Core Team, 2021) package `energy` (Rizzo and Szekely, 2022).
According to Székely and Rizzo (2017), the energy distance is invariant w.r.t. distance-preserving transformations (e.g. translation, reflection, angle-preserving rotation of coordinate axes) of data, i.e. rigid motion invariant. Moreover, it is scale invariant. It can be seen as a weighted L^2 distance between characteristic functions and the specific weight function is the only solution for such a weighted L^2 distance between characteristic functions so that the distance is rotation and scale invariant (under some technical assumptions). For equal sample sizes, the sample energy distance is the square of a metric on the sample space.

In Székely and Rizzo (2013) a discussion and illustration of the theory and application of energy statistics are given. A generalization of the energy statistic is given for which a continuous, monotonic decreasing function of the Euclidean distance between points needs to be chosen. Székely and Rizzo (2013) choose $-\log$ such that the test is scale invariant. Moreover, they recommend standardizing all variables with the mean and standard deviation of the pooled sample to avoid a single variable dominating the value of the test statistic.

Another generalization of the energy statistic is given by taking each distance to the power of α, $\alpha \in (0, 2]$. For $0 < \alpha < 2$, it still holds that the statistic is nonnegative with equality to zero if and only if both distributions are equal. The latter property does not hold in the case of $\alpha = 2$ (Székely and Rizzo, 2017). When using a different metric than the Euclidean metric, non-negativity of the resulting energy statistic is equivalent to the condition that the metric space in which the random variables take their values has negative type while the property that the statistic is equal to zero if and only if the distributions are equal is equivalent to the condition that that metric space has strong negative type. This holds e.g. for Euclidean spaces and separable Hilbert spaces (Székely and Rizzo, 2017). A metric is said to be of negative type if there exists a mapping $f : \mathcal{X} \to L^2$ such that $d(x, y) = \|f(x) - f(y)\|_2^2$ for every $x, y \in \mathcal{X}$.

Li (2018) derives the asymptotic null distribution of the energy statistic and shows under some assumptions that the test is more powerful for location than for scale differences.

Chakraborty and Zhang (2021) show that energy distance based on the usual Euclidean distance cannot completely characterize the homogeneity of two high-dimensional distributions, but only detects equality of means and the traces of covariance matrices in the high-dimensional setup. They criticize the energy distance based on Euclidean distances and define a new test with complexity linear in the dimension of the data that is capable of detecting homogeneity between the low-dimensional marginal distributions in the high-dimensional setup. They

generalize the energy statistic by replacing the Euclidean distances with a newly defined semimetric

$$K(x,y) := \sqrt{\rho_1(x_{(1)}, y_{(1)}) + \cdots + \rho_m(x_{(m)}, y_{(m)})},$$

where ρ_i are metrics or semimetrics on $\mathbb{R}^{p_i}, i = 1, \ldots, m$, and the vectors x and y are partitioned into m groups as $x = (x_{(1)}, \ldots, x_{(m)})$, where $x_{(i)} \in \mathbb{R}^{p_i}$ and $\sum_{i=1}^{m} p_i = p$ (analogously for y). Chakraborty and Zhang (2021) focus on the case where each ρ_i is a metric of strong negative type on $\mathbb{R}^{p_i}, i = 1, \ldots, m$. In that case $K(x,y)$ is a metric of strong negative type on $\mathbb{R}^p$. They define a t-test based on their newly proposed metric. They need several moment assumptions in their analysis and several other assumptions. The new test is shown to be able to detect a wider range of alternatives than the energy statistic but cannot detect differences beyond the equality of the low-dimensional marginal distributions with non-trivial power. No resampling-based inference is needed for their test, but a homogeneity metric as well as a grouping of samples is needed.

Rizzo and Székely (2010) show that the energy test can be seen as the treatment sum of squares in an ANOVA interpretation of the k-sample problem. They use a different measure of dispersion for univariate or multivariate responses based on all pairwise distances between-sample elements for ANOVA. With this, they derive their so-called *distance components (DISCO) decomposition* for powers of distances in $(0, 2]$ that gives a partition of the total dispersion in the samples into components analogous to the variance components in ANOVA. The resulting distance components determine a test for the general hypothesis of equal distributions. For each index in $(0, 2)$ this determines a nonparametric test for the multi-sample problem that is statistically consistent against general alternatives. For an index equal to two, it equals the usual ANOVA F-test. Their test statistic is somewhat similar to a generalization of the energy statistic where each of the differences is taken to the power α (given that $\mathbb{E}(\|X\|^\alpha) < \infty$, $\mathbb{E}(\|Y\|^\alpha) < \infty$). The new test is performed via permutation testing. Its asymptotic null distribution is a quadratic form (constants not given). The test is consistent against all alternatives with finite second moments. The choice of the index α is difficult. In general, the computational costs for calculating Gini means, in terms of which the test statistic can be formulated, is $\mathcal{O}(N^2)$, for $\alpha = 1$ it can be linearized and computation time reduces to $\mathcal{O}(N \log N)$. The simplest and most natural choice for α is one, for heavy-tailed distributions one may want to apply a small α. The test is implemented by permutation Bootstrap in the `R` package `energy` (Rizzo and Szekely, 2022).

Huang and Huo (2017) propose a *Randomly Projected Energy Statistics test* based on random projections and energy statistics to lower the computational costs from $\mathcal{O}(N^2)$ to $\mathcal{O}(mN \log N)$, with m denoting the number of random projections. For practical use, the number of random projections needs to be determined. Huang and Huo (2017) derive the asymptotic distribution of usual energy statistics and of the randomly projected one under conditions on expectations and variances and show that the modified version has nearly the same asymptotic efficiency as the usual energy statistic.

Deb and Sen (2021) present a rank version of energy statistic. Using the theory of measure transportation (optimal transport) a general framework for distribution-free, nonparametric tests based on multivariate ranks is provided. According to the authors, their test is nonparametric, exactly distribution-free, computationally feasible, and consistent against all alternatives under absolute continuity of the distributions. For consistency, no moment conditions are necessary, which enables the usage of heavy-tailed distributions. The test statistic is invariant under scaling and addition of a vector ($a + bZ, b \in \mathbb{R}, a \in \mathbb{R}^p$). The worst-case complexity for rank assignment is $\mathcal{O}(N^3)$. The calculation given ranks takes $\mathcal{O}(n_1 n_2 p)$. The resulting test is exactly equivalent to the Cramér-von Mises test for $p = 1$. An extension to the k-sample setting is possible. `R` code for the test is available on GitHub (`https://github.com/NabarunD/MultiDistFree.git`). The method can also be seen as a rank-based method.

Al-Labadi, Asl and Saberi (2022) propose an extension of the energy test to a Bayesian test for the k-sample problem, based on belief ratios. Their test is shown to be consistent. For the test, a prior has to be specified. Al-Labadi, Asl and Saberi (2022) choose a Dirichlet prior, but the choice of its parameters is not clear. Recommendations based on simulation are given. Additionally, a parameter in the belief ratio needs to be chosen. No implementation of the test is given, but a pseudocode algorithm is presented.

3.8.2. Other methods based on inter-point distances

Rigid motion invariant test Baringhaus and Franz (2010) define rigid motion (length and angle preserving transformation) invariant tests based on inter-point distances between samples and inter-point distances within each sample. The test is based on the Cramér test by Baringhaus and Franz (2004) which is equivalent to the energy test and the test by Szabo et al. (2002, 2003). Therefore it requires distributions with finite expectations. The Cramér test itself is rigid motion invariant (rigid motion $Qx + a$ where Q is an orthogonal matrix and a is a vector). The new test statistic generalizes the test statistic analogous to the Cramér test statistic by using a continuous function ϕ such that $\phi(\|x - y\|^2)$ is a negative definite kernel. The following conditions on the function ϕ are needed for consistency against all fixed alternatives. The resulting test statistic is nonnegative and zero if and only if H_0 is true, and one assumes w.l.o.g. that $\phi(0) = 0$ and ϕ is nonnegative. These assumptions are e.g. fulfilled for all distributions with finite support if and only if $\phi(\|x - y\|^2)$ is negative definite, which is equivalent to ϕ having a completely monotone derivative on $(0,\infty)$. For the existence of the test statistic, moment assumptions on distributions are needed that make sure that integrals over $\phi(\|X\|^2)$ exist. Different examples for functions are given, including as special cases the Cramér test, the test by Bahr (1996), and the test by Szabo et al. (2002). The test is not (asymptotically) distribution-free, but its asymptotic distribution can be approximated using a Bootstrap approach. It is shown to be consistent. Since the null distribution of the test statistic and also the asymptotic null distribution depend on the

common underlying distribution, the critical value needs to be approximated by Monte Carlo samples from the empirical distribution of the pooled sample or by bootstrapping. Efficiencies are examined under certain alternatives. Baringhaus and Franz (2010) give recommendations for the choice of the function ϕ based on simulations. Overall they recommend $\phi(z) = \log(1 + z)$ for general alternatives and for the Cramér test for location alternatives. An extension to the k-sample problem is possible. Tsukada (2019) give geometric interpretations of the tests. The tests are implemented with the recommended choices of ϕ for general use, for location alternatives, for scale alternatives, and ϕ corresponding to the Bahr (1996) test in the `R` package `cramer` (Franz, 2019).

Triangle test Liu and Modarres (2011) define a triangle test. First, one point from one of the samples and two points from the other sample are randomly selected. Then, it is examined how often the distance between the two observations from the same distribution is the largest, the middle, or the smallest in the triangle formed by these three observations. The test is asymptotically distribution-free under the null hypothesis of equal, but unknown continuous distribution functions, and it is well-defined when the number of variables p is larger than the number of observations N. Its computational complexity is independent of p. According to Biswas, Mukhopadhyay and Ghosh (2014), it is not distribution-free in finite samples. Biswas and Ghosh (2014) note that the test is rotation invariant.

Test for high dimension low sample size setting Biswas and Ghosh (2014) propose a test based on inter-point distances for high dimension, low sample size (HDLSS) setups, which is directly motivated by results of Maa, Pearl and Bartoszyński (1996). The test is invariant under location change, rotation, and homogeneous scale transformations and can be used even if the dimension is much larger than the sample size. Biswas and Ghosh (2014) derive results for increasing dimension and fixed sample sizes under assumptions (similar to those of Hall, Marron and Neeman (2005)) about increasing information with increasing dimensions, uniformly bounded fourth moments, weak dependence among component variables, and related to sample size. Under these assumptions, Biswas and Ghosh (2014) show consistency of their test for $p \to \infty$. If finite second moments of both distributions exist, additionally under H_0 the asymptotic distribution is shown to be a weighted chi-square distribution (asymptotically distribution free), and consistency for $N \to \infty$ and $n_1/n_2 \to$ const is shown. Sarkar and Ghosh (2018) show via simulations that the test has limitations in the HDLSS setup. The two distributions must differ in their locations or average variances to perform well in the HDLSS setup. Tsukada (2019) gives a geometric interpretation of the test.

Extensions of the Cramér test and the test for the HDLSS setting Sarkar and Ghosh (2018) aim to improve the tests based on mean inter-point distances that use the Euclidean distance (Cramér Test by Baringhaus and Franz (2004) and the test by Biswas and Ghosh (2014)), by using a new class

of distance functions instead. Block variants of the new tests are given, but the choice of the block size is left open. They show consistency under certain assumptions on the distributions and on the sample size for $p \to \infty$ for modified tests without and with blocking.

Cramér-von Mises test on inter-point distances Montero-Manso and Vilar (2019) develop a Cramér-von Mises test on inter-point distances motivated by Maa, Pearl and Bartoszyński (1996) that compares the whole distribution of inter-point distances instead of individual moments. Therefore, the univariate distributions of the pairwise distances within and between samples are compared using a Cramér-von Mises-type statistic. The resulting test statistic is called *distribution of distances (DD)* statistic. The test is applicable to a broad range of both continuous and discrete distributions since only the mild regularity conditions of Maa, Pearl and Bartoszyński (1996) are needed. Still, the theoretical results are only derived for continuous distributions. The asymptotic power of the test as p goes to infinity is not studied. For computing distances, a symmetric, real-valued, nonnegative function is required that fulfills mild regularity conditions but does not have to fulfill the triangle inequality. This function d needs to fulfill the condition $d(x, y) = 0$ iff $x = y$ and $d(ax+b, ay+b) = |a|d(x, y)$. For consistency of the test statistic also a bounded support of the distance is required. The test becomes asymptotically distribution-free under the null hypothesis and its critical value is obtained via a permutation approach. The computational cost is $\mathcal{O}(N^2 \log(N))$.

Modification of rigid motion invariant and HDLSS tests Tsukada (2019) propose new criteria based on the tests by Baringhaus and Franz (2010) and by Biswas and Ghosh (2014). The first test statistic is the length of the difference between the vector $\hat{\mu}$ consisting of estimated means of $\|X-Y\|, \|X-X'\|$ and $\|Y-Y'\|$ and the vector that projects $\hat{\mu}$ onto the line with direction vector $(1, 1, 1)^T$ via the origin. The Biswas and Ghosh (2014) test is the squared sum of $(2, -1, -1)^T\hat{\mu}$ and $(0, 1, -1)^T\hat{\mu}$. The second new test statistic uses a weighted sum of non-squared terms. Under the moment assumptions of Hall, Marron and Neeman (2005) (bounded fourth moments, ρ-mixing condition), results for fixed sample size and increasing dimension are derived. Under additional assumptions on the trace of the covariance matrices and the difference of the mean vectors and under the assumption of equal sample sizes that are not too small, consistency for $p \to \infty$ is shown for the second test. The asymptotic null distribution is derived for the second test under the assumption of finite second moments and for $N \to \infty$. The test is asymptotically distribution-free. Under the same assumptions, consistency for the second test is shown. In simulations, the power of the second test is stable for high-dimensional data and large samples. On the other hand, Tsukada (2019) state that they "expected the power of the [first proposed] test to be comparable to that of the [Biswas and Ghosh (2014)] test, but its performance was disappointing". Therefore they recommend the second test if there is no information that the two population covariance matrices are nearly identical, and they recommend the Baringhaus and Franz (2010) test if

it is known that the two population covariance matrices are equal.

3.9. Methods based on kernel (mean) embeddings

The general idea of kernel mean embeddings is to extend feature maps ϕ as used by other kernel methods (e.g. in the context of kernel support vector machines) to the space of probability distributions by representing each distribution F as a mean function

$$\phi(F) = \mu_F := \int_{\mathcal{X}} K(x, \cdot)\, \mathrm{d}F(x) = \mathbb{E}_F(K(X, \cdot)), \tag{7}$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric and positive definite kernel function. A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of functions on the domain $\mathcal{X}$ with kernel K is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ with dot product $\langle \cdot, \cdot \rangle$ that satisfies the reproducing property

$$\langle f(\cdot), K(x, \cdot) \rangle = f(x) \;\Rightarrow\; \langle K(x, \cdot), K(x', \cdot) \rangle = K(x, x'),$$

such that the linear map from a function to its value at x can be viewed as an inner product.

In the following, we always assume that the integral (7) exists. Then the kernel mean embedding as given above is essentially a transformation of the distribution F to an element in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ corresponding to the kernel K (Muandet et al., 2017). For characteristic kernels, the kernel mean representation captures all information about the distribution F, i.e. the map $F \mapsto \mu_F$ is injective, which implies $\|\mu_{F_1} - \mu_{F_2}\|_{\mathcal{H}} = 0 \Leftrightarrow F_1 = F_2$ (Fukumizu, Bach and Jordan, 2004; Sriperumbudur et al., 2008, 2010). Therefore the kernel mean embeddings can be used for comparing distributions. Conditions that ensure the characteristic property are given in Sriperumbudur et al. (2010) (e.g. by showing that integrally strictly positive definite kernels are characteristic) and in Sriperumbudur, Fukumizu and Lanckriet (2011). For more details on kernel mean embeddings and their applications refer to the comprehensive review of Muandet et al. (2017) and the papers cited therein. Here, we only give a brief overview of the main aspects regarding the problem of comparing two distributions.

3.9.1. Maximum mean discrepancy

The following section is largely based on the main points from Section 3.5 in Muandet et al. (2017), supplemented by additional findings from the sources cited therein as well as more recent findings.

Building on the ideas given above, a kernel mean embedding can be used to define a metric for probability distributions, the so-called *Maximum Mean Discrepancy (MMD)*

$$\mathrm{MMD}(\mathcal{H}, F_1, F_2) = \|\mu_{F_1} - \mu_{F_2}\|_{\mathcal{H}}. \tag{8}$$

It was proposed in the context of two-sample testing by Gretton et al. (2006) but enjoys increasing popularity in different applications like data integration (Borgwardt et al., 2006), generative adversarial networks (Li et al., 2017; Sutherland et al., 2017; Bińkowski et al., 2021), testing for independence (Gretton et al., 2012a), and goodness-of-fit testing Jitkrittum et al. (2018).
The MMD can equivalently be expressed as

$$\mathrm{MMD}(\mathcal{H}, F_1, F_1) = \sup_{f \in \mathcal{F}} \left(\int f(x)\, \mathrm{d}F_1(x) - \int f(x)\, \mathrm{d}F_2(x) \right),$$

with $\mathcal{F}$ the unit ball in a universal RKHS $\mathcal{H}$, and therefore belongs to the class of integral probability measures (Müller, 1997, cf. Section 3.6.1). MMD is bounded by the Wasserstein distance (11) and up to a constant also by the total variation distance (2) (Sriperumbudur et al., 2010, Theorem 2.1), so if two distributions are close w.r.t. one of those distances, they are also close according to MMD. The MMD can also be defined on other function spaces $\mathcal{F}$, which leads to a generalization of some further metrics like the Kolmogorov-Smirnov statistic or the Earth Mover's distances (Gretton et al., 2012a).

Another connection between MMD and other methods presented is that for translation invariant kernels, MMD can be written as

$$\mathrm{MMD}(\mathcal{H}, F_1, F_2) = \int_{\mathbb{R}^p} |\phi_{F_1}(\omega) - \phi_{F_2}(\omega)|^2 \, \mathrm{d}\Lambda(\omega),$$

where Λ is the spectral measure appearing in Bochner's theorem and ϕ_{F_1}, ϕ_{F_1} are the characteristic functions of F_1 and F_2 (Sriperumbudur et al., 2010, Corollary 4). So for translation invariant kernels, it can be interpreted as the $L^2(\Lambda)$ distance between the characteristic functions. See Section 3.3.2 for more methods based on comparisons of characteristic functions.
Another representation of MMD in terms of the associated kernel function that is useful for estimation is

$$\mathrm{MMD}^2(\mathcal{H}, F_1, F_2) = \mathbb{E}_{X,X'}\left[K(X, X')\right] - 2\,\mathbb{E}_{X,Y}\left[K(X, Y)\right] + \mathbb{E}_{Y,Y'}\left[K(Y, Y')\right]$$

where $X, X' \sim F_1$ and $Y, Y' \sim F_2$ are independent copies. $\mathrm{MMD}^2(\mathcal{H}, F_1, F_2)$ can be estimated by the U-statistic

$$\begin{aligned}
\widehat{\mathrm{MMD}}^2(\mathcal{H}, X, Y)_U &= \frac{1}{n_1(n_1-1)} \sum_{i=1}^{n_1} \sum_{\substack{j=1 \\ j \neq i}}^{n_1} K\left(x_i, x_j\right) \\
&\quad + \frac{1}{n_2(n_2-1)} \sum_{i=1}^{n_2} \sum_{\substack{j=1 \\ j \neq i}}^{n_2} K\left(y_i, y_j\right) - \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{\substack{j=1 \\ j \neq i}}^{n_2} K\left(x_i, y_j\right) \\
&= \frac{1}{n_1(n_1-1)} \sum_{i=1}^{n_1} \sum_{\substack{j=1 \\ j \neq i}}^{n_1} h_K\left((x_i, y_i), (x_j, y_j)\right), \text{ if } n_1 = n_2,
\end{aligned} \tag{9}$$

where $h_K((x,y),(x',y')) = K(x,x') - K(x,y') - K(y,x') + K(y,y')$ (Smola et al., 2007). This estimator is unbiased (Gretton et al., 2006). Sutherland (2019) presents an unbiased estimator of the variance of the squared MMD estimator and the difference of two correlated squared MMD estimators at essentially no additional computational cost.

Based on the above reformulations, MMD could also be seen as an IPM, a method based on comparing characteristic functions, or a method based on inter-point distances.

Under $H_0 : F_1 = F_2$, if $n_1 = n_2$ and $\mathbb{E}\left[h_K^2\right] < \infty$, it holds

$$n_1 \widehat{\mathrm{MMD}}^2(\mathcal{H}, X, Y)_U \xrightarrow{D} \sum_{l=1}^{\infty} \lambda_l [g_l^2 - 2]$$

with $g_l \sim N(0,2)$ i.i.d., λ_i solutions to $\int_{\mathcal{X}} \tilde{K}(x,x')\psi_i(x)\,\mathrm{d}p(x) = \lambda_i \psi_i(x')$, and centered RKHS kernel

$$\tilde{K}(X_i, X_j) = K(X_i, X_j) - \mathbb{E}_X(K(X_i, X)) - \mathbb{E}_X(K(X, X_j)) + \mathbb{E}_{X,X'}(K(X, X')).$$

Given a finite sample approximation of the $(1-\alpha)$-quantile of the null distribution of $n\widehat{\mathrm{MMD}}^2(X,Y)_U$, this can be used for testing H_0 against $H_1 : F_1 \neq F_2$. The quantiles can be approximated by bootstrapping or by fitting Pearson curves using the first four moments (Smola et al., 2007), or through a Gamma approximation of moments and by approximating the eigenvalues in the above expression by their empirical counterparts, which can be obtained from the Gram matrix (Gretton et al., 2009). The methods of Gretton et al. (2009) give a consistent estimate of the null distribution computed from the eigenspectrum of the Gram matrix on the pooled sample. They might therefore be preferable since Bootstrap is computationally costly and the Pearson curve fitting method has no consistency or accuracy guarantees. According to Song and Chen (2023) the MMD Bootstrap test performs poorly in experiments if (only) variance differs between high-dimensional distributions.

Alternatively, an asymptotic test based on the following asymptotic distribution shown by Muandet et al. (2017) based on the work of Gretton et al. (2012a) can be performed:

$$\sqrt{n_1}\left[\widehat{\mathrm{MMD}}^2(\mathcal{H}, X, Y)_U - \mathrm{MMD}^2(\mathcal{H}, F_1, F_2)\right] \xrightarrow{D} N(0, \sigma_{XY}^2),$$

where again $n_1 = n_2$ and $\mathbb{E}\left[h_K^2\right] < \infty$ is assumed and in addition it is assumed that $H_1 : F_1 \neq F_2$ holds and σ_{XY}^2 is defined as

$$\begin{aligned}\sigma_{XY}^2 = 4\Big(&\mathbb{E}_{(X,Y)}\left[\mathbb{E}_{(X',Y')}\left(h_K\left((X,Y),(X',Y')\right)\right)^2\right] \\ &- \left[\mathbb{E}_{(X,Y),(X',Y')}\left(h_K\left((X,Y),(X',Y')\right)\right)\right]^2\Big).\end{aligned}$$

The convergence rate of $1/\sqrt{n_1}$ of the statistic to its population value is independent of p (Sriperumbudur et al., 2012), but Muandet et al. (2017) warn that

the dimension may show up in a constant term which can make the upper bound arbitrarily large for high-dimensional data. Danafar et al. (2014) additionally note that the distribution of MMD degenerates under the null hypothesis and its estimator also degenerates under the null and has no consistency or accuracy guarantee.

Linear-time statistics The cost for computing $\widehat{\mathrm{MMD}}^2(\mathcal{H}, X, Y)_U$ is $\mathcal{O}\left(N^2\right)$ (Gretton et al., 2012a). To circumvent the quadratic cost, Gretton et al. (2012a) propose an unbiased linear-time statistic

$$\widehat{\mathrm{MMD}}^2(\mathcal{H}, X, Y)_{U,l} = \frac{1}{n_1} \sum_{i=1}^{\lfloor n_1/2 \rfloor} h_K\left((x_{2i-1}, y_{2i-1}), (x_{2i}, y_{2i})\right), \text{ if } n_1 = n_2, \quad (10)$$

for which the same convergence to a normal distribution can be shown with the only difference that its variance is only half as large as that for the quadratic-time statistic.

Another way to speed up the calculation is given by Zaremba, Gretton and Blaschko (2013). The motivation behind their modification is manifold. The test statistic is degenerate under the null hypothesis, and its asymptotic distribution takes the form of an infinite weighted sum of independent χ^2 variables. Further, the methods for estimating the null distribution in a consistent way (Bootstrap or method by Gretton et al. (2009)) are computationally demanding with costs of $\mathcal{O}(N^2)$ with a large constant or $\mathcal{O}(N^3)$ with a smaller constant, and Pearson curve fitting has no consistency guarantees. To solve these problems, they define a family of block tests for MMD. The choice of block size means a trade-off between power and computation time. To obtain an asymptotic Gaussian null distribution, the size of blocks B needs to be chosen such that $n_1/B \to \infty$ for $n_1 = n_2 \to \infty$. The assumptions made are the same as for quadratic-time MMD. Additional conditions for second moments are required for convergence of the test statistic. Due to the asymptotic Gaussian distribution, the critical values for testing are easy to compute. A choice for the size of blocks B is needed to perform the test, and only a heuristic choice of $\lfloor \sqrt{n_1} \rfloor$ is proposed by Zaremba, Gretton and Blaschko (2013). Moreover, like with the normal MMD test, the kernel needs to be chosen.

Zhao and Meng (2015) instead use the connection between MMD and characteristic functions to define an efficient test called fastMMD test. The idea is to equivalently transform MMD with shift-invariant kernels into amplitude expectation of a linear combination of sinusoid components based on Bochner's theorem and the Fourier transform (Rahimi and Recht, 2007). For this, they make use of sampling of Fourier transforms. By that, the complexity is reduced from $\mathcal{O}(N^2 p)$ to $\mathcal{O}(LNp)$, where L is the number of basis functions for approximating kernels, which determines the approximation accuracy. Spherically invariant kernels allow for further acceleration to $\mathcal{O}(LN \log p)$ by using the Fastfood technique (Le, Sarlos and Smola, 2013). Zhao et al. (2021) show convergence of their estimates.

Two important modifications to linear-time tests are given by Jitkrittum et al. (2016). They define two semimetrics on probability distributions using the sum of differences in expectations of analytic functions evaluated at either spatial or frequency locations. The goal is to choose features such that the distinguishability of distributions is maximized. Therefore, the lower bound on test power for tests using the features is optimized. This leads to two different linear-time tests.

The tests are based on analytic representations of probability distributions presented by Chwialkowski et al. (2015). The difference is that the features there are chosen at random, while here the lower bound for the test power is derived, which can be used to optimize the choice of the features.

The first test is the *Mean Embedding (ME)* test, which evaluates the difference of mean embeddings at locations chosen to maximize the test power lower bound (spatial features). The second test, the *Smooth Characteristic Function (SCF)* test, uses the difference of two smoothed empirical characteristic functions, evaluated at points in the frequency domain, which are chosen such that the same criterion is maximized (frequency features). The optimization of the mean embedding kernel/ frequency smoothing function is performed on held-out data. The ME and SCF test are defined in Chwialkowski et al. (2015) as follows: for the test samples, $X = \{X_1, \ldots, X_n\}$ and $Y = \{Y_1, \ldots, Y_n\}$ i.i.d. according to F_1 and F_2 are given. Both tests evaluate the hypotheses $H_0 : F_1 = F_2$ versus $H_1 : F_1 \neq F_2$.

The test statistic for the ME test is given by

$$
\begin{aligned}
T_{\mathrm{ME}} &= n \bar{Z}_n^T S_n^{-1} \bar{Z}_n, \text{ with} \\
\bar{Z}_n &= \frac{1}{n} \sum_{i=1}^{n} Z_i, \\
S_n &= \frac{1}{n-1} \sum_{i=1}^{n} (Z_i - \bar{Z}_n)(Z_i - \bar{Z}_n)^T, \\
Z_i &= \{K(X_i, V_j) - K(Y_i, V_j)\}_{j=1}^{J} \in \mathbb{R}^J.
\end{aligned}
$$

The test statistic depends on the positive definite kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^p$ and the set of J test locations $\mathcal{V} = \{V_1, \ldots, V_J\} \subseteq \mathbb{R}^p$. It is asymptotically chi-square distributed

$$
T_{\mathrm{ME}} \overset{H_0, asymp}{\sim} \chi^2_J
$$

and can be seen as a form of Hotelling's T^2 statistic. T_{ME} is a semimetric since it can be seen as squared normalized $L^2(\mathcal{X}, V_J)$ distance of the mean embeddings of the empirical measures $F_{1,n} = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ and $F_{2,n} = \frac{1}{n} \sum_{i=1}^{n} \delta_{Y_i}$ where $V_J = \frac{1}{J} \sum_{i=1}^{J} \delta_{V_i}$, and δ_x is the Dirac measure concentrated at x.

The SFC test statistic T_{SCF} is defined in the same way as T_{ME}, but uses a modified Z:

$$
\begin{aligned}
Z_i = &\{\hat{l}(X_i) \sin(X_i^T V_j) - \hat{l}(Y_i) \sin(Y_i^T V_j), \\
&\hat{l}(X_i) \cos(X_i^T V_j) - \hat{l}(Y_i) \cos(Y_i^T V_j)\}_{j=1}^{J} \in \mathbb{R}^{2J},
\end{aligned}
$$

$$\hat{l}(x) = \int_{\mathbb{R}^p} \exp(-iu^T x) l(u)\,\mathrm{d}u \quad \text{(Fourier transform of } l(x)\text{)}$$

Here $l : \mathbb{R}^p \to \mathbb{R}$ is an analytic translation-invariant kernel (i.e. $l(x-y)$ defines a positive definite kernel for x and y). The locations $\mathcal{V} = \{V_1, \ldots, V_J\} \subset \mathbb{R}^p$ are in the frequency domain. The test statistic is again asymptotically χ^2 distributed

$$T_{\mathrm{SCF}} \overset{H_0, asymp}{\sim} \chi^2_{2J}$$

and can be interpreted as a normalized version of the $L^2(\mathcal{X}, V_J)$ distance of the empirical smooth characteristic functions $\phi_{F_1}(v)$ and $\phi_{F_2}(v)$, where $\phi_F(v) = \int_{\mathbb{R}^p} \varphi_F(w) l(v-w)\,\mathrm{d}w$ with $\varphi_F(w) = \mathbb{E}_{X\sim F}\left[\exp\left(iw^T X\right)\right]$ is the characteristic function of F. Therefore, it could also be classified as a method based on comparing characteristic functions. Jitkrittum et al. (2016) denote the degrees of freedom of the χ^2 distribution for both tests as J'. They use a modification of the test statistic with regularization parameter γ_n:

$$T_{\mathrm{ME/SCF}} = n\bar{Z}_n^T (S_n + \gamma_n I)^{-1} \bar{Z}_n,$$

to obtain a higher stability of the matrix inversion. The asymptotic distribution under the null hypothesis stays the same as long as $\gamma_n \to 0$ for $n \to \infty$. Simulations on high-dimensional text and image data show that the tests are comparable to the state-of-the-art quadratic-time MMD test of Gretton et al. (2012b), but in contrast to the MMD tests return human-interpretable features explaining the test results. The test statistics depend on the set of test locations $\mathcal{V}$ and the kernel parameter σ. Jitkrittum et al. (2016) propose to set $\theta = \{\mathcal{V}, \sigma\} = \arg\max_\theta \lambda_n = \arg\max_\theta \mu^T \Sigma^{-1} \mu$, where $\lambda_n = n\mu^T\Sigma^{-1}\mu$ with $\mu = \mathbb{E}_{F_1,F_2}(Z_1)$, and $\Sigma = \mathbb{E}_{F_1,F_2}\left[(Z_1-\mu)(Z_1-\mu)^T\right]$ is the population counterpart of $T_{ME/SCF}$. Since a dependency between θ and the data used for testing would affect the null distribution, it is proposed to split the dataset in half and first use one half $\mathcal{D}^{tr}$ of $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ for optimizing θ via gradient ascent on $T^{tr}_{\mathrm{ME/SCF}}$ (in theory one should maximize λ_n but μ and Σ are unknown) and then perform the actual test using the test statistic $T^{te}_{\mathrm{ME/SCF}}$ on the other half $\mathcal{D}^{te}$ of the dataset. Convergence of the test statistic to λ_n is guaranteed for $n \to \infty$ over all kernels in a family of uniformly bounded kernels (e.g. Gaussian kernel class) and all test locations in an appropriate class. Jitkrittum et al. (2016) use the isotropic Gaussian kernel class $\mathcal{K}_g = \{K_\sigma : (x,y) \to \exp(-(2\sigma^2)^{-1}\|x-y\|_2^2) | \sigma > 0\}$, where σ is constrained to be in a compact set and $\mathbb{V} = \{\mathcal{V} |$ any two locations are at least ε distance apart, and all test locations have their norms bounded by $\zeta\}$, where $\mathcal{V}$ is the set of test locations as defined above. The authors conduct experiments to compare their proposed ME and SCF tests with the versions from Chwialkowski et al. (2015) (ME and SCF with σ optimized by grid search and random test locations) as well as with the quadratic-time and linear-time version of the MMD test (Gretton et al., 2012a) and the standard two-sample Hotelling's T^2 test. The newly proposed SCF test outperforms ME in terms of power and also

as the (quadratic-time) MMD test, while the linear-time MMD test performs worst. The quadratic-time MMD test becomes computationally infeasible for $p \in [5, 1500]$ and $n = 10000$. The observed type I error rate is too high for Hotelling's T^2 in high dimensions since an accurate estimation of the covariance matrix gets more difficult. The performance of the linear-time MMD test drops quickly with increasing dimension p, while the ME and SCF test with optimization show the slowest decrease in power with increasing dimension. On real data (text data/ image data) sometimes the ME test performs best and sometimes the ME and SCF test both perform well. Additionally, the learned location can be interpreted (e.g. by counting how often a specific word or pixel is chosen as a test location and looking at those that are chosen more often). The number of test locations J has to be chosen manually.

Other modifications to MMD There are several other modifications that do not aim at reducing the computational cost but focus on other aspects. Danafar et al. (2014) present a regularized Maximum Mean Discrepancy test for the comparison of multiple distributions. The regularizer is set provably optimal for maximal power such that there is no need for tuning by the user. The presented test is consistent under conditions on second moments. It has higher asymptotic power and higher power in small samples than the MMD and kernel Fisher discriminant analysis (KFDA) tests (Moulines, Bach and Harchaoui (2007), see below), but still a computational cost of $\mathcal{O}(N^2)$. Experiments show higher relative efficiency, compared to MMD and KFDA.

Cheng, Cloninger and Coifman (2020) propose a new kernel-based MMD statistic that can be made more powerful to distinguish certain alternatives when distributions are locally low-dimensional. The idea is to incorporate local covariance matrices and to construct an anisotropic kernel. The test's consistency is proven under mild assumptions on the kernel, as long as $\|f_1 - f_2\|\sqrt{n} \to \infty$. A finite-sample lower bound of the testing power is derived under the assumption that the distributions are continuous, compactly supported, and have densities w.r.t. the Lebesgue measure, and that $1 < p \ll \min(n_1, n_2)$. A set of reference points or a reference distribution and a covariance field, respectively, are required to conduct the test. Under the same assumptions as for consistency, Cheng, Cloninger and Coifman (2020) show that convergence of the power to 1 is at least as fast as $\mathcal{O}(N^{-1})$. The cost for computing one empirical estimate of the test statistic is $\mathcal{O}(NN_R)$, where N_R is the number of reference points.

Kirchler et al. (2020) propose a two-sample testing procedure based on a learned deep neural network representation. Instead of the kernel function that gives a feature representation, deep learning is used to obtain a suitable data representation. Kirchler et al. (2020) aim to overcome the problem that the MMD test depends critically on the choice of the kernel function and therefore "might fail for complex, structured data such as sequences and images, and other data where deep learning excels". At the same time, they want to improve the classifier two-sample test of Lopez-Paz and Oquab (2017) (see Section 3.10) that needs a train/test split of the data. The new test instead first maps the data onto a hidden layer of a deep neural network that was trained on an independent,

auxiliary dataset. This transformed data is then compared using the MMD test statistic of Gretton et al. (2012a) or a variant of it, or alternatively using the kernel FDA test (Harchaoui, Bach and Moulines, 2008) (see below). The corresponding procedures are called *Deep Maximum Mean Discrepancy (DMMD)* test and *Deep Fisher Discriminant Analysis (DFDA)* test, respectively. For the class of deep ReLU networks with a tanh activation function in the final layer, an asymptotic test based on an asymptotic normal or χ^2 distribution of the DMMD and DFDA test statistic is presented. For this, the covariance matrix of the learned feature map must exist and for DFDA it additionally must be invertible. Consistency of the tests can be shown under several assumptions on the neural network and its training and the assumption that the transfer task on which the deep neural network is fitted is not too far from the original task. There are no explicit directions on how to choose the transfer task since the theoretically optimal choice depends on the true distributions and the Bayes rate for the transfer task. So, if there is enough data, splitting is the safe way that guarantees the similarity of transfer and original task.

The new test of Song and Chen (2023) makes use of common patterns in moderate and high dimensions. It is aimed at solving the curse of dimensionality for kernel two-sample tests. It takes into account the variance-covariance matrix of the first two terms in (9). The test is implemented in the `R` package `kerTests` (Song and Chen, 2021). There are two corner cases in which the test statistic is not well-defined. In general, two conditions on the kernel and data are made that are usually fulfilled if there is no major outlier in the data and if one uses a Gaussian kernel with the median heuristic as described below. Under these, an asymptotic normal distribution for the test statistic is shown.

Choice of kernel function and parameters All methods described so far depend on a kernel function. The choice of this kernel function is nontrivial. Although there are many proposals on how to choose it, the optimal choice remains an open problem (Muandet et al., 2017).

In general, as stated at the beginning, characteristic kernels are preferred since they ensure that the MMD is zero if and only if the two distributions coincide. Details on conditions for kernels being characteristic are given in Sriperumbudur et al. (2009), Sriperumbudur, Fukumizu and Lanckriet (2011) and Simon-Gabriel and Schölkopf (2018). Concrete examples are listed in Table 3.1 of Muandet et al. (2017). Still, the class of characteristic kernels is large and leaves some room for decision.

Probably the most popular class of characteristic kernels are radial basis function (RBF) kernels. But even within this class, there are different proposals on how to choose the RBF kernel parameter. A heuristic for the choice of the kernel size for the RBF kernel is to set its parameter σ to the median distance between points in the pooled sample. The empirical MMD is zero both for a kernel size of zero and an infinitely large kernel size (Gretton et al., 2006).

A simulation study conducted by Gretton et al. (2006) shows that for low sample sizes, the threshold based on Pearson curves performs better in terms of type I error, while for high sample sizes, the Bootstrap threshold is preferred

due to the lower computational cost. In a simulation study, all in all, the method outperforms competing methods (t-test, Friedman-Rafsky Kolmogorov-Smirnov generalization (Friedman and Rafsky, 1979), Biau-Györfi test (Biau and Gyorfi, 2005), Hall-Tajvidi test (Hall and Tajvidi, 2002), or is at least close to the best-performing method (Gretton et al., 2006).

Later, Gretton et al. (2012b) propose to choose the kernel such that the test power is maximized for a given significance level. Therefore a kernel is selected from a particular family $\mathcal{K}$ of kernels. This family is defined as

$$\mathcal{K} = \left\{ K : K = \sum_{u=1}^{d} \beta_u K_u, \sum_{u=1}^{d} \beta_u = D, \beta_u \geqslant 0 \; \forall u \in \{1, \ldots, d\} \right\}$$

with a constant $D > 0$ and $\{K_u\}_{u=1}^{d}$ a set of positive definite functions $K_u : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which are assumed to be bounded, i.e. $|K_u| \leqslant C \, \forall u \in \{1, \ldots, d\}$. Then each kernel $K \in \mathcal{K}$ corresponds to exactly one RKHS $\mathcal{H}_K$ and the test statistic becomes

$$\widehat{\mathrm{MMD}}^2(\mathcal{H}_K, F_1, F_2)_{U,l} = \sum_{u=1}^{d} \beta_u \eta_u(F_1, F_2) = \mathbb{E}(\beta^T h) = \beta^T \eta,$$

where

$$\eta_u = \mathbb{E}_{XX'YY'}[h_{K_u}((X, Y), (X', Y'))]$$

and $h = (h_{K_1}, \ldots, h_{K_d})^T$, $\beta = (\beta_1, \ldots, \beta_d)^T$, and $\eta = (\eta_1, \ldots, \eta_d)^T \in \mathbb{R}^d$. The authors here make use of the asymptotically unbiased linear-time estimate of Gretton et al. (2012a) given in (10). To maximize the Hodges and Lehmann asymptotic relative efficiency (i.e. the power at a given significance level α) for the test based on the asymptotic normal distribution of the linear-time statistic, the following quadratic optimization program needs to be solved:

$$\min \left\{ \beta^T \left(\hat{Q} + \lambda_m I \right) \beta : \beta^T \hat{\eta} = 1, \beta \succeq 0 \right\},$$

if $\hat{\eta}$ has at least one positive entry. $\hat{Q}$ is a linear-time empirical estimate of the covariance matrix $\mathbb{C}\mathrm{ov}(h)$ and

$$\hat{\eta} = (\widehat{\mathrm{MMD}}^2(\mathcal{H}_{K_1}, F_1, F_2)_{U,l}, \ldots, \widehat{\mathrm{MMD}}^2(\mathcal{H}_{K_d}, F_1, F_2)_{U,l}).$$

The optimization is performed on a training set of m points (X_i, Y_i), $i = 1, \ldots, m$, and this training set and the data points used for testing are disjoint. In particular, all estimates needed for the optimization are calculated from this training data. If no entry of $\hat{\eta}$ is positive, a single base kernel K_u with the largest $\hat{\eta}_u / \hat{\sigma}_{K_u,\lambda}$ is arbitrarily selected since it is unlikely that the test statistic computed on the test data will exceed the always positive threshold.

$$\hat{\sigma}_{K,\lambda} = \sqrt{\beta^T \left(\hat{Q} + \lambda_m I \right) \beta} = \sqrt{\hat{\sigma}_K^2 + \lambda_m \|\beta\|_2^2}$$

is a regularized standard deviation estimate.
Gretton et al. (2012b) conducted simulations that show that their strategy for choosing an optimal kernel yields better results than other strategies, such as the aforementioned heuristic of setting the kernel size to the median distance between points in the aggregate sample or the strategy of maximizing the MMD test statistic proposed by Sriperumbudur et al. (2009). Their method is only outperformed by choosing the kernel with the highest ratio $\hat{\eta}_u/\hat{\sigma}_{K_u,\lambda}$ if a single best kernel exists. Otherwise, if a linear combination of kernels is needed, that strategy fails and the proposed optimal choice performs better in terms of power.
Another proposal for choosing the kernel is made by Liu et al. (2020) where deep kernels are used. The proposed kernel has the form

$$K_\omega(x, y) = [(1 - \varepsilon)\kappa(\phi_\omega(x), \phi_\omega(x)) + \varepsilon]q(x, y),$$

where ϕ_ω is a deep neural network with parameters ω that extracts features, and κ is a simple kernel (e.g. Gaussian with lengthscale σ_ϕ) on those features. q is a simple characteristic kernel on the input space and $0 < \varepsilon < 1$. This allows for an extremely flexible choice of kernels that can learn complex behavior. The parameters ω are selected by maximizing the ratio of the MMD to its variance, which asymptotically maximizes the power of the test. This is done in a similar train-test manner as in Gretton et al. (2012b), but here the proportion of the data assigned to the train set is optimized as well. Also, an improved estimator of the variance of the MMD estimator as proposed by Sutherland (2019) is used. This approach can be understood as a generalization of the evaluation of the accuracy of classifiers proposed by Lopez-Paz and Oquab (2017), but instead of cross-entropy, the test power is maximized.
The learning of the deep kernel is performed using minibatches of size m if the dataset is large. For each minibatch, the cost is $\mathcal{O}(mE + m^2C)$ with the term mE typically dominating for moderate m. Here, E denotes the cost of computing an embedding ϕ_ω and C the cost of computing the deep kernel. Testing is performed as a permutation test as proposed by Sutherland et al. (2017), instead of using the asymptotic distribution like proposed before, i.e. approximating the null distribution by drawing n_{perm} new samples X' and Y' from the pooled sample and calculating the test statistic on these samples. The permutation approach takes $\mathcal{O}(NE + N^2C + N^2 n_{\text{perm}})$ time.
It is shown theoretically that for reasonably large n and if the optimization process succeeds, the found kernel generalizes nearly optimally instead of just overfitting to the training data. Furthermore, the resulting test is compared to the one proposed by Gretton et al. (2012b) and the SCF and ME tests (Chwialkowski et al., 2015; Jitkrittum et al., 2016) as well as the classifier two-sample tests of Lopez-Paz and Oquab (2017) and Cheng and Cloninger (2022) in terms of type I error and power on several synthetic and real-world datasets. Liu et al. (2020) find that all tests keep the nominal type I levels and that the deep kernel MMD test generally has the highest power across a range of settings.
The MMD test along with different choices for kernels and many other kernel-based methods is implemented in the `R` package `kernlab` (Karatzoglou, Smola and Hornik, 2022; Karatzoglou et al., 2004).

3.9.2. Other kernel-based methods

Kernel Fisher discriminant analysis test Moulines, Bach and Harchaoui (2007) propose test statistics based on kernel Fisher discriminant analysis (kernel FDA). It is assumed that the kernel function is bounded for all probability measures $\mathbb{P}$ and that the RKHS associated with the kernel is dense in $L^2(\mathbb{P})$. Additionally, assumptions are made on eigenvalues of covariance matrices of both distributions. Both assumptions on the kernel are needed in the proof of consistency for the test, but only the first of each assumption is needed to show asymptotic normality. The resulting asymptotic normal distribution is independent of the kernel and an additional regularization parameter that must be chosen.

Tests based on symmetric kernels Fromont et al. (2012) present testing procedures based on a general symmetric kernel. Critical values of the tests are chosen by a wild Bootstrap or permutation Bootstrap approach. An aggregation method enables overcoming the difficulty of choosing a kernel and/or kernel parameters. It is demonstrated that the aggregated tests may be optimal in a classical statistical sense and non-asymptotic properties are shown for the aggregated tests. Therefore, the assumption is made that densities exist with respect to some non-atomic σ-finite measure and are square-integrable. A kernel needs to be chosen, but suggestions for this choice are given. An alternative test based on the conditional distribution of the test statistic given the sample is shown to be an exact level α test.

Findings on kernel and distance-based tests I Two important findings regarding kernel- and distance-based tests are given in Sejdinovic et al. (2013) and Ramdas et al. (2015).

Sejdinovic et al. (2013) establish a relationship between the energy test and the MMD test by showing that the energy statistic can be seen as a special case of MMD for a certain kernel function. For that, they give a generalized form of the energy statistic by replacing the Euclidean norm with other norms. They also determine the class of distributions for which these tests are consistent against all alternatives. In simulations, they show that the energy test is inferior regarding power. They make the same assumptions as Székely and Rizzo (2004) for the introduction and analysis of the energy statistic.

Ramdas et al. (2015) show that tests based on kernel embeddings or based on distances between pairs of points are not well-behaved for high-dimensional data, in contrast to general belief. Instead, they show that the power decreases at least polynomially in dimension for fair alternatives.

ME and SCF test based on L^1 distance Scetbon and Varoquaux (2019) present a test using the L^1 instead of the L^2 distance between kernel-based distribution representatives and define new ME and SCF test statistics based on that. They show that a sequence of Borel measures converges weakly towards a measure if and only if the $L^q, q \geqslant 1$, distance of their mean embeddings to

the mean embedding of that measure converges to zero, i.e. that the $L^q, q \geqslant 1$, distance metrizes weak convergence. They also show that their L^1 version rejects the null hypothesis better than the L^2 version under H_1 with high probability. The new tests are shown to be consistent for $N \to \infty$ and $n_1/N \to$ const, and the asymptotic distribution of the test statistic is shown to be a Nakagami distribution.

Kernel-based quadratic distance Chen and Markatou (2020) introduce a generalization of the MMD statistic to kernel-based quadratic distance. They give a review of two-sample tests that includes some of the tests presented here in less detail and also a review of the MMD literature that is not as detailed as that from Muandet et al. (2017). The test from Chen and Markatou (2020) is based on the *kernel-based quadratic distance* introduced by Lindsay et al. (2008)

$$d_K(F_1, F_2) = \int\int K(s,t)\,\mathrm{d}(F_1 - F_2)(s)\,\mathrm{d}(F_1 - F_2)(t)$$

with a nonnegative definite kernel K. An estimator is presented that relies on an appropriately centered kernel. Its limiting distributions under the null and the alternative and the exact variance under the null can be derived. However, those cannot be used for the construction of a critical value since the null distribution is an infinite sum that depends on eigenvalues of the centered kernel. Optimal tuning parameters can be chosen based on the ideas of Lindsay, Markatou and Ray (2014) for the one-sample test. Moreover, an extension to the k-sample problem is presented. The practical calculation of the test statistic under the assumption that the common distribution under H_0 belongs to a family of parametric distributions as well as the concrete form of the test statistic under a normal assumption are shown. Alternatively, a nonparametric calculation is possible by using the mixing distribution of F_1 and F_2, or its empirical counterpart, as the centering distribution for the kernel. Corresponding critical values for these two versions can be calculated by a parametric or nonparametric Bootstrap or in both cases with a permutation procedure.

Findings on kernel and distance-based tests II Zhu and Shao (2021) present situations in which the energy and MMD permutation tests are inconsistent. They show that the class of two-sample tests based on inter-point distances (generalized energy statistics) including MMD with Gaussian or Laplacian kernels and the energy statistic as well as the generalized energy statistic using the L^1 instead of the Euclidean distance are inconsistent when two high-dimensional distributions correspond to the same marginal distributions but differ in other aspects. Additionally, they derive the limiting distribution of a test statistic based on inter-point distances under low and medium sample sizes for increasing dimensions. They also show that under HDLSS and HDMSS, the energy statistic and MMD test are consistent if the sum of component-wise means or variances are not too small. On the other hand, if the sum of component-wise mean and variance differences are both of order $o(\sqrt{p}/\sqrt{n_1 n_2})$, then these tests suffer from a substantial power loss under HDLSS and have trivial power under

HDMSS. Under HDLSS, they have trivial power if additionally, the sum over squared covariance differences is $o(p)$. The L^1-norm-based test also experiences a power drop under HDLSS and has trivial power under HDMSS if the marginal univariate distributions are the same. Under HDLSS, it has trivial power when the distributions have the same bivariate marginal distributions. For the analysis, Zhu and Shao (2021) make assumptions on the existence of means and variances and additional moment and weak dependence assumptions on the components of X and Y. Zhu and Shao (2021) argue that in low dimensions the L^1-norm is not suitable since an L^1 distance of zero does not imply $F_1 = F_2$.

Bayesian kernel test Zhang et al. (2022) define a Bayesian kernel paired two-sample test based on modeling the difference between kernel mean embeddings in the RKHS. Their test is based on the framework of Flaxman et al. (2016) and automatically selects kernel parameters relevant to the problem. The use of a kernel allows for the use of the test beyond Euclidean spaces. In contrast to most other methods, they do not need the assumption that samples are independent of each other, but only the assumptions on the kernel as for the MMD. The test is conditional on the choice of the family of kernels. Zhang et al. (2022) focus on Gaussian RBF kernels in their analysis. The test statistic is based on the Bayes factor. Zhang et al. (2022) propose to model the witness function with a Gaussian process prior under the alternative model and to use a Gaussian noise model for the empirical witness vector given the bandwidth parameter. They derive the posterior distribution of the bandwidth parameter if it is unknown with a Gamma(2,2) prior under both null and alternative models and marginalize over it so that this parameter no longer has to be selected.

Kernel measure of multi-sample dissimilarity (KMD) Huang and Sen (2023) define a nonparametric kernel measure of multi-sample dissimilarity (KMD). Denote the dataset membership of each point in the pooled sample $\{Z_1, \dots, Z_N\}$ by $\{\Delta_1, \dots, \Delta_N\}$. If $\frac{n_i}{N} \to \pi_i \in (0,1)$ for $N \to \infty$ such that $\sum_i \pi_i = 1$ then $\{(\Delta_i, Z_i)\}_{i=1}^N$ can approximately be seen as an i.i.d. sample from $(\tilde{\Delta}, \tilde{Z})$ with distribution μ specified by $\mathbb{P}(\tilde{\Delta} = i) = \pi_i, i = 1, \dots, M$ and $\tilde{Z}|\tilde{\Delta} = i \sim F_i$. Let $(\tilde{Z}_1, \tilde{\Delta}_1), (\tilde{Z}_2, \tilde{\Delta}_2)$ i.i.d. samples from μ and $(\tilde{Z}, \tilde{\Delta}), (\tilde{Z}, \tilde{\Delta}') \sim \mu$ with $\tilde{\Delta}, \tilde{\Delta}'$ conditionally independent given $\tilde{Z}$. Denote by K a kernel function over the space $\{1, \dots, k\}$, e.g. the discrete kernel $K(x, y) := \mathbb{1}(x = y)$. Then the *kernel measure of multi-sample dissimilarity* (KMD) is defined as

$$\eta(F_1, \dots, F_k) := \frac{\mathbb{E}\left[K(\tilde{\Delta}, \tilde{\Delta}')\right] - \mathbb{E}\left[K(\tilde{\Delta}_1, \tilde{\Delta}_2)\right]}{\mathbb{E}\left[K(\tilde{\Delta}, \tilde{\Delta})\right] - \mathbb{E}\left[K(\tilde{\Delta}_1, \tilde{\Delta}_2)\right]}.$$

It has a lower bound of 0 that is attained if and only if the k distributions coincide and an upper bound of 1 that is attained if and only if all distributions are mutually singular. Monotonicity of η for location and scale alternatives for $k = 2$, $\mathcal{X} = \mathbb{R}^p, p \geqslant 1$ and log-concave distributions is shown such that values of KMD in $(0, 1)$ can be interpreted reasonably. Moreover, it is a member of the multi-distribution f-divergence as defined by García-García and Williamson

(2012) (see Section 3.6.2) and therefore fulfills all properties of the f-divergences. An estimator of η can be defined as follows. Given the pooled sample $Z_1, \ldots, Z_N$ and the corresponding sample memberships $\Delta_1, \ldots, \Delta_N$ let $\mathcal{G}$ be a geometric graph on $\mathcal{X}$ such that an edge between two points Z_i and Z_j in the pooled sample implies that Z_i and Z_j are close, e.g. the K-nearest neighbor graph with $K \geqslant 1$ or the MST. Denote by $(Z_i, Z_j) \in \mathcal{E}(\mathcal{G})$ that there is an edge in $\mathcal{G}$ connecting Z_i and Z_j. Moreover, let o_i be the out-degree of Z_i in $\mathcal{G}$. Then an estimator for η is defined as

$$\hat{\eta} := \frac{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{o_i}\sum_{j:(Z_i,Z_j)\in\mathcal{E}(\mathcal{G})}K(\Delta_i,\Delta_j) - \frac{1}{N(N-1)}\sum_{i\neq j}K(\Delta_i,\Delta_j)}{\frac{1}{N}\sum_{i=1}^{N}K(\Delta_i,\Delta_i) - \frac{1}{N(N-1)}\sum_{i\neq j}K(\Delta_i,\Delta_j)}.$$

This estimator is consistent and asymptotically normally distributed under assumptions similar to those of Deb and Sen (2021) on the geometric graph and for characteristic kernel functions. The k-sample test based on KMD is shown to be consistent against all alternatives where at least two distributions are unequal and Huang and Sen (2023) provide a complete characterization of the asymptotic power and detection threshold of the test for $\mathcal{X} = \mathbb{R}^p$ and assuming that P_i has a density w.r.t. the Lebesgue measure. Under H_0 the permutation and unconditional distribution of the estimator of KMD are both asymptotically normal and if $\mathcal{X}$ is a Euclidean space and the common distribution under H_0 has a Lebesgue density and under assumptions on the graph, the asymptotic null distribution is distribution-free. The test can be seen as a generalization of the two-sample statistic of the K-nearest neighbor test of Schilling (1986) and Henze (1988) or Petrie (2016). It could therefore also be assigned to the class of graph-based tests. It is implemented in the `R` package `KMD` (Huang, 2022). For the K-nearest neighbor graph (with $K \geqslant 1$ fixed) the calculation of $\hat{\eta}$ has computational complexity $\mathcal{O}(KN \log N)$. The use of the nearest neighbor graph rather than MST is recommended because of flexibility and computational convenience. Moreover, they recommend to use $K = 1$ for K-NN graph for estimation of η. For testing, larger values of K are recommended.

3.10. Methods based on binary classification

Classifier tests of Friedman The idea of measuring divergence between two distributions via separation and the misclassification error can be traced back as far as the 50's and 60's (Rao, 1952; Ali and Silvey, 1966). Later, Friedman (2004) brings up the idea of using a binary classifier to distinguish between distributions generating the two datasets. For that, a binary classifier is trained on the pooled dataset $\mathcal{D} = \{(X_i, 1)\}_{i=1}^{n_1} \cup \{(Y_i, -1)\}_{i=1}^{n_2} =: \{(Z_i, L_i)\}_{i=1}^{N}$. This binary classifier provides scores s_i for the confidence that sample i belongs to the first dataset ($L_i = 1$). The scores for the first and the second datasets can be seen as random samples from respective probability distributions with densities f_+ and f_-. Thus, a univariate two-sample test for equality of these densities, i.e. $H_0 : f_+(s) = f_-(s)$, e.g. chi-squared, Kolmogorov-Smirnov, Mann-Whitney,

or t-test, can be used to compare the distributions. To perform such a test, there are two options. First, the data are split into a training and test set and only the training set is used to train the classifier while the test set is used to perform the test, making use of the known null distribution of the respective test statistic. Second, all observations are used for training the classifier as well as for testing. Then the null distribution of the test statistic from the univariate test is not valid. Instead, a permutation test is performed by randomly permuting the labels and calculating the test statistic values for the classifiers trained on the permuted data. The empirical $(1-\alpha)$-quantile of these statistics can then be used as the critical value. The power of the test highly depends on the classifier but is likely not very sensitive to the choice of the univariate test statistic. The sensitivity in the choice of the classifier can be exploited to obtain a higher power by choosing a classifier that fits the differences of distributions that are of particular interest. Moreover, depending on the classifier, the differences in the distributions can further be examined after a rejection of the null hypothesis, e.g. for decision trees.

Classifier two-sample tests (C2ST) The general idea of Lopez-Paz and Oquab (2017) is to use a binary classifier for classifying to which of two datasets a sample belongs (here labeled by 0 and 1). If the datasets are generated from the same distribution, the accuracy should be close to chance level, otherwise, the classifier should be able to distinguish between the two distributions and hence the accuracy should be higher than chance level. A *Classifier Two-Sample Test (C2ST)* based on these considerations learns a representation of the data on the fly, and its test statistic is in interpretable units. Moreover, the predictive uncertainty allows interpreting where the distributions differ.
For the definition of the test statistic, w.l.o.g. assume that $n_1 = n_2$ and that two samples are given over the same sample space. The C2ST then consists of five steps:

1. Construct the dataset

$$\mathcal{D} = \{(X_i, 0)\}_{i=1}^{n_1} \cup \{(Y_i, 1)\}_{i=1}^{n_2} =: \{(Z_i, L_i)\}_{i=1}^{N}$$

   consisting of the samples from both datasets labeled with their membership to the two datasets.
2. Shuffle $\mathcal{D}$ at random and split it into a disjoint training and test set $\mathcal{D}^{\text{tr}}$ and $\mathcal{D}^{\text{te}}$ with $n_{\text{te}} = |\mathcal{D}^{\text{te}}|$.
3. Train a binary classifier $f : \mathcal{X} \to [0, 1]$ on $\mathcal{D}^{\text{tr}}$ such that $f(z_i)$ is an estimate of the conditional probability distribution $p(L_i = 1|Z_i)$.
4. Calculate the C2ST statistic on $\mathcal{D}^{te}$

$$\hat{T}_{\text{C2ST}} = \frac{1}{n_{\text{te}}} \sum_{(Z_i, L_i) \in \mathcal{D}^{\text{te}}} \mathbb{I}\left[\mathbb{I}\left(f(Z_i) > \frac{1}{2}\right) = L_i\right],$$

   which is the accuracy on the test set. $\mathbb{I}$ denotes the indicator function. The accuracy should be close to chance level if $F_1 = F_2$ and should be greater

than chance level for $F_1 \neq F_2$, since then the classifier should identify distributional differences between the two samples.

5. Calculate a p-value using the null distribution of the C2ST statistic, which is approximately $N(\frac{1}{2}, \frac{1}{4n_{\text{te}}})$.

Maximizing the power of a C2ST is a trade-off between a large training set, to optimize the classifier, and a large test set n_{te}, to better evaluate the performance of the classifier.

The test statistic is interpretable as the percentage of samples that are correctly classified. Furthermore, the values $f(z_i)$ along with the true labels l_i explain which samples were correctly or wrongly classified and with how much confidence. This provides information on where the two distributions differ. Using the classification-based approach also inherits the interpretability of the classifier to explain which features are most important for distinguishing between the two distributions.

In a simulation study, Lopez-Paz and Oquab (2017) compare C2ST using a neural network and C2ST using a K-NN classifier against the Wilcoxon-Mann-Whitney test, KS test, and Kuiper test for one-dimensional data, and additionally the MMD test, ME test and SCF test for one-dimensional as well as multi-dimensional data. They repeat the experiments from Jitkrittum et al. (2016). In all cases, C2ST shows a good performance. They observe that C2ST is better or nearly as good as SCF and MMD in the multi-dimensional case and nearly as good as the Kuiper and the ME test in the one-dimensional case.

Cai, Goggin and Jiang (2020) argue that disadvantages of the C2ST are that the use of train/test data for estimating the prediction accuracy makes the test less efficient in data utilization and can slow down the computation. They show that a more powerful test can be derived by not using the prediction accuracy directly (see below). The test is implemented in the `R` package `Ecume` (Roux de Bezieux, 2021).

Regression based test Kim, Lee and Lei (2019) derive a test that is intended for high-dimensional and complex data. A regression approach is used so the test can efficiently handle different types of data structures depending on the chosen regression model. Local differences can be identified with statistical confidence. The test gives a general framework for both global and local two-sample problems and for high-dimensional and non-Euclidean data. It is assumed that the densities of both distributions exist. The idea of the test is similar to that in other approaches based on binary classification. The equivalent null hypothesis based on regression for a binary outcome that determines the membership of data points is that the regression function does not depend on the features. The test statistic measures the empirical distance between the regression function $\mathbb{P}(Y = 1|X = x)$ and the class probability $\mathbb{P}(Y = 1)$ which both take values in $(0, 1)$. The power of the test can be related to the mean integrated squared error (MISE) of the chosen regression estimator. The null distribution of the test statistic is unknown and depends on the regression model and the distribution of the data. Therefore, a permutation test is performed.

Kim, Lee and Lei (2019) use Fisher's LDA as the regression method and show optimality under the assumption of normal distributions with equal covariance matrices. In general, a train/ test split is required for the method. Kim, Lee and Lei (2019) assume that the MISE is smaller than a positive constant times an $o(1)$ term and that the permutation critical value is uniformly bounded by this term up to some constant factor with high probability. Then, the procedure yields a level α test, and for sufficiently large N and for sufficiently large differences between the distributions, the type II error of the test is bounded. Kim, Lee and Lei (2019) use a linear smoother as the regression method (e.g. kNN regression, kernel regression, or local polynomial regression) for theoretical analysis. The convergence rates can be used for calculations on test errors. Note that the authors call their test regression-based, but model $\mathbb{P}(Y = 1|X = x)$ like in many of the other classification approaches.

Test based on the logit function of a classifier Cheng and Cloninger (2022) follow a slightly different approach for using a binary classifier network to distinguish between data from two different distributions. They train a classifier network and use the difference between both datasets of the provided logit function as the test statistic. An advantage of using networks is that the algorithm scales to large samples. Also, the use of networks is motivated by generalizing discriminative networks used in generative adversarial networks (GANs) from the goodness-of-fit problem to two-sample problems.
For the calculation of the test statistic, it is assumed w.l.o.g. that $N = n_1 + n_2$ is an even integer. Then the test is performed via the following steps:

1. Split the dataset $\mathcal{D}$ constructed as in Lopez-Paz and Oquab (2017) into two halves used as training and test set with n_1^{te} and n_2^{te} denoting the number of samples from datasets one and two, respectively, in the test set.
2. Training: Train a binary classification neural network on the training set using softmax loss. This gives estimated class probabilities

$$\mathbb{P}(l = 0|z) = \frac{\exp(u_\theta(z))}{\exp(u_\theta(z)) + \exp(v_\theta(z))},$$
$$\mathbb{P}(l = 1|z) = \frac{\exp(v_\theta(z))}{\exp(u_\theta(z)) + \exp(v_\theta(z))}$$

   with $u_\theta(z)$ and $v_\theta(z)$ activations in the last hidden layer of the network and θ the network parametrization. The *logit* is then defined as

$$f_\theta = u_\theta - v_\theta$$

3. Testing: The test statistic is computed as

$$\hat{T}_{CC} = \frac{1}{n_1^{\text{te}}} \sum_{x \in X^{1,\text{te}}} f_\theta(x) - \frac{1}{n_2^{\text{te}}} \sum_{y \in X^{2,\text{te}}} f_\theta(y)$$

   with f_θ parametrized by a trained neural network and $X^{1,\text{te}}$ and $X^{2,\text{te}}$ denoting the subsets of the test set corresponding to the first and the

second dataset. The critical value τ is calculated by a permutation test where the labels on the test set are randomly permuted m_{perm} times and the test statistic is recomputed each time using the permuted labels. τ is set to the empirical $(1-\alpha)$-quantile of these test statistics.

The test statistic can be viewed as estimating the symmetric KL divergence $\text{KL}(F_1, F_2) + \text{KL}(F_2, F1)$ (see Section 3.6.2).
Under the assumption that the training is terminated after a fixed number of epochs, the overall complexity of the test is $\mathcal{O}(N)$. Under certain assumptions regarding the neural network and the densities of F_1 and F_2, the test is asymptotically consistent. Moreover, a reduction of the needed network complexity for densities on or near low-dimensional manifolds in ambient space is shown.
In a simulation, the test is compared to the one proposed by Lopez-Paz and Oquab (2017) and to different kernel choices for the MMD test, where the kernel bandwidth is chosen as the median of the pairwise distances among all samples, as proposed in Gretton et al. (2012a). Cheng and Cloninger (2022) observe better performance of their test than for the C2ST and in certain settings (especially high dimensional data) also than for the MMD tests.

Test based on classification tree Yu et al. (2007) describe a two-sample test motivated by candidate gene association studies from the perspective of supervised machine learning. The estimated prediction error of a classification tree is used as a test statistic. A simulation study shows that the nominal type I error holds, but the power is sensitive to the chosen estimator for prediction error. The .632+ estimator results in the best overall performance. One advantage of the use of classification trees is that it enables the use of missing data since a tree can handle them via the use of surrogate variables.

Direction-projection-permutation (DiProPerm) test Wei et al. (2016) concentrate on the HDLSS setting and propose the so-called *direction-projection-permutation (DiProPerm)* test as a tool to assess whether a binary linear classifier detects statistically significant differences between high-dimensional distributions. The main idea is to work directly with the one-dimensional projections induced by the binary linear classifier. According to Wei et al. (2016), consistency is a nontrivial property in the HDLSS asymptotic regime, but certain variations of DiProPerm are consistent. In HDLSS settings, for ease of interpretability linear classifiers are preferable to more complicated ones like random forests. The test statistic is a univariate two-sample statistic applied to the projection onto the normal vector of a separating hyperplane. A permutation test is performed. In general, the choice of the classifier is open, but Wei et al. (2016) recommend using the distance weighted discrimination (DWD) classifier (Marron, Todd and Ahn, 2007). Also, different test statistics can be chosen (e.g. difference in means, t-test statistic, AUC). The theoretical analysis is performed only for the centroid projection direction and on the mean difference (MD) statistic and the t-statistic because these have simple closed-form expressions. Similar assumptions are needed for HDLSS asymptotic theory as

in Biswas and Ghosh (2014). Under these assumptions, the test is only shown to be consistent for the alternative of unequal means. The proof of consistency is performed only under certain alternatives (equal means, different covariance matrices) and only for centroid-t-statistic, while the test based on centroid-MD is inconsistent in this setting. Montero-Manso and Vilar (2019) mention that the test is not distribution-free. The test is implemented in the `R` package `diproperm` (Allmon, Marron and Hudgens, 2021).

Classification probability test Cai, Goggin and Jiang (2020) present a test, called the *Classification Probability Test (CPT)*, based on estimates of classification probabilities from a classifier trained on the samples. It can be applied whenever there is an appropriate classifier to consistently estimate the classification probabilities. In contrast to other classification-based tests, this test is not based on classification accuracy. Instead of testing $H_0 : F_1 = F_2$ directly, the idea is to equivalently test for hypotheses on the joint distribution of the data points and their dataset labels. For this, the odds ratio (OR) of probabilities that the label of a given feature point is one is used as a proxy for the likelihood ratio (LR) since $\text{LR} = \text{OR} \cdot \text{const}$ in this case. Since the test is an approximation of the LR test, asymptotically there should be no loss of information in contrast to the classification accuracy test proposed by Kim et al. (2021). For the test, it is assumed that a consistent estimator of the classification probability is given. According to Cai, Goggin and Jiang (2020), more research is needed on sufficient conditions for that. In addition, the assumption is made that the density functions of both distributions exist. A permutation test is performed. The test statistic estimates the KL divergence whenever the law of large numbers holds. An advantage of the test is that it does not need any density or density ratio estimation but only class probability estimates that can be obtained efficiently by different classification algorithms. The test performance generally depends on the underlying distribution and the classifier. Under the condition of uniform consistency for the estimation of class probabilities, the test is asymptotically most powerful. This uniform consistency condition is strong and artificial. Therefore, a second test is proposed based on more heuristic arguments that the two-sample test is equivalent to determining if the mapping of observations to class probabilities is a constant function. For this test, the variance of the estimated class probabilities is considered as a test statistic and again a permutation test is performed. In both cases, a classifier has to be chosen. Cai, Goggin and Jiang (2020) propose to choose it by K-fold cross-validation which is computationally intensive. Cai, Liu and Xia (2013) do not mention it, but probably some sort of training set is needed to train the classifier in the first step.

Testing for deviation of classification accuracy from chance Kim et al. (2021) analyze a general test based on checking if the accuracy of a classifier is significantly different from chance and compare it with Hotellings T^2 test. If the true error remains by at least $\varepsilon > 0$ better than chance as $p, N \to \infty$, then the permutation test is consistent. It is also computationally efficient. The

permutation test offers exact control of the type I error rate and is consistent if the number of permutations is greater than $(1 - \alpha)/\alpha$. A test based on a Gaussian approximation is also shown to be consistent. It is simple but has no finite sample guarantee. Kim et al. (2021) focus their analysis on tests for Gaussian or elliptical distributions. For performing the test, a train/ test split is required.

Test based on random forests Hediger, Michel and Näf (2022) provide a two-sample test based on the classification error of random forests that is applicable for any distribution. It requires almost no tuning, but for an asymptotic version of the test, both train and test sets are required. Alternatively, an out-of-bag (OOB) based permutation test can be performed. OOB statistics can be used to increase the sample efficiency compared to the test based on a holdout sample. The variable importance measures of the random forest provide insights into sources of distributional differences. The test is implemented in the `R` package `hypoRF` (Hediger, Michel and Näf, 2021).

Critique on accuracy based test Rosenblatt et al. (2021) criticize tests that analyze whether the estimated accuracy of a classifier is significantly better than chance level. Such tests can be underpowered compared to a "bona fide statistical test" and are also computationally more demanding. They examine candidate causes for low power, including the discrete nature of the accuracy test statistic, the types of signals that accuracy tests are designed to detect, the inefficient use of data, and a suboptimal regularization. For the analysis, they assume that the number of samples is in the order of the dimension or smaller. They demonstrate that in the high-dimensional regime accuracy tests never have more power than two-sample location or goodness-of-fit (GOF) tests. Problems with accuracy tests are that data splitting reduces the effective sample size, required regularization for testing seems to differ from that for predicting, and discretization makes the permutation tests conservative. The last point can not be captured in theoretical analyses as it decreases with sample size. Therefore, they recommend choosing a two-sample location or GOF test over an accuracy test and using appropriate regularization. For the use of accuracy tests, they recommend using larger test sets, regularization, and resampling with replacement. The results are fully based on a simulation study. No theoretical results are provided.

3.11. Distance and similarity measures for datasets

Distance and similarity based on metafeatures Feurer, Springenberg and Hutter (2015) define a distance measure between datasets. They intend to use it for speeding up Sequential Model-based Bayesian Optimization (SMBO) for hyperparameter tuning by using configurations that performed well on similar datasets for initialization (meta-learning). Under the assumption that each dataset $\mathcal{D}^{(i)}$ can be described by a set of K metafeatures $m^i = (m^i_1, \ldots, m^i_K)$

they propose two distance measures. The first one uses the q-norm of the difference between metafeatures of the datasets

$$D_{Fq}(\mathcal{D}^{(i)}, \mathcal{D}^{(j)}) = \|m^i - m^j\|_q.$$

The second one measures similarity w.r.t. performance of different hyperparameter settings by using the negative Spearman correlation between ranked results of a fixed set of n hyperparameter settings $\theta_l, l = 1, \ldots, n$, on both datasets

$$D_{Fc}(\mathcal{D}^{(i)}, \mathcal{D}^{(j)}) = 1 - \mathbb{C}\text{or}([g^{\mathcal{D}^{(i)}}(\theta_1), \ldots, g^{\mathcal{D}^{(i)}}(\theta_n)], [g^{\mathcal{D}^{(j)}}(\theta_1), \ldots, g^{\mathcal{D}^{(j)}}(\theta_n)]),$$

with $g^{\mathcal{D}^{(i)}}$ denoting the target function. In the context of finding the most similar of the datasets for which the hyperparameters have already been tuned to a new dataset for which the tuning has not yet been performed, the distance from the old datasets to the new one cannot be calculated, since the $g^{\mathcal{D}^{(i)}}(\theta_l)$ are not known for this new dataset. Instead, the distances are estimated using regression to learn a function mapping from pairs of metafeatures $\left(m^i, m^j\right)$ to $D_{Fc}(\mathcal{D}^{(i)}, \mathcal{D}^{(j)})$ based on the metafeatures and pairwise distances of the old datasets. Feurer, Springenberg and Hutter (2015) suggest 46 metafeatures found in the literature. These metafeatures can be categorized into

- simple metafeatures (describe basic dataset structure, e.g. number of features),
- PCA metafeatures,
- information-theoretic metafeatures (measure entropy),
- statistical metafeatures (use descriptive statistics to characterize dataset, e.g. kurtosis or dispersion of label distribution),
- landmarking metafeatures (are based on running several fast machine learning algorithms that can capture different properties of the dataset, e.g. linear separability).

Gromov-Hausdorff distance of metric measure spaces Mémoli (2017) defines a distance between datasets via the Gromov-Hausdorff metric between metric measure spaces. The idea is to represent data as a metric space endowed with a probability measure *(metric measure space)* and then determine the distance between these metric measure spaces. Given two metric measure spaces $(\mathcal{X}, d_\mathcal{X}, \mu_\mathcal{X})$ and $(\mathcal{Y}, d_\mathcal{Y}, \mu_\mathcal{Y})$ corresponding to the two datasets, denote by $\mathcal{U}(\mu_\mathcal{X}, \mu_\mathcal{Y})$ the collection of all couplings between $\mu_\mathcal{X}$ and $\mu_\mathcal{Y}$, i.e. of all measures μ over $\mathcal{X} \times \mathcal{Y}$ such that the push-forward of μ (i.e. the measure $\mu \circ f^1(A) = \mu(f^-1(A))$ for some measurable function f) for the first canonical projection π_1 is equal to $\mu_\mathcal{X}$, $\mu \circ \pi_1^{-1} = \mu_\mathcal{X}$, and analogously $\mu \circ \pi_2^{-1} = \mu_\mathcal{Y}$. Then the *Gromov-Wasserstein distance of order q* (Mémoli, 2011) is defined as

$$d_{\mathrm{GW},p}(\mathcal{X}, \mathcal{Y}) := \frac{1}{2} \inf_{\mu \in \mathcal{U}(\mu_\mathcal{X}, \mu_\mathcal{Y})} \left(\int \int |d_\mathcal{X}(x, x') - d_\mathcal{Y}(y, y')|^q \right.$$
$$\left. \mu(\mathrm{d}x \times \mathrm{d}y)\mu(\mathrm{d}x' \times \mathrm{d}y') \right)^{1/q}.$$

This means that the function $(x, y, x', y') \mapsto |d_{\mathcal{X}}(x, x') - d_{\mathcal{Y}}(y, y')|^q$ is integrated over the measure $\mu \otimes \mu$ for any $\mu \in \mathcal{U}(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ and the infimum with respect to μ is determined (Mémoli, 2017). For $q \geqslant 1$ this defines a proper distance on the collection of isomorphism classes of metric measure spaces (Mémoli, 2011, 2017). The calculation in practice remains unclear. Also the choice of the metrics $d_{\mathcal{X}}$ and $d_{\mathcal{X}}$ might be nontrivial in practice.

Similarity based on method ranking Leite, Brazdil and Vanschoren (2012) work on meta-learning in situations where it is not possible to evaluate and compare all combinations of learning algorithms and their possible parameter settings. For that, they develop a new technique called *active testing* that intelligently selects the most promising competitor for the next round of cross-validation based on prior duels between algorithms on similar datasets. Therefore, they characterize datasets based on the pairwise performance differences between algorithms. Their idea is that if the same algorithms win, tie, or lose in comparisons, then the datasets are expected to be similar at least in terms of effects on learning performance. They propose four ways to calculate dataset similarity. The first measure, called AT0, is not of interest since it assumes the same similarity for all pairs of datasets and is only used as a baseline. The second one, AT1, works as AT0 at the beginning before any tests on the new data were performed. Then, in each of the next iterations of cross-validation (CV) on the new data, the similarity is estimated based on the most recent CV test as follows. All datasets for which the new current best algorithm is better than the old one are assigned a similarity value of 1, and all other datasets have a similarity value of 0. An alternative is to set the similarity to the difference of relative landmarks (performance gain of the new best compared to the old best) for all datasets for which the new current best algorithm is better than the old one and then normalize these values to the range between 0 and 1. The third measure, ATW, works like AT1 but uses all CV tests carried out on the new dataset and calculates the Laplace-corrected ratio of results in which the datasets had the same results. The last measure, called ATx, works similarly to ATW but it is required that all pairwise comparisons yield the same outcome. In that case, the similarity is set to one, and otherwise to zero. Leite, Brazdil and Vanschoren (2012) present experiments to compare the different approaches. The results show that ATW and AT1 provide good performance using a small number of CV tests. Nonetheless, they believe that the results could be improved by using classical information-theoretic measures and/ or sampling landmarks for measuring the dataset similarity.

In Leite and Brazdil (2021), an improved version is presented that outperforms the previous active testing data similarity measures. For this, the performance gain of each algorithm on each dataset compared to the current best algorithm is estimated as the ratio of the performances of these algorithms divided by the ratio of the runtimes required for training the learners to the power of a parameter q. The authors recommend $q = 1/32$. The performance gain is estimated as this quantity minus one, if the resulting value is positive, and zero otherwise. The similarity of datasets is measured via the (weighted or

unweighted) correlation of these estimated performance gains of all algorithms on the respective datasets.

Deep dataset dissimilarity measures Calderon Ramirez et al. (2022) define another set of dataset dissimilarity measure, called the *deep dataset dissimilarity measures (DeDiMs)*. Their motivation is to asses a distribution mismatch between labeled and unlabelled data in Semi-supervised deep learning (SSDL) and therefore to quantify the difference between datasets. In total, four distances are defined: two Minkowski-based distance measures and two nonparametric density-based dataset divergence measures. The general steps presented for calculation, given two datasets $\mathcal{D}^a$ and $\mathcal{D}^b$, are as follows:

1. Draw a random subsample of $\mathcal{D}^a$ and $\mathcal{D}^b$ of size τ and denote these subsamples as $\mathcal{D}^{a,\tau}, \mathcal{D}^{b,\tau}$.
2. Transform the observation $x_{i\bullet} \in \mathbb{R}^p$ of dataset $i \in \{a, b\}$ using a feature extractor g to obtain the feature vector $h_i = g(x_{i\bullet}) \in \mathbb{R}^{p'}$. This yields the feature sets $H^{a,\tau}, H^{b,\tau}$.

For calculating the Minkowski-based distance sets, afterward, the following steps are performed:

1. Calculate $\hat{d}_i = \min_k \|h_i - h_k\|_q$ for $q = 1$ (Manhatten distance) or for $q = 2$ (Euclidean distance) for each of $\mathcal{C}$ samples h_i of $H^{a,\tau}$, where h_k is the closest feature vector from $H^{b,\tau}$. This yields a list of distances $d_{\ell_q}(\mathcal{D}^a, \mathcal{D}^b, \tau, \mathcal{C}) = \left\{\hat{d}_1, \ldots, \hat{d}_\mathcal{C}\right\}$.
2. Calculate a reference list of distances for the same samples of the dataset $\mathcal{D}^a$ to itself (intra-dataset distance) $d_{\ell_q}(\mathcal{D}^a, \mathcal{D}^a, \tau, \mathcal{C}) = \left\{\check{d}_1, \ldots, \check{d}_\mathcal{C}\right\}$.
3. Calculate the absolute differences between reference and inter-dataset distances $d_c = |\hat{d}_c - \check{d}_c|$ as well as their average reference subtracted distance $\bar{d}$ and the p-value of a Wilcoxon test on these differences.

This approach can be seen as a method based on inter-point distances. For the calculation of the density-based distances, the following steps are performed instead:

1. Compute the normalized histogram for each dimension $r = 1, \ldots, p'$ in the feature space to approximate the density function $f_{r,a}$ based on $H^{a,\tau}$ and $f_{r,b}$ based on $H^{b,\tau}$.
2. Compute the sum of the dissimilarities between the density functions $f_{r,a}$ and $f_{r,b}$ for the Jensen-Shannon divergence (d_{JS}) or the cosine distance (d_C): $\hat{d}_i = \sum_{r=1}^{p'} \delta_g(f_{r,a}, f_{r,b})$, $g = JS, C$ for all $\mathcal{C}$ samples (assumption: variables are statistically independent).
3. Compute the intra-dataset distances $\check{d}_1, \ldots, \check{d}_\mathcal{C}$.
4. Calculate the absolute differences between reference and inter-dataset distances $d_c = |\hat{d}_c - \check{d}_c|$ as well as their average reference subtracted distance $\bar{d}$ and the p-value of a Wilcoxon test on these differences.

This approach can be seen as a method based on comparing density functions or as a divergence. The dissimilarity measures do not fulfill the conditions of a metric or pseudo-metric since the distance of a dataset to itself is in general not exactly zero and symmetry properties are not fulfilled. The distances are evaluated in a simulation study with regard to their ability to detect a distribution mismatch and to increase SSDL performance. Both goals are achieved.

Distance based on optimal transport Alvarez-Melis and Fusi (2020) define a distance between datasets relying on optimal transport. They motivate the need for such distances by stating that methods to combine, adapt, and transfer knowledge across datasets need a notion of distance between datasets while "the notion of distance between datasets is an elusive one, and quantifying it efficiently and in a principled manner remains largely an open problem". They criticize that current methods to quantify the distance of two datasets are often heuristic, and highly dependent on tuning and on the architecture of a certain task. Also, many of the other proposals do not take the target variable into account. Therefore, Alvarez-Melis and Fusi (2020) propose a new distance between datasets that is model-agnostic, does not involve training, can compare datasets even if their label sets are disjoint, and has a theoretical footing. Their empirical results also show a good correlation with how hard a transfer-learning task is.
The definition of their distance heavily relies on the optimal transport (OT) problem. Therefore, we define this first in the following. Consider a complete, separable metric space $\mathcal{X}$ and a probability measures $\alpha, \beta \in \mathcal{P}(\mathcal{X})$. The optimal transport according to Kantorovitch (1958) is defined as

$$\mathrm{OT}(\alpha, \beta) := \min_{\pi \in \Pi(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y)\, \mathrm{d}\pi(x, y),$$

where $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is a cost function, the so-called ground cost, and

$$\Pi(\alpha, \beta) := \{\pi_{1,2} \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) | \pi_1 = \alpha, \pi_2 = \beta\}$$

is the set of joint distributions over the product space $\mathcal{X} \times \mathcal{X}$ with marginal distributions α and β. If $\mathcal{X}$ is provided with a metric $d_{\mathcal{X}}$, it is natural to use this as ground cost. In the special cases of $c(x, y) = d_{\mathcal{X}}(x, y)^q$ with $q \geqslant 1$, the term

$$W_q(\alpha, \beta) := \mathrm{OT}(\alpha, \beta)^{1/q} \tag{11}$$

is the q-Wasserstein distance, for $q = 1$ also called Earth Mover's Distance. Finite samples as usually given in practice implicitly define discrete measures for which the pairwise cost can be represented as a cost matrix. The OT then becomes a linear program. Solving this is often difficult due to its cubic complexity. The entropy-regularized problem

$$\mathrm{OT}_\varepsilon(\alpha, \beta) := \min_{\pi \in \Pi(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{X}} c(x, y)\, \mathrm{d}\pi(x, y) + \varepsilon H(\pi\alpha \otimes \beta),$$

where $H(\pi\alpha \otimes \beta) = \int \log(\mathrm{d}\pi/\,\mathrm{d}\alpha\,\mathrm{d}\beta)\,\mathrm{d}\pi$ is the relative entropy and ε gives a time vs. accuracy trade-off, is more efficient to solve. Based on this, the *Sinkhorn divergence* (Genevay, Peyre and Cuturi, 2018)

$$\mathrm{SD}_\varepsilon(\alpha,\beta) = \mathrm{OT}_\varepsilon(\alpha,\beta) - \frac{1}{2}\mathrm{OT}_\varepsilon(\alpha,\alpha) - \frac{1}{2}\mathrm{OT}_\varepsilon(\beta,\beta)$$

can be calculated.

Alvarez-Melis and Fusi (2020) define a dataset $\mathcal{D}$ as a set of feature-label pairs $z := (x,y) \in \mathcal{X} \times \mathcal{Y} =: \mathcal{Z}$ over a feature space $\mathcal{X}$ and a label set $\mathcal{Y}$. They focus on classification and therefore assume $\mathcal{Y}$ to be finite. Moreover, for simplicity, it is assumed that two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ are given whose feature spaces have the same dimensionality. It is not required as an assumption, but Alvarez-Melis and Fusi (2020) find it useful to think of samples in datasets as being drawn from joint distributions $F_1(x,y)$ and $F_2(x,y)$.

To define the distance without relying on external models or parameters, a metric on $\mathcal{Z}$ is needed. Given metrics on $\mathcal{X}$ and $\mathcal{Y}$ one could define $d_\mathcal{Z}(z,z') = (d_\mathcal{X}(x,x')^q + d_\mathcal{Y}(y,y')^q)^{1/q}$ for $q \geqslant 1$, but $d_\mathcal{Y}$ is rarely readily available. Since information about the occurrence of y in relation to feature vectors x is given, instead the metric in $\mathcal{X}$ can be used to compare labels. Let

$$N_\mathcal{D}(y) := \{x \in \mathcal{X} | (x,y) \in \mathcal{D}\}$$

be the set of feature vectors with label y and $n_y = |N_\mathcal{D}(y)|$ its cardinality. The labels are to be represented by their distribution over the feature space $y \mapsto \alpha_y(X) := \mathbb{P}(X|Y=y)$. The set $N_\mathcal{D}(y)$ can be understood as a finite sample of that. That given, choosing a distance between labels is equal to choosing a divergence between the associated distributions. Alvarez-Melis and Fusi (2020) propose OT as an ideal choice since it yields a true metric, it is computable from finite samples, and it is able to deal with sparsely supported distributions. $d_\mathcal{X}^q$ can be used as the optimal transport cost which results in the q-Wasserstein distance $W_q^q(\alpha_y, \alpha_{y'})$ (see (11)) between labels. With this, the distance between feature-label pairs can be defined as

$$d_\mathcal{Z}(z,z') := (d_\mathcal{X}(x,x')^q + W_q^q(\alpha_y,\alpha_{y'}))^{1/q}.$$

This distance can be used in optimal transport to finally define a distance between measures (i.e. datasets):

$$d_{\mathrm{OT}}(\mathcal{D}_1,\mathcal{D}_2) = \min_{\pi \in \Pi(\alpha,\beta)} \int_{\mathcal{Z}\times\mathcal{Z}} d_\mathcal{Z}(z,z')^q \,\mathrm{d}\pi(z,z')$$

This defines a true metric on $\mathcal{P}(\mathcal{Z})$ which Alvarez-Melis and Fusi (2020) call the *Optimal Transport Dataset Distance (OTDD)*.

There are different approaches to represent the distributions α_y, depending on the size of the dataset. In the first approach, the samples in $N_\mathcal{D}(y)$ can be treated as support points of a uniform empirical measure so that $\alpha_y = \sum_{x \in N_\mathcal{D}(y)} \frac{1}{n_y}\delta_x$. When applying this, in every evaluation of $d_\mathcal{Z}(z,z')$ an OT problem needs to be

solved which yields a total worst-case $\mathcal{O}(N^5 \log N)$ complexity and makes this approach only feasible for small to medium-sized datasets. For these, e.g. when $p \gg N$, in simulations for $N \lesssim 5000$, it might even be faster than a proposed second approach. For this second approach, each α_y is modeled as a Gaussian $N(\hat{\mu}_y, \hat{\Sigma}_y)$ with $\hat{\mu}_y$ the sample mean and $\hat{\Sigma}_y$ the covariance of $N_{\mathcal{D}}(y)$. Then, the 2-Wasserstein distance has an analytic form, known as Bures-Wasserstein distance. The distance defined using this approach is denoted as $d_{\mathrm{OT}-\mathcal{N}}$ or Bures-OTDD. It might be the only feasible approach for $n \gg p$ very large.
It holds $d_{\mathrm{OT}-\mathcal{N}}(\mathcal{D}_1, \mathcal{D}_2) \leqslant d_{\mathrm{OT}}(\mathcal{D}_1, \mathcal{D}_2) \leqslant d_{\mathrm{UB}}(\mathcal{D}_1, \mathcal{D}_2)$ for any two datasets, where d_{UB} is a distribution-agnostic OT upper bound defined by the OT distance using a certain cost function. For datasets of sizes n_1 and n_2 with k_1 and k_2 classes, dimension p and maximum class size m, both distances cause costs of $\mathcal{O}(n_1 n_2 \log(\max\{n_1, n_2\})\tau^{-3})$ for solving the outer OT problem τ-approximately, and the worst-case complexity for computing label-to-label pairwise distances is $\mathcal{O}(n_1 n_2(p + m^3 \log m + pm^2))$ for d_{OT} and $\mathcal{O}(n_1 n_1 p + k_1 k_2 p^3 + p^2 m(k_1 + k_2))$ for $d_{\mathrm{OT}-\mathcal{N}}$. Under more assumptions and simplifications, additional speed-ups are possible. To speed up the calculations it is also possible to use the Sinkhorn divergence with approximate OT solution for the inner OT problem.
Alvarez-Melis and Fusi (2020) suggest assessing how realistic assumptions such as the use of Gaussian distributions or the choice of the entropy regularization parameters are before using their method, in order to avoid an unreliable distance estimation. The OTDD can alternatively be seen as an inter-point distance-based method.

3.12. Comparison based on summary statistics

DataSpheres Johnson and Dasu (1998) aim to develop a fast, inexpensive method for massive high-dimensional datasets that does not rely on any distributional assumptions. The idea is to generate a so-called *DataSphere* (map of the dataset) which is a summary of the data, and compare these DataSpheres. The DataSphere can be generated in two passes over the data and can also be further aggregated. It partitions data into sections and represents each section through a set of summaries, which Johnson and Dasu (1998) call *profiles*. Then, tests for these profiles can be used to determine which datasets changed and where. For these tests, a set of weaker hypotheses that only need the profile information is used instead of testing if the joint distribution of the variables is the same for the two datasets.
For the construction of the DataSpheres, the following assumptions are made for the dataset $\mathcal{D}^T = (X_{1\bullet}, \ldots, X_{n\bullet})$: each $X_i \in \mathbb{R}^{p+1}$ with $p = v + c$ consists of $p + 1$ attributes of which c are categorical, v are value attributes and one attribute is the dataset membership with value 1 or 2. Let $S_j = \{X_{i\bullet}$: dataset membership attribute has value $j\}$, $j \in \{1, 2\}$ and let C be a particular value of the categorical variables in D. The *subpopulation* $\mathcal{D}[C_j]$ is defined as the tuples in $\mathcal{D}$ that have value C in their categorical features and value j in their

dataset membership attribute. $V[C_j]$ is defined as the projection of $\mathcal{D}[C_i]$ to the value attributes. Now, for each value of C that is present in D, it is examined if the distribution of $V[C_1]$ differs significantly from the distribution of $V[C_2]$. Therefore, $\mathcal{D}$ is partitioned into K *layers* $\{\mathcal{D}_\ell\}_{\ell=1}^K$ that are more homogeneous than the entire dataset. This is achieved by defining each layer as a set of data points that are within the same (Mahalanobis) distance range from a center of the data cloud (defined as the vector of trimmed means). The cutoffs for the ranges are defined using a fast approximative quantiling algorithm, so each layer contains the same number of data points. Additionally, directional information is included through the use of *pyramids*: a d dimensional set can be partitioned into $2d$ pyramids $P_{\ell\pm}, i = 1, \ldots, d$

$$P_{\ell+} = \{X_{i\bullet} : |\tilde{x}_{i\ell}| > |\tilde{x}_{ij}|, \tilde{x}_{i\ell} > 0, j = 1, \ldots, d, j \neq \ell\}$$
$$P_{\ell-} = \{X_{i\bullet} : |\tilde{x}_{i\ell}| > |\tilde{x}_{ij}|, \tilde{x}_{i\ell} < 0, j = 1, \ldots, d, j \neq \ell\}$$

with $\tilde{x}$ the normalized vectors. The tops of all pyramids meet at the center of the data cloud. A *section* $S(\mathcal{D}_\ell, P_{\ell\pm}, C)$ is now defined as the data points with categorical attributes C such that the value attributes lie in layer $\mathcal{D}_\ell$ and pyramid $P_{\ell\pm}$. Sections are summarized through sets of statistics, called profiles $P(\mathcal{D}_\ell, P_{\ell\pm}, C)$. For a dataset comparison the number of data points, the vector of means of value attributes, and the covariance matrix are used as statistics in the profile. A collection of profiles is called *data map* of a dataset. A data map can be seen as a representation of the dataset.

The authors propose the use of two different tests. The first test is the *multinomial test for proportions*. It compares the proportion of points falling into each section within a subpopulation. The second test is the *Mahalanobis* D^2 test (same as Hotelling's test), which is used to establish the closeness of the multivariate means of each layer within each subpopulation for the two datasets. Both tests are described in detail by Rao (1973). Two different tests are used since for passing the tests it is sufficient but not necessary that the joint distribution in the two datasets is the same.

Constrained minimum (CM) distance Tatti (2007) defines a distance of two datasets that is based on summary statistics but also takes into account their correlation. The so-called Constrained Minimum (CM) Distance can be computed in cubic time. Tatti (2007) lists several properties that a distance of datasets should fulfill: First of all, it should be a metric since metric theory is a well-known area and metrics have many theoretical and practical advantages. It also should take the statistical nature of the datasets into account, e.g. the distance should approach zero for an increasing number of data points when both datasets are generated from the same distribution. Finally, it should be quick to evaluate since data may be high dimensional. Motivated by these requirements the CM distance is defined.

For this, first, define a *feature function* $S : \mathcal{X} \to \mathbb{R}^m$ that maps points from the sample space $\mathcal{X}$ to a real vector. The *frequency* $\theta \in \mathbb{R}^m$ of S with respect to

dataset $\mathcal{D}$ is the average of the values of S

$$\theta = \frac{1}{N}\sum_{i=1}^{N} S(X_{i\bullet}).$$

Let $\mathcal{P}$ be the set of all distributions on $\mathcal{X}$. Then a distribution $F \in \mathcal{P}$ satisfies the frequency θ if $\mathbb{E}_F(S) = \theta$. Assume that the points in $\mathcal{X}$ can be enumerated as $\mathcal{X} = \{1, 2, \ldots, |\mathcal{X}|\}$. Then each distribution $F \in \mathcal{P}$ can be represented by a vector $u \in \mathbb{R}^{|\mathcal{X}|}$ with elements $u_i = f(i)$. Define a *constrained space*

$$\mathcal{C}(S,\theta) = \left\{ u \in \mathbb{R}^{|\mathcal{X}|} \,\middle|\, \sum_{i\in\mathcal{X}} S(i)u_i = \theta, \sum_{i\in\mathcal{X}} u_i = 1 \right\}$$

of distributions satisfying θ. Then, interpreting the distributions as geometrical objects, $\mathcal{C}(S,\theta)$ is an affine space since the constraints defining it are vector products. This implies that the constrained spaces for two different frequencies θ_1 and θ_2 are parallel. The distance between two parallel affine spaces can be measured by the shortest segment going from a point in the first space to a point in the second space, and this segment can be found by taking the points from both spaces that have the shortest norm. Motivated by this, the *Constrained Minimum (CM) Distance* is defined as follows. Given two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ pick a vector from each constrained space having the shortest norm

$$u_i = \arg\min_{u\in C(S,S(\mathcal{D}_i))} \|u\|_2, i = 1, 2$$

and define the CM distance between the datasets as

$$D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|S) = \sqrt{|\mathcal{X}|}\|u_1 - u_2\|_2.$$

The vectors u_1 or u_2 may have negative elements, thus the CM distance is not a distance between two distributions but rather a distance based on the frequencies of a given feature function motivated by the geometrical interpretation of the distribution sets. For calculation purposes, the CM distance can be rewritten as

$$D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|S)^2 = (\theta_1 - \theta_2)^T \mathbb{C}\mathrm{ov}^{-1}(S)(\theta_1 - \theta_2)$$

with

$$\mathbb{C}\mathrm{ov}(S) = \frac{1}{|\mathcal{X}|}\sum_{\omega\in\mathcal{X}} S(\omega)S(\omega)^T - \left(\frac{1}{|\mathcal{X}|}\sum_{\omega\in\mathcal{X}} S(\omega)\right)\left(\frac{1}{|\mathcal{X}|}\sum_{\omega\in\mathcal{X}} S(\omega)\right)^T.$$

The CM distance fulfills the following properties: $D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|S)$ is a pseudo metric. If $\mathcal{D}_1$ and $\mathcal{D}_2$ have the same number of items and $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ are datasets with the same features, then $D_{\mathrm{CM}}(\mathcal{D}_1 \cup \mathcal{D}_3, \mathcal{D}_2 \cup \mathcal{D}_3|S) = (1 - \varepsilon)D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|S)$ with $\varepsilon = \frac{|\mathcal{D}_3|}{|\mathcal{D}_1|+|\mathcal{D}_3|}$. This means that adding external data to the original datasets makes the distance smaller. Furthermore, adding extra

features cannot decrease the distance. Also, for $T(\omega) = AS(\omega) + b$ with an invertible $N \times N$ matrix A and a vector $b \in \mathbb{R}^N$, it holds that $D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|T) = D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|S)$.
Proposals for the choice of a feature function S are means of features or means and pairwise correlations or frequent itemsets.
For binary data and S chosen as the conjunction function, i.e. S is one if all components of an observation are one, and zero otherwise, or as the parity function, i.e. S is one if an odd number of components of an observation are one, and zero otherwise, the CM distance reduces to a more simple form. In these cases, it can be calculated as

$$D_{\mathrm{CM}}(\mathcal{D}_1, \mathcal{D}_2|S) = 2\|\theta_1 - \theta_2\|_2.$$

Note that the factor $\sqrt{2}$ instead of 2 that is stated in the original publication of Tatti (2007) in formula (4) and in Example 3 is not correct as can be seen from the proof of Lemma 8 from which these formulas follow. From $\mathbb{E}(S^2) = \mathbb{E}(S) = 0.5$, it follows that $\mathbb{V}\mathrm{ar}(S) = \mathbb{E}(S^2) - \mathbb{E}(S)^2 = 0.25$ and therefore $\mathbb{C}\mathrm{ov}(S) = 0.25I$ instead of $0.5I$ as claimed in Lemma 8.

3.13. Different testing approaches

General Bootstrap test Romano (1989) studies the asymptotic behavior of some nonparametric tests and shows that under fairly general conditions Bootstrap and randomization tests are equivalent (i.e. the difference in critical functions evaluated at the observed data tends to 0 in probability). The results hold for general applications and the k-sample problem is only one application among others. A very general test statistic for k-sample problems is presented. Its exact form is not specified. The test of Bickel (1969) is a special case for p-dimensional data and $k = 2$, the KS test is a special case for $p = 1$ and $k = 2$. Romano (1989) shows consistency for Bootstrap and permutation tests under some assumptions on the weights for the test statistic and on the distributions of the data.

Weighted Bootstrap test Burke (2000) designs a test using a weighted Bootstrap method based on independent random variables instead of sampling from the uniform distribution. Additionally, uniform confidence bands for the distribution function of multivariate data are constructed. Asymptotically consistent multivariate versions of the KS test and the Cramér-von Mises test are proposed.

Test based on projections I Ping (2000) considers the two- and k-sample problem. Projection pursuit-type statistics are used to overcome the sparseness of data points in high-dimensional space. The limiting distributions of the test statistics are not tractable and depend on the underlying distribution. Therefore, the properties of a Bootstrap approximation are examined. An approximation for statistics based on a number theoretic method is used for computational

reasons. This number-theoretic method chooses directions for projections from the unit sphere. The presented tests are projection versions of the KS-test, CvM-test, and Anderson test. For the theoretical results, only continuous distributions are considered. Consistency is discussed implicitly by proving that the test statistics tend to infinity with probability one as $n_i \to \infty$.

Test based on empirical Bayes factors In Chen and Hanson (2014), empirical Bayes factors constructed from independent Polya tree priors are proposed as a test statistic for the two-sample problem. From this, p-values can be obtained by permuting the group membership indicator. The test was proposed to test whether data distributions are the same across several subpopulations. Initially, it was designed for univariate distributions only but an extension to multivariate distributions is also provided. Both versions are applicable to the k-sample problem. The goal of Chen and Hanson (2014) is to design a test that performs almost as well as the t-test for approximately normal data, but substantially better for non-normal data. Their test statistic is the ratio of marginal densities under H_1 and H_0. The permutation test rejects H_0 for large values of the test statistic. In the limiting case, the test corresponds to the likelihood ratio test based on normal data. For approximately normal data, it behaves similarly to a t-test but pronounced data-driven deviations from normality are also taken into account. Chen and Hanson (2014) are able to give the exact closed-form expression for the marginal density due to the conjugate property of the Polya tree. However, this prior is only suitable for continuous data. Chen and Hanson (2014) center the Polya tree at the normal distribution since they assume that "many datasets are approximately normal, and therefore centering at normal can improve power compared to other nonparametric models that assume nothing". The test of Chen and Hanson (2014) extends the former approaches of Holmes et al. (2015) and Ma and Wong (2011) to the k-sample problem and to censored data. According to simulations, their new test has higher power. For testing, several parameters are chosen via heuristics. The computational cost is $\mathcal{O}(pN^2)$ in the multivariate case. According to Chen and Hanson (2014), in their examples computing permutation p-values took less than 5 minutes in each case, using `R` on an "old Windows-based laptop". For Bayes factors based on an infinite Polya tree, posterior consistency can be shown.

Projections obtained by maximization of a smooth test statistic Zhou, Zheng and Zhang (2017) propose a test that modifies Neyman's smooth test and extends it to the multivariate case based on projection pursue. They use a Bootstrap method to compute the critical value. Similar to Ghosh and Biswas (2016), they apply the idea that H_0 is equivalent to $H_0 : u^T X =_d u^T Y \, \forall u \in \mathcal{S}^{p-1}$ with $\mathcal{S}^{p-1}$ denoting the unit sphere in $\mathbb{R}^p$. They assume that the two sample sizes are comparable ($c_0 n_1 \leqslant n_2 \leqslant n_1, c_0 \in (0, 1]$) and that $n_2 \leqslant n_1$. For the projections in the directions of each u vector, Zhou, Zheng and Zhang (2017) use multiple vectors $u \in \mathcal{S}^{p-1}$ and calculate a univariate smooth-type test statistic which is the supremum norm of a vector of means of several orthonormal functions applied to values of the distribution function of u evaluated at the cross product

of u with the observations of the second dataset. The choice of the orthonormal functions remains unclear. The final test statistic is the (scaled) maximum of test statistics for different u vectors. H_0 is rejected for large values. The limiting distribution of the test statistic may not exist, therefore a Gaussian process approximation of the test statistic and its estimator are given. Multiplier Bootstrap is proposed for testing. For the analysis of the test, Zhou, Zheng and Zhang (2017) make the assumption of absolute continuous distribution functions and the assumption that the d orthonormal functions from $[0,1] \to \mathbb{R}$ are twice differentiable, with $d \leqslant n_1$. For $n_2 \to \infty$, additional assumptions on the maximum over the supremum norm of each function and its first and second derivative are made. The assumptions are fulfilled for normalized Legendre polynomials with $d = o((n_2/\log n_2)^{1/9})$ and for a trigonometric series with $d = o((n_2/\log n_2)^{1/4})$. Then the difference between the α level and the type I error of smooth test tends to zero for $n_2 \to \infty$. Moreover, power against local alternatives tends to 1 for $n_2, d \to \infty$, for normalized Legendre polynomials with $d = o(n_2^{1/9})$, and for trigonometric series with $d = o(n_2^{1/4})$. To show that the test asymptotically holds the level α for growing n_1, n_2 and possibly p, two assumptions are required. First, $d \leqslant \min\{n_1, n_2, \exp(C_0 p)\}$ has to hold for some positive constant C_0. Second, a bound for the maximum over the supremum norm of each of the first and second derivatives of orthonormal functions that grows with n_2 is required. The choice of d remains open, and according to Zhou, Zheng and Zhang (2017) in practice an optimal choice of d is also not possible. The computation of the multivariate test statistic requires solving an optimization problem with an ℓ_2-norm constraint. The best optimizer remains unclear.

Test based on ball divergence Pan et al. (2018) introduce a novel measure of the difference between two probability measures in separable Banach spaces, called *Ball Divergence*. The Ball Divergence is defined as the square of the measure difference over a given closed ball collection. It is equal to zero if and only if the probability measures are identical and does not require any moment assumptions. Based on the Ball Divergence, Pan et al. (2018) propose a metric rank test procedure. Its empirical test statistic is defined based on the difference between averages of the metric ranks. It is robust to outliers and heavy-tail data. The distribution of the test statistic converges to a mixture of χ^2 distributions under the null hypothesis, and it converges to a normal distribution with mean 0 and variance depending on the asymptotic proportion of the sample from the first distribution under the alternative hypothesis. The test does not depend on the ratio of sample sizes and thus can also be applied to imbalanced data. Pan et al. (2018) state that existing methods do not take extremely imbalanced data into account.

The newly proposed test relies on the fact that two Borel probability measures are identical if they agree on all balls in a separable Banach space (Preiss and Tišer, 1991). It can be applied for data in separable Banach spaces, which overcomes the limitation that many Banach spaces are not of the strong negative type or even of negative type (e.g. $\mathbb{R}^p$ with ℓ^q metric for $3 \leqslant p \leqslant \infty, 2 < q \leqslant \infty$)

such that e.g. the generalized energy distance is not applicable. The square root of the Ball Divergence is a symmetric divergence, but not a metric since it does not satisfy the triangle inequality. The testing procedure can be generalized further to the k-sample problem. A connection to the MMD and to the energy statistic is shown through a unified framework of variograms. Consistency against any general alternative can be shown without any additional assumptions and independent of the ratio between the smaller and the larger sample size. Li, Hu and Zhang (2022) conclude that the test by Pan et al. (2018) is model-free and not constrained by any arguments. The test is implemented in the `R` package `Ball` (Zhu et al., 2021).

Test based on Jackknife empirical likelihood Wan, Liu and Deng (2018) present a Jackknife Empirical Likelihood (JEL) test that is motivated by the fact that the energy statistic is zero if and only if the two distributions are equal, under the assumption that first moments exist. Wan, Liu and Deng (2018) aim to avoid the problem of an asymptotic distribution that depends on unknown parameters by using the estimated likelihood method to obtain a distribution-free asymptotic behavior. Their test statistic is asymptotically χ^2_1 distributed for any fixed dimension. A Jackknife Empirical Likelihood (EL) is used to circumvent solving nonlinear constraints for a U-statistic as the main obstacle of the EL method. The resulting test statistic is the nonparametric jackknife empirical log-likelihood ratio. To derive its asymptotic distribution, it is assumed that second moments for $\|X - Y\|$ and for the conditional expectations of $\|X - Y\|$, $\|X - X'\|$, $\|Y - Y'\|$ exist. Under these assumptions, it can also be shown that the resulting asymptotic test is consistent against all fixed alternatives. Under additional assumptions on expectations and on the covariance matrices of X and Y, the test is also shown to be consistent against contiguous alternatives $H_1 : F_1 = (1 - \delta_{n_1,n_2})F_2 + \delta_{n_1,n_2}Q$, where Q is a disturbance distribution and $\delta_{n_1,n_2} = \mathcal{O}(N^{-1/2})$.

Test based on projection averaging for Cramér-von Mises statistic Kim, Balakrishnan and Wasserman (2020) introduce a generalization of the Cramér-von Mises test to the multivariate two-sample problem via projection averaging. They show that the test is consistent against all fixed alternatives and minimax rate optimal against a certain class of alternatives. Moreover, it is robust to heavy-tailed data, free of tuning parameters, and computationally efficient even in high dimensions. The test is shown to have comparable power to existing high-dimensional mean tests under certain location models for $p \to \infty$. Kim, Balakrishnan and Wasserman (2020) propose a new metric called *angular distance* as a robust alternative to the Euclidean distance. This solves the problem of the energy statistic that requires that first moments exist, which might be violated for high-dimensional data where outlying observations occur frequently. By introducing the angular distance, a connection to the RKHS approach can be made. The newly proposed test statistic is an unbiased estimate of the squared multivariate Cramér-von Mises statistic and has a simple closed-form expression. It is invariant to orthogonal transformations, nonnegative, and

equal to zero if and only if the distributions are equal. Based on this statistic, a permutation test can be performed. It has the same asymptotic power as the oracle test and asymptotic tests that assume knowledge of the underlying distributions, for fixed and contiguous alternatives. Kim, Balakrishnan and Wasserman (2020) show that the new test has acceptable power in the contamination model while the energy statistic has very low power. They analyze the finite-sample power and prove minimax rate optimality against a class of alternatives that differ from the null in terms of the CvM-distance. They show that the energy test is not optimal in that context. Moreover, they show consistency in the HDLSS setting under certain conditions. It is also shown that the multivariate CvM-distance is a special case of the generalized energy statistic (Sejdinovic et al., 2013) and that it is equal to the MMD associated with the newly introduced angular distance.

Throughout their analysis, Kim, Balakrishnan and Wasserman (2020) make the assumption of $n_1, n_2 \geqslant 2$. The CvM statistic averages over K projections to approximate the integral over the unit sphere involved in the calculation of the CvM statistic. A resulting problem is that in high dimensions exponentially many projections may be required to achieve a certain accuracy. Instead, Kim, Balakrishnan and Wasserman (2020) give a closed-form expression for the squared multivariate CvM-distance that depends on the expected angles between the differences of X and Y under the assumption that $\beta^T X$ and $\beta^T Y$ have continuous distribution functions for λ-almost all β that lie in the p-dimensional unit sphere, where λ is the uniform probability measure on the p-dimensional unit sphere. The asymptotic null distribution of the test statistic is derived, but it is not applicable for calculating critical values. Therefore a permutation test is used instead. Kim, Balakrishnan and Wasserman (2020) show that the test is consistent under fixed alternatives if second moments of conditional expectations of the test statistic are assumed. If the distance between the distributions diminishes as the sample grows, the additional assumption of quadratic mean differentiable families and an assumption on eigenvalues is required to achieve power greater than α. The new test is more robust than the energy distance test since both can be represented as L^2-type differences between distribution functions but the energy distance gives uniform weight to the whole real line while the CvM statistic gives most weight on high-density regions. Moreover, the CvM distance is well-defined without moment assumptions in contrast to the energy distance that requires existing first moments. Kim, Balakrishnan and Wasserman (2020) prove that the permutation test is minimax rate optimal against a class of alternatives associated with the CvM-distance (CvM-distance of at least ε) and that the energy test is not minimax rate optimal in that context. Additionally, consistency under the HDLSS setting is shown under assumptions on the first and second moments. Under the HDLSS setting and additional moment assumptions (equal covariances, different means) and assumptions on the bandwidth parameter of the Gaussian kernel, they also show the equivalence of CvM, energy statistic, and MMD statistic with the Gaussian kernel. The projection averaging approach can also be used for other one-dimensional test statistics like the sign test, Wilcoxon test, and Kendall's tau. Li and Zhang (2020) note that

the test by Kim, Balakrishnan and Wasserman (2020) has cubic computational cost $\mathcal{O}(N^3)$.

Test based on projective ensemble Li and Zhang (2020) construct a robust test through a projective ensemble. The proposed test statistic is a generalization of the Cramér-von Mises statistic that has a simple closed-form expression without tuning parameters. It can be computed in quadratic time and is insensitive to the dimension. Li and Zhang (2020) show that the test based on a permutation procedure for approximating critical values is consistent against all fixed alternatives with rate $\sqrt{N}$. Their test does not require a moment assumption and is robust to outliers. The test is a generalization of the robust projection averaging test by Kim, Balakrishnan and Wasserman (2020) that does not need the continuity assumption. It is also a member of the class of MMD tests. The test statistic is nonnegative and equal to zero, if and only if the distributions are equal. Its limiting distribution is intractable since it depends on the unknown distributions.

Weighted log-rank-type test Liu et al. (2022) present a weighted log-rank-type test for the two- and k-sample problem using class of intensity centered score processes. Their idea is to convert multivariate data into survival data to make use of the powerful weighted log-rank test. The transformation can be viewed as a statistic examining the arrival pattern of data at a certain point in space. The test is computationally simple and applicable to high-dimensional data. Liu et al. (2022) show consistency against any fixed alternative for Kolmogorov-Smirnov-type and Cramér-von Mises-type statistics. Critical values for the tests are obtained by permutations or with a simulation-based resampling method. A regularity condition on the weight function is required as well as the existence of a bounded density for the first distribution. The choice of the weight function and the test set are left open. Three heuristic strategies are presented to choose the test set. Moreover, a (dis)similarity measure for points must be chosen. Typically the Euclidean distance is used for that.

Clustering-based k-sample tests Paul, De and Ghosh (2022a) propose different distribution-free k-sample tests intended for the high dimension low sample size (HDLSS) setting based on clustering the pooled sample. For the tests, first, the pooled sample is clustered using some clustering algorithm suitable for high-dimensional data, and then a contingency table of the cluster and dataset membership is created. The idea behind both tests is that if the datasets come from the same distribution, the cluster and dataset membership are independent while if the datasets come from different distributions, the clustering depends on the true dataset membership. For the first test, the Rand index of the clustering is used as a test statistic (RI test). It is zero when the clustering is perfect, i.e. when the cluster membership is a permutation of the true dataset membership. The Rand index should take higher values when all clusters have similar distributions of class labels. Therefore, $H_0 : F_1 = \cdots = F_k$ is rejected for large values. The critical value can be calculated using a generalized hypergeometric

distribution. Due to the discreteness of the Rand index, Paul, De and Ghosh (2022a) propose to use a randomized test. For the second test, the generalized Fisher's test statistic for $k \times \ell$ contingency tables is used (FS test). It is intended to assess whether there is a dependence between the dataset membership and the cluster. Again, a randomized test using the generalized hypergeometric distribution to find the critical values is proposed. As a clustering algorithm, Paul, De and Ghosh (2022a) suggest using K-means based on the generalized version of the *Mean Absolute Difference of Distances (MADD)*

$$\rho_{h,\varphi}(z_i, z_j) = \frac{1}{N-2} \sum_{m \in \{1,\dots,N\} \setminus \{i,j\}} \left| \varphi_{h,\psi}(z_i, z_m) - \varphi_{h,\psi}(z_j, z_m) \right|,$$

as proposed by Sarkar and Ghosh (2020) for the HDLSS setting. Here, $z_i, i = 1, \dots, N$, denote points from the pooled sample and

$$\varphi_{h,\psi}(z_i, z_j) = h\left(\frac{1}{p} \sum_{i=l}^{p} \psi |z_{il} - z_{jl}| \right),$$

where $h : \mathbb{R}^+ \to \mathbb{R}^+$ and $\psi : \mathbb{R}^+ \to \mathbb{R}^+$ are continous and strictly increasing functions. Paul, De and Ghosh (2022a) consider $h(t) = t$ and $\psi(t) = 1 - \exp(-t)$ for their examples. The number of clusters has to be chosen in advance for the RI and FS tests. A natural choice is to set the number of clusters to k. Paul, De and Ghosh (2022a) also present modified versions of the test where the number of clusters is estimated from the data using the Dunn index (MRI, MFS test). Setting the number of clusters to k might fail in the case of multimodal distributions. In that case, a larger number of clusters might be required where then multiple clusters can correspond to one dataset. Moreover, multiscale versions of the tests are presented (MSRI, MSFS test) for the case where the number of clusters is unclear. The RI or FS tests are then performed for different numbers of clusters and the results are aggregated using a Bonferroni adjustment for the individual tests. An upper limit for the number of clusters to be considered must be chosen. Under certain moment assumptions and assumptions on the functions h and ψ and on the sample size, consistency of the tests under the HDLSS setting (i.e. $p \to \infty$) is shown. The sample size requirements can already be fulfilled for very low sample sizes like $n_i = 4$, depending on the α level and the balance of the sample sizes. Slightly different assumptions are required for the RI, FS / MRI, MFS / MSRI, and MSFS tests. All presented tests are implemented in the `R` package `HDLSSkST` (Paul, De and Ghosh, 2022b).

4. Summary of data similarity methods

In the following, we give a brief summary of each of the ten classes that we divided the methods into, i.e. (i) comparison of cumulative distribution functions, density functions, or characteristic functions, (ii) methods based on multivariate ranks, (iii) discrepancy measures for distributions, (iv) graph-based methods, (v)

methods based on inter-point distances, (vi) kernel-based methods, (vii) methods based on binary classification, (viii) distance and similarity measures for datasets, (ix) comparison based on summary statistics, and (x) different testing approaches.

4.1. Comparison of cumulative distribution functions, density functions or characteristic functions

Since each of the cumulative distribution function, the density function (if it exists), and the characteristic function fully characterizes a distribution, it is natural to compare distributions by one of these functions. Given two datasets for which it is of interest to compare the underlying distributions, empirical versions of the functions can be used.

For univariate distributions, methods of the Kolmogorov-Smirnov (KS) type are particularly popular. They compare the maximal absolute difference of the respective cumulative distribution functions of the two datasets to be compared. The extension of KS-type methods to multivariate distributions is not straightforward. This class includes two generalizations, one that uses permutations (Bickel, 1969) and one that uses partitioning of the sample space (Biau and Gyorfi, 2005).

For the comparison of datasets based on their empirical density functions, different approaches to density estimation are utilized (e.g. kernel density estimation in Ahmad and Cerrito (1993); Anderson, Hall and Titterington (1994); Cao and van Keilegom (2006) or estimation of densities based on partitions in Ntoutsi, Kalousis and Theodoridis (2008); Ganti et al. (1999); Roederer et al. (2001); Wang and Pei (2005)) and the resulting estimates of both samples are then compared using different statistics, e.g. the L^2-norm between the estimates.

For comparison of distributions by characteristic functions, usually some type of distance, e.g. the L^2-norm, between the empirical characteristic functions is used (Alba-Fernández, Ibáñez-Pérez and Jiménez-Gamero, 2004; Alba Fernández, Jiménez Gamero and Muñoz García, 2008; Li, Hu and Zhang, 2022).

4.2. Methods based on multivariate ranks

In the univariate two-sample problem, nonparametric tests based on ranks are popular choices. Since $\mathbb{R}^p$ does not have a natural ordering, the generalization of these methods to the multivariate problem is not straightforward. For the multivariate case, rank-based methods are based either on projecting the multivariate observations to one-dimensional statistics and ranking those (Ghosh and Biswas, 2016) or on multivariate generalizations of ranks based on optimal transport (Ghosal and Sen, 2021; Deb, Bhattacharya and Sen, 2021). Yet another generalization uses graphs to define ranks for multivariate data (Zhou and Chen, 2023).

4.3. Discrepancy measures for distributions

There exist various approaches to measure the discrepancy of two distributions. So-called probability metrics are metrics in the mathematical sense (i.e. they are positive definite, symmetric, and fulfill the triangle inequality) while discrepancy measures that do not fulfill the triangle inequality are usually called semimetrics or pseudometrics. In general, discrepancy measures that may not fulfill all metric properties are known as divergences. The best-known class of probability metrics are integral probability metrics (IPM), also called probability metrics with a ξ-structure (Zolotarev, 1976, 1984), as introduced by Müller (1997). If two distributions are equal, any function has the same expectation under both distributions. Based on this idea, the supremum difference of the integrals under both distributions over functions belonging to a prespecified set of functions is evaluated. The choice of this set of functions determines the IPM. Divergences include the large class of f-divergences, which are also known as Ali-Silvey distances going back to Ali and Silvey (1966) or as Csiszár's Φ-divergences going back to Csiszár (1963). f-divergences use the idea that equal distributions assign the same likelihood to each point. Therefore, they measure how far the likelihood ratio of the distributions is from one by using a convex continuous function f that maps a ratio of one to the value zero. The expectation under the first distribution of this function f applied to the likelihood ratio of the two distributions to be compared is evaluated. The choice of the function f specifies the f-divergence. There are several other subclasses of probability metrics and divergences following a diverse set of approaches (e.g. Rényi, 1961; Zolotarev, 1984; Rachev, 1991; Muñoz et al., 2012; Zhao et al., 2021).

4.4. Graph-based methods

Graph-based methods for comparing distributions are particularly popular in two-sample testing. Most of these fit in the general framework presented for example by Arias-Castro and Pelletier (2016) or Mukhopadhyay and Wang (2020a). The pooled sample consisting of both datasets is used to construct a graph where each data point corresponds to one node. The methods differ in how edges between these nodes are inserted. Then, in most cases, the number of edges that connect points from different datasets, that is the number of adjacent nodes in the graph from different datasets, is counted. One particularly frequently used example is the K-nearest neighbor graph (Weiss, 1960; Friedman and Steppel, 1973; Schilling, 1986; Henze, 1988; Nettleton and Banerjee, 2001; Hall and Tajvidi, 2002; Chen, Chen and Su, 2018; Mondal, Biswas and Ghosh, 2015), where each point in the pooled sample corresponds to one node. An edge connects one node to another if the data point corresponding to the second node is one of the K nearest neighbors of the data point corresponding to the first node, with respect to some distance measure for the data points. If the number of edges connecting points from different datasets is high, points of both datasets are mixed well, so the datasets are similar. If the number is low,

the datasets are separated well, so they are not similar. Since it is unclear in general what constitutes a high or low number, this is usually determined via a permutation approach.

4.5. Methods based on inter-point distances

Many methods for comparing datasets are based on analyzing the distributions of inter-point distances within and between the datasets. A theoretical justification for methods based on inter-point comparisons using a univariate function (e.g. a distance) is given by Maa, Pearl and Bartoszyński (1996). For two datasets, they consider the two distributions of the in-sample comparisons (i.e. $\|X - X'\|$ and $\|Y - Y'\|$ for all pairs of X, X' points from the first dataset and Y, Y' points from the second dataset) and the distribution of the between-sample comparisons (i.e. $\|X - Y\|$). They show that the equality of these three distributions is equivalent to the equality of the distributions of the datasets. This holds in general for discrete distributions. For the continuous case, some restrictions on the density function are needed. These include the existence of expectations and a second condition that is for example fulfilled if one of the densities is bounded or continuous. Based on this theorem, many approaches compare the distributions of the within-sample distances and between-sample distances. The most popular statistic based on this idea is the energy statistic (Zech and Aslan, 2003), which compares the expectations of the distance distributions within and between samples.

4.6. Methods based on kernel (mean) embeddings

Kernel mean embeddings are a standard tool in machine learning. They map probability distributions to functions in so-called reproducing kernel Hilbert spaces (RKHS). Similarity between distributions can then be measured in this RKHS. More precisely, kernel mean embeddings extend feature maps ϕ as used by other kernel methods (e.g. in the context of kernel support vector machines) to the space of probability distributions by representing each distribution F on the feature space $\mathcal{X}$ as a so-called mean function

$$\mu_F(\cdot) := \int_{\mathcal{X}} K(x, \cdot)\, \mathrm{d}F(x) = \mathbb{E}_F(K(X, \cdot)),$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric and positive definite kernel function and $X \sim F$ a random variable defined on $\mathcal{X}$. When well-defined, the kernel mean embedding is essentially a transformation of the distribution F to an element in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ corresponding to the kernel K (Muandet et al., 2017). For characteristic kernels, this representation captures all information about the distribution F. This implies that the distance of the kernel mean embeddings of two distributions, measured in the metric that the RKHS is endowed with, is equal to zero if and only if the distributions coincide (Fukumizu, Bach and Jordan, 2004; Sriperumbudur et al., 2008, 2010).

Therefore, kernel mean embeddings can be used for comparing distributions. The difference of the kernel mean embeddings measured in the RKHS metric is called Maximum Mean Discrepancy (MMD) (Gretton et al., 2006), which is also the most popular method of this class.

4.7. Methods based on binary classification

The idea behind methods in this class is to perform a binary classification of the data points from two given datasets and to evaluate the quality of this classification. More detailed, the data points are labeled with their membership to the first or second dataset, respectively, and then some binary classification rule is fitted to the dataset labels on the pooled sample. If this classification rule performs well (e.g. measured by the classification error) the datasets are considered to be different in some sense, while for datasets that come from the same distribution, it is expected that the classification rule does not perform better than random guessing. To learn the classification rule, various classification methods like random forests or neural networks can be utilized and there are also different proposals on how to evaluate their performance (Yu et al., 2007; Lopez-Paz and Oquab, 2017; Kim, Lee and Lei, 2019; Cheng and Cloninger, 2022; Hediger, Michel and Näf, 2022). Alternatively, some approaches compare the whole (one-dimensional) distributions of scores, e.g. predicted probabilities, obtained from the classification of the data points (Friedman, 2004).

4.8. Distance and similarity measures for datasets

In contrast to defining a distance or similarity measure of the underlying distribution, some methods directly define the distance or similarity of the datasets themselves, using characteristics that are only indirectly connected to the underlying distributions. These methods are in part defined in the context of meta-learning. They use for example the correlation between meta-features like the number of variables in the datasets or other descriptive statistics (Feurer, Springenberg and Hutter, 2015), or the agreement of the performance of different learning algorithms on the datasets (Leite, Brazdil and Vanschoren, 2012; Leite and Brazdil, 2021). Moreover, there are approaches to define distances between datasets by viewing them as metric measure spaces (Mémoli, 2017) or by using optimal transport (Alvarez-Melis and Fusi, 2020).

4.9. Comparison based on summary statistics

The idea behind the methods of this class is to first summarize a dataset using different summary statistics. Then, a distance between these summary statistics is used as the distance between the datasets. This approach is less complex than using potentially complicated distances directly on the datasets, and it can also lead to simpler interpretations.

4.10. Different testing approaches

Comparing two datasets can be seen as a two-sample problem, i.e. testing for equality of their distributions. Many of the methods across all classes are introduced as test statistics for two- or k-sample testing. In this class, more two- and k-sample tests that do not fit in any of the other classes are collected.

5. Approach for comparison of data similarity methods

So far, we classified more than 100 different methods for quantifying the similarity of datasets into ten groups described above. Now, we rate these methods with regard to their applicability, interpretability, and theoretical properties, in order to be able to compare them with each other. This comparison can then facilitate the choice of an appropriate method for the data that researchers have at hand. For this comparison, we introduce 22 different criteria which are explained in the following Section 5.1. The procedure for the comparison of the methods is then described in Section 5.2. Note that the criteria do not include the performance of the methods, e.g. type I error rates and power for two- and k-sample tests, as this is hard to formalize, and for many methods, there are no empirical results yet. Moreover, to our knowledge, there are no neutral comparison studies of the methods yet. Rather, when comparisons are provided, they are usually presented in the context of an article proposing a new method (e.g. Biswas and Ghosh, 2014; Chwialkowski et al., 2015; Mondal, Biswas and Ghosh, 2015; Jitkrittum et al., 2016; Petrie, 2016; Chen and Friedman, 2017; Lopez-Paz and Oquab, 2017; Liu, Li and Póczos, 2018; Liu et al., 2020; Sarkar, Biswas and Ghosh, 2020). Therefore, we focus on criteria that can be judged without performing extensive simulations and leave a neutral comparison of the method performance open for further research.

5.1. Criteria for the comparison of data similarity measures

Applicability Favorable are methods that can be used for general applications. To judge the applicability, we introduce the following criteria.

Does the method allow incorporation of a target variable in a meaningful way?
Many datasets consist of influencing (independent) variables and a target (dependent) variable. Presumably, in most contexts, it is not reasonable to treat this target variable in the same way as the influencing variables. Therefore, dataset similarity measures should also take into account the different role of the target variable. This criterion is counted as fulfilled if the method explicitly accounts for a target variable in the datasets.

Does the method work on numeric data? Does the method work on categorical data?

Numeric and categorical data are often treated differently. Ideally, dataset similarity measures should be able to handle both kinds of data. Each of these criteria is counted as fulfilled if the method is defined for the respective type of data.

Does the method work for datasets that have different numbers of observations?
Some of the methods might not be able to handle different sample sizes. It is desirable that a method can handle differently-sized datasets. The criterion is counted as fulfilled if the method is explicitly defined for datasets of different sizes.

Does the method work if the number of variables exceeds the number of observations?
The case of more variables than observations might be hard to handle due to identifiability as well as the curse of dimensionality. However, since it is a common case in applications like the analysis of high-dimensional gene expression data, data similarity measures that work even for numbers of variables larger than the number of observations might be needed. This criterion is counted as fulfilled if the method can be applied to data where the number of variables is larger than the number of observations. We do not evaluate how well the method works in that case, but only if the measure can be applied at all.

Can the method be used to compare more than two datasets at a time?
In some applications, researchers might be faced with more than two datasets. In that case, it is useful if multiple datasets can be compared at once. In general, it is always possible to extend methods comparing two datasets to the k-sample case for $k > 2$ by aggregating the pairwise comparisons. For this criterion, we check if the method is explicitly defined for more than two datasets.

Can the method be used without a separate training dataset?
In some applications, data can be scarce, e.g. data derived from expensive experiments. In that case, it is undesirable or even impossible to hold out data for training a model involved in the dataset similarity measure. This criterion is counted as fulfilled if the method does not require holding out training data.

Is the method independent of further assumptions?
Further assumptions like continuity of distributions or the existence of certain moments reduce the applicability of a method and are therefore unwanted. This criterion is fulfilled if there are no explicit or implicit assumptions made that are not covered by the other criteria. For example, if the method requires numerical data, this criterion is fulfilled, while it is unfulfilled if continuous data is required.

Is the method free of parameters that need to be chosen or tuned?
Choosing good parameter values often requires good knowledge of the method

and of the datasets at hand and is therefore often a hard task. Thus, for ease of application, it is desirable for users that they do not need to choose parameters prior to applying the method. Default parameters or suggestions on how to choose the parameters are very helpful but do still leave some uncertainty for the parameter choice. Therefore, the criterion is only counted as fulfilled if the method has no free tuning parameters to choose.

Is the method implemented?
Implementation of a method highly increases its applicability for practitioners. This criterion is counted as fulfilled if an implementation in any software is publicly available, e.g. via an `R` package or as code (e.g. in `R`, `matlab`, `python`, or others) in any publicly available repository. Otherwise, we count the criterion as unknown since we cannot guarantee that there is no implementation if we find none. We searched the publications introducing or reviewing the respective methods themselves as well as CRAN (https://cran.r-project.org/) and Bioconductor (https://bioconductor.org/) for implementations of the methods.

What is the computational complexity of the method?
In times of big data, methods with high-cost complexity might be inapplicable. Therefore, a low complexity of the method is desirable. A value for this criterion is given if cost complexity is mentioned in the publications introducing or reviewing the respective methods. For this criterion, we do not decide whether it is fulfilled or not but simply report the complexity if known since in general it is unclear which complexities can be counted as "good". Moreover, usually only the complexity with regard to the number of observations is given while in some applications the number of features might be of higher interest.

Interpretability To judge the result of a dataset comparison, the interpretability of the used measure is very helpful. To rate the interpretability of each measure, we use the following criteria.

Does the measure have interpretable units?
Interpretable units allow the user to judge what an increase in the measure by one unit means. For example for accuracies given as percentages, one unit increase can be interpreted as classifying one additional observation in 100 correctly, or a one unit increase in many graph-based methods can be interpreted as one additional edge that connects points from different samples. In contrast, for example, one unit increase in the L^q metric of the density functions is not interpretable. This criterion is fulfilled if it is intuitively interpretable what an increase of the measure by one unit means.

Is the measure upper bounded? Is the measure lower bound?
Bounds allow us to set the observed value of a measure into context and thus to judge if the observed value represents a low or high similarity or distance, respectively. These criteria are fulfilled if the measure is bounded. If known, the

concrete bounds are provided.

Theoretical properties There are several desirable theoretical properties that a data similarity measure might have.

Is the measure invariant to rotation/ location change/ homogeneous scale transformations?
Invariance under certain transformations can be useful since it might for example allow to rescale or shift both datasets in the same way without influencing the similarity values. The criteria are counted as fulfilled if the respective transformation of the datasets does not change the value of the measure.

Does the measure fulfill the metric properties, i.e. is it positive definite, symmetric, and does it fulfill the triangle inequality?
Metrics are well-known in mathematics and used in many different contexts. The requirement of positive definiteness ensures that a value of zero is attained if and only if the datasets, respectively their distributions, coincide. Symmetry makes sure that the ordering of the datasets, i.e. which one is defined to be the first or second, does not change their similarity. The triangle inequality holds if the sum of the distance of one dataset to a second plus the distance of this second dataset to a third dataset cannot be smaller than the distance directly between the first and third dataset. Again, each of these criteria is fulfilled if the measure fulfills the respective property. For symmetry, this is often obvious even if not explicitly mentioned by the authors. Positive definiteness and the triangle inequality are counted as unknown if they are not explicitly mentioned in the publications introducing or reviewing the respective methods. If the measure is defined for more than two distributions, symmetry and the triangle inequality are checked for the special case of $k = 2$ distributions.

Is the two- or k-sample test based on the data similarity measure consistent?
This criterion is only applicable to methods for which a two- or k-sample test is defined. As such tests are defined for many of the presented methods, the testing performance is of interest. As direct power comparisons of the methods are infeasible, only consistency of the test is considered as it can be assessed without simulations. Following the presented literature, we distinguish between consistency under the usual limiting regime, i.e. $n_i \to \infty$, $n_i/N \to \pi_i \in (0,1), i = 1, \ldots, k$, and p fixed, and high dimension low sample size setup (HDLSS) consistency, i.e. n_i fixed and $p \to \infty$. In almost all cases, some additional assumptions on the distributions or on parts of the test statistic like the graph in graph-based tests or the kernel in kernel-based tests are required. As these differ fundamentally from test to test we only check whether there is some proof of consistency under certain assumptions. In that case, the criterion counts as fulfilled. If there is only proof for the test to not be consistent under the respective limiting regime, it is counted as unfulfilled. If there are known conditions under which it is consistent and known conditions under which it is not, it is counted as conditionally

fulfilled. It is also counted as conditionally fulfilled if consistency is only shown for a certain variant or special case of the test. If there are no statements regarding the consistency of the test in the literature, the criterion is counted as unknown. For methods for which no test is defined, the criterion is counted as inapplicable.

5.2. Method comparison procedure

The comparison of all presented methods is performed as follows. For each method, each of the criteria explained above is checked and the results are tabulated. If a criterion is fulfilled, the method gets a checkmark in the corresponding row of the criterion. If it is not fulfilled, the method gets a cross for that criterion. If it is neither described in the literature nor obvious whether the criterion is fulfilled, the field is left empty (referring to unknown). If a method has free parameters and a criterion is only fulfilled for certain choices of these parameters, the check is given in parentheses. For the lower and upper bounds and the complexity, concrete values are given if known.

In the end, to evaluate how good each method is with regard to our criteria, we count how many criteria are fulfilled, how many are fulfilled conditionally on some free parameters, how many are unfulfilled, and for how many it is unclear. The complexity is not considered in these numbers as it is unclear what a good complexity is in general. The distinction between criteria that are always fulfilled and criteria that are fulfilled for certain parameters allows down-weighting the latter in the comparison. This might be of interest since in many cases there is no single parameter setting that fulfills all properties that can (in principle) be fulfilled by some setting. We analyze which of the methods fulfill most of the criteria as these might be the most promising methods in a general setting. For concrete data at hand, some of the criteria might be irrelevant. To facilitate finding the best suiting method we complement this article with an online tool (`https://shiny.statistik.tu-dortmund.de/data-similarity`) which allows filtering by certain criteria that are relevant to the problem at hand.

6. Results of comparison of data similarity methods

In the following, we present the results of the method comparison. First, we demonstrate the criteria for one example method. Then we give an overview of the results for all methods. Finally, we present a detailed comparison. All figures presented are created using `R` (R Core Team, 2021).

6.1. Example for criteria evaluation: cross-match test

In the following, we check the criteria for one example method, namely the cross-match test statistic (Rosenbaum, 2005). The cross-match test is a graph-based method that uses the optimal non-bipartite matching. The optimal non-bipartite

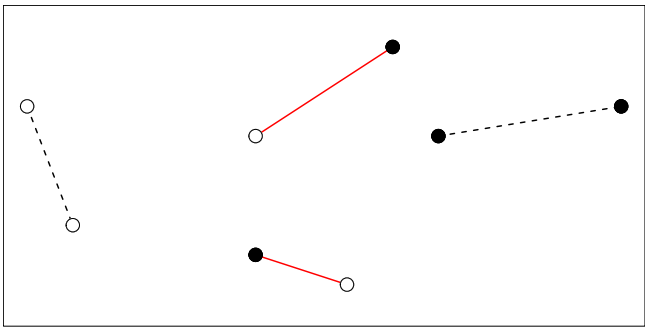


FIG 1. *Optimal non-bipartite matching for the pooled sample of two example datasets. White points correspond to the first dataset, black points correspond to the second dataset. Lines between points indicate edges. Edges between points from different datasets are indicated by red solid lines, and edges between points from the same dataset by black dashed lines.*

matching is the graph where each data point in the pooled sample is connected to exactly one other data point such that the sum over the edge lengths, i.e. the distances between the corresponding points, is minimal. In the case of an odd number of data points, one observation is left out such that the resulting matching has the lowest sum of edge lengths.

Figure 1 shows the optimal non-bipartite matching for an example dataset. The cross-match statistic is given as the number of edges that connect points from different datasets. For testing, the edge count standardized by the expectation and standard deviation under the null is used. For example, in Figure 1, the edges connecting points from different datasets are indicated by red and solid lines. The edge count statistic takes the value two.

We now check the criteria described in Section 5.1 for the cross-match test statistic.

Applicability:

- **Sensible inclusion of target variable?** Since the distances of the observations are taken, all variables are treated the same. ⇒ Unfulfilled
- **Numeric variables?** The test is intended for numeric data. ⇒ Fulfilled
- **Categorical variables?** Categorical data can lead to ties for which the statistic is not uniquely defined. ⇒ Unfulfilled
- **Unequal sample sizes permitted?** The datasets are pooled, so the sample sizes do not play a role in calculating the statistic. ⇒ Fulfilled
- $p > n_i$ **permitted?** Data is transformed into distances. ⇒ Fulfilled (see also Biswas and Ghosh, 2014)
- **Applicable to more than two datasets at a time** ($k > 2$)**?** The statistic is defined for exactly two datasets. ⇒ Unfulfilled
- **No additional training data / train test split required?** The calculation requires no training step. ⇒ Fulfilled
- **No further assumptions on distributions required?** The calculation

indirectly requires the uniqueness of the optimal non-bipartite matching, so no ties are allowed. ⇒ Unfulfilled
- **No tuning / choice of additional parameters required?** There are no additional parameters. ⇒ Fulfilled
- **Implemented in any software?** The cross-match test is implemented in the `R` (R Core Team, 2021) package `crossmatch` (Heller, Small and Rosenbaum, 2012). ⇒ Fulfilled
- **Computational complexity?** The complexity for calculating the optimal non-bipartite matching is $\mathcal{O}(N^3)$, where N denotes the total sample size, i.e. the size of the pooled sample (Rosenbaum, 2005).

Interpretability:

- **Interpretable units?** An increase of one unit for the cross-match statistic can be interpreted as one additional edge in the optimal non-bipartite matching that connects two points from different datasets. ⇒ Fulfilled
- **Lower bound?** If each observation is connected to another observation from the same dataset, the minimum value of zero is attained.
- **Upper bound?** If each observation from the smaller of the two datasets is connected to an observation from the other dataset, the maximum value of $\min\{n_1, n_2\}$ is attained, where n_1 and n_2 are the numbers of observations in the first and second datasets, respectively.

Theoretical properties:

- **Rotation invariant?** Distances are rotation invariant, so the optimal non-bipartite matching and therefore the edge count statistic stays the same under rotation. ⇒ Fulfilled
- **Location change invariant?** Distances are location change invariant, so the optimal non-bipartite matching and therefore the edge count statistic stays the same under location change. ⇒ Fulfilled
- **Scale invariant?** For a change in scale, all distances change by a constant factor, so the optimal non-bipartite matching and therefore the edge count statistic stays the same under scale transformations. ⇒ Fulfilled
- **Positive definite?** For more similar datasets, higher values are expected. ⇒ Unfulfilled
- **Symmetric?** The roles of the first and the second datasets are interchangeable since the data is pooled. ⇒ Fulfilled
- **Triangle inequality?** It is not known whether the triangle inequality is fulfilled.
- **Consistency?** Consistency under the usual limiting regime, $N \to \infty$, $n_i/N \to \pi_i \in (0, 1)$, is shown in the original article of Rosenbaum (2005). ⇒ Fulfilled. There is no proof of HDLSS consistency: ⇒ Unknown

6.2. General insights from overall results

Figure 2 shows a heatmap of all methods and criteria, where the color of each field indicates whether a criterion is fulfilled for the respective method. The methods are ordered first by the highest proportion of fulfilled criteria, then by the highest proportion of conditionally fulfilled criteria, and then by the lowest proportion of unfulfilled criteria. We take into account the proportions instead of absolute numbers of fulfilled criteria since we do not want to give a structural advantage to methods that define a test or are applicable to numeric data, since more criteria can be applied to such methods.

There are many graph-based methods at the top. Apart from this, overall the classes are mostly mixed up. The best method according to the ordering is the nonparametric (kernel) measure of multi-sample dissimilarity (KMD) of Huang and Sen (2023) which fulfills 16 out of 21 criteria. It uses the association between the features and the sample membership to quantify the dissimilarity of multiple distributions. The estimator for KMD is based on a graph in which two points of the pooled sample are connected by an edge if they are close in distance, e.g. the K-nearest neighbor graph. The second best method is the Energy statistic (Zech and Aslan, 2003; Székely and Rizzo, 2017), which is based on inter-point distances and compares the mean of between-sample distances to the means of within-sample distances and fulfills 14 out of 21 criteria and 1 conditionally. Following this method there are three methods that each fulfill 14 out of 21 criteria but none conditionally. These are all graph-based tests, namely the Friedman-Rafksy test (Friedman and Rafsky, 1979) that uses the minimum spanning tree, the cross-match test (Rosenbaum, 2005) that uses the optimal non-bipartite matching, and the graph-based test of Mukherjee et al. (2022) based on optimal non-bipartite matchings that generalizes the cross-match test to categorical data and multiple datasets by using the Mahalanobis distance of a matrix that consists of the pairwise cross-match statistics of all pairs of datasets.

We can see that certain criteria are fulfilled by most of the methods, such as applicability to numeric data, unequal sample sizes, and that there is a lower bound and symmetry. On the other hand, certain other criteria are unfulfilled for most methods, such as the sensible inclusion of a target variable, applicability to more than two datasets at a time, no further assumptions, no tuning parameters, and interpretable units. In many cases, it is unknown if a method is implemented, if it has an upper bound, or if the triangle inequality holds.

Figure 3 shows boxplots of the number of fulfilled criteria for each method, grouped by classes. The median number of fulfilled criteria ranges from five, for methods based on binary classification, to eleven for graph-based methods and methods based on inter-point distances. The number of fulfilled criteria also varies notably within the classes.

The number of fulfilled criteria on its own gives a first idea of the overall performance of the method with regard to our criteria. However, depending on the application, the criteria are not equally important, since some criteria might be mandatory and others negligible. For example, for a dataset comparison where

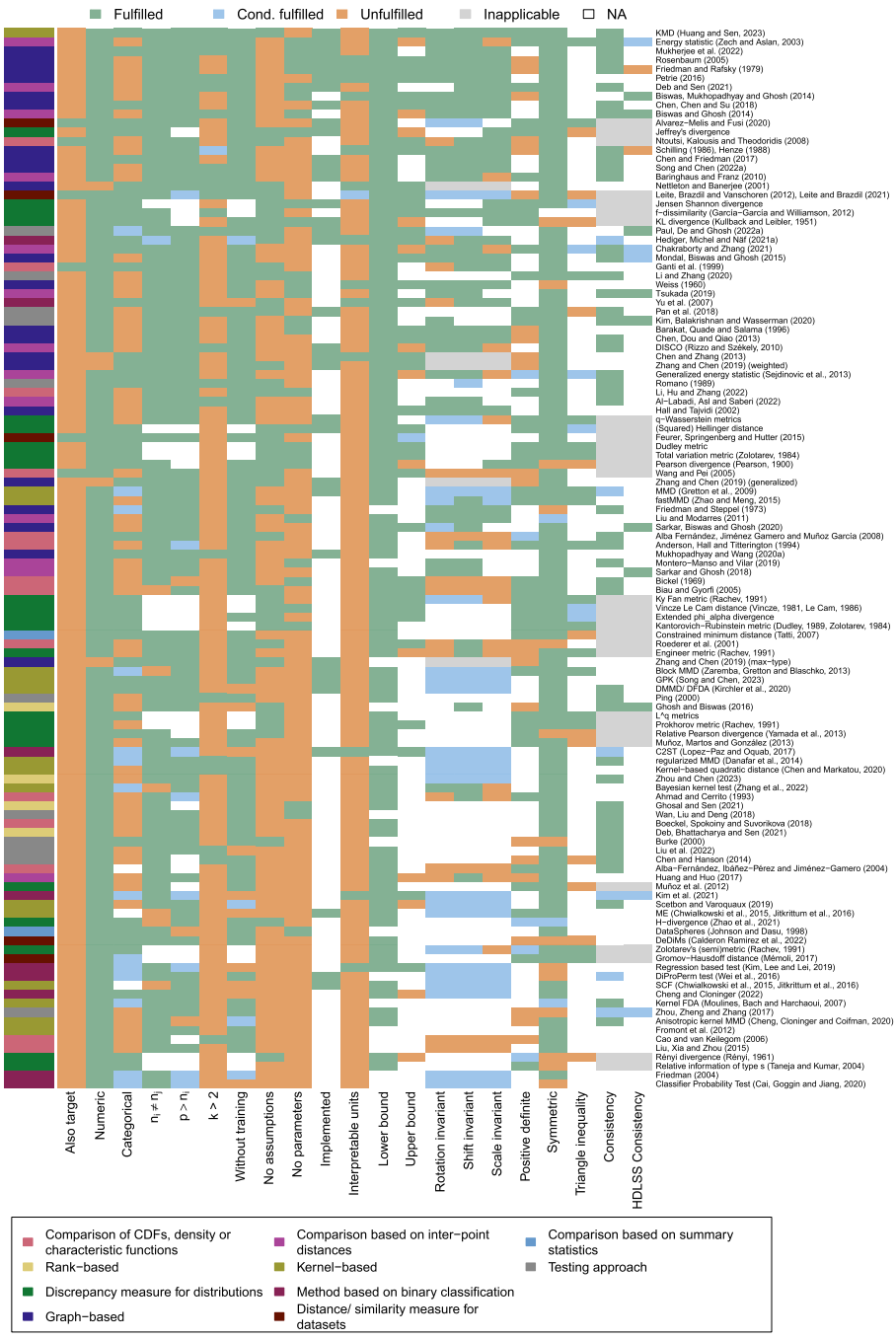


FIG 2. *Comparison of all methods regarding the theoretical criteria.* n_i, $i = 1, \ldots, k$, *denote the sample sizes,* k *denotes the number of datasets to compare,* p *denotes the number of features per dataset.*

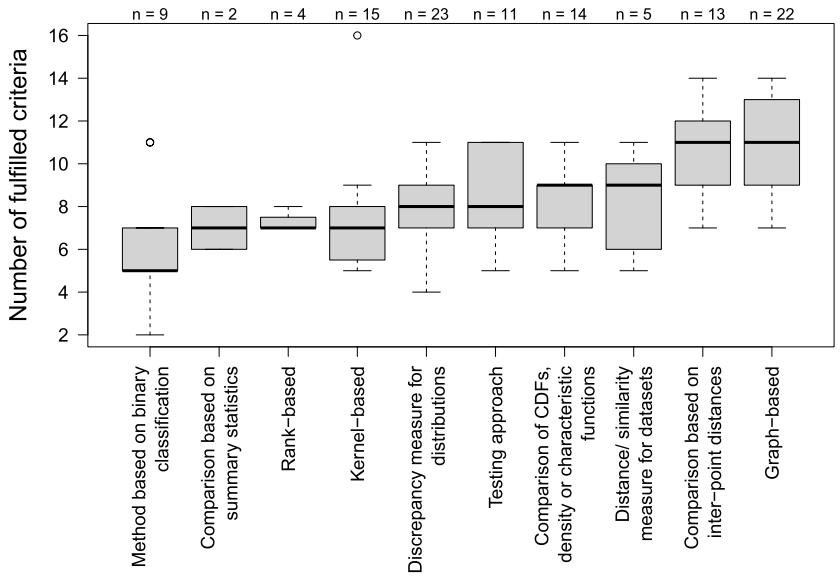


FIG 3. *Comparison of methods regarding the number of fulfilled criteria, grouped by classes. The classes are ordered by the median number of fulfilled criteria. n denotes the number of methods in the respective group.*

some of the variables are numeric and others are categorical, a method that can handle both types of data is required. Further, if the data is not transformed, the invariance properties of a method might not be of interest.

Therefore, in the following section, a detailed list of criteria is given for each method. To facilitate the choice and comparison of suitable methods for a dataset comparison, the online tool (`https://shiny.statistik.tu-dortmund.de/data-similarity`) can be used, which allows filtering and sorting of the tables of Section 6.3 by different criteria. In addition, the tool makes it easy to search for specific methods and it allows users to hide criteria that are not relevant to their application of interest, in order to make the comparison results more concise.

6.3. Detailed method comparison

Tables 2 to 11 show which of the methods fulfill which of our criteria and summarize how many of the criteria are fulfilled or unfulfilled for each method. The cells in the table are filled as explained in Section 5.1. For upper and lower bounds the criterion is fulfilled if a bound is given, all other criteria are fulfilled if they have a checkmark or a checkmark within parentheses. Parentheses around checkmarks mean that parameters can be chosen such that the criterion is fulfilled. Crosses in parentheses mean that for all but single choices the criterion is not fulfilled. Empty fields mean that it is neither described in literature

nor obvious whether the criterion is fulfilled. The complexity is not considered in the calculation of the score so the maximum number of fulfilled criteria is 21. For methods that are inapplicable to numeric data, the transformations of rotating, shifting, or scaling the data are not meaningful. Therefore, the invariance criteria are inapplicable for such methods. This is denoted by a dash. Similarly, consistency does not apply as a criterion to methods that do not define any two- or k-sample procedure. n_i, $i = 1, \ldots, k$ denote the sample sizes, $N = \sum n_i$ denotes the total sample size of the pooled sample, p the number of features, and k the number of datasets.

TABLE 2

Comparison of approaches based on the ***comparison of cumulative distribution functions, density or characteristic functions*** *regarding applicability, interpretability, and theoretical properties.*

Method/ Article	Bickel (1969)	Biau and Gyorfi (2005)	Boeckel, Spokoiny and Suvorikova (2018)	Ntoutsi, Kalousis and Theodoridis (2008)	Ganti et al. (1999)	Roederer et al. (2001)	Wang and Pei (2005)	Ahmad and Cerrito (1993)	Anderson, Hall and Titterington (1994)	Cao and van Keilegom (2006)	Alba-Fernández, Ibáñez-Pérez and Jiménez-Gamero (2004)	Alba Fernández, Jiménez Gamero and Muñoz García (2008)	Liu, Xia and Zhou (2015)	Li, Hu and Zhang (2022)
Also target?	×	×	×	✓	✓	×	✓	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	×	×	×	×	✓	×	×	×	×	×		✓	×	×
$n_i \neq n_j$?	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$p > n_i$?	×	✓	✓	✓	✓	✓	✓	(✓)	(✓)			✓	✓	✓
$k > 2$?	×	×	×	×	×	×	×	×	×	✓	×	×	×	×
Without training?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
No assumptions?	×	✓	×	✓	✓	✓	✓	×	×	×	×	✓	×	✓
No parameters?	✓	×	×	×	×	×	×	×	×	×	×	×	×	✓
Implemented?														
Complexity?	$\mathcal{O}(const^N)$													
Interpretable units?	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Lower bound?	0	0	0	0	0	0	0	0	0		0	0		0
Upper bound?	1	2		1		2	1							
Rotation invariant?	×	×		×	×	×	×	×	×	×	×	×	×	
Location change invariant?	×	×		✓	✓	✓	×	✓	✓	×	×	×	×	
Homogeneous scale invariant?	×	×		✓	✓	×	×	×	×	×	×	×	×	
Positive definite?	✓	✓		×		×	×	✓	✓	×	✓	(✓)		✓
Symmetric?	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓
Triangle inequality?									✓					
Consistency ($n_i \to \infty$)?	✓	✓	✓	–		–	–		✓		✓	✓		✓
Consistency ($p \to \infty$)?				–		–	–							
No. fulfilled criteria	9	9	7	11	11	8	9	7	9	5	7	9	5	10
No. cond. fulfilled criteria	0	0	0	0	0	0	0	1	1	0	0	1	0	0
No. unfulfilled criteria	9	9	6	6	4	9	8	8	8	9	8	7	9	4
No. NAs	3	3	8	2	6	2	2	5	3	7	6	4	7	7

TABLE 3

*Comparison of approaches based on **multivariate ranks** and **probability metrics** regarding applicability, interpretability, and theoretical properties. * assumptions only needed to show consistency for high dimension low sample size (HDLSS) setting.*

Method/ Article	Ghosh and Biswas (2016)	Ghosal and Sen (2021)	Deb, Bhattacharya and Sen (2021)	Zhou and Chen (2023)	Engineer metric (Rachev, 1991)	Zolotarev's (semi)metric (Rachev, 1991)	Ky Fan metric (Rachev, 1991)	Prokhorov metric (Rachev, 1991)	Dudley metric	Total variation metric (Zolotarev, 1984)	Kantorovich-Rubinstein metric (Zolotarev, 1984; Dudley, 1989)	L^q metrics	q-Wasserstein metrics
Also target?	×	×	×	×	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	×	×	×	×	×	✓	✓	×	✓	✓	✓	✓	×
$n_i \neq n_j$?	✓	✓	✓	✓	✓								✓
$p > n_i$?	✓		✓	✓	✓								✓
$k > 2$?	×	×	×	×	×	×	×	×	×	×	×	×	×
Without training?	×	✓	✓	✓	✓								✓
No assumptions?	✓*	×	×	×	×	×	✓	✓	✓	✓	✓	✓	×
No parameters?	×	✓	×	×	×	×	×	×	✓	✓	✓	×	×
Implemented?													✓
Complexity?			$\mathcal{O}(N^3)$										
Interpretable units?	×	×	×	×	×	×	×	×	×	×	×	×	×
Lower bound?		0	0	0	0	0	0	0	0	0	0	0	0
Upper bound?					×		1	1	2	2			
Rotation invariant?				(✓)	×	(✓)	(✓)						(✓)
Location change invariant?	✓			(✓)	✓	(✓)	(✓)						(✓)
Homogeneous scale invariant?				(✓)	×	×	×						×
Positive definite?	×				×	(✓)	✓	✓	✓	✓	✓	✓	✓
Symmetric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Triangle inequality?					✓	✓	✓	✓	✓	✓	✓	✓	✓
Consistency ($n_i \to \infty$)?	✓	✓	✓	✓	–	–	–	–	–	–	–	–	–
Consistency ($p \to \infty$)?	✓				–	–	–	–	–	–	–	–	–
No. fulfilled criteria	8	7	7	7	8	5	8	7	9	9	8	7	9
No. cond. fulfilled criteria	0	0	0	3	0	3	2	0	0	0	0	0	2
No. unfulfilled criteria	7	5	6	6	10	6	5	5	3	3	3	4	7
No. NAs	6	9	8	5	1	5	4	7	7	7	8	8	1

TABLE 4

*Comparison of **divergences** regarding applicability, interpretability, and theoretical properties. * holds for square root transformation. ** holds only in case of $\alpha = 1/2$, resp. $s = 1/2$. *** values calculated using the general formula given in Vajda (2009).*

Method/ Article	(Squared) Hellinger distance	Vincze Le Cam distance (Vincze, 1981; Le Cam, 1986)	KL divergence (Kullback and Leibler, 1951)	Jeffrey's divergence	Extended ϕ_α divergence	Jensen Shannon divergence	Pearson divergence (Pearson, 1900)	Relative Pearson divergence (Yamada et al., 2013)	f-dissimilarity (Györfi and Nemetz, 1975; García-García and Williamson, 2012)	Rényi divergence (Rényi, 1961)	Relative information of type s (Taneja and Kumar, 2004)	H-divergence (Zhao et al., 2021)	Muñoz et al. (2012)	Muñoz, Martos and González (2013)
Also target?	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×
$n_i \neq n_j$?			✓	✓			✓	✓	✓			×	✓	✓
$p > n_i$?												✓		✓
$k > 2$?	×	×	×	×	×	×	×	×	✓	×	×	×	×	×
Without training?			✓	✓	✓		✓	✓				✓	✓	✓
No assumptions?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×	×
No parameters?	✓	✓	✓	✓	×	✓	✓	×	×	×	×	×	×	×
Implemented?	✓		✓	✓		✓								
Complexity?														
Interpretable units?	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Lower bound?	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Upper bound?	4***	1/2***	×	×	✓	log(2)	×		(✓)	×				
Rotation invariant?									✓					
Location change invariant?									✓					
Homogeneous scale invariant?			✓	✓		✓	✓		✓					
Positive definite?	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)	✓	(✓)	✓	✓
Symmetric?	✓	✓	×	✓	✓	✓	×	×		(×)**	(×)**	(✓)	✓	✓
Triangle inequality?	(✓)*	(✓)*	×	×	(✓)*	(✓)*	×	×		×			×	×
Consistency ($n_i \to \infty$)?	–	–	–	–	–	–	–	–	–	–	–		–	–
Consistency ($p \to \infty$)?	–	–	–	–	–	–	–	–	–	–	–		–	–
No. fulfilled criteria	9	8	10	11	8	10	9	7	10	4	4	6	6	7
No. cond. fulfilled criteria	1	1	0	0	1	1	0	0	1	1	0	2	0	0
No. unfulfilled criteria	3	3	6	5	4	3	6	6	3	7	6	5	7	7
No. NAs	6	7	3	3	6	5	4	6	5	7	9	8	6	5

TABLE 5

*Comparison of **graph-based methods** regarding applicability, interpretability, and theoretical properties. * no assumptions mentioned in the original articles, but the method implicitly requires uniqueness of the constructed graph (Chen and Zhang, 2013). ** K minimum number of categories, M number of minimum spanning trees. *** for unstandardized statistic.*

Method/ Article	Friedman and Rafsky (1979)	Rosenbaum (2005)	Chen and Zhang (2013)	Biswas, Mukhopadhyay and Ghosh (2014)	Petrie (2016)	Chen and Friedman (2017)	Chen, Chen and Su (2018)	Zhang and Chen (2019) (weighted)	Zhang and Chen (2019) (generalized)	Zhang and Chen (2019) (max-type)	Sarkar, Biswas and Ghosh (2020)	Mukhopadhyay and Wang (2020a)	Mukherjee et al. (2022)	Song and Chen (2022a)
Also target?	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	✓	✓	✓
Categorical?	×	×	✓	×	×	✓	✓	✓	✓	✓	×	✓	✓	×
$n_i \neq n_j$?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$p > n_i$?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$k > 2$?	×	×	×	×	✓	×	×	×	×	×	×	✓	✓	✓
Without training?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
No assumptions?	×*	×*	×	×*	×	×	×	✓	✓	✓	×	×	×*	×*
No parameters?	✓	✓	×	✓	×	×	×	×	×	×	×	×	✓	×
Implemented?	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓
Complexity?		$\mathcal{O}(N^3)$	$\mathcal{O}(K^2)$, $\mathcal{O}(M)$**	$\mathcal{O}(N^2 \log N)$	$\mathcal{O}(N^2 \log N)$, $\mathcal{O}(N^3)$, $\mathcal{O}(N \log N)$									
Interpretable units?	✓	✓	✓	✓	✓	×	×	×	×	×	×	×	×	×
Lower bound?	2	0***	✓***	1	✓	0	0	0	0		0	0	0	0
Upper bound?	N	$\min\{n_1, n_2\}$***	✓***	$\min(n_1, n_2)$	✓		✓	✓			✓			
Rotation invariant?	✓	✓	–	✓	✓	✓	✓	–	–	–	(✓)		✓	✓
Location change invariant?	✓	✓	–	✓	✓	✓	✓	–	–	–	✓		✓	✓
Homogeneous scale invariant?	✓	✓	–	✓	✓	✓	✓	–	–	–			✓	✓
Positive definite?	×	×	×	×				×	×	×				
Symmetric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Triangle inequality?														
Consistency ($n_i \to \infty$)?	✓	✓				✓	✓						✓	✓
Consistency ($p \to \infty$)?	×			✓							✓			
No. fulfilled criteria	14	14	9	13	13	12	13	9	8	7	9	9	14	12
No. cond. fulfilled criteria	0	0	0	0	0	0	0	0	0	0	1	0	0	0
No. unfulfilled criteria	6	5	6	5	4	5	5	6	6	6	6	4	3	5
No. NAs	1	2	3	3	4	4	3	3	4	5	5	8	4	4

TABLE 6

*Comparison of methods based on **nearest neighbors** regarding applicability, interpretability, and theoretical properties. * only in combination with numerical data.*

Method/ Article	Weiss (1960)	Friedman and Steppel (1973)	Schilling (1986); Henze (1988)	Barakat, Quade and Salama (1996)	Nettleton and Banerjee (2001)	Hall and Tajvidi (2002)	Chen, Dou and Qiao (2013)	Mondal, Biswas and Ghosh (2015)
Also target?	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	×	✓	✓	✓
Categorical?	×	(✓)*	×	×	✓	×	×	×
$n_i \neq n_j$?	✓	✓	✓	✓	✓	✓	✓	✓
$p > n_i$?	✓	✓	✓	✓	✓	✓	✓	✓
$k > 2$?	×	×	(✓)	×	✓	×	×	×
Without training?	✓	✓	✓	✓	✓	✓	✓	✓
No assumptions?	×	✓	×	×	✓	×	×	×
No parameters?	✓	×	×	✓	×	×	×	×
Implemented?								
Complexity?		$\mathcal{O}(2^p N \log_2 N)$, $\mathcal{O}(pN^2)$						
Interpretable units?	✓	×	✓	×	✓	×	×	×
Lower bound?	0		0	0	0	0	0	0
Upper bound?	1		1	$N(n_1(n_1-1)+n_2(n_2-1))/2$ $+n_1 n_2(N-2)/2$	$\min(n_1, n_2)$	✓	1	✓
Rotation invariant?	✓	✓	✓	✓	–	✓	✓	✓
Location change invariant?	✓	✓	✓	✓	–	✓	✓	✓
Homogeneous scale invariant?	✓	✓	✓	✓	–	✓	✓	✓
Positive definite?			×	×			×	
Symmetric?	×	×	✓	✓	✓	✓	✓	✓
Triangle inequality?								
Consistency ($n_i \to \infty$)?		✓	✓				✓	✓
Consistency ($p \to \infty$)?			×					(✓)
No. fulfilled criteria	11	9	12	11	10	10	11	11
No. cond. fulfilled criteria	0	1	1	0	0	0	0	1
No. unfulfilled criteria	5	5	6	6	3	6	7	6
No. NAs	5	6	2	4	5	5	3	3

TABLE 7

*Comparison of methods based on **inter-point distances** regarding applicability, interpretability, and theoretical properties. * m denotes the number of random projections.*

Method/ Article	Energy statistic (Zech and Aslan, 2003)	Generalized energy statistic (Sejdinovic et al., 2013)	Chakraborty and Zhang (2021)	DISCO (Rizzo and Székely, 2010)	Huang and Huo (2017)	Deb and Sen (2021)	Al-Labadi, Asl and Saberi (2022)	Baringhaus and Franz (2010)	Liu and Modarres (2011)	Biswas and Ghosh (2014)	Sarkar and Ghosh (2018)	Montero-Manso and Vilar (2019)	Tsukada (2019)
Also target?	×	×	×	×	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	×	✓	✓	×	×	×	×	×	×	×	×	×	×
$n_i \neq n_j$?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$p > n_i$?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$k > 2$?	✓	×	×	✓	×	✓	✓	✓	×	✓	×	×	×
Without training?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
No assumptions?	×	×	×	×	×	×	×	×	×	×	×	×	×
No parameters?	✓	×	×	×	×	✓	×	×	✓	✓	×	✓	✓
Implemented?	✓			✓		✓		✓					
Complexity?	$\mathcal{O}(N^2)$		$\mathcal{O}(p)$	$\mathcal{O}(N^2)$	$\mathcal{O}(mN \log N)$ $+\mathcal{O}(n_1 n_2 p)$*	$\mathcal{O}(N^3)$			Indep. of p			$\mathcal{O}(N^2 \log N)$	
Interpretable units?	×	×	×	×	×	×	×	×	×	×	×	×	×
Lower bound?	0	0	0	0	0	0	0	0	0	0	0		0
Upper bound?	×	×	×	×	×					×			
Rotation invariant?	✓	✓	✓	✓	×		✓	✓	✓	✓	✓	✓	✓
Location change invariant?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Homogeneous scale invariant?	×	×	×	×	×	✓		×	✓	✓			
Positive definite?	✓	(✓)	✓		×	✓		✓					
Symmetric?	✓	✓	✓	✓	✓	✓	✓	✓	(✓)	✓	✓	✓	✓
Triangle inequality?	✓	(✓)	(✓)										
Consistency ($n_i \to \infty$)?	✓	✓	✓	✓		✓	✓	✓		✓		✓	✓
Consistency ($p \to \infty$)?	(✓)		(✓)							✓	✓		✓
No. fulfilled criteria	14	10	11	11	7	13	10	12	9	13	9	9	11
No. cond. fulfilled criteria	1	2	2	0	0	0	0	0	1	0	0	0	0
No. unfulfilled criteria	6	7	7	7	10	4	5	6	5	5	6	5	5
No. NAs	0	2	1	3	4	4	6	3	6	3	6	7	5

TABLE 8

*Comparison of methods based on **variations of MMD** regarding applicability, interpretability, and theoretical properties. * for block size $B = \lfloor n^\gamma \rfloor$. ** L denotes the number of basis functions for approximating kernels. *** N_R denotes the number of reference points.*

Method/ Article	(Linear) MMD^2 (Gretton et al., 2009; Muandet et al., 2017; Gretton et al., 2012a)	Block MMD (Zaremba, Gretton and Blaschko, 2013)	fastMMD (Zhao and Meng, 2015)	ME (Chwialkowski et al., 2015; Jitkrittum et al., 2016)	SCF (Chwialkowski et al., 2015; Jitkrittum et al., 2016)	regularized MMD (Danafar et al., 2014)	Anisotropic kernel MMD (Cheng, Cloninger and Coifman, 2020)	DMMD/ DFDA (Kirchler et al., 2020)	GPK (Song and Chen, 2023)
Also target?	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	(✓)	(✓)	×			(✓)	×	✓	✓
$n_i \neq n_j$?	✓	×	✓	×	×	✓	✓	✓	✓
$p > n_i$?	✓	✓	✓	✓	✓	✓	×	✓	✓
$k > 2$?	×	×	×	×	×	✓	×	×	×
Without training?	✓	✓	✓	×	×	✓	(✓)	×	✓
No assumptions?	×	×	×	×	×	×	×	×	×
No parameters?	×	×	×	×	×	×	×	×	×
Implemented?	✓	✓		✓	✓			✓	✓
Complexity?	$\mathcal{O}(N^2p)$, $\mathcal{O}(Np)$	$\mathcal{O}(N^{1+\gamma}p)$, $\gamma \in (0,1)$*	$\mathcal{O}(LNp)$, $\mathcal{O}(LN \log p)$**	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(NNN_R)$***	$\mathcal{O}(N)$, $\mathcal{O}(p)$	
Interpretable units?	×	×	×	×	×	×	×	×	×
Lower bound?	0	0	0	0	0		0	0	0
Upper bound?									
Rotation invariant?	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)		(✓)	(✓)
Location change invariant?	(✓)	(✓)	✓	(✓)	(✓)	(✓)		(✓)	(✓)
Homogeneous scale invariant?	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)		(✓)	(✓)
Positive definite?	✓	✓	✓	✓			×		
Symmetric?	✓	✓	✓	✓	✓	✓	✓	✓	✓
Triangle inequality?	✓	✓	✓						
Consistency ($n_i \to \infty$)?	(✓)					✓	✓	✓	
Consistency ($p \to \infty$)?									
No. fulfilled criteria	9	8	9	6	5	7	5	8	8
No. cond. fulfilled criteria	5	4	2	3	3	4	1	3	3
No. unfulfilled criteria	5	6	6	7	7	4	8	6	5
No. NAs	2	3	4	5	6	6	7	4	5

TABLE 9

*Comparison of **kernel-based methods other than MMD** and of methods based on **binary classification** regarding applicability, interpretability, and theoretical properties. * assumptions are only needed for showing properties of the estimator. ** for K-nearest neighbor graph. *** for permutation version.*

Method/ Article	Kernel FDA (Moulines, Bach and Harchaoui, 2007)	Fromont et al. (2012)	Scetbon and Varoquaux (2019)	Kernel-based quadratic distance (Chen and Markatou, 2020)	Bayesian kernel test (Zhang et al., 2022)	KMD (Huang and Sen, 2023)	Friedman (2004)	C2ST (Lopez-Paz and Oquab, 2017)	Regression based test (Kim, Lee and Lei, 2019)	Cheng and Cloninger (2022)	Yu et al. (2007)	DiProPerm test (Wei et al., 2016)	Classifier Probability Test (Cai, Goggin and Jiang, 2020)	Kim et al. (2021)	Hediger, Michel and Näf (2022)
Also target?	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	(✓)	×	×	×	(✓)	✓	(✓)	(✓)	(✓)	✓	✓	(✓)	(✓)	(✓)	✓
$n_i \neq n_j$?	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)
$p > n_i$?	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	✓	✓	✓	(✓)	(✓)	✓
$k > 2$?	×	×	×	×	×	✓	×	×	×	×	×	×	×	×	×
Without training?	✓	✓	(✓)	✓	✓	✓	(✓)	×	×	×	×	✓	×	×	(✓)
No assumptions?	×	×	×	×	×	✓*	×	×	×	×	✓	×	×	×	✓
No parameters?	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
Implemented?						✓		✓				✓			✓
Complexity?						$\mathcal{O}(KN \log N)$**				$\mathcal{O}(N)$					
Interpretable units?	×	×	×	×	×	×	×	✓	✓	×	✓	×	×	✓***	✓***
Lower bound?			0	0	0	0		0	0	×	0			0***	0***
Upper bound?			×			1		1	1	×	1			1***	1***
Rotation invariant?			(✓)	(✓)	✓	✓	(✓)	(✓)	(✓)	(✓)	×	(✓)	(✓)	(✓)	×
Location change invariant?			(✓)	(✓)	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	(✓)	(✓)	(✓)	✓
Homogeneous scale invariant?			(✓)	(✓)	×	✓	(✓)	(✓)	(✓)	(✓)	✓	(✓)	(✓)	(✓)	✓
Positive definite?						✓									
Symmetric?	(✓)	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	×	×	✓	✓
Triangle inequality?															
Consistency ($n_i \to \infty$)?	✓		✓			✓		(✓)				(✓)		(✓)	(✓)
Consistency ($p \to \infty$)?														(✓)	
No. fulfilled criteria	5	5	6	7	7	16	3	7	5	5	11	5	2	6	11
No. cond. fulfilled criteria	2	0	4	3	1	0	6	6	5	3	0	5	5	7	3
No. unfulfilled criteria	5	6	7	5	7	3	5	5	6	8	5	6	7	5	4
No. NAs	9	10	4	6	6	2	7	3	5	5	5	5	7	3	3

TABLE 10

Comparison of ***distance and similarity measures for datasets*** *and* ***comparison based on summary statistics*** *regarding applicability, interpretability, and theoretical properties.* * *combination of numeric and categorical data required.* ** *finite sample space required.*

Method/ Article	Feurer, Springenberg and Hutter (2015)	Gromov-Hausdoff distance (Mémoli, 2017)	Leite, Brazdil and Vanschoren (2012); Leite and Brazdil (2021)	DeDiMs (Calderon Ramirez et al., 2022)	Alvarez-Melis and Fusi (2020)	DataSpheres (Johnson and Dasu, 1998)	Constrained minimum distance (Tatti, 2007)
Also target?	✓	×	✓	×	✓	×	×
Numeric?	✓	✓	✓	✓	✓	✓*	✓**
Categorical?	✓	(✓)	✓	✓	✓	✓*	✓
$n_i \neq n_j$?	✓		✓	✓	✓	✓	✓
$p > n_i$?	✓		(✓)	✓	✓	×	✓
$k > 2$?	×	×	✓	×	×	×	×
Without training?	✓		✓	✓	✓	✓	✓
No assumptions?	✓	×	✓	×	×	×	×
No parameters?	×	×	×	×	×	✓	×
Implemented?					✓		
Complexity?							$\mathcal{O}(N^3)$
Interpretable units?	×	×	(✓)	×	×	×	×
Lower bound?	0	0	0 or −1	0	0		0
Upper bound?	(2)		1				
Rotation invariant?			(✓)		(✓)		
Location change invariant?			(✓)		(✓)		
Homogeneous scale invariant?			(✓)				
Positive definite?		✓	×	×	✓		✓
Symmetric?	✓	✓	✓	×	✓	✓	✓
Triangle inequality?		✓	×	×	✓		×
Consistency ($n_i \to \infty$)?	–	–	–		–		–
Consistency ($p \to \infty$)?	–	–	–		–		–
No. fulfilled criteria	9	5	10	6	11	6	8
No. cond. fulfilled criteria	1	1	5	0	2	0	0
No. unfulfilled criteria	3	5	3	8	4	5	6
No. NAs	6	8	1	7	2	10	5

TABLE 11
*Comparison of **different testing approaches** regarding applicability, interpretability, and theoretical properties.* for RI version.*

Method/ Article	Romano (1989)	Burke (2000)	Ping (2000)	Chen and Hanson (2014)	Zhou, Zheng and Zhang (2017)	Pan et al. (2018)	Wan, Liu and Deng (2018)	Kim, Balakrishnan and Wasserman (2020)	Li and Zhang (2020)	Liu et al. (2022)	Paul, De and Ghosh (2022a)
Also target?	×	×	×	×	×	×	×	×	×	×	×
Numeric?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Categorical?	✓	✓	×	×	×	×	×	×	×	×	(✓)
$n_i \neq n_j$?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$p > n_i$?	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
$k > 2$?	✓	×	✓	✓	×	✓	×	×	×	✓	✓
Without training?	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
No assumptions?	✓	✓	×	×	×	✓	×	×	✓	×	✓
No parameters?	×	×	×	×	×	×	✓	✓	✓	×	×
Implemented?						✓					✓
Complexity?				$\mathcal{O}(pN^2)$				$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$		
Interpretable units?	×	×	×	×	×	×	×	×	×	×	×
Lower bound?	0		0	0	0	0		0	0	0	0
Upper bound?											1*
Rotation invariant?								✓			(✓)
Location change invariant?	(✓)										(✓)
Homogeneous scale invariant?											
Positive definite?		×		×	×	✓		✓	✓		
Symmetric?	✓	×	✓	✓	×	✓	✓	✓	✓	✓	✓
Triangle inequality?				×		×			✓		
Consistency ($n_i \to \infty$)?	✓	✓	✓	✓	(✓)	✓	✓	✓	✓	✓	
Consistency ($p \to \infty$)?					(✓)			✓			✓
No. fulfilled criteria	10	7	8	7	5	11	7	11	11	7	11
No. cond. fulfilled criteria	1	0	0	0	2	0	0	0	0	0	3
No. unfulfilled criteria	3	6	5	7	8	5	5	5	4	6	3
No. NAs	7	8	8	7	6	5	9	5	6	8	4

7. Conclusion

In statistics and machine learning, measuring the similarity between two or more datasets has widespread applications. Extremely many approaches for quantifying dataset similarity have been proposed in the literature. We examined more than 100 methods for quantifying the similarity of datasets. The methods were selected from an extensive literature search by using the following criteria:

- The method is applicable for multivariate datasets.
- The method requires no specific parametric or distributional assumptions on the underlying distributions of the datasets (e.g. normal distribution).
- The method does not focus on a particular property of the data (e.g. means), but on the entire dataset or its entire distribution.

We classified the methods into ten classes based on their main ideas, including

1. Comparison of cumulative distribution functions, density functions, or characteristic functions
2. Methods based on multivariate ranks
3. Discrepancy measures for distributions
4. Graph-based methods
5. Methods based on inter-point distances
6. Kernel-based methods
7. Methods based on binary classification
8. Distance and similarity measures for datasets
9. Comparison based on summary statistics
10. Different testing approaches.

We presented an extensive review of these methods. For each method, we introduced the underlying ideas, formal definitions, and important properties. An overview of the methods can be found in Table 1 and a summary of the classes can be found in Section 4.

Moreover, we compared all these methods with respect to 22 criteria that can be divided into the three categories applicability (e.g. is the method applicable to numeric or categorical data), interpretability (e.g. is the statistic bounded), and theoretical properties (e.g. metric properties). The criteria can be used to judge which methods are best suited for quantifying the similarity of given datasets. Overall, we found that graph-based methods had the highest numbers of fulfilled criteria.

To facilitate the choice of an appropriate data similarity measure for a concrete application, we provided detailed comparisons of the methods. Moreover, we designed an online tool (https://shiny.statistik.tu-dortmund.de/data-similarity) that allows for custom filtering of the criteria and sorting of the methods. Therefore, the online tool can provide more specific guidance for the choice of a suitable dataset similarity method for concrete data at hand in addition to the overall comparison presented in this paper. We intend to expand this online tool over time. Suggestions for new methods to be included, as well as additional entries for criteria not yet marked as fulfilled or unfulfilled, are welcome. These can be added as an issue in the GitHub repository (https://github.com/MariekeStolte/ComparisonToolDatasetSimilarity.git).

Note that the comparison so far does not include the performance of the methods, e.g. type I error rates and power for two- and k-sample tests. Therefore, no statements can be made as to whether the methods that perform well in this theoretical comparison also perform well in practice. There are limited simulation results on the performance of the methods available in some of the respective articles.

Moreover, the discussion is restricted to datasets with the same number of variables since the aspect of comparing datasets with different dimensions is very rarely discussed in the literature. Further, in the applications we have in mind the comparison of datasets with different dimensions is also not relevant.

For future research, we plan to incorporate the best-performing methods into

a comparison of parametric and Plasmode simulation studies. Within this comparison, a critical step is to quantify how far assumptions of the simulations deviate from a true data-generating process. It is desirable to quantify this deviation in terms of a dataset similarity or distance rather than in terms of specific parameters that are changing. Moreover, we plan to conduct an empirical comparison of the methods to evaluate how well the methods perform in practice and to provide a fair comparison of the method performance.

Acknowledgments

The authors would like to thank the anonymous referees and the Editor for their constructive comments that improved the quality of this paper.

Funding

This work has been supported (in part) by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project P1) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation – Project Number 427806116).

References

Agarwal, S. M. D., Bhattacharya, B. and Zhang, N. R. (2020). multicross: A graph-based test for comparing multivariate distributions in the multi sample framework. R package version 2.1.0.

Ahmad, I. A. and Cerrito, P. B. (1993). Goodness of fit tests based on the L2-norm of multivariate probability density functions. *Journal of Nonparametric Statistics* **2** 169–181. https://doi.org/10.1080/10485259308832550. MR1256380

Al-Labadi, L., Asl, F. F. and Saberi, Z. (2022). A Bayesian nonparametric multi-sample test in any dimension. *AStA Advances in Statistical Analysis* **106** 217–242. https://doi.org/10.1007/s10182-021-00419-3. MR4426855

Alba, M. V., Barrera, D. and Jiménez, M. D. (2001). A homogeneity test based on empirical characteristic functions. *Computational Statistics* **16** 255–270. https://doi.org/10.1007/s001800100064. MR1857131

Alba-Fernández, V., Ibáñez-Pérez, M. J. and Jiménez-Gamero, M. D. (2004). A bootstrap algorithm for the two-sample problem using trigonometric Hermite spline interpolation. *Communications in Nonlinear Science and Numerical Simulation* **9** 275–286. https://doi.org/10.1016/S1007-5704(03)00117-5. MR2044139

Alba Fernández, V., Jiménez Gamero, M. D. and Muñoz García, J. (2008). A test for the two-sample problem based on empirical characteristic functions. *Computational Statistics & Data Analysis* **52** 3730–3748. https://doi.org/10.1016/j.csda.2007.12.013. MR2427377

Ali, S. M. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)* **28** 131–142. https://doi.org/10.1111/j.2517-6161.1966.tb00626.x. MR0196777

Allmon, A. G., Marron, J. S. and Hudgens, M. G. (2021). diproperm: Conduct direction-projection-permutation tests and display plots. R package version 0.2.0.

Alvarez-Melis, D. and Fusi, N. (2020). Geometric dataset distances via optimal transport. In *Advances in Neural Information Processing Systems* **33** 21428–21439. Curran Associates, Inc.

Anderson, N. H., Hall, P. and Titterington, D. M. (1994). Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis* **50** 41–54. https://doi.org/10.1006/jmva.1994.1033. MR1292607

Arias-Castro, E. and Pelletier, B. (2016). On the consistency of the crossmatch test. *Journal of Statistical Planning and Inference* **171** 184–190. https://doi.org/10.1016/j.jspi.2015.10.003. MR3458077

Aslan, B. and Zech, G. (2005a). New test for the multivariate two-sample problem based on the concept of minimum energy. *Journal of Statistical Computation and Simulation* **75** 109–119. https://doi.org/10.1080/00949650410001661440. MR2117010

Aslan, B. and Zech, G. (2005b). Statistical energy as a tool for binning-free, multivariate goodness-of-fit tests, two-sample comparison and unfolding. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **537** 626–636. https://doi.org/10.1016/j.nima.2004.08.071.

Bahr, R. (1996). Ein neuer Test für das mehrdimensionale Zwei-Stichproben-Problem bei allgemeiner Alternative, PhD thesis, Universität Hannover.

Barakat, A. S., Quade, D. and Salama, I. A. (1996). Multivariate homogeneity testing using an extended concept of nearest neighbors. *Biometrical Journal* **38** 605–612. https://doi.org/10.1002/bimj.4710380509.

Baringhaus, L. and Franz, C. (2004). On a new multivariate two-sample test. *Journal of Multivariate Analysis* **88** 190–206. https://doi.org/10.1016/S0047-259X(03)00079-4. MR2021870

Baringhaus, L. and Franz, C. (2010). Rigid motion invariant two-sample tests. *Statistica Sinica* **20** 1333–1361. MR2777328

Basu, A., Shioya, H. and Park, C. (2011). *Statistical Inference: The Minimum Distance Approach*. CRC Press. MR2830561

Bhattacharya, B. B. (2020). Asymptotic distribution and detection thresholds for two-sample tests based on geometric graphs. *The Annals of Statistics* **48** 2879–2903. https://doi.org/10.1214/19-AOS1913. MR4152627

Biau, G. and Gyorfi, L. (2005). On the asymptotic properties of a nonparametric L_1-test statistic of homogeneity. *IEEE Transactions on Information Theory* **51** 3965–3973. https://doi.org/10.1109/TIT.2005.856979. MR2239012

Bickel, P. J. (1969). A distribution free version of the Smirnov two sample test in the p-variate case. *The Annals of Mathematical Statistics* **40** 1–23. MR0256519

Birnbaum, Z. and Orlicz, W. (1931). Über die Verallgemeinerung des Begriffes der zueinander konjugierten Potenzen. *Studia Mathematica* **3** 1–67.

Biswas, M. and Ghosh, A. K. (2014). A nonparametric two-sample test applicable to high dimensional data. *Journal of Multivariate Analysis* **123** 160–171. https://doi.org/10.1016/j.jmva.2013.09.004. MR3130427

Biswas, M., Mukhopadhyay, M. and Ghosh, A. K. (2014). A distribution-free two-sample run test applicable to high-dimensional data. *Biometrika* **101** 913–926. https://doi.org/10.1093/biomet/asu045. MR3286925

Bińkowski, M., Sutherland, D. J., Arbel, M. and Gretton, A. (2021). Demystifying MMD GANs. arXiv:1801.01401 [cs, stat]. https://doi.org/10.48550/arXiv.1801.01401.

Boeckel, M., Spokoiny, V. and Suvorikova, A. (2018). Multivariate Brenier cumulative distribution functions and their application to non-parametric testing. arXiv:1809.04090 [math, stat]. https://doi.org/10.48550/arXiv.1809.04090.

Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B. and Smola, A. J. (2006). Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics (Oxford, England)* **22** e49–57. https://doi.org/10.1093/bioinformatics/btl242.

Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics* **7** 200–217. https://doi.org/10.1016/0041-5553(67)90040-7. MR0215617

Burbea, J. and Rao, C. (1982). On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory* **28** 489–495. https://doi.org/10.1109/TIT.1982.1056497. MR0672884

Burke, M. D. (2000). Multivariate tests-of-fit and uniform confidence bands using a weighted bootstrap. *Statistics & Probability Letters* **46** 13–20. https://doi.org/10.1016/S0167-7152(99)00082-6. MR1730878

Cai, H., Goggin, B. and Jiang, Q. (2020). Two-sample test based on classification probability. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **13** 5–13. https://doi.org/10.1002/sam.11438. MR4063880

Cai, T., Liu, W. and Xia, Y. (2013). Two-sample covariance matrix testing and support recovery in high-dimensional and sparse settings. *Journal of the American Statistical Association* **108** 265–277. https://doi.org/10.1080/01621459.2012.758041. MR3174618

Calderon Ramirez, S., Oala, L., Torrentes-Barrena, J., Yang, S., Elizondo, D., Moemeni, A., Colreavy-Donnelly, S., Samek, W., Molina-Cabello, M. and Lopez-Rubio, E. (2022). Dataset similarity to assess semi-supervised learning under distribution mismatch between the labelled and unlabelled datasets. *IEEE Transactions on Artificial Intelligence* **4** 282–291. https://doi.org/10.1109/TAI.2022.3168804.

Cao, R. and van Keilegom, I. (2006). Empirical likelihood tests for two-

sample problems via nonparametric density estimation. *Canadian Journal of Statistics* **34** 61–77. https://doi.org/10.1002/cjs.5550340106. MR2267710

Chakraborty, S. and Zhang, X. (2021). A new framework for distance and kernel-based metrics in high dimensions. *Electronic Journal of Statistics* **15** 5455–5522. MR4352549

Chen, H., Chen, X. and Su, Y. (2018). A weighted edge-count two-sample test for multivariate and object data. *Journal of the American Statistical Association* **113** 1146–1155. https://doi.org/10.1080/01621459.2017.1307757. MR3862346

Chen, L., Dou, W. W. and Qiao, Z. (2013). Ensemble subsampling for imbalanced multivariate two-sample tests. *Journal of the American Statistical Association* **108** 1308–1323. https://doi.org/10.1080/01621459.2013.800763. MR3174710

Chen, H. and Friedman, J. H. (2017). A new graph-based two-sample test for multivariate and object data. *Journal of the American Statistical Association* **112** 397–409. https://doi.org/10.1080/01621459.2016.1147356. MR3646580

Chen, Y. and Hanson, T. E. (2014). Bayesian nonparametric k-sample tests for censored and uncensored data. *Computational Statistics & Data Analysis* **71** 335–346. https://doi.org/10.1016/j.csda.2012.11.003. MR3131974

Chen, Y. and Markatou, M. (2020). Kernel Tests for One, Two, and K-Sample Goodness-of-Fit: State of the Art and Implementation Considerations. In *Statistical Modeling in Biomedical Research: Contemporary Topics and Voices in the Field* (Y. Zhao and D.-G. D. Chen, eds.). *Emerging Topics in Statistics and Biostatistics* 309–337. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-33416-1_14.

Chen, H. and Zhang, N. R. (2013). Graph-based tests for two-sample comparisons of categorical data. *Statistica Sinica* **23** 1479–1503. MR3222245

Chen, H. and Zhang, J. (2017). gTests: Graph-based two-sample tests. R package version 0.2.

Cheng, X., Cloninger, A. and Coifman, R. R. (2020). Two-sample statistics based on anisotropic kernels. *Information and Inference: A Journal of the IMA* **9** 677–719. https://doi.org/10.1093/imaiai/iaz018. MR4146351

Cheng, X. and Cloninger, A. (2022). Classification logit two-sample testing by neural networks. *IEEE Transactions on Information Theory* **68** 6631–6662. https://doi.org/10.1109/TIT.2022.3175691.

Chernozhukov, V., Galichon, A., Hallin, M. and Henry, M. (2017). Monge–Kantorovich depth, quantiles, ranks and signs. *The Annals of Statistics* **45** 223–256. https://doi.org/10.1214/16-AOS1450. MR3611491

Choi, K. and Marden, J. (1997). An approach to multivariate rank tests in multivariate analysis of variance. *Journal of the American Statistical Association* **92** 1581–1590. https://doi.org/10.1080/01621459.1997.10473680. MR1615267

Chwialkowski, K. P., Ramdas, A., Sejdinovic, D. and Gretton, A. (2015). Fast two-sample testing with analytic representations of probability measures. In *Advances in Neural Information Processing Systems* **28**. Curran Associates, Inc.

Csiszár, I. (1963). Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizität von Markoffschen Ketten. *A Magyar Tudományos Akadémia. Matematikai Kutató Intézetének Közleményei* **8** 85–108. MR164374.

Danafar, S., Rancoita, P. M. V., Glasmachers, T., Whittinstal, K. and Schmidhuber, J. (2014). Testing hypotheses by regularized maximum mean discrepancy. *International Journal of Computer and Information Technology* **02** 223–232.

Roux de Bezieux, H. (2021). Ecume: Equality of 2 (or k) continuous univariate and multivariate distributions. R package version 0.9.1.

Deb, N., Bhattacharya, B. B. and Sen, B. (2021). Efficiency lower bounds for distribution-free hotelling-type two-sample tests based on optimal transport. arXiv:2104.01986 [math, stat]. https://doi.org/10.48550/arXiv.2104.01986.

Deb, N. and Sen, B. (2021). Multivariate rank-based distribution-free nonparametric testing using measure transportation. *Journal of the American Statistical Association* **118** 1–16. https://doi.org/10.1080/01621459.2021.1923508. MR4571116

Dudley, R. M. (1989). *Real Analysis and Probability*. Wadsworth and Brooks, New York. https://doi.org/10.1201/9781351076197. MR0982264

Fan, K. (1943). Entfernung zweier zufälligen Größen und die Konvergenz nach Wahrscheinlichkeit. *Mathematische Zeitschrift* **49** 681–683. https://doi.org/10.1007/BF01174225. MR0011903

Feurer, M., Springenberg, J. and Hutter, F. (2015). Initializing Bayesian hyperparameter optimization via meta-learning. *Proceedings of the AAAI Conference on Artificial Intelligence* **29**. https://doi.org/10.1609/aaai.v29i1.9354.

Flaxman, S., Sejdinovic, D., Cunningham, J. P. and Filippi, S. (2016). Bayesian learning of kernel embeddings. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence. UAI'16* 182–191. AUAI Press, Arlington, Virginia, USA.

Fokianos, K., Qin, J., Kedem, B. and Short, D. A. (2001). A semiparametric approach to the one-way layout. *Technometrics* **43** 56–65. MR1819908

Franz, C. (2019). cramer: Multivariate nonparametric Cramer-test for the two-sample-problem. R package version 0.9-3.

Friedman, J. (2004). On Multivariate Goodness-of-Fit and Two-Sample Testing Technical Report, SLAC National Accelerator Lab., Menlo Park, CA (United States).

Friedman, J. H. and Rafsky, L. C. (1979). Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *The Annals of Statistics* **7** 697–717. MR0532236

Friedman, J. H. and Steppel, S. (1973). A nonparametric procedure for

comparing multivariate point sets. *Stanford Linear Accelerator Center Computation Research Group Technical Memo* **153**.

Fromont, M., Laurent, B., Lerasle, M. and Reynaud-Bouret, P. (2012). Kernels based tests with non-asymptotic bootstrap approaches for two-sample problems. In *Proceedings of the 25th Annual Conference on Learning Theory* 23.1–23.23. JMLR Workshop and Conference Proceedings.

Fukumizu, K., Bach, F. R. and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research* **5** 73–99. MR2247974

Ganti, V., Gehrke, J., Ramakrishnan, R. and Loh, W.-Y. (1999). A framework for measuring changes in data characteristics. In *Proceedings of the 18th Symposium on Principles of Database Systems* 126–137.

García-García, D. and Williamson, R. C. (2012). Divergences and risks for multiclass experiments. In *Proceedings of the 25th Annual Conference on Learning Theory* 28.1–28.20. JMLR Workshop and Conference Proceedings ISSN: 1938-7228.

Genevay, A., Peyre, G. and Cuturi, M. (2018). Learning generative models with Sinkhorn divergences. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics* 1608–1617. PMLR ISSN: 2640-3498.

Gerber, H. U. (1979). *An Introduction to Mathematical Risk Theory*. Huebner Foundation Monograph. MR0579350

Ghosal, P. and Sen, B. (2021). Multivariate ranks and quantiles using optimal transport: Consistency, rates, and nonparametric testing. arXiv:1905.05340 [math, stat]. https://doi.org/10.48550/arXiv.1905.05340. MR4404927

Ghosh, A. K. and Biswas, M. (2016). Distribution-free high-dimensional two-sample tests based on discriminating hyperplanes. *TEST* **25** 525–547. https://doi.org/10.1007/s11749-015-0467-x. MR3531841

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. and Smola, A. (2006). A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems* **19**. MIT Press.

Gretton, A., Fukumizu, K., Harchaoui, Z. and Sriperumbudur, B. K. (2009). A fast, consistent kernel two-sample test. In *Advances in Neural Information Processing Systems* **22**. Curran Associates, Inc.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. and Smola, A. (2012a). A kernel two-sample test. *Journal of Machine Learning Research* **13** 723–773. MR2913716

Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K. and Sriperumbudur, B. K. (2012b). Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems* **25**. Curran Associates, Inc.

Györfi, L. and Nemetz, T. (1975). f-dissimilarity: A general class of separation measures of several probability measures. *Topics in Information Theory. Colloq. Math. Soc. János Bolyai* **16** 309–321. MR0459923

Hall, P., Marron, J. S. and Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society:*

Series B (Statistical Methodology) **67** 427–444. https://doi.org/10.1111/j.1467-9868.2005.00510.x. MR2155347

Hall, P. and Tajvidi, N. (2002). Permutation tests for equality of distributions in high-dimensional settings. *Biometrika* **89** 359–374. MR1913964

Hallin, M., Hlubinka, D. and Hudecová, v. (2022). Efficient fully distribution-free center-outward rank tests for multiple-output regression and MANOVA. *Journal of the American Statistical Association* **118** 1–17. https://doi.org/10.1080/01621459.2021.2021921. MR4646617

Harchaoui, Z., Bach, F. and Moulines, E. (2008). Testing for homogeneity with kernel Fisher discriminant analysis. In *Advances in Neural Information Processing Systems* **20**. Curran Associates, Inc.

Hediger, S., Michel, L. and Näf, J. (2021). hypoRF: Random forest two-sample tests. R package version 1.0.0.

Hediger, S., Michel, L. and Näf, J. (2022). On the use of random forest for two-sample testing. *Computational Statistics & Data Analysis* **170** 107435. https://doi.org/10.1016/j.csda.2022.107435. MR4374771

Heller, R., Small, D. and Rosenbaum, P. (2012). crossmatch: The cross-match test. R package version 1.3-1.

Heller, R., Jensen, S. T., Rosenbaum, P. R. and Small, D. S. (2010). Sensitivity analysis for the cross-match test, with applications in genomics. *Journal of the American Statistical Association* **105** 1005–1013. https://doi.org/10.1198/jasa.2010.ap09260. MR2752596

Henze, N. (1988). A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics* **16** 772–783. MR0947577

Henze, N. and Penrose, M. D. (1999). On the multivariate runs test. *The Annals of Statistics* **27** 290–298. MR1701112

Henze, N. and Voigt, B. (1992). Almost sure convergence of certain slowly changing symmetric one- and multi-sample statistics. *The Annals of Probability* **20** 1086–1098. https://doi.org/10.1214/aop/1176989819. MR1159587

Hettmansperger, T. P., Möttönen, J. and Oja, H. (1998). Affine invariant multivariate rank tests for several samples. *Statistica Sinica* **8** 785–800. MR1651508

Hettmansperger, T. P. and Oja, H. (1994). Affine invariant multivariate multisample sign tests. *Journal of the Royal Statistical Society: Series B (Methodological)* **56** 235–249. MR1257810

Holmes, C. C., Caron, F., Griffin, J. E. and Stephens, D. A. (2015). Two-sample Bayesian nonparametric hypothesis testing. *Bayesian Analysis* **10** 297–320. https://doi.org/10.1214/14-BA914. MR3420884

Huang, Z. (2022). KMD: Kernel measure of multi-sample dissimilarity. R package version 0.1.0.

Huang, C. and Huo, X. (2017). An efficient and distribution-free two-sample test based on energy statistics and random projections. arXiv:1707.04602 [stat]. https://doi.org/10.48550/arXiv.1707.04602.

Huang, Z. and Sen, B. (2023). A kernel measure of dissimilarity between M

distributions. *Journal of the American Statistical Association* 1–27. https://doi.org/10.1080/01621459.2023.2298036.

Hušková, M. and Meintanis, S. G. (2008). Tests for the multivariate k-sample problem based on the empirical characteristic function. *Journal of Nonparametric Statistics* **20** 263–277. https://doi.org/10.1080/10485250801948294. MR2421770

Jitkrittum, W., Szabó, Z., Chwialkowski, K. P. and Gretton, A. (2016). Interpretable distribution features with maximum testing power. In *Advances in Neural Information Processing Systems* **29**. Curran Associates, Inc.

Jitkrittum, W., Kanagawa, H., Sangkloy, P., Hays, J., Schölkopf, B. and Gretton, A. (2018). Informative features for model comparison. In *Advances in Neural Information Processing Systems* **31**. Curran Associates, Inc.

Johnson, T. and Dasu, T. (1998). Comparing massive high-dimensional data sets. In *KDD* 229–233.

Kanamori, T., Suzuki, T. and Sugiyama, M. (2012). f-Divergence estimation and two-sample homogeneity test under semiparametric density-ratio models. *IEEE Transactions on Information Theory* **58** 708–720. https://doi.org/10.1109/TIT.2011.2163380. MR2917977

Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science* **6** 366–422. https://doi.org/10.1287/mnsc.6.4.366. MR0129016

Kantorovitch, L. (1958). On the translocation of masses. *Management Science* **5** 1–4. https://doi.org/10.1287/mnsc.5.1.1. MR0096552

Karatzoglou, A., Smola, A. and Hornik, K. (2022). kernlab: Kernel-based machine learning lab. R package version 0.9-31.

Karatzoglou, A., Smola, A., Hornik, K. and Zeileis, A. (2004). kernlab – An S4 package for kernel methods in R. *Journal of Statistical Software* **11** 1–20. https://doi.org/10.18637/jss.v011.i09.

Kim, I., Balakrishnan, S. and Wasserman, L. (2020). Robust multivariate nonparametric tests via projection averaging. *The Annals of Statistics* **48** 3417–3441. https://doi.org/10.1214/19-AOS1936. MR4185814

Kim, I., Lee, A. B. and Lei, J. (2019). Global and local two-sample tests via regression. *Electronic Journal of Statistics* **13** 5253–5305. https://doi.org/10.1214/19-EJS1648.

Kim, I., Ramdas, A., Singh, A. and Wasserman, L. (2021). Classification accuracy as a proxy for two-sample testing. *The Annals of Statistics* **49** 411–434. https://doi.org/10.1214/20-AOS1962.

Kirchler, M., Khorasani, S., Kloft, M. and Lippert, C. (2020). Two-sample testing using deep learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* 1387–1398. PMLR.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics* **22** 79–86. https://doi.org/10.1214/aoms/1177729694.

Le, Q., Sarlos, T. and Smola, A. (2013). Fastfood – computing Hilbert space expansions in loglinear time. In *Proceedings of the 30th International Conference on Machine Learning* 244–252. PMLR.

Le Cam, L. (1986). *Asymptotic Methods in Statistical Decision Theory. Springer Series in Statistics*. Springer, New York, NY.

Leite, R., Brazdil, P. and Vanschoren, J. (2012). Selecting classification algorithms with active testing. In *Machine Learning and Data Mining in Pattern Recognition* (P. Perner, ed.). *Lecture Notes in Computer Science* 117–131. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31537-4_10.

Leite, R. and Brazdil, P. (2021). Exploiting performance-based similarity between datasets in metalearning. In *AAAI Workshop on Meta-Learning and MetaDL Challenge* 90–99. PMLR.

Li, J. (2018). Asymptotic normality of interpoint distances for high-dimensional data with applications to the two-sample problem. *Biometrika* **105** 529–546. https://doi.org/10.1093/biomet/asy020.

Li, X., Hu, W. and Zhang, B. (2022). Measuring and testing homogeneity of distributions by characteristic distance. *Statistical Papers* **64** 529–556. https://doi.org/10.1007/s00362-022-01327-7.

Li, Z. and Zhang, Y. (2020). On a projective ensemble approach to two sample test for equality of distributions. In *Proceedings of the 37th International Conference on Machine Learning* 6020–6027. PMLR.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y. and Poczos, B. (2017). MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems* **30**. Curran Associates, Inc.

Liese, F. and Vajda, I. (1987). *Convex Statistical Distances. Teubner-Texte zur Mathematik* **95**. Teubner.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* **37** 145–151. https://doi.org/10.1109/18.61115

Lindsay, B. G., Markatou, M. and Ray, S. (2014). Kernels, degrees of freedom, and power properties of quadratic distance goodness-of-fit tests. *Journal of the American Statistical Association* **109** 395–410. https://doi.org/10.1080/01621459.2013.836972.

Lindsay, B. G., Markatou, M., Ray, S., Yang, K. and Chen, S.-C. (2008). Quadratic distances on probabilities: A unified foundation. *The Annals of Statistics* **36** 983–1006. https://doi.org/10.1214/009053607000000956.

Liu, Y., Li, C.-L. and Póczos, B. (2018). Classifier two sample test for video anomaly detections. In *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018* 71. BMVA Press.

Liu, Y., Liu, Z. and Zhou, W. (2019). A test for equality of two distributions via integrating characteristic functions. *Statistica Sinica* **29** 1779–1801.

Liu, Z. and Modarres, R. (2011). A triangle test for equality of distribution functions in high dimensions. *Journal of Nonparametric Statistics* **23** 605–615.

https://doi.org/10.1080/10485252.2010.485644.

Liu, R. Y. and Singh, K. (1993). A quality index based on data depth and multivariate rank tests. *Journal of the American Statistical Association* **88** 252–260. https://doi.org/10.1080/01621459.1993.10594317.

Liu, Z., Xia, X. and Zhou, W. (2015). A test for equality of two distributions via jackknife empirical likelihood and characteristic functions. *Computational Statistics & Data Analysis* **92** 97–114. https://doi.org/10.1016/j.csda.2015.06.004.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A. and Sutherland, D. J. (2020). Learning deep kernels for non-parametric two-sample tests. In *Proceedings of the 37th International Conference on Machine Learning* 6316–6326. PMLR.

Liu, L., Meng, Y., Wu, X., Ying, Z. and Zheng, T. (2022). Log-rank-type tests for equality of distributions in high-dimensional spaces. *Journal of Computational and Graphical Statistics* 1–13. https://doi.org/10.1080/10618600.2022.2051530.

Lopez-Paz, D. and Oquab, M. (2017). Revisiting classifier two-sample tests. In *International Conference on Learning Representations*.

Ma, L. and Wong, W. H. (2011). Coupling optional Pólya trees and the two sample problem. *Journal of the American Statistical Association* **106** 1553–1565. https://doi.org/10.1198/jasa.2011.tm10003

Maa, J.-F., Pearl, D. K. and Bartoszyński, R. (1996). Reducing multidimensional two-sample data to one-dimensional interpoint comparisons. *The Annals of Statistics* **24** 1069–1074. https://doi.org/10.1214/aos/1032526956.

Marron, J. S., Todd, M. J. and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association* **102** 1267–1271.

Meintanis, S. G. (2016). A review of testing procedures based on the empirical characteristic function. *South African Statistical Journal* **50** 1–14.

Mondal, P. K., Biswas, M. and Ghosh, A. K. (2015). On high dimensional two-sample tests based on nearest neighbors. *Journal of Multivariate Analysis* **141** 168–178. https://doi.org/10.1016/j.jmva.2015.07.002.

Montero-Manso, P. and Vilar, J. A. (2019). Two-sample homogeneity testing: A procedure based on comparing distributions of interpoint distances. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **12** 234–252. https://doi.org/10.1002/sam.11417.

Moulines, E., Bach, F. and Harchaoui, Z. (2007). Testing for homogeneity with kernel Fisher discriminant analysis. In *Advances in Neural Information Processing Systems* **20**. Curran Associates, Inc.

Muandet, K., Fukumizu, K., Sriperumbudur, B. and Schölkopf, B. (2017). Kernel Mean Embedding of Distributions: A Review and Beyond. *Foundations and Trends® in Machine Learning* **10** 1–141. https://doi.org/10.1561/2200000060.

Mukherjee, S., Agarwal, D., Zhang, N. R. and Bhattacharya, B. B. (2022). Distribution-free multisample tests based on optimal matchings with applications to single cell genomics. *Journal of the American Statisti-*

cal Association **117** 627–638. https://doi.org/10.1080/01621459.2020.1791131.

Mukhopadhyay, S. and Wang, K. (2020a). A nonparametric approach to high-dimensional k-sample comparison problems. *Biometrika* **107** 555–572. https://doi.org/10.1093/biomet/asaa015.

Mukhopadhyay, S. and Wang, K. (2020b). LPKsample: LP nonparametric high dimensional K-sample comparison. R package version 2.1.

Munoz, A. and Moguerza, J. M. (2006). Estimation of high-density regions using one-class neighbor machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28** 476–480. https://doi.org/10.1109/TPAMI.2006.52.

Muñoz, A., Martos, G. and González, J. (2013). A New distance for data sets in a reproducing kernel Hilbert space context. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (J. Ruiz-Shulcloper and G. Sanniti di Baja, eds.). *Lecture Notes in Computer Science* 222–229. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41822-8_28.

Muñoz, A., Martos, G., Arriero, J. and Gonzalez, J. (2012). A new distance for probability measures based on the estimation of level sets. In *Artificial Neural Networks and Machine Learning – ICANN 2012* (A. E. P. Villa, W. Duch, P. Érdi, F. Masulli and G. Palm, eds.). *Lecture Notes in Computer Science* 271–278. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33266-1_34.

Mémoli, F. (2011). Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics* **11** 417–487. https://doi.org/10.1007/s10208-011-9093-5.

Mémoli, F. (2017). Distances Between Datasets. In *Modern Approaches to Discrete Curvature* (L. Najman and P. Romon, eds.). *Lecture Notes in Mathematics* 115–132. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-58002-9_3.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability* **29** 429–443. https://doi.org/10.2307/1428011.

Nettleton, D. and Banerjee, T. (2001). Testing the equality of distributions of random vectors with categorical components. *Computational Statistics & Data Analysis* **37** 195–208. https://doi.org/10.1016/S0167-9473(01)00015-9.

Nguyen, X., Wainwright, M. J. and Jordan, M. I. (2010). Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory* **56** 5847–5861. https://doi.org/10.1109/TIT.2010.2068870.

Ntoutsi, I., Kalousis, A. and Theodoridis, Y. (2008). A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM). Proceedings* 810–821. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.

9781611972788.73.

Oja, H. and Randles, R. H. (2004). Multivariate nonparametric tests. *Statistical Science* **19** 598–605. https://doi.org/10.1214/088342304000000558.

Pan, W., Tian, Y., Wang, X. and Zhang, H. (2018). Ball divergence: Nonparametric two sample test. *Annals of Statistics* **46** 1109–1137. https://doi.org/10.1214/17-AOS1579.

Paul, B., De, S. K. and Ghosh, A. K. (2022a). Some clustering-based exact distribution-free k-sample tests applicable to high dimension, low sample size data. *Journal of Multivariate Analysis* **190** 104897. https://doi.org/10.1016/j.jmva.2021.104897.

Paul, B., De, S. K. and Ghosh, A. K. (2022b). HDLSSkST: Distribution-free exact high dimensional low sample size k-sample tests. R package version 2.1.0.

Pearson, K. (1900). On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **50** 157–175. https://doi.org/10.1080/14786440009463897.

Petrie, A. (2016). Graph-theoretic multisample tests of equality in distribution for high dimensional data. *Computational Statistics & Data Analysis* **96** 145–158. https://doi.org/10.1016/j.csda.2015.11.003.

Ping, J. (2000). Bootstrap tests for the equality of distributions. *Korean Journal of Computational & Applied Mathematics* **7** 347–362. https://doi.org/10.1007/BF03012197.

Preiss, D. and Tišer, J. (1991). Measures in Banach spaces are determined by their values on balls. *Mathematika. A Journal of Pure and Applied Mathematics* **38** 391–397 (1992). https://doi.org/10.1112/S0025579300006744 MR1147839.

Prokhorov, Y. V. (1956). Convergence of Random processes and limit theorems in probability theory. *Theory of Probability & Its Applications* **1** 157–214. https://doi.org/10.1137/1101016.

Puri, M. L., Sen, P. K. et al. (1971). Nonparametric methods in multivariate analysis.

R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

Rachev, S. T. (1991). *Probability Metrics and the Stability of Stochastic Models*. John Wiley & Sons, Chichester.

Rachev, S. T. and Rüschendorf, L. (1990). Approximation of sums by compound Poisson distributions with respect to stop-loss distances. *Advances in Applied Probability* **22** 350–374. https://doi.org/10.2307/1427540.

Rachev, S. T. and Rüschendorf, L. (1998). *Mass Transportation Problems Volume 1: Theory. Probability and its Applications*. Springer, New York. https://doi.org/10.1007/b98893.

Rachev, S. T., Stoyanov, S. and Fabozzi, F. J. (2008). *Advanced Stochastic Models, Risk Assessment, and Portfolio Optimization: The Ideal Risk, Uncer-*

tainty, and Performance Measures. The Frank J. Fabozzi series. John Wiley & Sons.

Rachev, S. T., Stoyanov, S. V. and Fabozzi, F. J. (2011). *A Probability Metrics Approach to Financial Risk Measures*. John Wiley & Sons, Ltd, New York.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems* **20**. Curran Associates, Inc.

Rahmatallah, Y., Zybailov, B., Emmert-Streib, F. and Glazko, G. (2017). GSAR: Bioconductor package for gene set analysis in R. *BMC Bioinformatics* **18** 61.

Ramdas, A., Trillos, N. G. and Cuturi, M. (2017). On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy* **19** 47. https://doi.org/10.3390/e19020047.

Ramdas, A., Reddi, S. J., Poczos, B., Singh, A. and Wasserman, L. (2015). On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. *Proceedings of the AAAI Conference on Artificial Intelligence* **29**. https://doi.org/10.1609/aaai.v29i1.9692.

Randles, R. H. and Peters, D. (1990). Multivariate rank tests for the two-sample location problem. *Communications in Statistics – Theory and Methods* **19** 4225–4238. https://doi.org/10.1080/03610929008830439.

Rao, C. R. (1952). *Advanced Statistical Methods in Biometric Research*. John Wiley & Sons.

Rao, C. R. (1973). *Linear Statistical Inference and is Applications*, 2 ed. John Wiley & Sons, Incorporated.

Rizzo, M. and Szekely, G. (2022). energy: E-Statistics: Multivariate inference via the energy of data. R package version 1.7-10.

Rizzo, M. L. and Székely, G. J. (2010). DISCO analysis: A nonparametric extension of analysis of variance. *The Annals of Applied Statistics* **4** 1034–1055. https://doi.org/10.1214/09-AOAS245.

Roederer, M., Moore, W., Treister, A., Hardy, R. R. and Herzenberg, L. A. (2001). Probability binning comparison: A metric for quantitating multivariate distribution differences. *Cytometry* **45** 47–55.

Rogers, W. H. (1978). *Some Convergence Properties of K-Nearest Neighbor Estimates*. Stanford University.

Romano, J. P. (1989). Bootstrap and randomization tests of some nonparametric hypotheses. *The Annals of Statistics* **17** 141–159.

Rosenbaum, P. R. (2005). An exact distribution-free test comparing two multivariate distributions based on adjacency. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **67** 515–530.

Rosenblatt, J. D., Benjamini, Y., Gilron, R., Mukamel, R. and Goeman, J. J. (2021). Better-than-chance classification for signal detection. *Biostatistics* **22** 365–380. https://doi.org/10.1093/biostatistics/kxz035.

Rényi, A. (1961). On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics* **4.1** 547–562.

Sarkar, S., Biswas, R. and Ghosh, A. K. (2020). On some graph-based two-sample tests for high dimension, low sample size data. *Machine Learning* **109** 279–306. https://doi.org/10.1007/s10994-019-05857-4.

Sarkar, S. and Ghosh, A. K. (2018). On some high-dimensional two-sample tests based on averages of inter-point distances. *Stat* **7** e187. https://doi.org/10.1002/sta4.187.

Sarkar, S. and Ghosh, A. K. (2020). On perfect clustering of high dimension, low sample size data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42** 2257–2272. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.2019.2912599.

Sason, I. and Verdú, S. (2016). f-Divergence inequalities. *IEEE Transactions on Information Theory* **62** 5973–6006. https://doi.org/10.1109/TIT.2016.2603151.

Scetbon, M. and Varoquaux, G. (2019). Comparing distributions: ℓ_1 geometry improves kernel two-sample testing. arXiv:1909.09264 [cs, stat]. https://doi.org/10.48550/arXiv.1909.09264.

Schilling, M. F. (1986). Multivariate two-sample tests based on nearest neighbors. *Journal of the American Statistical Association* **81** 799–806. https://doi.org/10.2307/2289012.

Sejdinovic, D., Sriperumbudur, B., Gretton, A. and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-BASED statistics in hypothesis testing. *The Annals of Statistics* **41** 2263–2291.

Simon-Gabriel, C.-J. and Schölkopf, B. (2018). Kernel distribution embeddings: Universal kernels, characteristic kernels and kernel metrics on distributions. *Journal of Machine Learning Research* **19** 1–29.

Smola, A., Gretton, A., Song, L. and Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *Algorithmic Learning Theory* (M. Hutter, R. A. Servedio and E. Takimoto, eds.). *Lecture Notes in Computer Science* 13–31. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-75225-7_5.

Song, H. and Chen, H. (2021). kerTests: Generalized kernel two-sample tests. R package version 0.1.3.

Song, H. and Chen, H. (2022a). New graph-based multi-sample tests for high-dimensional and non-Euclidean data. arXiv:2205.13787 [stat]. https://doi.org/10.48550/arXiv.2205.13787.

Song, H. and Chen, H. (2022b). gTestsMulti: New graph-based multi-sample tests.

Song, H. and Chen, H. (2023). Generalized kernel two-sample tests. *Biometrika* **111** 755–770. https://doi.org/10.1093/biomet/asad068.

Sriperumbudur, B. K., Fukumizu, K. and Lanckriet, G. R. G. (2011). Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research* **12** 2389–2410.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Lanckriet, G. and Schölkopf, B. (2008). Injective Hilbert space embeddings of probability measures. In *21st Annual Conference on Learning Theory (COLT 2008)*

111–122. Omnipress.
Sriperumbudur, B., Fukumizu, K., Gretton, A., Lanckriet, G. and Schölkopf, B. (2009). Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Advances in Neural Information Processing Systems 22* 1750-1758. Max-Planck-Gesellschaft. Curran, Red Hook, NY, USA.
Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B. and Lanckriet, G. R. G. (2010). Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research* **11** 1517–1561.
Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B. and Lanckriet, G. R. G. (2012). On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics* **6** 1550–1599. https://doi.org/10.1214/12-EJS722.
Sugiyama, M., Liu, S., du Plessis, M. C., Yamanaka, M., Yamada, M., Suzuki, T. and Kanamori, T. (2013a). Direct divergence approximation between probability distributions and its applications in machine learning. *Journal of Computing Science and Engineering* **7** 99–111. https://doi.org/10.5626/JCSE.2013.7.2.99.
Sugiyama, M., Kanamori, T., Suzuki, T., Plessis, M. C. d., Liu, S. and Takeuchi, I. (2013b). Density-difference estimation. *Neural Computation* **25** 2734–2775. https://doi.org/10.1162/NECO_a_00492.
Sutherland, D. J. (2019). Unbiased estimators for the variance of MMD estimators. https://doi.org/10.48550/ARXIV.1906.02104.
Sutherland, D. J., Tung, H.-Y., Strathmann, H., De, S., Ramdas, A., Smola, A. and Gretton, A. (2017). Generative models and model criticism via optimized maximum mean discrepancy. In *International Conference on Learning Representations*.
Szabo, A., Boucher, K., Carroll, W. L., Klebanov, L. B., Tsodikov, A. D. and Yakovlev, A. Y. (2002). Variable selection and pattern recognition with gene expression data generated by the microarray technology. *Mathematical Biosciences* **176** 71–98. https://doi.org/10.1016/s0025-5564(01)00103-1.
Szabo, A., Boucher, K., Jones, D., Tsodikov, A. D., Klebanov, L. B. and Yakovlev, A. Y. (2003). Multivariate exploratory tools for microarray data analysis. *Biostatistics* **4** 555–567. https://doi.org/10.1093/biostatistics/4.4.555.
Székely, G. J. and Rizzo, M. L. (2004). Testing for equal distributions in high dimension. *InterStat* **5** 1249–1272.
Székely, G. J. and Rizzo, M. L. (2013). Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference* **143** 1249–1272. https://doi.org/10.1016/j.jspi.2013.03.018.
Székely, G. J. and Rizzo, M. L. (2017). The energy of data. *Annual Review of Statistics and Its Application* **4** 447–479. https://doi.org/10.1146/annurev-statistics-060116-054026.
Taneja, I. J. and Kumar, P. (2004). Relative information of type s, Csiszár's f-divergence, and information inequalities. *Information Sciences* **166**

105–125. https://doi.org/10.1016/j.ins.2003.11.002. MR2104249

Tatti, N. (2007). Distances between data sets based on summary statistics. *Journal of Machine Learning Research* **8** 131–154. MR2280217

Thas, O. (2010). *Comparing Distributions*. Springer, New York. MR2547894

Tsukada, S.-i. (2019). High dimensional two-sample test based on the inter-point distance. *Computational Statistics* **34** 599–615. MR3953674

Vajda, I. (2009). On metric divergences of probability measures. *Kybernetika* **45** 885–900. MR2650071

van Erven, T. and Harremoës, P. (2014). Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* **60** 3797–3820. https://doi.org/10.1109/TIT.2014.2320500. MR3225930

Vincze, I. (1981). On the concept and measure of information contained in an observation. In *Contributions to Probability* 207–214. Elsevier. MR0618690

Wan, Y., Liu, Z. and Deng, M. (2018). Empirical likelihood test for equality of two distributions using distance of characteristic functions. *Statistics* **52** 1379–1394. https://doi.org/10.1080/02331888.2018.1520855. MR3868887

Wang, J., Gao, R. and Xie, Y. (2021). Two-sample test using projected Wasserstein distance. In *2021 IEEE International Symposium on Information Theory (ISIT)* 3320–3325. https://doi.org/10.1109/ISIT45174.2021.9518186.

Wang, J., Gao, R. and Xie, Y. (2022). Two-sample test with kernel projected Wasserstein distance. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics* 8022–8055. PMLR ISSN: 2640-3498.

Wang, Q., Kulkarni, S. R. and Verdu, S. (2005). Divergence estimation of continuous distributions based on data-dependent partitions. *IEEE Transactions on Information Theory* **51** 3064–3074. https://doi.org/10.1109/TIT.2005.853314. MR2239136

Wang, Q., Kulkarni, S. R. and Verdu, S. (2006). A nearest-neighbor approach to estimating divergence between continuous random vectors. In *2006 IEEE International Symposium on Information Theory* 242–246. https://doi.org/10.1109/ISIT.2006.261842.

Wang, H. and Pei, J. (2005). A random method for quantifying changing distributions in data streams. In *European Conference on Principles of Data Mining and Knowledge Discovery* 684–691. Springer.

Wei, S., Lee, C., Wichers, L. and Marron, J. S. (2016). Direction-projection-permutation for high-dimensional hypothesis tests. *Journal of Computational and Graphical Statistics* **25** 549–569. https://doi.org/10.1080/10618600.2015.1027773. MR3499694

Weiss, L. (1960). Two-sample tests for multivariate distributions. *The Annals of Mathematical Statistics* **31** 159–164. MR0119305

Xu, P. (2019). testOTM: Multivariate ranks and quantiles by optimal transportation. R package version 0.11.2.

Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H. and Sugiyama, M. (2013). Relative density-ratio estimation for robust distribution comparison. *Neural Computation* **25** 1324–1370. https://doi.org/10.1162/NECO_a_

00442. MR3075782

Yu, K., Martin, R., Rothman, N., Zheng, T. and Lan, Q. (2007). Two-sample comparison based on prediction error, with applications to candidate gene association studies. *Annals of Human Genetics* **71** 107–118. https://doi.org/10.1111/j.1469-1809.2006.00306.x.

Zaremba, W., Gretton, A. and Blaschko, M. (2013). B-test: A non-parametric, low variance kernel two-sample test. In *Advances in Neural Information Processing Systems* **26**. Curran Associates, Inc.

Zech, G. and Aslan, B. (2003). A new test for the multivariate two-sample problem based on the concept of minimum energy. arXiv:math/0309164 version: 1. https://doi.org/10.48550/arXiv.math/0309164.

Zhang, J. and Chen, H. (2019). Graph-based two-sample tests for data with repeated observations. arXiv:1711.04349 [stat]. https://doi.org/10.48550/arXiv.1711.04349. MR4359638

Zhang, Q., Wild, V., Filippi, S., Flaxman, S. and Sejdinovic, D. (2022). Bayesian kernel two-sample testing. *Journal of Computational and Graphical Statistics* **31** 1164–1176. https://doi.org/10.1080/10618600.2022.2067547. MR4513378

Zhao, J. and Meng, D. (2015). FastMMD: Ensemble of circular discrepancy for efficient two-sample test. *Neural Computation* **27** 1345–1372. https://doi.org/10.1162/NECO_a_00732. MR3865034

Zhao, S., Sinha, A., He, Y., Perreault, A., Song, J. and Ermon, S. (2021). Comparing distributions by measuring differences that affect decision making. In *International Conference on Learning Representations*.

Zhou, D. and Chen, H. (2023). A new ranking scheme for modern data and its application to two-sample hypothesis testing. In *Proceedings of Thirty Sixth Conference on Learning Theory* 3615–3668. PMLR ISSN: 2640-3498.

Zhou, W.-X., Zheng, C. and Zhang, Z. (2017). Two-sample smooth tests for the equality of distributions. *Bernoulli* **23** 951–989. https://doi.org/10.3150/15-BEJ766. MR3606756

Zhu, Y. and Chen, H. (2024). Limiting distributions of graph-based test statistics on sparse and dense graphs. *Bernoulli* **30** 770–796. https://doi.org/10.3150/23-BEJ1616. MR4665597

Zhu, C. and Shao, X. (2021). Interpoint distance based two sample tests in high dimension. *Bernoulli* **27** 1189–1211. https://doi.org/10.3150/20-BEJ1270. MR4255231

Zhu, J., Pan, W., Zheng, W. and Wang, X. (2021). Ball: An R package for detecting distribution difference and association in metric spaces. *Journal of Statistical Software* **97** 1–31. https://doi.org/10.18637/jss.v097.i06.

Zolotarev, V. M. (1976). Metric distances in spaces of random variables and their distributions. *Mathematics of the USSR-Sbornik* **30** 373. https://doi.org/10.1070/SM1976v030n03ABEH002280. MR0467869

Zolotarev, V. M. (1984). Probability metrics. *Theory of Probability & Its Applications* **28** 278–302. https://doi.org/10.1137/1128025. MR0700210

4. Article 4: DataSimilarity: an R Package for Quantifying Similarity of Datasets and for Multivariate Two- and k-Sample Testing

DataSimilarity: An R Package for Quantifying Similarity of Datasets and for Multivariate Two- and k-Sample Testing

Marieke Stolte
TU Dortmund University

Luca Sauer
TU Dortmund University

Jörg Rahnenführer
TU Dortmund University

Andrea Bommert
TU Dortmund University

Abstract

Quantifying the similarity of two or more datasets is a common task in various applications of statistics and machine learning, including two- or k-sample testing and meta- or transfer learning. We present a new R package called **DataSimilarity**, which contains a variety of methods for quantifying the similarity of datasets. The package includes 36 methods, of which 14 are completely new implementations of methods that were not available in R before. The remaining functions are wrapper functions for methods with already existing implementations that unify and simplify the various input and output formats of the different implementations and bundle the methods of many existing R packages in a single package.

Keywords: dataset similarity, two-sample testing, multi-sample testing, R.

1. Introduction

Quantifying the similarity of two or more datasets has numerous applications in statistics and machine learning. Typical example applications include two- and k-sample testing to check whether two or more distributions coincide, as well as transfer and meta-learning where the similarity of training datasets is used to transfer insights from one learning task to another. Emerging from the widespread applications, a variety of methods for quantifying the similarity of two or more datasets have been proposed in the literature. For a comprehensive review, taxonomy, and comparison based on the theoretical properties of such methods, refer to Stolte, Kappenberg, Rahnenführer, and Bommert (2024).

Unfortunately, relatively few of the methods have been implemented so far. Out of the 118 methods that Stolte *et al.* (2024) identified, an implementation could be found only for 34. These implementations are, however, spread across different programming languages and packages, and some are not even included in any package but only uploaded to GitHub as a supplement to the original article presenting that method. The newly implemented R (R Core Team 2024) package **DataSimilarity** (Stolte and Sauer 2025) fills this gap. It is available on CRAN (R Foundation for Statistical Computing 2025) and offers a large collection of

dataset similarity methods applicable to different types of data that can be used in a unified framework. The methods from the theoretical comparison (Stolte *et al.* 2024) were selected to be included in the package if they fulfilled at least one of the following properties:

1. The method is implemented in R.

2. The method fulfills at least 11 (i.e. more than half) of the criteria considered in the theoretical comparison of that article, excluding the consistency criteria.

3. The method is the best in its subclass defined in the theoretical comparison, and no other method from this subclass was chosen based on the first two criteria. The subclasses are based on the underlying ideas of the methods. For more details, see Section 2.

The package unifies the input and output format of the already existing R implementations and offers new implementations of additional methods to provide the most relevant methods in a single easy-to-use package. Former R packages usually include only a few, and in many cases, only a single method for quantifying dataset similarity. Moreover, the input and output formats of the existing packages differ widely, making it hard for practitioners to use a new method if some aspect of their data has changed. With the new package, a large set of methods can be used with the same input and output format. Moreover, the new input and output format is kept very simple. The input consists of the datasets to compare and potentially some additional parameters that are specific to the method. Supplying the datasets directly is notably simpler than some of the input formats of the original implementations, where users often had to do more pre-processing steps before calculating the dataset similarity measure. The output format in the new package is a '`htest`' object, including the dataset similarity or distance calculated with the corresponding method as a statistic, the p value (if applicable), the dataset names, the alternative, and potentially other output related to the specific method. This has the advantage that the output is automatically printed in an appealing format and that its format is standardized such that, for example, the observed statistic value calculated with each method can always be accessed in the same way, independent of the chosen method.

In Section 2, the theoretical background, and one example method for each of six application domains are explained. In Section 3, the general ideas behind the implementations in the **DataSimilarity** package are described. Afterwards, the functionality of the package is demonstrated for the six example methods (Section 4). Section 5 gives an overview of all implemented methods and their applicability. Lastly, Section 6 gives a summary and discussion of the package.

In the Appendix A, all methods are briefly described, and more detailed information on the implementation is given where appropriate.

2. Methods

In the following, we describe the general setup in the two- or k-sample problem that most of the implemented methods have in common. Moreover, we discuss the selection of the implemented methods and present one example method for each application domain in more detail.

2.1. The two- and k-sample problem

Most methods for quantifying the similarity of datasets are proposed in the literature as test statistics for two- or k-sample testing. For this, a dataset is seen as a sample from a set of random variables that follow some true underlying distribution. Often, the similarity or distance of these underlying distributions is estimated.

In the following, we assume that at least two different datasets $X^{(1)}$ and $X^{(2)}$ are given consisting of n_1 and n_2 samples $X_1^{(1)}, \ldots, X_{n_1}^{(1)} \sim F_1$ and $X_1^{(2)}, \ldots, X_{n_2}^{(2)} \sim F_2$, respectively. We assume $X_i^{(1)}, X_j^{(2)} : \mathcal{X} \to \mathbb{R}^p \, \forall i \in \{1, \ldots, n_1\}, j \in \{1, \ldots, n_2\}$ and call the p components of each sample features or variables. The two-sample problem is defined as the testing problem

$$H_0 : F_1 = F_2 \text{ vs. } H_1 : F_1 \neq F_2. \tag{1}$$

This testing problem is sometimes also called testing for homogeneity of the two distributions.

In some cases, it is assumed that there are n_i observations of a target variable Y in each dataset. However, most methods only require the feature variables and cannot deal with a target variable in a meaningful way.

Analogously to the two-sample problem, the k-sample or multi-sample problem is defined for $k \geqslant 2, k \in \mathbb{N}$, datasets $X^{(1)}, \ldots, X^{(k)}$ with sample sizes $n_i, i = 1, \ldots, k$, as

$$H_0 : F_1 = F_2 = \cdots = F_k \text{ vs. } H_1 : \exists i \neq j \in \{1, \ldots, k\} : F_i \neq F_j,$$

where F_i denotes the distribution from which each observation in the ith dataset is drawn.

Each of the considered methods can be seen as a measure of similarity or distance between the F_i, $i = 1, \ldots, k$. Not all of these methods include a hypothesis test.

We use the hat symbol to denote estimators. We denote the pooled sample as $\{Z_1, \ldots, Z_N\} = \{X_1^{(1)}, \ldots, X_{n_1}^{(1)}, \ldots, X_1^{(k)}, \ldots, X_{n_k}^{(k)}\}$, where $N = \sum_{i=1}^{k} n_i$ is the total sample size. Additionally, we assume that all Z_i are distributed independently.

2.2. Selection of methods

Previously, in a comprehensive literature review (Stolte *et al.* 2024), 118 methods were described and divided into the ten classes:

1. Comparison of cumulative distribution functions, density functions, or characteristic functions,
2. Methods based on multivariate ranks,
3. Discrepancy measures for distributions,
4. Graph-based methods,
5. Methods based on inter-point distances,
6. Kernel-based methods,
7. Methods based on binary classification,
8. Distance and similarity measures for datasets,

9. Comparison based on summary statistics, and
10. Testing approaches.

Moreover, the methods were compared with respect to 22 criteria judging their applicability, interpretability, and theoretical properties. The **DataSimilarity** package comprises 36 methods that fulfill at least one of the following properties:

1. The method is implemented in R.
2. The method is one of the top methods ordered by the highest number of fulfilled criteria and fulfills at least 11 criteria of the 20 criteria, excluding the consistency criteria.
3. The method is the best in its subclass in the theoretical comparison, and no other method from this subclass was chosen based on the first two criteria.

To avoid preferring methods that define a test over methods that do not and therefore can by definition not fulfill the consistency criteria, consistency is not counted for determining the top methods. We chose 11 as the cutoff for the number of fulfilled criteria, as this is the median number of fulfilled criteria (excluding consistency) for the already implemented methods, and it ensures that at least more than half of the criteria are fulfilled. All methods from the theoretical comparison fulfilling the above-mentioned properties are included except for the method of Weiss (1960), for which no concrete test is defined but only the general idea. Moreover, that test lacks symmetry, i.e., the test result depends on the order in which the datasets are supplied, which is highly undesirable in practice.
There are additional methods (DMMD, DFDA by Kirchler, Khorasani, Kloft, and Lippert (2020), and ME, CFS by Chwialkowski, Ramdas, Sejdinovic, and Gretton (2015); Jitkrittum, Szabó, Chwialkowski, and Gretton (2016)) implemented in Python, but these are not compatible with current versions of Python and the used packages and can therefore no longer be run directly. As these methods also performed poorly in the theoretical comparison and other methods based on similar ideas (MMD) are implemented in this package, we did not take the effort to re-implement the methods in R from scratch. The same holds for the block MMD (Zaremba 2022) for which a MATLAB implementation exists. Since it is just a block-wise estimation of the already implemented MMD, it is not included here.

2.3. Definition of example methods

In the following, we differentiate six cases with regard to the applicability of the selected methods. These are summarized in Table 1. We always indicate which method is applicable in which case. In the following, we explain one example method for each case. These methods are used later in examples for applying the **DataSimilarity** package. Brief descriptions of the remaining methods can be found in Appendix A.

1. Methods applicable to exactly two numeric datasets without target variables

One example method for this case is the Rosenbaum (2005) cross-match test. It is a graph-based method. Most graph-based methods work by constructing a similarity graph on the pooled sample and counting the edges that connect points from different samples. Here, the optimal non-bipartite matching is used, i.e., a graph where pairs of two observations

Scenario no.	No. datasets	Scale level	Target variable
1	$k = 2$	Numeric	No
2	$k \geqslant 2$	Numeric	No
3	$k = 2$	Numeric	Yes
4	$k = 2$	Categorical	No
5	$k \geqslant 2$	Categorical	No
6	$k = 2$	Categorical	Yes

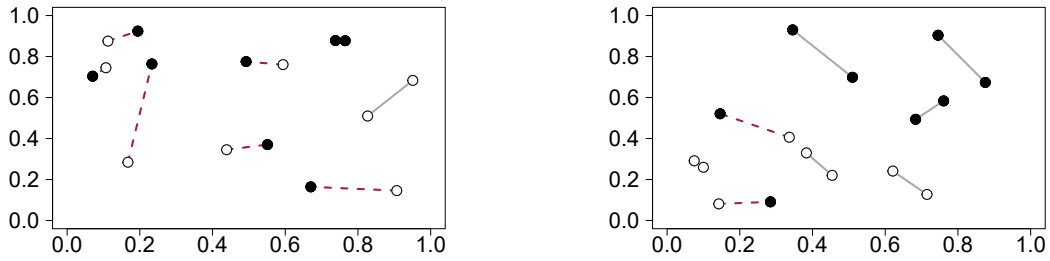
Table 1: Overview of considered cases for applicability of the dataset similarity methods. The target variable (if applicable) has to be categorical.

in the pooled sample are connected such that the sum over the edge lengths (= Euclidean distances of connected observations) is minimized. The optimal non-bipartite matching for two example data situations is shown in Figure 1. In the case of an odd pooled sample size, a ghost observation is added that has maximal distance to all real observations. This ghost observation and the observation that was matched with it are then discarded in the calculation of the test statistic.

The test statistic of the cross-match test is given by the standardized cross-match count

$$\frac{\text{CMC} - \mathsf{E}_{H_0}(\text{CMC})}{\sqrt{\mathsf{VAR}_{H_0}(\text{CMC})}},$$

where CMC denotes the cross-match count and E_{H_0} and VAR_{H_0} its expectation and variance, respectively, under $H_0 : F_1 = F_2$. The cross-match count is the number of edges connecting points from different datasets. The exact distribution of the test statistic under H_0 is known. For small samples, it can be used for computing an exact p value. For large samples, the asymptotic standard normal distribution of the test statistic can be used. The idea of the test is that for similar datasets, the number of edges connecting points from different samples is expected to be higher than in datasets that differ. This is illustrated in Figure 1a compared to Figure 1b. In the case of data drawn from different datasets, fewer edges connect points from different datasets, as indicated by the lower number of red edges in Figure 1b.



(a) Datasets drawn from the same distribution. (b) Datasets drawn from different distributions.

Figure 1: Optimal non-bipartite matching for example datasets. Dataset 1 is indicated by white points and Dataset 2 by black points. Edges connecting points from different datasets are indicated by red, dashed lines. Edges connecting points from the same sample are indicated by grey, solid lines.

2. Methods applicable to two or more numeric datasets without target variables.

The method of Mukherjee, Agarwal, Zhang, and Bhattacharya (2022) is an extension of the Rosenbaum (2005) cross-match test for multiple samples. The cross-match counts $A = (a_{12}, a_{13}, \ldots, a_{1k}, a_{23}, \ldots, a_{2k}, \ldots, a_{k-1,k})^\top$ for all pairs of datasets are calculated using the optimal non-bipartite matching on the pooled sample. The test statistic then is the Mahalanobis distance of the observed cross-counts under the null hypothesis $H_0 : F_1 = F_2 = \cdots = F_k$

$$\text{MMCM} = (A - \mathsf{E}_{H_0}(A))^\top \mathsf{COV}_{H_0}^{-1}(A)(A - \mathsf{E}_{H_0}(A)).$$

The expectation and covariance matrix of the cross-count vector A under H_0 can be calculated analytically and depend only on the sample sizes $n_i, i = 1, \ldots, k$. Small values of the multi-sample Mahalanobis cross-match (MMCM) statistic indicate similarity. However, as there is no known upperbound, it is hard to interpret the MMCM value. The MMCM statistic follows a $\chi^2_{\binom{k}{2}}$ distribution asymptotically under the null, which can be used for testing.

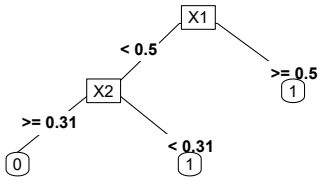
3. Methods applicable to exactly two numeric datasets with target variables

Ntoutsi, Kalousis, and Theodoridis (2008) propose measuring dataset similarity based on probability density estimates derived from decision trees. For this, it is assumed that in addition to both covariate datasets $X^{(1)}$ and $X^{(2)}$, categorical target variables $Y^{(1)}$ and $Y^{(2)}$ with the same levels are given. On each dataset $X^{(j)}$, a classification tree is constructed with $Y^{(j)}$ as the target variable, $j = 1, 2$. The splits defined by the decision trees induce a partition of the feature space $\mathcal{X}$ such that each leaf node corresponds to one segment in the partition. Figure 2 demonstrates the procedure for two example datasets. First, trees are fit to each dataset (Figure 2a and 2b). Then, the sample space is divided into segments based on the splits performed in each tree (Figure 2c and 2d). These partitions are intersected (Figure 2e) and based on the joint partition, the probability densities $P_D(\mathcal{X})$ and $P_D(Y^{(j)}, \mathcal{X})$ are estimated for $D \in \{X^{(1)}, X^{(2)}, Z\}$.

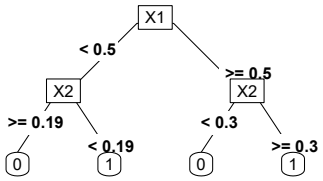
Let n_r denote the number of segments in the joint partition and n_c the number of classes of $Y^{(j)}, j = 1, 2$. $\hat{P}_D(\mathcal{X}) \in \mathbb{R}^{n_r}$ uses the proportion of observations in D that fall into each segment of the joint partition. This means that for each of the n_r segments of the partition, the number of observations from dataset D that fall into that segment is counted and divided by the total number of observations in D. For the estimation of the joint density $P_D(Y, \mathcal{X})$, the proportion of observations that fall into each segment of the joint partition and belong to each class is determined, $\hat{P}_D(Y, \mathcal{X}) \in \mathbb{R}^{n_r \times n_c}$. Here, for each of the n_r segments of the partition and each of the n_c classes, the number of observations in D where the corresponding target variable has the respective class value and that fall into the respective segment is counted and divided by the total number of observations in D. The conditional density $P_D(Y|\mathcal{X})$ is estimated by calculating the proportion of observations belonging to each class separately for each segment, $\hat{P}_D(Y|\mathcal{X}) \in \mathbb{R}^{n_r \times n_c}$. Here, for each of the n_r segments of the partition and each of the n_c classes, the number of observations in D where the corresponding target variable has the respective class value and that fall into the respective segment is counted and divided by the total number of observations in D that fall into the respective segment.

Then, Ntoutsi *et al.* (2008) consider the similarity index

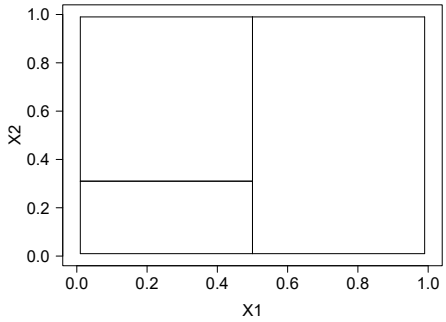
$$s(p, q) = \sum_i \sqrt{p_i \cdot q_i}$$



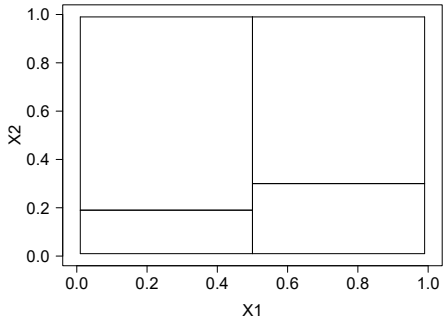
(a) Fitted Tree for Dataset 1.



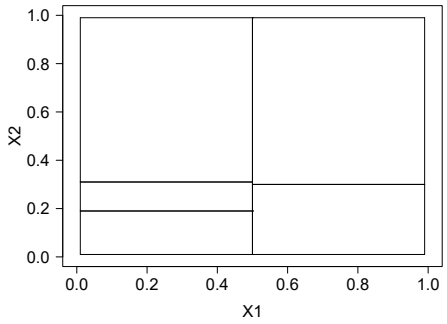
(b) Fitted Tree for Dataset 2.



(c) Partition of sample space derived from fitted tree for Dataset 1.



(d) Partition of sample space derived from fitted tree for Dataset 2.



(e) Intersected partition (greatest common refinement, GCR) from fitted trees for Datasets 1 and 2. Each dataset includes two covariates and a binary target variable.

Figure 2: Partitioning of sample space by fitting trees to two example datasets. Names in the tree nodes refer to variables in the respective datasets. 0 and 1 in the tree leaves refer to the levels of the target variables in the respective datasets.

for vectors p and q, where $(n_r \times n_c)$-matrices are interpreted as $(n_r \cdot n_c)$-dimensional vectors. For the conditional distribution, the similarity vector $S(Y|\mathcal{X}) \in \mathbb{R}^{n_r}$ is computed with $S(Y|\mathcal{X})_i = s(\hat{P}_{X^{(1)}}(Y|\mathcal{X})_{i\bullet}, \hat{P}_{X^{(2)}}(Y|\mathcal{X})_{i\bullet})$ and index $i\bullet$ denoting the i-th row. Based on this, three similarity measures for datasets are proposed:

1. NTO1 $= s(\hat{P}_{X^{(1)}}(\mathcal{X}), \hat{P}_{X^{(2)}}(\mathcal{X}))$

2. NTO2 $= s(\hat{P}_{X^{(1)}}(Y, \mathcal{X}), \hat{P}_{X^{(2)}}(Y, \mathcal{X}))$
3. NTO3 $= S(Y|\mathcal{X})^\top \hat{P}_Z(\mathcal{X})$.

All three measures have values in the interval $[0, 1]$, where high values correspond to high similarity.

4. Methods applicable to exactly two categorical datasets without target variables

Hediger, Michel, and Näf (2022) provide a two-sample test based on random forests. It is applicable for categorical data and can also be used with numeric variables or a mix of categorical and numeric variables. For this, a pooled dataset is created where each observation is labeled according to its original dataset membership, and a random forest is trained to distinguish between the dataset labels. The idea is that if the datasets are generated from the same distribution, the classification error of the random forest should be close to the chance level, otherwise, the classifier should be able to distinguish between the two distributions and hence the classification error should be lower than the chance level. One advantage of using random forests as the classifier is that it requires almost no tuning. An asymptotic test is proposed. For this, the pooled dataset has to be split into a training set on which the random forest is trained and a test set on which its classification error is evaluated to ensure that the test is performed on data that is independent of the data on which the classifier was trained. In the implementation, both datasets are split in half to create a training and a test dataset. Alternatively, an out-of-bag (OOB) based permutation test can be performed that does not require data splitting. OOB statistics can be used to increase the sample efficiency compared to the test based on a holdout sample. Both the OOB-based permutation test and the asymptotic version of the test using data splitting are implemented. The test statistic is either the mean of the per-class OOB or test classification errors, or the overall OOB or test classification error over both classes, respectively. In the asymptotic case, a binomial test is performed in case of the overall classification error, or a Z test is performed in case of the mean per-class classification error. Otherwise, a permutation test is performed. The variable importance measures of the random forest can provide additional insights into sources of distributional differences.

5. Methods applicable to two or more categorical datasets without target variables

The general idea of Lopez-Paz and Oquab (2017) is to use a classifier to determine which of two or more datasets an observation belongs to. The *classifier two-sample test (C2ST)* uses the classification accuracy of this classifier as its test statistic.
The C2ST consists of five steps:

1. Construct the dataset consisting of the samples from all datasets labeled with their membership to each dataset.
2. Assign the observations of the dataset constructed in 1. randomly to a training and test set.
3. Train a classifier that predicts to which of the datasets $X^{(j)}, j = 1, \ldots, k$, an observation belongs.

4. Calculate the C2ST statistic, which is the accuracy on the test set. The accuracy should be close to the chance level for $F_1 = \cdots = F_k$, and it should be greater than the chance level if $\exists i \neq j \in \{1, \ldots, k\} : F_i \neq F_j$ since in the latter case the classifier should identify distributional differences between the samples.

5. Calculate a p value using a binomial test for comparing the accuracy to the chance level.

Maximizing the power of a C2ST is a trade-off between using a large training set to optimize the classifier and a large test set to better evaluate the performance of the classifier.
The test statistic is interpretable as the percentage of samples that are correctly classified on the unseen test data. The above-mentioned test of Hediger *et al.* (2022) can be seen as a special case of the general framework proposed by Lopez-Paz and Oquab (2017). One difference in the implementation of the tests is that for the C2ST, categorical data is dummy coded, while for the test of Hediger *et al.* (2022) the categorical variables are passed to `ranger::ranger()` directly. Moreover, the use of OOB predictions and feature importance is specific to the random forest-based test and cannot be used for all of the available classifiers for the C2ST. Further, the C2ST uses the accuracy as its test statistic, while the test of Hediger *et al.* (2022) uses the classification error, i.e., $1 -$ accuracy.

6. Methods applicable to exactly two categorical datasets with target variables

Alvarez-Melis and Fusi (2020a) define a distance based on optimal transport between datasets that include a target (class) variable Y. The *optimal transport dataset distance* (OTDD) is defined as

$$d_{\text{OT}}(X^{(1)}, X^{(2)}) = \min_{\pi \in \Pi(F_1, F_2)} \int_{\mathcal{Z} \times \mathcal{Z}} d_{\mathcal{Z}}(z, z')^q \, \mathrm{d}\pi(z, z')$$

where $X^{(1)}, X^{(2)}$ denote the two datasets,

$$\Pi(F_1, F_2) := \{\pi_{1,2} \in \mathcal{P}(\mathcal{Z} \times \mathcal{Z}) | \pi_1 = F_2, \pi_2 = F_2\}$$

is the set of joint distributions over the product space $\mathcal{Z} \times \mathcal{Z}$ over the sample space of the pooled sample with marginal distributions F_2 and F_2, and

$$d_{\mathcal{Z}}(z, z') := (d_{\mathcal{X}}(x, x')^{q'} + W_{q'}(\alpha_y, \alpha_{y'})^{q'})^{1/q'}.$$

defines a distance of two points $z^\top = (x^\top, y)$, and $z'^\top = (x'^\top, y')$ in the pooled sample. $d_{\mathcal{X}}$ defines a distance on the covariate space, e.g., the Euclidean distance, and $W_{q'}(\alpha_y, \alpha_{y'})$ is the q'-Wasserstein distance of the distribution of the subset of covariate data with corresponding response value y and the distribution of the subset of covariate data with corresponding response value y'. The powers q and q' have to be chosen in advance to calculate the OTDD. The optimal transport problem can intuitively be motivated by imagining each probability density as a pile of dirt. Then, the cost function corresponds to the cost of transporting the dirt from one point to another, which is proportional to the distance between the two points. The optimal transport then corresponds to the lowest cost required for moving one pile of dirt fully to the shape and location of the other. Therefore, distributions can be regarded as more similar if the optimal transport between them is lower. For an intuitive explanation and visualization of the OTDD, also refer to Alvarez-Melis and Fusi (2020b).

3. General comments on implementation

Where possible, existing implementations are used. If methods already have a name in the article where they were proposed or in the secondary literature, the corresponding functions are named after that, e.g., `Wasserstein()` for the Wasserstein distance, `MMD()` for the maximum mean discrepancy (MMD), or `CMDistance()` for the constrained minimum (CM) distance. Otherwise, the function names are composed of the first letters of the surnames of all authors of the article where the respective method was originally proposed, e.g., `FR()` for the Friedman-Rafsky test proposed by Friedman and Rafsky (1979), or the full surname in case of a single author, e.g., `Bahr()` for the test proposed by Bahr (1996). The input and output of the methods from different existing packages and of the newly implemented methods are unified. To achieve this, for some existing methods, it was sufficient to implement a wrapper calling the original function.

In other cases, we re-implemented the method from scratch if the R package was archived and additional issues with the original implementation occurred. This was the case for the DiProPerm test (Wei, Lee, Wichers, and Marron 2016) for which the original implementation in the **diproperm** package (Allmon, Marron, and Hudgens 2021) yields non-reproducible results. Moreover, the implementations of the multi-sample cross-match test of Petrie (2016) and the previously mentioned multi-sample Mahalanobis cross-match test (MMCM) of Mukherjee *et al.* (2022) in the **multicross** package (Agarwal, Bhattacharya, and Zhang 2020) could not be used due to the output format that made it impossible to access the test statistic and p value. More details on the new implementations compared to the aforementioned versions can be found in Appendix A.

Each method gets two (or more) datasets as its first input parameters. After that, arguments specific to the method follow. For example, many methods perform a permutation test for which the number of permutations (`n.perm`) has to be specified. The output is of class '`htest`' and includes

- `statistic`: The test statistic
- `parameter` (optional): A parameter specifying the null distribution (e.g., degrees of freedom for a χ^2 distribution).
- `p.value`: The p value (if an asymptotic or permutation / Bootstrap test is performed).
- `estimate`: The sample estimate(s) (if available, e.g., the unstandardized edge count for edge-count tests, `NULL` for many methods).
- `alternative`: The alternative hypothesis. For two datasets, this is $F_1 \neq F_2$, for k datasets it is $\exists i \neq j \in \{1, \ldots, k\} : F_i \neq F_j$.
- `data.name`: Names of the supplied datasets.
- Further elements specific to the method (optional), e.g., the variable importances for the test of Hediger *et al.* (2022).

We use the '`htest`' class as it is widely adopted for storing results of hypothesis tests in R, and most of the implemented methods are two- or k-sample tests. Objects of class '`htest`' will be automatically printed in an appealing format using the `print.htest()` function from

the **stats** package. For methods for which no test is performed, the `p.value` is set to `NULL`. This allows pretty printing of the results and a unified output format for the corresponding functions. For many of the newly implemented permutation tests, we use the `boot()` function from the **boot** package that is included in R for implementing the permutation.

In typical applications, users should choose a test a priori and not based on test results. Therefore, the new functions perform exactly one test and return only the results corresponding to that single test. Some of the former implementations used to perform multiple tests based on the same metrics or always returned the asymptotic p value in addition to a permutation p value. This could lead to unscientific practices like choosing the test based on the desired result. As an exception, for implementations that output multiple related tests, we offer wrapper functions that also perform these multiple tests. Often, conducting them at once is computationally faster than performing each test individually when large parts of the calculation are the same. This option might be useful in certain situations where multiple tests need to be applied to the same data, e.g., when performing method comparison studies. We do not advise applying multiple tests for the same hypothesis on the same datasets when conducting inference for a specific real-life application.

Some of the existing implementations already include setting a random seed, and some do not. Therefore, for unity, the new methods all include a random seed argument and set the random seed to the supplied value for reproducibility.

4. Illustrations

In the following, the example methods for the six cases from Section 2.3 are applied to some real-world datasets. These are typically subsets of a dataset defined in such a way that, from the application background, it is clear that the subsets should or should not differ. The datasets were selected from the datasets included in the R packages that the **DataSimilarity** package suggests, so no additional packages are needed. To apply all the methods, we simply need to load the **DataSimilarity** package.

```
R> library("DataSimilarity")
```

4.1. Exactly two numeric datasets without target variables

The dataset `dhfr` (Sutherland and Weaver 2004) from the **caret** package (Kuhn and Max 2008) is a binary classification dataset (regarding Dihydrofolate Reductase inhibition) consisting of 325 compounds, of which 203 are labeled as 'active' and 122 as 'inactive'. The variables are 228 molecular descriptors. As the active and inactive compounds should differ in their descriptors, we divide the dataset according to the first variable that indicates the activity status.

```
R> data(dhfr, package = "caret")
R> act <- dhfr[dhfr$Y == "active", -1]
R> inact <- dhfr[dhfr$Y == "inactive", -1]
```

We apply the Rosenbaum cross-match test to check whether the active and inactive compounds differ. As the combined sample size is smaller than 340, we can apply the exact test:

```
R> Rosenbaum(act, inact, exact = TRUE)

        Exact cross-match test

data:  act and inact
z = -9.4098, p-value < 2.2e-16
alternative hypothesis: The distributions of act and inact are unequal.
sample estimates:
edge.count
        20
```

The cross-match count is equal to 20. At most, there could be 122 cross-matches if each observation from the 'inactive' dataset was connected to an observation in the 'active' dataset. Therefore, the cross-match count of 20 can be considered a rather small value. This is also reflected by the z score of -9.41. Consequently, we see that the hypothesis of equal distributions can be rejected with a p value smaller than $2.2 \cdot 10^{-16}$.

We obtain a warning that informs us that a ghost value was introduced when calculating the optimal non-bipartite matching due to the odd pooled sample size. This means that an artificial point was added to the sample that has the highest distance to all other points in the sample, such that the optimal non-bipartite matching, which needs an even sample size, could be calculated. The ghost value and the point with which it was matched are then discarded from the subsequent calculations.

4.2. More than two numeric datasets without target variables

The well-known `iris` dataset (Fisher 1936) included in the **datasets** package that comes with base R (R Core Team 2024) includes measurements of sepal and petals of 50 flowers each of three iris species. We compare the datasets for the three species: Iris setosa, versicolor, and virginica.

```
R> data("iris")
R> setosa <- iris[iris$Species == "setosa", -5]
R> versicolor <- iris[iris$Species == "versicolor", -5]
R> virginica <- iris[iris$Species == "virginica", -5]
```

For camparing the three datasets, we use the Mukherjee *et al.* (2022) Mahalanobis multisample crossmatch (MMCM) test for the three datasets.

```
R> MMCM(setosa, versicolor, virginica)

        Approximative MMCM test

data:  setosa, versicolor, virginica
chisq = 129.78, df = 3, p-value < 2.2e-16
alternative hypothesis: At least one pair of distributions are unequal.
```

The MMCM statistic value on its own is hard to interpret. However, the test rejects the null hypothesis of equal distributions with $p < 2.2 \cdot 10^{-16}$. Therefore, we can conclude that the

observed MMCM value presents an extreme value when assuming the null hypothesis. Thus, the datasets are dissimilar.

4.3. Exactly two numeric datasets with target variables

The `segmentationData` dataset (Hill, LaPan, Li, and Haney 2007) in the **caret** package (Kuhn and Max 2008) includes cell body segmentation data. The dataset contains 119 imaging measurements of 2019 cells to predict the segmentation that is divided into the two classes `PS` for 'poorly segmented' and `WS` for 'well segmented'. Moreover, there is a division into 1009 observations used for training and 1010 observations used as a test set. We compare this training and test set. Ideally, the distributions of the training and test set should be equal.

```
R> data(segmentationData, package = "caret")
R> test <- segmentationData[segmentationData$Case == "Test", -(1:2)]
R> train <- segmentationData[segmentationData$Case == "Train", -(1:2)]
```

To check the similarity of the training and test sets, we apply the method of Ntoutsi *et al.* (2008). For demonstration, we use all three proposed similarity measures: NTO1, NTO2, and NTO3. In all cases, we do not tune the decision trees that are used to define the partitions. The `target1` and `target2` arguments have to be specified as the column names of the target variable in the first and second supplied datasets, respectively. Here, the target variable is named `"Class"` in both cases.

```
R> NKT(train, test, target1 = "Class", target2 = "Class", tune = FALSE)

        Data similarity according to Ntoutsi et al. (2008), version 1

data:  train and test
s = 0.96931
alternative hypothesis: The distributions of train and test are unequal.

R> NKT(train, test, target1 = "Class", target2 = "Class", tune = FALSE,
+      version = 2)

        Data similarity according to Ntoutsi et al. (2008), version 2

data:  train and test
s = 0.92444
alternative hypothesis: The distributions of train and test are unequal.

R> NKT(train, test, target1 = "Class", target2 = "Class", tune = FALSE,
+      version = 3)

        Data similarity according to Ntoutsi et al. (2008), version 3

data:  train and test
s = 0.96648
alternative hypothesis: The distributions of train and test are unequal.
```

We observe high similarity between the training and test datasets with all three methods, reflected by the similarity values `s` that are all close to the maximal value 1. For the method of Ntoutsi *et al.* (2008), no test is proposed, and therefore, no p value is calculated.

4.4. Exactly two categorical datasets without target variables

The `banque` dataset from the **ade4** package (Dray and Dufour 2007) consists of bank survey data of 810 customers. All variables are categorical and contain socio-economic information of the customers. We divide the data into bank card owners and non-bank card owners and compare these two groups. In total, 243 out of the 810 customers own a bank card.

```
R> data(banque , package = "ade4")
R> card <- banque[banque$cableue == "oui", -7]
R> no.card <- banque[banque$cableue == "non", -7]
```

We use the random forest test of Hediger *et al.* (2022) to compare these two groups. For easier interpretation, we look at the overall out-of-bag (OOB) prediction error instead of the per-class OOB prediction error.

```
R> HMN(card, no.card, n.perm = 1000, statistic = "OverallOOB")
```

```
        Permutation OverallOOB random forest based two-sample test

data:  card and no.card
p.hat = 0.16076, p-value = 0.000999
alternative hypothesis: The distributions of card and no.card are unequal.
```

The overall OOB prediction error is 0.161, which is considerably smaller than the naive prediction error of 243/810 = 0.3. Therefore, the random forest is able to distinguish between the datasets, so we can conclude that the datasets differ. This is also reflected by the p value of 9.990e-04.

4.5. More than two categorical datasets without target variables

We consider the `banque` dataset from the **ade4** package (Dray and Dufour 2007) again. This time we split it by the nine socio-professional categories given by 'csp'.

```
R> data(banque, package = "ade4")
R> agric <- banque[banque$csp == "agric", -1]
R> artis <- banque[banque$csp == "artis", -1]
R> cadsu <- banque[banque$csp == "cadsu", -1]
R> inter <- banque[banque$csp == "inter", -1]
R> emplo <- banque[banque$csp == "emplo", -1]
R> ouvri <- banque[banque$csp == "ouvri", -1]
R> retra <- banque[banque$csp == "retra", -1]
R> inact <- banque[banque$csp == "inact", -1]
R> etudi <- banque[banque$csp == "etudi", -1]
```

We apply the classifier two-sample test (C2ST). First, we use the default K-NN classifier. Categorical variables are dummy-coded.

```
R> C2ST(agric, artis, cadsu, inter, emplo, ouvri, retra, inact, etudi)

        Approximative Classifier Two-Sample Test using knn

data:  agric, artis, cadsu, inter, emplo, ouvri, retra, inact, etudi
p.hat = 0.26389, size = 567.00000, prob = 0.22593, p-value =
8.041e-05
alternative hypothesis: At least one pair of distributions are unequal.
```

The accuracy of the K-NN classifier is 0.264. It is larger than the naive accuracy for always predicting the largest class, which is given by `prob = 0.226` in the output. The classifier seems to be able to distinguish between the datasets, and we can, therefore, regard them as dissimilar. Moreover, the null hypothesis of equal distributions can be rejected with a p value of 8.041e-05.

For demonstration, we additionally perform the C2ST with a multilayer perceptron classifier.

```
R> C2ST(agric, artis, cadsu, inter, emplo, ouvri, retra, inact, etudi,
+       classifier = "nnet", train.args = list(trace = FALSE))

        Approximative Classifier Two-Sample Test using nnet

data:  agric, artis, cadsu, inter, emplo, ouvri, retra, inact, etudi
p.hat = 0.25, size = 567.00000, prob = 0.22593, p-value =
0.00025
alternative hypothesis: At least one pair of distributions are unequal.
```

The results are very similar to using K-NN.

4.6. Exactly two categorical datasets with target variables

We consider the `banque` dataset from the **ade4** package (Dray and Dufour 2007) again. In this case, we interpret the savings bank amount (`eparliv`) variable as the target variable, which is again supplied via the `target1` and `target2` arguments. It is divided into the three categories '> 20000', '> 0 and < 20000', and 'nulle'. We divide the data into the socio-professional categories as before. We use the optimal transport dataset distance (OTDD) to compare the resulting datasets for craftsmen, shopkeepers, and company directors ('artis') to that of higher intellectual professions ('cadsu') and to that of manual workers ('ouvri'). As all variables are categorical, we use the Hamming distance instead of the default Euclidean distance.

```
R> OTDD(artis, cadsu, target1 = "eparliv", target2 = "eparliv",
+       feature.cost = hammingDist)
```

```
        Optimal Transport Dataset Distance

data:  artis and cadsu
OTDD = 44.166
alternative hypothesis: Distributions of artis and cadsu are unequal
```

We obtain a dataset distance of 44.166 between craftsmen/shopkeepers/company directors and executives/higher intellectual professions. For the OTDD, low values correspond to high similarity, and the minimum value is 0. The observed value is clearly larger than zero, so the datasets are not exactly similar. How dissimilar they are, however, is hard to interpret from the observed OTDD value on its own. For the OTDD, no test is proposed, and therefore, no p value is calculated.

```
R> OTDD(artis, ouvri, target1 = "eparliv", target2 = "eparliv",
+       feature.cost = hammingDist)

        Optimal Transport Dataset Distance

data:  artis and ouvri
OTDD = 49.427
alternative hypothesis: Distributions of artis and ouvri are unequal
```

We obtain a dataset distance of 49.427 between craftsmen/shopkeepers/company directors and manual workers. Again, this value on its own is hard to interpret. However, we can compare the values and conclude that the data of craftsmen/shopkeepers/company directors is more similar to that of executives/higher intellectual professions than to that of manual workers.

5. Implementation overview

Table 2 gives an overview of all wrapper functions included in the package. For each method, the original implementation, the new function name, and the applicability to data with a target variable, numerical data, categorical data, and multiple samples are given. Note that the applicability statements refer to the specific implementation of the method. Some of the methods are, in theory, applicable to a broader range of data types than those implemented. Moreover, note that most implementations are only applicable to either numerical or categorical data except for the classifier-based methods `HMN()` and `C2ST()`, which can handle both data types simultaneously as long as the selected classifier can do so. The `MMD()` implementation can also handle both data types, but a matching kernel function has to be implemented. Note that the graph-based tests cannot deal with both numerical and categorical data due to ties, even if a distance function that can handle both is supplied. More details on the methods and their implementation can be found in Appendix A.

Table 3 gives an overview of the newly implemented methods and their applicability. A few of these methods were already implemented in another programming language, as described in the implementation details in Appendix A.

Method	Original function	New function	y	Num	Cat	k>2
KMD (Huang and Sen 2024)	`KMD::KMD()`, `KMD::KMD_test()` (Huang 2022)	`KMD()`	✗	✓	✗*	✓
Friedman and Rafsky (1979)	`gTests::g.tests()` (Chen and Zhang 2017)	`FR()`	✗	✓	✓	✗
Cross-match test (Rosenbaum 2005)	`crossmatch::crossmatch()` (Heller, Small, and Rosenbaum 2024)	`Rosenbaum()`	✗	✓	✗	✗
Cramér test (Baringhaus and Franz 2004)	`cramer::cramer.test()` (Franz 2024)	`Cramer()`	✗	✓	✗	✗
Energy statistic (Székely and Rizzo 2017)	`energy::eqdist.test()` (Rizzo and Szekely 2024)	`Energy()`	✗	✓	✗	✓
Hediger *et al.* (2022)	`hypoRF::hypoRF()` (Hediger, Michel, and Naef 2024)	`HMN()`	✗	✓	✓	✗
Baringhaus and Franz (2010)	`cramer::cramer.test()` (Franz 2024)	`BF()`	✗	✓	✗	✗
Bahr (1996)	`cramer::cramer.test()` (Franz 2024)	`Bahr()`	✗	✓	✗	✗
Wasserstein distance	`Ecume::wasserstein_permut()` (Roux de Bezieux 2024)	`Wasserstein()`	✗	✓	✗	✗
Chen and Friedman (2017)	`gTests::g.tests()` (Chen and Zhang 2017)	`CF()`	✗	✓	✓	✗
Chen, Chen, and Su (2018)	`gTests::g.tests()` (Chen and Zhang 2017)	`CCS()`	✗	✓	✓	✗
Ball divergence (Pan, Tian, Wang, and Zhang 2018)	`Ball::bd.test()` (Zhu, Pan, Zheng, and Wang 2021)	`BallDivergence()`	✗	✓	✗	✓
Song and Chen (2022)	`gTestsMulti::gtestsmulti()` (Song and Chen 2023b)	`SC()`	✗	✓	✗	✓
DISCO (Rizzo and Székely 2010)	`energy::eqdist.test()` (Rizzo and Szekely 2024)	`DISCOB()`, `DISCOF()`	✗	✓	✗	✓
Zhang and Chen (2022)	`gTests::g.tests()` (Chen and Zhang 2017)	`ZC()`	✗	✓	✓	✗
RI test (Paul, De, and Ghosh 2022b)	`HDLSSkST::RItest()` (Paul, De, and Ghosh 2022a)	`RItest()`	✗	✓	✗	✓
FS test (Paul *et al.* 2022b)	`HDLSSkST::FStest()` (Paul *et al.* 2022a)	`FStest()`	✗	✓	✗	✓
Maximum Mean Discrepancy (MMD) (Gretton, Borgwardt, Rasch, Schölkopf, and Smola 2006)	`kernlab::kmmd()` (Karatzoglou, Smola, Hornik, and Zeileis 2004)	`MMD()`	✗	✓	✗*	✗
Song and Chen (2023a)	`kerTests::kertests()` (Song and Chen 2023c)	`GPK()`	✗	✓	✗*	✗
Mukhopadhyay and Wang (2020b)	`LPKsample::GLP()` (Mukhopadhyay and Wang 2020a)	`MW()`	✗	✓	✗*	✓
Chen, Dou, and Qiao (2013)	`gTests::g.tests_cat()` (Chen and Zhang 2017)	`FR_cat()`, `CF_cat()`, `CCS_cat()`, `ZC_cat()`	✗	✓	✓	✗

Classifier Two-Sample Test (Lopez-Paz and Oquab 2017)	`Ecume::classifier_test()` (Roux de Bezieux 2024)	`C2ST()`	✗	✓	✓	✓

Table 2: Implemented wrapper functions. y: Can the method deal with a target variable in the dataset? Num: Is the method as implemented applicable to numeric data? Cat: Is the method as implemented applicable to categorical data? $k > 2$: Is the method as implemented applicable to more than two datasets at a time? ✗*: Method is, in theory, applicable, but implementation does not work in this case. ✓*: Implementation can be used, although this case is not described in the literature.

Method	New function	y	Num	Cat	k>2
Mukherjee *et al.* (2022)	`MMCM()`	✗	✓	✓*	✓
Petrie (2016)	`Petrie()`	✗	✓	✓*	✓
Biswas, Mukhopadhyay, and Ghosh (2014)	`BMG()`	✗	✓	✗	✓
Deb and Sen (2021)	`DS()`	✗	✓	✗	✗
Ntoutsi *et al.* (2008)	`NKT()`	✓	✓	✗	✗
Ganti, Gehrke, Ramakrishnan, and Loh (1999)	`GGRL()`	✓	✓	✗*	✗
Alvarez-Melis and Fusi (2020a)	`OTDD()`	✓	✓	✓	✗
Jeffreys divergence	`Jeffreys()`	✗	✓	✗	✗
Biswas and Ghosh (2014)	`BG2()`	✗	✓	✗	✗
Engineer metric	`engineerMetric()`	✗	✓	✗	✗
Schilling (1986) and Henze (1988)	`SH()`	✗	✓	✗	✗
Barakat, Quade, and Salama (1996)	`BQS()`	✗	✓	✗	✗
Yu, Martin, Rothman, Zheng, and Lan (2007)	`YMRZL()`	✗	✓	✓	✗
Li, Hu, and Zhang (2022)	`LHZ()`	✗	✓	✗	✗
Constrained Minimum Distance (Tatti 2007)	`CMDistance()`	✗	✗	✓	✗
Biau and Gyorfi (2005)	`BG()`	✗	✓	✗	✓
DiProPerm test (Wei *et al.* 2016)	`DiProPerm()`	✗	✓	✗	✗

Table 3: Newly implemented functions. y: Can the method deal with a target variable in the dataset? Num: Is the method as implemented applicable to numeric data? Cat: Is the method as implemented applicable to categorical data? $k > 2$: Is the method as implemented applicable to more than two datasets at a time? ✗*: Method is, in theory, applicable, but implementation does not work in this case. ✓*: Implementation can be used, although this case is not described in the literature.

6. Summary and discussion

We presented the **DataSimilarity** package, which includes methods for quantifying the similarity of two or more datasets. Such methods are particularly popular as test statistics in two- or k-sample testing but have numerous additional applications in statistics and machine learning. The new package includes the most relevant methods from a previous review and theoretical comparison (Stolte *et al.* 2024). In total, 36 of such methods are included in the package. This covers methods that were already implemented, for which the new package includes wrapper functions to unify and simplify the various input and output formats, as well as newly implemented methods. The unified in- and output format makes the variety of methods easy to use in practice, as demonstrated in the application examples in this article. Overall, the new package is a valuable toolbox for measuring the similarity or distance of two or more datasets in general and for two or k-sample testing in particular.

Computational details

The results in this paper were obtained using R 4.4.3. The **rpart** and **rpart.plot** package (Therneau and Atkinson 2025; Milborrow 2024) were used for the visualization of fitted classification trees. All packages used and R itself are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

Acknowledgments

This work has been supported (in part) by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project P1) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation - Project Number 427806116).
We would like to thank Nabarun Deb and Bodhisattva Sen for allowing us to use their R implementation of their test for the `DS()` function in the package. Moreover, we would like to thank David Alvarez-Melis, whose Python implementation of the OTDD was the basis for our R implementation.

References

Agarwal A, Tewari A, Errickson J (2023). *rlemon: R Access to LEMON Graph Algorithms*. R package version 0.2.1, URL https://CRAN.R-project.org/package=rlemon.

Agarwal SMD, Bhattacharya B, Zhang NR (2020). *multicross: A Graph-Based Test for Comparing Multivariate Distributions in the Multi Sample Framework*. R package version 2.1.0, URL https://CRAN.R-project.org/package=multicross.

Allmon AG, Marron J, Hudgens MG (2021). *diproperm: Conduct Direction-Projection-Permutation Tests and Display Plots*. R package version 0.2.0, URL https://CRAN.R-project.org/package=diproperm.

Alvarez-Melis D, Fusi N (2020a). "Geometric Dataset Distances via Optimal Transport." In *Advances in Neural Information Processing Systems*, volume 33, pp. 21428–21439. Curran Associates, Inc.

Alvarez-Melis D, Fusi N (2020b). "Measuring dataset similarity using optimal transport." URL https://www.microsoft.com/en-us/research/blog/measuring-dataset-similarity-using-optimal-transport/.

Aslan B, Zech G (2005). "New Test for the Multivariate Two-Sample Problem Based on the Concept of Minimum Energy." *Journal of Statistical Computation and Simulation*, **75**(2), 109–119. ISSN 0094-9655. doi:10.1080/00949650410001661440.

Bahr R (1996). *Ein neuer Test für das mehrdimensionale Zwei-Stichproben-Problem bei allgemeiner Alternative*. Ph.D. thesis, Universität Hannover.

Barakat AS, Quade D, Salama IA (1996). "Multivariate Homogeneity Testing Using an Extended Concept of Nearest Neighbors." *Biometrical Journal*, **38**(5), 605–612. ISSN 1521-4036. doi:10.1002/bimj.4710380509.

Baringhaus L, Franz C (2004). "On a New Multivariate Two-Sample Test." *Journal of Multivariate Analysis*, **88**(1), 190–206. ISSN 0047-259X. doi:10.1016/S0047-259X(03)00079-4.

Baringhaus L, Franz C (2010). "Rigid Motion Invariant Two-Sample Tests." *Statistica Sinica*, **20**(4), 1333–1361. ISSN 1017-0405.

Beck C, Lu B, Greevy R (2024). *nbpMatching: Functions for Optimal Non-Bipartite Matching*. R package version 1.5.6, URL https://CRAN.R-project.org/package=nbpMatching.

Beygelzimer A, Kakadet S, Langford J, Arya S, Mount D, Li S (2024). *FNN: Fast Nearest Neighbor Search Algorithms and Applications*. R package version 1.1.4, URL https://CRAN.R-project.org/package=FNN.

Biau G, Gyorfi L (2005). "On the Asymptotic Properties of a Nonparametric L_1-Test Statistic of Homogeneity." *IEEE Transactions on Information Theory*, **51**(11), 3965–3973. ISSN 1557-9654. doi:10.1109/TIT.2005.856979.

Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, Thomas J, Ullmann T, Becker M, Boulesteix AL, Deng D, Lindauer M (2021). "Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges." *arXiv:2107.05847 [cs, stat]*.

Biswas M, Ghosh AK (2014). "A Nonparametric Two-Sample Test Applicable to High Dimensional Data." *Journal of Multivariate Analysis*, **123**, 160–171. ISSN 0047-259X. doi:10.1016/j.jmva.2013.09.004.

Biswas M, Mukhopadhyay M, Ghosh AK (2014). "A Distribution-Free Two-Sample Run Test Applicable to High-Dimensional Data." *Biometrika*, **101**(4), 913–926. ISSN 0006-3444. doi:10.1093/biomet/asu045.

Chen H, Chen X, Su Y (2018). "A Weighted Edge-Count Two-Sample Test for Multivariate and Object Data." *Journal of the American Statistical Association*, **113**(523), 1146–1155. ISSN 0162-1459. doi:10.1080/01621459.2017.1307757.

Chen H, Friedman JH (2017). "A New Graph-Based Two-Sample Test for Multivariate and Object Data." *Journal of the American Statistical Association*, **112**(517), 397–409. ISSN 0162-1459. doi:10.1080/01621459.2016.1147356.

Chen H, Zhang J (2017). *gTests: Graph-Based Two-Sample Tests*. R package version 0.2, URL https://CRAN.R-project.org/package=gTests.

Chen H, Zhang NR (2013). "Graph-Based Tests for Two-Sample Comparisons of Categorical Data." *Statistica Sinica*, **23**(4), 1479–1503. ISSN 1017-0405.

Chen L, Dou WW, Qiao Z (2013). "Ensemble Subsampling for Imbalanced Multivariate Two-Sample Tests." *Journal of the American Statistical Association*, **108**(504), 1308–1323. ISSN 0162-1459. doi:10.1080/01621459.2013.800763.

Christophe D, Petr S (2024). *randtoolbox: Generating and Testing Random Numbers*. R package version 2.0.5.

Chu L, Chen H (2019). "Asymptotic Distribution-Free Change-Point Detection for Multivariate and Non-Euclidean Data." *The Annals of Statistics*, **47**(1), 382–414. ISSN 0090-5364, 2168-8966. doi:10.1214/18-AOS1691.

Chwialkowski KP, Ramdas A, Sejdinovic D, Gretton A (2015). "Fast Two-Sample Testing with Analytic Representations of Probability Measures." In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Deb N, Sen B (2021). "Multivariate Rank-Based Distribution-Free Nonparametric Testing Using Measure Transportation." *Journal of the American Statistical Association*, **118**(541), 1–16. ISSN 0162-1459. doi:10.1080/01621459.2021.1923508.

Dray S, Dufour AB (2007). "The ade4 Package: Implementing the Duality Diagram for Ecologists." *Journal of Statistical Software*, **22**(4), 1–20. doi:10.18637/jss.v022.i04.

Dunipace EA (2024). *approxOT: approximate optimal transport*. R package version 1.1, URL https://github.com/ericdunipace/approxOT.

Fisher RA (1936). "The Use of Multiple Measurements in Taxonomic Problems." *Annals of Eugenics*, **7**(2), 179–188. ISSN 2050-1439. doi:10.1111/j.1469-1809.1936.tb02137.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1936.tb02137.x.

Franz C (2024). *cramer: Multivariate Nonparametric Cramer-Test for the Two-Sample-Problem*. R package version 0.9-4, URL https://CRAN.R-project.org/package=cramer.

Friedman JH, Rafsky LC (1979). "Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests." *The Annals of Statistics*, **7**(4), 697–717. ISSN 0090-5364.

Fukumizu K, Bach FR, Jordan MI (2004). "Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces." *Journal of Machine Learning Research*, **5**, 73–99.

Ganti V, Gehrke J, Ramakrishnan R, Loh WY (1999). "A Framework for Measuring Changes in Data Characteristics." In *Proceedings of the 18th Symposium on Principles of Database Systems*, pp. 126–137.

Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A (2006). "A Kernel Method for the Two-Sample-Problem." In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Hahsler M, Piekenbrock M, Doran D (2019). "dbscan: Fast Density-Based Clustering with R." *Journal of Statistical Software*, **91**(1), 1–30. doi:10.18637/jss.v091.i01.

Hediger S, Michel L, Naef J (2024). *hypoRF: Random Forest Two-Sample Tests*. R package version 1.0.1, URL https://CRAN.R-project.org/package=hypoRF.

Hediger S, Michel L, Näf J (2022). "On the Use of Random Forest for Two-Sample Testing." *Computational Statistics & Data Analysis*, **170**, 107435. ISSN 0167-9473. doi:10.1016/j.csda.2022.107435. URL https://www.sciencedirect.com/science/article/pii/S0167947322000159.

Heller R, Small D, Rosenbaum P (2024). *crossmatch: The Cross-Match Test*. R package version 1.4-0, URL https://CRAN.R-project.org/package=crossmatch.

Henze N (1988). "A Multivariate Two-Sample Test Based on the Number of Nearest Neighbor Type Coincidences." *The Annals of Statistics*, **16**(2), 772–783. ISSN 0090-5364.

Hill AA, LaPan P, Li Y, Haney S (2007). "Impact of Image Segmentation on High-Content Screening Data Quality for SK-BR-3 Cells." *BMC Bioinformatics*, **8**(1), 340. ISSN 1471-2105. doi:10.1186/1471-2105-8-340. URL https://doi.org/10.1186/1471-2105-8-340.

Hornik K (2005). "A CLUE for CLUster Ensembles." *Journal of Statistical Software*, **14**(12). doi:10.18637/jss.v014.i12.

Hornik K (2024). *clue: Cluster Ensembles*. R package version 0.3-66, URL https://CRAN.R-project.org/package=clue.

Huang Z (2022). *KMD: Kernel Measure of Multi-Sample Dissimilarity*. R package version 0.1.0, URL https://CRAN.R-project.org/package=KMD.

Huang Z, Sen B (2024). "A Kernel Measure of Dissimilarity between M Distributions." *Journal of the American Statistical Association*, **119**(548), 3020–3032. doi:10.1080/01621459.2023.2298036.

Jeffreys H (1997). "An Invariant Form for the Prior Probability in Estimation Problems." *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, **186**(1007), 453–461. doi:10.1098/rspa.1946.0056.

Jitkrittum W, Szabó Z, Chwialkowski KP, Gretton A (2016). "Interpretable Distribution Features with Maximum Testing Power." In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004). "kernlab – An S4 Package for Kernel Methods in R." *Journal of Statistical Software*, **11**(9), 1–20. doi:10.18637/jss.v011.i09.

Kirchler M, Khorasani S, Kloft M, Lippert C (2020). "Two-sample Testing Using Deep Learning." In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 1387–1398. PMLR. ISSN 2640-3498.

Kuhn, Max (2008). "Building Predictive Models in R Using the caret Package." *Journal of Statistical Software*, **28**(5), 1–26. doi:10.18637/jss.v028.i05. URL https://www.jstatsoft.org/index.php/jss/article/view/v028i05.

Kullback S, Leibler RA (1951). "On Information and Sufficiency." *The Annals of Mathematical Statistics*, **22**(1), 79–86. ISSN 0003-4851, 2168-8990. doi:10.1214/aoms/1177729694.

Li X, Hu W, Zhang B (2022). "Measuring and Testing Homogeneity of Distributions by Characteristic Distance." *Statistical Papers*. ISSN 1613-9798. doi:10.1007/s00362-022-01327-7.

Lopez-Paz D, Oquab M (2017). "Revisiting Classifier Two-Sample Tests." In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=SJkXfE5xx.

Maechler M, Dutang C, Goulet V (2024). *expm: Matrix Exponential, Log, 'etc'*. R package version 1.0-0, URL https://CRAN.R-project.org/package=expm.

Makiyama K (2019). *densratio: Density Ratio Estimation*. R package version 0.2.1, URL https://CRAN.R-project.org/package=densratio.

Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2024). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-16, URL https://CRAN.R-project.org/package=e1071.

Milborrow S (2024). *rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'*. R package version 3.1.2, URL https://CRAN.R-project.org/package=rpart.plot.

Muandet K, Fukumizu K, Sriperumbudur B, Schölkopf B (2017). "Kernel Mean Embedding of Distributions: A Review and Beyond." *Foundations and Trends® in Machine Learning*, **10**(1-2), 1–141. ISSN 1935-8237, 1935-8245. doi:10.1561/2200000060.

Mukherjee S, Agarwal D, Zhang NR, Bhattacharya BB (2022). "Distribution-Free Multisample Tests Based on Optimal Matchings With Applications to Single Cell Genomics." *Journal of the American Statistical Association*, **117**(538), 627–638. ISSN 0162-1459. doi:10.1080/01621459.2020.1791131.

Mukhopadhyay S, Wang K (2020a). *LPKsample: LP Nonparametric High Dimensional K-Sample Comparison*. R package version 2.1, URL https://CRAN.R-project.org/package=LPKsample.

Mukhopadhyay S, Wang K (2020b). "A Nonparametric Approach to High-Dimensional k-Sample Comparison Problems." *Biometrika*, **107**(3), 555–572. ISSN 0006-3444. doi:10.1093/biomet/asaa015.

Ntoutsi I, Kalousis A, Theodoridis Y (2008). "A General Framework for Estimating Similarity of Datasets and Decision Trees: Exploring Semantic Similarity of Decision Trees." In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)*, pp. 810–821. Society for Industrial and Applied Mathematics. ISBN 978-0-89871-654-2. doi:10.1137/1.9781611972788.73.

Pan W, Tian Y, Wang X, Zhang H (2018). "Ball Divergence: Nonparametric Two Sample Test." *The Annals of Statistics*, **46**(3), 1109–1137. ISSN 0090-5364. doi:10.1214/17-AOS1579.

Paul B, De SK, Ghosh AK (2022a). *HDLSSkST: Distribution-Free Exact High Dimensional Low Sample Size k-Sample Tests*. R package version 2.1.0, URL https://CRAN.R-project.org/package=HDLSSkST.

Paul B, De SK, Ghosh AK (2022b). "Some Clustering-Based Exact Distribution-Free k-Sample Tests Applicable to High Dimension, Low Sample Size Data." *Journal of Multivariate Analysis*, **190**, 104897. ISSN 0047-259X. doi:10.1016/j.jmva.2021.104897. URL https://www.sciencedirect.com/science/article/pii/S0047259X21001743.

Petrie A (2016). "Graph-Theoretic Multisample Tests of Equality in Distribution for High Dimensional Data." *Computational Statistics & Data Analysis*, **96**, 145–158. ISSN 0167-9473. doi:10.1016/j.csda.2015.11.003.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

R Foundation for Statistical Computing (2025). "The Comprehensive R Archive Network." URL https://cran.r-project.org/.

Rahmatallah Y, Zybailov B, Emmert-Streib F, Glazko G (2017). "GSAR: Bioconductor package for gene set analysis in R." *BMC Bioinformatics*, **18**, 61.

Rizzo M, Szekely G (2024). *energy: E-Statistics: Multivariate Inference via the Energy of Data*. R package version 1.7-12, URL https://CRAN.R-project.org/package=energy.

Rizzo ML, Székely GJ (2010). "DISCO Analysis: A Nonparametric Extension of Analysis of Variance." *The Annals of Applied Statistics*, **4**(2), 1034–1055. ISSN 1932-6157, 1941-7330. doi:10.1214/09-AOAS245.

Robert CP (1996). "Instrinsic losses." *Theory and Decision*, **40**, 191–214. doi:10.1007/BF00133173.

Rosenbaum PR (2005). "An Exact Distribution-Free Test Comparing Two Multivariate Distributions Based on Adjacency." *Journal of the Royal Statistical Society B*, **67**(4), 515–530. ISSN 1369-7412.

Roux de Bezieux H (2024). *Ecume: Equality of 2 (or k) Continuous Univariate and Multivariate Distributions*. R package version 0.9.2, URL https://CRAN.R-project.org/package=Ecume.

Sarkar S, Ghosh AK (2020). "On Perfect Clustering of High Dimension, Low Sample Size Data." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**(9), 2257–2272. ISSN 1939-3539. doi:10.1109/TPAMI.2019.2912599. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, URL https://ieeexplore.ieee.org/document/8695805.

Schilling MF (1986). "Multivariate Two-Sample Tests Based on Nearest Neighbors." *Journal of the American Statistical Association*, **81**(395), 799–806. ISSN 0162-1459. doi:10.2307/2289012.

Song H, Chen H (2022). "New Graph-Based Multi-Sample Tests for High-Dimensional and Non-Euclidean Data." doi:10.48550/arXiv.2205.13787. ArXiv:2205.13787 [stat], URL http://arxiv.org/abs/2205.13787.

Song H, Chen H (2023a). "Generalized Kernel Two-Sample Tests." *Biometrika*, pp. 755–770. ISSN 1464-3510. doi:10.1093/biomet/asad068.

Song H, Chen H (2023b). *gTestsMulti: New Graph-Based Multi-Sample Tests*. R package version 0.1.1, URL https://CRAN.R-project.org/package=gTestsMulti.

Song H, Chen H (2023c). *kerTests: Generalized Kernel Two-Sample Tests*. R package version 0.1.4, URL https://CRAN.R-project.org/package=kerTests.

Southworth LK, Kim SK, Owen AB (2009). "Properties of Balanced Permutations." *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, **16**(4), 625–638. ISSN 1557-8666. doi:10.1089/cmb.2008.0144.

Sriperumbudur B, Fukumizu K, Gretton A, Lanckriet G, Schölkopf B (2009). "Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions." In *Advances in Neural Information Processing Systems 22*, pp. 1750–1758. Max-Planck-Gesellschaft, Curran, Red Hook, NY, USA.

Sriperumbudur BK, Gretton A, Fukumizu K, Lanckriet G, Schölkopf B (2008). "Injective Hilbert space embeddings of probability measures." In *21st Annual Conference on Learning Theory (COLT 2008)*, pp. 111–122. Omnipress.

Sriperumbudur BK, Gretton A, Fukumizu K, Schölkopf B, Lanckriet GRG (2010). "Hilbert Space Embeddings and Metrics on Probability Measures." *Journal of Machine Learning Research*, **11**(50), 1517–1561. ISSN 1533-7928.

Stolte M, Kappenberg F, Rahnenführer J, Bommert A (2024). "Methods for Quantifying Dataset Similarity: A Review, Taxonomy and Comparison." *Statistics Surveys*, **18**, 163–298. ISSN 1935-7516. doi:10.1214/24-SS149.

Stolte M, Sauer L (2025). *DataSimilarity: Quantifying Similarity of Datasets and Multivariate Two- And k-Sample Testing*. R package version 0.2.0, URL https://CRAN.R-project.org/package=DataSimilarity.

Sugiyama M, Liu S, du Plessis MC, Yamanaka M, Yamada M, Suzuki T, Kanamori T (2013). "Direct Divergence Approximation between Probability Distributions and Its Applications in Machine Learning." *Journal of Computing Science and Engineering*, **7**(2), 99–111. ISSN 1976-4677. doi:10.5626/JCSE.2013.7.2.99.

Sutherland JJ, Weaver DF (2004). "Three-Dimensional Quantitative Structure-Activity and Structure-Selectivity Relationships of Dihydrofolate Reductase Inhibitors." *Journal of Computer-Aided Molecular Design*, **18**(5), 309–331. ISSN 0920-654X. doi:10.1023/b:jcam.0000047814.85293.da.

Szabo A, Boucher K, Carroll WL, Klebanov LB, Tsodikov AD, Yakovlev AY (2002). "Variable selection and pattern recognition with gene expression data generated by the microarray technology." *Mathematical Biosciences*, **176**(1), 71–98. ISSN 0025-5564. doi:10.1016/s0025-5564(01)00103-1.

Székely GJ, Rizzo ML (2017). "The Energy of Data." *Annual Review of Statistics and Its Application*, **4**(1), 447–479. doi:10.1146/annurev-statistics-060116-054026.

Tatti N (2007). "Distances between Data Sets Based on Summary Statistics." *Journal of Machine Learning Research*, **8**(1).

Therneau T, Atkinson B (2025). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1.24, URL https://CRAN.R-project.org/package=rpart.

Vaserstein LN (1969). "Markov Processes Over Denumerable Products of Spaces, Describing Large Systems of Automata." *Problemy Peredachi Informatsii*, **5**(3), 64–72.

Volker TB (2024). "densityratio: Distribution Comparison through Density Ratio Estimation." doi:10.5281/zenodo.13881689. URL https://github.com/thomvolker/densityratio.

Wei S, Lee C, Wichers L, Marron JS (2016). "Direction-Projection-Permutation for High-Dimensional Hypothesis Tests." *Journal of Computational and Graphical Statistics*, **25**(2), 549–569. ISSN 1061-8600. doi:10.1080/10618600.2015.1027773.

Weiss L (1960). "Two-Sample Tests for Multivariate Distributions." *The Annals of Mathematical Statistics*, **31**(1), 159–164. ISSN 0003-4851, 2168-8990.

Yu K, Martin R, Rothman N, Zheng T, Lan Q (2007). "Two-sample Comparison Based on Prediction Error, with Applications to Candidate Gene Association Studies." *Annals of Human Genetics*, **71**(1), 107–118. ISSN 1469-1809. doi:10.1111/j.1469-1809.2006.00306.x.

Zaremba W (2022). "B - test." URL https://github.com/wojzaremba/btest.

Zhang J, Chen H (2022). "Graph-Based Two-Sample Tests for Data with Repeated Observations." *Statistica Sinica*, **32**(1), 391–415. ISSN 1017-0405. Publisher: Institute of Statistical Science, Academia Sinica, URL https://www.jstor.org/stable/27108529.

Zhu J, Pan W, Zheng W, Wang X (2021). "Ball: An R Package for Detecting Distribution Difference and Association in Metric Spaces." *Journal of Statistical Software*, **97**(6), 1–31. doi:10.18637/jss.v097.i06.

A. Implementation details

A.1. KMD (Huang and Sen 2024)

The *kernel measure of multi-sample dissimilarity* (KMD) introduced by Huang and Sen (2024) is a kernel-based test using the association between the variables and the sample membership to quantify the dissimilarity of multiple samples. Denote the dataset membership of each point in the pooled sample $\{Z_1, \ldots, Z_N\}$ by $\{\Delta_1, \ldots, \Delta_N\}$. $\{(\Delta_i, Z_i)\}_{i=1}^N$ can be seen as an i.i.d. sample from $(\tilde{\Delta}, \tilde{Z})$ with distribution μ defined by $\mathrm{P}(\tilde{\Delta} = i) = \pi_i$ and $\tilde{Z}|\tilde{\Delta} = i \sim F_i$, $i = 1, \ldots, k$. Let $(\tilde{Z}_1, \tilde{\Delta}_1), (\tilde{Z}_2, \tilde{\Delta}_2)$ be i.i.d. samples from μ and $(\tilde{Z}, \tilde{\Delta}), (\tilde{Z}, \tilde{\Delta}') \sim \mu$ with $\tilde{\Delta}, \tilde{\Delta}'$ conditionally independent given $\tilde{Z}$. Denote by K a kernel function over $\{1, \ldots, k\}$, e.g., the discrete kernel $K(x, y) := \mathbb{1}(x = y)$. Then, the KMD is defined as

$$\eta(P_1, \ldots, P_k) := \frac{\mathsf{E}\left[K(\tilde{\Delta}, \tilde{\Delta}')\right] - \mathsf{E}\left[K(\tilde{\Delta}_1, \tilde{\Delta}_2)\right]}{\mathsf{E}\left[K(\tilde{\Delta}, \tilde{\Delta})\right] - \mathsf{E}\left[K(\tilde{\Delta}_1, \tilde{\Delta}_2)\right]}.$$

It can be estimated using a similarity graph $\mathcal{G}$, e.g., the K-nearest neighbor (NN) graph or the minimum spanning tree (MST) on the pooled sample. Denote by $(Z_i, Z_j) \in \mathcal{E}(\mathcal{G})$ that there is an edge in $\mathcal{G}$ connecting Z_i and Z_j. Moreover, let o_i be the out-degree of Z_i in $\mathcal{G}$. Then, an estimator for η is defined as

$$\hat{\eta} := \frac{\frac{1}{N}\sum_{i=1}^N \frac{1}{o_i} \sum_{j:(Z_i,Z_j)\in\mathcal{E}(\mathcal{G})} K(\Delta_i, \Delta_j) - \frac{1}{N(N-1)} \sum_{i\neq j} K(\Delta_i, \Delta_j)}{\frac{1}{N}\sum_{i=1}^N K(\Delta_i, \Delta_i) - \frac{1}{N(N-1)} \sum_{i\neq j} K(\Delta_i, \Delta_j)}.$$

An asymptotic and a permutation k-sample test are proposed based on the KMD.
The implementation of the new function `KMD()` combines the calculation of KMD and the corresponding p value using the functions `KMD()` and `KMD_test()`, respectively, from the **KMD** package (Huang 2022). Moreover, the inputs of the new function are simply the individual datasets instead of the pooled data matrix and sample IDs. By default, the asymptotic test is performed (`n.perm = 0`) using a discrete kernel and the K-N graph with $K = \lceil N/10 \rceil$, where N denotes the total sample size of the pooled sample. The options for the graph are restricted to `"knn"` and `"mst"` by the implementations from the **KMD** package. A user-specified kernel can be used only when a kernel matrix is supplied instead of the keyword `"discrete"` for the `kernel` argument of the new function.

A.2. Edge-count tests (Friedman and Rafsky 1979; Chen and Zhang 2013; Chen *et al.* 2018; Zhang and Chen 2022)

The tests by Friedman and Rafsky (1979), Chen and Friedman (2017), Chen *et al.* (2018), and Zhang and Chen (2022) are graph-based two-sample tests that use the edge counts in a similarity graph like the (K-)MST on the pooled sample. They make use of the number of edges that connect points within the first sample, R_1, the number of edges that connect points within the second sample, R_2, and the number of edges that connect points from different samples R_{12}. The original edge-count test by Friedman and Rafsky (1979) takes the standardized between-sample edge-count

$$T_{\mathrm{FR}} = \frac{R_{12} - \mathsf{E}_{H_0}(R_{12})}{\sqrt{\mathsf{VAR}_{H_0}(R_{12})}}$$

as its test statistic. The expectation and variance under the null can be calculated analytically. Chen and Friedman (2017) noted that this has low power against scale alternatives and proposed the *generalized edge-count test* using

$$T_{\mathrm{CF}} = (R_1 - \mathsf{E}_{H_0}(R_1), R_2 - \mathsf{E}_{H_0}(R_2))\, \mathsf{COV}_{H_0}^{-1} \left(\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} \right) \begin{pmatrix} R_1 - \mathsf{E}_{H_0}(R_1) \\ R_2 - \mathsf{E}_{H_0}(R_2) \end{pmatrix}.$$

Chen *et al.* (2018) found some problems with the original edge-count test for unequal sample sizes of the two datasets, based on which they proposed the *weighted edge-count test* using the weighted edge-counts

$$R_w = \frac{n_1}{N} R_1 + \frac{n_2}{N} R_2,$$

where n_1 denotes the sample size of the first dataset, n_2 the sample size of the second dataset, and $N = n_1 + n_2$ the total sample size in the pooled sample. The *weighted edge-count test* statistic is then defined as the standardized weighted edge count

$$T_{\mathrm{CCS}} = \frac{R_w - \mathsf{E}_{H_0}(R_w)}{\sqrt{\mathsf{VAR}_{H_0}(R_w)}}.$$

Lastly, the *max-type edge count* (Zhang and Chen 2022) test (based on the method of Chu and Chen (2019)) additionally uses the difference of the edge counts in the samples, i.e.,

$$R_d = R_1 - R_2.$$

Its test statistic is defined as

$$T_{\mathrm{ZC}} = \max \left(\kappa \frac{R_w - \mathsf{E}_{H_0}(R_w)}{\sqrt{\mathsf{VAR}_{H_0}(R_w)}}, \left| \frac{R_d - \mathsf{E}_{H_0}(R_d)}{\sqrt{\mathsf{VAR}_{H_0}(R_d)}} \right| \right),$$

where κ is a constant that has to be chosen prior to performing the test. $\kappa \in \{1, 1.14, 1.31\}$ is recommended based on a small power simulation for normal data with shift or scale alternatives.

Wrapper functions around `g.tests()` from the **gTests** package (Chen and Zhang 2017) are implemented. These do not need a pre-calculated graph as input but allow specifying a distance function (`dist.fun`) and a function for calculating a similarity graph (`graph.fun`) and then calculating the similarity graph internally. The new input also includes both datasets. We find this more intuitive and less error-prone than supplying an edge matrix and two vectors of indices specifying the dataset membership as for the original `g.tests()` function. The new implementation forces the user to choose one of the tests first and then perform it instead of performing all tests at once. Moreover, the users have to decide whether they want to perform the permutation test or the approximative test.

For the Friedman-Rafsky test, there is an additional implementation in the **GSAR** package (Rahmatallah, Zybailov, Emmert-Streib, and Glazko 2017), but there, the test statistic is standardized by the empirical mean and standard deviation rather than the theoretical mean and standard deviation of the test statistic under the null hypothesis as proposed in the original article. Therefore, we use the **gTests** implementation here.

A.3. Edge-count tests for categorical data (Chen and Zhang 2013; Zhang and Chen 2022)

These methods are adaptations of the previously mentioned edge-count tests for categorical data. With categorical data, the problem of ties in the distance matrix arises. Ties lead to

non-unique solutions for the similarity graph construction and, therefore, also to non-unique values of the proposed test statistics. This can be solved by either taking the union of all optimal graphs and calculating the respective statistic on this union graph or by averaging the test statistic values over all optimal graphs. The new implementation of the categorical graph-based tests is again a wrapper function that includes the calculation of the edge matrix. For this, the function `getGraph()` from the **gTests** package is used. Therefore, the choice of the similarity graph is restricted to the K-nearest neighbors and the K-MST. Still, a distance function can be supplied. By default, this is the sum of unequal classes. The calculation of the frequency table of all observations and the similarity graph on this are performed internally; thus, again, only the datasets have to be supplied by the user. Moreover, the method for aggregating the graphs has to be supplied. Possible options are averaging (`"a"`) and union (`"u"`) over graphs.

A.4. Cross-match test (Rosenbaum 2005)

The Rosenbaum cross-match test uses a similar approach as the Friedman-Rafsky test but is based on the optimal non-bipartite matching instead of the MST as a similarity graph (see Section 2.3). The new function `Rosenbaum()` is a wrapper around the `crossmatchtest()` function from the **crossmatch** package (Heller *et al.* 2024). Again, a distance function can be supplied. By default, this is `stats::dist()`, i.e., the Euclidean distance. The new function then calculates the distance matrix internally. Again, we find this more straightforward from a user perspective than supplying a distance matrix on the pooled sample and a vector specifying the dataset membership of each observation. The output of the function includes the raw edge count, its standard error, and expectation under the null like for the **crossmatch** implementation. In contrast, only either the exact or the approximative p value is returned. By default (`exact = TRUE`), the exact p value is returned. This is appropriate for samples that are not too large. Note that with a pooled sample size of 340 or more, it is numerically impossible to derive the exact distribution due to the factorials involved in the calculation, and `crossmatchtest()` will return a missing value for the exact p value.

A.5. Energy statistic and generalizations by Baringhaus and Franz (2010)

The energy statistic is a popular two- and k-sample statistic based on interpoint distances. The k-sample statistic is defined as

$$T_{\text{Energy}} = \sum_{1 \leqslant i < j \leqslant k} \frac{n_i n_j}{n_i + n_j} \left(\frac{2}{n_i n_j} \sum_{u=1}^{n_i} \sum_{v=1}^{n_j} \|X_u^{(i)} - X_v^{(j)}\|_2 - \frac{1}{n_i^2} \sum_{u=1}^{n_i} \sum_{v=1}^{n_i} \|X_u^{(i)} - X_v^{(i)}\|_2 - \frac{1}{n_j^2} \sum_{u=1}^{n_j} \sum_{v=1}^{n_j} \|X_u^{(j)} - X_v^{(j)}\|_2 \right).$$

For a comprehensive review of the literature on the energy statistic and its applications, please refer to Székely and Rizzo (2017). A permutation test can be performed based on the energy statistic. In the two-sample case, the energy statistic is equal to two times the Cramér test statistic of Baringhaus and Franz (2004), and therefore, the tests are equivalent. However, a Bootstrap instead of a permutation test is proposed for the Cramér test. Baringhaus and Franz (2010) propose a test statistic that generalizes the Cramér test statistic by using a continuous function ϕ such that $\phi(\|x - y\|^2)$ is a negative definite kernel instead of the

Euclidean distances. Different examples for ϕ are given, including as special cases the Cramér test, the test by Bahr (1996), and the test by Szabo, Boucher, Carroll, Klebanov, Tsodikov, and Yakovlev (2002). Overall, $\phi(z) = \log(1 + z)$ is recommended for general alternatives based on a simulation study, and the Cramér test is recommended for location alternatives. The tests of Baringhaus and Franz (2010) are implemented in the **cramer** package (Franz 2024). The new implementation is a simple wrapper to unify input and output naming and types. The energy statistic is implemented in the **energy** package (Rizzo and Szekely 2024). For the corresponding wrapper, the input type was changed to a greater extent since the original implementation had the pooled sample and the sample sizes as the input. The **energy** implementation outsourced the calculation of the energy statistic to C, which gives it a notable advantage with regard to computing time over the **cramer** implementation.

A.6. Random forest-based test (Hediger *et al.* 2022)

The random forest-based method of Hediger *et al.* (2022) is briefly described above in Section 2.3. The function here is a wrapper around the `hypoRF()` function from the **hypoRF** package (Hediger *et al.* 2024) that only renames arguments for consistency with the other methods. Note that the implemented per-class OOB statistics differ for the permutation test and the approximate test: for the permutation test, the sum of the per-class OOB errors is returned; for the asymptotic version, the standardized sum is returned.

A.7. Wasserstein distance

The q-Wasserstein distance (Vaserstein 1969) of two distributions F_1 and F_2 on $\mathcal{X}$ is defined as

$$\mathrm{W}(F_1, F_2) := \left(\min_{\pi \in \Pi(F_1, F_2)} \int_{\mathcal{X} \times \mathcal{X}} d_{\mathcal{X}}(x, y)^q \, \mathrm{d}\pi(x, y) \right)^{1/q},$$

where $d_{\mathcal{X}}$ is the metric that $\mathcal{X}$ is provided with, and

$$\Pi(F_1, F_2) := \{\pi_{1,2} \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) | \pi_1 = F_1, \pi_2 = F_2\}$$

is the set of joint distributions over the product space $\mathcal{X} \times \mathcal{X}$ with marginal distributions F_1 and F_2.
In the **Ecume** package (Roux de Bezieux 2024), a permutation test based on the Wasserstein distance is implemented.

A.8. Ball divergence (Pan *et al.* 2018)

The Ball divergence measures the difference between two probability measures. It is defined as the square of the measure difference over a given closed ball collection. It can be estimated as

$$\widehat{\mathrm{BD}} = A + C,$$

where

$$A = \frac{1}{n_1^2} \sum_{i,j=1}^{n_1} \left(A_{ij}^{(1)} - A_{ij}^{(2)} \right)^2,$$

$$C = \frac{1}{n_2^2} \sum_{l,m=1}^{n_2} \left(C_{lm}^{(1)} - C_{lm}^{(2)} \right)^2,$$

and

$$A_{ij}^{(1)} = \frac{1}{n_1} \sum_{u=1}^{n_1} \mathbb{1}(X_u^{(1)} \in \bar{B}(X_i^{(1)}, d(X_i^{(1)}, X_j^{(1)}))),$$
$$A_{ij}^{(2)} = \frac{1}{n_2} \sum_{v=1}^{n_2} \mathbb{1}(X_v^{(2)} \in \bar{B}(X_i^{(1)}, d(X_i^{(1)}, X_j^{(1)}))),$$
$$C_{lm}^{(1)} = \frac{1}{n_1} \sum_{u=1}^{n_1} \mathbb{1}(X_u^{(1)} \in \bar{B}(X_l^{(2)}, d(X_l^{(2)}, X_m^{(2)}))),$$
$$C_{lm}^{(2)} = \frac{1}{n_2} \sum_{v=1}^{n_2} \mathbb{1}(X_v^{(2)} \in \bar{B}(X_l^{(2)}, d(X_l^{(2)}, X_m^{(2)}))),$$

with $\bar{B}(X_i^{(l)}, d(X_i^{(l)}, X_j^{(l)}))$ denoting the closed Ball around $X_i^{(l)}$ with radius equal to the distance d of the points $X_i^{(l)}$ and $X_j^{(l)}, l \in \{1, 2\}$. Therefore, the first part of the Ball divergence, A, consists of squared distances of proportions of data points from the first sample lying within closed balls around data points from the first sample and of data points from the second sample lying within closed balls around data points from the first sample. The second part, C, consists of squared distances of proportions of data points from the first sample lying within closed balls around data points from the second sample and of data points from the second sample lying within closed balls around data points from the second sample. For both parts, the mean over all such Balls with radii equal to the distances of the center point of the ball to all other points from the same sample is taken. For multiple samples, the pairwise test statistics can be summarized by summing up the pairwise divergences, or by taking the maximum of sums of the Ball divergences from each sample to all other samples, or by summing up the largest $k-1$ pairwise Ball divergences.
The implementation here is a wrapper around the `bd.test()` function from the **Ball** package (Zhu *et al.* 2021). In contrast to the original implementation, the new wrapper returns an object of class '`htest`' in the multi-sample case, although, in that case, no test is conducted. Moreover, only the summarized statistic according to the specified `kbd.type`, which determines how the pairwise Ball divergences are summarized, is returned.

A.9. Multisample graph-based tests (Song and Chen 2022)

Song and Chen (2022) propose three new tests for the k-sample problem that use the between-sample edges and the within-sample edges of a similarity graph on the pooled sample. Let R^W denote the vector containing the numbers of within-sample edges for each of the k samples and R^B denote the vector containing the numbers of between-sample edges for all $k(k-1)$ pairs of different samples. Then, the first test statistic is given by

$$S = S^W + S^B, \text{ where}$$
$$S^W = \left(R^W - \mathsf{E}_{H_0}(R^W)\right)^\top \mathsf{COV}_{H_0}^{-1}\left(R^W\right)\left(R^W - \mathsf{E}_{H_0}(R^W)\right),$$
$$S^B = \left(R^B - \mathsf{E}_{H_0}(R^B)\right)^\top \mathsf{COV}_{H_0}^{-1}\left(R^B\right)\left(R^B - \mathsf{E}_{H_0}(R^B)\right).$$

The second test statistic is based on the vector R^A of all linearly independent numbers of edges between and within samples, i.e., all numbers of edges between all pairs of samples,

including the pairs of a sample with itself, except for the pair of sample $(k-1)$ and sample k. The test statistic is then defined as

$$S^A = \left(R^A - \mathsf{E}_{H_0}(R^A)\right)^\top \mathsf{COV}_{H_0}^{-1}\left(R^A\right)\left(R^A - \mathsf{E}_{H_0}(R^A)\right).$$

All expectations and covariances under the null can be calculated analytically again. While $\mathsf{COV}_{H_0}\left(R^W\right)$ is shown to be always invertible, no such proof exists for $\mathsf{COV}_{H_0}\left(R^B\right)$ and $\mathsf{COV}_{H_0}\left(R^A\right)$. Therefore, Song and Chen (2022) suggest checking the invertability numerically before applying the test and using a generalized inverse if necessary. This is already done within their implementation. Based on S^A, an asymptotic test can easily be performed. The asymptotic distribution of S is more complicated and hard to compute in practice, therefore, a fast test is suggested instead. It combines the tests using S^W and S^B and takes the Bonferroni-adjusted p value of both these tests. Alternatively, a permutation test can be performed for either S^A or S.

The implementation here for the test of Song and Chen (2022) is a wrapper around the `gtestsmulti()` function form **gTestsMulti** (Song and Chen 2023b). The input is simplified as for the wrapper around `g.tests()`. The user has to choose whether the original (S) or the fast (S^A) version of the test should be performed. If the number of permutations for the permutation test (`n.perm`) is set to 0, the approximate test is performed; otherwise, the permutation p value is reported.

A.10. DISCO

Rizzo and Székely (2010) show that the energy test can be seen as the treatment sum of squares in an ANOVA interpretation of the k-sample problem. As the measure of dispersion for univariate or multivariate responses based on all pairwise distances between-sample elements for ANOVA

$$d_\alpha(X^{(1)}, X^{(2)}) = \frac{n_1 n_2}{n_1 + n_2}[2g_\alpha(X^{(1)}, X^{(2)}) - g_\alpha(X^{(1)}, X^{(1)}) - g_\alpha(X^{(2)}, X^{(2)})]$$

is proposed with

$$g_\alpha(X^{(i)}, X^{(j)}) = \frac{1}{n_i n_j}\sum_{u=1}^{n_i}\sum_{v=1}^{n_j}\|X_u^{(i)} - X_v^{(j)}\|_2^\alpha, i, j \in \{1, 2\}.$$

With this, Rizzo and Székely (2010) derive their so-called *distance components (DISCO) decomposition* for $\alpha \in (0, 2]$. It partitions the total dispersion in the samples

$$T_\alpha = \frac{N}{2} g_\alpha(Z, Z),$$

into components

$$T_\alpha = S_\alpha + W_\alpha$$

analogous to the variance components in ANOVA. Here, Z denotes the pooled sample. The between-sample measure of dispersion S_α and the within-sample measure of dispersion W_α, respectively, are defined as

$$S_\alpha = \sum_{1 \leqslant i < j \leqslant k} \frac{n_i + n_j}{2N} d_\alpha(X^{(i)}, X^{(j)}),$$

$$W_\alpha = \sum_{i=1}^{k} \frac{n_i}{2} g_\alpha(X^{(i)}, X^{(i)}).$$

The between-sample measure of dispersion S_α can be used directly in a k-sample permutation test (`DISCOB()`). Alternatively, the statistic

$$F_\alpha = \frac{S_\alpha/(k-1)}{W_\alpha/(N-k)}$$

can be used in a k-sample permutation test (`DISCOF()`). For each index $\alpha \in (0, 2)$, this determines a nonparametric test for the multi-sample problem that is statistically consistent against general alternatives. For $\alpha = 2$, it equals the usual ANOVA F-test. The choice of the index α is difficult. In general, the computational costs for calculating Gini means g_α, in terms of which the test statistic can be formulated, are $\mathcal{O}(N^2)$. For $\alpha = 1$, it can be linearized, and computation time reduces to $\mathcal{O}(N \log N)$. The simplest and most natural choice for α is one. For heavy-tailed distributions, a small α is recommended.
The test is implemented by permutation Bootstrap in the `R` package `energy` (Rizzo and Szekely 2024). The new implementations of the between-sample and of the DISCO F-test are wrappers, which mainly unify the inputs and outputs that differed between the two tests in the original implementation. Moreover, the input format is again changed from the pooled sample and the dataset labels to the individual datasets.

A.11. (Modified / multiscale / aggregated) RI and FS test

Paul *et al.* (2022b) propose distribution-free k-sample tests intended for the high dimension low sample size (HDLSS) setting. The tests are based on clustering the pooled sample and comparing the resulting clustering to the true dataset membership via a contingency table. If the datasets come from the same distribution, the cluster and dataset membership are independent, while if the datasets come from different distributions, the clustering depends on the true dataset membership. As a clustering algorithm, Paul *et al.* (2022b) suggest using K-means based on the generalized version of the *mean absolute difference of distances (MADD)*

$$\rho_{h,\varphi}(z_i, z_j) = \frac{1}{N-2} \sum_{m \in \{1,\dots,N\} \setminus \{i,j\}} \left| \varphi_{h,\psi}(z_i, z_m) - \varphi_{h,\psi}(z_j, z_m) \right|,$$

as proposed by Sarkar and Ghosh (2020) for the HDLSS setting. Here, $z_i, i = 1, \dots, N$, denote realizations from the pooled sample and

$$\varphi_{h,\psi}(z_i, z_j) = h\left(\frac{1}{p} \sum_{l=1}^{p} \psi |z_{il} - z_{jl}| \right),$$

where $h : \mathbb{R}^+ \to \mathbb{R}^+$ and $\psi : \mathbb{R}^+ \to \mathbb{R}^+$ are continuous and strictly increasing functions. $\psi(t) = t^2$, $\psi(t) = 1 - \exp(-t)$, $\psi(t) = 1 - \exp(-t^2)$, $\psi(t) = \log(1 + t)$, and $\psi(t) = t$ are considered in combination with $h(t) = \sqrt{t}$ and $h(t) = t$. The number of clusters has to be chosen in advance. A natural choice is to set the number of clusters to the number of datasets k. For the RI test, the Rand index of the clustering is used as a test statistic. It is zero when the clustering is perfect, i.e., when the cluster membership is a permutation of the true dataset membership. The test rejects for low values since the Rand index should take higher values when all clusters have similar distributions of class labels. The critical value can be calculated using a generalized hypergeometric distribution. Due to the discreteness of the Rand index, Paul *et al.* (2022b) propose to use a randomized test. For the FS test, the

generalized Fisher's test statistic for testing for independence in a $k \times \ell$ contingency table is used. Again, a randomized test using the generalized hypergeometric distribution to find the critical values is proposed.
Paul *et al.* (2022b) additionally propose modified versions of the tests (MRI, MFS test). For these, the number of clusters is estimated from the data using the Dunn index, since setting the number of clusters to k might fail in case of multimodal distributions where a larger number of clusters might be required, where then multiple clusters can correspond to one dataset.
Moreover, multiscale versions of the tests are presented (MSRI, MSFS test) for the case where the number of clusters is unclear. The respective tests are then performed for different numbers of clusters, and the results are aggregated using a Bonferroni adjustment for the individual tests. Still, an upper limit for the number of clusters to be considered must be chosen. The implementation also includes aggregated tests (AFS / ARI test) that perform all pairwise FS / MFS or RI / MRI tests, respectively, on the samples and aggregate the results by taking the minimum test statistic value and applying a multiple testing procedure.
The tests are implemented in the R package **HDLSSkST** (Paul *et al.* 2022a). The main difference between the new wrapper functions and the original implementation is that the modified and multiscale versions of the RI and FS tests can be performed with the same function as the original tests. The test can be chosen via the newly introduced `version` argument of the `FStest()` and `RItest()` functions. One advantage of this is that the input and output formats are unified between the versions of the test. In the original implementation of the test, the elements of the output list differ both content-wise and also in their names between the tests. Moreover, the input of the tests differs slightly between the original functions for the different tests. The input is also unified to match the input of the other functions in the **DataSimilarity** package and, therefore, consists simply of the datasets instead of a pooled data matrix, a vector with the dataset affiliation of each observation, and a vector of the sample sizes. We think this is easier to understand and less error-prone from a user perspective.

A.12. MMD

The *maximum mean discrepancy* (MMD) uses a kernel mean embedding to define a metric for probability distributions. Kernel mean embeddings extend feature maps ϕ to the space of probability distributions by representing each distribution F as a mean function

$$\phi(F)(\cdot) = \mu_F(\cdot) := \int_{\mathcal{X}} K(x, \cdot)\, \mathrm{d}F(x) = \mathsf{E}_F(K(X, \cdot)),$$

where $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric and positive definite kernel function. A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ of functions on the domain $\mathcal{X}$ with kernel K is a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ with dot product $\langle \cdot, \cdot \rangle$ that satisfies the reproducing property

$$\langle f(\cdot), K(x, \cdot) \rangle = f(x) \;\Rightarrow\; \langle K(x, \cdot), K(x', \cdot) \rangle = K(x, x'),$$

such that the linear map from a function to its value at x can be seen as an inner product. Then the kernel mean embedding as given above is a transformation of the distribution F to an element in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ corresponding to the kernel K (Muandet, Fukumizu, Sriperumbudur, and Schölkopf 2017). For characteristic kernels, the

kernel mean representation captures all information about the distribution F, which implies $\|\mu_{F_1} - \mu_{F_2}\|_{\mathcal{H}} = 0 \Leftrightarrow F_1 = F_2$ (Fukumizu, Bach, and Jordan 2004; Sriperumbudur, Gretton, Fukumizu, Lanckriet, and Schölkopf 2008; Sriperumbudur, Gretton, Fukumizu, Schölkopf, and Lanckriet 2010). Therefore, the MMD measures the difference between two distributions as

$$\text{MMD}(\mathcal{H}, F_1, F_2) = \|\mu_{F_1} - \mu_{F_2}\|_{\mathcal{H}}.$$

Here, the implementation `kmmd()` from the **kernlab** package (Karatzoglou *et al.* 2004) is used. The alternative implementation from the **Ecume** package (Roux de Bezieux 2024) does not include an automatic choice of the kernel parameter. The new implementation adds a permutation test to the **kernlab** implementation.

A.13. GPK (Song and Chen 2023a)

Song and Chen (2023a) propose another kernel-based test for which they decompose the squared MMD estimator as

$$\widehat{\text{MMD}}^2 = \alpha + \beta - 2\gamma,$$

where

$$\alpha = \frac{1}{n_1(n_1-1)} \sum_{i=1}^{n_1} \sum_{\substack{j=1 \\ j \neq i}}^{n_1} K(X_i^{(1)}, X_j^{(1)}),$$

$$\beta = \frac{1}{n_2(n_2-1)} \sum_{i=1}^{n_2} \sum_{\substack{j=1 \\ j \neq i}}^{n_2} K(X_i^{(2)}, X_j^{(2)}),$$

$$\gamma = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} K(X_i^{(1)}, X_j^{(2)}).$$

As a new statistic, they propose to use

$$\text{GPK} = (\alpha - \mathsf{E}_{H_0}(\alpha), \beta - \mathsf{E}_{H_0}(\beta)) \mathsf{COV}_{H_0}^{-1} \left(\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right) \begin{pmatrix} \alpha - \mathsf{E}_{H_0}(\alpha) \\ \beta - \mathsf{E}_{H_0}(\beta) \end{pmatrix}.$$

The GPK can be decomposed into $\text{GPK} = Z_W^2 + Z_D^2$, where Z_W and Z_D are the standardized versions (with expectation and variance under H_0) of

$$W = \frac{n_1}{N}\alpha + \frac{n_2}{N}\beta,$$
$$D = n_1(n_1-1)\alpha - n_2(n_2-1)\beta.$$

Based on this observation, they further generalize W to

$$W_r = r\frac{n_1}{N}\alpha + \frac{n_2}{N}\beta$$

and Z_W to $Z_{W,r}$. Fast tests based on $Z_{W,r}$ are proposed as the asymptotic distribution of $Z_W = Z_{W,1}$ is complicated but that of $Z_{W,r}, r \neq 1$, is a standard normal under mild assumptions. One fast test fGPK uses the Bonferroni adjusted test result of the tests based on Z_D, $Z_{W,1.2} =: ZW_1$ and $Z_{W,0.8} =: ZW_2$, the other fast test fGPK_M uses the Bonferroni

adjusted test result of the tests based on $Z_{W,1.2}$ and $Z_{W,0.8}$. For GPK (as well as for fGPK and fGPK$_M$), a permutation test can be performed.
The new implementation `GPK()` based on the `kertests()` function from the **kerTests** package (Song and Chen 2023c) performs by default the fast test version instead of a permutation test, and the bandwidth parameter σ of the RBF kernel that is used as the kernel K is chosen via the median heuristic using the function `med_sigma()` of the **kerTests** package. The median heuristic sets the bandwidth of the kernel to the median value of all pairwise distances in the pooled sample (Sriperumbudur, Fukumizu, Gretton, Lanckriet, and Schölkopf 2009). When the fast test is performed, all three test statistics, ZW_1, ZW_2, and Z_D, are returned together with the asymptotic p value if `n.perm = 0` or the permutation p value if `n.perm > 0`, respectively. For the GPK statistic, only the permutation test is available as its null distribution cannot be accessed. Therefore, if the number of permutations is set to zero, the fast test is always performed. This holds even if `fast` is set to `FALSE` (with a warning).

A.14. LP test Mukhopadhyay and Wang (2020b)

For the test of Mukhopadhyay and Wang (2020b), a nonparametrically designed set of orthogonal functions (LP polynomials) is obtained by orthonormalizing a set of functions constructed as orthonormal polynomials of mid-distribution transforms. These are used for the construction of a polynomial kernel of degree 2 that encodes the similarity between two data points in the LP-transformed domain. The values of the kernel Gram matrix are then used as weights on a graph with the pooled sample as vertices. The idea is to cluster points for the graph into k groups that have higher connectivity and compare how closely this clustering is related to the true memberships of the k distributions. Then, the problem reduces to testing independence, which can be accomplished by determining whether all of the LP comeans are zero.
The test is implemented in the **LPKsample** package (Mukhopadhyay and Wang 2020a). The new implementation offers the additional option to sum over all components instead of summing over the significant components only. This might be of interest when using the statistic as a data similarity measure without testing. By default, this is disabled (`sum.all = FALSE`). When only summing over the significant components, the returned test statistic is always equal to zero when no component is significant.

A.15. C2ST (Lopez-Paz and Oquab 2017)

The *classifier two-sample test* is already described in Section 2.3. For the C2ST, the classifier can be specified by the user and defaults to K-nearest neighbors. Possible options are all models accepted by `caret::train()`. For a list of these classification models, call e.g.

```
R> names(caret::getModelInfo())[sapply(caret::getModelInfo(), function(x) {
+    "Classification" %in% x$type
+  })]
```

A.16. Multisample cross-match tests of Mukherjee *et al.* (2022) and Petrie (2016)

The tests of Mukherjee *et al.* (2022) and Petrie (2016) generalize the Rosenbaum cross-match test to multiple samples by calculating the cross-counts for all pairs of samples based on the

optimal non-bipartite matching on the pooled sample and taking the Mahalanobis distance or simply the sum of the cross-counts, respectively, as the test statistics. New functions `MMCM()` and `Petrie()` were implemented. There exist implementations of these methods in the R package **multicross** (Agarwal *et al.* 2020), but the package is archived on CRAN, and the implementation makes it impossible to access the test statistic and p value as numeric values. Therefore, here, the functions were re-implemented from scratch. To ensure that the new functions are not derivations of the **multicross** versions, they were implemented by an author who had not looked at the **multicross** implementations before. The functions implement the formulas from Section 2 of Mukherjee *et al.* (2022). The new output is again of class '`htest`' and contains the test statistic value and the p value as a numeric value. The **nbpMatching** package (Beck, Lu, and Greevy 2024) is used for calculating the optimal non-bipartite matching. Note that in case of ties in the distance matrix, the optimal non-bipartite matching might not be defined uniquely. In the current implementation, the observations in the pooled sample are ordered as supplied by the user. When searching for a match, the **nbpMatching** implementation of the optimal non-bipartite matching algorithm starts at the end of the pooled sample. Therefore, with many ties (e.g., for categorical data), observations from the first dataset are often matched with ones from the last dataset, and so on. This might affect the validity of the test negatively since, even under the null, more cross counts than expected are observed. A random ordering of the pooled sample might help solve this issue, but would result in the observed test statistic value depending on this random ordering and is therefore not implemented.

A.17. Test using the shortest Hamiltonian path (Biswas *et al.* 2014)

Biswas *et al.* (2014) suggest a graph-based test similar to those of Friedman and Rafsky (1979) and Rosenbaum (2005) but using the shortest Hamiltonian path as the similarity graph. Since calculating the Hamiltonian path is an NP hard problem, the implementation of `BMG()` is based on Kruskal's algorithm, which is a heuristic approach to find the shortest Hamilton Path within the pooled dataset as suggested in Biswas *et al.* (2014). Here, it is implemented as follows:

1. Create an edge list of the fully connected graph on the pooled sample, sorted by increasing Euclidean distance of the corresponding vertices.
2. For each edge, check if (i) an addition of this edge leads to a cyclic graph (using `IsAcyclic()` from the **rlemon** package (Agarwal, Tewari, and Errickson 2023)) and (ii) an addition of this edge leads to a degree larger than two in any (used) vertex. If both criteria are not met, keep the corresponding edge.
3. Return the reduced edge list, containing only edges needed to construct the Hamilton path.

For pooled sample sizes $N < 1030$, an exact test can be performed. For $N \geqslant 1030$, calculation of the exact runs statistic cannot be performed due to terms involved in the calculation becoming too large for representing them as floating point numbers in R. In the exact case, the p values using the null distribution of the univariate runs statistic (Biswas *et al.* 2014) are calculated. If an asymptotic test is performed, the asymptotic null distribution is used instead.

A.18. Rank Energy statistic (Deb and Sen 2021)

The test of Deb and Sen (2021) is a rank version of the Energy statistic. The multivariate ranks are assigned using optimal transport. The implementation is based on R code accompanying the original article (`https://github.com/NabarunD/MultiDistFree`). It wraps up tidied-up versions of the `computestatistic()` and `gensamdist()` given there. The implementation uses the **randtoolbox** package (Christophe and Petr 2024) for random number generation, the **clue** package (Hornik 2005, 2024) to solve the assignment problem for ranking, and the **energy** package (Rizzo and Szekely 2024) for implementation of the Energy statistic.

A.19. Decision tree-based dataset similarity: Ganti *et al.* (1999) and Ntoutsi *et al.* (2008)

The methods of Ganti *et al.* (1999) and Ntoutsi *et al.* (2008) work by determining the partition induced by a decision tree fit to each dataset and then intersecting these partitions and calculating certain probability estimates on the resulting intersection. A description of the method of Ntoutsi *et al.* (2008) is given in Section 2.3. Ganti *et al.* (1999) calculate a decision tree model for each of the two datasets and calculate the greatest common refinement (GCR) induced by these trees. That is the intersection of the partitions of the sample space induced by each tree. A visualization of the computation of the GCR is given in Figure 2. Ganti *et al.* (1999) then compare the distribution of both datasets over this GCR. Let n_r denote the number of segments of the GCR, p_i the proportion of observations of $X^{(1)}$ that map to the i-th segment, and q_i the respective proportion of observations of $X^{(2)}$ mapping to the i-th segment. Then Ganti *et al.* (1999) compare the vectors p and q by a difference function $f : \mathbb{R}^{2n_r} \to \mathbb{R}^{n_r}$ and aggregate the results from that by an aggregate function $g : \mathbb{R}^{n_r} \to \mathbb{R}$ to obtain a measure of distance between the two datasets

$$\text{GAN} = g(f(p, q)).$$

Large values then indicate differences between the datasets. They propose the absolute difference function

$$f_a(p, q)_i = |p_i - q_i|,$$

and the scaled difference function

$$f_s(p, q)_i = \begin{cases} \frac{|p_i - q_i|}{(p_i + q_i)/2}, & \text{if } (p_i + q_i) > 0, i = 1, \ldots, n_r \\ 0, & \text{otherwise} \end{cases}.$$

For the aggregate function, they propose the sum or maximum of the values from the difference function. For using the sum as the aggregate function together with either f_a or f_s, it can be shown that the GCR is optimal in the sense that it gives the lowest value over all common refinements. For using the maximum, this property is not fulfilled. Ganti *et al.* (1999) propose using a Bootstrap test procedure for assessing whether or not the two datasets are generated by the same data-generating process.

We use **rpart** package (Therneau and Atkinson 2025) for tree estimation. In the frame of a tree object fit with `rpart()`, the nodes are numbered starting with 1 at the root, following the rule that the left child node gets the ID of the parent times 2, and the right child node gets the ID of the parent times 2 plus 1. This allows us to easily trace back the decision

rules from a leaf node to the root using integer division by 2. Moreover, the split rules can be easily accessed using the `labels()` function on the tree object. We iterate over leaves and collect all split rules on each path from the leaf to the root. Suppose no upper or lower limit is specified by any split rule for a certain variable in this way. In that case, we set this limit to the minimum or maximum, respectively, of this variable over both datasets. This ensures that each observation in any of the two datasets falls into some part of the intersected partition later on. The resulting set of ranges for all variables for each leaf node gives us the partition induced by the tree. The resulting partitions are intersected as described in Ganti *et al.* (1999) and Ntoutsi *et al.* (2008). For Ntoutsi *et al.* (2008), all three methods presented in the original article (see also Section 2.3) are implemented. No test is performed. For Ganti *et al.* (1999), the difference and aggregation functions can be supplied by the users. The suggested choices f_a and f_s, i.e., taking the absolute differences between the joint probabilities calculated on the GCR or normalizing this difference with the sum of both probabilities, are readily implemented. The default difference function is set to f_a, and the default aggregation function is set to the sum. A permutation test can be performed.
Neither Ntoutsi *et al.* (2008) nor Ganti *et al.* (1999) discuss the hyperparameter choice for the decision trees. Here, we offer the options to use the default parameter settings of `rpart()` or to tune the hyperparameters. For tuning the hyperparameters, we use the `best.rpart()` function of the **e1071** package (Meyer, Dimitriadou, Hornik, Weingessel, and Leisch 2024). The parameters `minsplit`, `minbucket`, and `cp` of the tree can be tuned. The ranges that are used here for tuning are chosen based on the recommendations by Bischl, Binder, Lang, Pielok, Richter, Coors, Thomas, Ullmann, Becker, Boulesteix, Deng, and Lindauer (2021). Tuning is enabled by default but can be disabled by setting `tune` to `FALSE`. Cross-validation is used for tuning. The number of evaluations (`n.eval`) is set to 100 as a default, and the number of folds (`k`) is set to 5. Both values can be customized by the user. The remaining calculation works the same for a tuned or untuned tree model. By default, the number of permutations is set to 0, corresponding to not performing any test.

An implementation for categorical data for the method of Ganti *et al.* (1999) is also supplied. This comes with the following difficulties. If a category is only observed in one dataset and not in the other or even if just not all combinations of categories are observed, it might happen that at a certain split, not all levels of the respective variable are observed in the remaining data at that split. Then, it is unclear to which child node the missing level is assigned. In the `rpart::rpart()` implementation that we use here, the label is not assigned at all. If now in the other dataset, the combination with this label is present, the respective data points do not fit anywhere in the intersected partition. Therefore, the calculated probabilities in the joint distribution do not sum to one anymore. In these cases, a warning is printed. The function might still return a useful measure of dataset distance, but the interpretation and theoretical results might not hold anymore. Also note that for deep trees, the intersection in practice often reduces to all combinations of categories of the variables. Therefore, the measure reduces to the differences in frequency of all category combinations in these cases, but is far more complicated and time-consuming to calculate.

A.20. OTDD (Alvarez-Melis and Fusi 2020a)

A description of the optimal transport dataset distance can be found in Section 2.3. There is a Python implementation of the method (`https://github.com/microsoft/otdd`) that was used as a rough orientation here. Compared to that, the JDOT option is deprecated. The

new implementation uses the Wasserstein distance implementation from the **approxOT** package (Dunipace 2024) and the matrix square root from the **expm** package (Maechler, Dutang, and Goulet 2024). Note that the solution of the optimal transport between two distributions is given by their q-Wasserstein distance to the power of q. There are different options for the method to calculate the optimal transport based on the dataset distance. First case: chosen method is `"augmentation"`. In this case, the variable means and the covariance matrix of each dataset, reduced to each target observation value in that dataset, are calculated. The mean vector and the vectorized covariance matrix (column-wise) corresponding to the target value are appended to each observation in each dataset. Then, the q-Wasserstein distance to the power of q of these augmented datasets is calculated. Note that this calculation assumes commuting covariance matrices of all label distributions (rarely fulfilled in practice) and that the feature space metric coincides with the ground cost of the optimal transport problem on the labels (Alvarez-Melis and Fusi 2020a). Second case: chosen method is `"precomputed.labeldist"`. In this case, both the distance matrix for the label distributions and the distance matrix for the features are calculated, and the corresponding distances are added with weights `lambda.x` and `lambda.y`, respectively, to calculate a cost matrix of all observations. In case of `sinkhorn = FALSE`, i.e., for the exact calculation, only the costs from each observation from the first dataset to each observation from the second dataset are needed. When using debiased Sinkhorn approximation, additionally, the costs within each dataset are needed. For calculating the distance matrices of the label distributions, there are again different options:

1. `inner.ot.method = "exact"`. The Wasserstein distance for each label pair is calculated between the datasets reduced to the observations where the target value equals the corresponding label. There are options for using the (debiased) Sinkhorn approximation and changing the parameters of the Wasserstein distance and the ground cost metric.

2. `inner.ot.method = "gaussian.approx"`. The label distributions are approximated by Gaussians, which leads to a simple closed-form solution of the optimal transport problem that uses only the means and covariances. The calculation includes calculating multiple matrix square roots of covariance matrices, which might get costly if the number of variables is high. Moreover, this calculation fails if the estimated covariance matrix is not numerically positive semi-definite. This might happen especially for $N < p$ settings.

3. `inner.ot.method = "only.means"`. The former is further simplified by using only the means (i.e., assuming equal covariance matrices in all label distributions).

4. `inner.ot.method = "naive.upperbound"`. A distribution-agnostic upper bound for the optimal transport between the label distributions is calculated that again relies only on the means and covariance matrices of these distributions.

A.21. Jeffreys Divergence

Jeffreys divergence (Jeffreys 1997) is the symmetrized version

$$J(F_1, F_2) = \mathrm{KL}(F_1, F_2) + \mathrm{KL}(F_2, F_1)$$

of the Kullback Leibler (KL) divergence (Kullback and Leibler 1951)

$$\mathrm{KL}(F_1, F_2) := \int \log\left(\frac{f_1(x)}{f_2(x)}\right) f_1(x)\,\mathrm{d}x.$$

Within the `Jeffreys()` function, Jeffreys divergence is calculated as the sum of the two KL-divergences (Kullback and Leibler 1951) where each dataset is used as the first once. The KL-divergences are calculated using density ratio estimation as recommended in Sugiyama, Liu, du Plessis, Yamanaka, Yamada, Suzuki, and Kanamori (2013). For this, the `densratio()` function from the **densratio** package (Makiyama 2019) is used. By default, the method `KLIEP` is chosen as suggested by Sugiyama *et al.* (2013). The **densratio** package was preferred here over the alternative package **densityratio** (Volker 2024) for two reasons. It is available on CRAN and the resulting KL divergences for some simple examples of data sampled from two univariate normal distributions were not worse compared to the true values, which are known for that simple case, than the resulting KL divergences using the alternative package. The theoretical values of the KL divergence between data samples from two univariate normal distributions

$$\mathrm{KL}(F_1, F_2) = \log\left(\frac{\sigma_2}{\sigma_1}\right) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}$$

can be calculated using the mean μ_i and the standard deviation σ_i of both distributions, $i = 1, 2$, respectively (Robert 1996). For our examples, we calculated (i) the KL divergence using the `densratio()` function from the **densratio** package, (ii) the KL divergence using the `kliep()` function from the **densityratio** package, and (iii) the theoretical KL divergence using the above formula, as we sample data `X1`, `X2` from two univariate normal distributions.

```
R> library(densityratio)
R> library(densratio)
R> KL_densratio <- function(X1, X2, dens.method = "KLIEP", verbose = FALSE) {
+    new_x1 <- X1
+    ratio_obj1 <- densratio(X1, X2, method = dens.method, verbose = verbose)
+    hatR1 <- ratio_obj1$compute_density_ratio(new_x1)
+    return(mean(log(hatR1)))
+  }
R> KL_densityratio <- function(X1, X2, dens.method = "KLIEP",
+                               verbose = FALSE) {
+    new_x1 <- X1
+    ratio_obj1 <- kliep(X1, X2, progressbar = verbose)
+    hatR1 <- predict(ratio_obj1, newdata = new_x1)
+    return(mean(log(hatR1)))
+  }
R> KL_theoretical <- function(mu1 = 0, mu2 = 0, sd1 = 1, sd2 = 1) {
+    return(log(sd2 / sd1) + (sd1^2 + (mu1 - mu2)^2) / (2*sd2^2) - 0.5)
+  }
```

To compare the results, we run a small simulation study where, for each setting, we sample 100 random numbers from the setting-specific univariate normal distributions and calculate the resulting KL divergences using both density ratio estimation methods and both datasets as the first once. We repeat this ten times for each setting and return the mean KL divergences over the ten repetitions. We calculate the theoretical KL divergence once.

```
R> simKL <- function(mu1 = 0, mu2 = 0, sd1 = 1, sd2 = 1, L = 10) {
+    res12_densratio <- res21_densratio <- numeric(L)
+    res12_densityratio <- res21_densityratio <- numeric(L)
+    for(i in 1:L) {
+      X1 <- rnorm(100, mu1, sd1)
+      X2 <- rnorm(100, mu2, sd2)
+      res12_densratio[i] <- KL_densratio(X1, X2)
+      res21_densratio[i] <- KL_densratio(X2, X1)
+      res12_densityratio[i] <- KL_densityratio(X1, X2)
+      res21_densityratio[i] <- KL_densityratio(X2, X1)
+    }
+    res <- matrix(c(KL_theoretical(mu1, mu2, sd1, sd2),
+                    KL_theoretical(mu2, mu1, sd2, sd1),
+                    mean(res12_densratio), mean(res21_densratio),
+                    mean(res12_densityratio), mean(res21_densityratio)),
+                  nrow = 2)
+    colnames(res) <- c("theoretical", "densratio", "densityratio")
+    rownames(res) <- c("KL(X1, X2)", "KL(X2, X1)")
+    return(res)
+  }
```

As a first example, we use the standard normal distribution to sample both datasets. Since the underlying distribution is the same for both data sets, we expect a KL divergence of 0.

```
R> set.seed(0)
R> simKL()
```

```
           theoretical   densratio densityratio
KL(X1, X2)           0 0.008009434   0.08110945
KL(X2, X1)           0 0.014909099   0.07706323
```

Using both datasets as the first once, the calculated KL divergence using both packages is close to the theoretical value of 0. For the second example, we shift the mean of one of the distributions by one while maintaining the scale parameters.

```
R> set.seed(0)
R> simKL(mu1 = 1, mu2 = 2, sd1 = 1, sd2 = 1)
```

```
           theoretical densratio densityratio
KL(X1, X2)         0.5 0.4666214    0.4807512
KL(X2, X1)         0.5 0.4328822    0.4502845
```

Using both datasets as the first once, the theoretical KL divergence is 0.5 in both cases. The calculation of the KL divergences using both packages results in very similar values. In both cases, both versions underestimate the theoretical value slightly. For the third example, we change both the mean and the variance of one of the univariate normal distributions. Thus, we compare the KL divergences between data sampled from a standard normal distribution

and data sampled from a normal distribution with a mean of one and a variance of two. The calculated KL divergences using both packages slightly overestimate the theoretical KL divergence of 0.2784 when using the altered normally distributed data as the first data set. The version using the **densityratio** package performs slightly better. Using the standard normal data as the first data set, both versions slightly overestimate the theoretical value of 0.1591. The version using the **densratio** package is closer to the theoretical value.

```
R> set.seed(0)
R> simKL(mu1 = 1, mu2 = 1.5, sd1 = 1, sd2 = sqrt(2))

           theoretical densratio densityratio
KL(X1, X2)   0.1590736 0.1774662    0.2258464
KL(X2, X1)   0.2784264 0.1282312    0.1913686
```

A.22. Biswas and Ghosh (2014)

The statistic of Biswas and Ghosh (2014) uses inter-point distances and is defined as

$$T = ||\hat{\mu}_{D_{F_1}} - \hat{\mu}_{D_{F_2}}||_2^2, \text{ where}$$

$$\hat{\mu}_{D_{F_1}} = \left[\hat{\mu}_{F_1F_1} = \frac{2}{n_1(n_1-1)} \sum_{i=1}^{n_1} \sum_{j=i+1}^{n_1} ||X_i^{(1)} - X_j^{(1)}||, \hat{\mu}_{F_1F_2} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} ||X_i^{(1)} - X_j^{(2)}||\right]^\top,$$

$$\hat{\mu}_{D_{F_2}} = \left[\hat{\mu}_{F_1F_2} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} ||X_i^{(1)} - X_j^{(2)}||, \hat{\mu}_{F_2F_2} = \frac{2}{n_2(n_2-1)} \sum_{i=1}^{n_2} \sum_{j=i+1}^{n_2} ||X_i^{(2)} - X_j^{(2)}||\right]^\top.$$

For testing, the scaled statistic

$$\begin{aligned}
T^* &= \frac{N\hat{\lambda}(1-\hat{\lambda})}{2\hat{\sigma}_0^2} T \text{ with} \\
\hat{\lambda} &= \frac{n_1}{N}, \\
\hat{\sigma}_0^2 &= \frac{n_1 S_1 + n_2 S_2}{N}, \text{ where} \\
S_1 &= \frac{1}{\binom{n_1}{3}} \sum_{1 \leqslant i < j < l \leqslant n_1} ||X_i^{(1)} - X_j^{(1)}|| \cdot ||X_i^{(1)} - X_l^{(1)}|| - \hat{\mu}_{F_1F_1}^2 \text{ and} \\
S_2 &= \frac{1}{\binom{n_2}{3}} \sum_{1 \leqslant i < j < l \leqslant n_2} ||X_i^{(2)} - X_j^{(2)}|| \cdot ||X_i^{(2)} - X_l^{(2)}|| - \hat{\mu}_{F_2F_2}^2
\end{aligned}$$

is used as it is asymptotically χ_1^2-distributed. The new function `BG2()` implements the Biswas and Ghosh (2014) test from scratch. `stats::dist()` is used to calculate the Euclidean distance matrix on the pooled sample. The statistic T and the scaled test statistic T^* are implemented according to the formulas above. A permutation test is implemented by permuting the distance matrix, recalculating the test statistic T for the permuted distances, and calculating the p value as the proportion of permuted test statistics larger than the observed test statistic An asymptotic test is implemented using the asymptotic result from Theorem 4.1

of Biswas and Ghosh (2014), i.e., calculating the p value as `stats::pchisq(`T^*`, lower.tail = FALSE)`.

A.23. Engineer metric

The L_q-*Engineer metric* is defined as

$$\mathrm{EN}(X, Y; q) = \left[\sum_{i=1}^{p} |\mathsf{E}(X_i) - \mathsf{E}(Y_i)|^q\right]^{\min(q,1/q)} \quad \text{with } q > 0,$$

where X_i, Y_i denote the ith component of the p-dimensional random vectors $X \sim F_1$ and $Y \sim F_2$. A new function `engineerMetric()` is implemented. Since the Engineer metric is simply the L_q-distance of the expectations of two random vectors, it is estimated as the L_q-distance of the column means of the datasets. For the distance calculation, the **base** function `norm()` is used, and different options for the L_q norm are available via the `type` argument.

A.24. Schilling (1986) and Henze (1988) test

The Schilling-Henze test uses the mean within-sample edge-count, i.e.,

$$\mathrm{SH} := L := \frac{1}{KN}(R_1 + R_2)$$

in a K-nearest neighbor graph as the test statistic. It is implemented from scratch as follows.

1. Calculate K-nearest neighbor (NN) edge matrix on the pooled sample (distance function returning a distance matrix and K are inputs of the function), i.e. create a matrix where the first column is each observation number repeated K times, and the second column are the corresponding K nearest neighbors of that observation. For the calculation of the K-NN graph, a function can be supplied by the user. Pre-implemented options include a brute-force search, a wrapper for the `kNN()` function from the **dbscan** package (Hahsler, Piekenbrock, and Doran 2019), and the fast (approximative) K-NN algorithm implemented in the `get.knn()` function from the **FNN** package (Beygelzimer, Kakadet, Langford, Arya, Mount, and Li 2024).

2. Count the number L of rows where both observations come from the same sample (i.e., either both have observation number $\leqslant n_1$ or both have observation number $> n_1$).

3. Calculate the quantities $\mathsf{E}_{H_0}(L)$ and $\mathsf{VAR}_{H_0}(L)$ from proposition 2.1 in Henze (1988).

4. Calculate the standardized test statistic $L^* = (L - \mathsf{E}_{H_0}(L))/\sqrt{\mathsf{VAR}_{H_0}(L)}$.

5. When performing a permutation test, permute the distance matrix on the pooled sample, recalculate L, and calculate the proportion of permuted test statistics that are larger than the observed value of L.

6. When performing an asymptotic test, use the asymptotic normal distribution of Z as proposed in Remark 5.1 of Henze (1988).

7. The observed value of L^* is returned in the result as the `statistic`, the observed L is returned as the `estimate`.

The default for K is set to one. This is rather arbitrary based on computational speed as there is no good rule for choosing K so far proposed in the literature (Aslan and Zech 2005).

A.25. Barakat *et al.* (1996) Generalization of the Schilling-Henze Test

Barakat *et al.* (1996) generalize the Schilling-Henze nearest neighbor test to circumvent choosing the number of nearest neighbors. Their test statistic is the sum of edge counts for all values of K for the K-nearest neighbor graph. The resulting test is equivalent to a sum of Wilcoxon rank sums. It requires samples in the Euclidean space $\mathbb{R}^p$ and it is assumed that there are no ties in ranking w.r.t. to nearness.
Within our implementation, we do not explicitly calculate the K-nearest neighbor graph for all possible values of K as this would be highly inefficient. Instead, the distance matrix on the pooled sample is calculated with a user-specified distance function (Euclidean distance calculated via `stats::dist()` by default), and the column-wise orderings of the distances, excluding the diagonal elements, are calculated. Then, the cumulative numbers of the elements smaller than n_1 are calculated for the first n_1 columns of the orderings, corresponding to the numbers of within-sample edges in the first sample in the K-nearest neighbor graph for $K = 1, \ldots, N-1$. Analogously, the cumulative numbers of the elements greater than n_1 are calculated for the remaining n_2 columns of the orderings, corresponding to the numbers of within-sample edges in the second sample in the K-nearest neighbor graph for $K = 1, \ldots, N-1$. Lastly, all these cumulative numbers are summed up, which corresponds to the Barakat *et al.* (1996) test statistic. A permutation test is implemented using the `boot::boot()` function. For that, the distances are permuted directly, and the calculation is repeated for the permuted distance matrix, which circumvents the costly recalculation of the distances for each permutation.

A.26. Tree-based test (Yu *et al.* 2007)

Yu *et al.* (2007) propose a permutation test that uses the classification error of a classification tree that distinguishes between the two datasets. The implementation of the test is based on the `C2ST()` function as the methods work very similarly. Here, we set the classifier to `"rpart"`, i.e., a CART. Instead of the classification accuracy as for the C2ST, the classification error, i.e., 1− accuracy is returned. A permutation test is implemented using the `boot::boot()` framework, and the permutation p value is calculated as the proportion of the number + 1 of permuted test statistics smaller than or equal to the observed value divided by the number of permutations. Yu *et al.* (2007) do not propose any asymptotic test, but since their test fits into the framework of Lopez-Paz and Oquab (2017), the binomial test proposed there and implemented in the `Ecume::classifier_test()` function utilized by `C2ST()` is still valid and therefore kept in the implementation.

A.27. Characteristic distance (Li *et al.* 2022)

The characteristic distance is defined as

$$\begin{aligned}\mathrm{CD}(X,Y) = \mathsf{E}\big[\|\mathsf{E}\left(\exp\left(i\langle X'', X - X'\rangle\right)|X - X'\right) \\ - \mathsf{E}\left(\exp\left(i\langle Y, X - X'\rangle\right)|X - X'\right)\|^2\big] \\ + \mathsf{E}\big[\|\mathsf{E}\left(\exp\left(i\langle X, Y - Y'\rangle\right)|Y - Y'\right) \\ - \mathsf{E}\left(\exp\left(i\langle Y'', Y - Y'\rangle\right)|Y - Y'\right)\|^2\big],\end{aligned}$$

where X', X'' and Y', Y'' denote independent copies of $X \sim F_1$ and $Y \sim F_2$, respectively. An empirical version is obtained by replacing the conditional expectations with empirical means. The implementation calculates the empirical characteristic distance between two datasets. For both summands, Euler's formula is used for every entry of the inner product defined in Li *et al.* (2022). Both mean values are calculated, and the squared complex modulus of the difference between both means is calculated. Since the inner product leads to a symmetric matrix, only an upper triangular matrix is calculated, and the final sum is multiplied by two. For reproducability, a permutation test with `n.perm` permutations and random seed `seed` is performed.

A.28. Constrained Minimum Distance (Tatti 2007)

The *constrained minimum (CM) distance* uses a *feature function* $S : \mathcal{X} \to \mathbb{R}^m$ that maps points from the sample space $\mathcal{X}$ to a real vector. The *frequency* $\theta \in \mathbb{R}^m$ of S with respect to dataset $X^{(j)}$ is the average of the values of S

$$\theta_j = \frac{1}{N}\sum_{i=1}^{n_j} S(X_i^{(j)}), j = 1, 2.$$

The CM distance is then defined as

$$D_{\mathrm{CM}}(X^{(1)}, X^{(2)}|S)^2 = (\theta_1 - \theta_2)^\top \mathsf{COV}^{-1}(S)(\theta_1 - \theta_2)$$

with

$$\mathsf{COV}(S) = \frac{1}{|\mathcal{X}|}\sum_{\omega\in\mathcal{X}} S(\omega)S(\omega)^\top - \left(\frac{1}{|\mathcal{X}|}\sum_{\omega\in\mathcal{X}} S(\omega)\right)\left(\frac{1}{|\mathcal{X}|}\sum_{\omega\in\mathcal{X}} S(\omega)\right)^\top.$$

It has to be assumed that the feature space $\mathcal{X}$ is finite and can be enumerated. For binary data and S chosen as the conjunction function, i.e., S is one if all components of an observation are one, and zero otherwise, or as the parity function, i.e., S is one if an odd number of components of an observation are one, and zero otherwise, the CM distance reduces to

$$D_{\mathrm{CM}}(X^{(1)}, X^{(2)}|S) = 2\|\theta_1 - \theta_2\|_2.$$

This special case for binary data is implemented first. It includes the option to use either the means as features (example 3 in Tatti (2007)) or the means and covariances (example 4 in Tatti (2007)). Note that there is an error in the calculation of the covariance matrix in A.4 Proof of Lemma 8 in Tatti (2007). The correct covariance matrix has the form $\mathsf{COV}[T_{\mathcal{F}}] = 0.25I$ since $\mathsf{VAR}[T_A] = \mathsf{E}[T_A^2] - \mathsf{E}[T_A]^2 = 0.5 - 0.5^2 = 0.25$ following from the correct statement that $\mathsf{E}[T_A^2] = \mathsf{E}[T_A] = 0.5$. Therefore, formula (4) changes to $d_{CM}(D_1, D_2|S_{\mathcal{F}}) = 2||\theta_1 - \theta_2||_2$ and

the formula in example 3 changes to $d_{CM}(D_1, D_2|S_1) = 2||\theta_1 - \theta_2||_2$. Our implementation is based on these corrected formulas. If the original formula was used, the results on the same data calculated with the formula for the binary special case and the results calculated with the general formula differ by a factor of $\sqrt{2}$. For the general case for categorical data, the user has to specify a feature function S mapping a point in the sample space to a real vector. Additionally, either the covariance matrix $\mathsf{COV}[S]$, if known, or the sample space has to be given. If both are given, the supplied covariance matrix is used and not recalculated. The constrained minimum distance is calculated using Theorem 1 in Tatti (2007), i.e., the formulas given above. Therefore, the supplied or calculated $\mathsf{COV}[S]$, respectively, has to be invertible.

A.29. Biau and Gyorfi (2005)

Biau and Gyorfi (2005) test for homogeneity of two (multivariate) datasets by calculating the L_1-distance between the two empirical distributions restricted to a finite partition. For this, a finite partition of the subspace spanned by the two datasets has to be defined. By default, we define a rectangular partition under the assumption of approximately equal cell probabilities. The number of elements of the partition m_n are chosen according to the convergence criteria in Biau and Gyorfi (2005) as $n^{0.8}$, where the exponent can be varied as an argument (`exponent`). For each dimension, $m_n^{1/p} + 1$ equidistant cut-points are created along the range of both datasets to define the partition. It must be ensured that there are at least three cut-points per dimension (min, max, and one point splitting the data into two bins). The argument `eps` ensures that the partition covers all data points by adding some small value to the data range. Alternative partition functions can be provided via the `partition` argument. After calculating the partition, all data points are assigned to an element of the partition along the defined cut-points. Lastly, the L_1 distance between the empirical distribution functions restricted to the elements of the partition is calculated.

A.30. DiProPerm test (Wei *et al.* 2016)

Wei *et al.* (2016) propose their *direction-projection-permutation (DiProPerm)* test for which a univariate two-sample statistic is applied to the projection of the datasets onto the normal vector of a separating hyperplane. For this, a linear classification method like a support vector machine (SVM) or distance weighted discrimination (DWD) is used to calculate such a separating hyperplane. A permutation test is then performed for the univariate statistic applied to the projection onto the normal vector. Possible options for the univariate statistic would be the mean difference, the two-sample t-statistic, or the area under the curve (AUC). There is an implementation in the **diproperm** package (Allmon *et al.* 2021), which is currently archived. Our implementation is independent of that implementation. It has the following advantages.

- All suggested univariate two-sample statistics from the paper, i.e., mean difference, t test statistic, and AUC, are implemented. An additional two-sample statistic can be used if a suitable function is supplied via the `stat.fun` argument.
- Additional binary linear classifiers other than the DWD and SVM suggested in the original paper can easily be used by supplying a suitable function via the `dipro.fun` argument.

- The results of the new function are reproducible by setting a random seed.
- The new implementation does not rely on global variables.
- The p value is returned as numeric instead of character.
- The output is an object of class '`htest`' for pretty displaying of the results.

One restriction of the new function is that it no longer supports balanced permutation. That was necessary to ensure the reproducibility, which we consider to be a trade-off worth making since the use of balanced permutation is controversial anyway, see Southworth, Kim, and Owen (2009), and reproducibility is essential for permutation tests.

Affiliation:

Marieke Stolte
Department of Statistics
TU Dortmund University
Vogelpothsweg 87
44227 Dortmund, Germany
E-mail: stolte@statistik.tu-dortmund.de

5. Article 5: An Empirical Comparison of Methods for Quantifying the Similarity of Categorical Datasets

An Empirical Comparison of Methods for Quantifying the Similarity of Categorical Datasets

Marieke Stolte[1*] Jörg Rahnenführer[1] Andrea Bommert[1]

[1]Department of Statistics, TU Dortmund University

Abstract

Quantifying the similarity of two or more datasets has widespread applications in statistics and machine learning. The choice of an appropriate dataset similarity method is, however, difficult due to the abundance of proposed methods and the lack of neutral comparison studies. A comprehensive and neutral theoretical comparison of dataset similarity methods exists. However, no empirical comparison studies are available, especially in the case of categorical data. Here, the first step in closing this gap is taken by comparing the most promising dataset similarity measures for two or more categorical datasets selected based on the theoretical comparison. It is evaluated how well the methods detect certain differences between datasets and how computationally demanding their application is. Moreover, the methods are clustered based on their performance to find groups of methods that perform similarly well for detecting certain differences. The results show that the edge count tests perform very well when comparing two datasets (two-sample case). For certain scenarios, the constrained minimum (CM) distance performs even better. For categorical data consisting of variables with five categories each, two cases were considered for differences between the class probability distributions of each variable in different datasets. In one case, the probabilities of all categories were altered with increasing probabilities for higher classes, resulting in a skewed probability distribution. In the other case, the probability of one class was increased while the probability of another class was decreased accordingly. The CM distance and certain graph-based tests performed better in the former setting, while certain classifier-based tests performed better in the latter case. This tendency was even clearer in the multi-sample case. It might, however, also result from differences in the distance functions used by the methods, which highlights the importance of choosing distance functions appropriate for the coding and scale level of the data.

*Corresponding author, e-mail: stolte@statistik.tu-dortmund.de

1 Introduction

Methods for quantifying the similarity of two or more datasets are relevant in many applications in statistics and machine learning. One typical example application is two- and k-sample testing, where the hypothesis of equal distributions is checked. There are, however, other applications like meta- or transfer learning or the comparison of simulated and real-world datasets that do not require a hypothesis test. Stolte et al. (2024) performed an extensive review of methods for quantifying the similarity of multivariate datasets and presented a taxonomy of such methods as well as a theoretical comparison that rated the applicability, interpretability, and theoretical properties of each method. This comparison does, however, not cover the performance of the methods in practice. There are some simulations in the literature that evaluate the performance of newly presented methods or compare new methods with parametric or univariate alternatives, especially for earlier methods (Friedman and Rafsky, 1979; Schilling, 1986; Baringhaus and Franz, 2004; Rosenbaum, 2005; Yu et al., 2007; Baringhaus and Franz, 2010; Zhang and Chen, 2022). There are also some limited comparisons of dataset similarity methods for more recent methods (Székely and Rizzo, 2004; Gretton et al., 2012; Biswas et al., 2014; Biswas and Ghosh, 2014; Petrie, 2016; Chen and Friedman, 2017; Lopez-Paz and Oquab, 2017; Chen et al., 2018; Pan et al., 2018; Mukhopadhyay and Wang, 2020a; Hediger et al., 2021; Li et al., 2022; Mukherjee et al., 2022; Song and Chen, 2022a; Zaremba, 2022; Huang and Sen, 2024; Song and Chen, 2023a). None of these is a neutral comparison study in the sense of Boulesteix et al. (2013), though, since all of them are conducted in the context of presenting new methods. Neutral comparison studies are, however, very important for making informed method choices (Boulesteix et al., 2013). Moreover, previous studies almost exclusively compare power values of asymptotic or permutation / Bootstrap two- or k-sample tests for numeric data. There is a lack of studies for categorical data and for methods that do not define a two- or k-sample test.

To close this gap, an extensive comparison of methods for quantifying the similarity of categorical datasets is performed. This comparison study is neutral in the sense that its focus is the comparison itself, and none of the authors of the current study were involved in the development of any of the compared methods. The most promising methods are selected from the theoretical comparison if they fulfill any of the following three criteria:

1. The method is implemented in R.
2. The method fulfills at least 11 (i.e. more than half of the) criteria in the theoretical comparison, excluding the consistency criteria.
3. The method is the best in its subclass in the theoretical comparison, and no other method from this subclass was chosen with the first two criteria.

Consistency is not considered here to ensure that tests have no structural advantage over dataset similarity methods that define no test and thus cannot fulfill the consistency criteria. Moreover, the focus here is on methods applicable to categorical datasets that do not include a target variable since the categorical case is most neglected in the literature.

The study aims are to:

- Compare how good the methods are at detecting certain differences between datasets. It is not expected to find a single method that compares best in all scenarios (Strobl and Leisch, 2024).

- Identify groups of methods that act similarly across different analysis scenarios and determine which deviations between datasets these groups of methods can detect well.

- Find out which methods are computationally feasible and numerically stable.

Note that since the comparison is not limited to two- and k-sample tests, this study does not conduct power comparisons. A similar quantity is used instead that compares the statistic values simulated for datasets drawn from different distributions to those values simulated for datasets drawn from the same distribution.

For the two-sample case, a very good performance of different graph-based edge count tests (Friedman and Rafsky, 1979; Chen and Friedman, 2017; Chen et al., 2018; Zhang and Chen, 2022) is observed. The nearest neighbor (NN) graph using the union of graphs for dealing with ties can be recommended for low-dimensional data ($p = 2$). Denser graphs like the 5-minimum spanning tree using the union can be recommended for higher-dimensional data ($p = 10, 50$). For certain alternatives and balanced sample sizes, the constrained minimum (CM) distance (Tatti, 2007) performs even better than the edge count tests. It takes into account the differences between the datasets in the variable mean vectors.

For the k-sample case with $k = 4$, only certain classifier-based tests (Lopez-Paz and Oquab, 2017) and graph-based tests using the optimal non-bipartite matching (Petrie, 2016; Mukherjee et al., 2022) can be used. For binary data, the graph-based tests are often superior. For multinomial data, the method performance depends on the chosen alternative, with the graph-based tests performing better than the classifier-based tests under one of the considered alternatives and vice versa for the other alternative.

In both the two- and the k-sample case, it could be observed that certain graph-based tests did not work for $p = 2$, which can be attributed to the handling of ties or to the degeneration of the theoretical null distribution. Often, the classifier-based methods performed poorly for unbalanced sample sizes.

The remaining manuscript is structured as follows. In Section 2, the simulation setup is presented according to the ADEMP structure (Morris et al., 2019). Next, in Section 3, the results of the method comparison with respect to the ability to detect certain differences in datasets are presented. Afterward, in Section 4, the method performances are clustered to reveal groups of methods that work well for detecting certain differences between datasets. In Section 5, the runtime, memory consumption, and occurring numerical errors of the methods are compared. Last, in Section 6, the results are summarized and discussed, and an outlook on open research questions is given.

2 Simulation Setup

The following describes the simulation setup according to the *ADEMP* (Morris et al., 2019) structure (aims, data-generating mechanisms, estimands / other targets, methods, and performance measures).

2.1 Aims

The aims of the simulation study are to:

1. Compare dataset similarity measures with respect to their performance in detecting differences of datasets drawn from distributions that differ in certain aspects.

2. Identify groups of dataset similarity measures that act similarly across different alternatives.

3. Compare dataset similarity measures with respect to their consumption of computational resources.

2.2 Data-Generating Mechanisms

In the following, the data-generating mechanisms of the simulation study are explained. The data-generating mechanisms can be divided into two cases:

1. Comparison of two datasets (without a target variable)

2. Comparison of four datasets (without a target variable).

In each case, multiple true data-generating mechanisms for the datasets are considered. Two or more datasets are generated from the same underlying distributions or different underlying distributions. The k-sample case has different options for how many and which distributions can differ. The number of possible settings increases with increasing k. Here, only $k = 4$ is considered for the k-sample case as a compromise between comparing multiple samples but still having a reasonably low number of possible settings. For $k = 4$, there are four possible settings for how many distributions differ from each other:

a) $3 + 1$:
One distribution differs from the others, which are equal, e.g. $F_1 = F_2 = F_3 \neq F_4$.

b) $2 + 2$:
Two groups of two distributions each, where the distributions within the groups are equal but the distributions between the groups differ, e.g. $F_1 = F_2 \neq F_3 = F_4$.

c) $2 + 1 + 1$:
Two distributions are equal, and the other two distributions are different from these and each other, e.g. $F_1 = F_2 \neq F_3 \neq F_4, F_1 \neq F_4$.

d) $1 + 1 + 1 + 1$:
All distributions differ, $F_i \neq F_j, i \neq j \in \{1, \ldots, 4\}$.

These settings are considered for case 2 of the data-generating mechanisms. Settings a)–c) are mostly neglected previously (e.g. Mukherjee et al., 2022; Song and Chen, 2022b).

2.2.1 Numbers of Observations and Variables

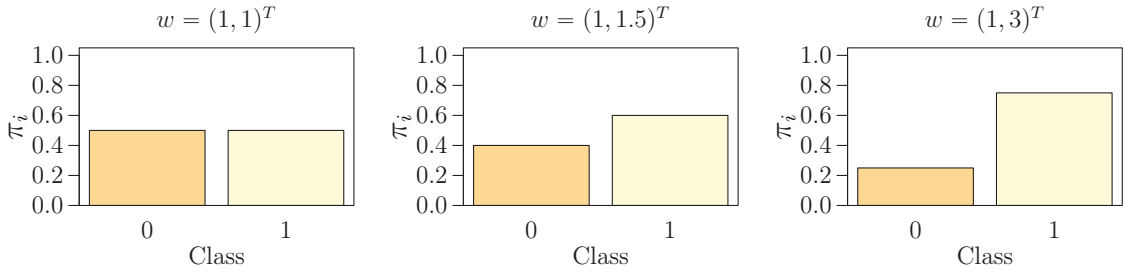
For each setting of the underlying distributions, different numbers of variables and observations are considered. Additionally, the imbalance of the number of observations of different datasets is also varied, as this might impact the method performance (Chen et al., 2018). The number of variables p is varied over $p \in \{2, 10, 50\}$, which represents low- to middle-dimensional data. The lower-dimensional data is chosen since not all methods are intended for high-dimensional data, and many previous studies only considered lower numbers of variables as well (Friedman and Rafsky, 1979; Schilling, 1986; Baringhaus and Franz, 2004; Székely and Rizzo, 2004; Rosenbaum, 2005; Baringhaus and Franz, 2010; Lopez-Paz and Oquab, 2017; Pan et al., 2018; Li et al., 2022). Moreover, the runtime of many methods increases both in p and in the sample size N such that high values are infeasible in the scope of a simulation study. For $k = 2$, the overall sample size is varied as $N \in \{50, 100, 200, 500, 1000\}$, and the individual sample sizes are set to $n_1 = \pi \cdot N$ and $n_2 = (1 - \pi) \cdot N$ with $\pi \in \{0.2, 0.5\}$ to cover typical sample sizes and one balanced and unbalanced sample size setting, respectively. For $k = 4$, the sample size is varied as $N \in \{100, 200, 400\}$ and the individual sample sizes are set to $n_1 = n_2 = n_3 = n_4 = 0.25 \cdot N$ or $n_1 = 0.1 \cdot N, n_2 = 0.2 \cdot N, n_3 = 0.3 \cdot N, n_4 = 0.4 \cdot N$. A full factorial design is used, i.e. all combinations of p, N, and the settings for the individual sample sizes are used in each scenario, and in the four-sample case also for each of the settings a)–d).

2.2.2 Generation of Categorical Data

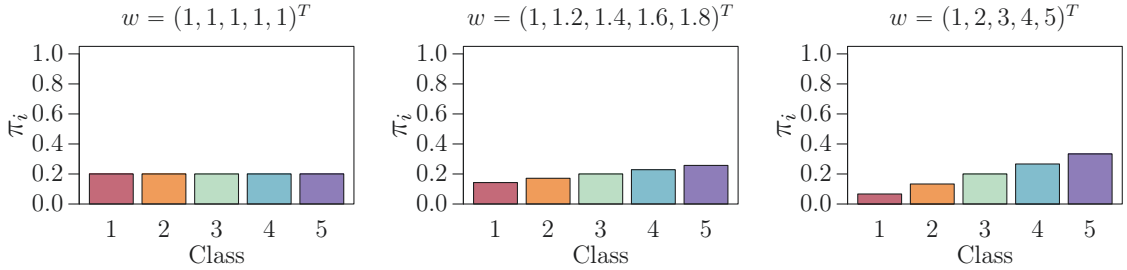
For categorical data, there are not many simulation studies comparing dataset similarity methods. Chen et al. (2013) compare their tests to the χ^2 test for $p = 1$ and use normal and uniform distributions discretized into twelve classes. For the alternatives, the second distribution is shifted, which changes the class distribution. This approach is inflexible with respect to the resulting deviations of the class distributions. Here, data are generated in a more flexible way using Bernoulli distributions and multinomial distributions with five classes, motivated by the common use of binary and 5-point Likert scale-like data, e.g. in questionnaire data in social sciences or for rating the severity of disease in medical applications. For most methods, a potential ordering of the categories is not taken into account. Some methods use the ordering indirectly by using the Euclidean distances of the observations. Only independent variables are considered in this study.

The probability distributions of each variable in the datasets are varied. Some selected probability distributions are visualized in Figure 1. For the null situation, all classes are equally likely. For binary data, as an alternative, the class probabilities are varied to gradually become more unbalanced. This is shown for two example scenarios in Figure 1a. The left barplot shows the balanced case, which is used in the null situation. The other barplots show unbalanced class distributions that are used as alternatives for one or more of the datasets. For the multinomial data, two types of alternatives are considered. First, the class probability distribution is varied in such a way that more probability mass is given to the higher classes. This is referred to as "skewed" for convenience, even though no strict ordering of the classes is assumed. Second, the class probability of one class is increased while the other is decreased by the same amount. The two cases are visualized for two example scenarios each in Figure 1b and Figure 1c, respectively. For the concrete class probabilities for all cases, see Tables 2 and 3 in Appendix A.

(a) Binary data.



(b) Multinomial data: Skewed class distribution.



(c) Multinomial data: One class up, one class down.

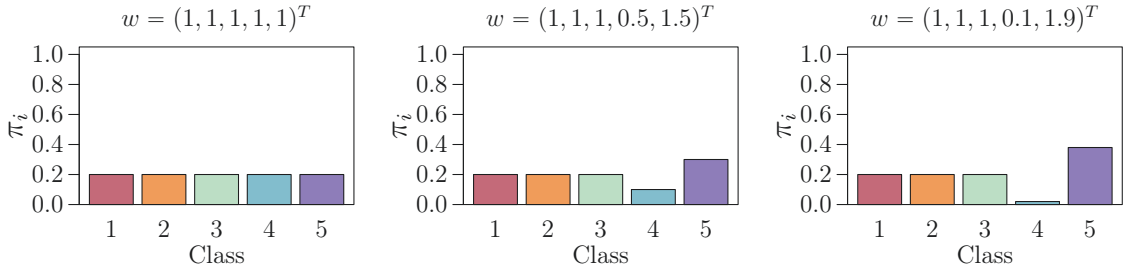


Figure 1: Class probability distribution for selected scenarios given by the weight vectors w. It holds $\pi_i = w_i / \sum_j w_j$.

2.3 Estimands

The population quantity of interest is the similarity or distance of the underlying distributions of the datasets. This is estimated by each method.

2.4 Methods

The most promising methods from the previous review and theoretical comparison (Stolte et al., 2024) are included in this empirical comparison. Methods are selected from the review if any of the following criteria are fulfilled:

1. The method is implemented in R.

2. The method fulfills at least 11 (i.e. more than half of the) criteria in the theoretical comparison, excluding the consistency criteria.

3. The method is the best in its subclass in the theoretical comparison, and no other method from this subclass was chosen with the first two criteria.

An overview of all methods that fulfill these criteria is provided in Table 4 in Appendix B. We restrict the analysis here to the subset of those methods that are applicable to categorical data without a target variable. These ten methods are explained in the following. Most of these methods are only applicable to two samples. For the methods that are applicable to multiple samples, this is explicitly stated. All methods are used with default parameters based on recommendations from the literature, if available. If no sensible default is available, different options are compared. The choices of these are explained in Appendix C. The methods are applied using parameter choices that a practitioner with good knowledge of the underlying literature but without expert knowledge of the methods could use.

- Classifier two-sample test (C2ST, Lopez-Paz and Oquab, 2017): The pooled dataset is split into a training and test set, and a classifier is trained on the training set to distinguish between the datasets. The classifier's accuracy on the test set is used as the statistic. For similar datasets, an accuracy close to the accuracy of the naive prediction of the larger dataset is expected, while for different datasets, higher accuracies are expected. A Binomial test can be used to compare the accuracy of the naive prediction. The procedure can also be used for multiple samples.

- Random forest-based test by Hediger et al. (2021) (HMN): A random forest is trained on the entire pooled dataset to distinguish between the individual datasets. The out-of-bag prediction error is used as a test statistic. If the datasets are similar, the error should be close to that expected for always predicting the larger dataset.

- Tree-based test by Yu et al. (2007) (YMRZL): A classification tree is trained to distinguish between the datasets using a training dataset that is a subset of the pooled sample. Its classification error on the left-out test set is used as the statistic. If the datasets are similar, the error should be close to that expected for always predicting the larger sample.

- Original edge count test by Friedman and Rafsky (1979) (FR): A graph (originally the minimum spanning tree, MST) is constructed on the pooled sample using an appropriate distance measure. Here, the Hamming distance is used. For categorical data, the optimal graph might not be unique due to ties in the inter-point distances. Therefore, either the arithmetic mean of the test statistics on all optimal graphs ("a") or the test statistic on the union of all optimal graphs ("u"), i.e. the graph that includes all edges of all optimal graphs, is calculated (Chen et al., 2013). For calculating the test statistic for a given similarity graph, the number of edges connecting points from different samples is counted. The expectation and variance of this edge-count statistic under the null hypothesis of equal distributions are known and can be calculated analytically. The standardized edge count using this null expectation and standard deviation is used as the test statistic. It follows a standard normal distribution asymptotically under the null. Certain assumptions

on the size and density of the similarity graph have to be made to prove the asymptotic normality of the test statistic. For similar datasets, higher numbers of edges connecting points from different samples are expected.

- Generalized edge count test by Chen and Friedman (2017) (CF): The Friedman-Rafsky test is generalized to improve the power for detecting both location and scale alternatives. The number of edges connecting points within each of the two samples, R_1, R_2, respectively, is counted in a similarity graph on the pooled sample. The Mahalanobis distance

$$(R_1 - \mathbb{E}_{H_0}(R_1), R_2 - \mathbb{E}_{H_0}(R_2))\, \mathbb{C}\mathrm{ov}_{H_0}^{-1}(R) \begin{pmatrix} R_1 - \mathbb{E}_{H_0}(R_1) \\ R_2 - \mathbb{E}_{H_0}(R_2) \end{pmatrix}$$

  of the vector $R = (R_1, R_2)^T$ is used as the test statistic. Under the null, it is asymptotically χ^2_2-distributed under certain assumptions on the graph. Small values of the statistic indicate the similarity of the datasets. Again, for categorical data, averaging ("a") or the union ("u") can be used (Zhang and Chen, 2022).

- Weighted edge count test by Chen et al. (2018) (CCS): The Friedman-Rafsky test is generalized to improve the power in settings with unequal sample sizes. The weighted statistic is defined as

$$R_w = \frac{n_1}{N} R_1 + \frac{n_2}{N} R_2,$$

  where R_1, R_2 are defined as above and n_i denotes the sample size of the i-th sample, $i = 1, 2$, $N = n_1 + n_2$. Again, the expectation and standard deviation of R_w can be calculated analytically and are used to define a standardized test statistic that is asymptotically standard normally distributed under the null given certain assumptions on the graph. Small numbers of edges connecting points within the same sample indicate similar datasets. Therefore, small values of R_w or its standardized version indicate similarity. Again, for categorical data, averaging ("a") or the union ("u") can be used (Zhang and Chen, 2022).

- Max-type edge count test by Zhang and Chen (2022) (ZC): The test is yet another generalization of the Friedman-Rafsky test. The test statistic is given by

$$R_m = \max\{\kappa R_w, |R_1 - R_2|\},$$

  where κ is a parameter that has to be chosen prior to testing. Again, a standardized version is given by standardizing R_w and $R_d = |R_1 - R_2|$ with their expectations and standard deviations under the null, with both components asymptotically following a standard normal distribution under the null, given certain assumptions on the graph. Small values of the statistic indicate similarity. Again, for categorical data, averaging ("a") or the union ("u") can be used (Zhang and Chen, 2022).

- Multi-sample Cross-Match statistic by Petrie (2016): The optimal non-bipartite matching is calculated on the pooled sample, and the overall number of edges connecting points from different samples is calculated. It is standardized by the analytical expectation and standard deviation under the null hypothesis. The standardized statistic is asymptotically standard normally distributed under the null. High values of the cross-match statistic indicate similarity between the datasets.

- Multi-sample Mahalanobis Cross-Match (MMCM) statistic (Mukherjee et al., 2022): The optimal non-bipartite matching is calculated on the pooled sample. The numbers of edges a_{ij} connecting points from sample i and sample j, $i \neq j \in \{1, \ldots, k\}$, is calculated. The Mahalanobis distance of the cross-match vector $A = a_{12}$ in the two-sample case and $A = (a_{12}, a_{13}, a_{23}, a_{24})^T$ in the four-sample case, respectively, is used as the test statistic

$$\text{MMCM} = (A - \mathbb{E}_{H_0}(A))^T \, \mathbb{C}\text{ov}_{H_0}^{-1}(A)(A - \mathbb{E}_{H_0}(A)),$$

  where again expectations and covariances under the null can be calculated analytically. The MMCM statistic follows a χ^2_{k-1}-distribution asymptotically under the null. For similar datasets, low MMCM values are expected. For two samples, this is analytically equivalent to Petrie's test.

- Constrained Minimum (CM) Distance (Tatti, 2007): The CM distance is based on a *feature function* $S : \mathcal{X} \to \mathbb{R}^m$ that maps points from the sample space $\mathcal{X}$ to a real vector. The *frequency* $\theta \in \mathbb{R}^m$ of S with respect to dataset $X^{(j)}$ is the average of the values of S

$$\theta_j = \frac{1}{N} \sum_{i=1}^{n_j} S(X_i^{(j)}), j = 1, 2.$$

  The CM distance is then defined as

$$D_{\text{CM}}(X^{(1)}, X^{(2)}|S)^2 = (\theta_1 - \theta_2)^T \, \mathbb{C}\text{ov}^{-1}(S)(\theta_1 - \theta_2),$$

  with

$$\mathbb{C}\text{ov}(S) = \frac{1}{|\mathcal{X}|} \sum_{\omega \in \mathcal{X}} S(\omega) S(\omega)^T - \left(\frac{1}{|\mathcal{X}|} \sum_{\omega \in \mathcal{X}} S(\omega) \right) \left(\frac{1}{|\mathcal{X}|} \sum_{\omega \in \mathcal{X}} S(\omega) \right)^T.$$

  The recommended feature function S is used here, i.e. the independent means of the variables are considered (see Appendix C).

All of these methods are applied in the two-sample case. The C2ST, MMCM, and the method of Petrie (2016) are also applied in the multi-sample case. For the other methods, no generalizations to the multi-sample case are available in the literature. For all methods except for the CM distance, a permutation test is proposed in addition. However, no permutation test is performed here due to the high runtime. More details on the methods can be found in Stolte et al. (2024) and the references therein.

2.5 Performance Measures

Two aspects are evaluated here. First, it is evaluated how well the methods can detect the differences between the distributions that were described in the previous subsections. No classical power comparison can be conducted since not all methods define a test. Moreover, such a comparison would not be possible due to the very high runtimes of the many permutation tests. Instead, the methods are compared as follows. As the ranges of the method values vary heavily from method to method, the values cannot be compared directly. Therefore, the performance of the methods has to be made comparable. The approach here is based on the observation that a typical power comparison evaluates the proportion of simulation repetitions in which the observed test statistic is more extreme

than some quantile of the (permutation) null distribution. The approach is illustrated in Figure 2. For each setting in which the distributions do not differ, a quantile of the simulated statistic values for a certain method is calculated. The statistic values of that method for deviations of one (or more) distribution(s) from the first are then compared to this quantile. The proportion of simulation repetitions in which a more extreme statistic value than this threshold is observed is used to quantify the performance of the method. For methods for which high values correspond to the similarity of the distributions, the proportion of simulation repetitions is used for which the resulting statistic value is smaller than the 5% quantile of the corresponding statistic values simulated under equal distributions. For methods for which low values correspond to similarity, the proportion of repetitions with values higher than the 95% quantile is used. We abbreviate this *Proportion of Extreme Simulation Repetitions* w.r.t. the null threshold as *PESR* in the following. It can be used to evaluate how well the methods detect the difference in the distributions, and it can be compared between different methods. Note that the determined PESR does not directly equal the testing power since only one specific null situation is considered. Moreover, the PESR might still show high values in cases where the asymptotic test does not find any differences or in cases where the asymptotic test does not hold its α level. This can happen when the asymptotic null distribution does not match the empirical distribution. For methods for which an asymptotic test exists, the asymptotic testing power can be simulated without relevant increases in runtime. See Appendix D for a comparison and discussion of simulated asymptotic power and the PESR for selected methods.

The number of simulation repetitions is set to 500 repetitions per scenario. The Monte Carlo standard error (MCSE) for proportions like the PESR is highest for a proportion of 0.5. For 500 iterations, the MCSE is then $\sqrt{0.5(1-0.5)/500} \approx 0.022$. For a proportion of 0.05, which corresponds to the PESR in null situations, the MCSE is $\sqrt{0.05(1-0.05)/500} \approx 0.01$. This is considered sufficiently small here. Using 1000 iter-

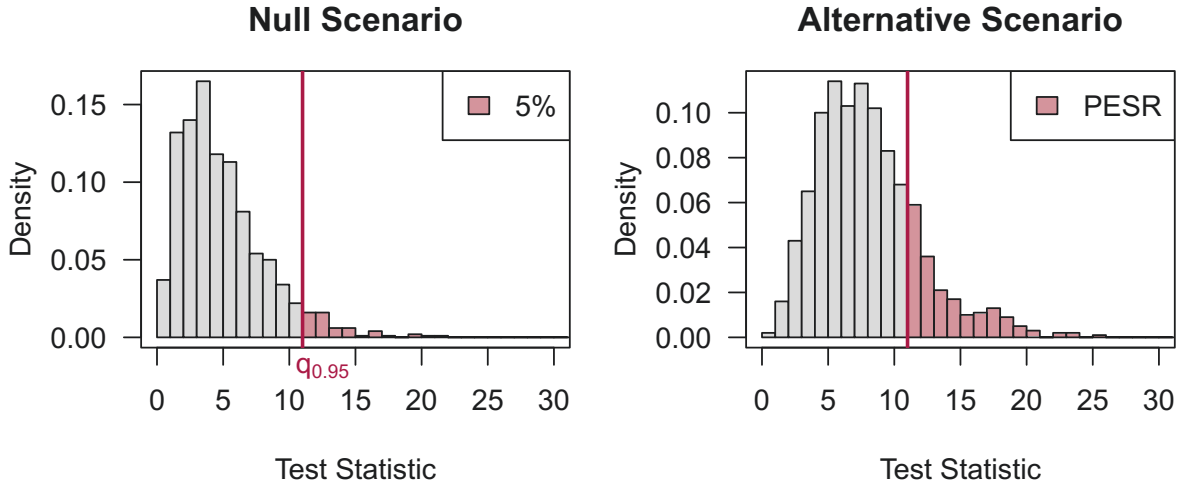


Figure 2: Illustration of PESR calculation for the case where low test statistic values correspond to similarity between the datasets. The 95% quantile $q_{0.95}$ of the test statistic values simulated under a null scenario is used as the threshold under the alternative scenario. The proportion of values that are more extreme than this threshold is evaluated and denoted as the PESR. Test statistic values for this figure were artificially generated from a χ^2_5 and χ^2_8 distribution, respectively, for demonstration.

ations would only bring down the MCSEs to ≈ 0.015 and ≈ 0.006, respectively, but this would double the runtime.

In case of computational errors resulting in missing or infinite values of the statistics, the affected repetitions are excluded from the PESR calculation. If there are missing values in more than 100 of the 500 iterations, the corresponding PESR value is set to missing. This ensures that the calculated PESR values are based on a reasonably high number of repetitions.

In addition to the PESR, we consider the applicability of the methods in practice. To do this, runtime, memory consumption, and any numerical problems are considered. These are measured for selected scenarios only. Here, the null situations for two balanced classes are chosen. All combinations of N and p as discussed before are used except for $N = 1000$ for the two-sample case, since it is infeasible with the RAM configuration of the used computer. Datasets are generated once for each combination of N and p for that scenario. On each dataset, each similarity method is then applied once to measure the memory consumption and afterward at least 10 times to measure the runtime per method call. For methods with low runtimes, the number of repetitions is increased such that the method is run for at least 1 second to get stable estimates of the runtime per method call. Each method is called once before starting the benchmark to ensure that all required packages and objects are already loaded at the start of the benchmark and the results are not distorted by lazy loading. The benchmarks are performed on a Lenovo ThinkPad laptop with an AMD Ryzen 5 PRO 4650U processor with six cores and 16 GB of RAM under Windows 10. Benchmarks are run during the nighttime when the laptop is not used for any other work to ensure that the results are not disturbed by other computations.

2.6 Software

All simulations are performed using `R` version 4.4.0 (R Core Team, 2024) on the Linux-HPC-Cluster (LiDO3) at TU Dortmund University. Further analyses and benchmarking are performed using `R` version 4.4.2 (R Core Team, 2024) on a personal computer. The implementation of all methods can be found in the `DataSimilarity` package (Stolte and Sauer, 2025). The `bench` package (Hester and Vaughan, 2025) is used for measuring runtime and memory consumption. The `pheatmap` (Kolde, 2019) and the `cba` (Buchta and Hahsler, 2024) packages are used for visualizing and clustering the PESR values of the methods across scenarios. The `rpart.plot` (Milborrow, 2016) package is used for visualizing the decision rules for finding the best-performing methods for a specific scenario. The full `R` code of the study can be found on Zenodo (Stolte, 2025).

3 Sensitivity in Detecting Differences Between Datasets

In the following, the proportions of extreme simulation repetitions (PESR) are compared between the methods, first for the two-sample and then for the multi-sample setting.

3.1 Two-sample Setting

In the following, the results for the two-sample setting ($k = 2$) are discussed. Since in any scenario for any method, either all repetitions encountered an error or a maximum of 8 out of the 500 repetitions encountered an error, the results are either missing or the effect of errors can be seen as negligible. For a more detailed discussion of the occurring errors, see Section 5.3. First, a pre-selection of methods is performed to exclude variants that are inferior in all scenarios from the following comparisons for clarity. Next, the results of the selected methods are discussed for datasets consisting of binary data. Afterward, the results for datasets consisting of categorical data with five categories are discussed. In the former case, the success probability in the second dataset is varied. In the latter case, two alternatives for varying the probabilities of the five classes are considered: a skewed probability distribution and increasing the probability of one class while decreasing the probability of another class (see Figure 1 for an illustration). Last, the best-performing methods are summarized over all scenarios.

3.1.1 Pre-Selection of Methods

Counting all variants, a total of 56 methods are applied in the two-sample setting. In the following, the best variants of similar methods are pre-selected to facilitate the comparison. For that, the variants are compared, and variants that are inferior to others in all considered scenarios are eliminated from the following analyses.

The first group is the classifier two-sample tests. The C2ST itself is used with two classifiers, the multilayer perceptron (NN) and the K-nearest neighbor classifier (KNN). Moreover, the YMRZL method of Yu et al. (2007) can also be seen as a variant of the C2ST that uses a decision tree as the classifier. When comparing these three methods, the YMRZL is worse than the C2ST (NN) and the C2ST (KNN) in almost all cases. Which of the C2ST (NN) and the C2ST (KNN) performs better depends on the scenario. The comparison of the three methods for binary data and balanced sample sizes is shown as an example in Figure 38 in Appendix F.1. In the following analyses, the results for the YMRZL method will not be shown.

For each of the edge-count tests by Friedman and Rafsky (1979), Chen and Friedman (2017), Chen et al. (2018), and Zhang and Chen (2022), the graph-type is varied as a K-minimum spanning tree (KMST), or a K-nearest neighbor-graph (KNN) with $K = 1$ and 5. Moreover, for each graph, averaging ("a") or the union ("u") is applied to handle ties. Therefore, there are in total six variants for each of the methods: FR, CF, and CCS. The ZC method has an additional parameter κ that is varied over the three recommended values $\kappa = 1, 1.14, 1.31$. Thus, there are in total 18 variants for ZC. The PESR comparisons between the different graphs and averaging and union for each method are shown for binary datasets with balanced sample sizes as an example in Figures 39 to 42 in Appendix F.1. The comparisons can be summarized over all scenarios as follows. For $p = 2$, the 5NN, "u" versions fail, resulting in either only `NaN` (not a number) or only 0

test statistic values due to standardization with variances that are analytically equal to zero. The same holds for the CF and ZC 1NN, "a". These issues are further discussed in Appendix E. For binary data, the differences between the remaining methods are very small. For multinomial data, the differences are clearer, and typically, the 1NN, "u" version performs best. For $p = 10$, the $K = 5$ graphs typically perform better than the $K = 1$ graphs. Typically, either the 5MST, "u" is best, or it is only best for small N, and the 5NN, "u" is better (or very similar to the 5MST) for large N. For $p = 50$, there are no differences between the "a" and "u" versions. One reason for this might be that with so many variables, there are fewer tied observations and therefore fewer or even only one optimal graph, such that the differences between averaging over the optimal graphs or using their union diminish. The superiority of the $K = 5$ versions over the $K = 1$ versions is even clearer for $p = 50$ than for $p = 10$. Often, the 5NN performs best. Sometimes, the 5MST performs better for small N. Generally speaking, the 5MST performs best for CF and ZC in general and FR in the case of unbalanced data. So overall, the $K = 1$ methods lose PESR with increasing p, while the $K = 5$ methods gain PESR with increasing p. "u" tends to work better than "a" except for the $p = 1$ and 5NN case, but in many cases the differences between "a" and "u" are small or even negligible. The differences are clearer for unbalanced sample sizes where the "a" variants are slightly more affected than the "u" variants. Regarding the choice of κ for ZC, the differences between the PESR values for a given graph and "a"/"u" combination are negligible (see e.g. Figure 45 in Appendix F.1). Therefore, for the remaining analysis, the methods are restricted to three versions: the 1NN, "u" performs best for $p = 2$, the 5MST, "u" performs best in many cases for $p > 2$, and the 5NN, "a" is considered since 5NN sometimes outperforms 5MST for $p > 2$ and 5NN, "u" fails for $p = 2$. For ZC, only the default $\kappa = 1.14$ is used.

For the random forest-based test HMN by Hediger et al. (2021), there are two options that are compared. The classification error can be calculated per class or overall. There are often no differences with regard to PESR between these variants (see e.g. Figure 46 in Appendix F.1), but if there are differences, the per-class OOB version shows a higher PESR. Therefore, only that version is used in the following.

The method of Petrie (2016) and the MMCM (Mukherjee et al., 2022) are based on the same graph-based quantities and can, therefore, be seen as variants of each other. In many cases, there are no differences between the two, but if there are, Petrie's method typically performs better (see e.g. Figure 47 in Appendix F.1). Both methods mostly fail for $p = 2$, which is discussed in Section E. Thus, the MMCM is not shown in the following.

So, in total, 17 methods and variants are considered: FR, CF, CCS, and ZC ($\kappa = 1.14$) each with the three versions 1NN, "u", 5MST, "u", and 5NN, "a", C2ST (KNN) and C2ST (NN), HMN (per class OOB), Petrie, and the CM distance, for which no variants were tested. For balanced sample sizes, CF and FR are equivalent, such that the methods can be reduced further to 14 methods, and then only FR is shown.

3.1.2 Binary Data

Figure 3 shows the proportions of extreme simulation repetitions (PESR) for two binary datasets of equal sample sizes for each of the pre-selected methods. The Monte Carlo standard errors (MCSEs) are also displayed by error bars. The MCSEs are small here and in all following analyses, indicating low sampling variability of the simulation results. For $p = 2$, many methods perform similarly well. In particular, the CM distance and the edge count tests for 1NN, "u" perform comparably well. The HMN, C2ST, and Petrie's method perform considerably worse than the rest. For $p > 2$, the CM distance performs

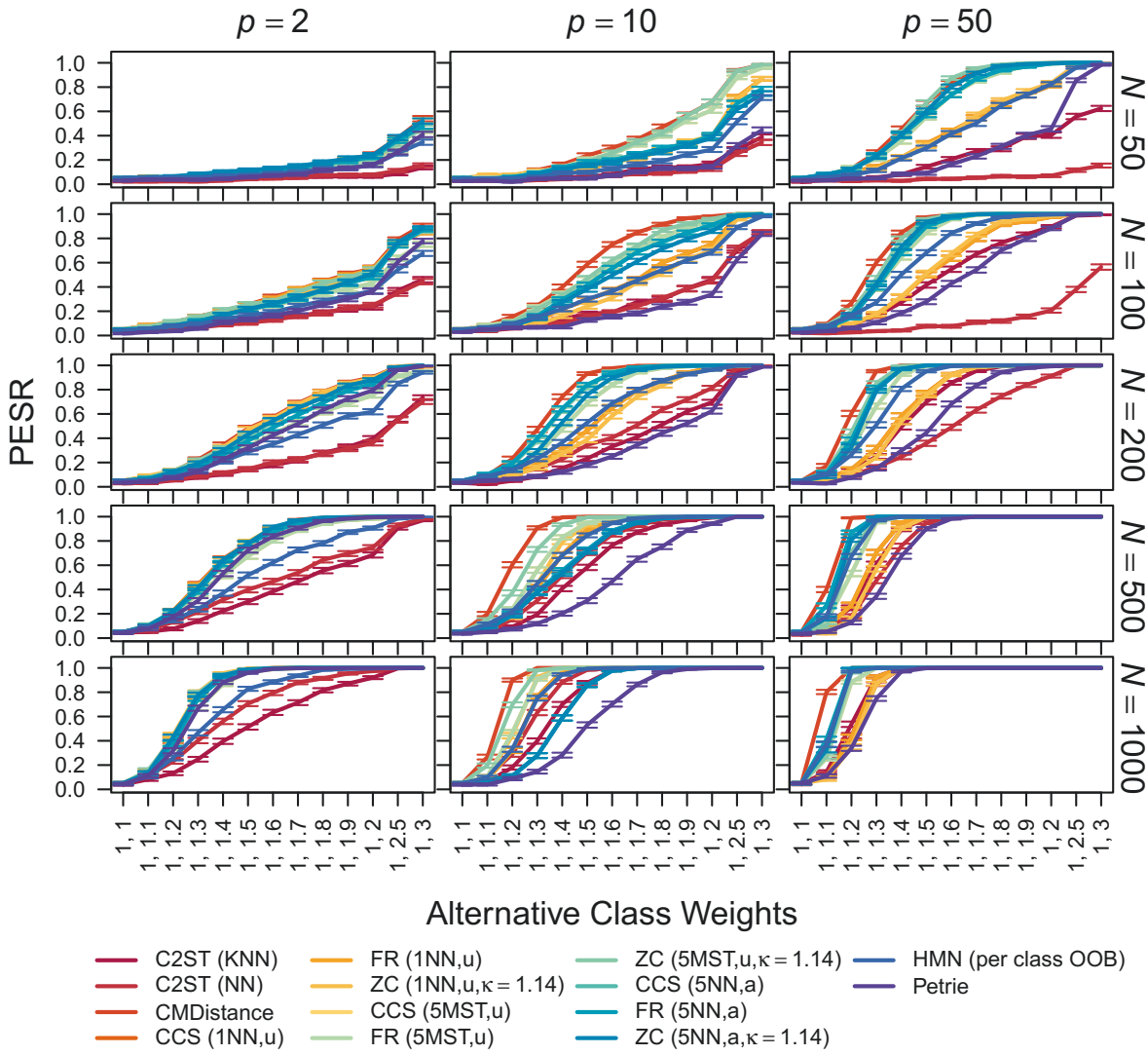


Figure 3: PESR (proportion of extreme simulation repetitions) for $k = 2$ binary datasets of equal sample sizes. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

best. The edge count tests for the $K = 5$ graphs also perform well. For $p = 10$, the ZC (5MST, u) is the second best method, for $p = 50$, it is the CCS (5NN, a). The classifier-based tests HMN and C2ST, as well as Petrie's method, perform worse. As expected, for all methods, the PESR increases with an increasing number of observations.

For unbalanced sample sizes, the PESR of all methods decreases. This is, for example, illustrated in Figure 4 for $p = 50$. The full results for the unbalanced case are shown in Figure 48 in Appendix F.2. The performance of the CM distance and the classifier-based tests is more heavily impacted than that of the graph-based tests. Therefore, the CM distance is no longer the best-performing method, but typically, CF (5MST, u) becomes the best method. The method CF, which was specifically designed as an alternative to FR for unbalanced sample sizes, is less affected than that of FR; see e.g. Figure 4. The HMN method breaks down completely for unbalanced sample sizes.

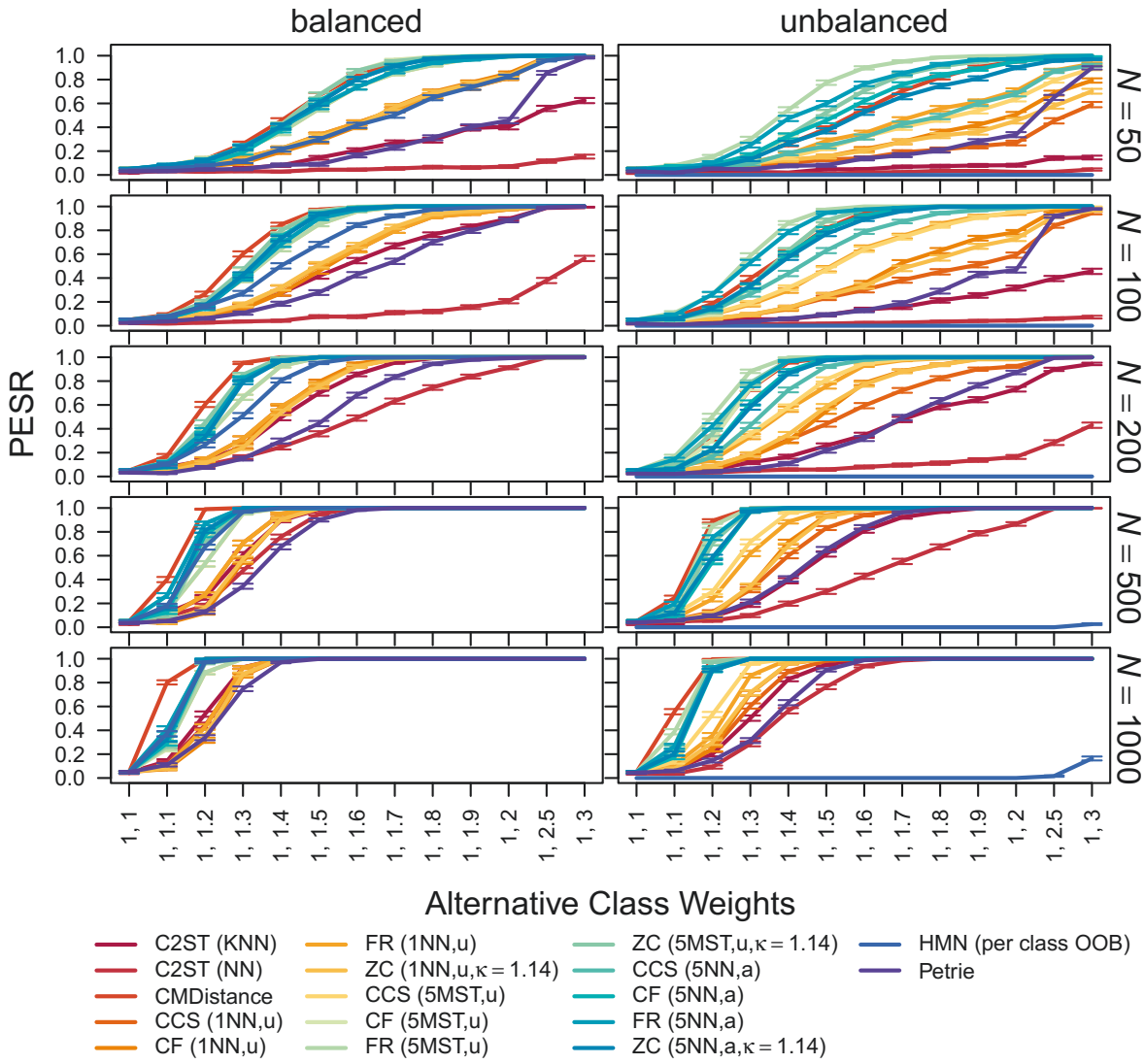


Figure 4: PESR (proportion of extreme simulation repetitions) for $k = 2$ binary datasets of equal ("balanced") and unequal ("unbalanced") sample sizes with $p = 50$ variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

3.1.3 Multinomial Data

Figure 5 shows the proportions of extreme simulation repetitions (PESR) for two datasets of equal sample sizes with categorical variables. The probability distribution for the five classes in the second dataset gets more and more skewed (from left to right in each single plot), while the probability distribution for the five classes in the first dataset is a uniform distribution. For $p = 2$ and $p = 10$, the CM distance is again the best-performing method. However, for $p = 50$, it cannot be computed anymore as it requires the enumeration of the whole sample space, which becomes numerically infeasible for the $5^{50} \approx 8.9 \cdot 10^{34}$ possible values due to memory restrictions. For $p = 2$, the Petrie statistic fails, and the estimated PESR is very high, irrespective of the differences in probability distributions between the datasets. The edge count tests FR, CCS, and ZC (1NN, u) perform quite well. The classifier-based tests HMN, C2ST (NN), and C2ST (KNN) perform poorly in

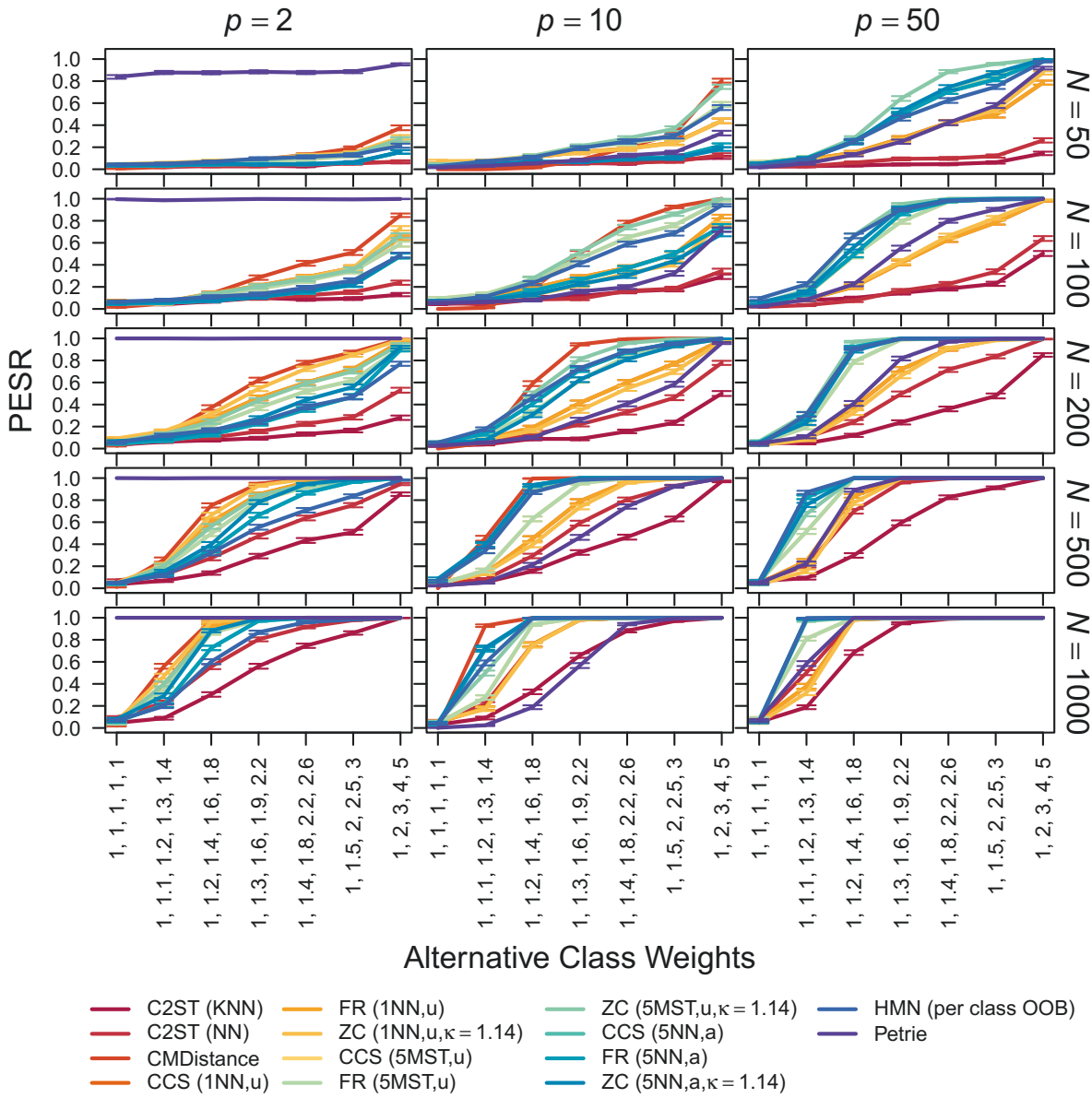


Figure 5: PESR (proportion of extreme simulation repetitions) for $k = 2$ datasets of equal sample sizes with categorical variables. The class weights give the unnormalized probabilities $(1, 1 + \delta, 1 + 2\delta, 1 + 3\delta, 1 + 4\delta)$ for the values 1 to 5 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1, 1, 1, 1)$. Error bars indicate Monte Carlo standard errors.

the comparison. For $p = 10$, the results are similar, but Petrie's method is working as intended. Its PESR values are, however, comparably low. For $p = 50$, the graph-based tests with $K = 5$ and the HMN work best, followed by the $K = 1$ versions and Petrie's method. The C2ST versions, especially the C2ST (KNN), perform the worst.

For unbalanced sample sizes, the PESR decreases again for all methods (see Figure 49 in Appendix F.2). As for binary data, the C2ST and the CM distance are more affected than the edge count tests. The HMN fails again in the case of unbalanced sample sizes. The FR performs best (1NN, "u" for $p = 2$ and 5MST, "u" for $p > 2$). Here, the CF is no improvement for the FR performance.

Figure 6 shows proportions of extreme simulation repetitions (PESR) for two datasets

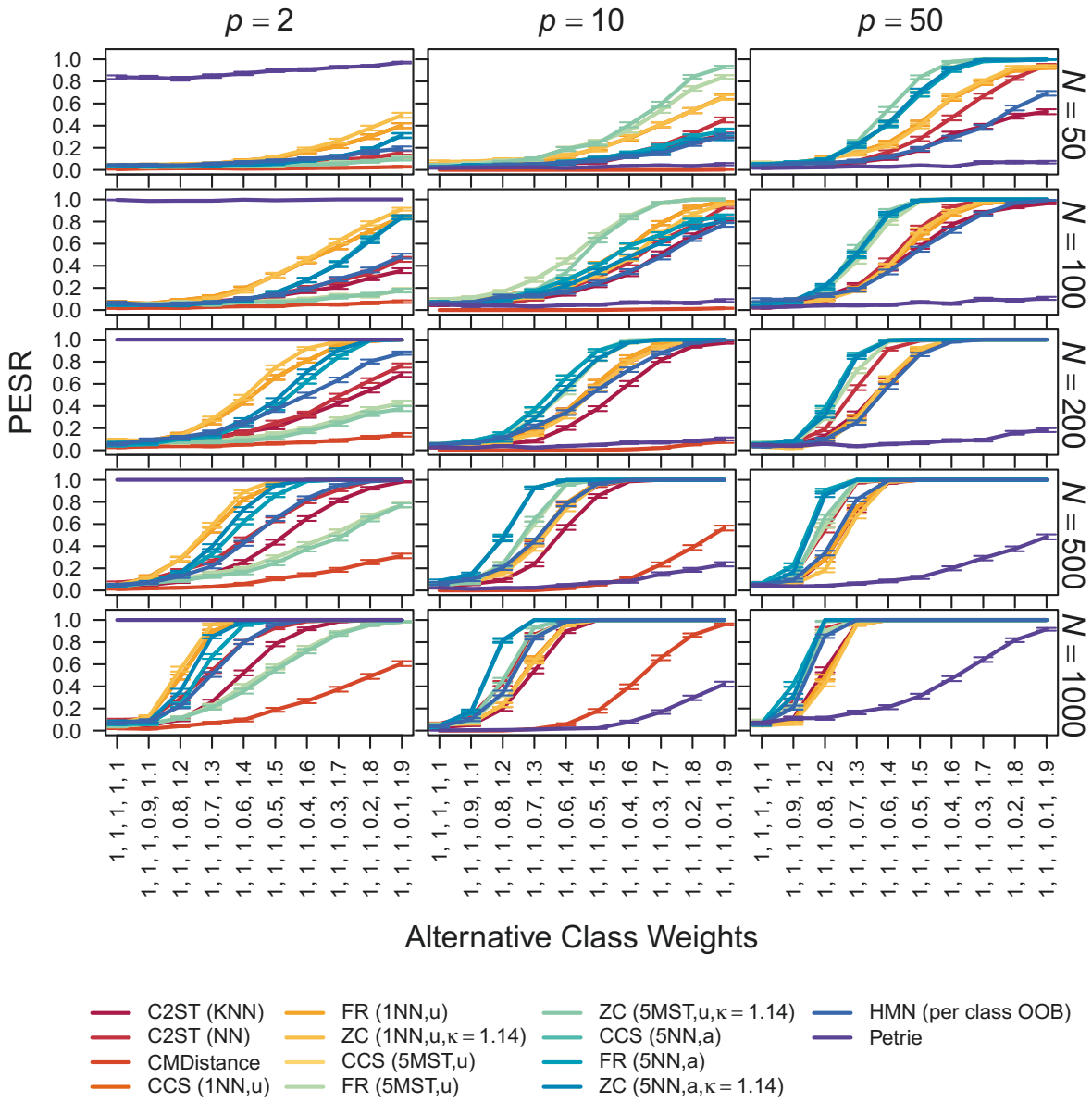


Figure 6: PESR (proportion of extreme simulation repetitions) for $k = 2$ datasets of equal sample sizes with categorical variables. The class weights give the unnormalized probabilities $(1, 1, 1, 1+\delta, 1-\delta)$ for the values 1 to 5 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1, 1, 1, 1)$. Error bars indicate Monte Carlo standard errors.

of equal sample sizes with categorical variables when the probability of one class increases while the probability of another class decreases in the second dataset. Contrary to the results for the other settings, here, the CM distance is not the best-performing method but almost the worst. For $p = 50$, it is again infeasible to compute. For $p = 2$, the edge count tests using 1NN, "u" are best, with the ZC slightly outperforming the others. Petrie's method again fails. The CM distance achieves the lowest PESR values. For $p = 10, 50$, and small N, the edge count tests using 5MST, "u" perform best. For larger N, their performances are exceeded by that for the 5NN, "a". The method by Petrie (2016) and the CM distance (for $p = 10$) perform the worst. The C2ST methods perform comparatively better than for the skewed probability distribution alternative.

For unbalanced sample sizes, the performances of all methods decrease again (see Figure 50 in Appendix F.2). The classifier-based methods especially have very low performance in that case. The FR (1NN, u for $p = 2$, 5MST, u for $p > 2$) performs best, followed by the CF and ZC.

3.1.4 Summary of Best-performing Methods

Figure 7 shows a decision tree in which the decision rules for determining the best method are given. These rules depend on the type of alternative (binary, multinomial "skewed", multinomial "1 up, 1 down"), the balance of the sample sizes, the number of variables p, and the number of observations N in the pooled sample. Depending on these, the best-performing methods in terms of the highest PESR curve are given. For the binary and "skewed" case and balanced sample sizes, the CM distance is the best choice, except for $p = 50$ for which it cannot be calculated. In the latter case, the ZC (5MST, u) is best for low sample sizes and the HMN (per class OOB) for large sample sizes. For the "1 up, 1 down" alternative and balanced sample sizes, different edge count tests using dense graphs are best depending on the dataset dimensions (N and p). For unbalanced sample sizes and $p = 2$, the FR (1NN, u) performs best. For larger p, the FR (5MST, u) performs best, except for binary data, $p = 50$, and large N, where the CM distance is better.

Since the true deviations between the datasets are unknown in practice, a method for quantifying the similarity or distance of the datasets has to be chosen independently of that. Therefore, the question arises of how much worse a fixed method would perform than the ideal method choice in each scenario. Therefore, for each alternative scenario, the difference of the PESR value of each method from the highest PESR value across all methods for that scenario is calculated. The null scenarios are not considered for this comparison since, for these, the PESR should be at 0.05 for all methods per definition.

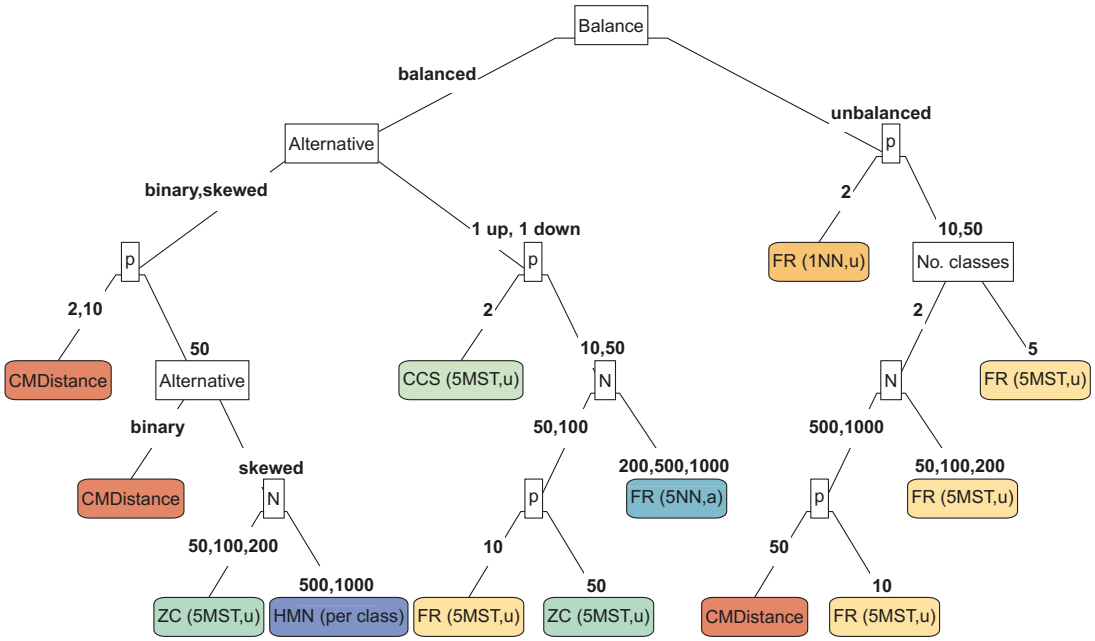


Figure 7: Summary of best-performing method per scenario for the two-sample case.

The median of the differences is calculated for each combination of the sample size balance, the number of categories, and p, i.e. the differences are aggregated over the different types of deviations and N. The aggregation over the deviations is performed since these would be unknown in real-world applications. The aggregation over N is performed for clarity since the best method rarely differs depending on N. Missing PESR values are assigned the maximum difference of 1 to penalize errors of the method. The same holds for PESR values that are constantly too high. These are discarded from calculating the maximum, and the corresponding method is assigned a difference of 1. This is the case for Petrie's test for $p = 2$ and multinomial data, see e.g. Figure 6.

Figure 8 shows the median difference of the PESR values per method to the highest PESR value, for each combination of the sample size balance, the number of categories and p, as well as the overall median differences of the PESR values to the scenario-specific best method (see last column). It can be seen that overall, the FR (5MST, u) has the lowest median difference and is, therefore, typically not much worse than the best-performing method. It is closely followed by the CF (5MST, u) and the ZC (5MST, u), and FR (1NN, u). Again, it can be seen that the edge count methods perform well overall, followed by the CM Distance, the classifier-based tests, and last, Petrie's test.

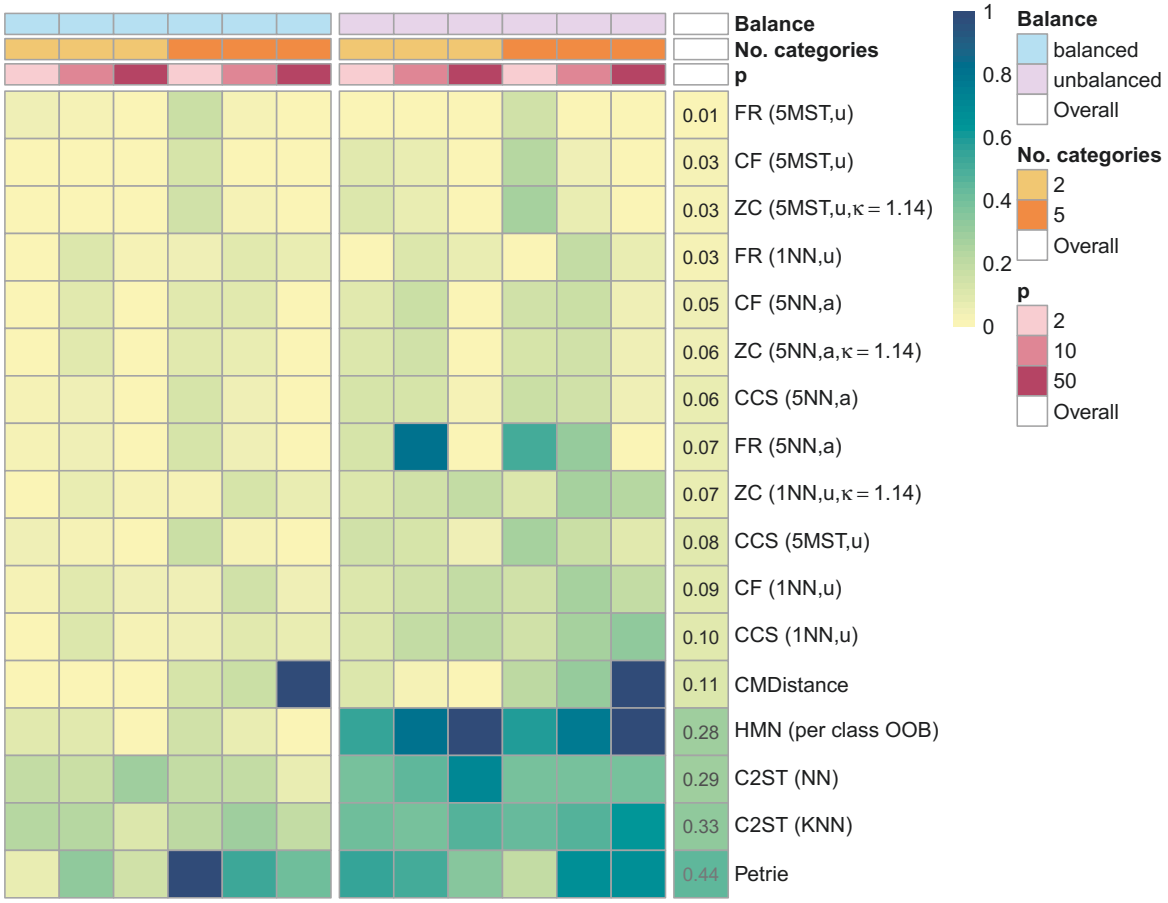


Figure 8: Median difference of the PESR values to that of the best-performing method per scenario for the two-sample case.

3.2 Multi-sample Setting

In the following, the results for the multi-sample setting ($k = 4$) are discussed. No errors occurred during the simulations. First, the results for datasets consisting of binary data are discussed. Afterward, the results for datasets consisting of categorical data with five categories are discussed. Last, the best-performing methods are summarized.

3.2.1 Binary Data

Figure 9 shows the PESR (proportion of extreme simulation repetitions) for binary data and the "1+1+1+1" case with balanced sample sizes. Similar results can be observed for other groupings except where explicitly mentioned below (see Figures 51 to 53 in Appendix F.3). In general, the proportions increase with increasing differences in the class weights of the datasets. Moreover, the proportions increase with the overall sample size N. The proportion also increases with the number of variables p, except for the method C2ST (NN), for which the proportions are highest for $p = 10$. The alternatives here are chosen to affect all variables in the datasets such that the true differences between the datasets increase with increasing dimension, which might explain the generally increased proportions for higher-dimensional data. For C2ST (NN) using the multilayer perceptron as the classifier, this effect of the larger true differences might be canceled out by the decreasing performance of the classifier for higher-dimensional data.

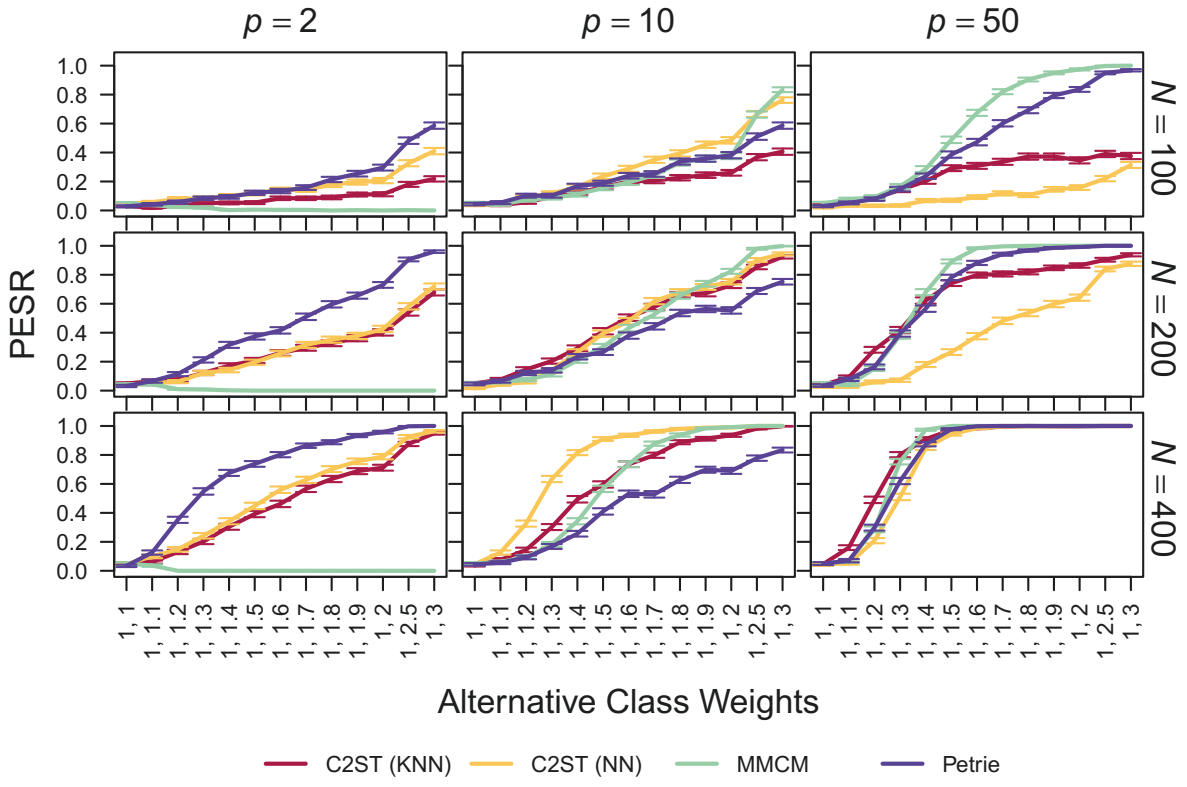


Figure 9: PESR (proportion of extreme simulation repetitions) for $k = 4$ binary datasets of equal sample sizes. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, in the second dataset to $(1, 1 + \delta)$, in the third to $(1, 1 + 2\delta)$, and in the fourth to $(1, 1 + 3\delta)$. Error bars indicate Monte Carlo standard errors.

For $p = 2$, MMCM fails, the method of Petrie (2016) outperforms the other methods, and C2ST (NN) either outperforms C2ST (KNN) or these two are comparable. This also holds for the other groupings, except for the "3+1" case where none of the methods performs well. For $p = 10$ and $N < 400$, the methods perform similarly for small to medium deviations. For large deviations, clearer differences are visible, and C2ST (NN) or MMCM outperforms the other methods. For $p = 10$ and $N = 400$, C2ST (NN) performs better than other methods, the method of Petrie (2016) often performs worst, and C2ST (KNN) performs similarly to MMCM, with the latter performing slightly worse for small deviations and slightly better for large deviations. For $p = 50$, MMCM outperforms the other methods for the small value $N = 100$. For larger N, C2ST (KNN) performs better, and the method of Petrie (2016) is equal to MMCM for small deviations. For $N = 100$, C2ST (NN) and C2ST (KNN) perform poorly compared to the other methods. C2ST (NN) still performs poorly in the comparison for $N = 200$ while C2ST (KNN) is competitive there for small and medium deviations. For other groupings with fewer numbers of differing datasets, MMCM (and Petrie's method) tends to perform worse than the C2ST (KNN), especially for small deviations and large N. This can be seen in Figure 10, where the PESR curves are compared for the different groupings for $p = 50$.

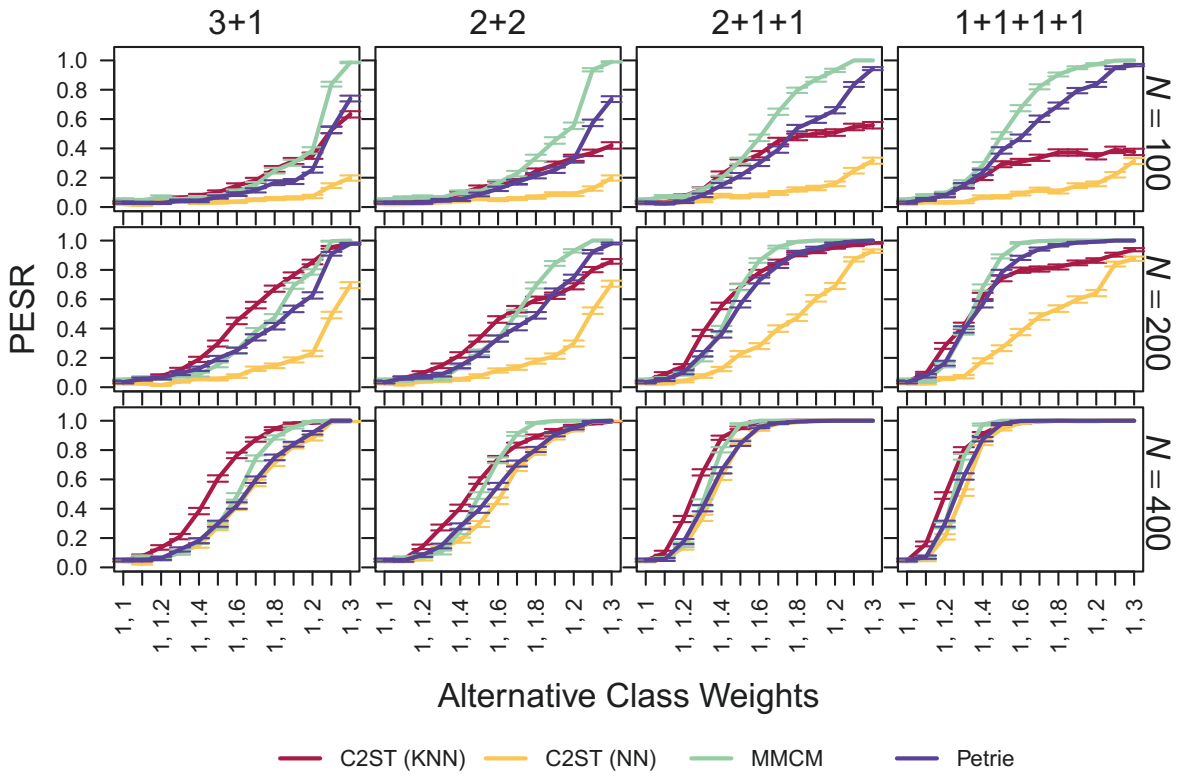


Figure 10: PESR (proportion of extreme simulation repetitions) for $k = 4$ binary datasets of equal sample sizes with $p = 50$ variables. The class weights give the unnormalized probabilities $(1, 1+\delta)$. The weights in the first dataset are always set to $(1, 1)$. For "3+1", the weights in the second and third datasets are $(1, 1)$, and in the fourth dataset $(1, 1+\delta)$. For "2+2", the weights in the third dataset are also $(1, 1+\delta)$. For "2+1+1", the weights in the third dataset are $(1, 1+\delta)$, and in the fourth $(1, 1+2\delta)$. For "1+1+1+1", the weights are $(1, 1+\delta)$ in the second dataset, $(1, 1+2\delta)$ in the third, and $(1, 1+3\delta)$ in the fourth. Error bars indicate Monte Carlo standard errors.

Overall, the PESRs are increasing with an increasing number of differing datasets, so the performances of the methods for fixed N, p, and alternative class weights are increasing from "3+1" to "2+2" to "2+1+1" to "1+1+1+1" as can be seen in Figure 10 for the special case of $p = 50$ (see Figures 51 to 53 in Appendix F.3 for full results). However, the largest difference between the two datasets seems to be crucial, and this grows with the number of differing datasets. Often, the proportions for a certain combination of N and p and a certain weight vector for "3+1" (see e.g. Figure 51 in Appendix F.3) are comparable to the proportions for the same combination of N and p for "1+1+1+1" and the second weight in the weight vector minus 0.2. In that case, the weights for the fourth dataset of the "1+1+1+1" case coincide with the ones in the "3+1" case.

Regarding the balance of the sample sizes, typically, the performance of each method for a certain combination of N, p, and the alternative class weights is higher for equal sample sizes than for unbalanced sample sizes. This is illustrated in Figure 11 for $p = 50$ and the "1+1+1+1" grouping. See Figures 54 to 57 in Appendix F.4 for the other cases. Especially, the C2ST method's performance suffers severely from unbalanced sample sizes. The performance of MMCM and the method of Petrie (2016) decreases only slightly, except for "2+1+1", "1+1+1+1" and $p = 2$, where the performance of MMCM breaks down, and for "1+1+1+1" and $p = 10$, where the proportion decreases for large deviations for the method of Petrie (2016).

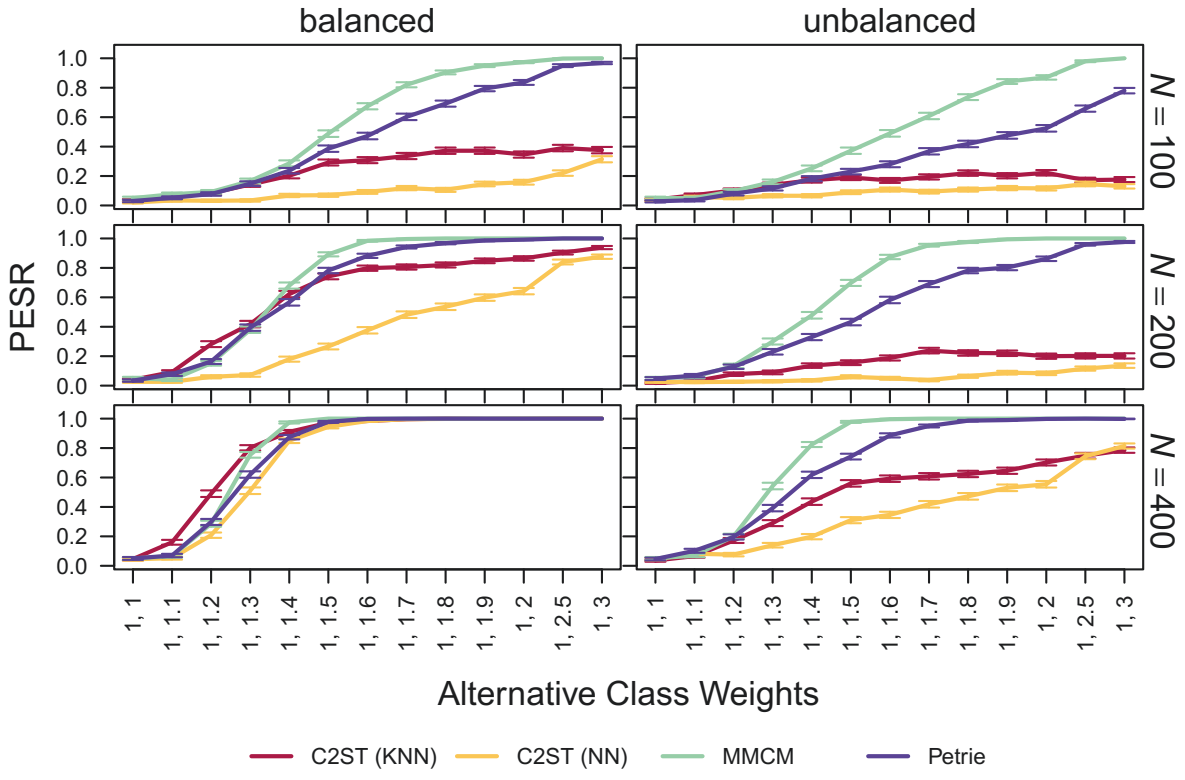


Figure 11: PESR (proportion of extreme simulation repetitions) for $k = 4$ binary datasets of equal ("balanced") or unequal ("unbalanced") sample sizes with $p = 50$ variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, in the second dataset to $(1, 1 + \delta)$, in the third to $(1, 1 + 2\delta)$, and in the fourth to $(1, 1 + 3\delta)$. Error bars indicate Monte Carlo standard errors.

3.2.2 Multinomial Data

Figure 12 shows the method performances for the "1+1+1+1+1" case and balanced sample sizes for increasing skewness of the class probability distribution. Again, the performance increases with the number of differing datasets (see Figure 58 to 60 in Appendix F.5). Moreover, the method performance generally increases with increasing N and p as before. The method of Petrie (2016) is not working as intended for $p = 2$ as the proportions are almost constant regardless of the alternative class weights and at the same time already far from 0.05 for the null situation. The other methods are also performing poorly for $p = 2$ as the proportions are quite low. Only for the highest sample size $N = 400$, the C2ST variants show some increase in the PESR for increasing class weights. For $p = 10$ and $p = 50$, MMCM typically performs best, followed by Petrie's method, then C2ST (NN), and C2ST (KNN) performs worst. For small deviations, the MMCM and Petrie's method often perform similarly, but for larger deviations, MMCM is mostly clearly superior. For $p = 10$ and $N = 100$, C2ST (NN) performs best for small deviations.

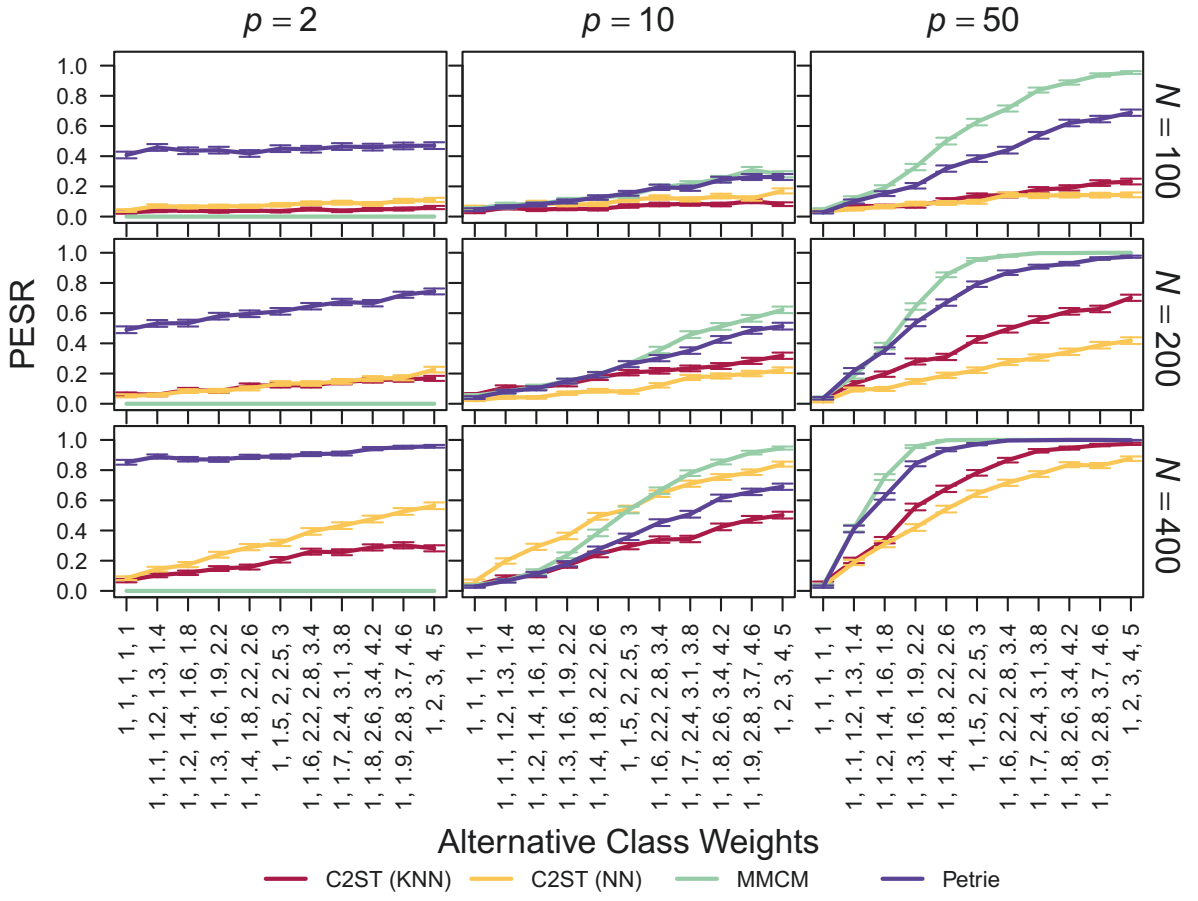


Figure 12: PESR (proportion of extreme simulation repetitions) for $k = 4$ multinomial datasets of equal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first dataset are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta, 1 + 2\delta, 1 + 3\delta, 1 + 4\delta)$ for each variable in the second dataset. The weights in the third dataset are given by $(1, 1 + (\delta + 0.1), 1 + 2(\delta + 0.1), 1 + 3(\delta + 0.1), 1 + 4(\delta + 0.1))$, and the weights for the fourth dataset are given by $(1, 1 + (\delta + 0.2), 1 + 2(\delta + 0.2), 1 + 3(\delta + 0.2), 1 + 4(\delta + 0.2))$. Error bars indicate Monte Carlo standard errors.

As in the binary case, the methods suffer from imbalance, except for the method of Petrie (2016) in the "3+1" case, which gets better and outperforms other methods in that case. The performances of the MMCM and Petrie's method are not as severely impacted as those of the C2ST methods.

Figure 13 shows the method performances for the "1+1+1+1" case and balanced sample sizes for increasing the class probability of one class and decreasing the class probability of another class. The performance for the other groupings is again increasing in the number of differing datasets (see Figure 65 to 67 in Appendix F.7). For $p = 2$, the method of Petrie (2016) and the MMCM statistic give unacceptable results, while the C2ST variants work as intended but not very well. For $p > 2$, the C2ST methods clearly outperform MMCM and Petrie's method, which perform very poorly. Using a neural network performs better than or is comparable to using KNN for the C2ST in most cases. The performances are again overall increasing in N and p. Again, the C2ST methods suffer from unbalanced sample sizes and perform quite poorly for low N. C2ST (KNN) is more heavily impacted than C2ST (NN) (see Figure 68 to 71 in Appendix F.8).

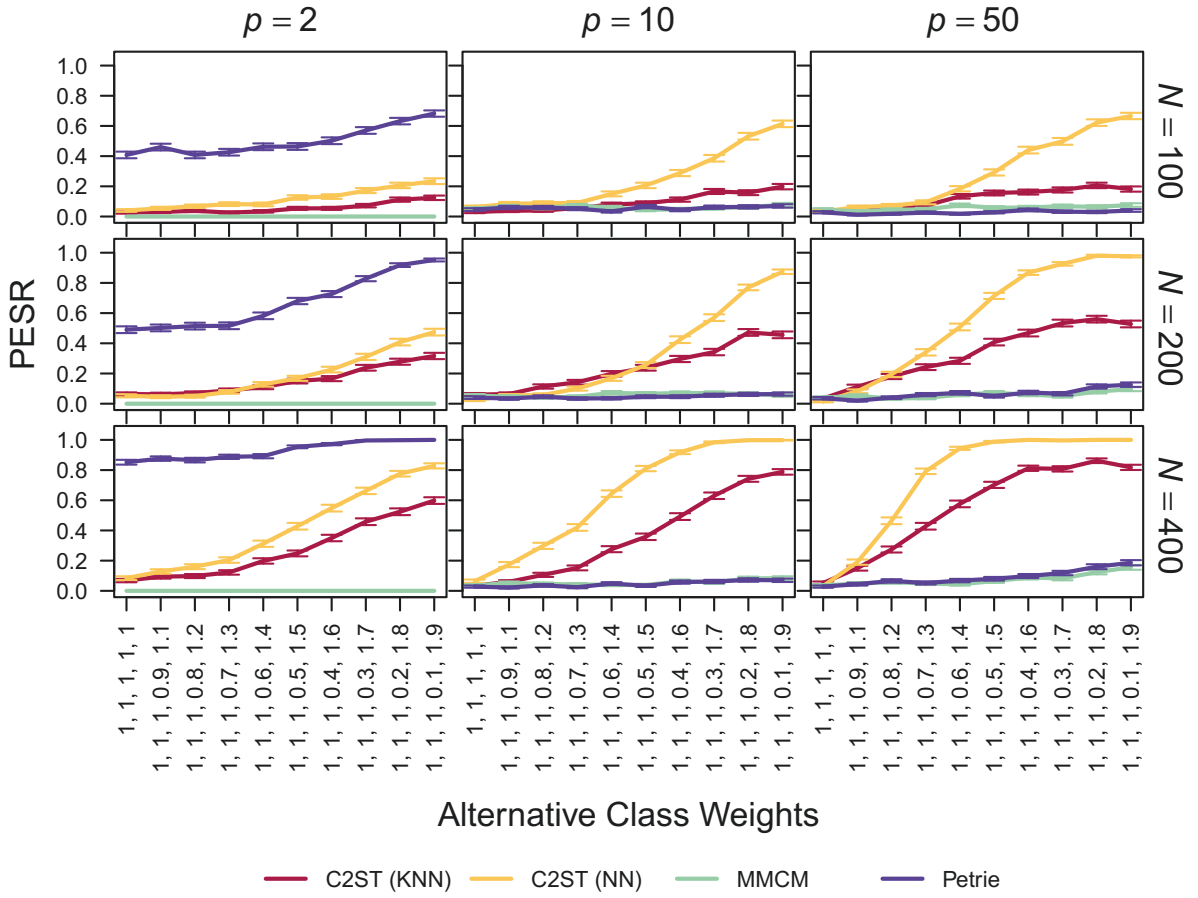


Figure 13: PESR (proportion of extreme simulation repetitions) for $k = 4$ binary datasets of equal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first dataset are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1+\delta, 1-\delta)$ in the second dataset. The weights in the third dataset are given by $(1, 1, 1, 1 + \delta + 0.1, 1 - \delta - 0.1)$, and the weights for the fourth dataset are given by $(1, 1, 1, 1 + \delta + 0.2, 1 - \delta - 0.2)$. Error bars indicate Monte Carlo standard errors.

3.2.3 Summary of Best-performing Methods

Figure 14 summarizes the findings for the multi-sample case. A decision tree is shown in which the decision rules for determining the best method are given. These decision rules depend on the type of alternative, the grouping, the balance of the sample sizes, the number of variables p, and the number of observations N in the pooled sample. For the "1 up, 1 down" alternative, the C2ST (NN) performs best. For the binary and multinomial "skewed" case, $p > 2$, and balanced sample sizes, the MMCM performs best in high variable/low sample size settings, otherwise, the C2ST (NN/KNN) is better. For the binary and multinomial "skewed" case, $p > 2$, and balanced sample sizes, the MMCM performs best except for the binary "3+1" case in which Petrie's method is better. For the multinomial "skewed" case, and $p = 2$, the C2ST (NN) performs best in the case of balanced sample sizes, and no method performs well for unbalanced sample sizes. For binary data and $p = 2$, Petrie's method is best for balanced data or unbalanced data and the "3+1" grouping. For other groupings and unbalanced data, the C2ST (NN) is best.

As in the two-sample case, the median of the differences between the PESR of each method and that of the scenario-specific best-performing method is calculated for each combination of the sample size balance, the number of categories, and p. Figure 15 shows these median differences and the overall median differences. It can be seen that overall, the C2ST (NN) has the lowest median difference and, therefore, performs best. It is followed by the MMCM and C2ST (KNN). Petrie's test performs the worst. This can mainly be attributed to the high penalty applied for $p = 2$ in the multinomial setting, where the method has too high PESR values even in the null setting.

The ordering of the methods in the four-sample case is mostly consistent with that in the two-sample case. However, it should be noted that all methods that are available in the four-sample case are among the worst-performing methods from the two-sample case.

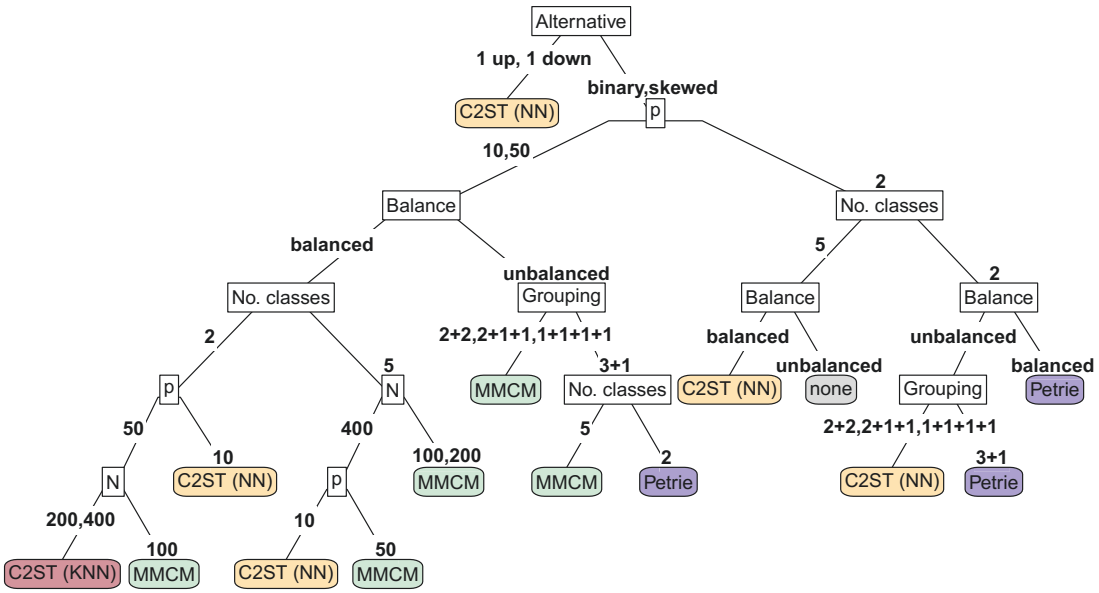


Figure 14: Summary of best-performing method per scenario for the two-sample case.

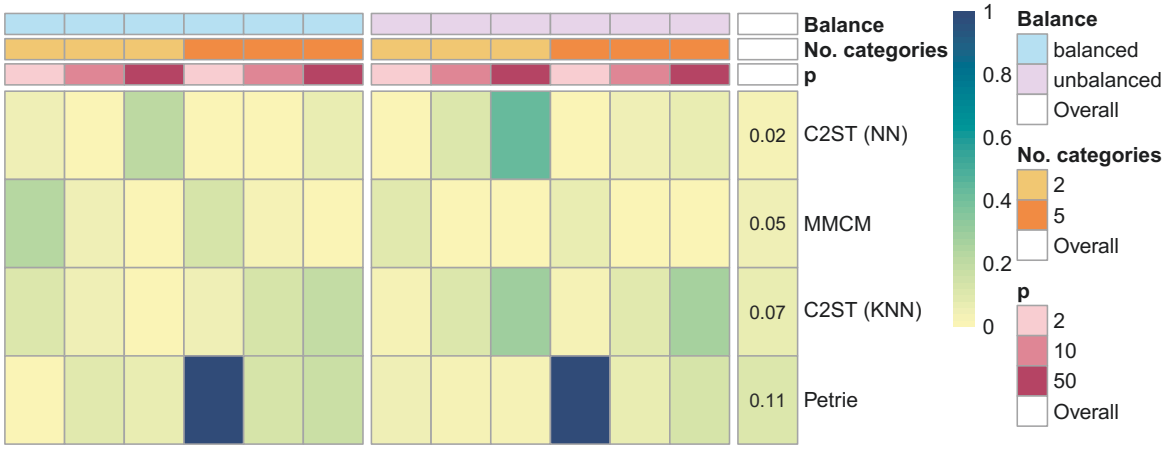


Figure 15: Median difference of the PESR values to that of the best-performing method per scenario for the two-sample case.

4 Clustering of Methods

In the following, the aim is to identify groups of methods that perform similarly across deviations. To find such groups, the proportions of extreme simulation repetitions (PESR) values are clustered using hierarchical clustering with complete linkage and Euclidean distances. For clustering, the PESR values per combination of a deviation and grouping are considered as variables. The combinations of the method, N, p, and sample size balance are considered as observations. The idea behind this division is that the deviation and grouping would be unknown in a real-world dataset comparison, while the dimensions of the datasets and the chosen method are known. Both observations (rows) and variables (columns) are clustered to see which methods act similarly for different N, p, and balance combinations across different alternatives. The clustering of the deviations thereby facilitates identifying groups of deviations for which the identified groups of methods perform particularly well or poorly.

The PESR value clustering is visualized using heatmaps. The clustering is shown in dendrograms. The dendrograms are ordered using an optimal leaf ordering that minimizes the sum of the distances along the path connecting the leaves in the given order (Bar-Joseph et al., 2001). This optimal ordering is used since it is expected to give a sensible order of the deviations in increasing strength.

4.1 Two-sample Setting

For $k = 2$, the clustering is performed for each N and p combination separately since the large number of methods makes it impossible to show all methods for all N and p simultaneously, and the results from the PESR curve comparisons suggest that using the dataset dimensions represents a sensible stratification. The resulting clusterings are discussed in the following, first for binary and afterward for multinomial data. All methods are considered; no pre-selection of methods is used here.

4.1.1 Binary Data

Figure 16 shows the clustering results for two binary datasets with $N = 200$ and $p = 10$ as a heatmap with dendrograms. The results for other N and p are similar, with the overall

trend of increasing PESR values for increasing N and p (see Appendix F.9). There are no distinct clusters visible. The optimal sorting of the deviations corresponds to an increasing (or in some cases decreasing) ordering of the deviations on the x-axis. The methods are sorted from top to bottom by decreasing PESR values. Typically, more balanced scenarios are among the higher-performing part of the ordering, and more unbalanced scenarios are among the lower-performing part. The ordering of the methods matches the comparison of PESR curves discussed before. The edge count methods are clearly sorted by the used graph type, with the $K = 5$ versions in the well-performing methods for $p > 2$ and the $K = 1$ versions in the well-performing methods for $p = 2$.

Overall, the CM distance and the edge count tests perform better than the MMCM, Petrie, C2ST, and YMRZL. HMN is competitive in the balanced sample size case but breaks down completely in the unbalanced sample size case.

4.1.2 Multinomial Data

For categorical data with five classes, the resulting clusterings differ depending on N and p. For lower N and p, the results are similar to the binary case in the sense that there is no clear clustering of the methods, but they are rather ordered by gradually decreasing PESR. This is, for example, shown in Figure 17 as a heatmap with dendrograms for $N = 200$ and $p = 10$. The other heatmaps can be found in Appendix F.10. Again, the scenarios with higher PESR are mostly ones with balanced sample sizes, while most of the unbalanced scenarios are among the lower-performing cases. In contrast to the binary case, for five categories, two types of deviations are considered: the class probability distribution becoming more and more skewed or the probability of one class going up and the probability of another class going down. For lower N and p, these are not well separated in the clustering but rather mixed between these two cases and instead sorted by strength of deviation. One noteworthy exception for the different deviations is the CM distance, which only shows a high PESR for the "skewed" alternatives but a low PESR for the "1 up, 1 down" alternatives.

For higher N and p, there is a clearer distinction between the deviations. This is, for example, shown in Figure 18 as a heatmap with dendrograms for $N = 500$ and $p = 10$. With respect to the deviations, there are four groups: low deviations and medium deviations (each a mix of "skewed" and "1 up, 1 down"), high deviations "skewed", and high deviations "1 up, 1 down".

The edge count tests and the HMN for balanced data show high PESR for all medium to high deviations. These are again mostly sorted by graph type and balance of the sample sizes, with $K = 5$ performing better than $K = 1$ for $p > 2$ (vice versa for $p = 2$) and higher PESR for balanced than for unbalanced settings. The C2ST variants and the YMRZL show mostly higher PESR for "1 up, 1 down" and lower PESR for "skewed" deviations. In contrast, Petrie's method, the MMCM, and the CM distance show high PESR values for "skewed" but very low PESR values for "1 up, 1 down" deviations. The HMN and YRZL for unbalanced sample sizes show (very) low PESR values across all deviations.

Overall, the edge count tests perform best across all deviations for categorical data with five categories. HMN is also competitive in the case of balanced sample sizes, but useless for unbalanced sample sizes. For only detecting "skewed" deviations, the CM distance also performs very well. For only detecting "1 up, 1 down" deviations, the C2ST variants are a competitive alternative.

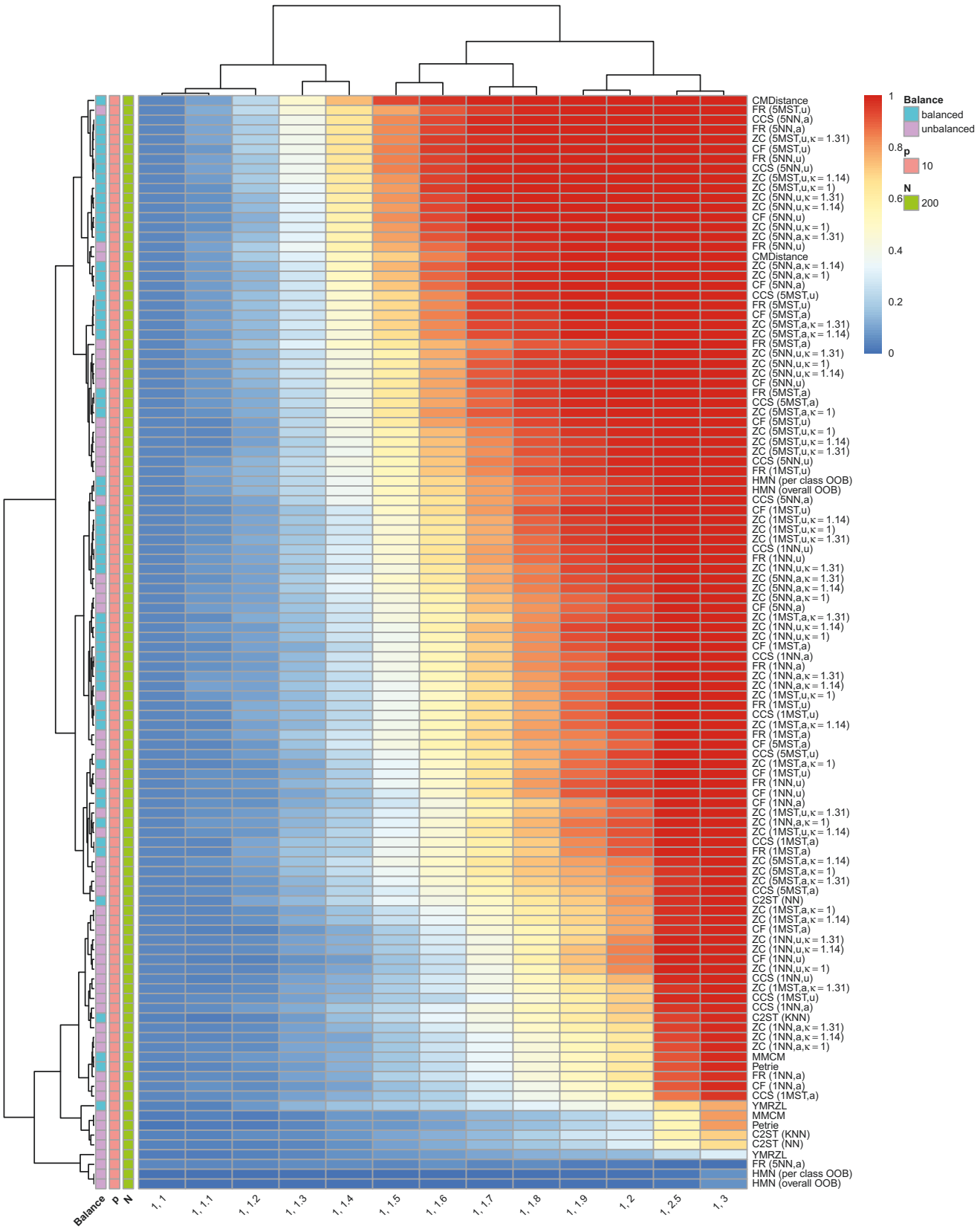


Figure 16: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 200$ and $p = 10$. The labels on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

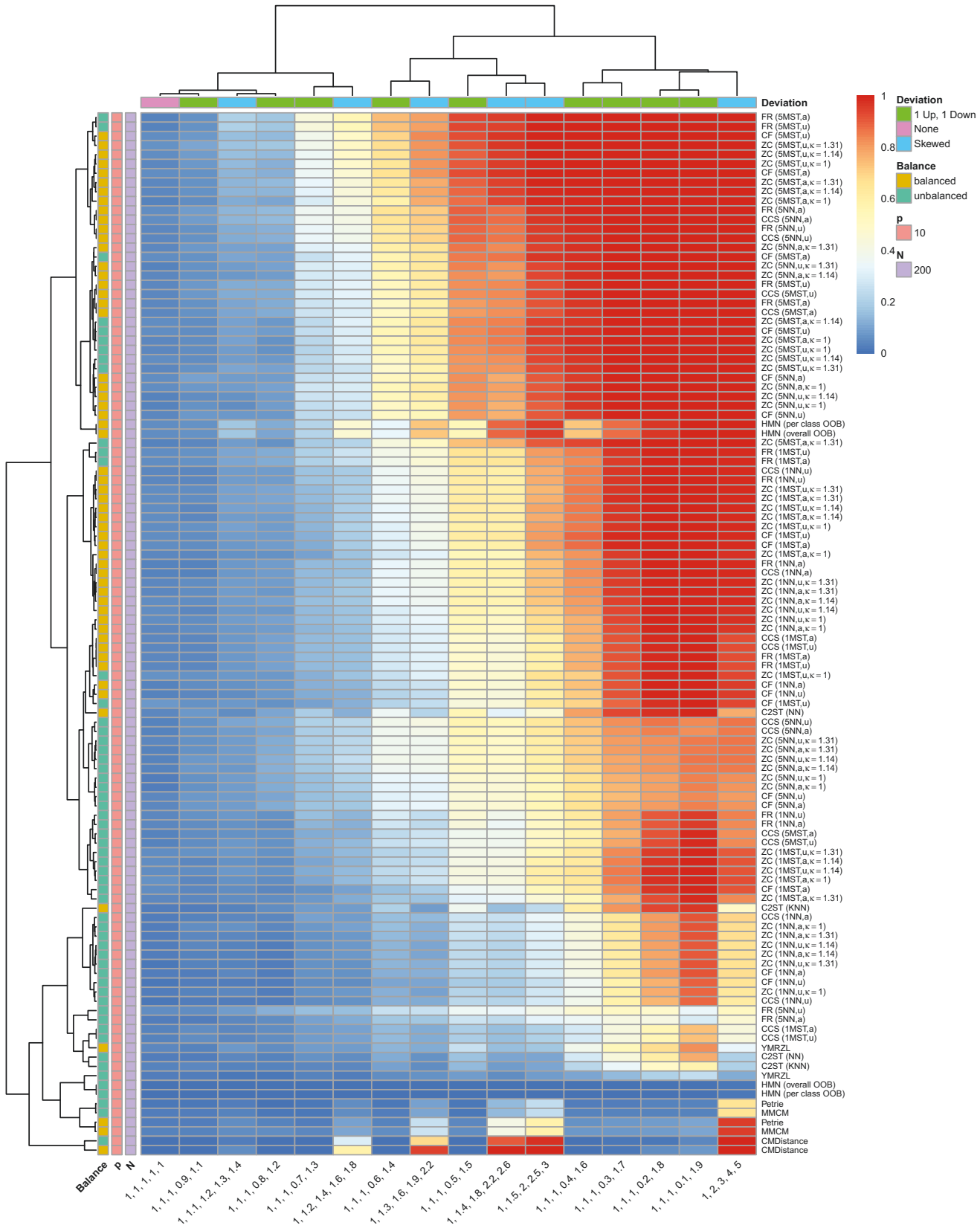


Figure 17: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 200$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

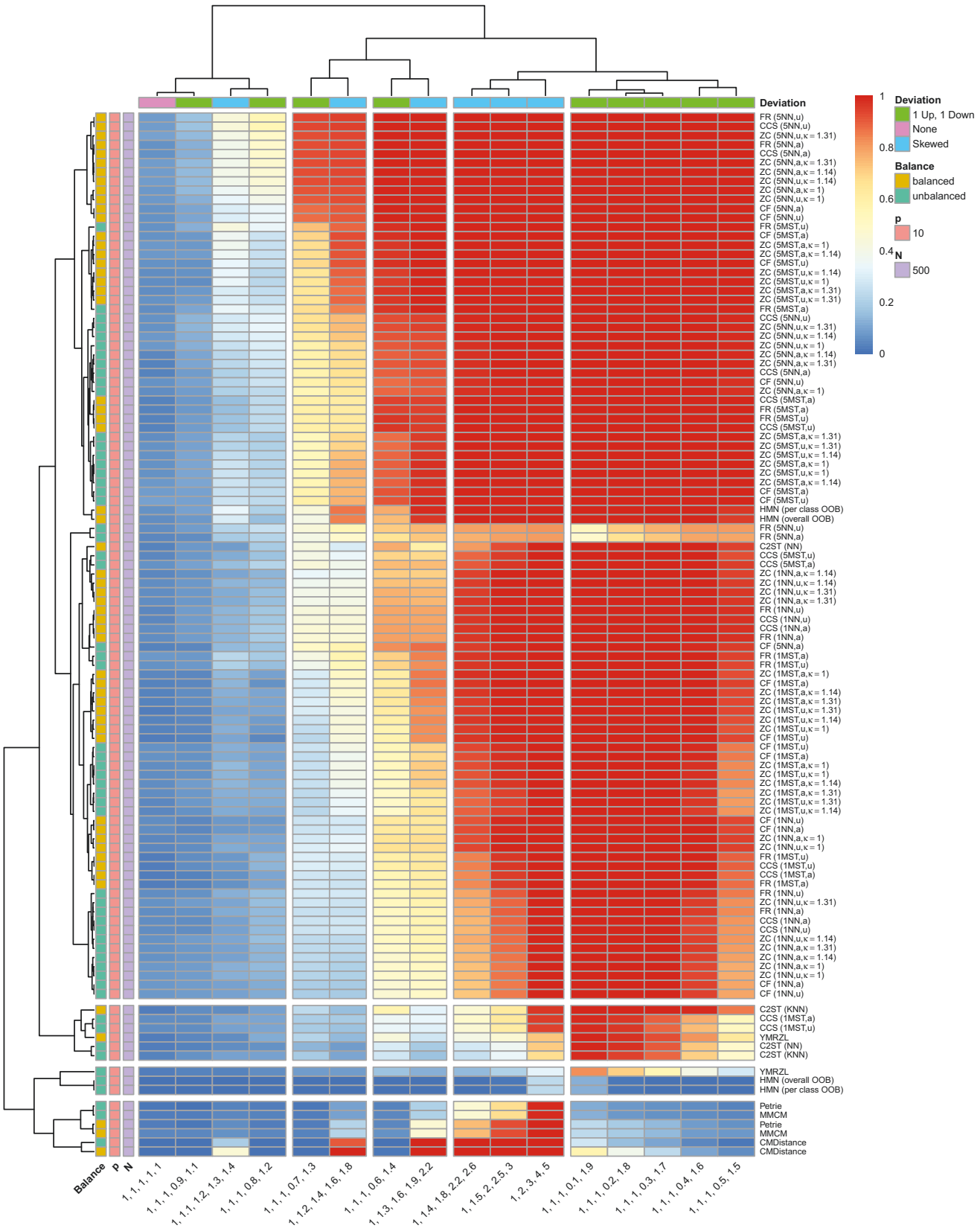


Figure 18: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 500$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

4.2 Multi-sample Setting

In the following, the clusterings for $k = 4$ are presented, first for binary and afterward for multinomial data. Here, all N and p are considered together, which is possible due to the considerably lower number of methods.

4.2.1 Binary Data

The resulting clustering for binary data is shown in the heatmap with dendrograms in Figure 19. As for the two-sample case, there is no clear clustering visible. The scenarios are again ordered by strength. Scenarios with large deviations are on the left, scenarios with medium deviations in the middle, and scenarios with high deviations on the right. The strength of a deviation here seems to depend both on the alternative class weights and on the grouping. Scenarios for groupings with a lower number of differing datasets and high class weights for ones are comparable to scenarios for groupings with a higher number of differing datasets and lower class weights.

The methods are ordered from top to bottom according to decreasing PESR. The MMCM and Petrie's method for medium and high-dimensional datasets, as well as the C2ST (NN or KNN) for the highest dimensional datasets with balanced sample sizes, are among the best-performing methods with high PESR for medium and high deviations

Next in the ordering are MMCM and Petrie's method for medium-dimensional data (mostly unbalanced sample sizes) and the C2ST for high-dimensional data with balanced sample sizes, which have high PESR for high deviations but low PESR for medium and low deviations.

Last in the ordering are mostly the MMCM and Petrie for $p = 2$ and C2ST for low-dimensional data, and especially unbalanced sample sizes.

4.2.2 Multinomial data

Figure 20 shows the resulting clustering for the case of five classes as a heatmap with dendrograms. The scenarios are clustered into three overall blocks based on the severity of the deviation. The left block includes large deviations with skewed class distribution, the block in the middle includes small deviations or groupings "2+2" and "3+1", and the block on the right includes large deviations with one class probability up and one class probability down. The deviations are very clearly split into "skewed" (on the left) and "1 up, 1 down" (on the right).

There are four clusters of methods and N, p, balance combinations. The first block includes methods with high PESR for both kinds of deviations. However, the PESR values for methods in this block are mostly already high under the null scenario and for small deviations, so these are rather methods that do not work as intended and not methods that perform particularly well. These include Petrie's method for $p = 2$ (performing badly), the C2ST (NN) for balanced sample sizes and $N = 400, p \geqslant 10$, and the C2ST (KNN) for balanced sample sizes, $N = 400, p = 50$ (performing well).

The next block includes methods with high PESR for detecting one up, one down deviations, but low PESR for skewed class distributions. These consist of the C2ST variants for high N or p and mostly balanced sample sizes, or for unbalanced sample sizes and the highest N, p combination, as well as Petrie's method for $p = 2$ and $N = 200$ and unbalanced sample sizes.

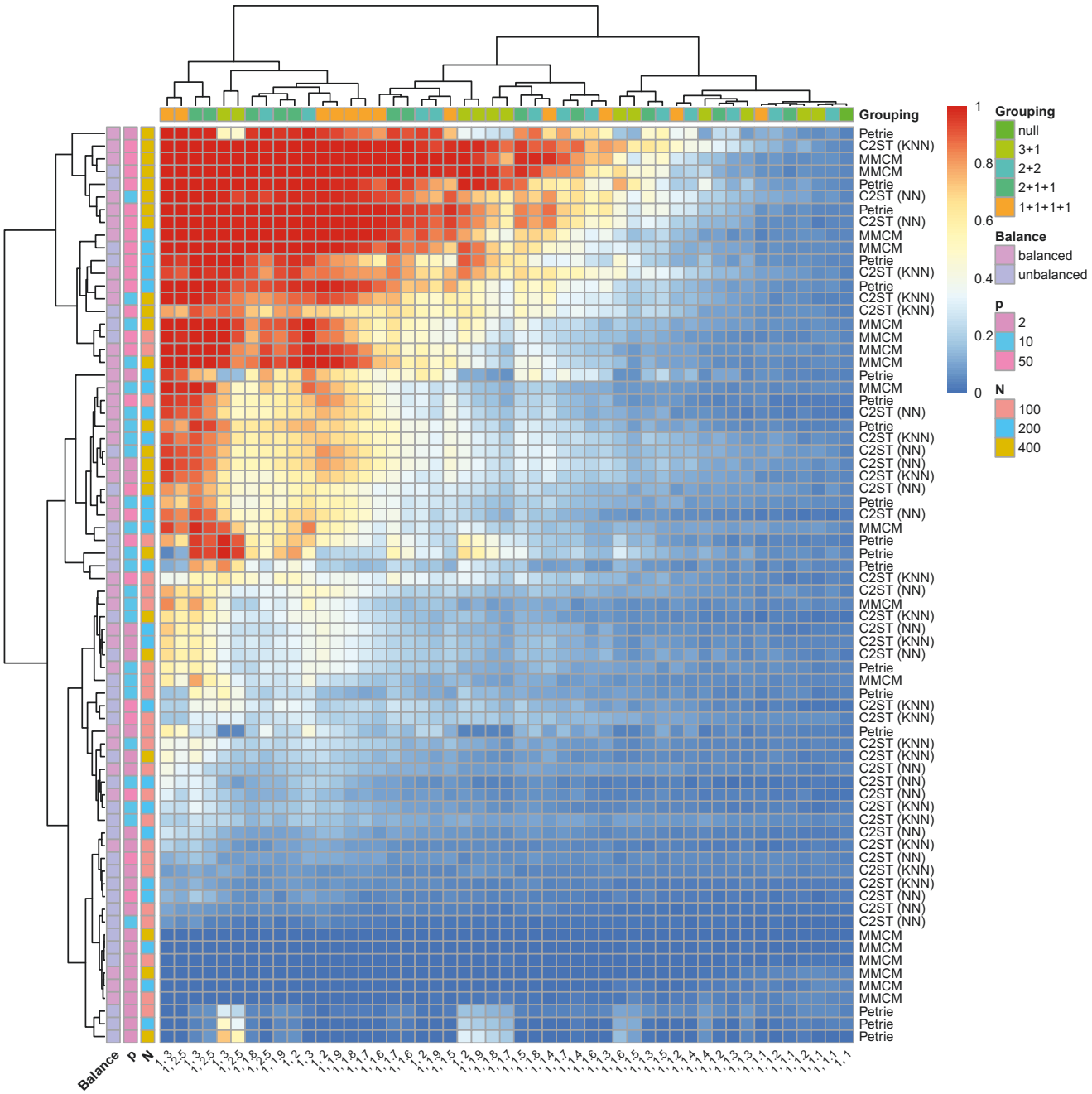


Figure 19: Clustering of PESR values per deviation (x-axis) and per method and dataset dimension (y-axis) for four binary datasets. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

The third block includes methods with (very) low PESR values. These are Petrie's method for $p = 2, 10$, and mostly unbalanced sample sizes, the C2ST (NN) for low N and p, and the C2ST (KNN) for low p or low N or unbalanced sample sizes.

The last block includes methods with high PESR for skewed class distribution deviations but low PESR for one up, one down deviations. These include the MMCM and Petrie's method for high p or N settings.

Across all blocks of methods, the PESR values for the cluster of small deviations are low. Overall, the clustering shows that the graph-based methods MMCM and Petrie are better at detecting skewed class distribution, while the C2ST variants are better at detecting

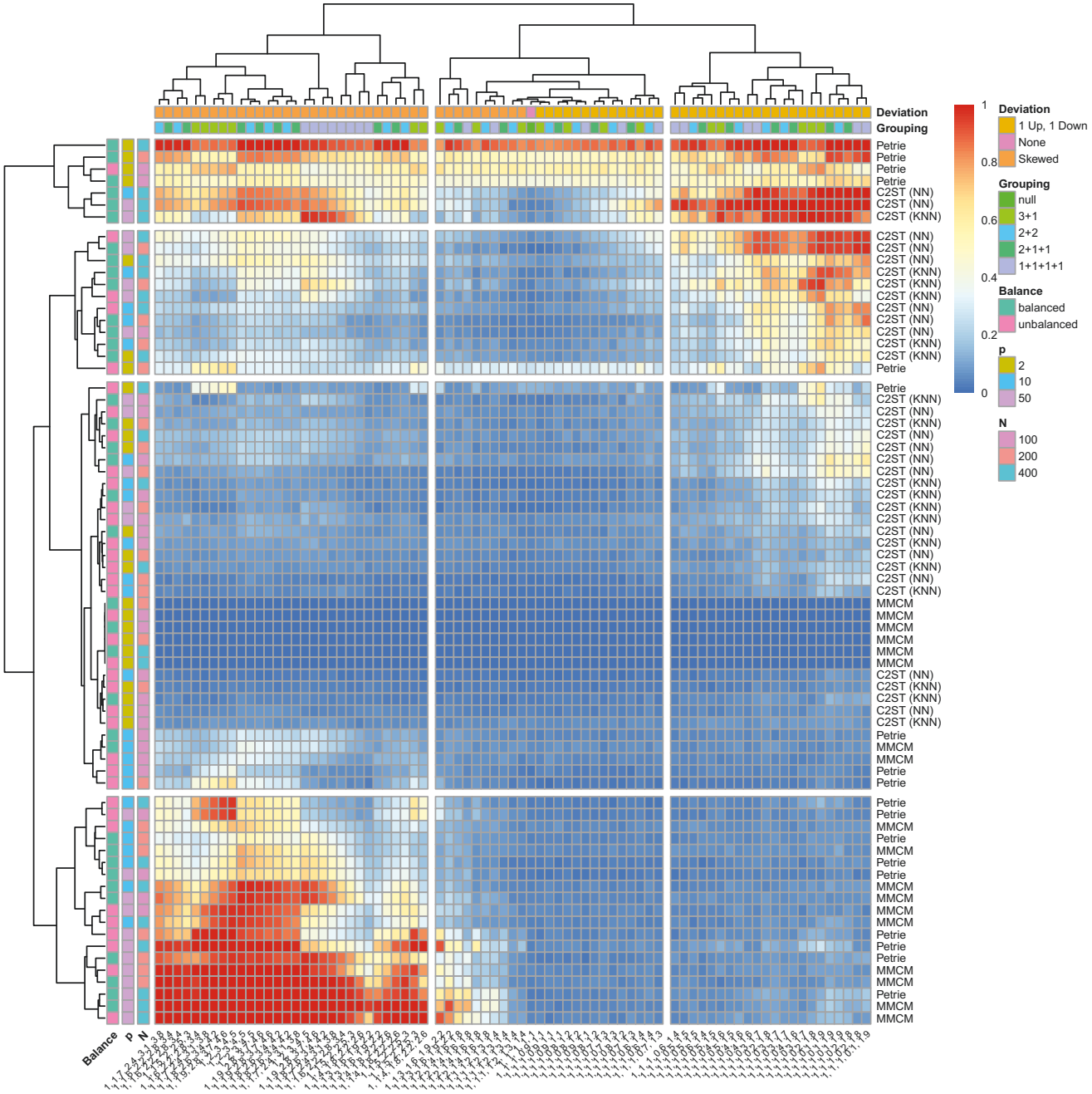


Figure 20: Clustering of PESR values per deviation (x-axis) and per method and dataset dimension (y-axis) for four categorical datasets where each variable is drawn from a multinomial distribution. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

one up, one down alternatives. Moreover, the graph-based methods suffer heavily from low p, while the classifier-based methods suffer heavily from unbalanced dataset sizes and low N and p.

5 Applicability in Practice

In the following, the runtime and memory consumption of the methods are compared, and the errors and numerical problems that occur are discussed.

5.1 Runtime

The runtimes of the methods are compared, first for the two-sample setting and then for the multi-sample setting.

5.1.1 Two-sample Setting

Figure 21 shows boxplots of the runtimes for each of the pre-selected methods for the scenario with two binary datasets with balanced class probabilities and equal sample sizes. The full results for all methods can be found in Figure 99 in Appendix F.11. For $p = 2$, the runtimes of Petrie's method and the CM distance are the lowest, followed by the edge count tests and HMN. The C2ST variants take considerably longer, with C2ST (KNN) being faster than C2ST (NN). The edge count tests' runtimes increase with increasing p and N, while the other runtimes remain the same. Thus, for increasing N and p, the edge count tests have higher runtimes than HMN and consequently also than the C2ST variants. There are very slight differences in the runtimes of the edge count tests between the graph types. The 1NN, "u" versions are the fastest, followed by 5MST, "u". The 5NN, "a" takes the longest. There are no differences between the different edge count tests, which might be due to the implementation in which the required quantities for all tests are calculated regardless of which test is actually performed.

5.1.2 Multi-sample Setting

Figure 22 shows boxplots of the runtimes for each method for the scenario with four binary datasets with balanced class probabilities and equal sample sizes. The results are similar to those for the two-sample setting. There is a clear method ranking with regard to runtime in the considered scenario. The method of Petrie (2016) and the MMCM have the lowest runtimes. The C2ST variants need considerably more time. The C2ST (NN) again has considerably higher runtimes than the C2ST (KNN). The runtimes of all methods do not seem to be impacted much by the dimension of the dataset in this setting.

5.2 Memory Consumption

The memory consumption is compared between the methods in addition to the runtime. In the following, the results of that comparison are presented, again, first for the two-sample case and then for the multi-sample case.

5.2.1 Two-sample Setting

Figure 23 shows boxplots of the memory allocation for each method for the scenario with two binary datasets with balanced class probabilities and equal sample sizes for the pre-selected methods that were compared before. The full results for all methods can be found in Figure 100 in Appendix F.11.

Figure 21: Runtime comparison for the scenario with two binary datasets with balanced class probabilities and equal sample sizes. At least ten repetitions were performed for each method.

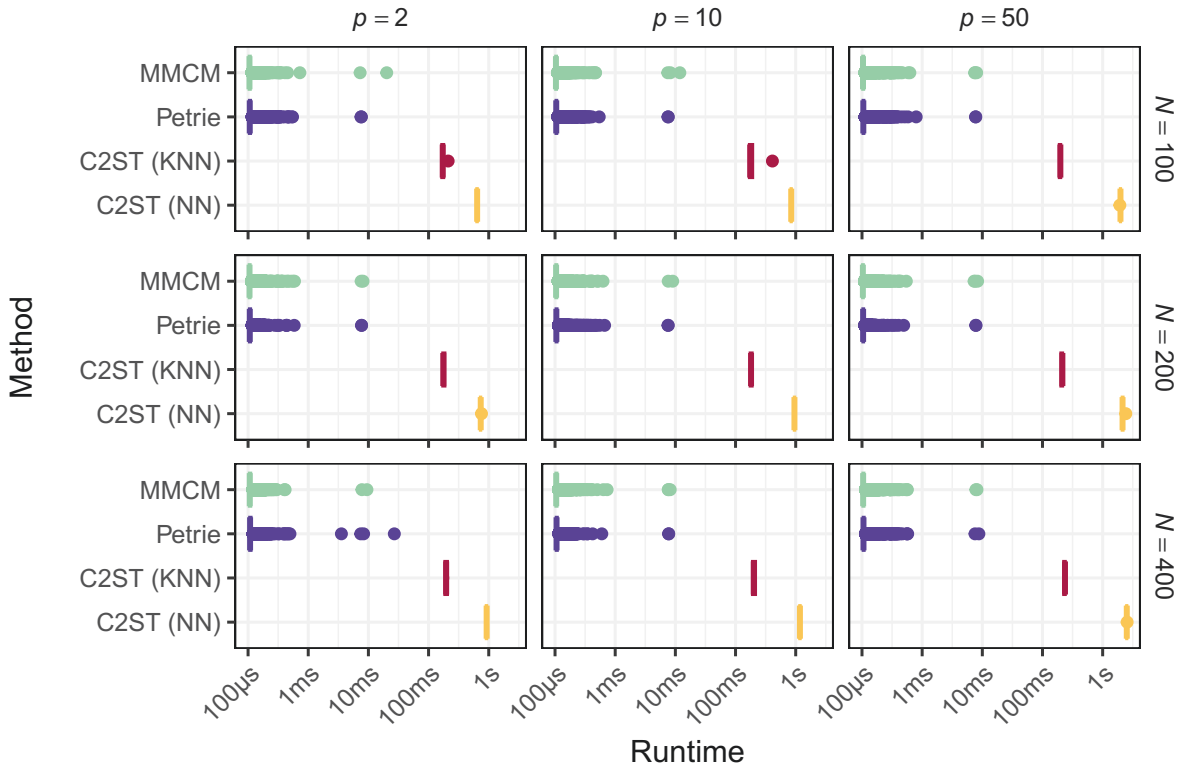


Figure 22: Runtime comparison for the scenario with four binary datasets with balanced class probabilities and equal sample sizes. Ten repetitions were performed for each method.

For too small runtimes, the memory allocation cannot be measured. This is the case for Petrie's method and for some N and p combinations also for the CM distance. In the cases where a measurement could be retrieved, the CM distance has the lowest memory allocation. The remaining order is similar to that for the runtime: for low p, the edge count tests need the second least memory, followed by the HMN, and the C2ST needs the most. For increasing N and p, the memory allocation of the edge count tests increases while the other methods are not affected, so the edge count tests again need the most resources for the higher p and N combinations.

With regard to memory, there are clearer differences between the graphs. The 1NN, "u" versions use the least memory, followed by 5MST, "u". The 5NN, "a" requires the most memory. When looking at the full results, this is not a difference between "a" and "u", but the $K = 5$ versions consistently require more memory than the $K = 1$ versions, and calculating the nearest neighbor graph allocates more memory than calculating the minimum spanning tree.

5.2.2 Multi-sample setting

The memory consumption of each method for the scenario with four binary datasets with balanced class probabilities and equal sample sizes is shown in Figure 24. The memory of the MMCM and the method of Petrie (2016) cannot be measured due to their very short runtimes. Therefore, in the figure, no values are plotted for these two methods. For C2ST, the memory consumption increases with the number of variables and the number of observations.

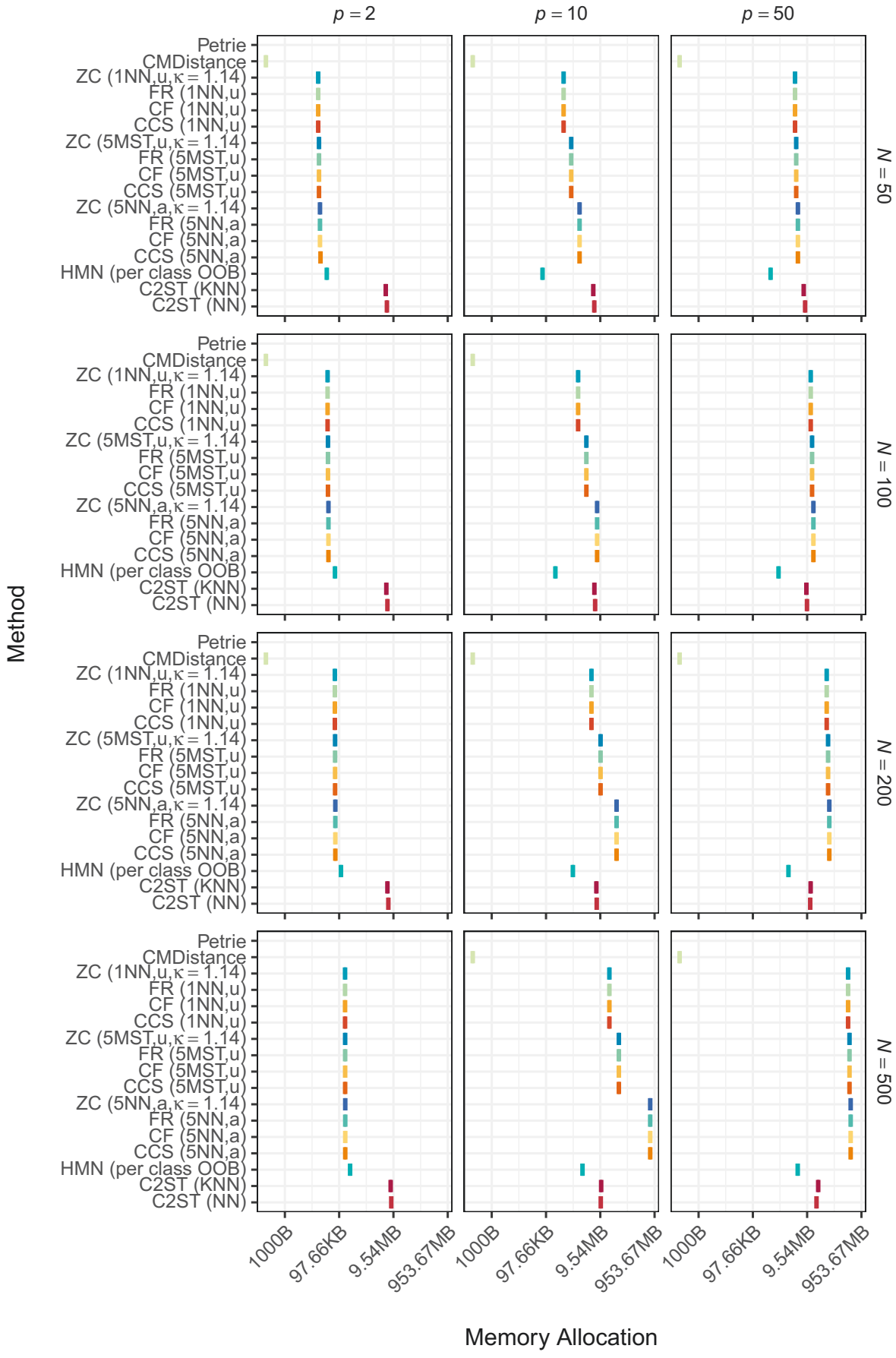


Figure 23: Memory consumption comparison for the scenario with two binary datasets with balanced class probabilities and equal sample sizes. One repetition was performed for each method.

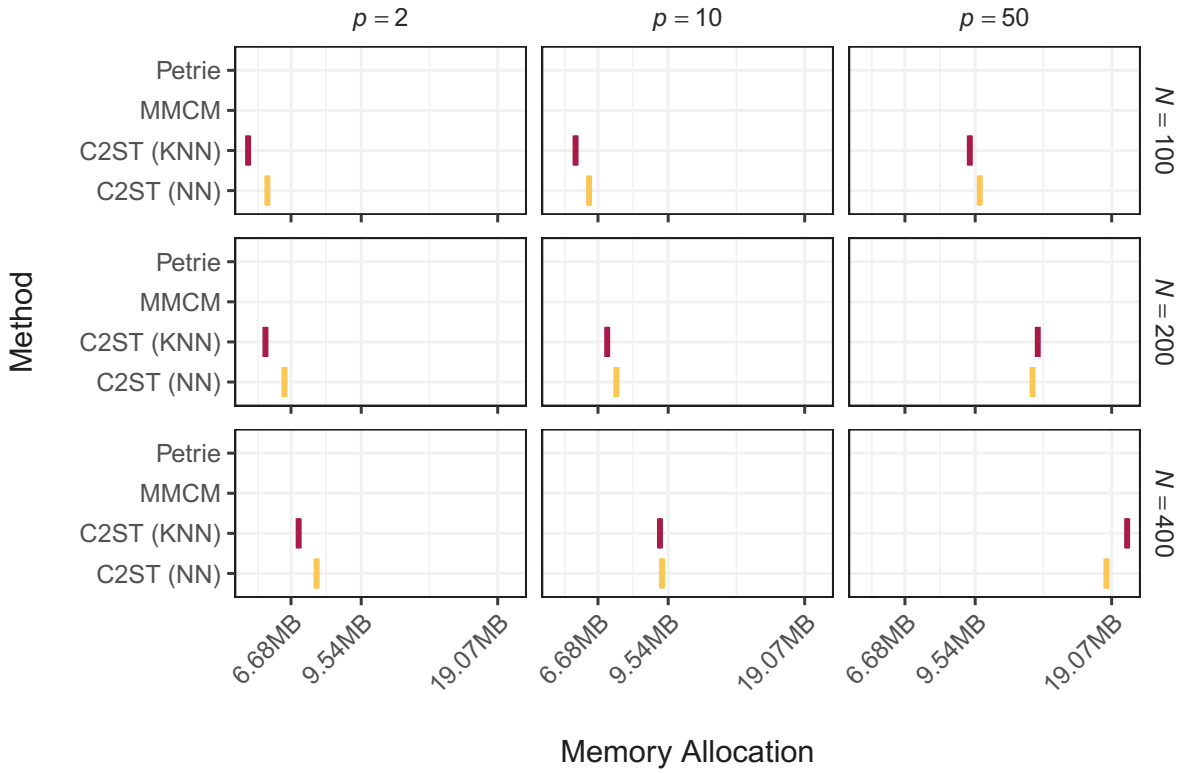


Figure 24: Memory consumption comparison for the scenario with four binary datasets with balanced class probabilities and equal sample sizes. One repetition was performed for each method.

5.3 Errors

In the following, the errors that occurred in the simulation study are discussed.

5.3.1 Two-sample Setting

Table 1 gives an overview of the number of errors per scenario and method that occurred during the simulations for $k = 2$. For most method and scenario combinations, no errors occurred. For some combinations, one to eight errors occurred, which is negligible considering the total number of 500 iterations. For 160 method and scenario combinations, errors occurred in all repetitions such that no results could be retrieved. These are exactly the scenarios with $p = 50$ and 5 categories for the CM distance where the enumeration of the sample space is infeasible.

The remaining errors are related to technical problems while running the simulation study. There were sporadic problems with the file system or the parallel loading of packages. These are not related to the methods themselves and are unlikely to occur again when repeating the simulation study.

In addition to the occurring errors, there were also infinite, missing, and `NaN` (not a number) values for the test statistics that were not the result of an error. For HMN (per class OOB), the test statistic is not defined for perfect classification since the variance of the classification error used to standardize the statistic is zero in that case. The test statistic is then set to infinity in the implementation. This occurred 50 times within the simulation study, mostly for the highest deviations but spread across scenarios such that

No. Errors	No. Scenarios
0	376026
1	94
2	7
5	13
6	13
7	6
8	1
500	160

Table 1: Frequency of numbers of occurring errors. No. Errors: Number of repetitions of the simulation in which an error occurred. No. Scenarios: Number of methods and scenarios (combination of N, p, balance, deviation) in which a certain number or errors occurred.

for each individual scenario, only a few repetitions were affected. For $p = 2$, the test statistics of all edge count tests using the 5NN, "u" are undefined due to standardization with a variance of zero. For CF and ZC, this also holds for the 1NN, "a". For more details on this issue, see Appendix E.

5.3.2 Multi-sample Setting

No errors occurred in any repetition of any scenario. All methods appear to be numerically stable in the chosen scenarios for the four-sample case.

6 Discussion and Conclusion

Quantifying the similarity of two or more datasets is an important task in statistical and machine learning applications. There are various methods for quantifying dataset similarity proposed in the literature, but there are no neutral comparisons of the empirical performance of such methods. Stolte et al. (2024) presented a taxonomy and theoretical comparison of such methods. That article, however, did not include method performance in practice. Here, such a comparison is performed for selected methods from the aforementioned review that are applicable to categorical data and that performed well with regard to the theoretical criteria. The aims of this comparison study are to:

1. Compare dataset similarity measures with respect to their performance in detecting differences of datasets drawn from distributions that differ in certain aspects.

2. Identify groups of dataset similarity measures that act similarly across different alternatives.

3. Compare dataset similarity measures with respect to their consumption of computational resources.

For this, a simulation study was performed. As no proper power comparison of the methods was feasible, the proportion of extreme simulation repetitions (PESR) is considered instead. In short, this is the proportion of repetitions in which the observed statistic

value is more extreme than a threshold calculated from simulations under a null scenario in which the true distributions from which the datasets are generated do not differ. Three main scenarios were considered for the two- and k-sample case with $k = 4$.

1. Binary datasets with equal or differing class distributions.

2. Categorical dataset with equal class distributions or differing class distributions that gradually become more skewed.

3. Categorical dataset with equal class distributions or differing class distributions where the probability for one class increases while the probability of another class decreases accordingly ("1 up, 1 down").

In the two-sample case, a total of 56 variants of 10 methods were included. Out of these, 17 could be pre-selected for the overall comparisons by excluding variants that consistently performed worse than the selected variants. The excluded methods were:

- The YMRZL (Yu et al., 2007), which was consistently worse than the C2ST (Lopez-Paz and Oquab, 2017).

- The edge count tests (Friedman and Rafsky, 1979; Chen and Friedman, 2017; Chen et al., 2018; Zhang and Chen, 2022) using the 1NN graph and averaging over optimal graphs ("a"), the 5NN, "a", and the 1MST, both for averaging and union ("u") over optimal graphs. These were consistently outperformed by 1NN, "u"; 5MST, "u"; and 5NN, "a".

- The MMCM (Mukherjee et al., 2022), which was consistently outperformed by the method by Petrie (2016).

- The HMN (Hediger et al., 2021) using the overall out-of-bag (OOB) error, which was consistently outperformed by the per-class OOB.

The results for the remaining methods in the two-sample case can be summarized as follows. In general, the PESR values for all methods increased with increasing sample sizes N. For most methods, the PESR values also increased with increasing numbers of variables p. Exceptions from this were the edge count tests using the 1NN or 1MST, for which the PESR decreased in most cases for increasing p, and the C2ST (NN) for which the PESR decreased in some cases for increasing p. The PESR values for each method and scenario are lower for unbalanced sample sizes than for balanced sample sizes. For binary datasets and for multinomial datasets with the "skewed" alternative and small numbers of variables $p = 2, 10$ and with equal sample sizes, the constrained minimum (CM) distance (Tatti, 2007) performed best. It is, however, infeasible to calculate for categorical data with five categories and $p = 50$ variables, and it is affected by the imbalance of the sample sizes. Moreover, it was unable to detect the "1 up, 1 down" alternatives in the multinomial case. In the cases where the CM distance has its weaknesses, one of the edge count tests was best. Otherwise, these were the next best alternatives to the CM distance.

The tests using the 1NN, "u" were then best for $p = 2$ and the tests using the 5MST, "u", or 5NN, "a" for $p = 10, 50$. The differences between the edge count tests were small for balanced sample sizes. For unbalanced sample sizes and binary data, the weighted edge count test (Chen et al., 2018) was best. This was expected as it was specifically intended for unbalanced sample sizes. For unbalanced multinomial data, however, the original

edge count test (Friedman and Rafsky, 1979) was best. The HMN was competitive for balanced data and higher sample sizes N and numbers of variables p, but its performance broke down in the case of unbalanced sample sizes. The C2ST variants (using a KNN classifier or a multilayer perceptron) were typically outperformed by the CM distance, the edge count tests, and the HMN. They were also more heavily affected by the imbalance of the sample sizes than the edge count tests. Petrie's method was often the worst in the comparison. For datasets with $p = 2$, it was not working as intended.

Overall, there was a tendency for the edge count tests to show high performance for all alternatives. Especially the FR (5MST, u) can be recommended. The CM distance showed high performance for binary data or multinomial data with the "skewed alternative" but only for balanced sample sizes. In those cases, it had the highest PESR values even for small deviations. The HMN is somewhere in the middle field for balanced sample sizes and the worst for unbalanced sample sizes. The C2ST had comparably low PESR values and was better for detecting the "1 up, 1 down" alternative than for the "skewed" alternative in the multinomial case.

The observation that denser graphs, such as the 5MST and 5NN, instead of the MST and 1NN, perform better empirically for higher-dimensional data is in line with earlier simulation studies. For a detailed discussion of the use of denser graphs, refer to Zhu and Chen (2024). In particular, they derive less strict assumptions for the asymptotic distributions of the edge count statistics that allow for denser graphs than the assumptions made in the original articles.

For the multi-sample case, four datasets were considered, and all possible combinations of how many of these four datasets can differ from each other. Only the C2ST, Petrie's method, and the MMCM were applicable to more than two samples. The comparison of these can be summarized as follows. Again, the PESR values increase for increasing N and p. They also increase with an increasing number of differing pairs of datasets. The PESR values for each method and scenario are again lower for unbalanced sample sizes than for balanced sample sizes. For binary datasets, the MMCM or Petrie's method are often the best. These do, however, fail again for $p = 2$ due to the presence of many ties. In the multinomial case, MMCM and Petrie's method are better at detecting the "skewed" alternatives while the C2ST is better at detecting "1 up, 1 down". Often, the C2ST (NN) is better than the C2ST (KNN). The C2ST is more heavily affected by unequal sample sizes than the MMCM and Petrie's method are.

With regard to runtime and memory, the CM distance, the MMCM, and Petrie's method performed best. The edge count tests' performances depended on the number of samples and the number of variables. For small p, their computational costs are lower than those of the HMN and the C2ST variants; for increasing p (and N), the costs increase while those of the HMN and C2ST variants are almost constant, such that the resource consumption of the edge count tests exceeds the others. For the classifier-based tests, the HMN had considerably lower resource consumption than the C2ST variants, and the C2ST (KNN) had lower runtime and memory allocation than the C2ST (NN). The memory allocation for Petrie's method and the MMCM could not be evaluated due to their very short runtime. These results should, however, be interpreted with caution as they might depend on the operating system, the configuration of the computer, and `R` (R Core Team, 2024) and the required packages. Moreover, memory can only be measured for computations that are performed in `R` itself, while computations that are performed in other programming languages cannot be considered. This might give an unfair advantage to the tests based on the optimal non-bipartite matching, where the matching itself is

done in `Fortran` as well as for the HMN, where the random forest calculation is done in `C`. This might be a reason that their memory did not depend on the dataset dimension. The MST calculation for the edge count tests is also performed in `C`, while the NN calculation is performed in `R` itself. The C2ST using KNN and NN also uses internal `C` code for training the models.

One potential limitation of the current study is that the scenarios could not be chosen in a way that, with an increase in the number of variables, the true difference between the datasets increases as each variable follows the same distribution. Therefore, the decreasing performance of methods for increasing numbers of variables might be covered by the increasing true dataset difference. Similarly, the maximum difference in the class probabilities increases with an increasing number of differing datasets, which might explain the increasing ability of the methods to detect differences for an increasing number of differing datasets.

Another aspect that could be improved is the differing implementations of the methods, which allowed for the choice of an appropriate distance measure for calculating the graph for the edge count tests, but not for the MMCM and Petrie's method, where only Euclidean distances could be used. This could in particular influence the performance for the "1 up, 1 down" scenario since for the Euclidean distance, observing category 5 instead of 4 (which was the case that became more likely here in this scenario) is a smaller change than observing category 5 instead of 1, which was far more likely in the "skewed" case. For all other methods, these changes have equal distances, which would be appropriate for nominal data. This might, in part, explain the comparably bad performance of MMCM and Petrie's method for the former scenario and the comparably good performance for the latter. This issue has been fixed in the new implementation used by the `DataSimilarity` package (Stolte and Sauer, 2025), which allows choosing a distance function. Moreover, for the CM distance, only the suggested feature functions were used. Again, the coding in the "1 up, 1 down" scenario might influence the results here. Finding a feature function that is more suitable for detecting the "1 up, 1 down" alternative might be considered in future research, as the CM distance performed so well in the other cases. This highlights the importance of choosing the distance function or feature function, respectively, appropriately for the coding and scale level of the data at hand.

Other aspects that could be considered in the future are the comparison of dataset similarity methods for categorical datasets that include a target variable or for numerical datasets. Based on the comparisons, one could try to combine methods that worked well for different alternatives, like the MMCM / Petrie's method and the C2ST in the multi-sample case, to create a new method or a group of methods that can detect various types of differences between datasets.

Acknowledgments

This work has been supported (in part) by the Research Training Group "Biostatistical Methods for High-Dimensional Data in Toxicology" (RTG 2624, Project P1) funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation – Project Number 427806116).
The authors gratefully acknowledge the computing time provided on the Linux HPC cluster at TU Dortmund University (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

References

Agarwal, S. M. D., Bhattacharya, B., and Zhang, N. R. (2020): *multicross: A Graph-Based Test for Comparing Multivariate Distributions in the Multi Sample Framework*, R package version 2.1.0, URL: https://CRAN.R-project.org/package=multicross.

Alvarez-Melis, D. and Fusi, N. (2020): "Geometric Dataset Distances via Optimal Transport", in: *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., pp. 21428–21439.

Bahr, R. (1996): "Ein neuer Test für das mehrdimensionale Zwei-Stichproben-Problem bei allgemeiner Alternative", PhD thesis, Universität Hannover.

Bar-Joseph, Z., Gifford, D. K., and Jaakkola, T. S. (2001): "Fast optimal leaf ordering for hierarchical clustering", in: *Bioinformatics* 17 (suppl_1), S22–S29, ISSN: 1367-4803, DOI: 10.1093/bioinformatics/17.suppl_1.S22.

Baringhaus, L. and Franz, C. (2004): "On a new multivariate two-sample test", in: *Journal of Multivariate Analysis* 88 (1), pp. 190–206, ISSN: 0047-259X, DOI: 10.1016/S0047-259X(03)00079-4.

Baringhaus, L. and Franz, C. (2010): "Rigid Motion Invariant Two-Sample Tests", in: *Statistica Sinica* 20 (4), pp. 1333–1361, ISSN: 1017-0405.

Biau, G. and Gyorfi, L. (2005): "On the asymptotic properties of a nonparametric L_1-test statistic of homogeneity", in: *IEEE Transactions on Information Theory* 51 (11), pp. 3965–3973, ISSN: 1557-9654, DOI: 10.1109/TIT.2005.856979.

Biswas, M. and Ghosh, A. K. (2014): "A nonparametric two-sample test applicable to high dimensional data", in: *Journal of Multivariate Analysis* 123, pp. 160–171, ISSN: 0047-259X, DOI: 10.1016/j.jmva.2013.09.004.

Biswas, M., Mukhopadhyay, M., and Ghosh, A. K. (2014): "A distribution-free two-sample run test applicable to high-dimensional data", in: *Biometrika* 101 (4), pp. 913–926, ISSN: 0006-3444, DOI: 10.1093/biomet/asu045.

Boulesteix, A.-L., Lauer, S., and Eugster, M. J. A. (2013): "A Plea for Neutral Comparison Studies in Computational Sciences", in: *PLOS ONE* 8 (4), e61562, ISSN: 1932-6203, DOI: 10.1371/journal.pone.0061562.

Buchta, C. and Hahsler, M. (2024): *cba: Clustering for Business Analytics*, R package version 0.2-25, URL: https://CRAN.R-project.org/package=cba.

Chen, H., Chen, X., and Su, Y. (2018): "A Weighted Edge-Count Two-Sample Test for Multivariate and Object Data", in: *Journal of the American Statistical Association* 113 (523), pp. 1146–1155, ISSN: 0162-1459, DOI: 10.1080/01621459.2017.1307757.

Chen, H. and Friedman, J. H. (2017): "A New Graph-Based Two-Sample Test for Multivariate and Object Data", in: *Journal of the American Statistical Association* 112 (517), pp. 397–409, ISSN: 0162-1459, DOI: 10.1080/01621459.2016.1147356.

Chen, H. and Zhang, J. (2017): *gTests: Graph-Based Two-Sample Tests*, R package version 0.2, URL: https://CRAN.R-project.org/package=gTests.

Chen, H. and Zhang, N. R. (2022): *gCat: Graph-Based Two-Sample Tests for Categorical Data*, R package version 0.2, URL: https://CRAN.R-project.org/package=gCat.

Chen, L., Dou, W. W., and Qiao, Z. (2013): "Ensemble Subsampling for Imbalanced Multivariate Two-Sample Tests", in: *Journal of the American Statistical Association* 108 (504), pp. 1308–1323, ISSN: 0162-1459, DOI: 10.1080/01621459.2013.800763.

Deb, N. and Sen, B. (2021): "Multivariate Rank-Based Distribution-Free Nonparametric Testing Using Measure Transportation", in: *Journal of the American Statistical Association* 118 (541), pp. 1–16, ISSN: 0162-1459, DOI: 10.1080/01621459.2021.1923508.

Franz, C. (2019): *cramer: Multivariate Nonparametric Cramer-Test for the Two-Sample-Problem*, R package version 0.9-3, URL: https://CRAN.R-project.org/package=cramer.

Friedman, J. H. and Rafsky, L. C. (1979): "Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests", in: *The Annals of Statistics* 7 (4), pp. 697–717, ISSN: 0090-5364.

Ganti, V., Gehrke, J., Ramakrishnan, R., and Loh, W.-Y. (1999): "A Framework for Measuring Changes in Data Characteristics", in: *Proceedings of the 18th Symposium on Principles of Database Systems*, pp. 126–137.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2012): "A Kernel Two-Sample Test", in: *Journal of Machine Learning Research* 13, pp. 723–773.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., and Smola, A. (2006): "A Kernel Method for the Two-Sample-Problem", in: *Advances in Neural Information Processing Systems*, vol. 19, MIT Press.

Hediger, S., Michel, L., and Näf, J. (2021): *On the Use of Random Forest for Two-Sample Testing*, arXiv:1903.06287 [stat], DOI: 10.48550/arXiv.1903.06287.

Heller, R., Small, D., and Rosenbaum, P. (2012): *crossmatch: The Cross-match Test*, R package version 1.3-1, URL: https://CRAN.R-project.org/package=crossmatch.

Henze, N. (1988): "A Multivariate Two-Sample Test Based on the Number of Nearest Neighbor Type Coincidences", in: *The Annals of Statistics* 16 (2), pp. 772–783, ISSN: 0090-5364.

Hester, J. and Vaughan, D. (2025): *bench: High Precision Timing of R Expressions*, R package version 1.1.4, URL: https://CRAN.R-project.org/package=bench.

Huang, Z. (2022): *KMD: Kernel Measure of Multi-Sample Dissimilarity*, R package version 0.1.0, URL: https://CRAN.R-project.org/package=KMD.

Huang, Z. and Sen, B. (2024): "A Kernel Measure of Dissimilarity between M Distributions", in: *Journal of the American Statistical Association* 119 (548), pp. 3020–3032, DOI: 10.1080/01621459.2023.2298036.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004): "kernlab – An S4 Package for Kernel Methods in R", in: *Journal of Statistical Software* 11 (9), pp. 1–20, DOI: 10.18637/jss.v011.i09.

Kolde, R. (2019): *pheatmap: Pretty Heatmaps*, R package version 1.0.12, URL: https://CRAN.R-project.org/package=pheatmap.

Li, X., Hu, W., and Zhang, B. (2022): "Measuring and testing homogeneity of distributions by characteristic distance", in: *Statistical Papers*, ISSN: 1613-9798, DOI: 10.1007/s00362-022-01327-7.

Lopez-Paz, D. and Oquab, M. (2017): "Revisiting Classifier Two-Sample Tests", in: *International Conference on Learning Representations*, URL: https://openreview.net/forum?id=SJkXfE5xx.

Milborrow, S. (2016): *rpart.plot: Plot rpart Models. An Enhanced Version of plot.rpart*, R package, URL: http://CRAN.R-project.org/package=rpart.plot.

Morris, T. P., White, I. R., and Crowther, M. J. (2019): "Using simulation studies to evaluate statistical methods", in: *Statistics in Medicine* 38 (11), pp. 2074–2102, ISSN: 1097-0258, DOI: 10.1002/sim.8086.

Mukherjee, S., Agarwal, D., Zhang, N. R., and Bhattacharya, B. B. (2022): "Distribution-Free Multisample Tests Based on Optimal Matchings With Applications to Single Cell Genomics", in: *Journal of the American Statistical Association* 117 (538), pp. 627–638, ISSN: 0162-1459, DOI: 10.1080/01621459.2020.1791131.

Mukhopadhyay, S. and Wang, K. (2020a): "A nonparametric approach to high-dimensional k-sample comparison problems", in: *Biometrika* 107 (3), pp. 555–572, ISSN: 0006-3444, DOI: 10.1093/biomet/asaa015.

Mukhopadhyay, S. and Wang, K. (2020b): *LPKsample: LP Nonparametric High Dimensional K-Sample Comparison*, R package version 2.1, URL: https://CRAN.R-project.org/package=LPKsample.

Ntoutsi, I., Kalousis, A., and Theodoridis, Y. (2008): "A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees", in: *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)*, Proceedings, Society for Industrial and Applied Mathematics, pp. 810–821, ISBN: 978-0-89871-654-2, DOI: 10.1137/1.9781611972788.73.

Pan, W., Tian, Y., Wang, X., and Zhang, H. (2018): "BALL DIVERGENCE: NONPARAMETRIC TWO SAMPLE TEST", in: *Annals of statistics* 46 (3), pp. 1109–1137, ISSN: 0090-5364, DOI: 10.1214/17-AOS1579.

Petrie, A. (2016): "Graph-theoretic multisample tests of equality in distribution for high dimensional data", in: *Computational Statistics & Data Analysis* 96, pp. 145–158, ISSN: 0167-9473, DOI: 10.1016/j.csda.2015.11.003.

R Core Team (2024): *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, URL: https://www.R-project.org/.

Rizzo, M. and Szekely, G. (2022): *energy: E-Statistics: Multivariate Inference via the Energy of Data*, R package version 1.7-10, URL: https://CRAN.R-project.org/package=energy.

Rizzo, M. L. and Székely, G. J. (2010): "DISCO analysis: A nonparametric extension of analysis of variance", in: *The Annals of Applied Statistics* 4 (2), pp. 1034–1055, ISSN: 1932-6157, 1941-7330, DOI: 10.1214/09-AOAS245.

Rosenbaum, P. R. (2005): "An Exact Distribution-Free Test Comparing Two Multivariate Distributions Based on Adjacency", in: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67 (4), pp. 515–530, ISSN: 1369-7412.

Roux de Bezieux, H. (2021): *Ecume: Equality of 2 (or k) Continuous Univariate and Multivariate Distributions*, R package version 0.9.1, URL: https://CRAN.R-project.org/package=Ecume.

Schilling, M. F. (1986): "Multivariate Two-Sample Tests Based on Nearest Neighbors", in: *Journal of the American Statistical Association* 81 (395), pp. 799–806, ISSN: 0162-1459, DOI: 10.2307/2289012.

Simon, H., Michel, L., and Naef, J. (2021): *hypoRF: Random Forest Two-Sample Tests*, R package version 1.0.0, URL: https://CRAN.R-project.org/package=hypoRF.

Song, H. and Chen, H. (2021): *kerTests: Generalized Kernel Two-Sample Tests*, R package version 0.1.3, URL: https://CRAN.R-project.org/package=kerTests.

Song, H. and Chen, H. (2022a): *gTestsMulti: New Graph-Based Multi-Sample Tests*, URL: https://CRAN.R-project.org/package=gTestsMulti (visited on 08/25/2022).

Song, H. and Chen, H. (2022b): *New graph-based multi-sample tests for high-dimensional and non-Euclidean data*, arXiv:2205.13787 [stat], DOI: 10.48550/arXiv.2205.13787, URL: http://arxiv.org/abs/2205.13787.

Song, H. and Chen, H. (2023a): "Generalized kernel two-sample tests", in: *Biometrika*, asad068, ISSN: 1464-3510, DOI: 10.1093/biomet/asad068.

Song, H. and Chen, H. (2023b): *gTestsMulti: New Graph-Based Multi-Sample Tests*, R package version 0.1.1, URL: https://CRAN.R-project.org/package=gTestsMulti.

Stolte, M. (2025): *R code for "An Empirical Comparison of Methods for Quantifying Similarity of Categorical Datasets"*, version v.1.1, DOI: 10.5281/zenodo.15074829.

Stolte, M., Kappenberg, F., Rahnenführer, J., and Bommert, A. (2024): "Methods for Quantifying Dataset Similarity: A Review, Taxonomy and Comparison", in: *Statistics Surveys* 18, pp. 163–298, ISSN: 1935-7516, DOI: 10.1214/24-SS149.

Stolte, M. and Sauer, L. (2025): *DataSimilarity: Quantifying Similarity of Datasets and Multivariate Two- And k-Sample Testing*, R package version 0.1.1, URL: https://CRAN.R-project.org/package=DataSimilarity.

Strobl, C. and Leisch, F. (2024): "Against the "one method fits all data sets" philosophy for comparison studies in methodological research", in: *Biometrical Journal* 66 (1), p. 2200104, ISSN: 1521-4036, DOI: 10.1002/bimj.202200104.

Székely, G. J. and Rizzo, M. L. (2004): "Testing for equal distributions in high dimension", in: *InterStat* 5 (16.10), pp. 1249–1272.

Székely, G. J. and Rizzo, M. L. (2017): "The Energy of Data", in: *Annual Review of Statistics and Its Application* 4 (1), pp. 447–479, DOI: 10.1146/annurev-statistics-060116-054026.

Tatti, N. (2007): "Distances between Data Sets Based on Summary Statistics.", in: *Journal of Machine Learning Research* 8 (1).

Wei, S., Lee, C., Wichers, L., and Marron, J. S. (2016): "Direction-Projection-Permutation for High-Dimensional Hypothesis Tests", in: *Journal of Computational and Graphical Statistics* 25 (2), pp. 549–569, ISSN: 1061-8600, DOI: 10.1080/10618600.2015.1027773.

Yu, K., Martin, R., Rothman, N., Zheng, T., and Lan, Q. (2007): "Two-sample Comparison Based on Prediction Error, with Applications to Candidate Gene Association Studies", in: *Annals of Human Genetics* 71 (1), pp. 107–118, ISSN: 1469-1809, DOI: 10.1111/j.1469-1809.2006.00306.x.

Zaremba, W. (2022): *B - test*, URL: https://github.com/wojzaremba/btest (visited on 09/02/2022).

Zaremba, W., Gretton, A., and Blaschko, M. (2013): "B-test: A Non-parametric, Low Variance Kernel Two-sample Test", in: *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc.

Zhang, J. and Chen, H. (2022): "Graph-Based Two-Sample Tests for Data with Repeated Observations", in: *Statistica Sinica* 32 (1), Publisher: Institute of Statistical Science, Academia Sinica, pp. 391–415, ISSN: 1017-0405.

Zhu, J., Pan, W., Zheng, W., and Wang, X. (2021): "Ball: An R Package for Detecting Distribution Difference and Association in Metric Spaces", in: *Journal of Statistical Software* 97 (6), pp. 1–31, DOI: 10.18637/jss.v097.i06.

Zhu, Y. and Chen, H. (2024): "Limiting distributions of graph-based test statistics on sparse and dense graphs", in: *Bernoulli* 30 (1), Publisher: Bernoulli Society for Mathematical Statistics and Probability, pp. 770–796, ISSN: 1350-7265, DOI: 10.3150/23-BEJ1616.

A Scenario Parameter Settings

Distribution	Alternative	Parameters
Bernoulli	Null: $F_j = \text{Bin}(0.5)$, $j = 1, 2$	–
	Unbalanced: $F_j = \text{Bin}(\pi^{(j)})$, $j = 1, 2$	$\pi^{(1)} = 0.5$, $\pi^{(2)} = \frac{1+\delta}{2+\delta}$, $\delta =$ 0.1,0.2,0.3,0.4,0.5,0.6, 0.7,0.8,0.9,1,1.5,2
Multinomial	Null: $F_j = \text{Mult}(1/5 \cdot \mathbf{1}_5)$, $j = 1, 2$	–
	Skewed: $F_j = Mult(\pi^{(j)})$	$\pi^{(1)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(2)} = w/\sum_{l=c}^{5} w_c$, $w_c = 1 + (c-1)\delta, c = 1, \ldots, 5$, $\delta =$ $0.1, 0.2, 0.3$,$0.4, 0.5, 1$
	1 up, 1 down: $F_j = \text{Mult}(\pi^{(j)})$, $j = 1, 2$	$\pi^{(1)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(2)} = w/5$, $w_1 = w_2 =$ $w_3 = 1$, $w_4 = 1 - \delta$, $w_5 = 1 + \delta$, $\delta =$ $0.1, 0.2, 0.3$,$0.4, 0.5, 0.6, 0.7, 0.8, 0.9$

Table 2: Scenarios for categorical data generation for $k = 2$. $X_i^{(j)} \overset{\text{iid}}{\sim} F_j$, $i = 1, \ldots, n_j$, $j = 1, 2$.

Distribution	Alternative	Parameters
Bernoulli	Null: $F_j = \text{Bin}(0.5)$, $j = 1,\ldots,4$	–
	Unbalanced: $F_j = \text{Bin}(\pi^{(j)})$, $j = 1,\ldots,4$	$\pi^{(1)} = \pi^{(2)} = \pi^{(3)} = 0.5$, $\pi^{(4)} = \frac{1+\delta}{2+\delta}$, $\delta = 0.1$,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1, 1.5,2
		$\pi^{(1)} = \pi^{(2)} = 0.5$, $\pi^{(3)} = \pi^{(4)} = \frac{1+\delta}{2+\delta}$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5, 0.6, 0.7,0.8,0.9, 1,1.5, 2
		$\pi^{(1)} = \pi^{(2)} = 0.5$, $\pi^{(3)} = \frac{1+\delta}{2+\delta}$, $\pi^{(4)} = \frac{1+2\delta}{2+2\delta}$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5,0.6, 0.7, 0.8, 0.9, 1, 1.5, 2
		$\pi^{(j)} = \frac{1+(j-1)\delta}{2+(j-1)\delta}$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2
Multinomial	Null: $F_j = \text{Mult}(1/5 \cdot \mathbf{1}_5)$, $j = 1,\ldots,4$	–
	Skewed: $F_j = \text{Mult}(\pi^{(j)})$	$\pi^{(1)} = \pi^{(2)} = \pi^{(3)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(4)} = w/\sum_{l=c}^{5} w_c$, $w_c = 1 + (c-1)\delta$, $c = 1,\ldots,5$, $\delta = 0.1, 0.2, 0.3$,0.4,0.5, 0.6, 0.7, 0.8, 0.9, 1
		$\pi^{(1)} = \pi^{(2)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(3)} = \pi^{(4)} = w/\sum_{l=c}^{5} w_c$, $w_c = 1 + (c-1)\delta$, $c = 1,\ldots,5$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
		$\pi^{(1)} = \pi^{(2)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(3)} = w/\sum_{l=c}^{5} w_c$, $w_c = 1 + (c-1)\delta$, $\pi^{(4)} = w/\sum_{l=c}^{5} w_c$, $w_c = 1 + (c-1)(\delta + 0.1)$, $c = 1,\ldots,5$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5,0.6, 0.7,0.8, 0.9, 1
		$\pi^{(1)} = 1/5 \cdot \mathbf{1}_5$,$\pi^{(j)} = w/\sum_{l=c}^{5} w_c$, $w_c = 1 + (c-1)(\delta + (j-1)0.1)$, $c = 1,\ldots,5$, $j = 2, 3, 4$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
	1 up, 1 down: $F_j = \text{Mult}(\pi^{(j)})$, $j = 1,\ldots,4$	$\pi^{(1)} = \pi^{(2)} = \pi^{(3)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(4)} = w/5$, $w_1 = w_2 = w_3 = 1$, $w_4 = 1 - \delta$, $w_5 = 1 + \delta$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5,0.6, 0.7, 0.8, 0.9
		$\pi^{(1)} = \pi^{(2)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(3)} = \pi^{(4)} = w/5$, $w_1 = w_2 = w_3 = 1$, $w_4 = 1 - \delta$, $w_5 = 1 + \delta$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5, 0.6, 0.7, 0.8, 0.9
		$\pi^{(1)} = \pi^{(2)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(3)} = w/5$, $w_1 = w_2 = w_3 = 1$, $w_4 = 1 - \delta$, $w_5 = 1 + \delta$, $\pi^{(4)} = w/5$, $w_1 = w_2 = w_3 = 1$, $w_4 = 1 - \delta - 0.1$, $w_5 = 1 + \delta + 0.1$, $\delta = 0.1, 0.2, 0.3$,0.4, 0.5, 0.6, 0.7, 0.8, 0.9

Distribution	Alternative	Parameters
		$\pi^{(1)} = 1/5 \cdot \mathbf{1}_5$, $\pi^{(j)} = w/5$, $w_1 = w_2 = w_3 = 1$, $w_4 = 1 - \delta - (j-1)0.1$, $w_5 = 1 + \delta + (j-1)0.1$, $\delta = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$

Table 3: Scenarios for categorical data generation for $k = 4$. $X_i^{(j)} \overset{\text{iid}}{\sim} F_j$, $i = 1, \ldots, n_j$, $j = 1, \ldots, 4$.

B Complete Method list

Method	(Sub)class	No. fulfilled	Implementation	Inclusion	y?	Num.?	Cat.?	$K > 2$?
KMD (Huang and Sen, 2024)	Kernel-based	15	R package KMD (Huang, 2022)	Implemented & $\geqslant$ 11 criteria	✗	✓	✗*	✓
Mukherjee et al. (2022)	Graph-based	13	R package multicross (Agarwal et al., 2020)	Implemented & $\geqslant$ 11 criteria	✗	✓	✓	✓
Biswas et al. (2014)	Graph-based	12	Own implementation	$\geqslant$ 11 criteria	✗	✓	✗	✓
Friedman and Rafsky (1979)	Graph-based	13	R package gTests (Chen and Zhang, 2017)	Implemented & $\geqslant$ 11 criteria	✗	✓	✓	✗
Cross-match test (Rosenbaum, 2005)	Graph-based	13	R package crossmatch (Heller et al., 2012)	Implemented & $\geqslant$ 11 criteria	✗	✓	✗	✗
Cramér test (Baringhaus and Franz, 2004)	Inter-point distances	11	R package cramer (Franz, 2019)	Implemented & $\geqslant$ 11 criteria	✗	✓	✗	✗
Energy statistic (Székely and Rizzo, 2017)	Inter-point distances	13	R package energy (Rizzo and Szekely, 2022)	Implemented & $\geqslant$ 11 criteria	✗	✓	✗	✓
Deb and Sen (2021)	Inter-point distances / Rank-based	12	Implementation based on R code for paper (`https://github.com/NabarunD/MultiDistFree`)	Implemented & $\geqslant$ 11 criteria	✗	✓	✗	✗
Ntoutsi et al. (2008)	Comparison of density functions	11	Own implementation	$\geqslant$ 11 criteria	✓	✓	✗	✗
Ganti et al. (1999)	Comparison of density functions	11	Own implementation	$\geqslant$ 11 criteria	✓	✓	✗*	✗
Hediger et al. (2021)	Binary classification	11	R package hypoRF (Simon et al., 2021)	Implemented & $\geqslant$ 11 criteria	✗	✓	✓	✗
Petrie (2016)	Graph-based	13	R package multicross (Agarwal et al., 2020)	Implemented & $\geqslant$ 11 criteria	✗	✓	✓*	✓

Method	(Sub)class	No. fulfilled	Implementation	Inclusion	y?	Num.?	Cat.?	$K > 2$?
Alvarez-Melis and Fusi (2020)	Distance/ similarity measure for datasets	11	own implementation based on python implementation (`https://github.com/microsoft/otdd`)	$\geqslant$ 11 criteria	✔	✔	✔	✖
Jeffreys divergence	Divergence	11	Own implementation	$\geqslant$ 11 criteria	✖	✔	✖	✖
Baringhaus and Franz (2010)	Inter-point distances	11	R package cramer (Franz, 2019)	Implemented & $\geqslant$ 11 criteria	✖	✔	✖	✖
Bahr (1996)	Inter-point distances	11	R package cramer (Franz, 2019)	Implemented & $\geqslant$ 11 criteria	✖	✔	✖	✖
Biswas and Ghosh (2014)	Inter-point distances	11	Own implementation	$\geqslant$ 11 criteria	✖	✔	✖	✖
Schilling (1986) and Henze (1988)	Graph-based	11	Own implementation	$\geqslant$ 11 criteria	✖	✔	✖	✖
Yu et al. (2007)	Binary classification	11	Own implementation using R package Ecume (Roux de Bezieux, 2021)	(Almost) implemented & $\geqslant$ 11 criteria	✖	✔	✔	✖
Wasserstein distance	Probability metric	9	R package Ecume (Roux de Bezieux, 2021)	Implemented	✖	✔	✖	✖
Chen and Friedman (2017)	Graph-based	11	R package gTests (Chen and Zhang, 2017)	Implemented & $\geqslant$ 11	✖	✔	✔	✖
Chen et al. (2018)	Graph-based	12	R package gTests (Chen and Zhang, 2017)	Implemented & $\geqslant$ 11	✖	✔	✔	✖
Ball divergence (Pan et al., 2018)	Testing	9	R package ball (Zhu et al., 2021)	Implemented	✖	✔	✖	✔
Song and Chen (2022b)	Graph-based	11	R package gTestsMulti (Song and Chen, 2023b)	Implemented & $\geqslant$ 11	✖	✔	✖	✔
DISCO (Rizzo and Székely, 2010)	Inter-point distances	10	R package energy (Rizzo and Szekely, 2022)	Implemented	✖	✔	✖	✔

Method	(Sub)class	No. fulfilled	Implementation	Inclusion	y?	Num.?	Cat.?	$K > 2$?
Li et al. (2022)	Comparison of characteristic functions	9	Own implementation	Best in (sub)class	✗	✓	✗	✗
Maximum Mean Discrepancy (MMD) (Gretton et al., 2006)	Kernel-based (MMD)	9	R packages kernlab (Karatzoglou et al., 2004) and Ecume (Roux de Bezieux, 2021)	Implemented	✗	✓	✗*	✗
Mukhopadhyay and Wang (2020a)	Graph-based	9	R package LPKsample (Mukhopadhyay and Wang, 2020b)	Implemented	✗	✓	✗*	✓
Chen et al. (2013)	Graph-based	9	R packages gTests (Chen and Zhang, 2017), gCat (Chen and Zhang, 2022)	Implemented	✗	✓	✓	✗
Block MMD (Zaremba et al., 2013)	Kernel-based (MMD)	8	R implementation based on matlab code for paper (`https://github.com/wojzaremba/btest`)	Implemented	✗	✓	✗*	✗
Song and Chen (2023a)	Kernel-based (MMD)	8	R package kerTests (Song and Chen, 2021)	Implemented	✗	✓	✗*	✗
Constrained Minimum Distance (Tatti, 2007)	Comparison based on summary statistics	8	Own implementation	Best in (sub)class	✗	✗	✓	✗
Biau and Gyorfi (2005)	Comparison of CDFs	8	Own implementation	Best in (sub)class	✗	✓	✗	✓
Classifier Two-Sample Test (Lopez-Paz and Oquab, 2017)	Binary classification	7	R package Ecume (Roux de Bezieux, 2021)	Implemented	✗	✓	✓	✓

Method	(Sub)class	No. fulfilled	Implementation	Inclusion	y?	Num.?	Cat.?	$K > 2$?
DiProPerm test (Wei et al., 2016)	Binary classification	5	Own implementation	Implemented	✗	✓	✗	✗

Table 4: Methods from theoretical comparison chosen for the empirical comparison. Methods applicable to two or more categorical datasets without target variables are highlighted. These are compared in the current study. No. fulfilled: Number of fulfilled criteria in theoretical comparison. y?: Can the method deal with a target variable in the dataset? Num.?: Is the method as implemented applicable to numeric data? Cat.?: Is the method as implemented applicable to categorical data? $K > 2$?: Is the method as implemented applicable to more than two datasets at a time? ✗*: Method is, in theory, applicable, but implementation is not. ✓*: Implementation is applicable, although this case is not described in the literature.

C Method Parameter Settings

The classifier for the classifier two-sample test (C2ST) of Lopez-Paz and Oquab (2017) is chosen as a K-nearest neighbor classifier (KNN) or as a multilayer perceptron network (neural net, NN) optimized by stochastic gradient descent. A K-nearest neighbor classifier, as well as a one-layer neural network, were compared in the simulations in the original publication. The neural network performed superior there. The KNN classifier is the default in the R implementation of the test. Note that Lopez-Paz and Oquab (2017) did not tune the hyperparameters of the classifiers, which might affect the performance. The implementation that is used here tunes the hyperparameters. The categorical data is dummy-coded in the present implementation. The KNN implementation uses the Euclidean distances of these dummy-coded data, which is equivalent to using the Hamming distance on the original categorical data.

For the random forest-based test of Hediger et al. (2021), both methods of estimating the prediction accuracy, namely taking the overall OOB accuracy or averaging the per-class OOB accuracies, are compared.

For the edge count tests, a similarity graph has to be chosen. Typical choices are the K-minimum spanning tree (MST) and the K-nearest neighbor (NN) graph. The choice of K is difficult, and there are no guidelines on how to choose this parameter. Many of the tests were originally proposed for $K = 1$, therefore, this value is used. Previous simulation studies show better empirical performance for higher values of K (Zhu and Chen, 2024). Often, $K = 5$ is used, so this is included as a second choice here.

For the max-type test of Zhang and Chen (2022), all recommended values for the parameter κ are compared, i.e. $\kappa \in \{1,1.14,1.31\}$.

For the graph-based tests for categorical data (Zhang and Chen, 2022), both strategies of averaging the results of all optimal graphs and using the union of all optimal graphs are compared.

For the CM distance, a feature function has to be chosen. There are no clear recommendations for this choice in the literature. The original paper uses the conjunction function for binary data. In a comparison of different feature functions on real data, the CM distance values for the independent means only and the independent means along with pairwise correlations were used, and were highly correlated. The other feature functions considered focused on frequent itemsets, which are not part of this study. Therefore, here, the independent means are used since they are computationally less demanding than calculating pairwise correlations in addition.

D Comparison of PESR and Asymptotic Power

Figure 25 to 34 show the simulated power of the asymptotic test and PESR curves for methods where an asymptotic test is defined. The power of the asymptotic test is estimated by the proportion of simulation repetitions in which the asymptotic p value is smaller than 0.05. As an example, the two-sample case with binary data and balanced sample sizes is selected. The unbalanced sample size case is shown additionally for methods for which it makes a difference in the comparison of power and PESR. The C2ST (NN) is selected as a representative for C2ST (KNN) and YMRZL, and the FR (1NN, u) was selected as a representative for the edge count tests since the methods within these groups acted similarly.

For the C2ST (Figures 25 and 26) and YMRZL, there are some differences between the

asymptotic power and the PESR. In most cases, the PESR is lower than the simulated asymptotic power. These differences vanish for increasing N where the asymptotic test is expected to be most accurate. For unbalanced sample sizes, however, there are still some differences for the largest N.

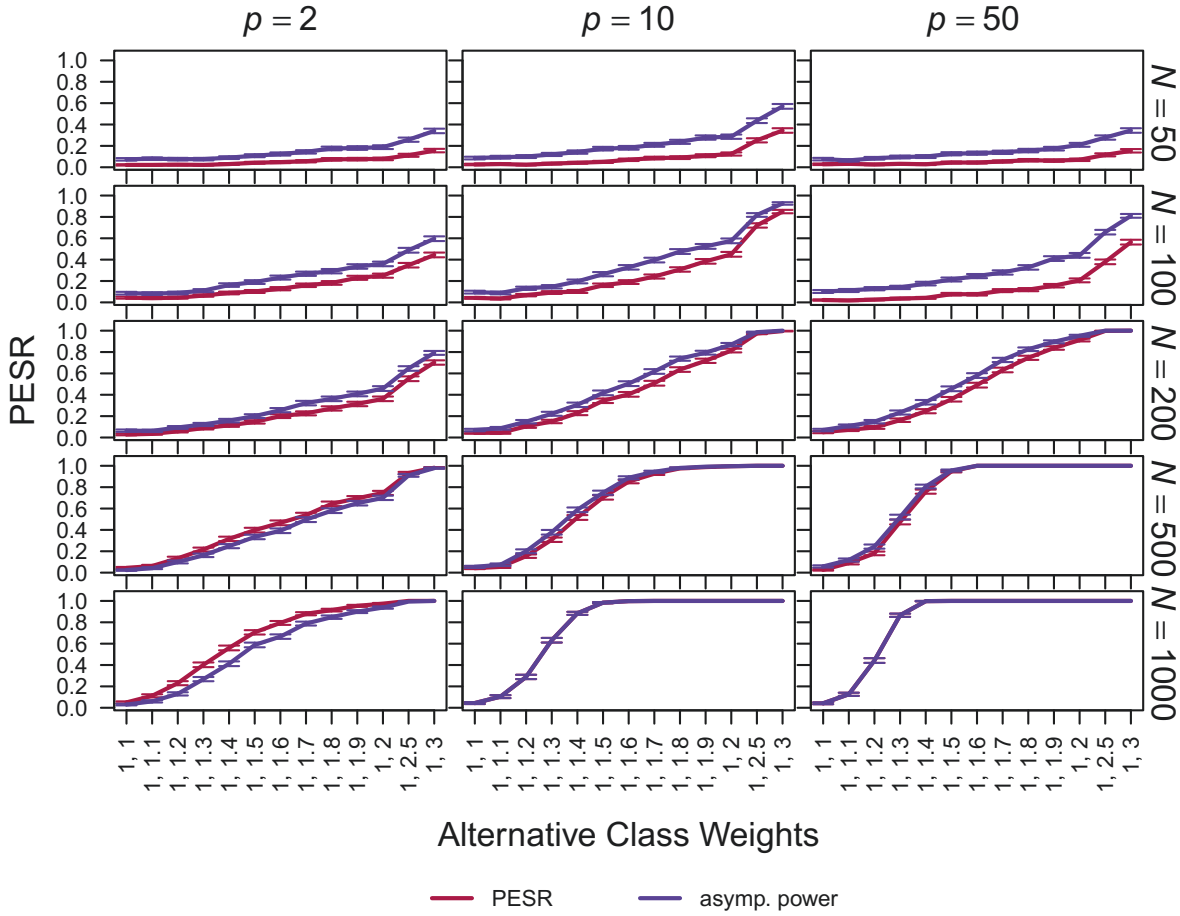


Figure 25: Comparison of the PESR to the asymptotic power for two datasets of the same sample sizes with binary variables for **C2ST (NN)**. The class weights give the unnormalized probabilities $(1, 1+\delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

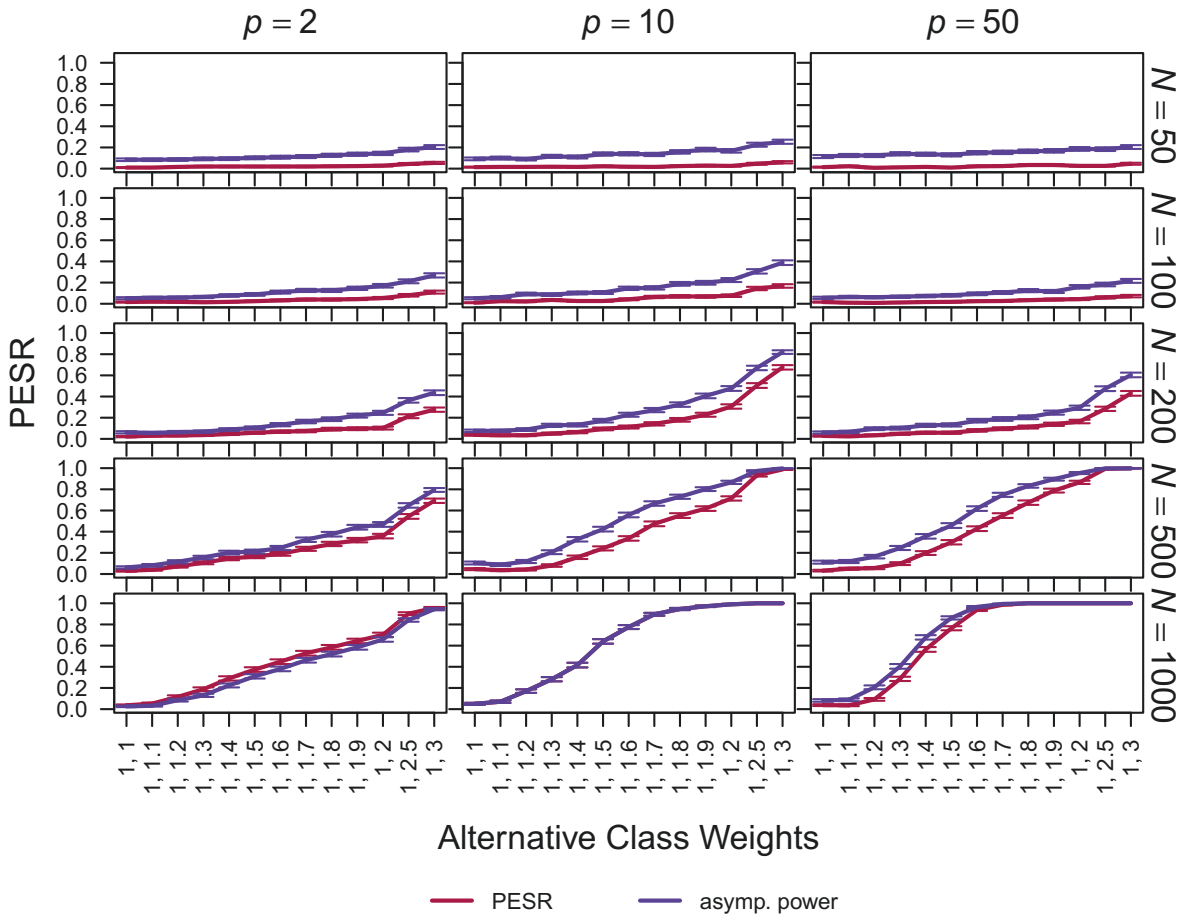


Figure 26: Comparison of the PESR to the asymptotic power for two datasets of unequal sample sizes with binary variables for **C2ST (NN)**. The class weights give the unnormalized probabilities $(1, 1+\delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

For the edge count tests, the power and PESR curves are very similar with some small differences for low N (see e.g. Figure 27).

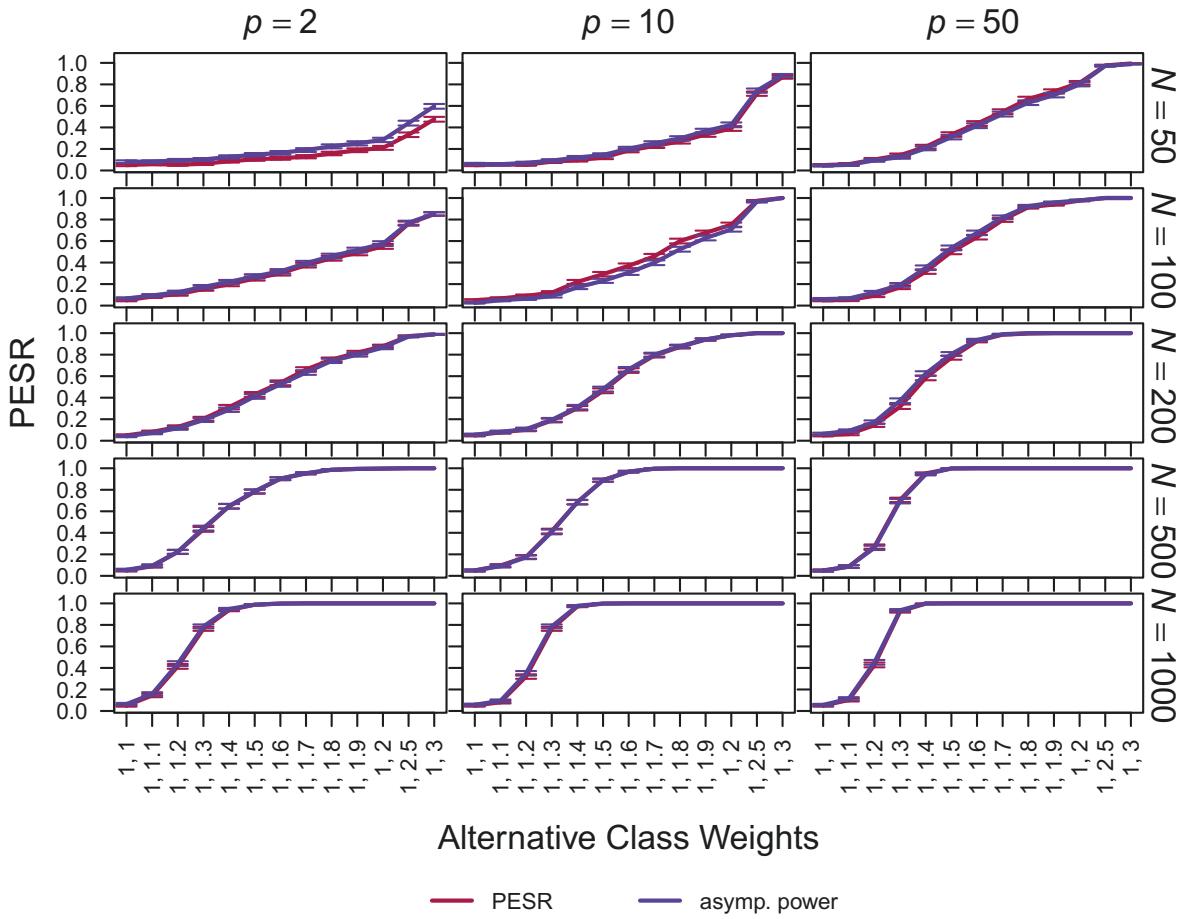


Figure 27: Comparison of the PESR to the asymptotic power for two datasets of the same sample sizes with binary variables for **FR (1NN, u)**. The class weights give the unnormalized probabilities $(1, 1+\delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

For the HMN and balanced sample sizes, the PESR and power curves are also typically close with some differences for small to moderate N (Figure 28 and 30). For unbalanced sample sizes and the HMN (overall OOB), the PESR is almost constantly zero, while the simulated power is constantly equal to one (Figure 29). The reason for this is that the test statistics in the simulation are high (even under the null) but almost constant. Therefore, the test rejects in all cases, but the PESR does not find a difference between the distributions under the alternatives and the null distribution. For the HMN (per class OOB), on the other hand, the PESR and power are close in the unbalanced sample size and both are mostly zero (Figure 31). The test statistic here is almost constant but close to zero as expected under the null, so the test does not reject, and the PESR also does not find differences between the test statistics under the null and the alternatives.

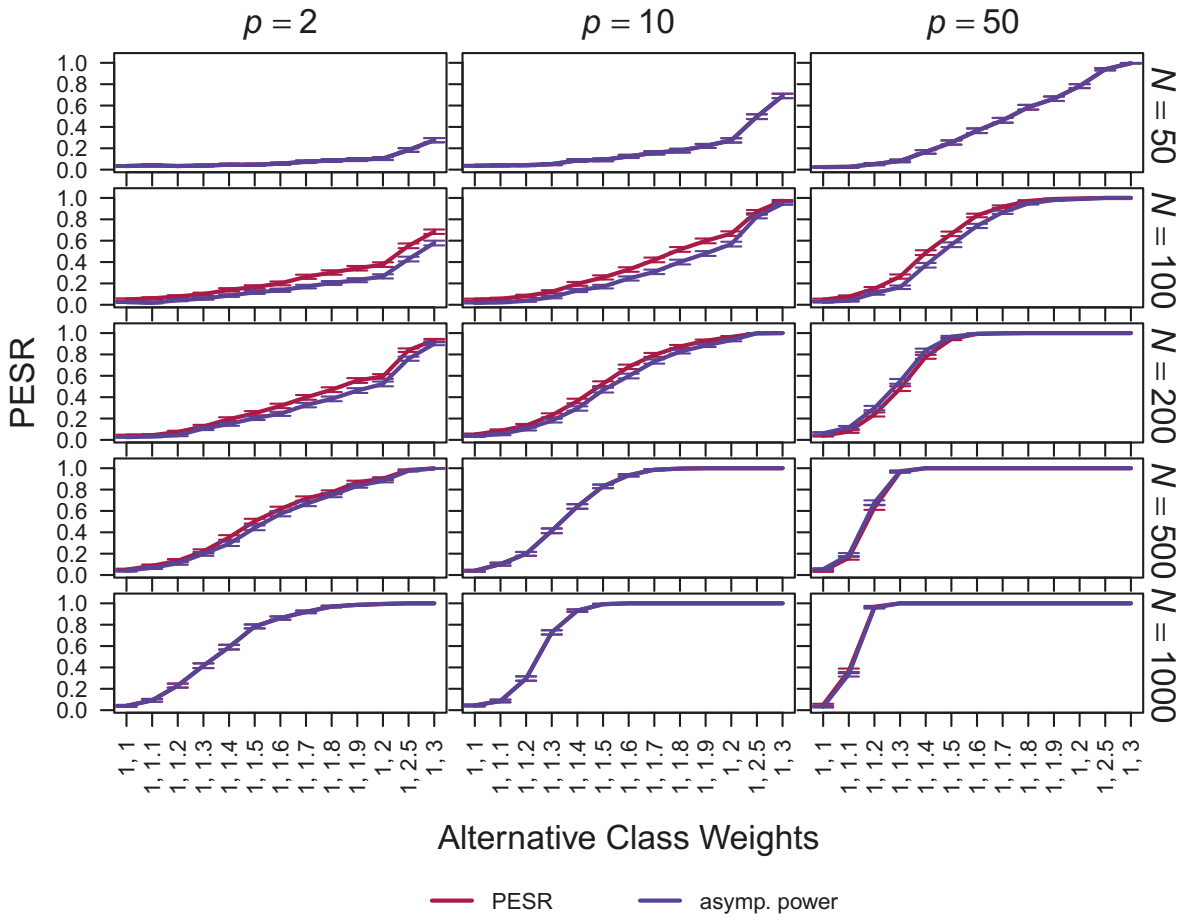


Figure 28: Comparison of the PESR to the asymptotic power for two datasets of the same sample sizes with binary variables for **HMN (overall OOB)**. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

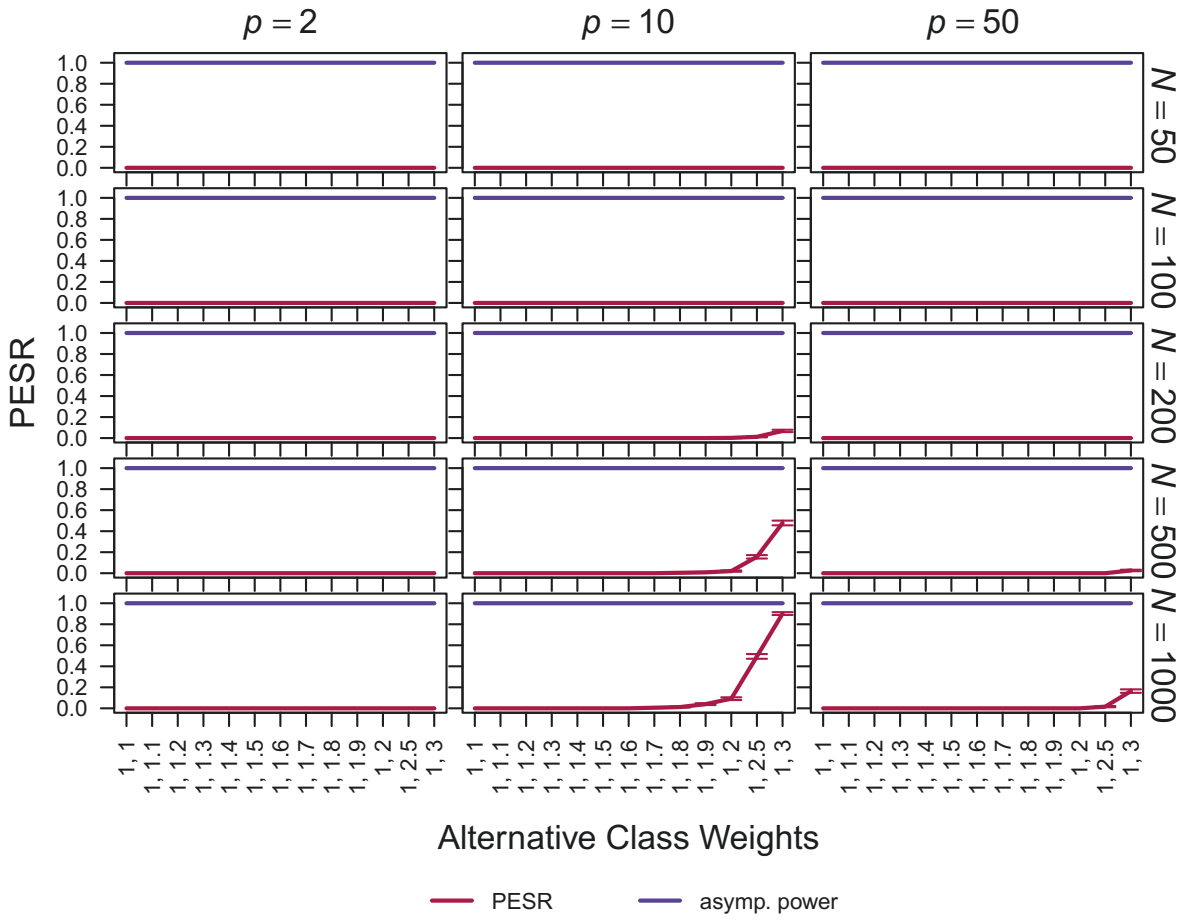


Figure 29: Comparison of the PESR to the asymptotic power for two datasets of unequal sample sizes with binary variables for **HMN (overall OOB)**. The class weights give the unnormalized probabilities $(1, 1+\delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

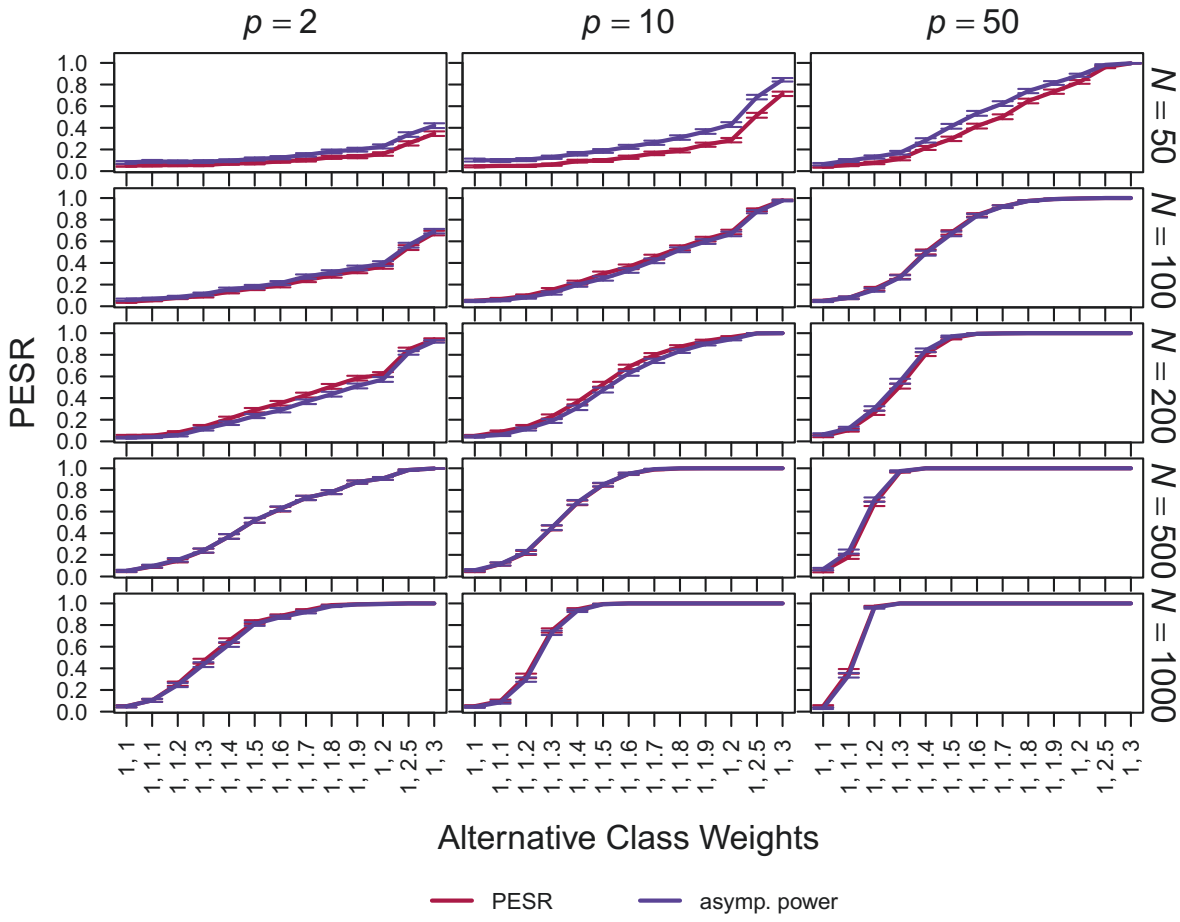


Figure 30: Comparison of the PESR to the asymptotic power for two datasets of the same sample sizes with binary variables for **HMN (per class OOB)**. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

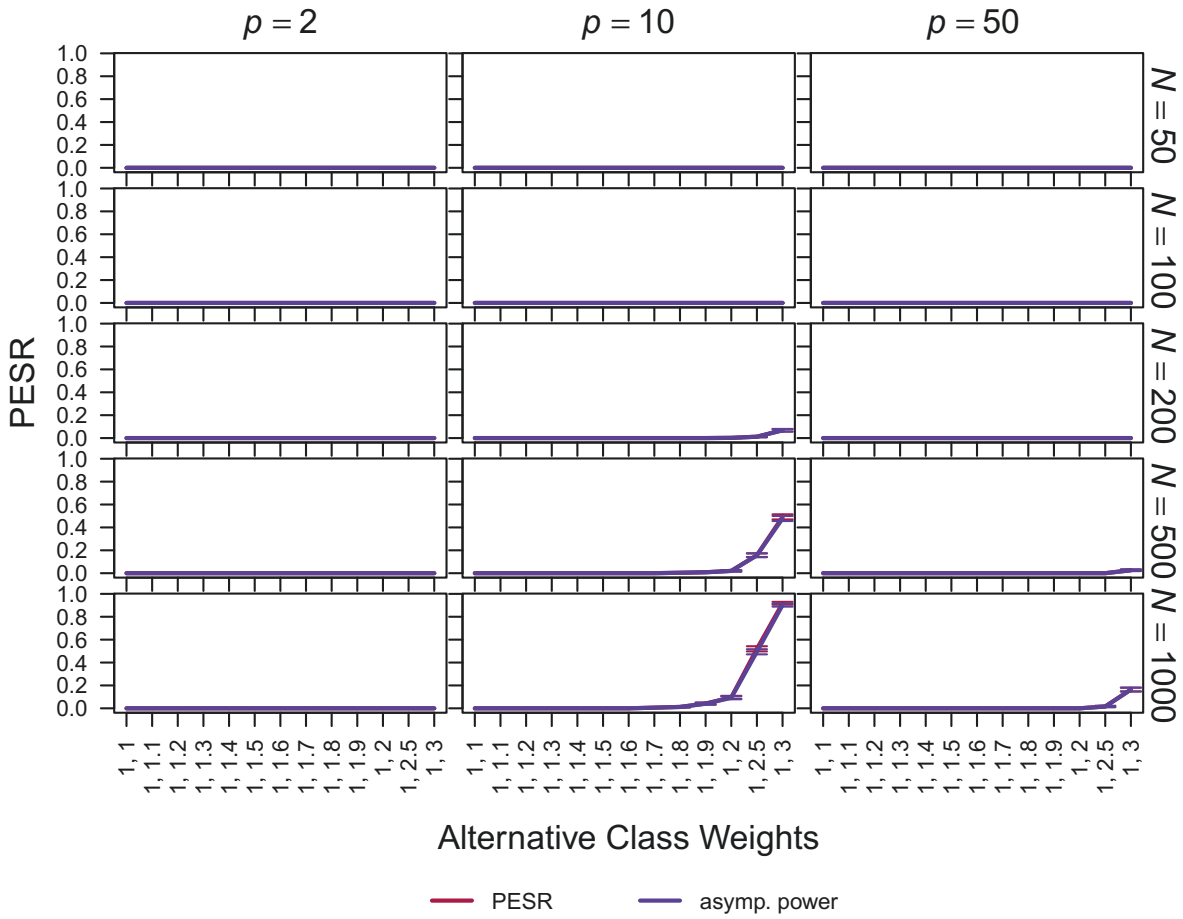


Figure 31: Comparison of the PESR to the asymptotic power for two datasets of unequal sample sizes with binary variables for **HMN (per class OOB)**. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

Similar observations can be made for the MMCM and Petrie's method for $p = 2$, where very large test statistic values are observed regardless of whether there are differences between the distributions or not (Figure 32 to 34). For Petrie's method in the binary case, the PESR is increasing while the power is constantly zero (Figure 33). The reason for this is that the observed test statistic values are small with low variance but slightly decreasing with the alternatives. Thus, the test does not find any differences, but the PESR picks up a slight decrease. For $p > 2$, there are some differences between the PESR and the power, even for higher N. Typically, the power is higher than the PESR. For $p = 10$, the tests are even liberal.

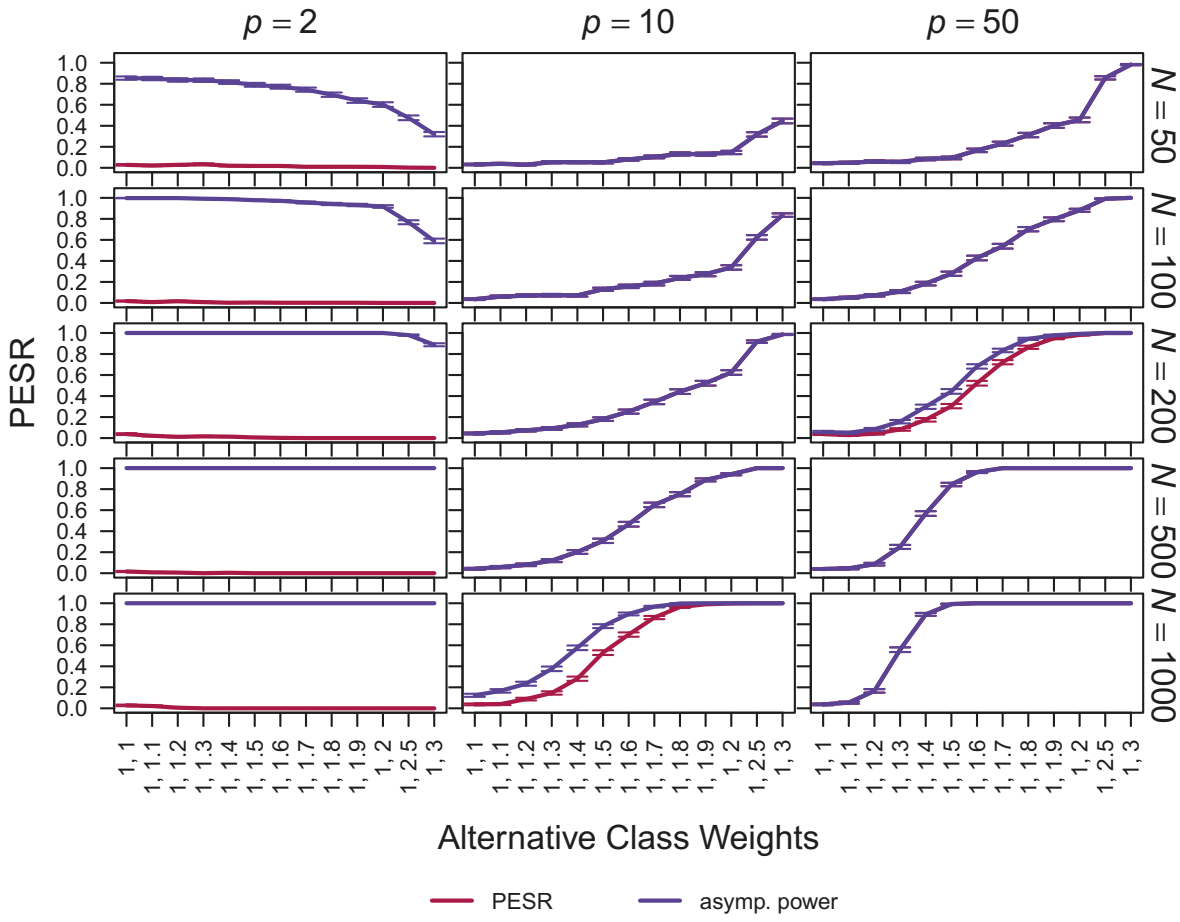


Figure 32: Comparison of the PESR to the asymptotic power for two datasets of the same sample sizes with binary variables for **MMCM**. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

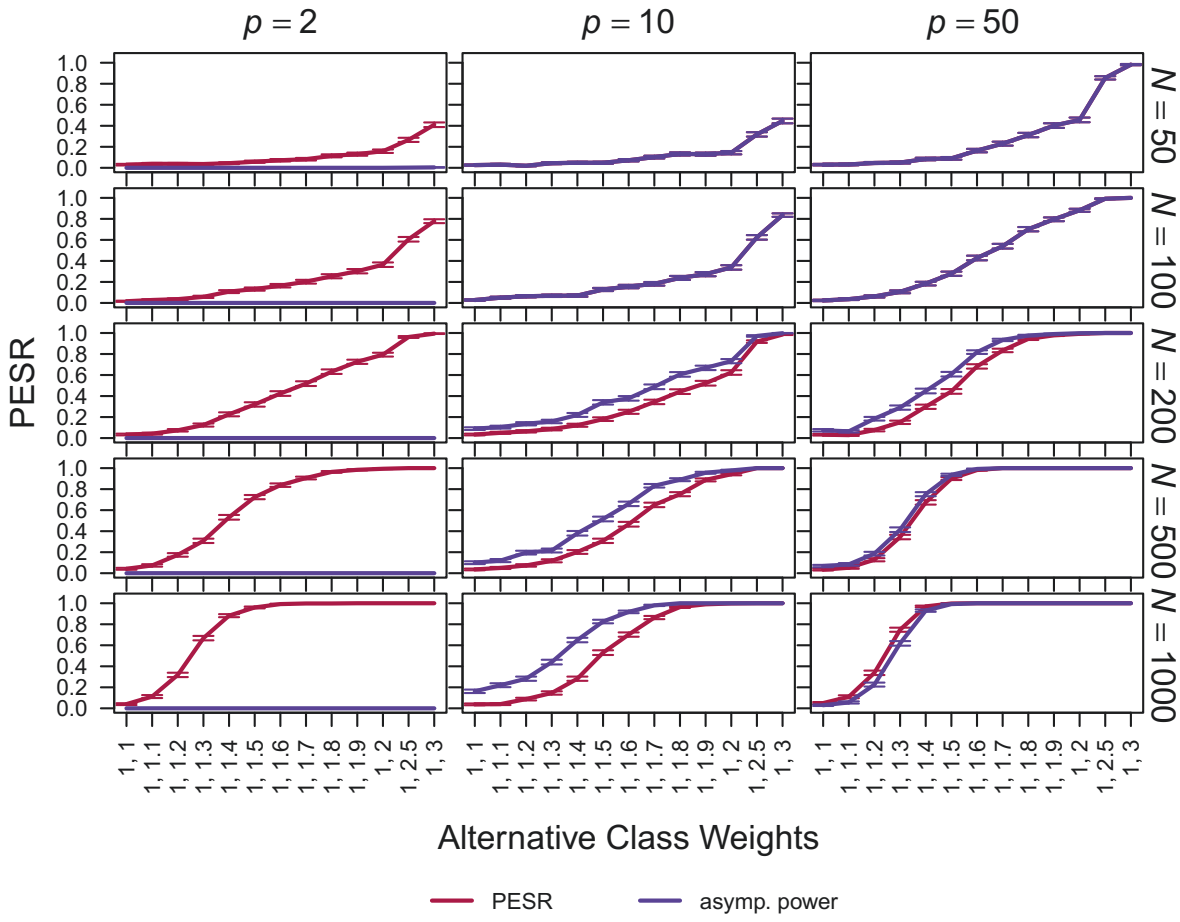


Figure 33: Comparison of the PESR to the asymptotic power for two datasets of the same sample sizes with binary variables for **Petrie**. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

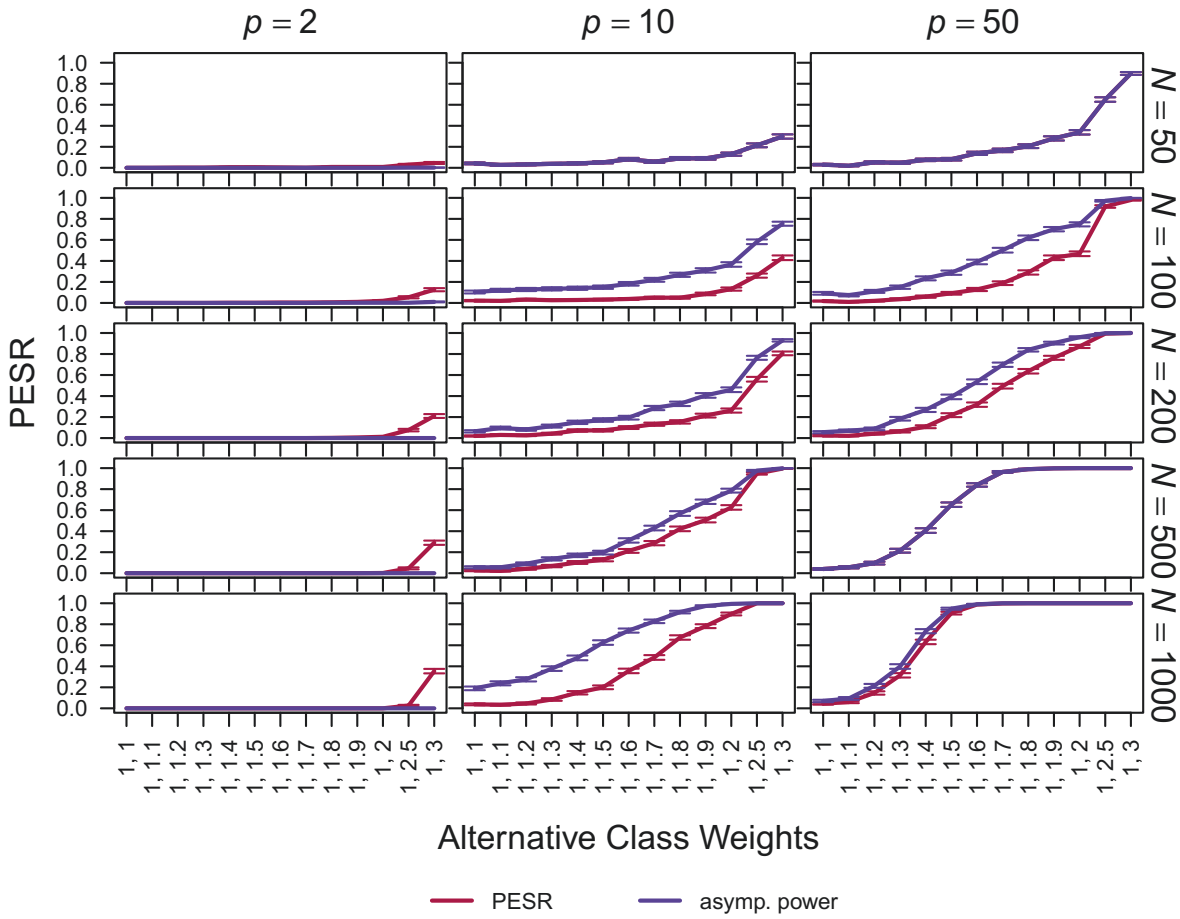


Figure 34: Comparison of the PESR to the asymptotic power for two datasets of unequal sample sizes with binary variables for **Petrie**. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

E Graph-Based Methods for Low-Dimensional Categorical Data

In all settings, clear problems with the performance of MMCM and the method of Petrie (2016) were visible for $p = 2$. In the following, the reasons for this issue are investigated. Both methods use the optimal non-bipartite matching as a basis where, based on the Euclidean distances, pairs of observations are matched such that the sum of the edge lengths, i.e. the sum of the Euclidean distances of matched observations, is minimized. With categorical data and few variables, there are only a few possible observations, which leads to ties in the distance matrix. For $p = 2$ variables, only 2^c combinations are possible, where c denotes the number of categories, which is either 2 or 5 here. When calculating the Euclidean distances as is done for the two above-mentioned methods, there are even fewer possible distance values. For $c = 2$, the possible distances are only $0, 1, \sqrt{2}$. Therefore, when calculating the optimal non-bipartite matching that is used in

both methods, there are many optimal solutions. The implemented matching algorithm goes through the observations in the order of the samples and starts looking for a match in the reverse order. Therefore, for the two-sample case with ties, the observations from the first dataset mostly get matched with observations from the second sample. In the four-sample case, they mostly get matched with observations from the fourth sample, then from the third, then from the second, and only lastly from the first. The observations from the second sample mostly get matched with the third sample, as most observations from the fourth sample already had a perfect match after going through the first sample. Therefore, in the two-sample case, the cross counts of the first and second datasets are very high, while the numbers of counts within the samples are very low. In the multi-sample case, the cross counts of the first and last datasets and the second and third datasets are very high; the cross counts of the first and third datasets and the second and fourth datasets are already lower than expected under the null, and the within-sample counts are typically very low. Thus, even under the null, the observed cross counts deviate largely from the expected values, resulting in very large test statistic values. When changing the class weights of the last sample, fewer observations from the first sample get matched with observations of the last sample, which leads to a redistribution of the cross matches that is more evenly than under the null, so counterintuitively, the test statistic values often decrease for increasing differences in the class weights, as can be seen for the MMCM test statistic in Figure 35 for $k = 2$ and in Figure 36 for $k = 4$. For Petrie's test, this decrease in the binary and balanced setting results in good PESR values as decreasing values are what would be expected under the alternative, and the PESR does not detect the too-high values in the null situation, which is a shortcoming of this approach. However, for $p = 2$, both asymptotic tests would reject the null for all considered class weights in almost all simulation repetitions. For more than two categories, the problem is less severe but still present.

Randomly permuting the order of the observations before calculating the matching could prevent the peculiar matching due to the ties, but it introduces randomness in the calculation of the test statistic value.

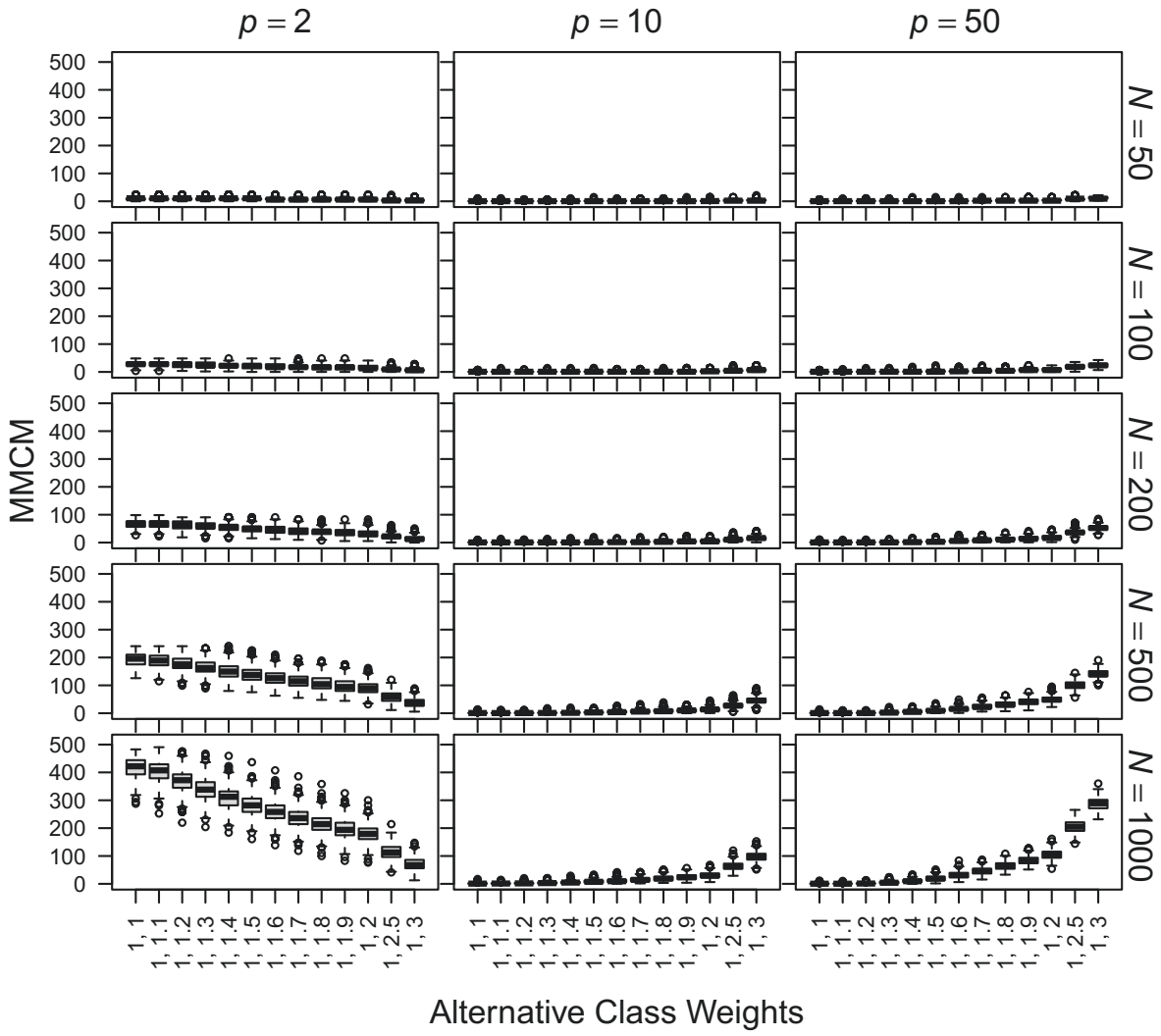


Figure 35: MMCM test statistic values for $k = 2$ datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

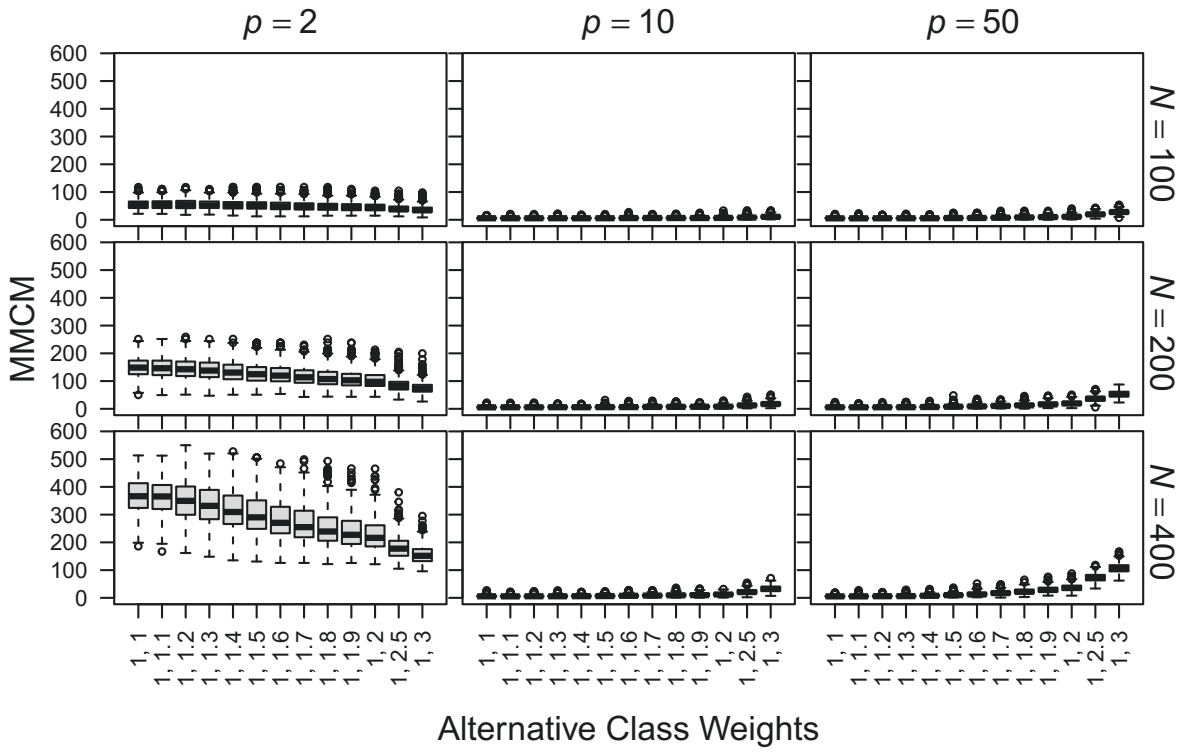


Figure 36: MMCM test statistic values for $k = 4$ datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the fourth dataset. The weights in the first to third dataset are always set to $(1, 1)$. Error bars indicate Monte Carlo standard errors.

In the case with two categories and two variables, the CF and ZC using 1NN, "a" break down with test statistic values mostly simulated as `NaN` (not a number) or zeros. The (undirected) 1NN graph on the distinct values that results in this case is displayed in Figure 37. The CF and ZC both use the standardized difference between the edge counts within the first and second samples. The variance under the null hypothesis used for standardization can be calculated as

$$\mathbb{V}\mathrm{ar}_{H_0}(R_{d,a}) = \frac{4n_1n_2}{N(N-1)}\left[\sum_{u=1}^{c}\frac{\left(\left|\mathcal{E}_u^{C_0}\right|-2\right)^2}{4m_u} - \frac{\left(|C_0|-4\right)^2}{N}\right],$$

where u takes on all distinct values. $\left|\mathcal{E}_u^{C_0}\right|$ denotes the number of edges of node u in the graph on the distinct values, which is 2 here, see Figure 37. $|C_0|$ denotes the number of edges in the graph on the distinct values, which is 4 for the 1NN graph. Thus, $\mathbb{V}\mathrm{ar}(R_{d,a}) = 0$. However, the variance is calculated by a different but equivalent expression in the implementation and is sometimes numerically not exactly equal to zero but some very small positive number in which case the resulting test statistic value is equal to zero as the observed and expected $R_{d,a}$ are also equal to zero here with similar arguments.

For the 5NN, "u", a similar problem is given with $p = 2$. In the extreme case for $c = 2$, the 5NN is always the full graph as for each distinct value, there are only 3 other distinct values such that it is connected with all three of those. In the $c = 5$ case, there are $2^5 = 32$ distinct values, so it is also very likely that the union of all optimal 5NN graphs is the full graph. In the case where the 5NN, "u" is equal to the full graph, the variance of each test

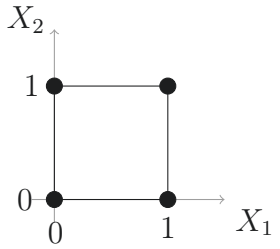


Figure 37: Undirected 1NN graph on the distinct values for two binary variables.

statistic is equal to zero. Since the parts of the test statistics of FR, CCS, CF, and ZC that are standardized can be expressed in terms of linear combinations of the edge counts within the first and second sample, $R_{1,u}$ and $R_{2,u}$, respectively, in the union graph $\bar{G}$ (Zhang and Chen, 2022), it suffices to show $\mathbb{V}\text{ar}_{H_0}(R_{1,u}) = \mathbb{V}\text{ar}_{H_0}(R_{2,u}) = \mathbb{C}\text{ov}_{H_0}(R_{1,u}, R_{2,u}) = 0$ to show that the null variance of these statistics is equal to zero. For the (undirected) full graph it holds that the number of edges is given by $|\bar{G}| = \frac{N(N-1)}{2}$ and the degree of each node i is given by $|\mathcal{E}_i^{\bar{G}}| = N-1, i = 1, \ldots, N$. Inserting these quantities in the expressions for the variances and covariances given in the supplemental material of Zhang and Chen (2022) yields

$$
\begin{aligned}
\mathbb{V}\text{ar}_{H_0}(R_{1,u}) =& \left[\frac{n_1(n_1-1)}{N(N-1)} - \frac{n_1(n_1-1)(n_1-2)(n_1-3)}{N(N-1)(N-2)(N-3)}\right]\frac{N(N-1)}{2} \\
&+ \left[\frac{n_1(n_1-1)(n_1-2)}{N(N-1)(N-2)} - \frac{n_1(n_1-1)(n_1-2)(n_1-3)}{N(N-1)(N-2)(N-3)}\right] N(N-1)(N-2) \\
&+ \left[\frac{n_1(n_1-1)(n_1-2)(n_1-3)}{N(N-1)(N-2)(N-3)} - \left(\frac{n_1(n_1-1)}{N(N-1)}\right)^2\right]\left(\frac{N(N-1)}{2}\right)^2 \\
=& \frac{n_1(n_1-1)}{2} - \frac{n_1(n_1-1)(n_1-2)(n_1-3)}{2(N-2)(N-3)} \\
&+ n_1(n_1-1)(n_1-2) - \frac{n_1(n_1-1)(n_1-2)(n_1-3)}{N-3} \\
&+ \frac{n_1(n_1-1)(n_1-2)(n_1-3)}{4(N-2)(N-3)} N(N-1) - \left(\frac{n_1(n_1-1)}{2}\right)^2 \\
=& \frac{n_1(n_1-1)}{2} - \left(\frac{n_1(n_1-1)}{2}\right)^2 + n_1(n_1-1)(n_1-2) \\
&- \frac{2+4(N-2)-N(N-1)}{4(N-2)(N-3)}\left[n_1(n_1-1)(n_1-2)(n_1-3)\right] \\
=& n_1(n_1-1)\left[\frac{1}{2} - \frac{1}{4}n_1(n_1-1) + n_1 - 2\right] \\
&- \frac{2+4N-8-N^2+N}{4(N^2-5N+6)}\left[n_1(n_1-1)(n_1-2)(n_1-3)\right] \\
=& n_1(n_1-1)\left[-\frac{1}{4}n_1^2 + \frac{5}{4}n_1 - \frac{3}{2}\right] + \frac{1}{4}\left[n_1(n_1-1)(n_1-2)(n_1-3)\right] \\
=& \frac{n_1(n_1-1)}{4}\left[-n_1^2 + 5n_1 - 6\right] + \frac{n_1(n_1-1)}{4}\left[n_1^2 - 5n_1 + 6\right] \\
=& 0.
\end{aligned}
$$

Analogously, $\mathbb{V}\mathrm{ar}_{H_0}(R_{1,u}) = 0$ can be shown by replacing the sample size n_1 of the first sample with the sample size n_2 of the second sample in the above calculation.
For the covariance, it holds

$$\begin{aligned}
\mathbb{C}\mathrm{ov}_{H_0}(R_{1,u}, R_{2,u}) =& \frac{n_1(n_1-1)n_2(n_2-1)}{N(N-1)(N-2)(N-3)} \\
& \cdot \left[\left(\frac{N(N-1)}{2} \right)^2 - \frac{N(N-1)}{2} - N(N-1)(N-2) \right] \\
& - \frac{n_1(n_1-1)}{N(N-1)} \frac{n_2(n_2-1)}{N(N-1)} \left(\frac{N(N-1)}{2} \right)^2 \\
=& \frac{n_1(n_1-1)n_2(n_2-1)}{4(N-2)(N-3)} N(N-1) - \frac{n_1(n_1-1)n_2(n_2-1)}{2(N-2)(N-3)} \\
& - \frac{n_1(n_1-1)n_2(n_2-1)}{(N-3)} - \frac{n_1(n_1-1)n_2(n_2-1)}{4} \\
=& n_1(n_1-1)n_2(n_2-1) \left[\frac{N(N-1)-2-4(N-2)}{4(N-2)(N-3)} - \frac{1}{4} \right] \\
=& n_1(n_1-1)n_2(n_2-1) \left[\frac{N^2-N-2-4N+8}{4(N^2-5N+6)} - \frac{1}{4} \right] \\
=& n_1(n_1-1)n_2(n_2-1) \left[\frac{1}{4} - \frac{1}{4} \right] \\
=& 0.
\end{aligned}$$

Therefore, all variances used for standardization in the calculation of the edge count test statistics are analytically equal to zero, which results in the `NaN` results. Numerically, the calculated variances are sometimes not exactly equal to zero but always have a very small absolute value. If this very small value is positive, the resulting test statistics are equal to zero. This happens especially for unequal sample sizes and small N. Sometimes, the numerically calculated variance estimate is even slightly negative, which also results in `NaN` values of the test statistic.

F Additional Figures

F.1 Pre-Selection of Methods for $k = 2$

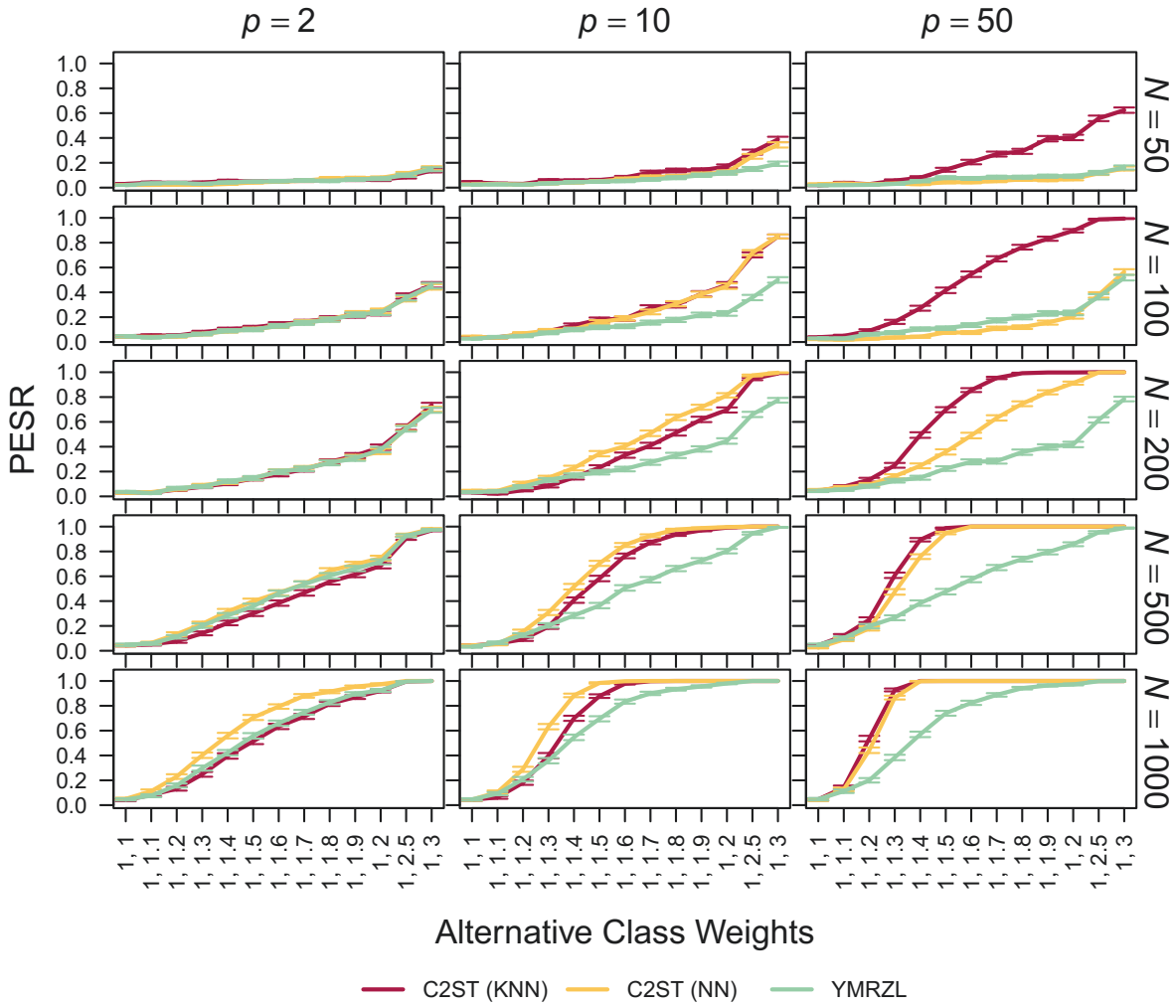


Figure 38: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

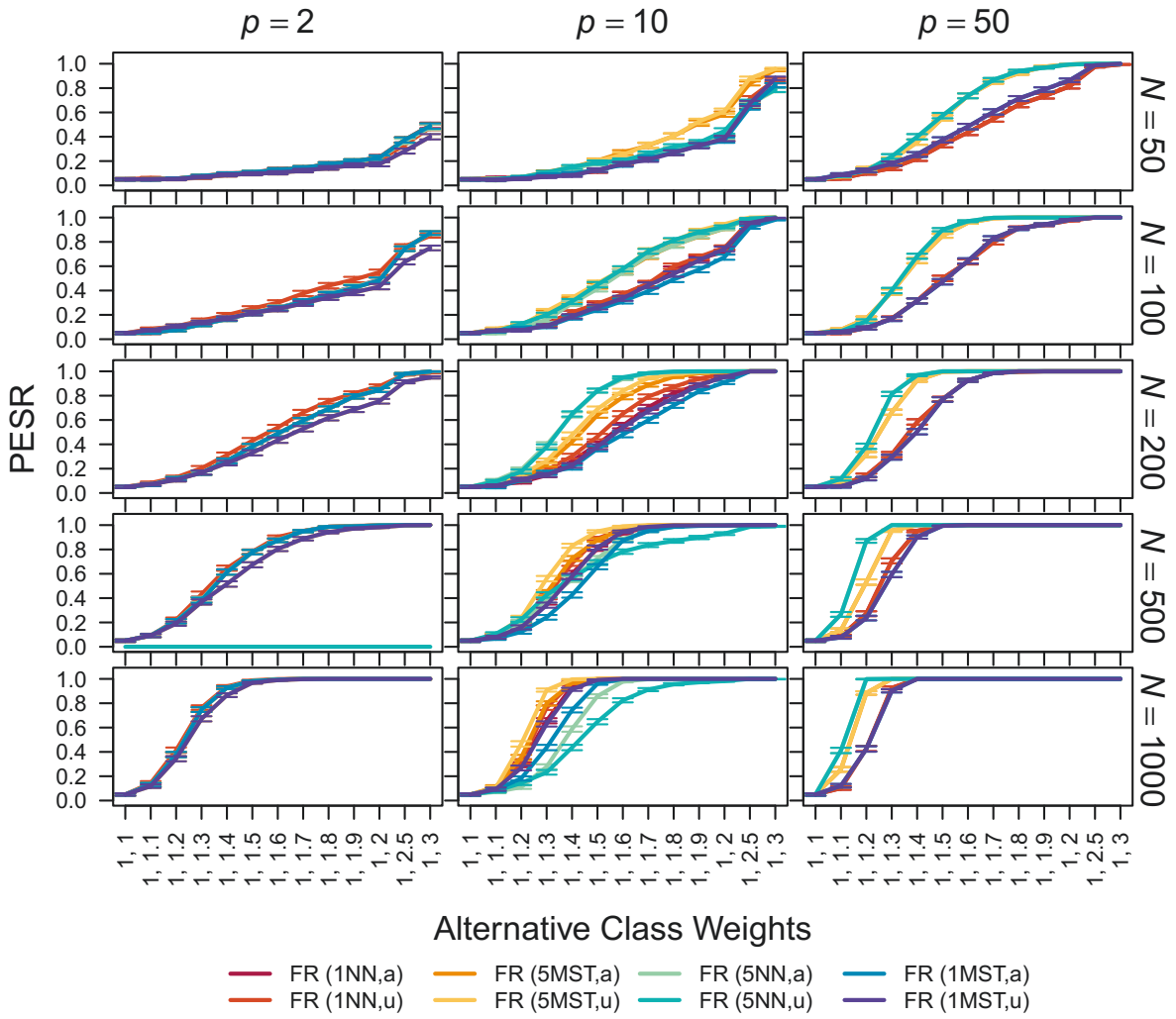


Figure 39: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1+\delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1+\delta)$. Error bars indicate Monte Carlo standard errors.

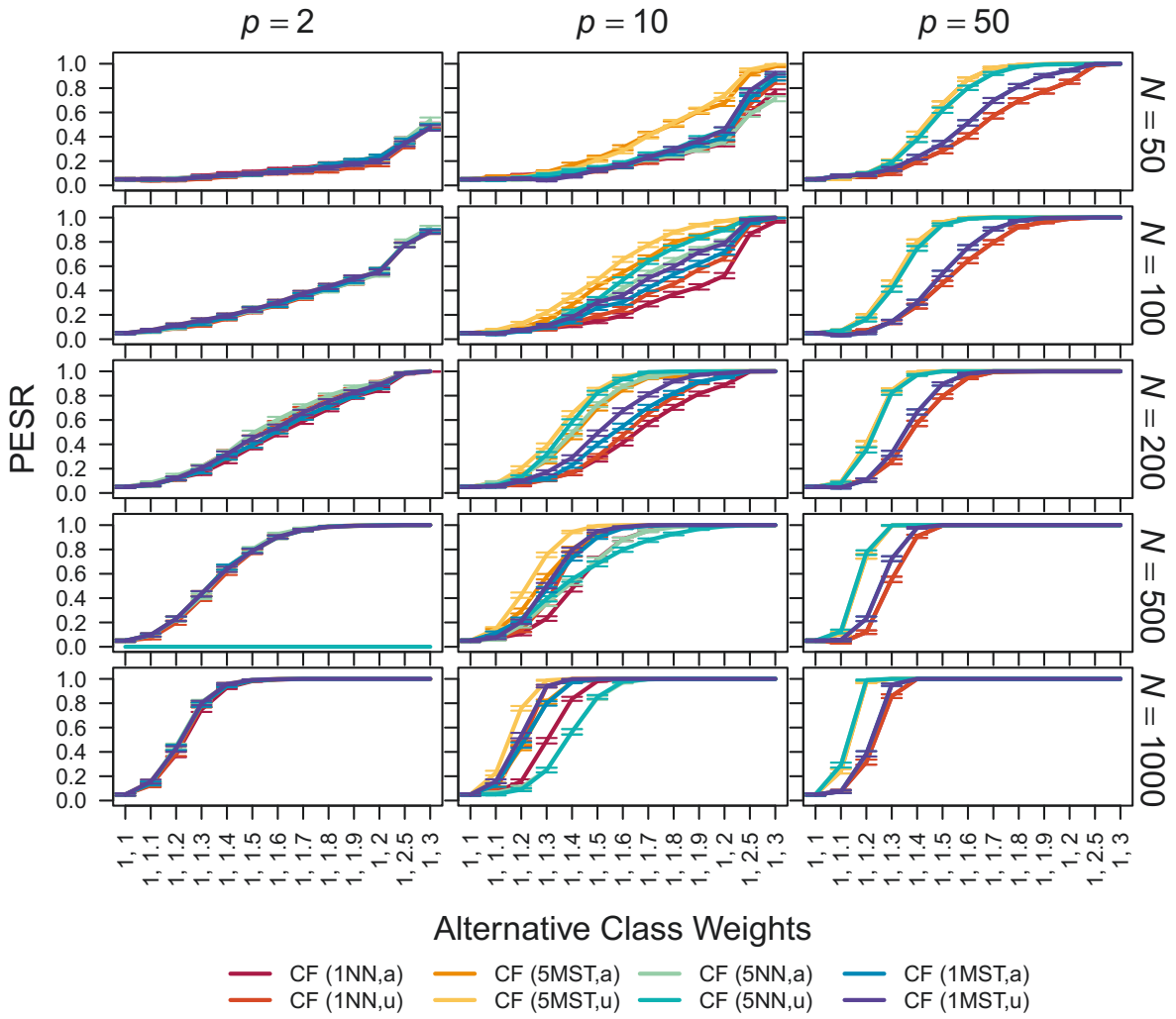


Figure 40: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1+\delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1+\delta)$. Error bars indicate Monte Carlo standard errors.

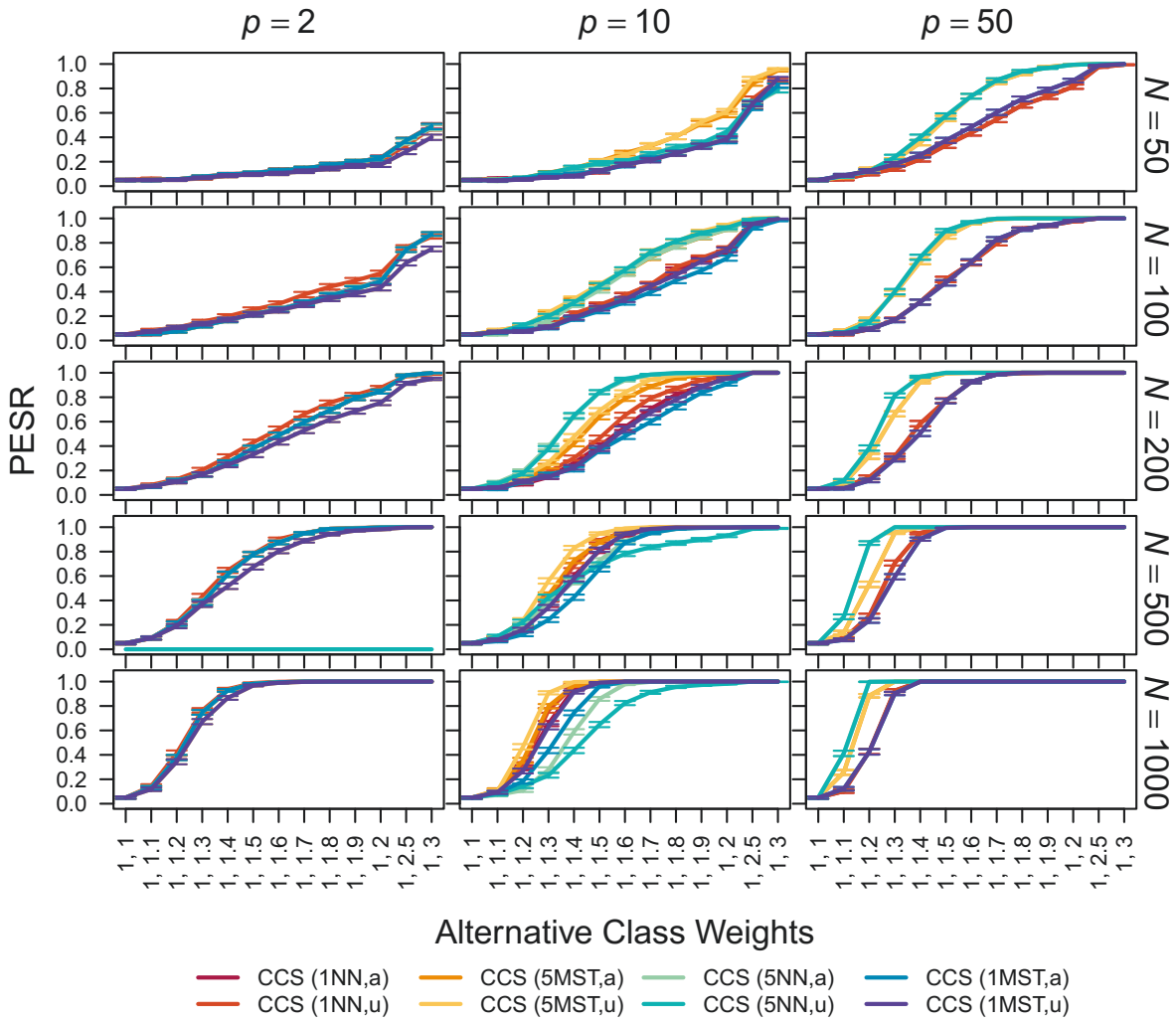


Figure 41: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

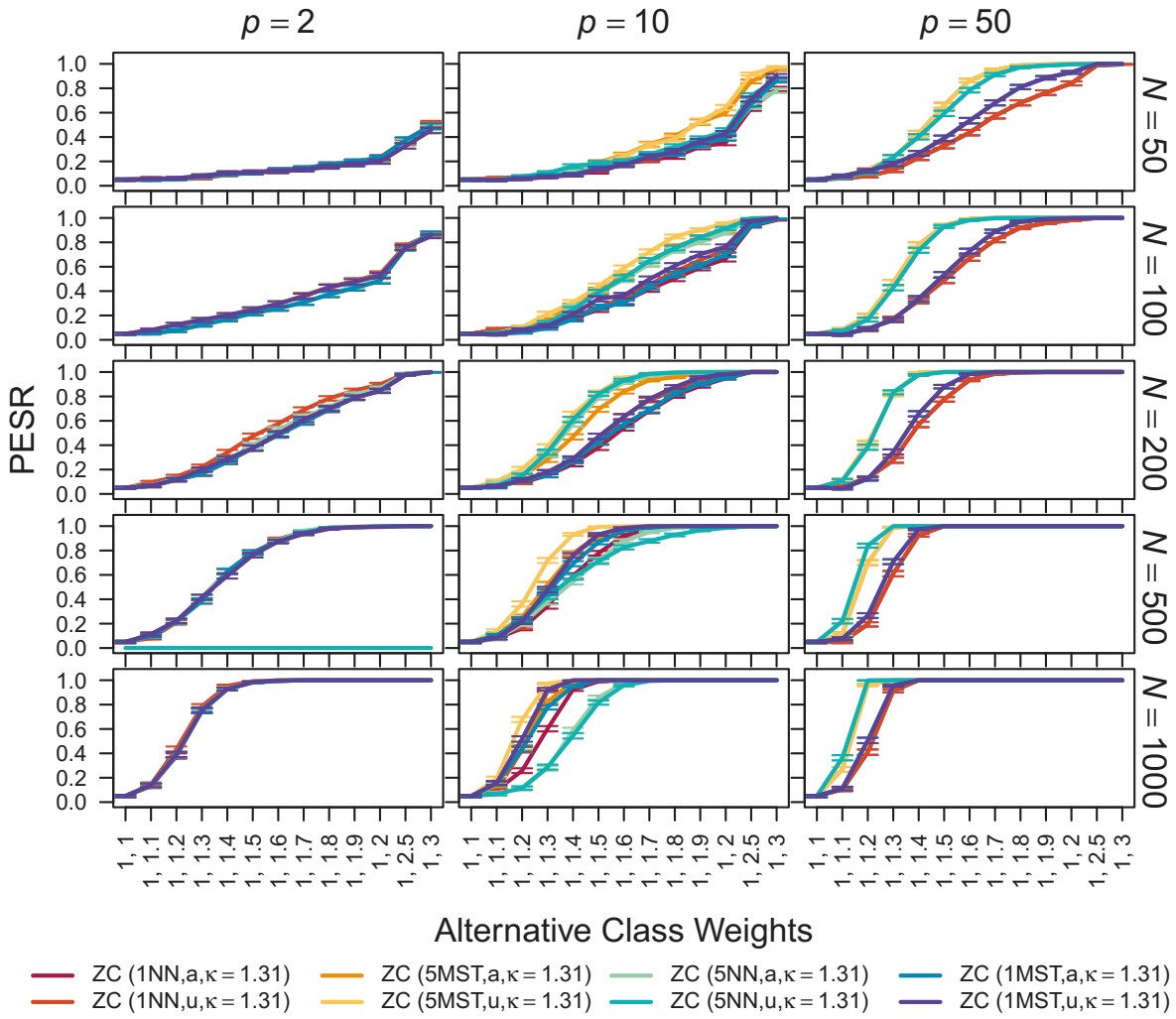


Figure 42: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

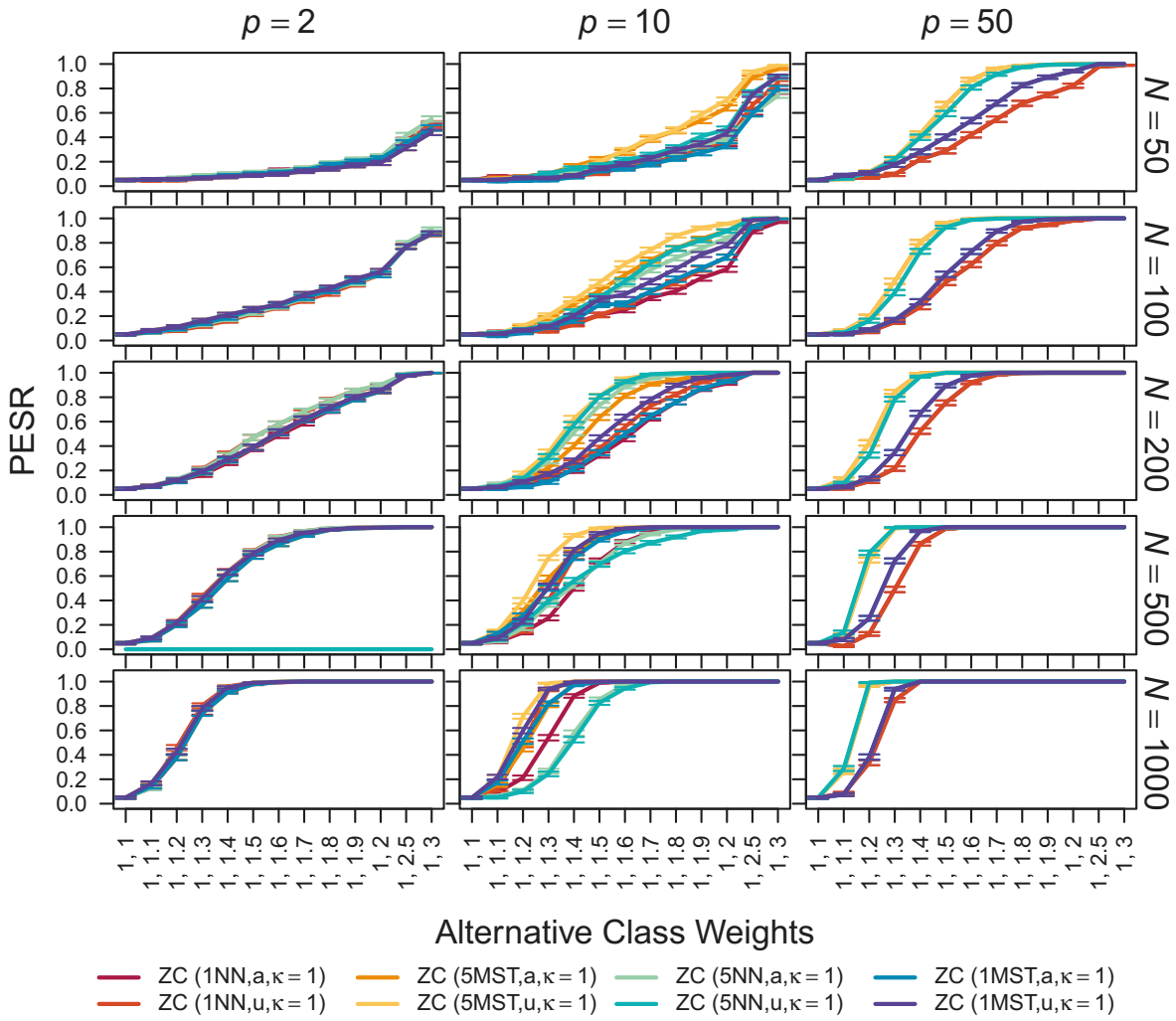


Figure 43: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

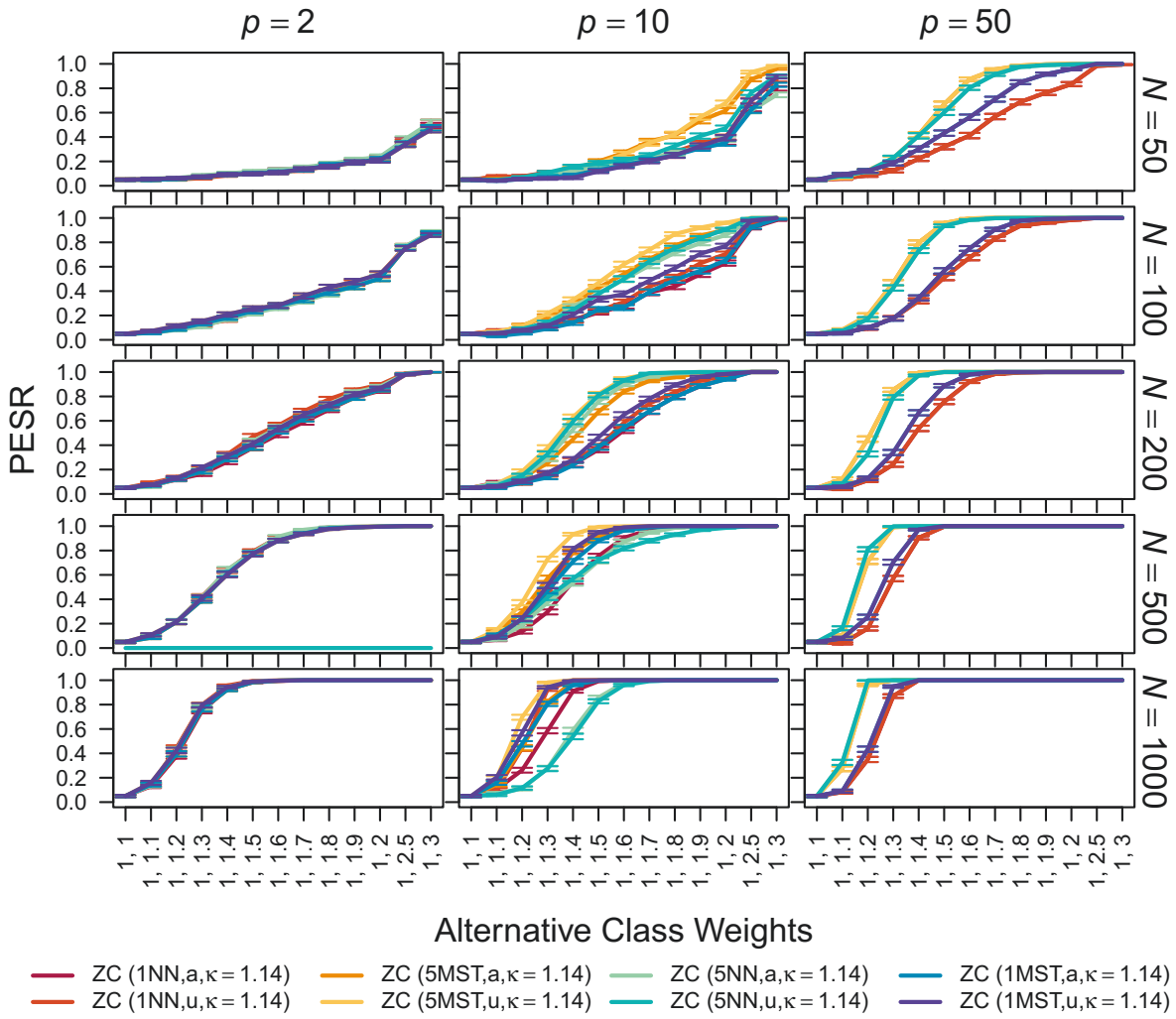


Figure 44: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

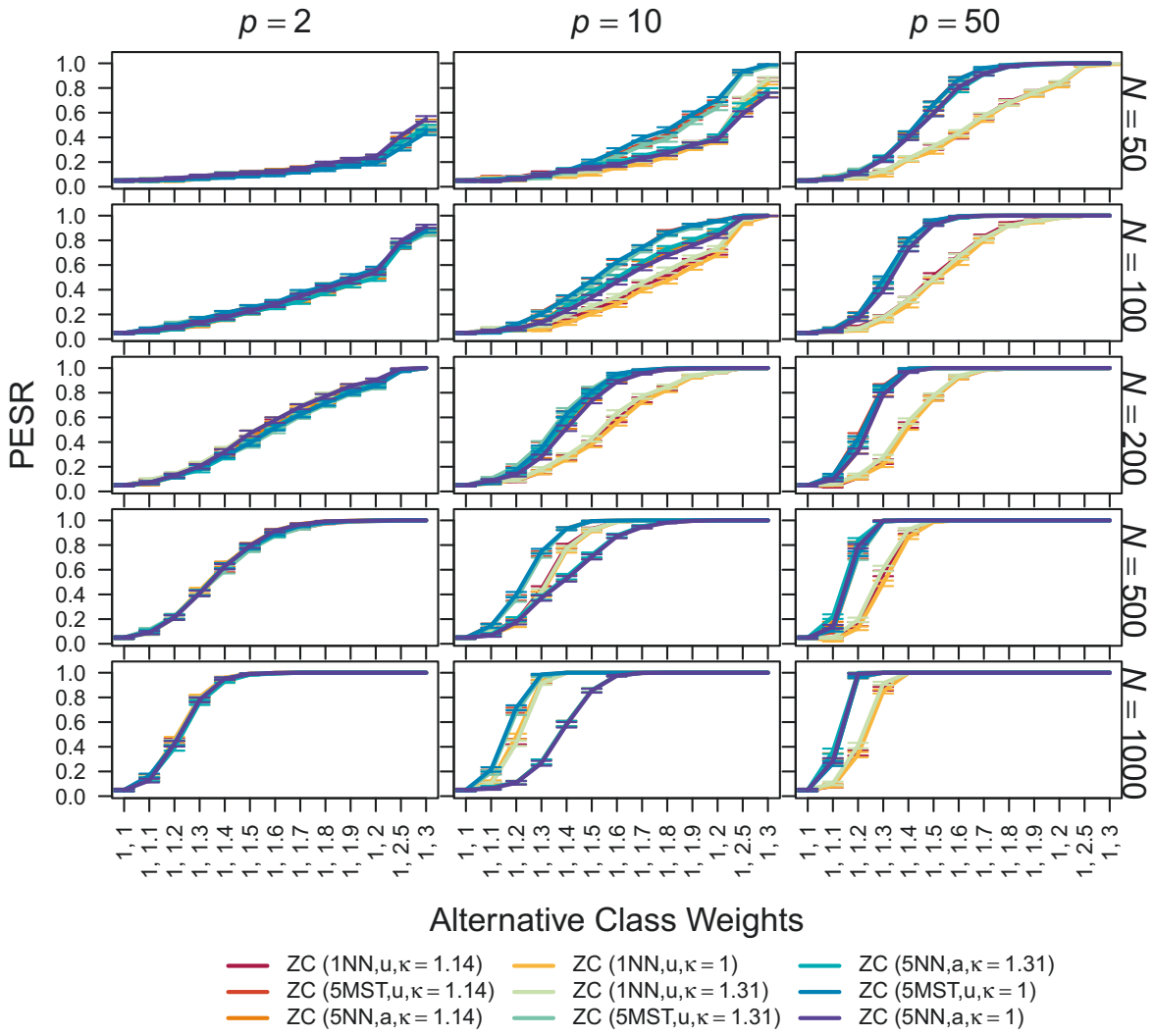


Figure 45: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

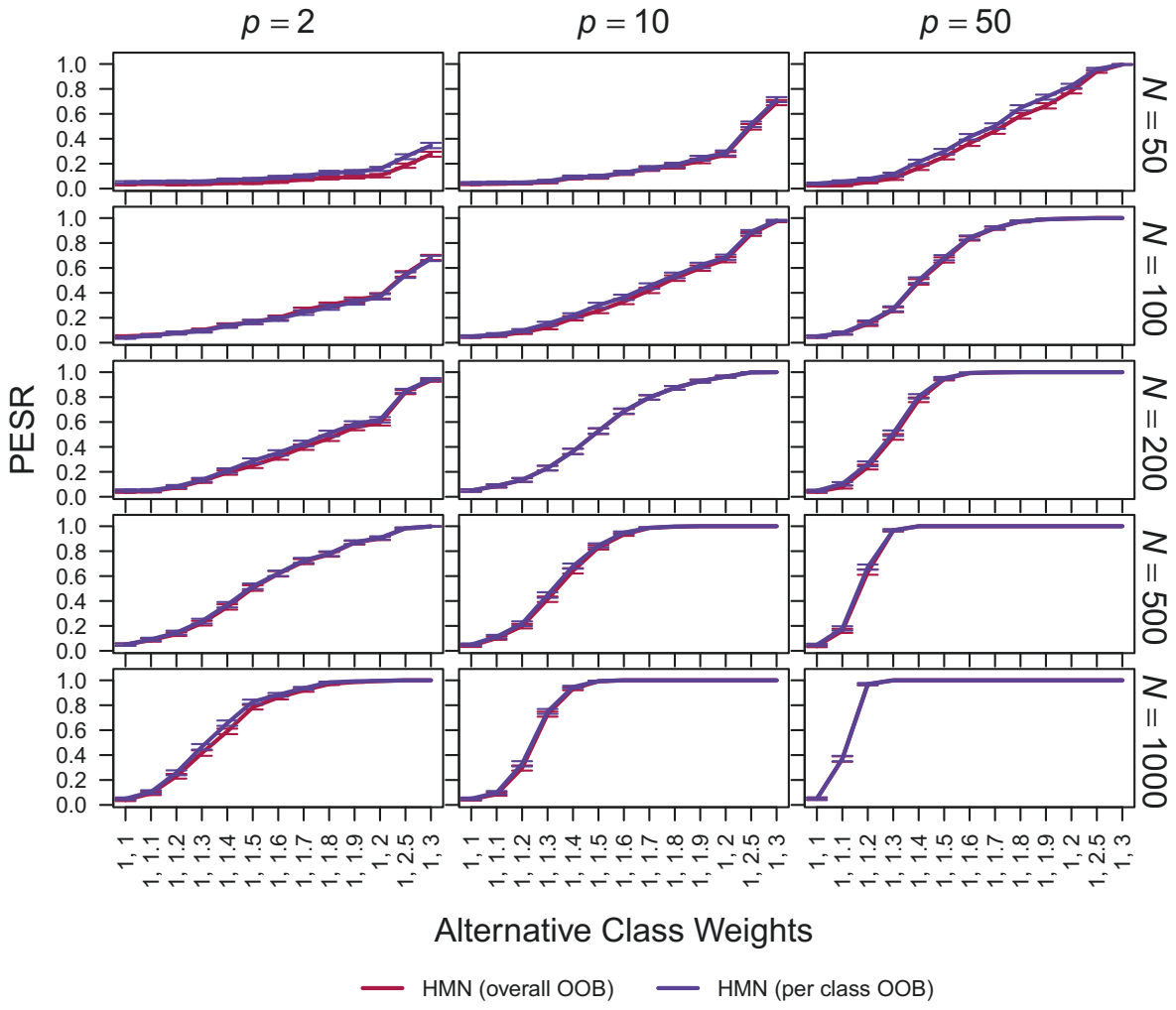


Figure 46: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

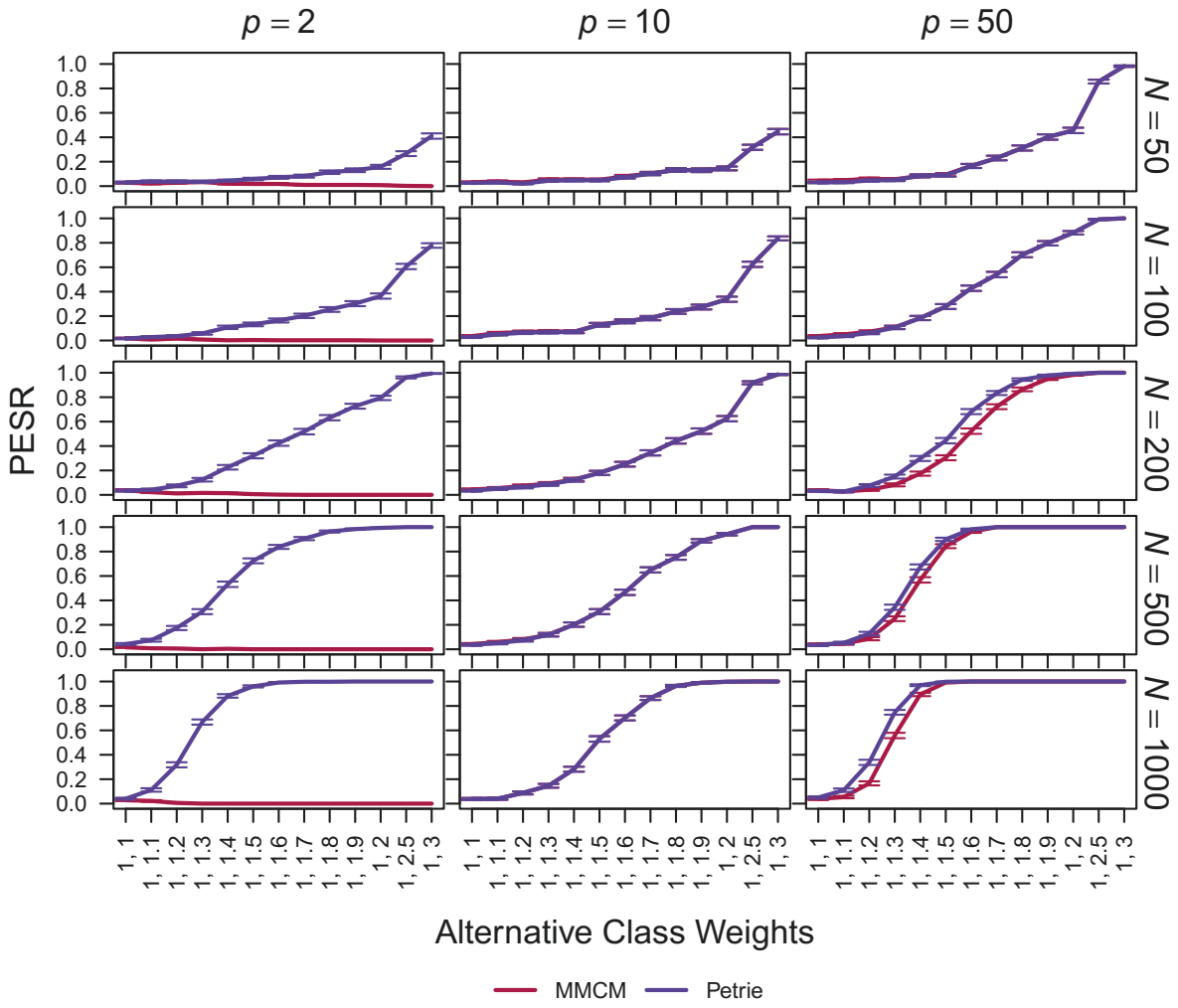


Figure 47: Proportion of extreme simulation repetitions (PESR) for two datasets of the same sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

F.2 $k = 2$, Unbalanced Sample Sizes

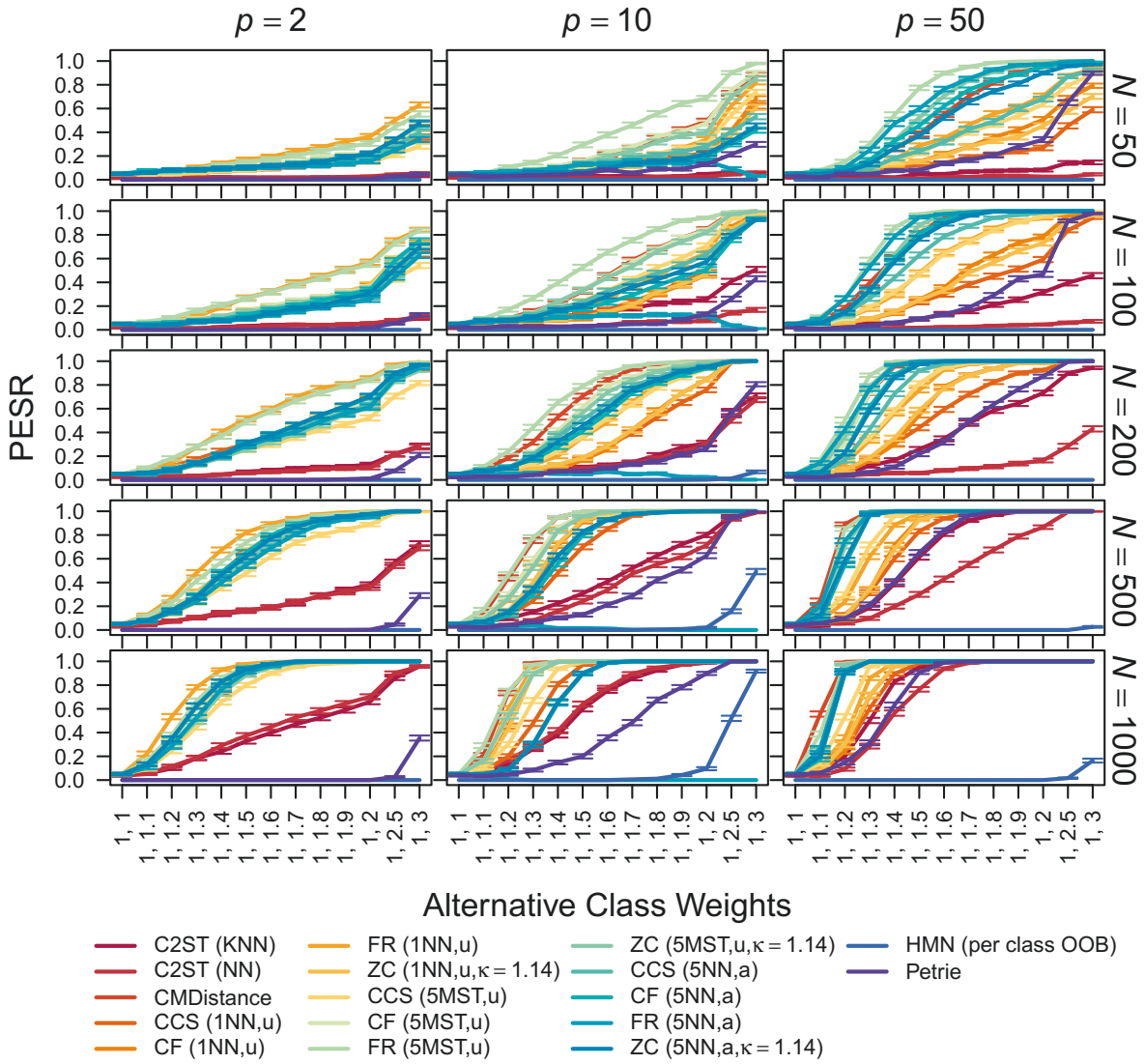


Figure 48: Proportion of extreme simulation repetitions (PESR) for two datasets of unequal sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the second dataset. This means the weights in the first dataset are set to $(1, 1)$, and in the second dataset to $(1, 1 + \delta)$. Error bars indicate Monte Carlo standard errors.

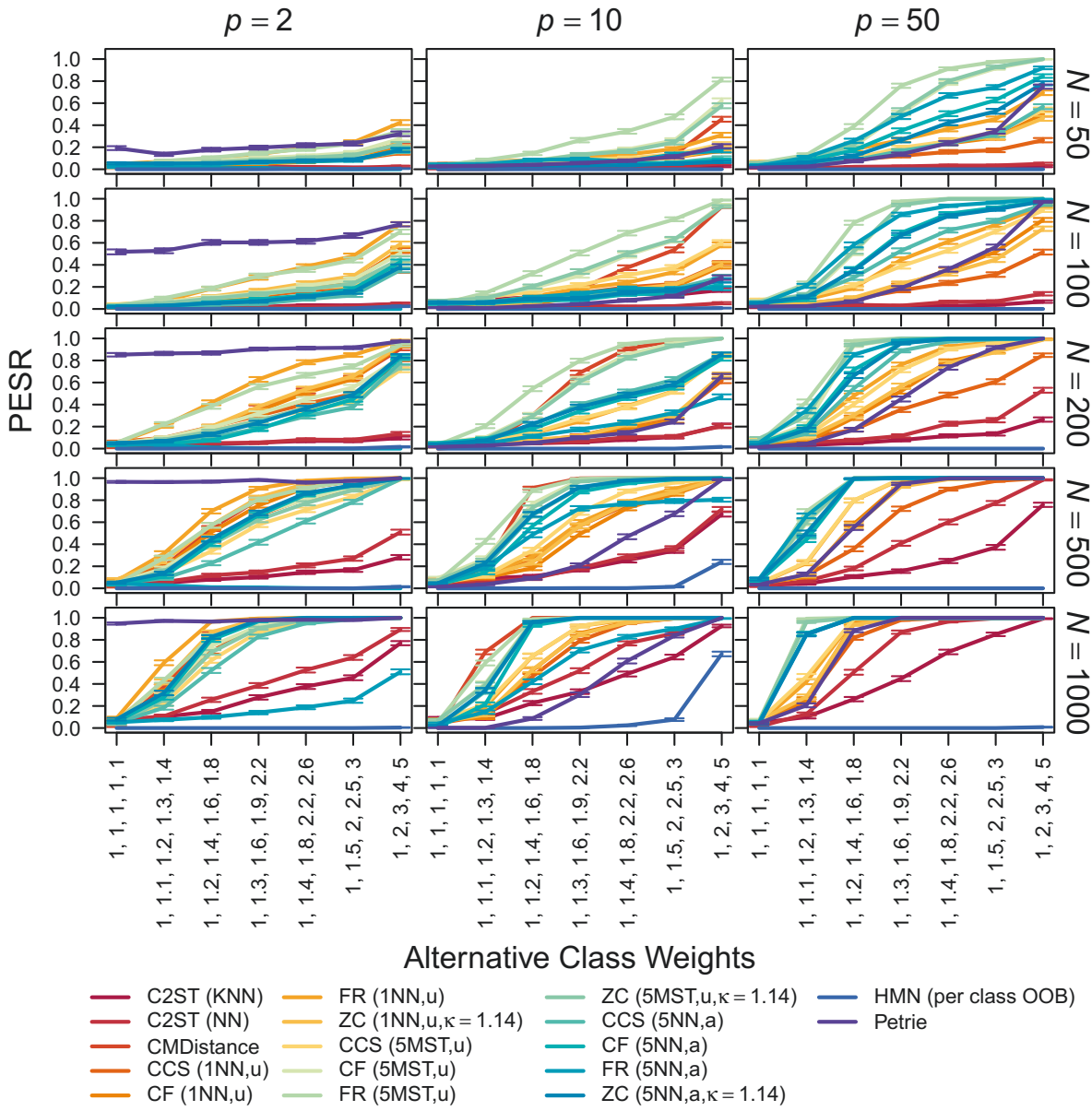


Figure 49: Proportion of extreme simulation repetitions (PESR) for two datasets of unequal sample sizes. The class weights give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for the values 1 to 5 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1, 1, 1, 1)$. Error bars indicate Monte Carlo standard errors.

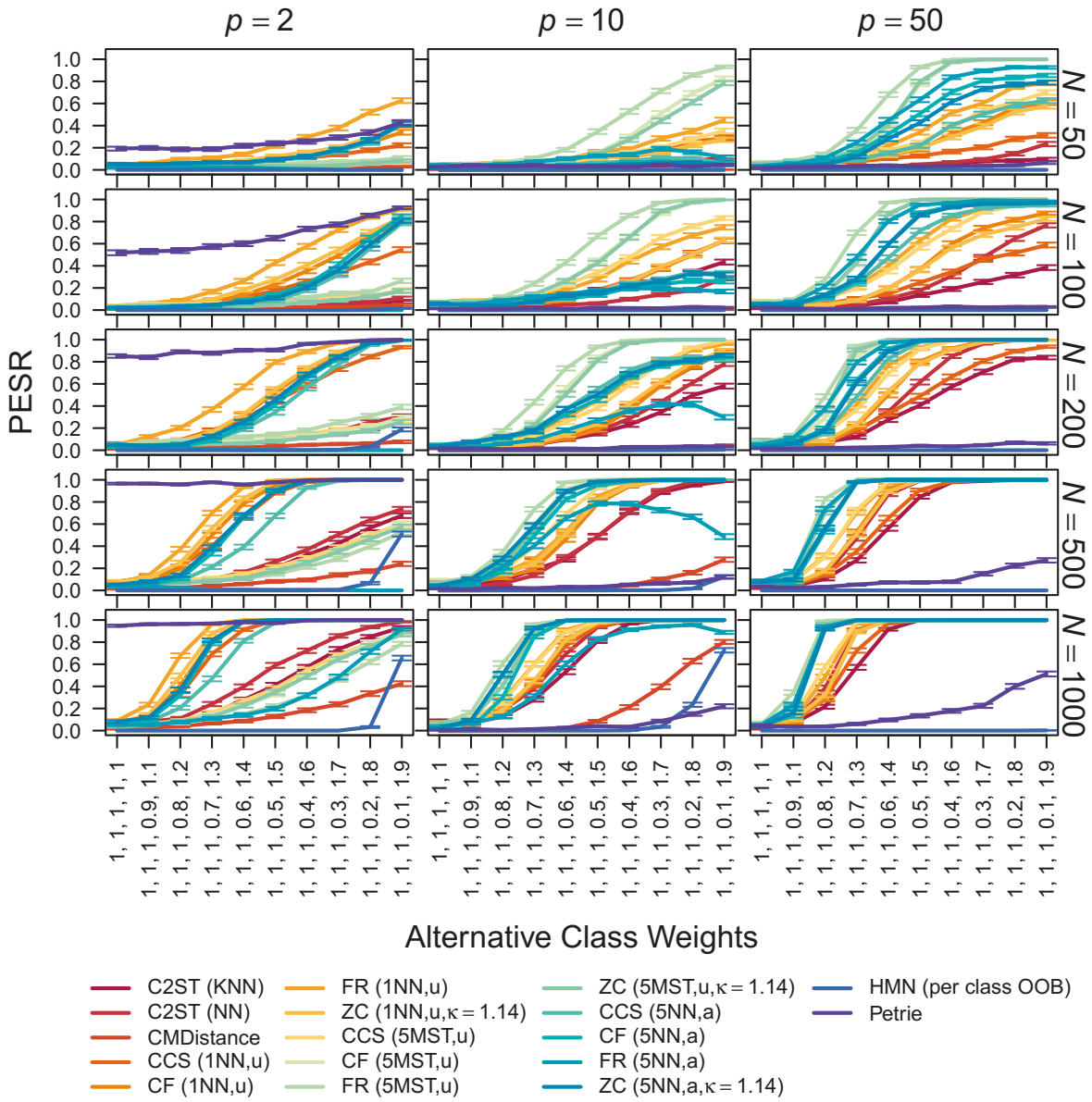


Figure 50: Proportion of extreme simulation repetitions (PESR) for two datasets of unequal sizes. The class weights give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ for the values 1 to 5 for each variable in the second dataset. The weights in the first dataset are always set to $(1, 1, 1, 1, 1)$. Error bars indicate Monte Carlo standard errors.

F.3 $k = 4$, Binary Data, Balanced Sample Sizes

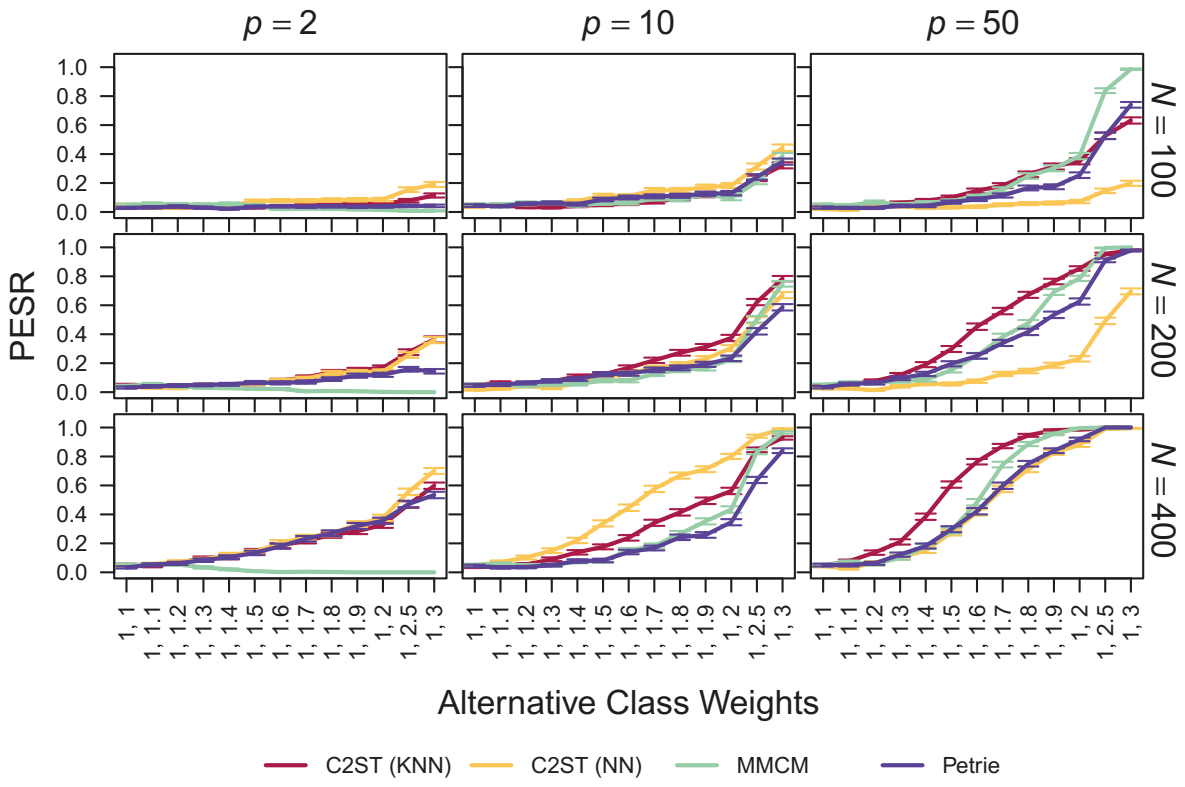


Figure 51: Proportion of extreme simulation repetitions (PESR) for four datasets of the same sample sizes with binary variables. The class weights, i.e. the unnormalized probabilities for the values 0 and 1, in the first, second, and third datasets are always set to $(1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta)$ for each variable in the fourth dataset. Error bars indicate Monte Carlo standard errors.

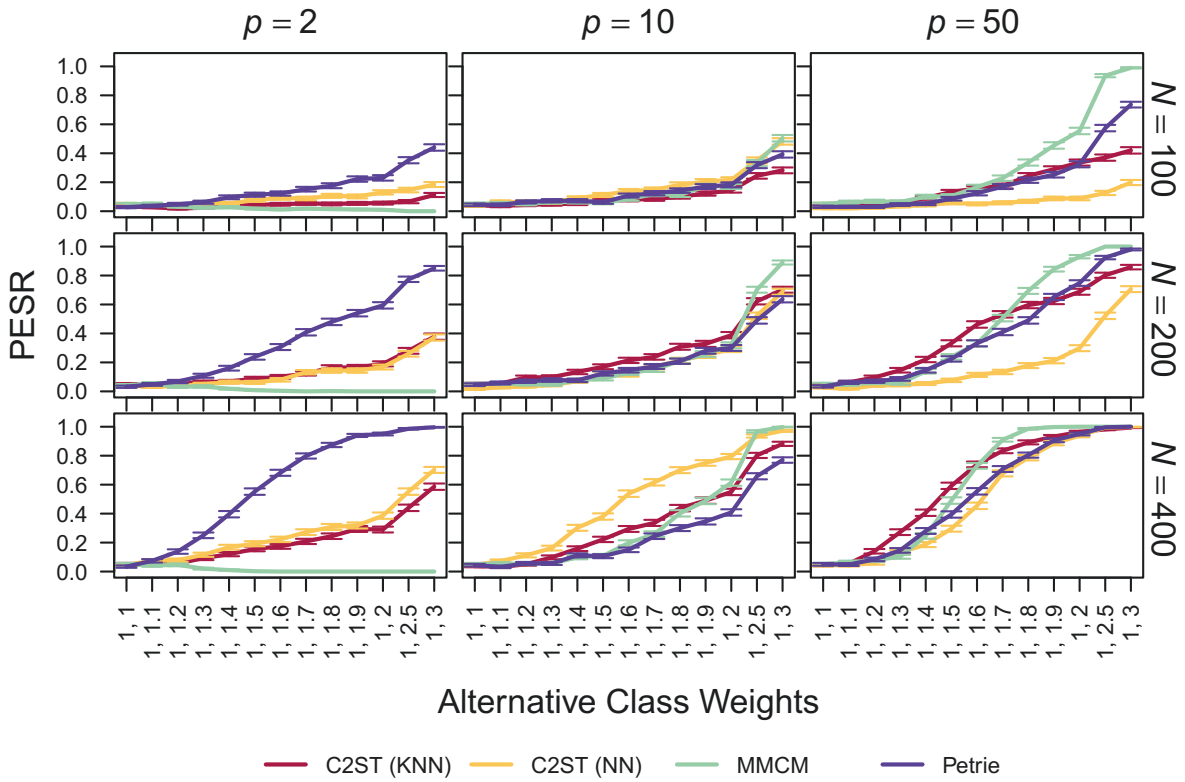


Figure 52: Proportion of extreme simulation repetitions (PESR) for four datasets of the same sample sizes with binary variables. The class weights, i.e. the unnormalized probabilities for the values 0 and 1, in the first and second datasets, are always set to $(1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta)$for each variable in the third and fourth datasets. Error bars indicate Monte Carlo standard errors.

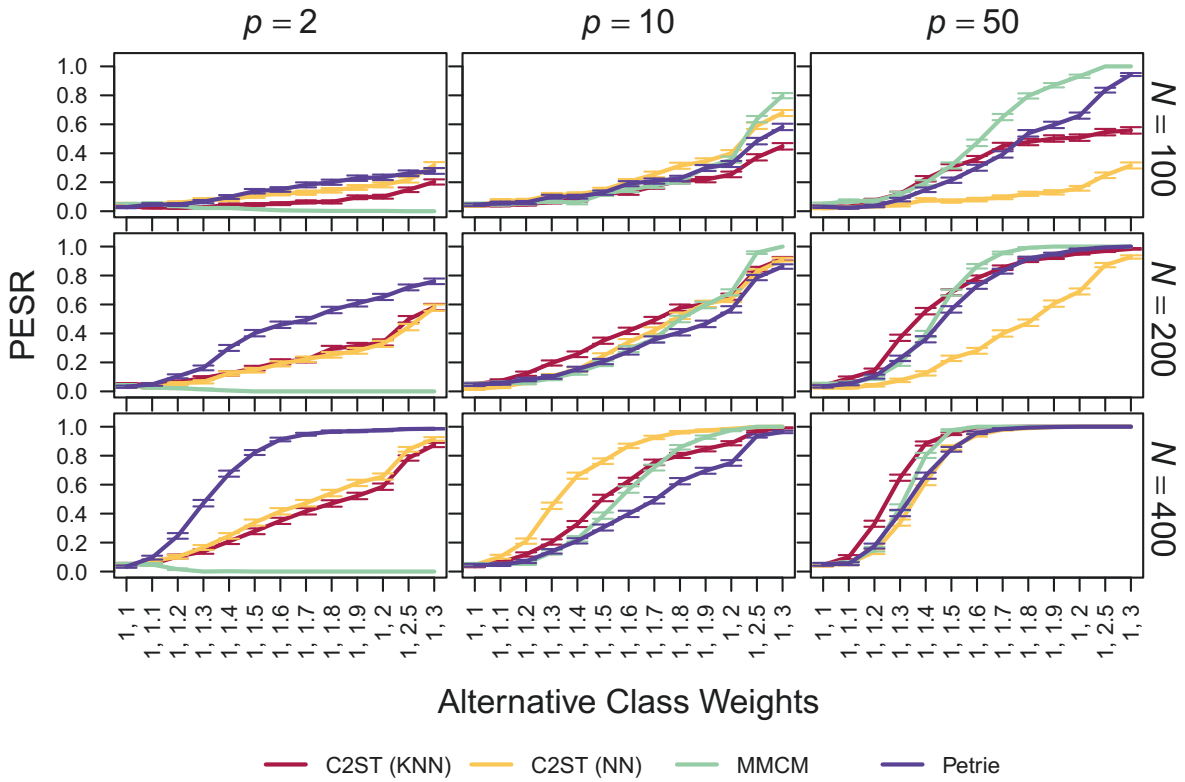


Figure 53: Proportion of extreme simulation repetitions (PESR) for four datasets of the same sample sizes with binary variables. The class weights, i.e. the unnormalized probabilities for the values 0 and 1, in the first and second datasets, are always set to $(1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta)$ for each variable in the third dataset. The weights in the fourth dataset are set to $(1, 1 + 2\delta)$. Error bars indicate Monte Carlo standard errors.

F.4 $k = 4$, Binary Data, Unbalanced Sample Sizes

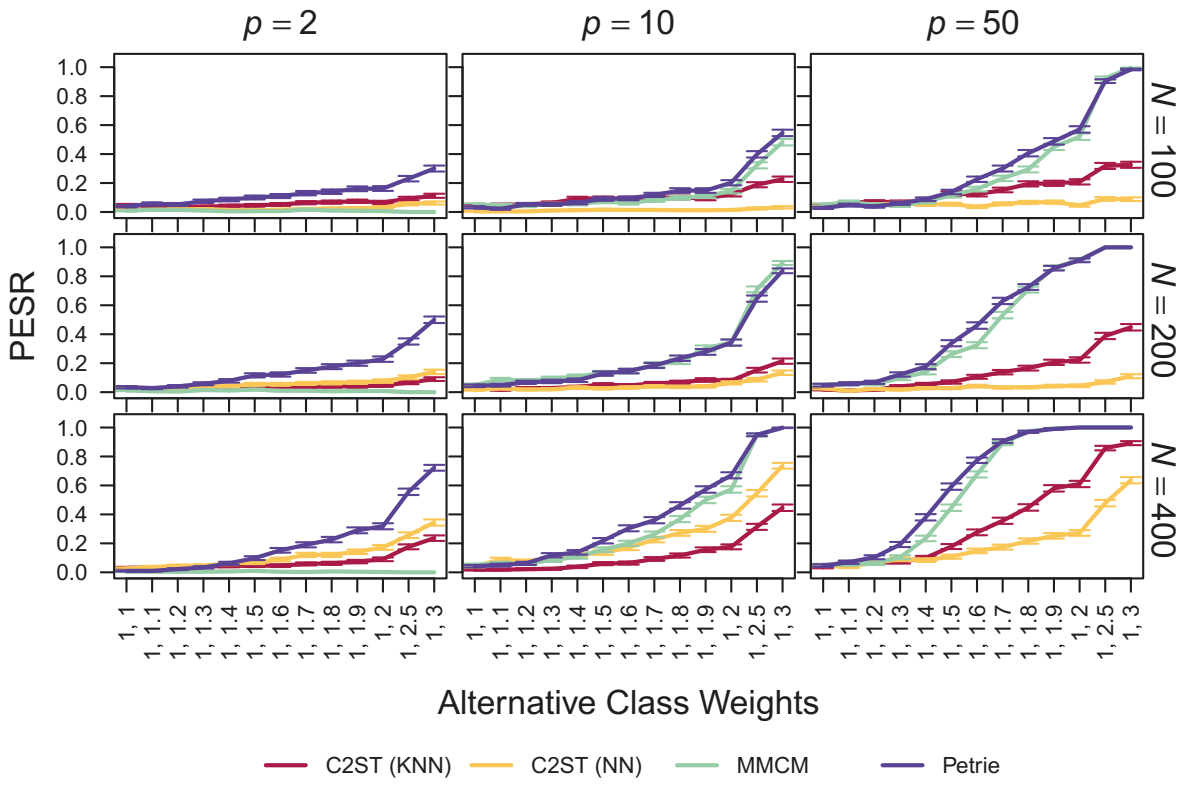


Figure 54: Proportion of extreme simulation repetitions (PESR) for four datasets of unequal sample sizes with binary variables. The class weights, i.e. the unnormalized probabilities for the values 0 and 1, in the first, second, and third datasets are always set to $(1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta)$ for each variable in the fourth dataset. Error bars indicate Monte Carlo standard errors.

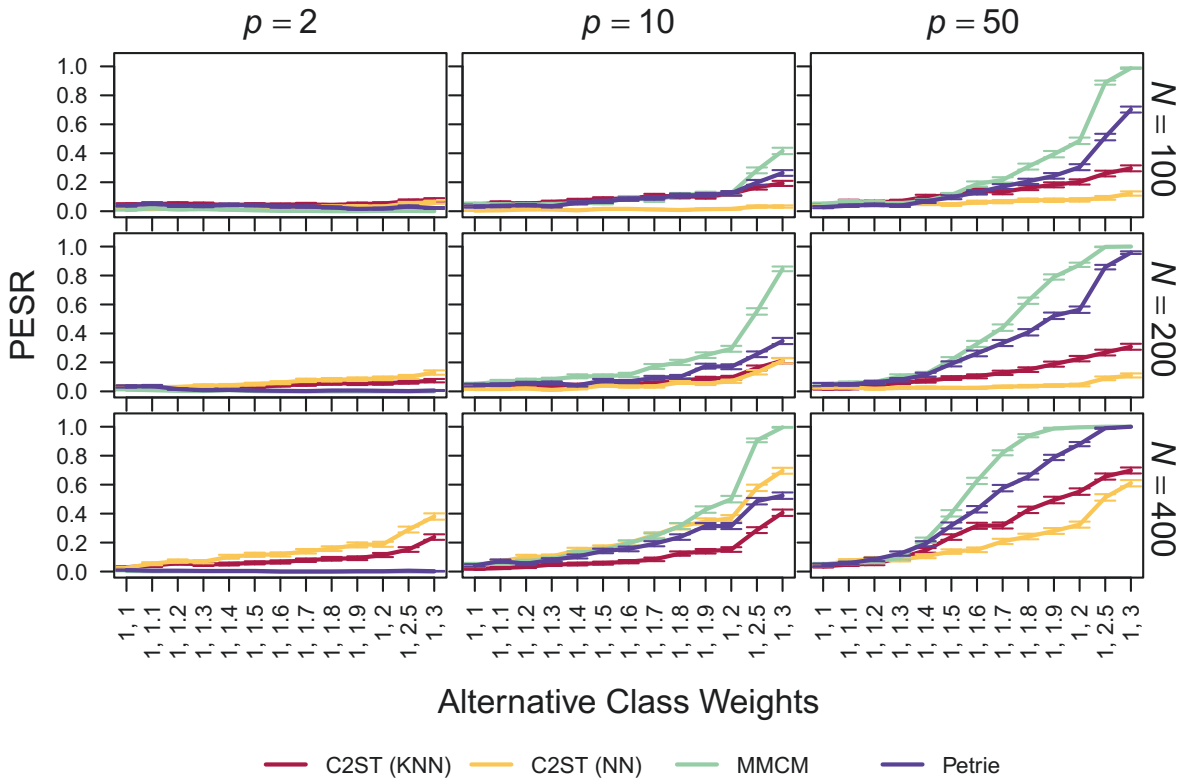


Figure 55: Proportion of extreme simulation repetitions (PESR) for four datasets of unequal sample sizes with binary variables. The class weights, i.e. the unnormalized probabilities for the values 0 and 1, in the first and second datasets, are always set to $(1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta)$for each variable in the third and fourth datasets. Error bars indicate Monte Carlo standard errors.

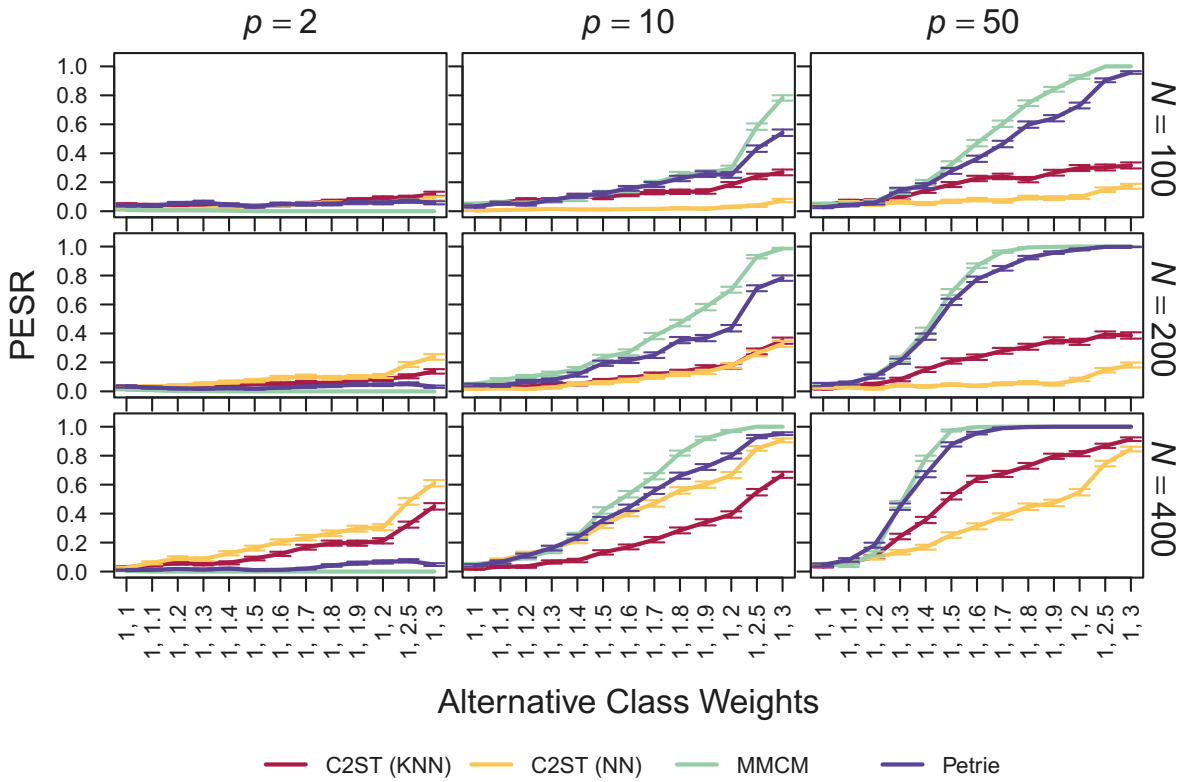


Figure 56: Proportion of extreme simulation repetitions (PESR) for four datasets of unequal sample sizes with binary variables. The class weights give the unnormalized probabilities $(1, 1 + \delta)$ for the values 0 and 1 for each variable in the third dataset. The weights in the first and second datasets are always set to $(1, 1)$. The weights in the fourth dataset are set to $(1, 1 + 2\delta)$. Error bars indicate Monte Carlo standard errors.

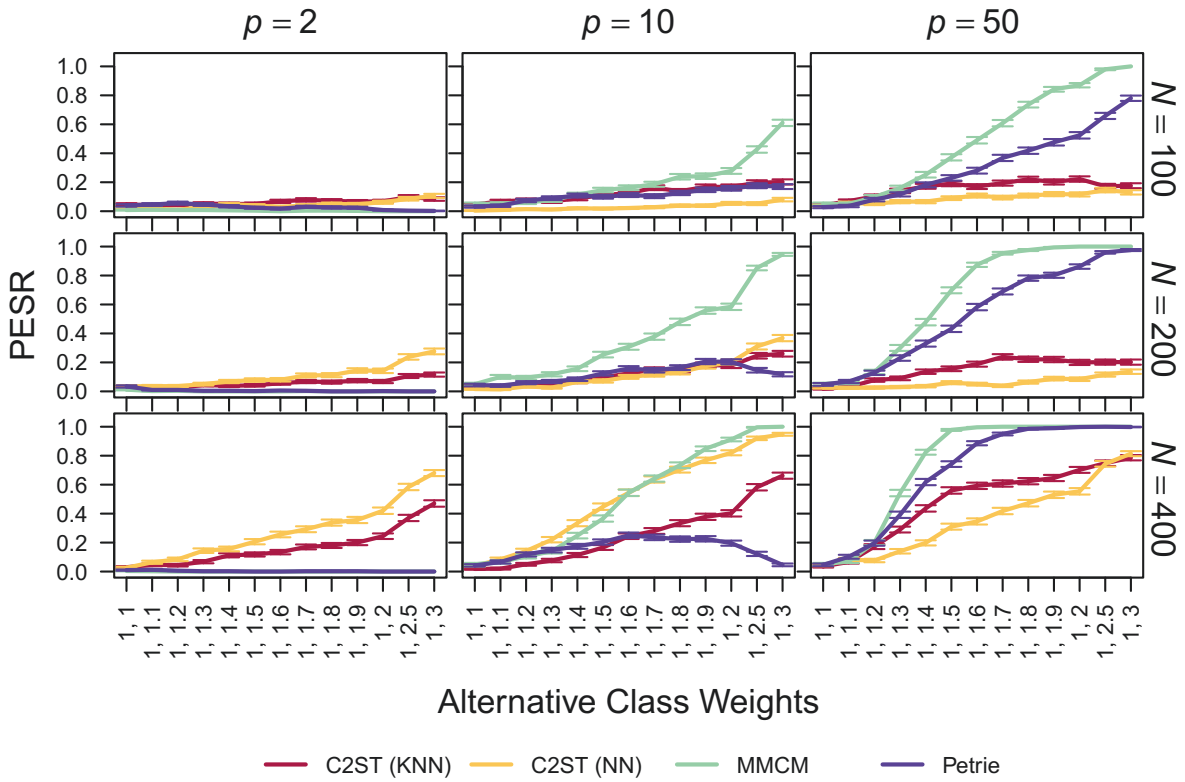


Figure 57: Proportion of extreme simulation repetitions (PESR) for four datasets of unequal sample sizes with binary variables. The class weights, i.e. the unnormalized probabilities for the values 0 and 1, in the first dataset, are always set to $(1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta)$ for each variable in the second dataset. The weights in the third dataset are set to $(1, 1 + 2\delta)$. The weights in the fourth dataset are set to $(1, 1 + 3\delta)$. Error bars indicate Monte Carlo standard errors.

F.5 $k = 4$, Multinomial Data, Skewed Probability Distribution, Balanced Sample Sizes

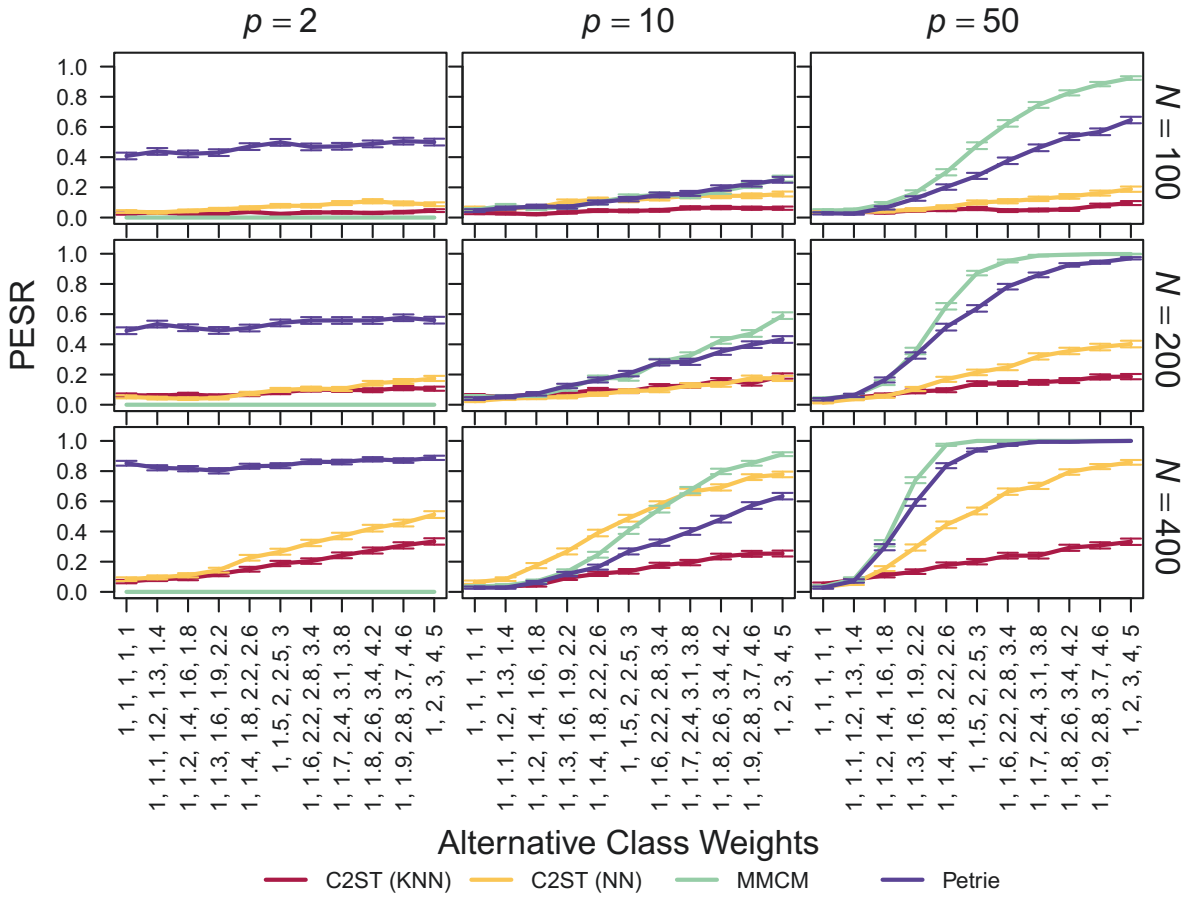


Figure 58: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the same sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first to the third dataset, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for each variable in the fourth dataset. Error bars indicate Monte Carlo standard errors.

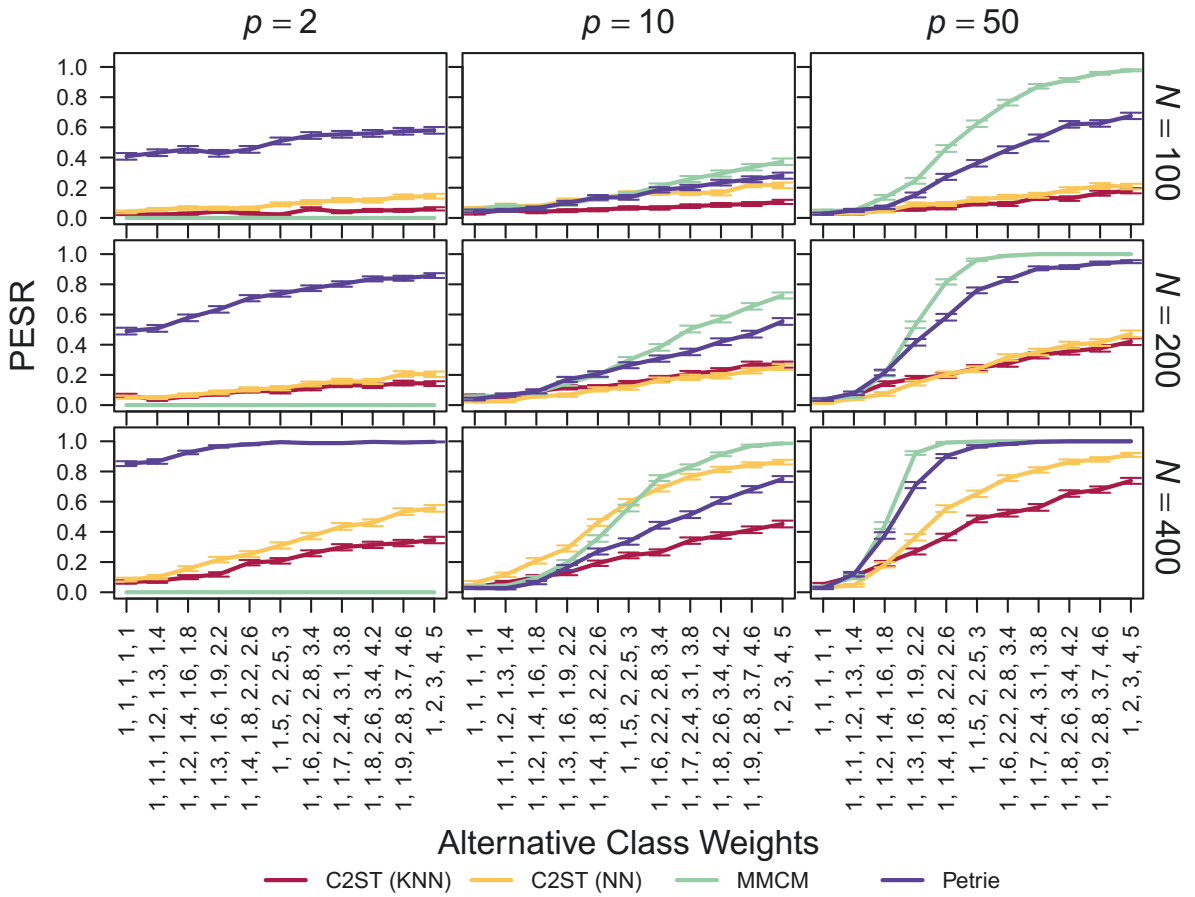


Figure 59: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the same sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for each variable in the third and fourth dataset. Error bars indicate Monte Carlo standard errors.

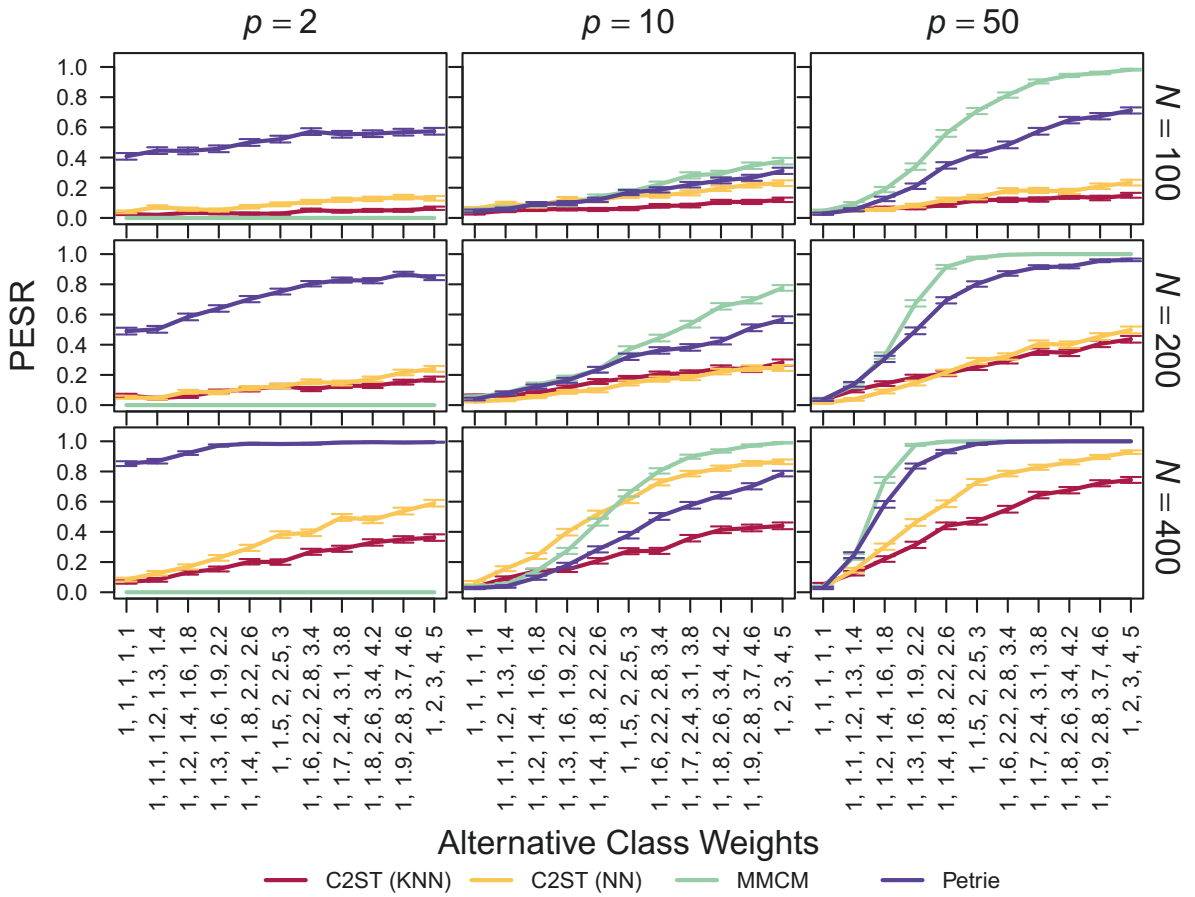


Figure 60: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the same sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1 + \delta, 1 + 2\delta, 1 + 3\delta, 1 + 4\delta)$ for each variable in the third dataset. The weights in the fourth dataset are given by $(1, 1 + (\delta + 0.1), 1 + 2(\delta + 0.1), 1 + 3(\delta + 0.1), 1 + 4(\delta + 0.1))$. Error bars indicate Monte Carlo standard errors.

F.6 $k = 4$, Multinomial Data, Skewed Probability Distribution, Unbalanced Sample Sizes

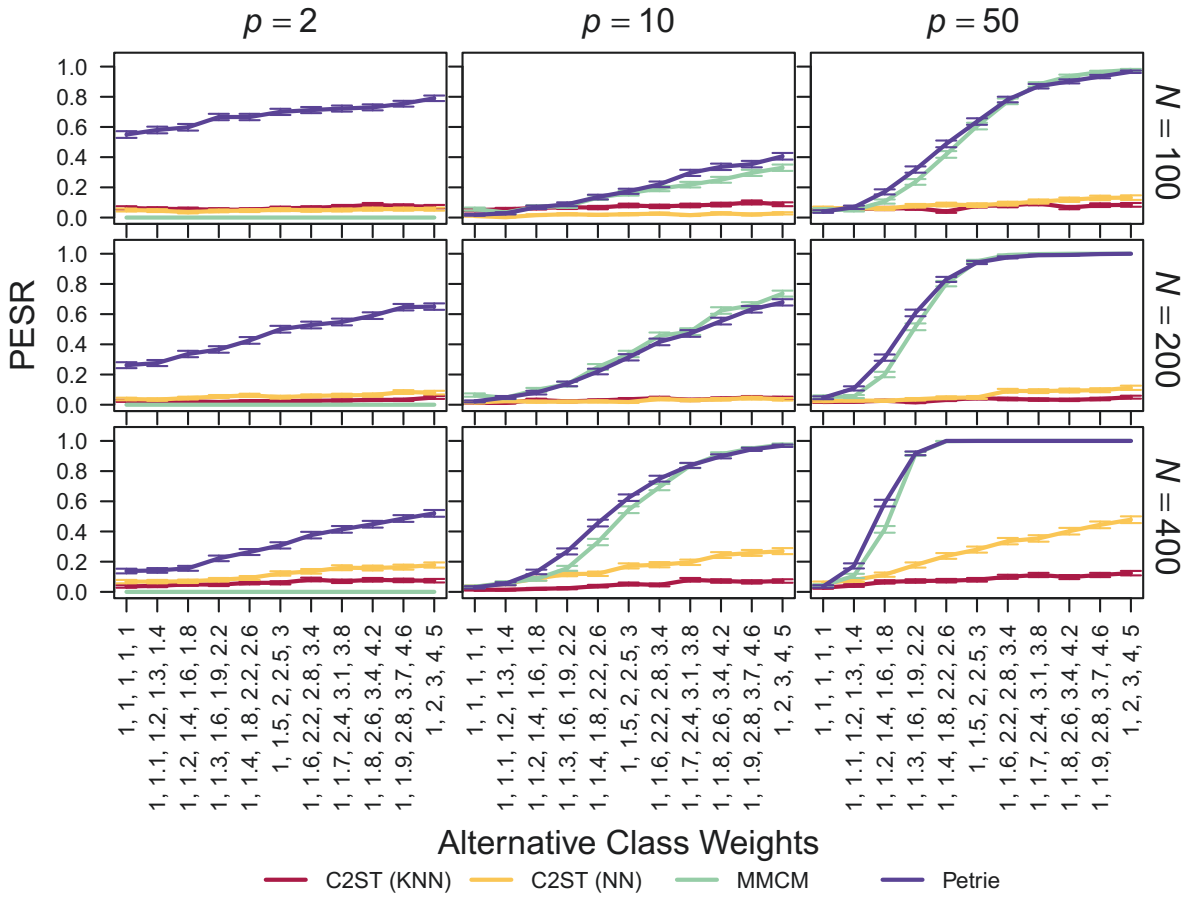


Figure 61: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first to the third dataset, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for each variable in the fourth dataset. Error bars indicate Monte Carlo standard errors.

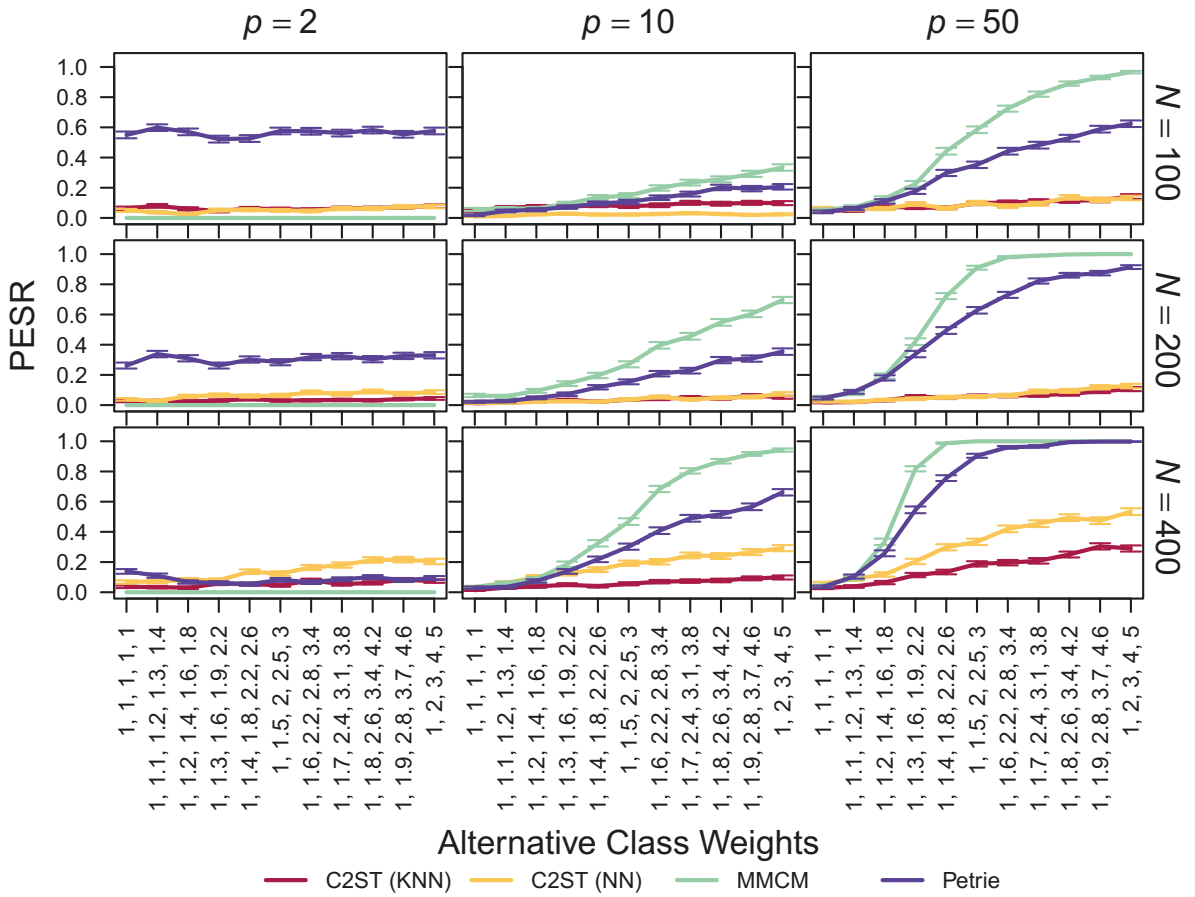


Figure 62: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for each variable in the third and fourth dataset. Error bars indicate Monte Carlo standard errors.

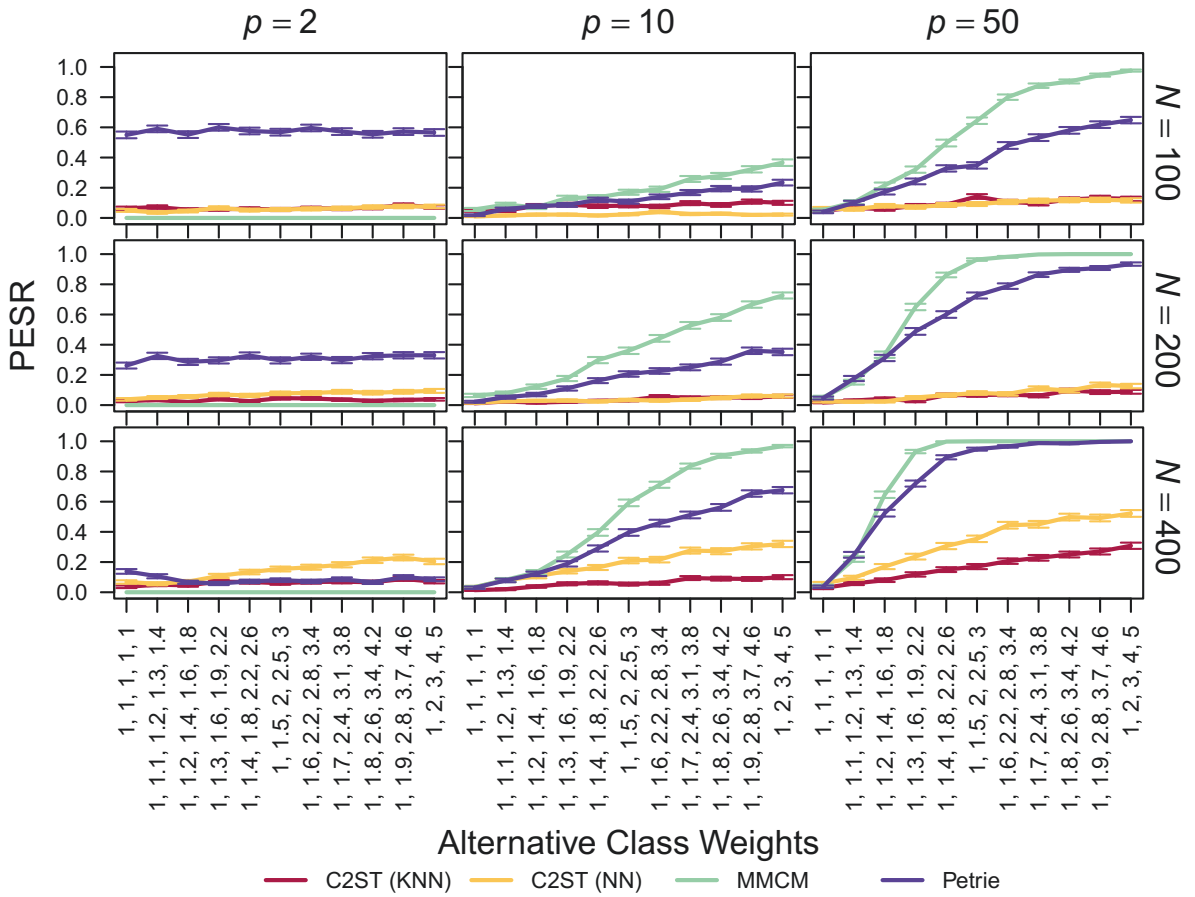


Figure 63: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for each variable in the third dataset. The weights in the fourth dataset are given by $(1, 1 + (\delta + 0.1), 1 + 2(\delta + 0.1), 1 + 3(\delta + 0.1), 1 + 4(\delta + 0.1))$. Error bars indicate Monte Carlo standard errors.

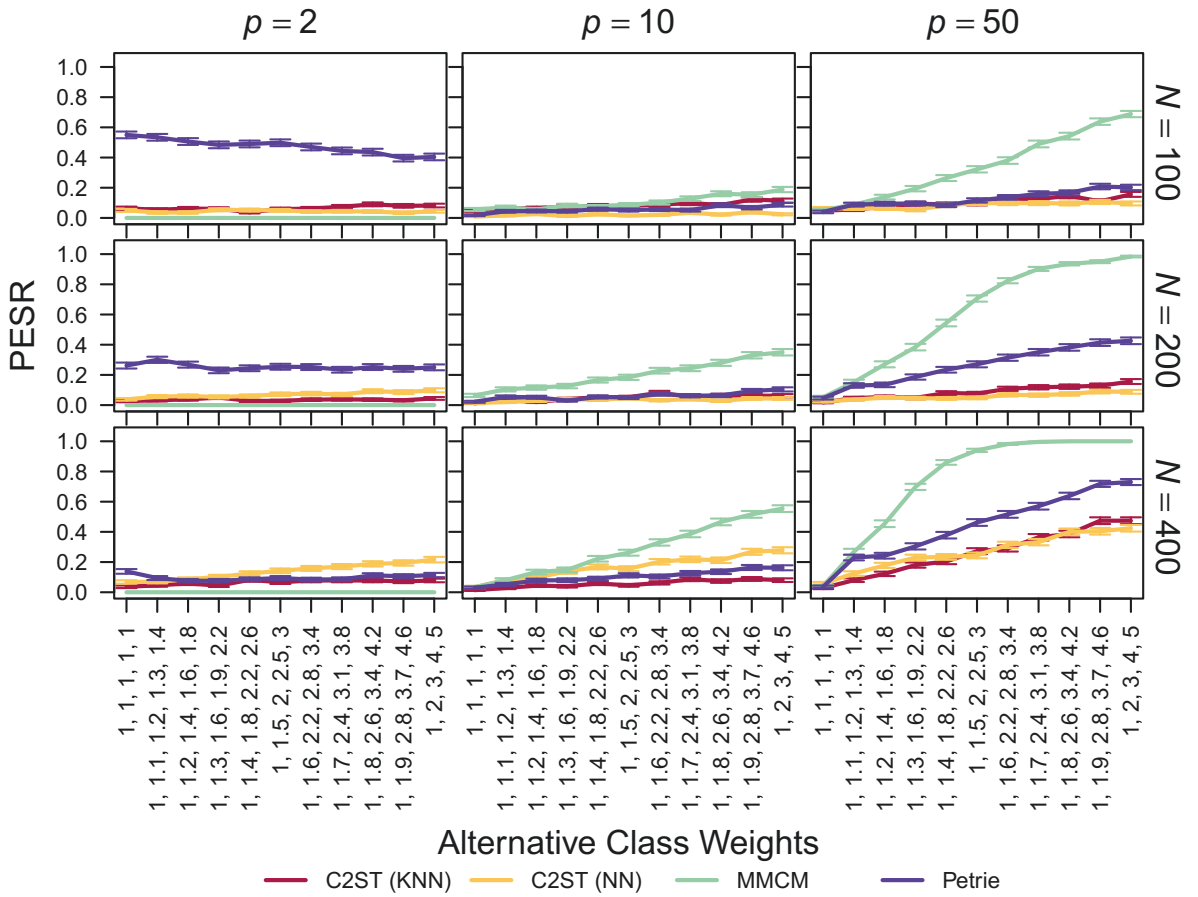


Figure 64: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first dataset are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1+\delta, 1+2\delta, 1+3\delta, 1+4\delta)$ for each variable in the second dataset. The weights in the third dataset are given by $(1, 1+(\delta+0.1), 1+2(\delta+0.1), 1+3(\delta+0.1), 1+4(\delta+0.1))$. The weights in the fourth dataset are given by $(1, 1+(\delta+0.2), 1+2(\delta+0.2), 1+3(\delta+0.2), 1+4(\delta+0.2))$. Error bars indicate Monte Carlo standard errors.

F.7 $k = 4$, Multinomial Data, One Class Up, One Class Down, Balanced Sample Sizes

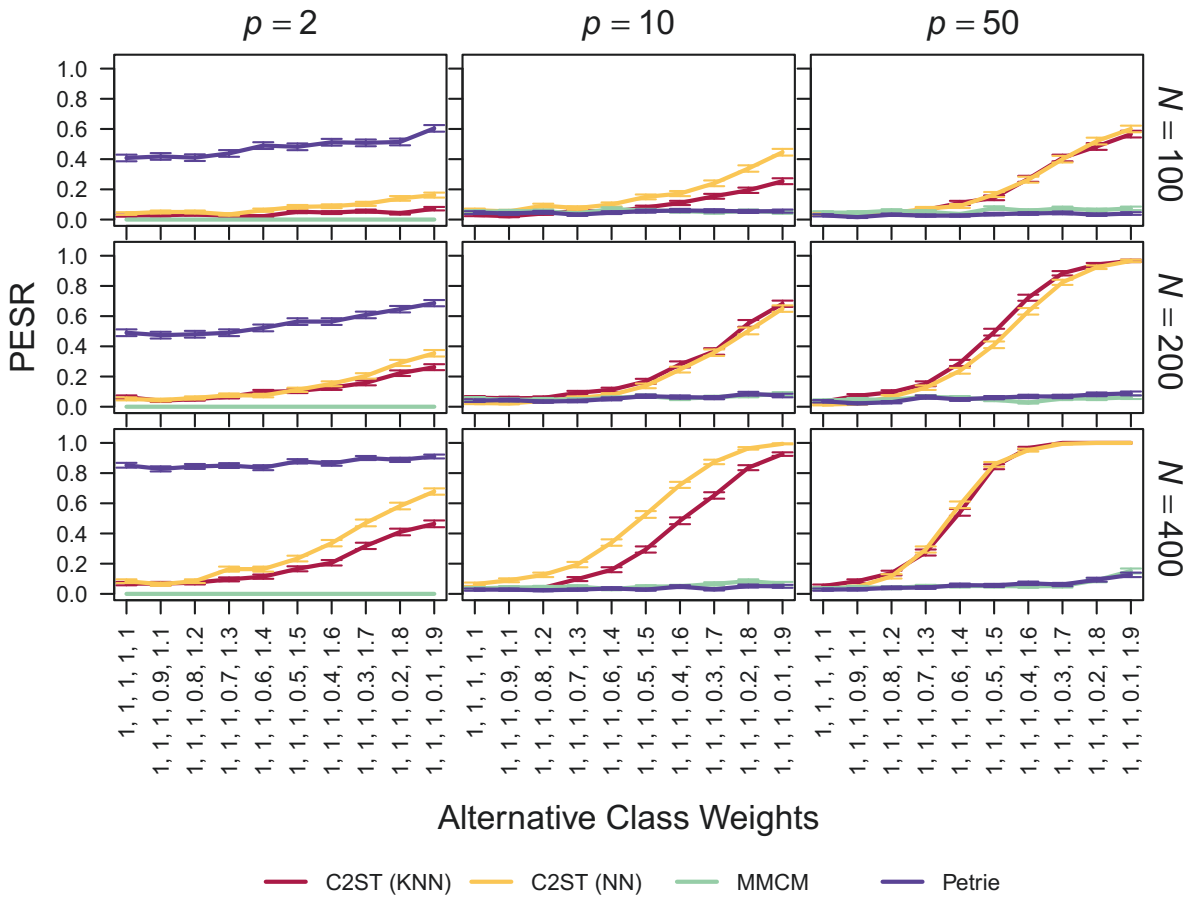


Figure 65: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the same sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first to the third dataset, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ in the fourth dataset. Error bars indicate Monte Carlo standard errors.

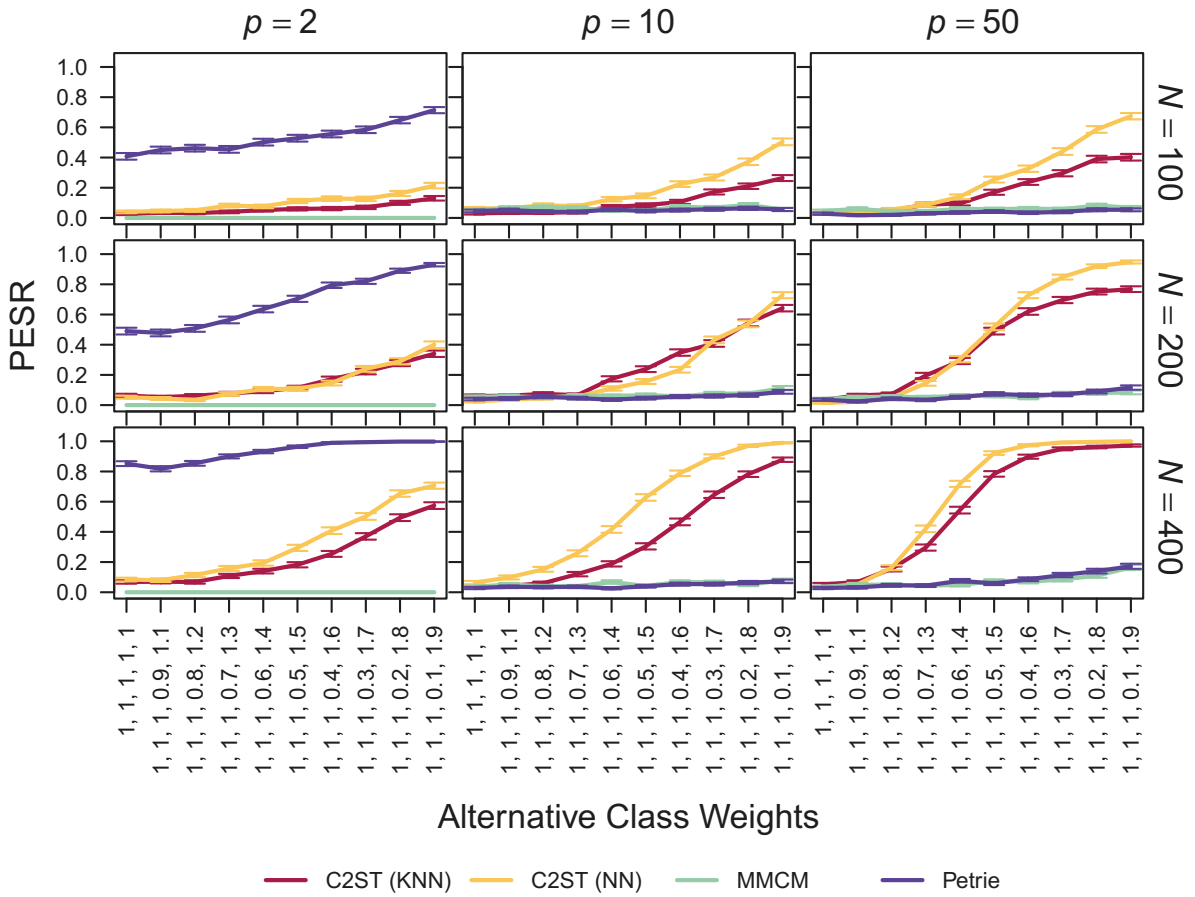


Figure 66: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the same sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ for each variable in the third and fourth datasets. Error bars indicate Monte Carlo standard errors.

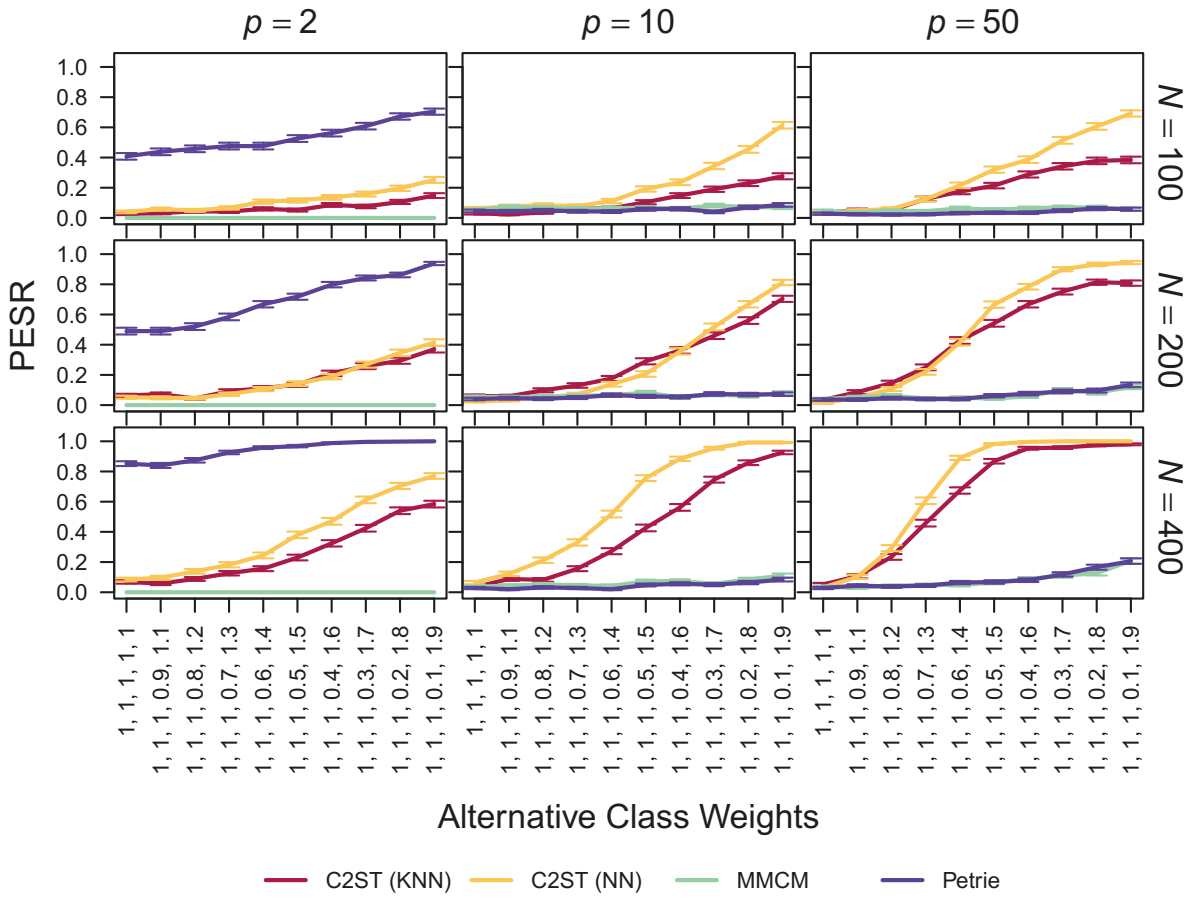


Figure 67: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the same sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ for each variable in the third dataset. The weights in the fourth dataset are given by $(1, 1, 1, 1 + \delta + 0.1, 1 - \delta - 0.1)$. Error bars indicate Monte Carlo standard errors.

F.8 $k = 4$, Multinomial Data, One Class Up, One Class Down, Unbalanced Sample Sizes

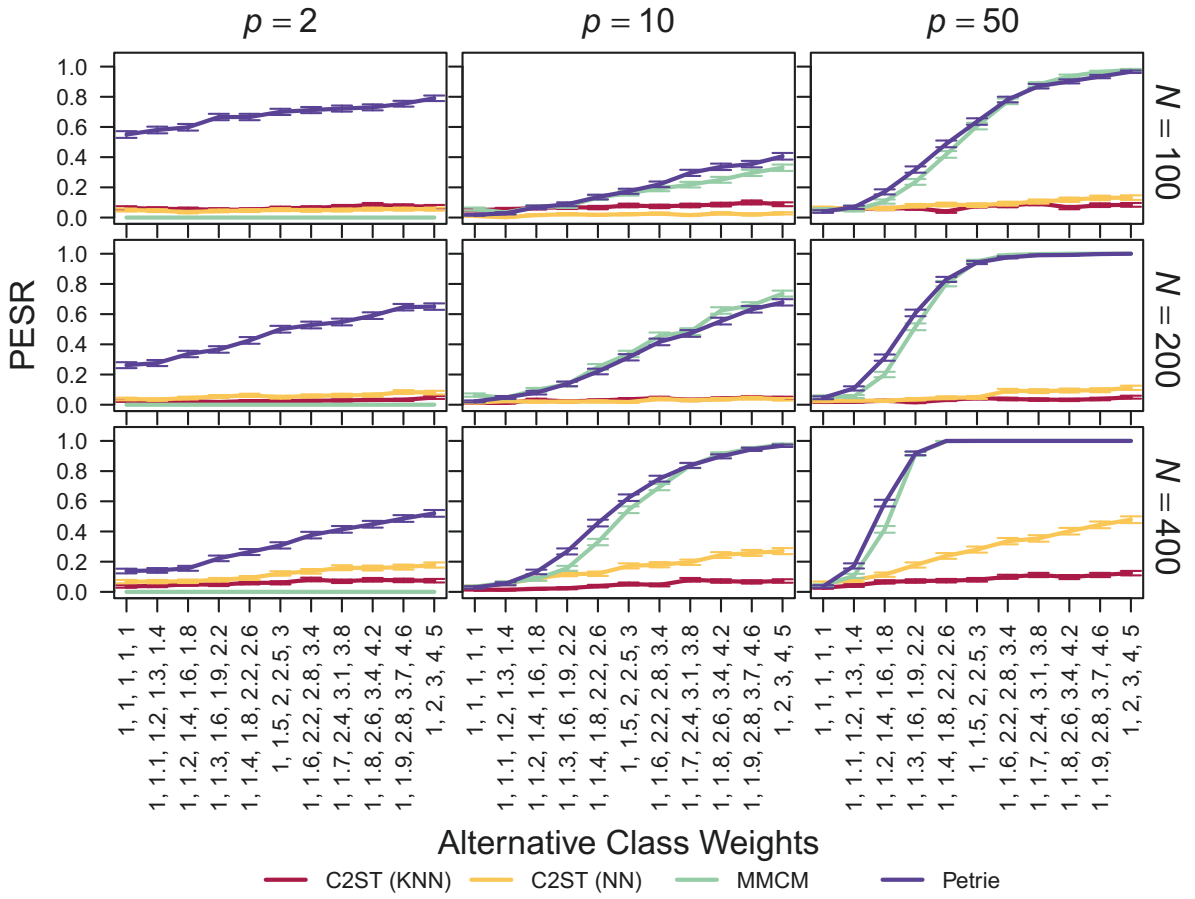


Figure 68: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first to the third dataset, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ in the fourth dataset. Error bars indicate Monte Carlo standard errors.

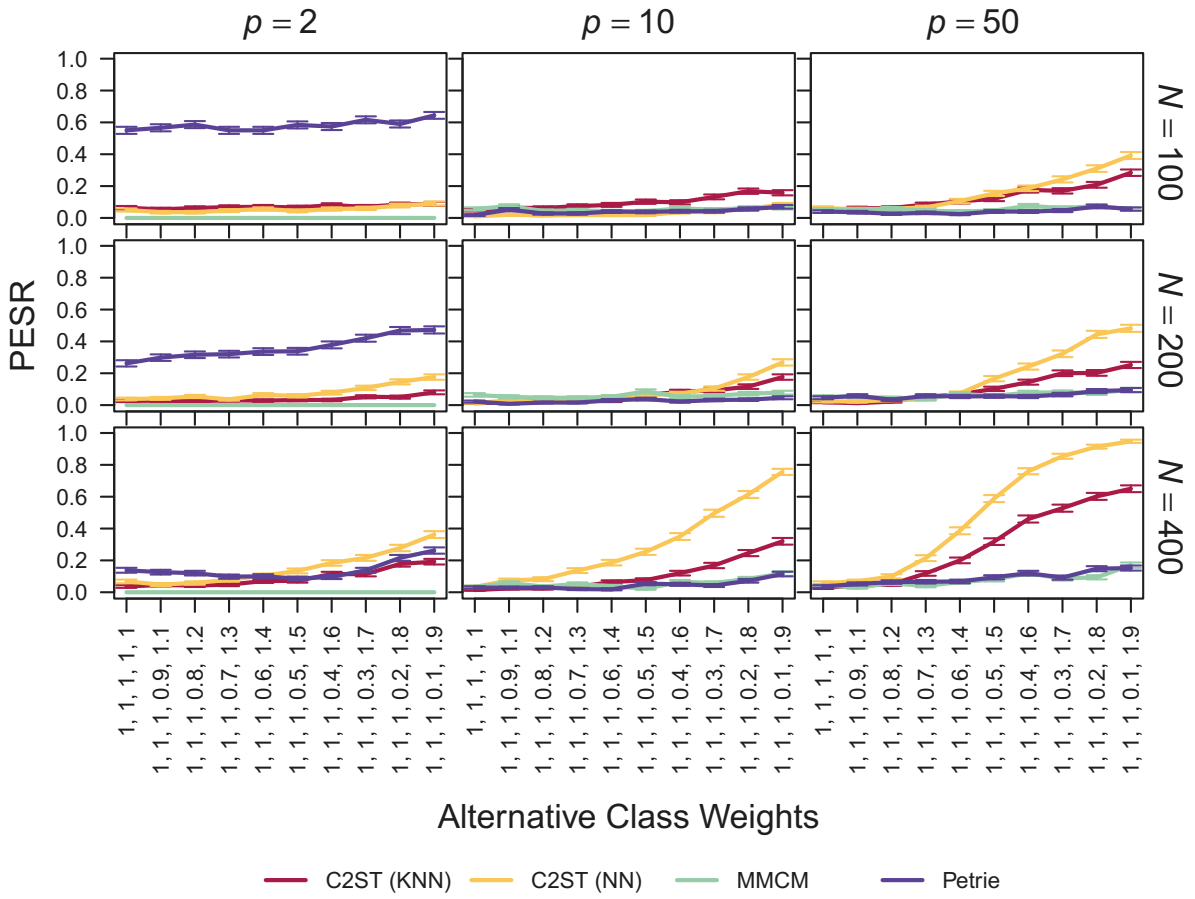


Figure 69: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ for each variable in the third and fourth datasets. Error bars indicate Monte Carlo standard errors.

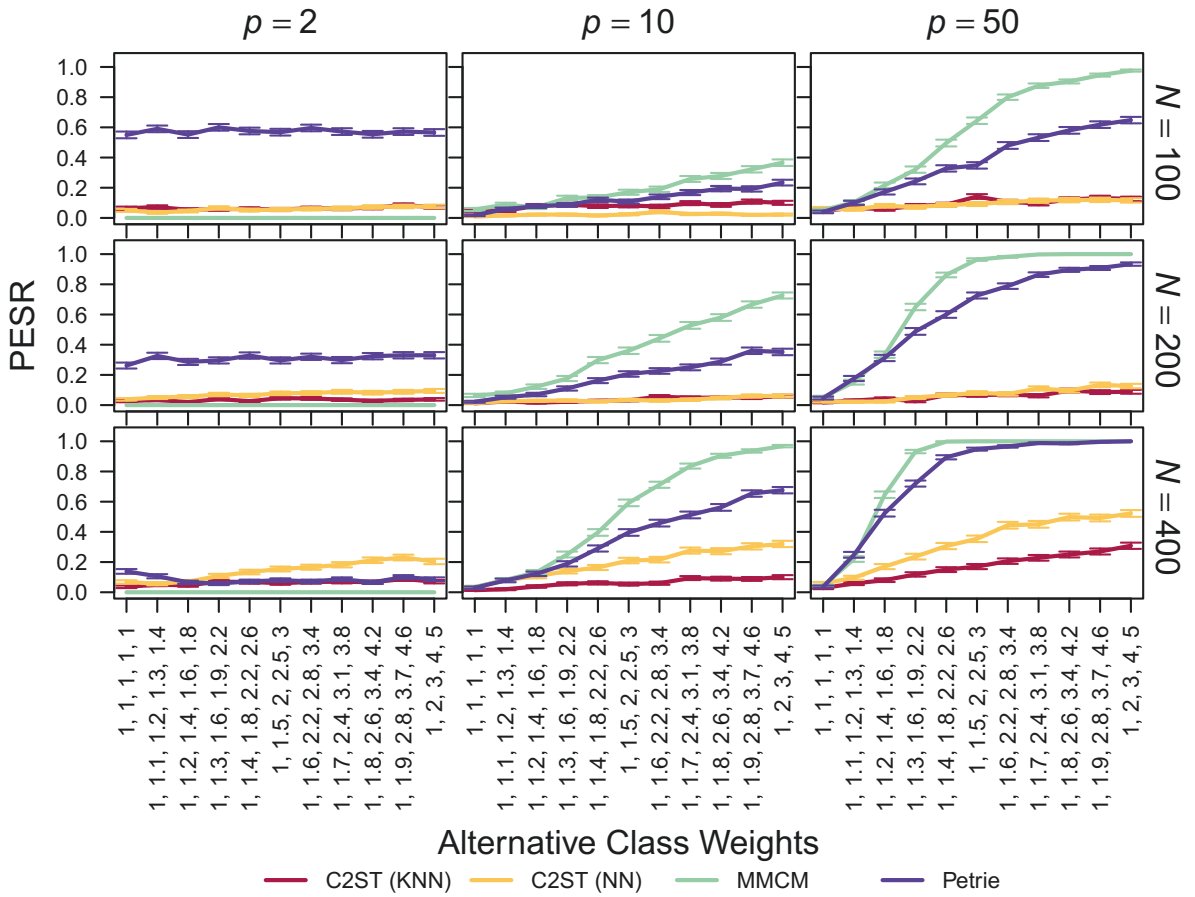


Figure 70: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ for each variable in the third dataset. The weights in the fourth dataset are given by $(1, 1, 1, 1 + \delta + 0.1, 1 - \delta - 0.1)$. Error bars indicate Monte Carlo standard errors.

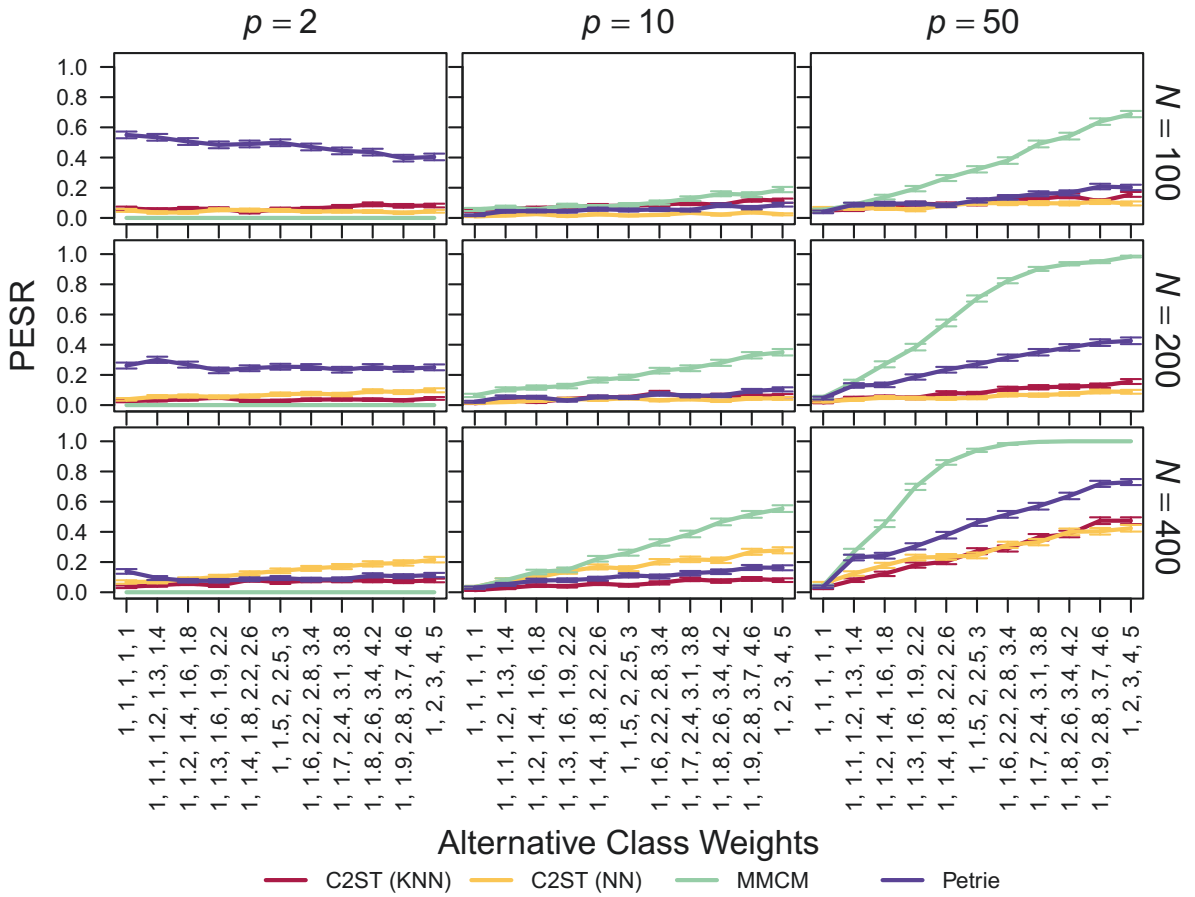


Figure 71: Proportion of extreme simulation repetitions (PESR) for four multinomial datasets of the unequal sample sizes. The class weights, i.e. the unnormalized probabilities for the values 1 to 5, in the first and second datasets, are always set to $(1, 1, 1, 1, 1)$. The class weights on the x-axis give the unnormalized probabilities $(1, 1, 1, 1 + \delta, 1 - \delta)$ for each variable in the third dataset. The weights in the third dataset are given by $(1, 1, 1, 1 + \delta + 0.1, 1 - \delta - 0.1)$, and the weights for the fourth dataset are given by $(1, 1, 1, 1 + \delta + 0.2, 1 - \delta - 0.2)$. Error bars indicate Monte Carlo standard errors.

F.9 Clustering for $k = 2$, Binary Data

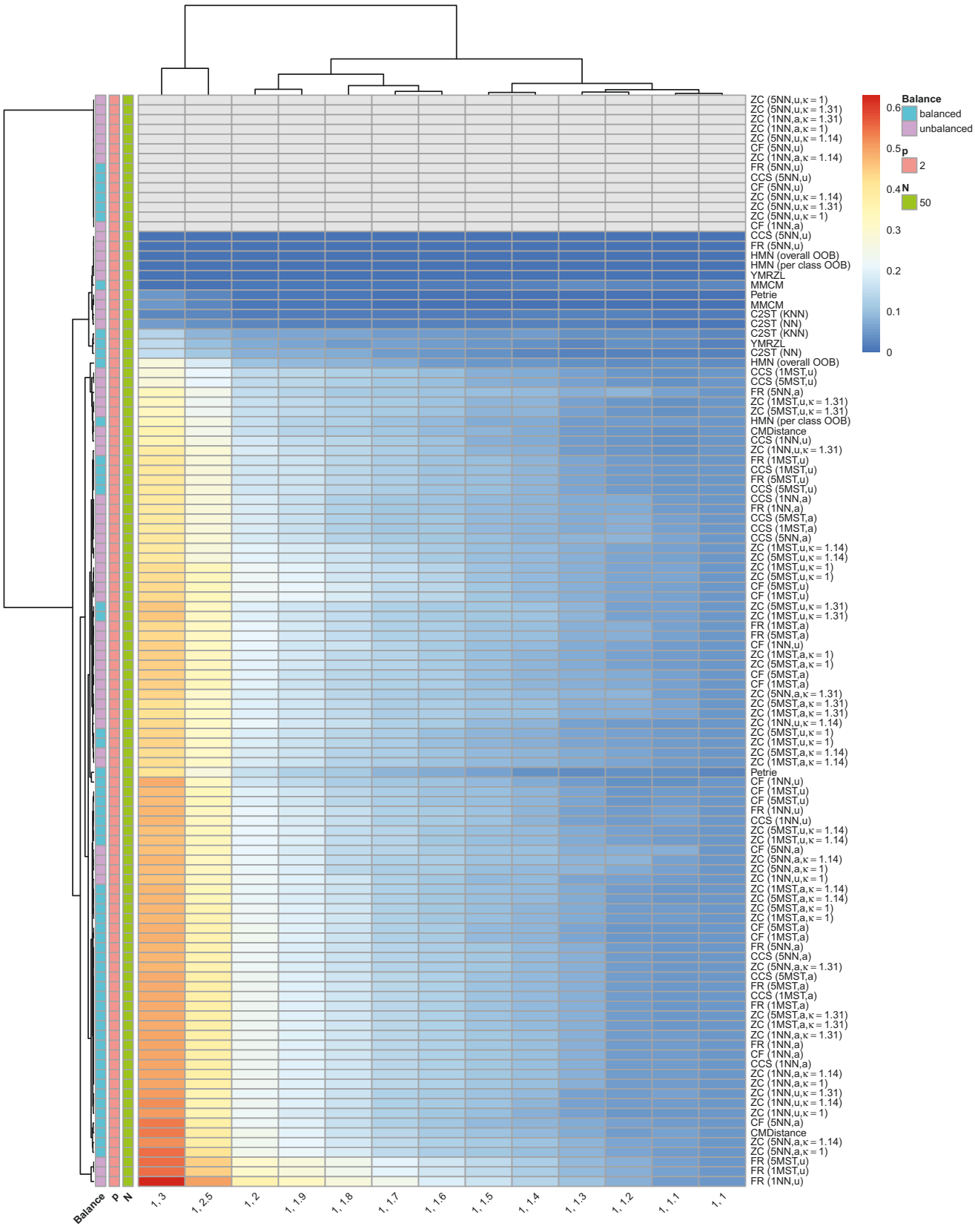


Figure 72: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 50$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

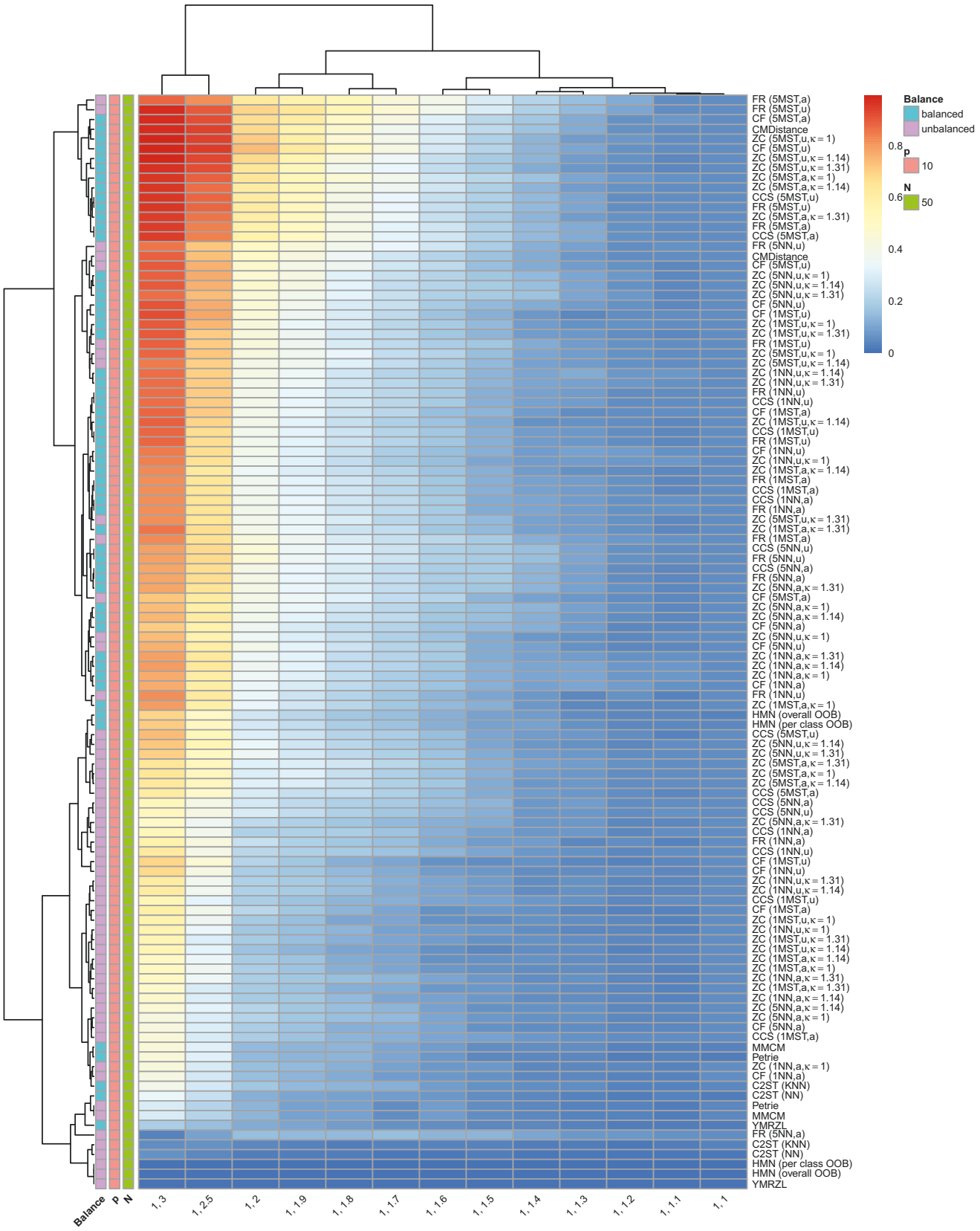


Figure 73: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 50$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

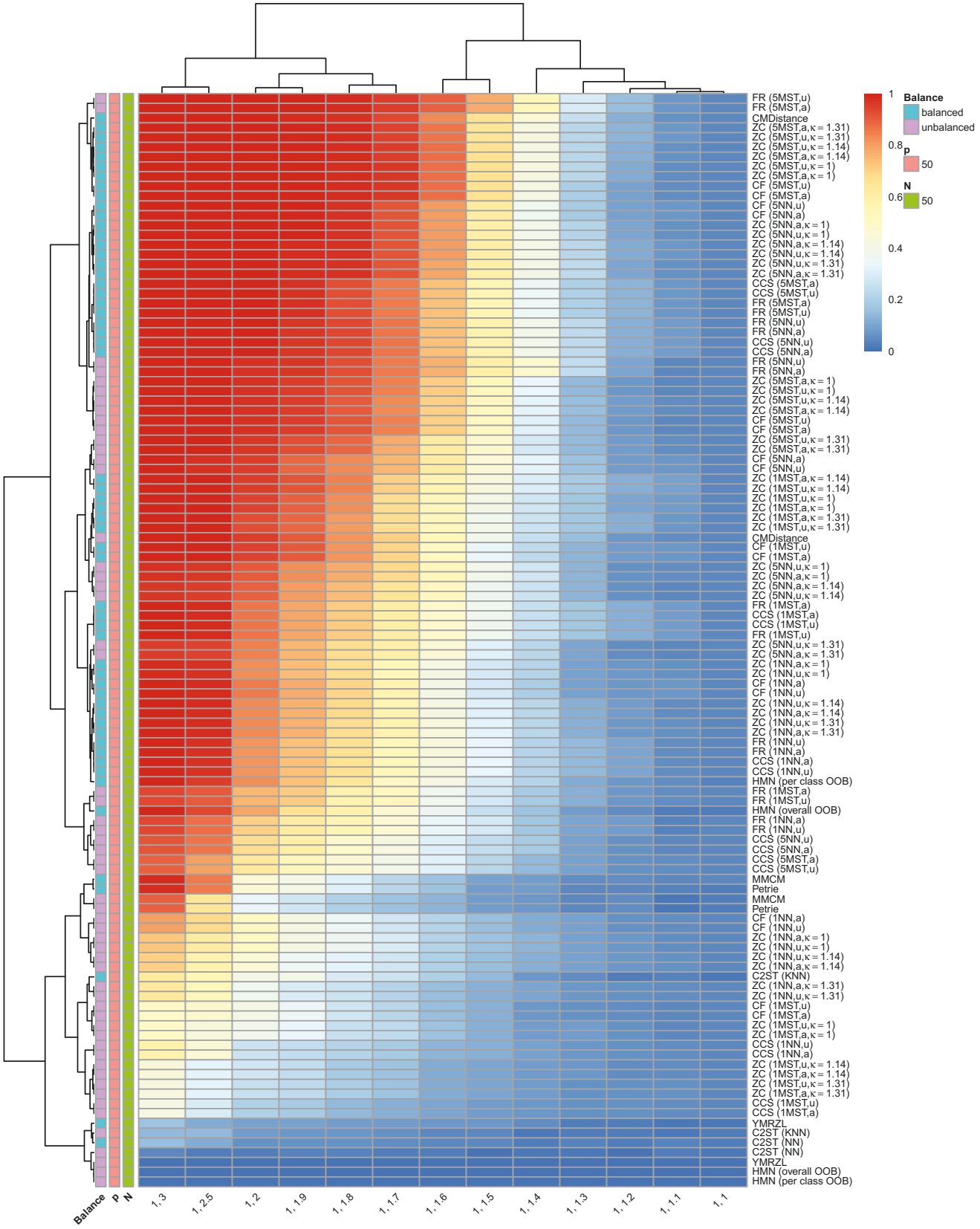


Figure 74: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 50$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

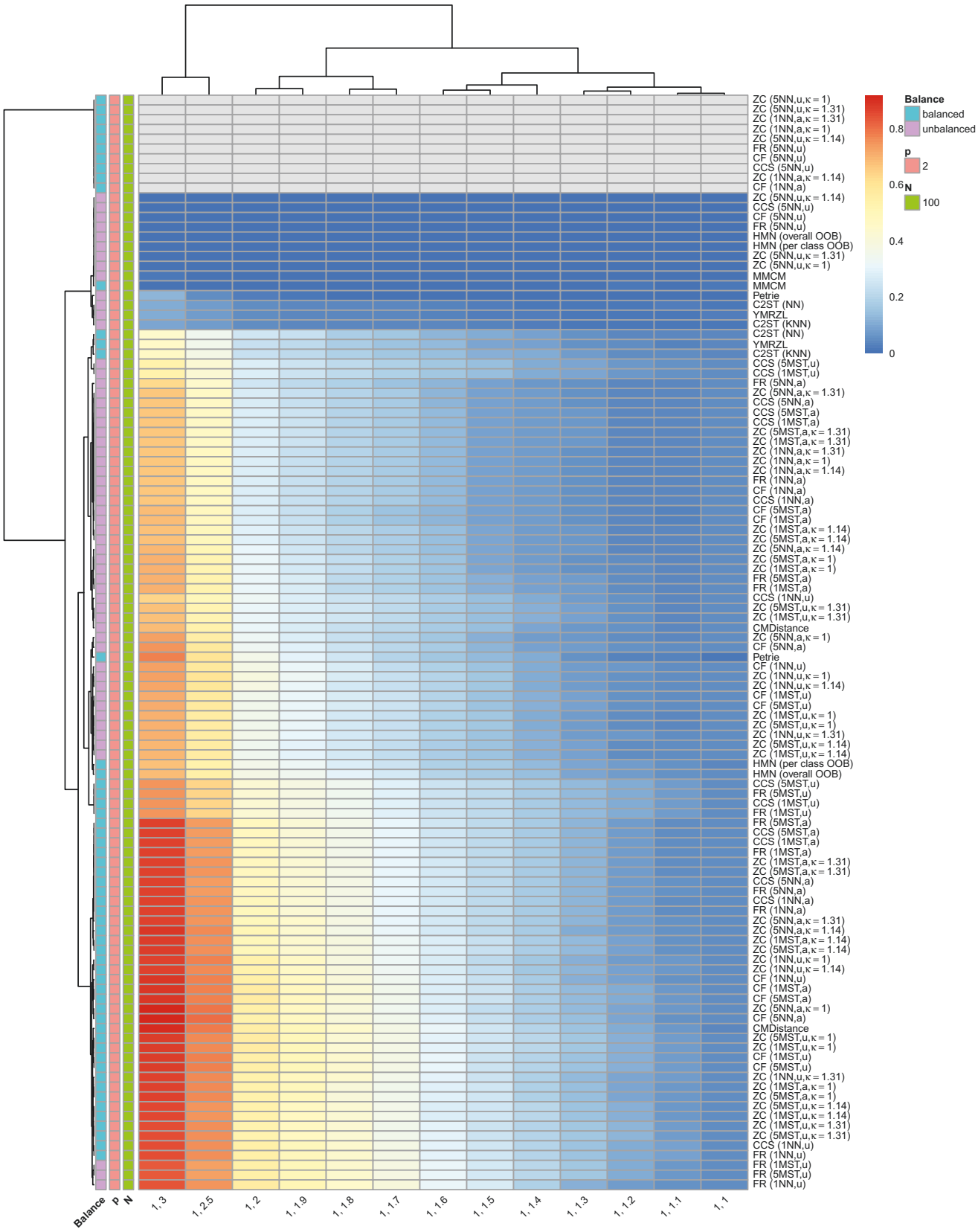


Figure 75: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 100$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

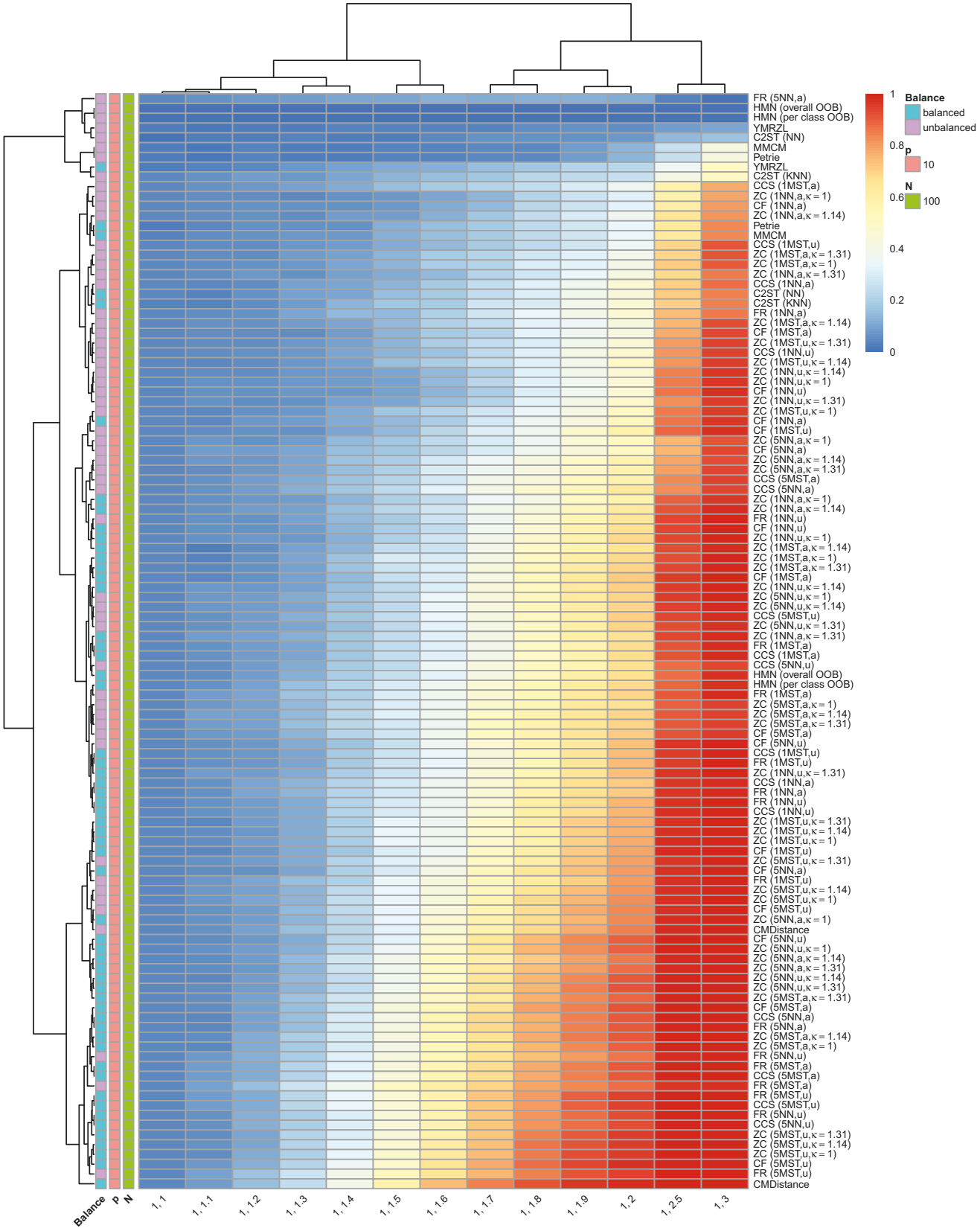


Figure 76: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 100$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

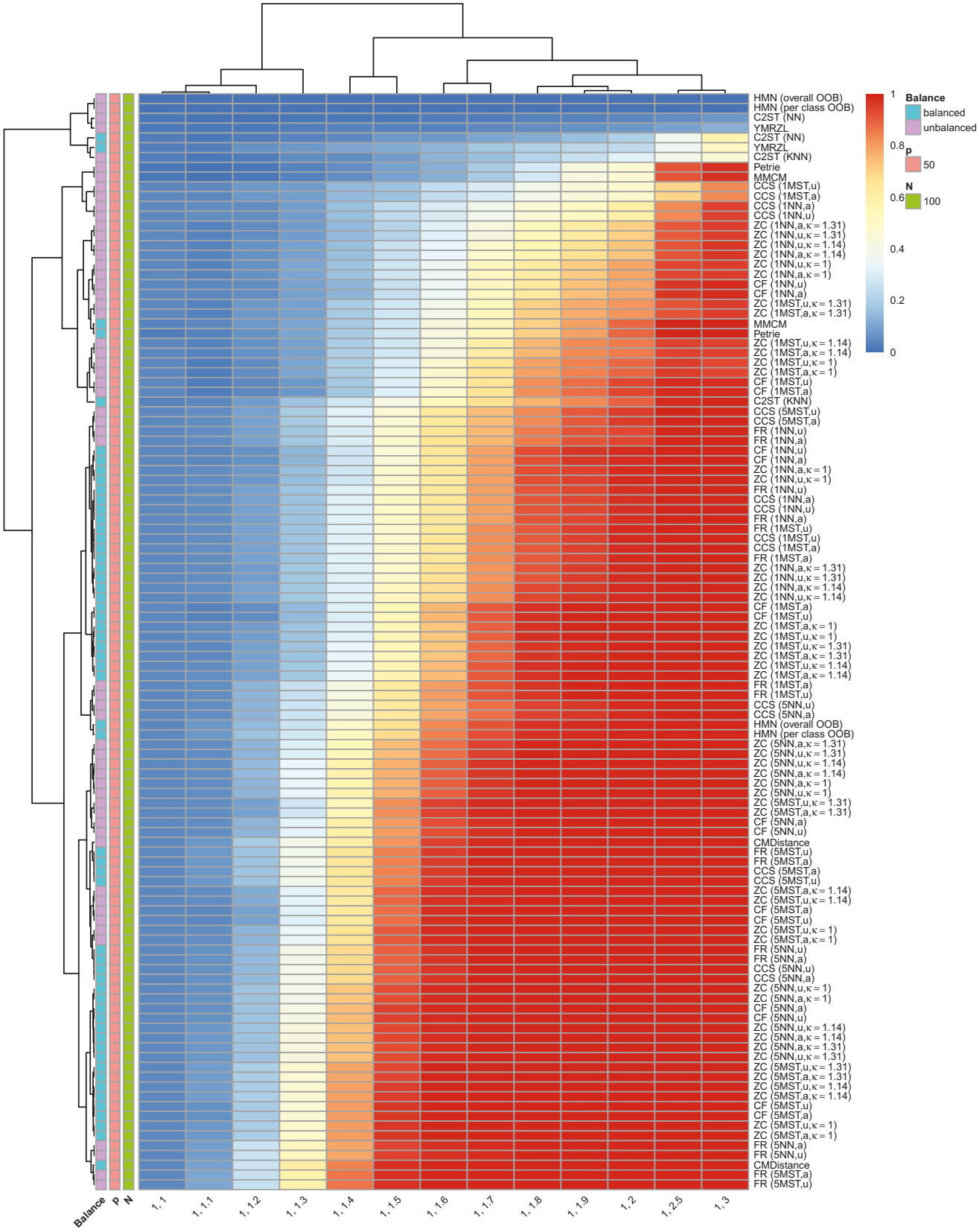


Figure 77: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 100$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

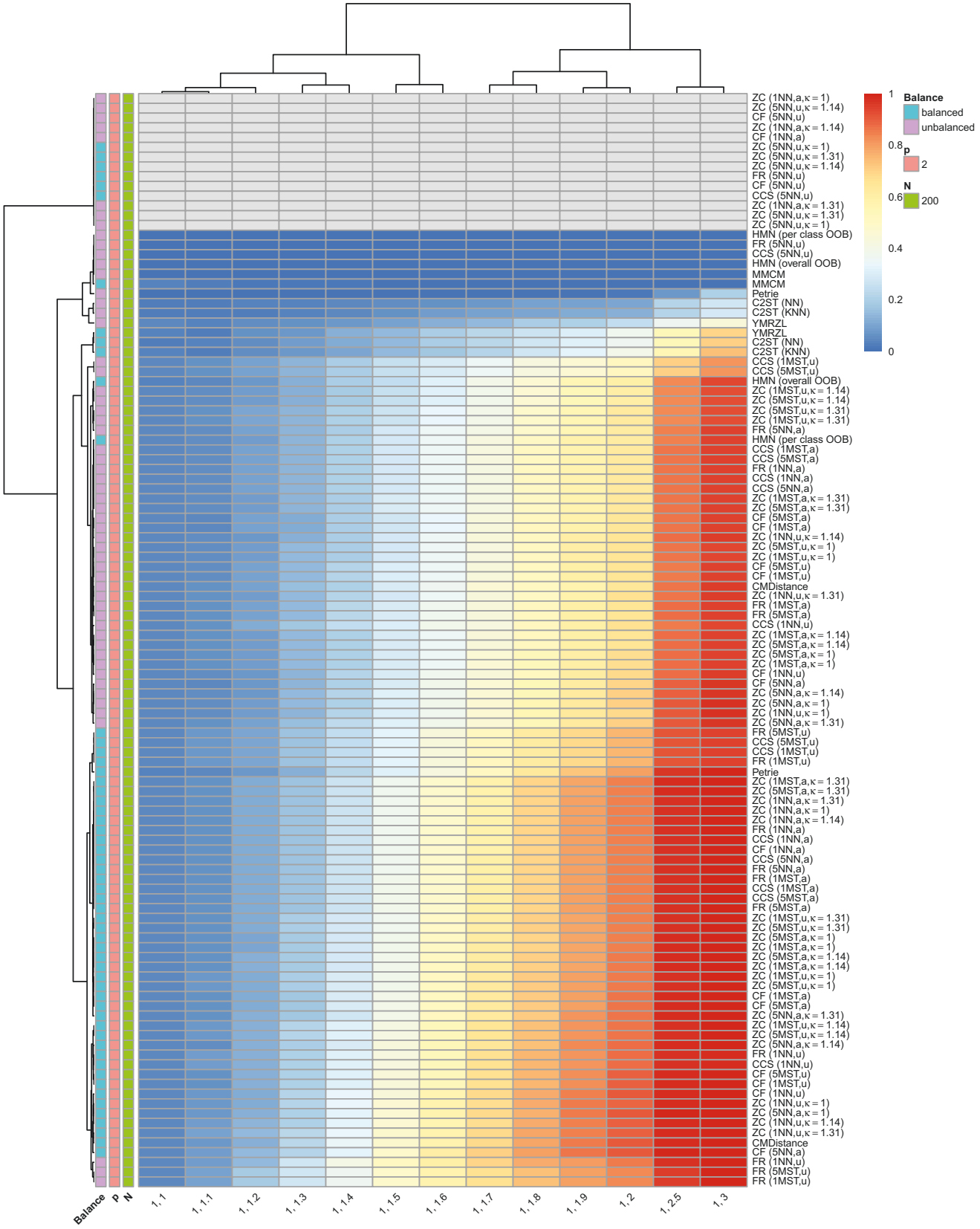


Figure 78: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 200$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

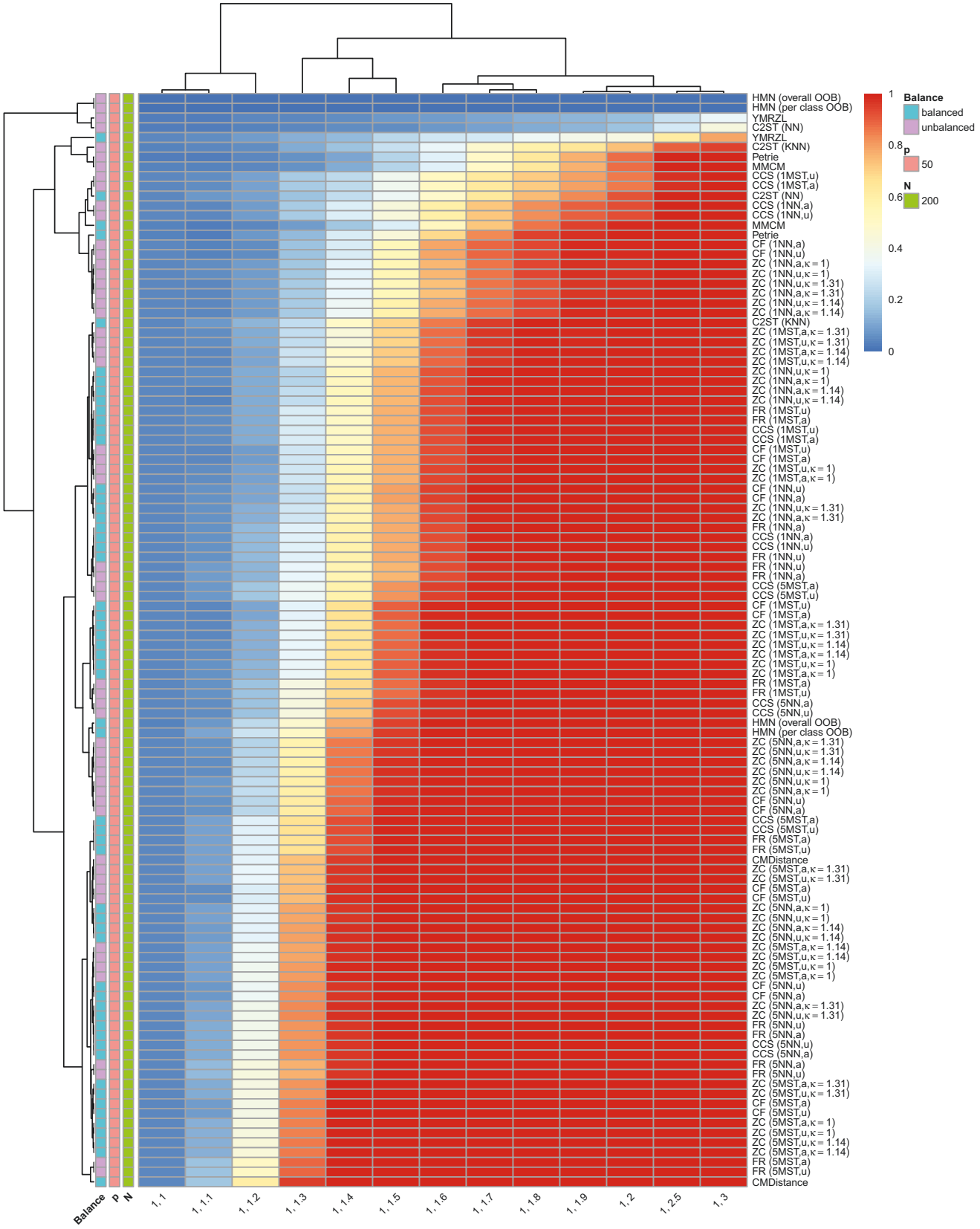


Figure 79: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 200$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

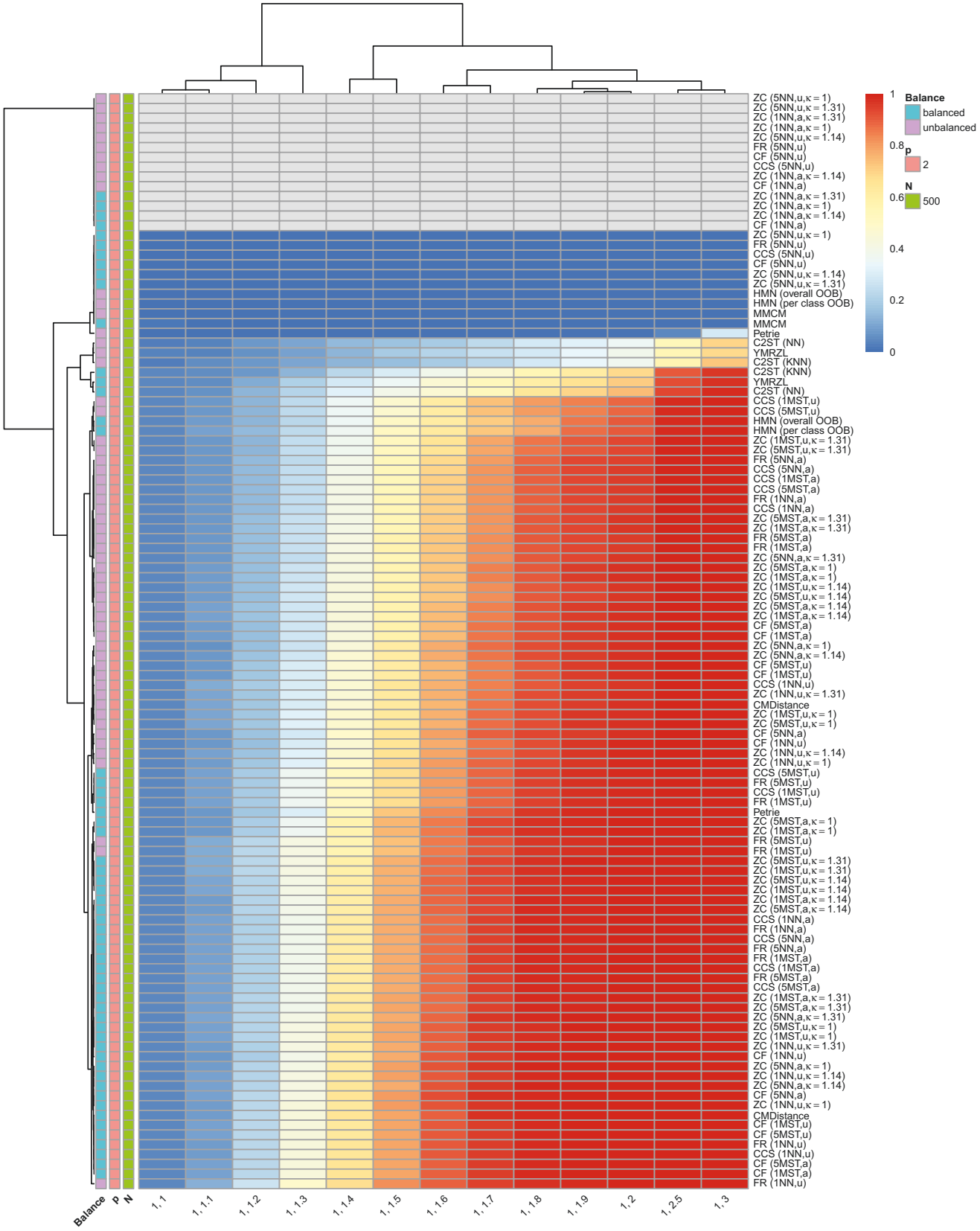


Figure 80: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 500$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

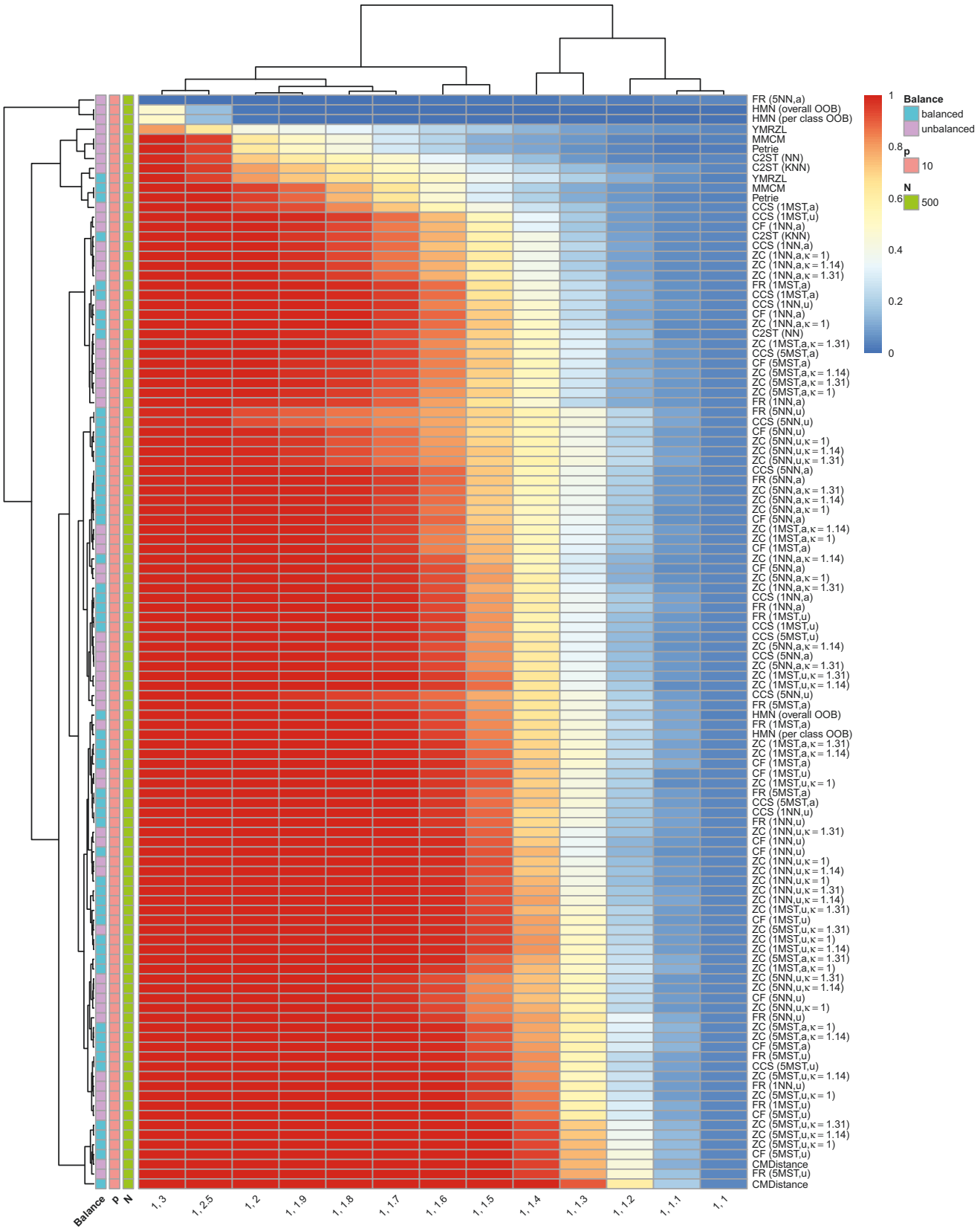


Figure 81: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 500$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

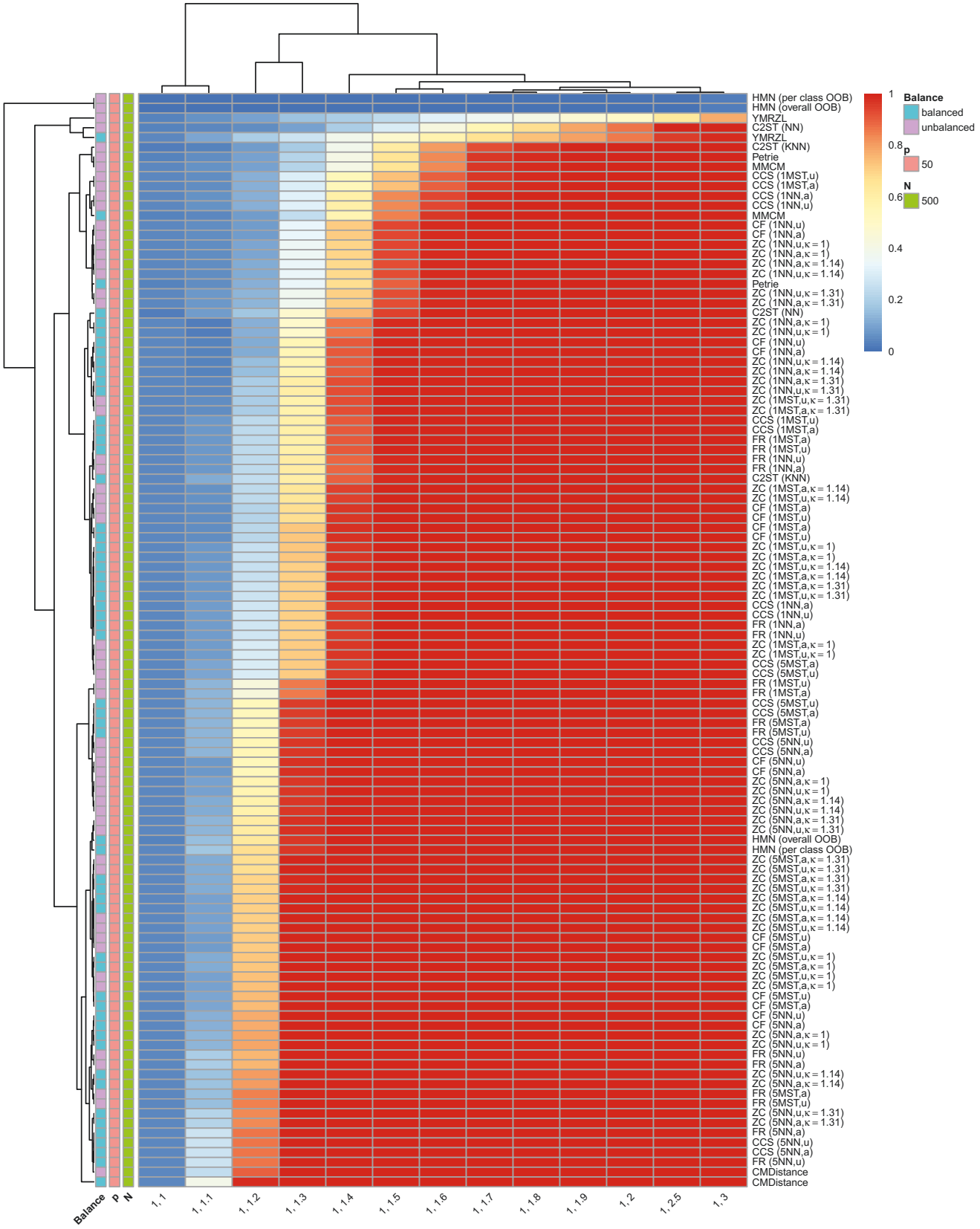


Figure 82: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 500$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

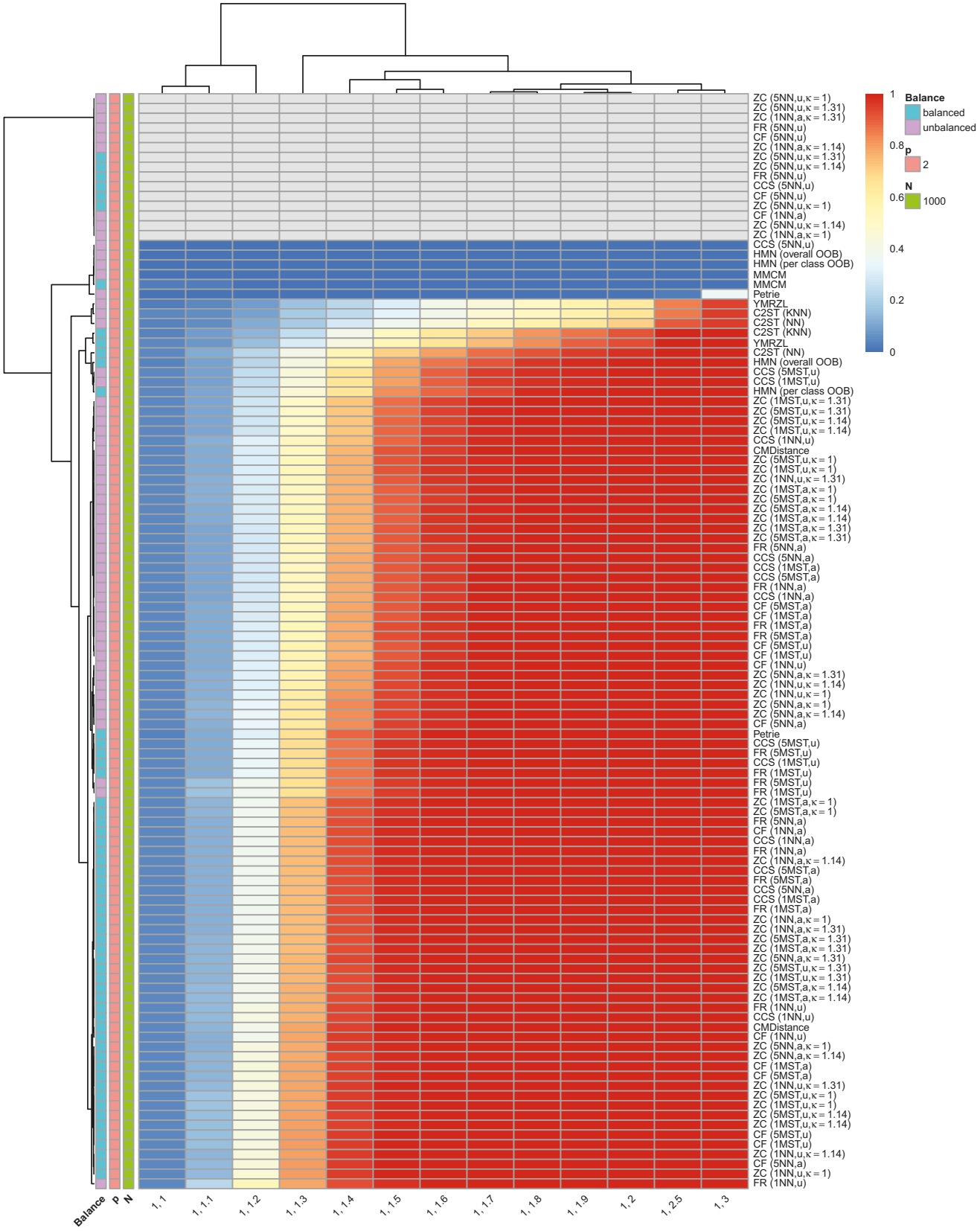


Figure 83: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 1000$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

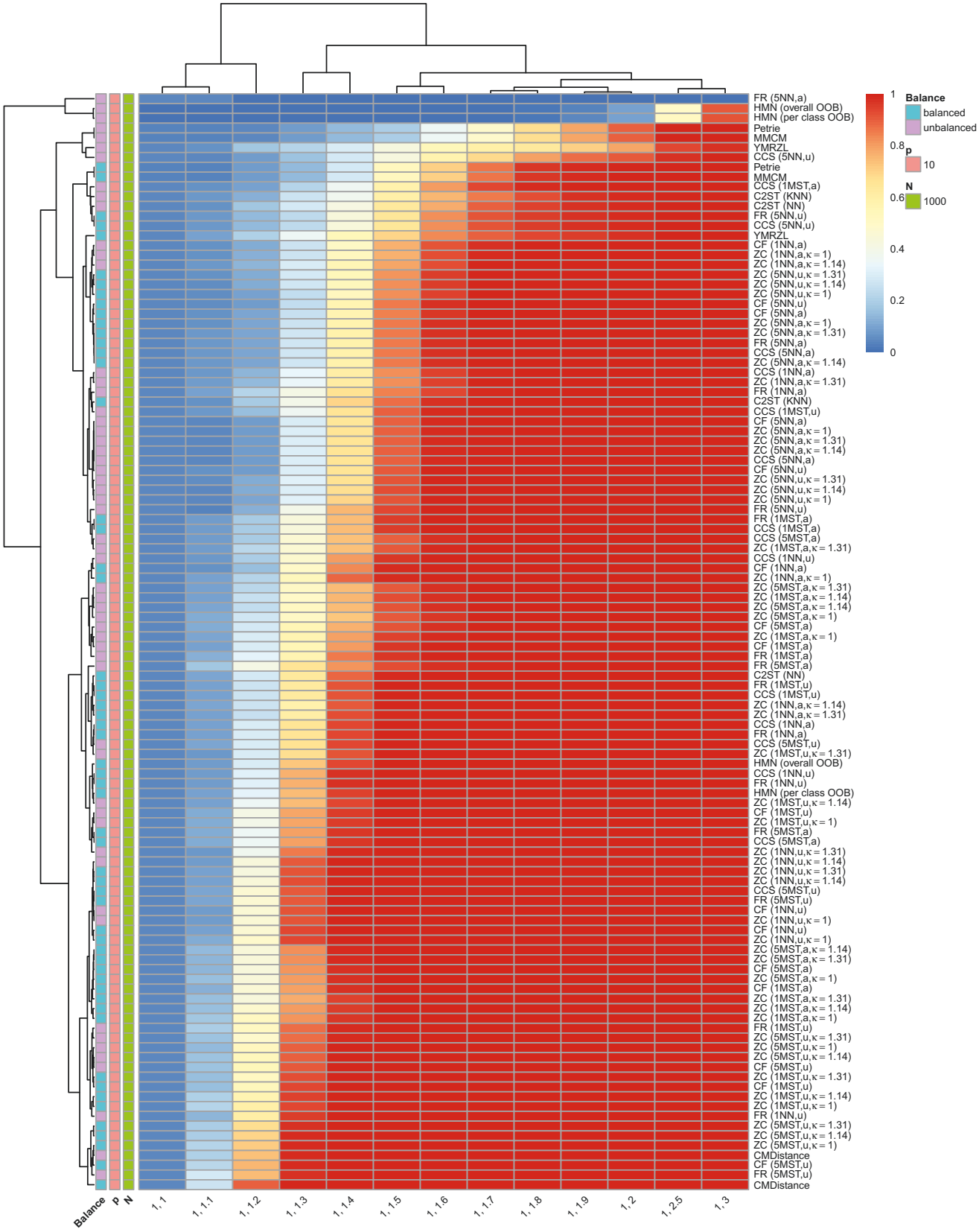


Figure 84: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 1000$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

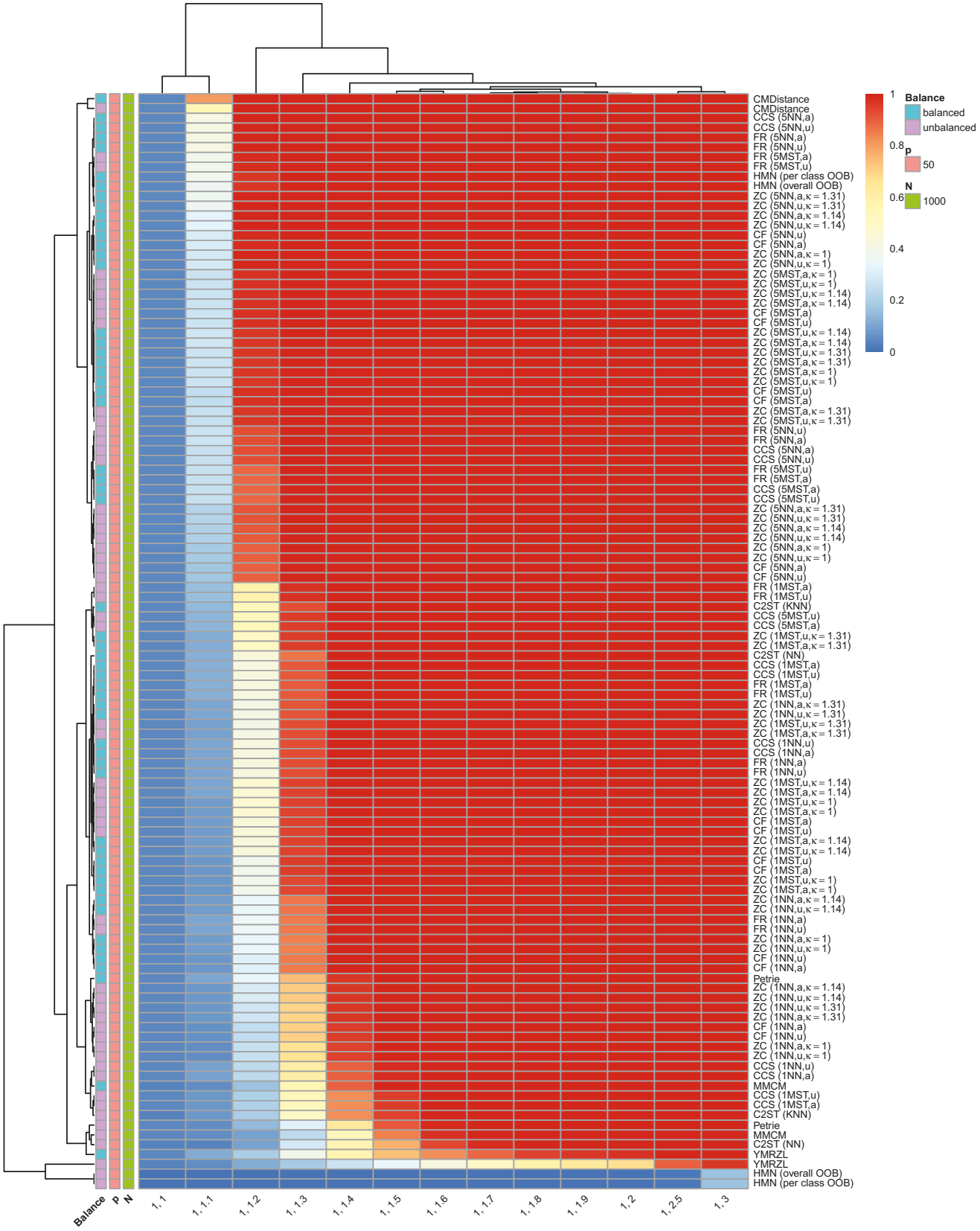


Figure 85: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two binary datasets with $N = 1000$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

F.10 Clustering for $k = 2$, Multinomial Data

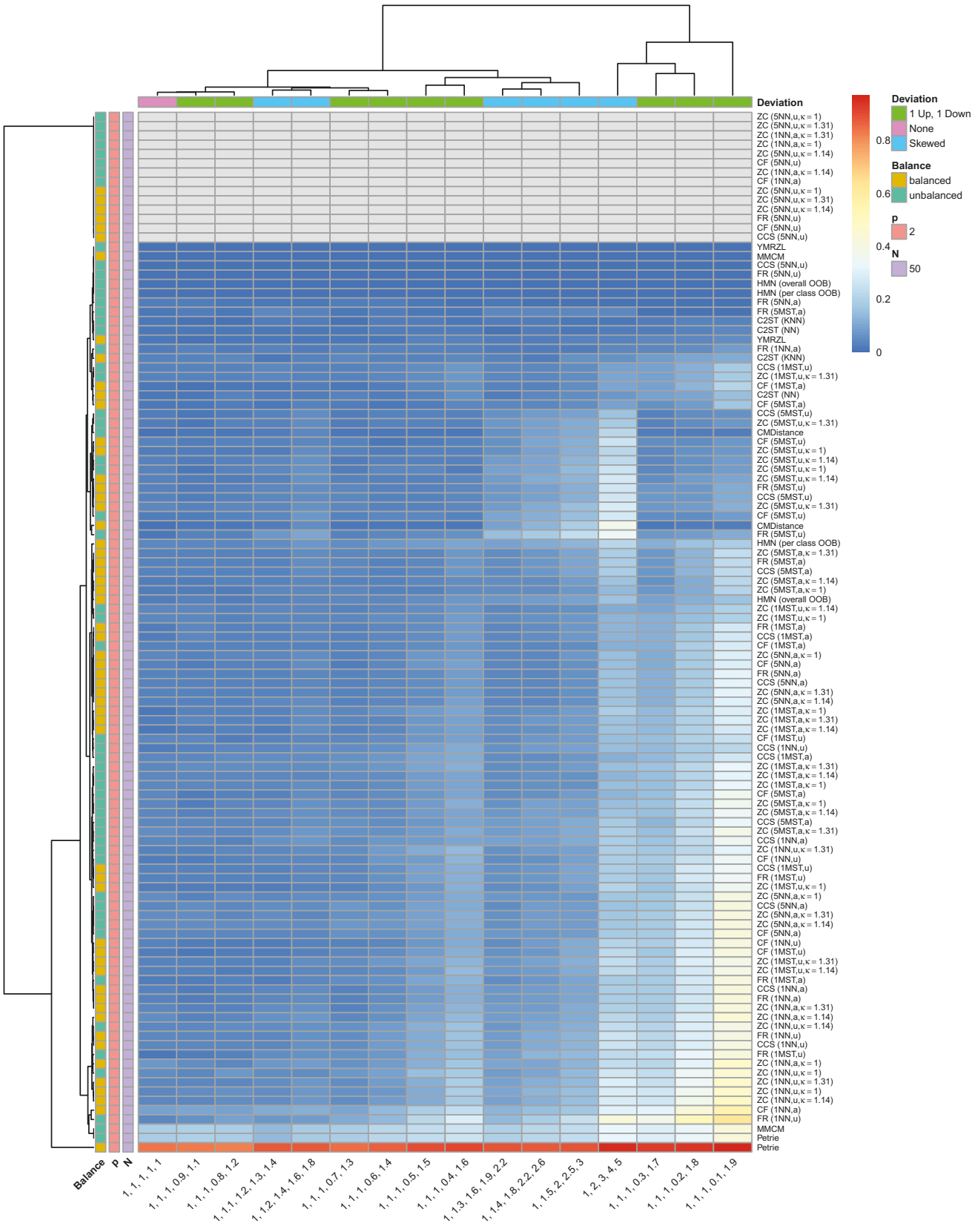


Figure 86: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 50$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

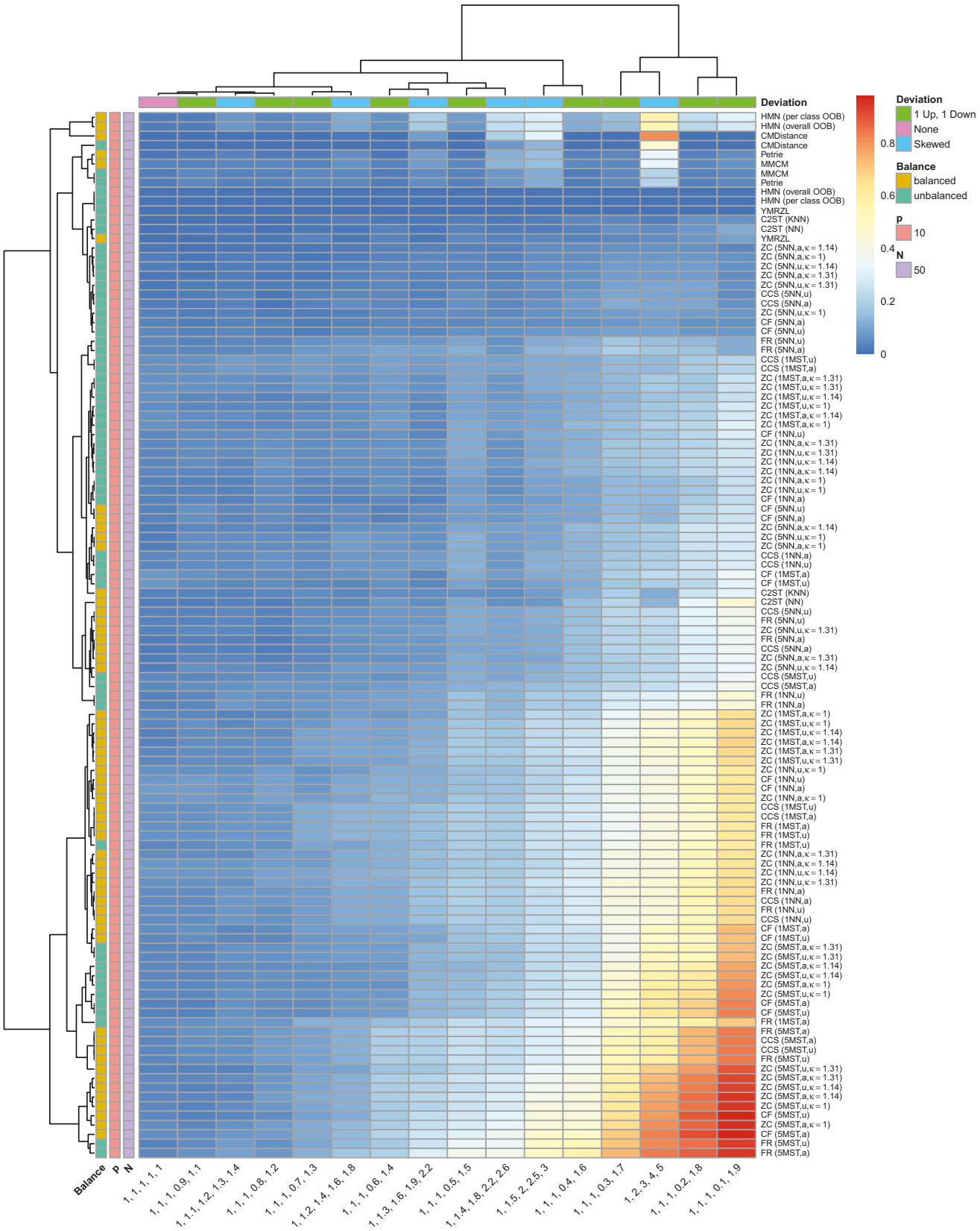


Figure 87: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 50$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

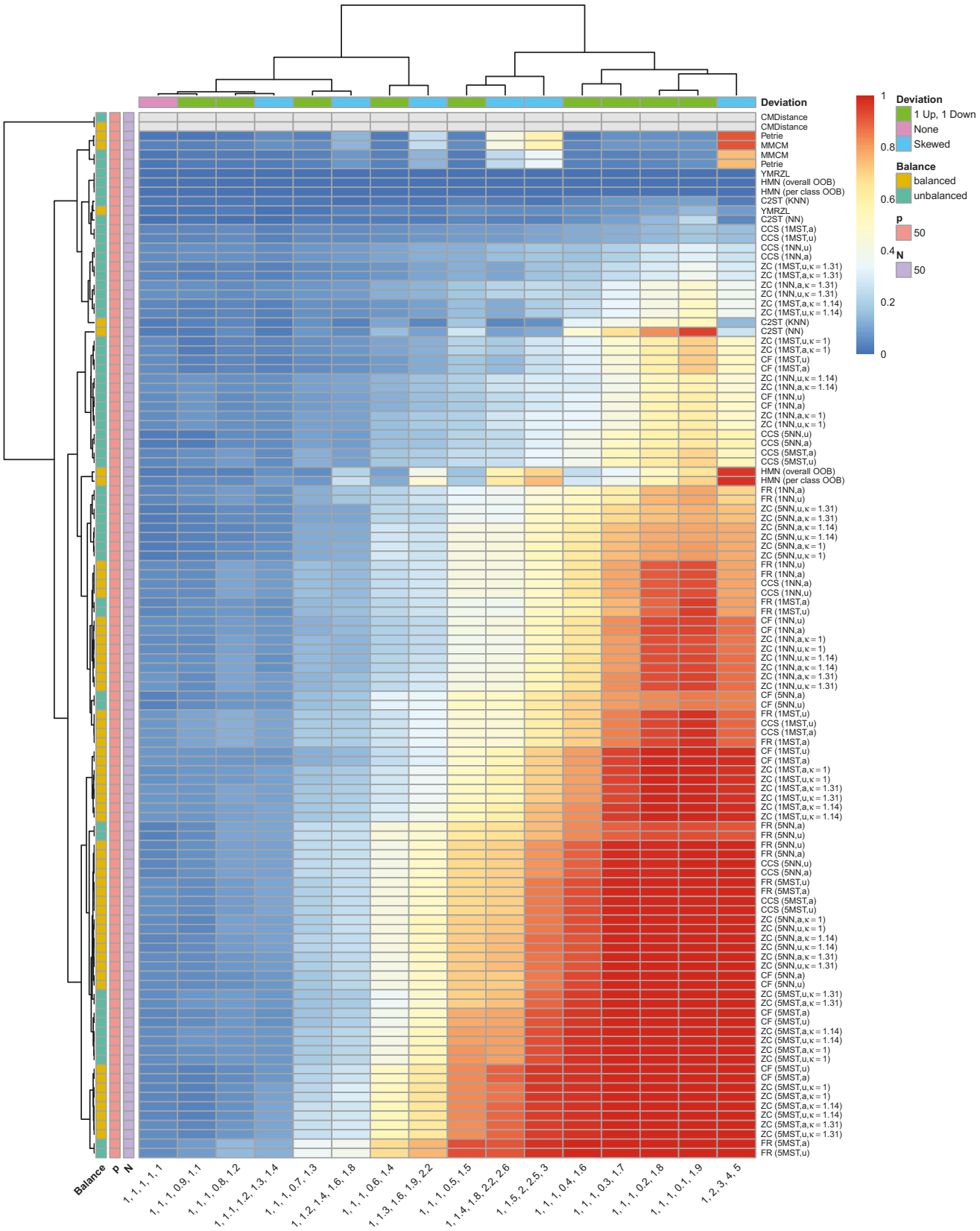


Figure 88: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 50$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

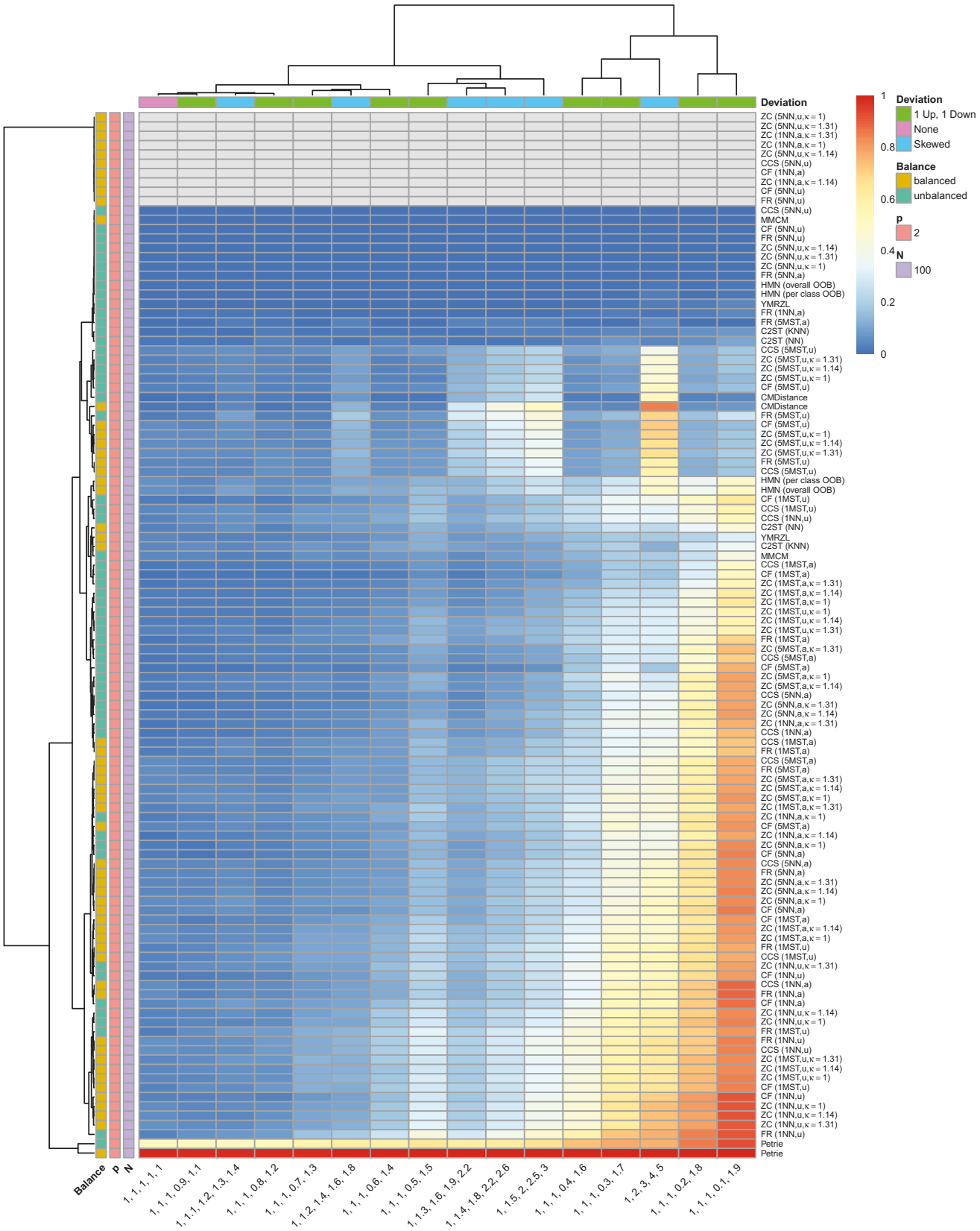


Figure 89: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 100$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

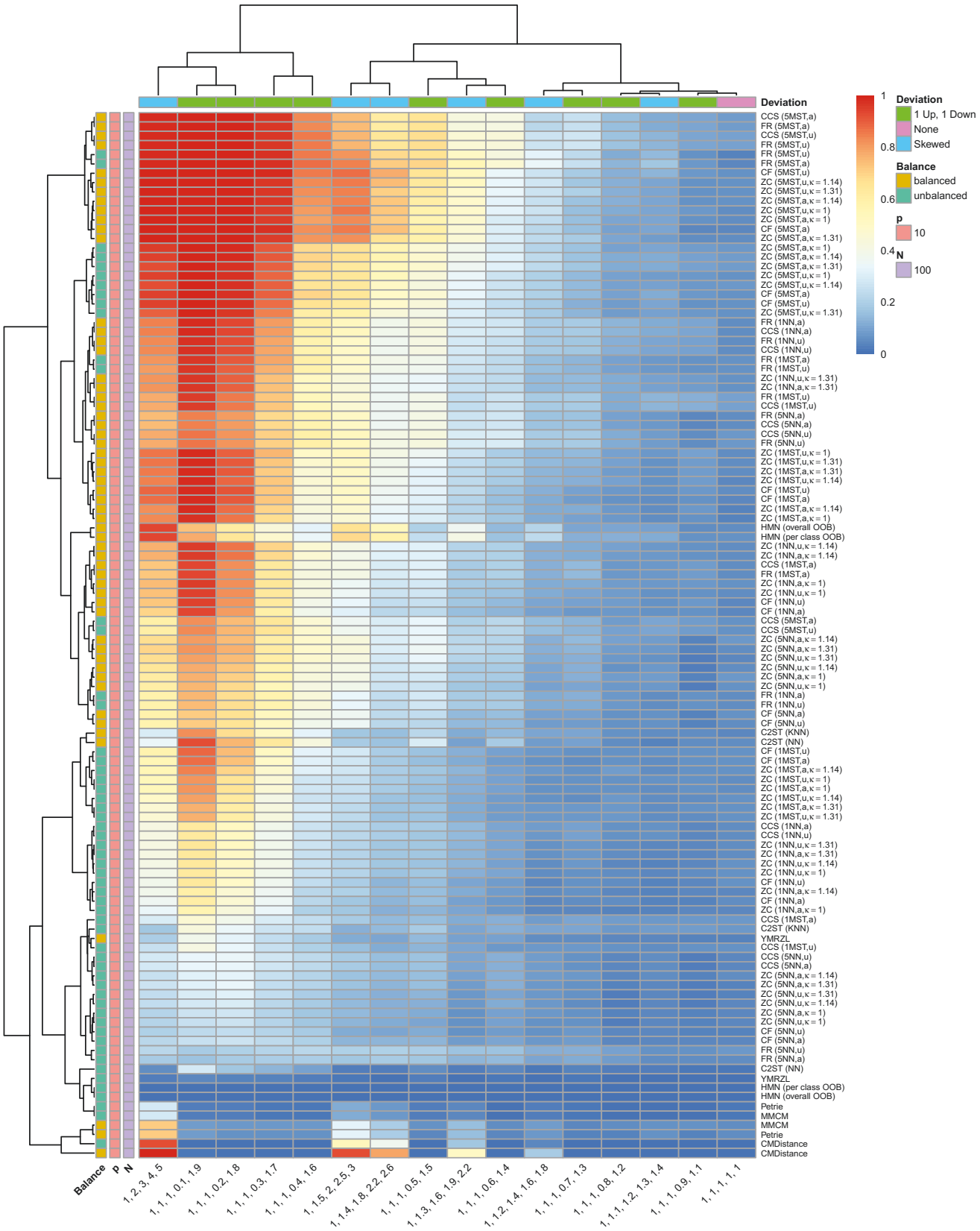


Figure 90: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 100$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

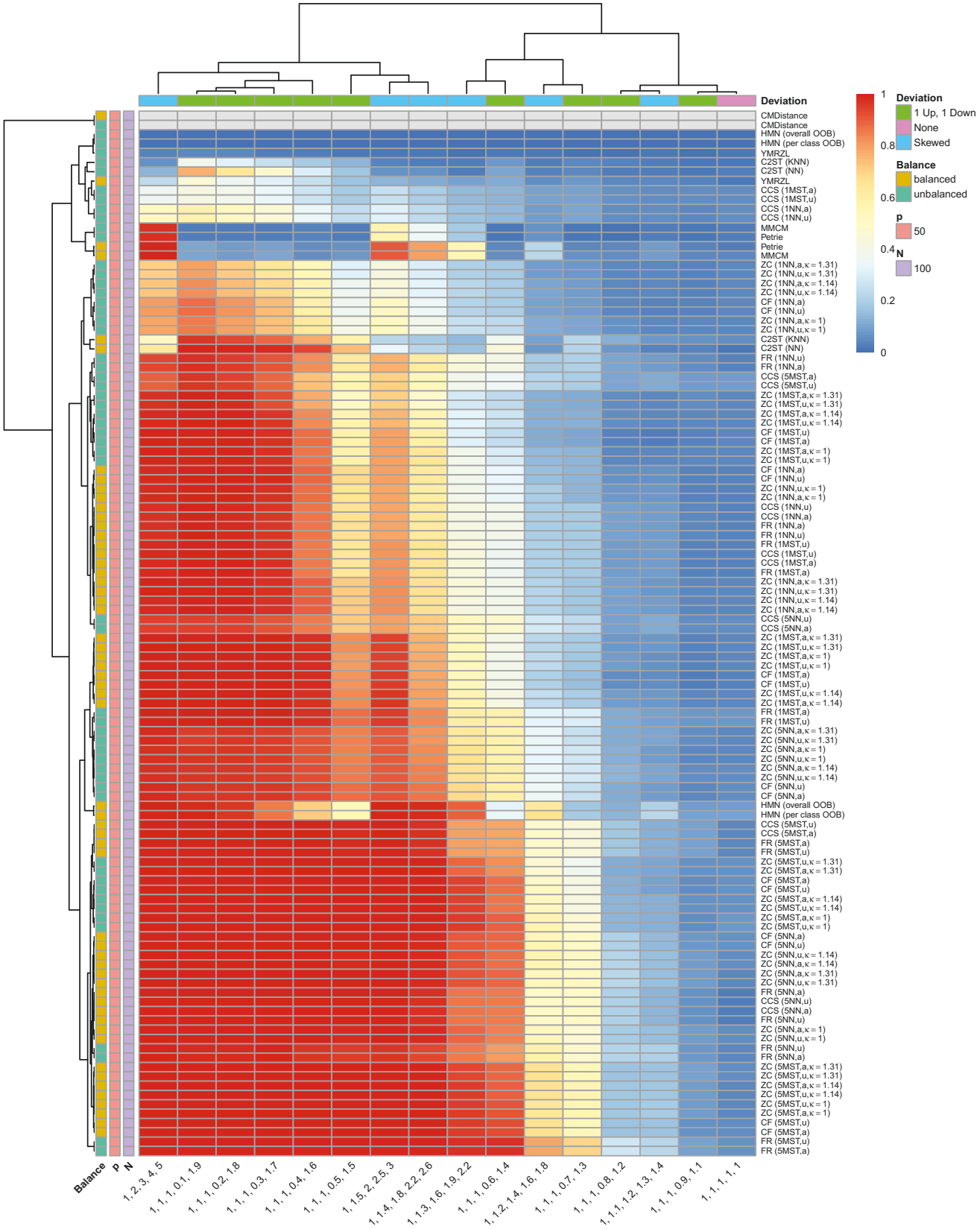


Figure 91: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 100$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

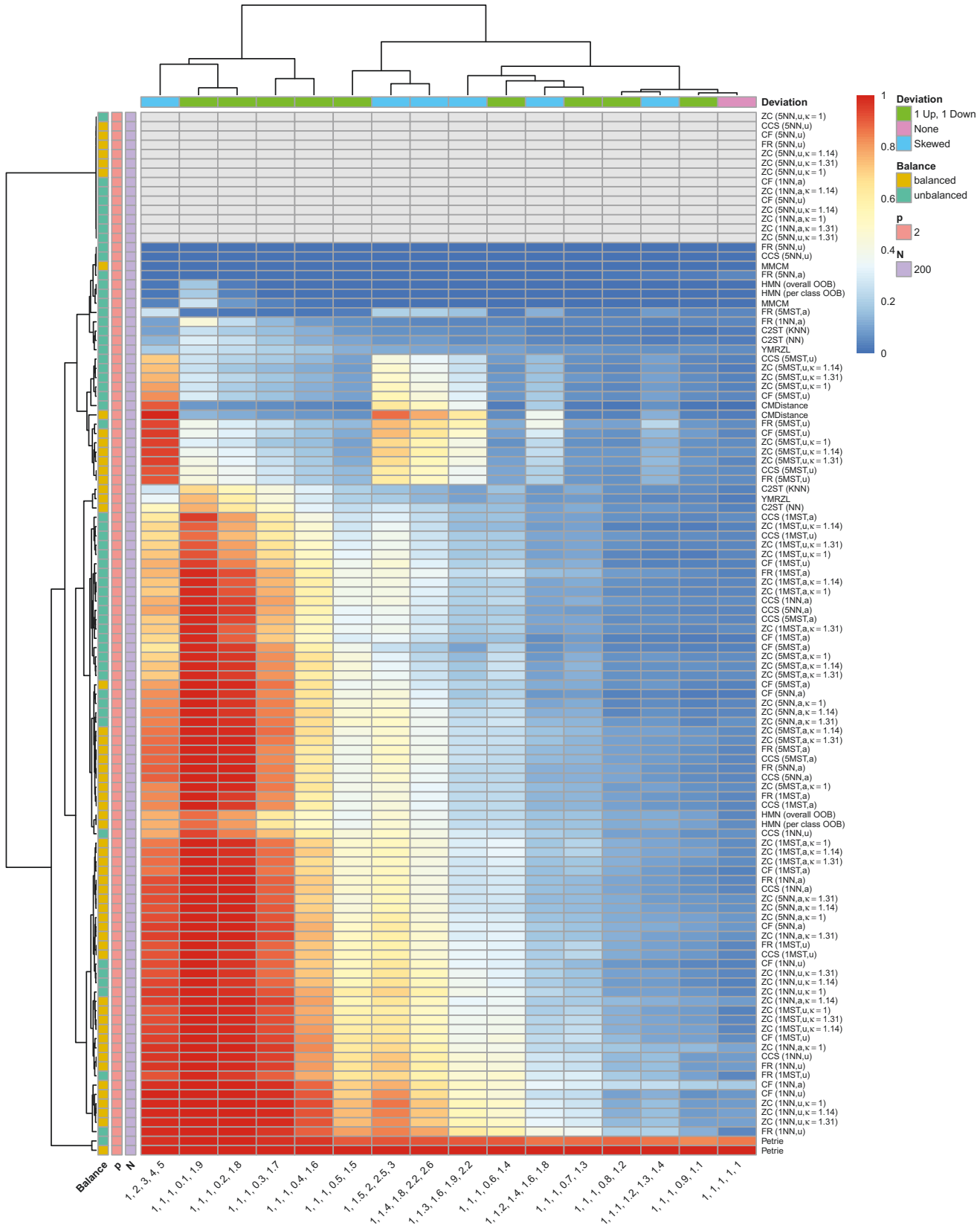


Figure 92: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 200$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

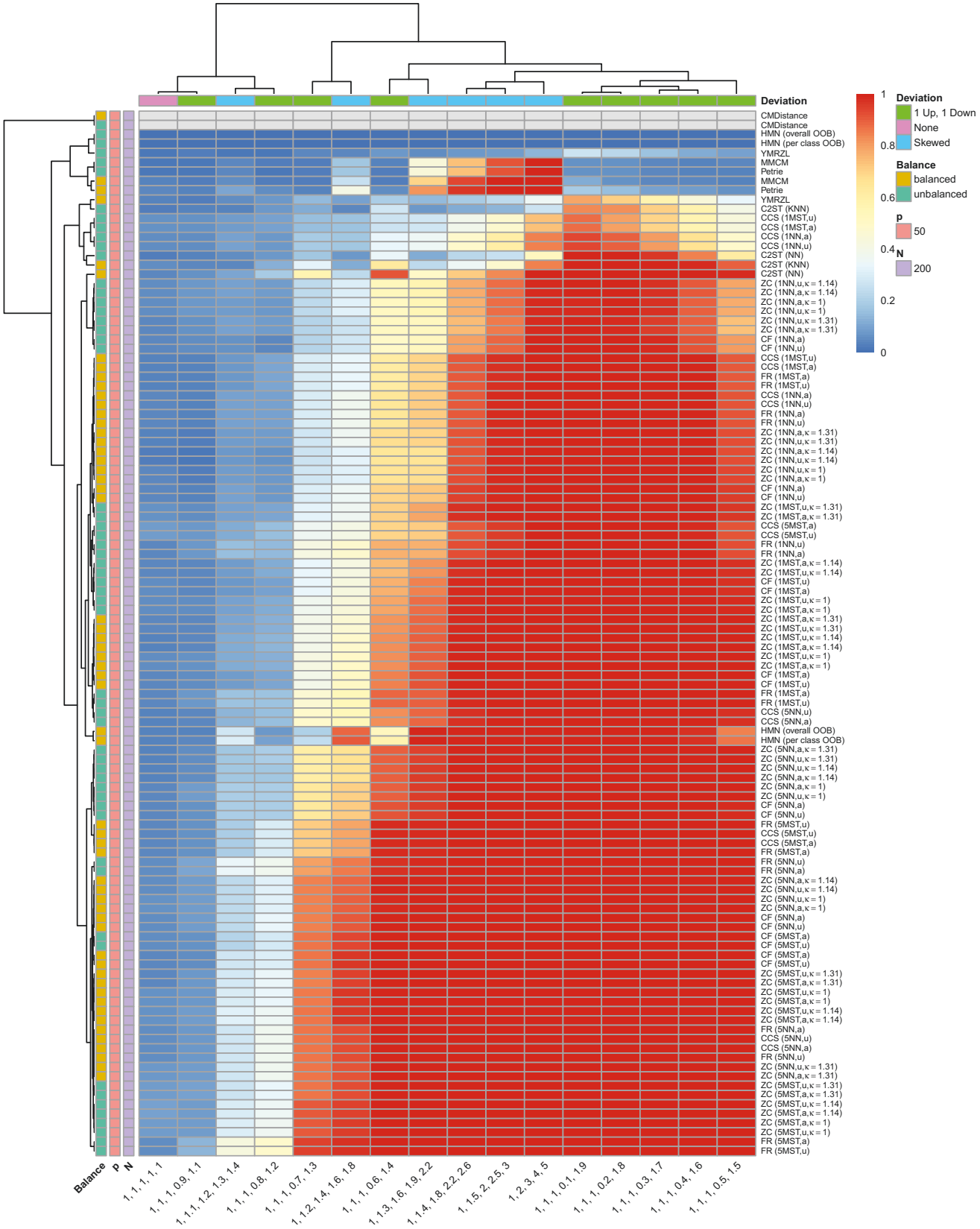


Figure 93: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 200$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

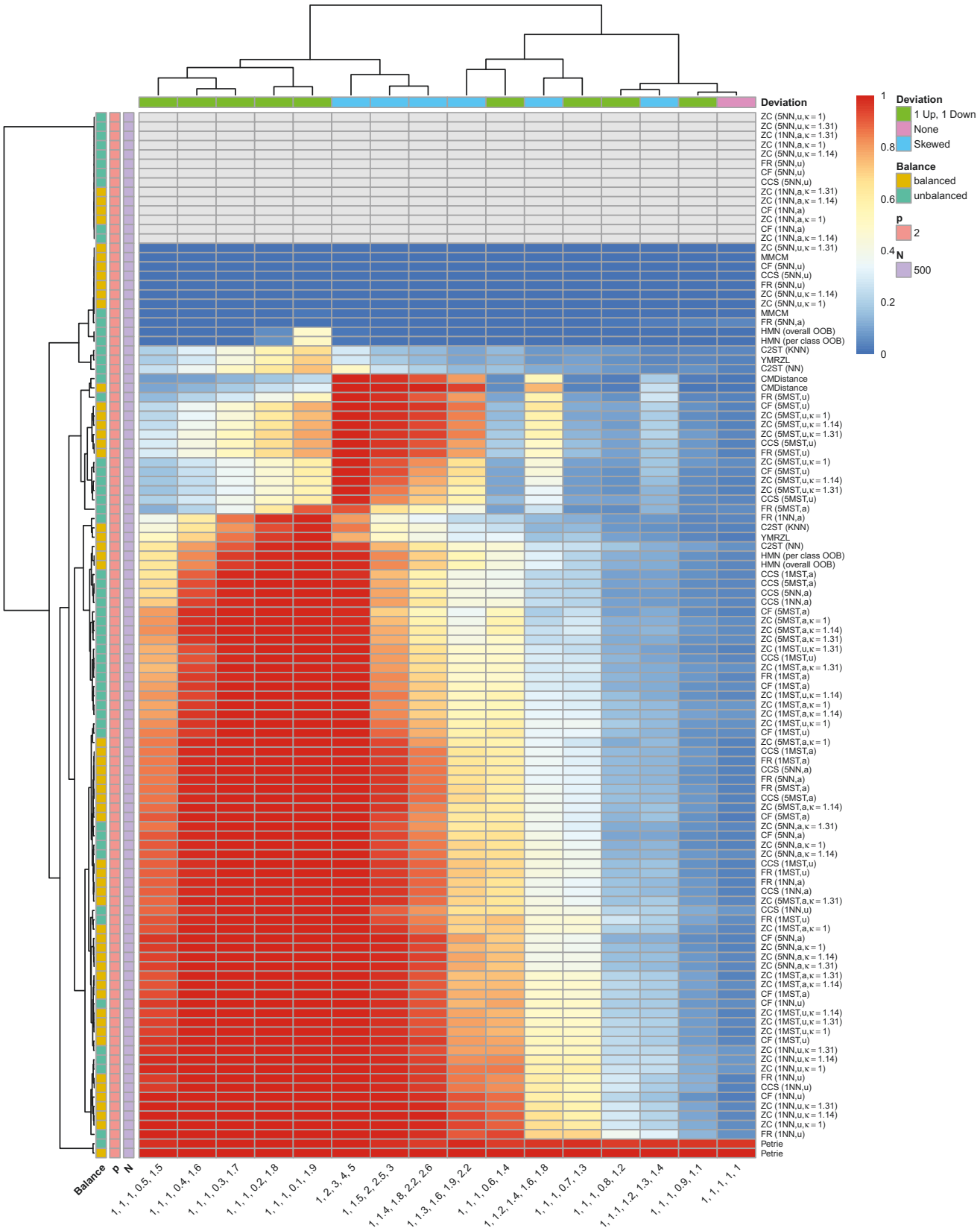


Figure 94: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 500$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

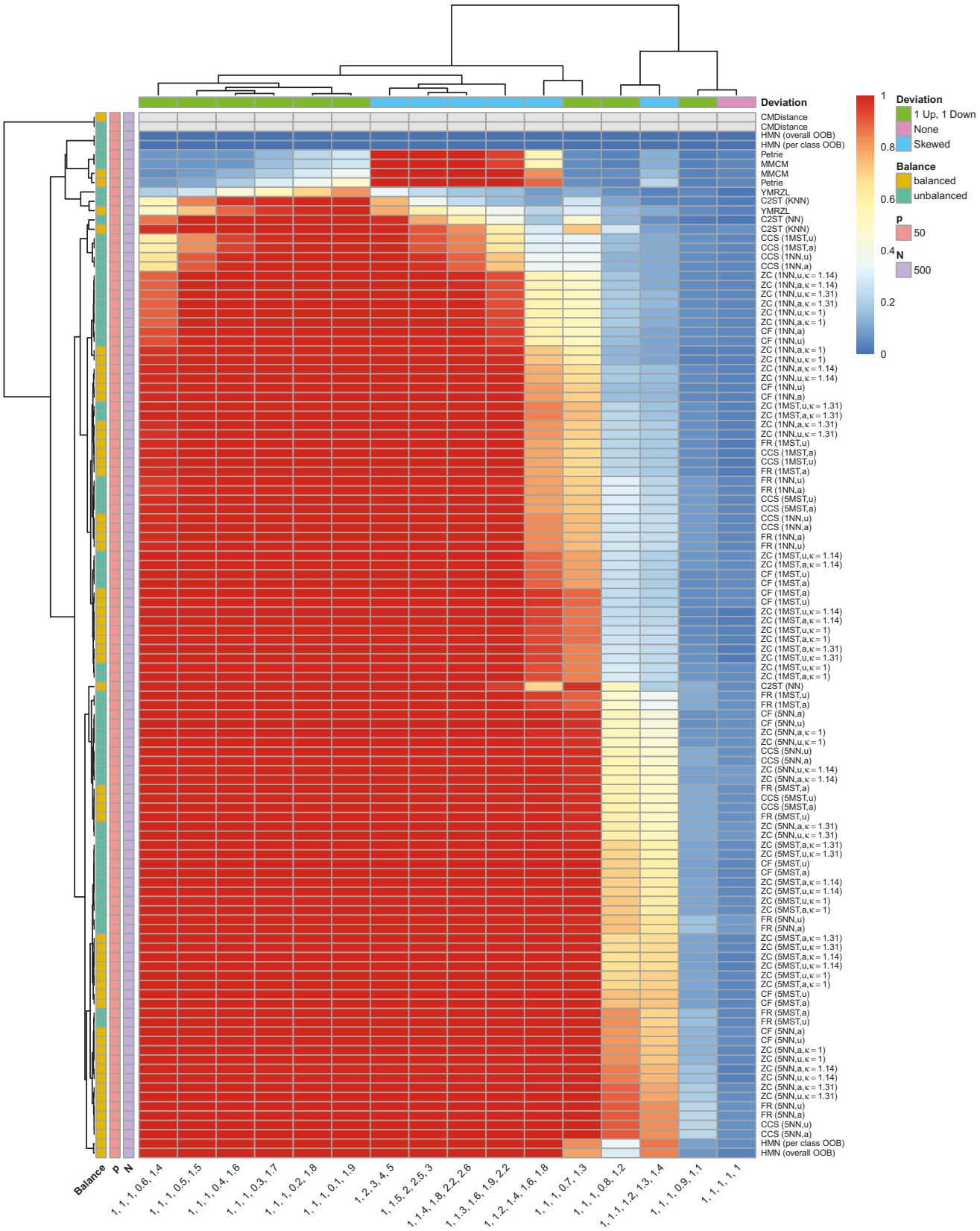


Figure 95: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 500$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

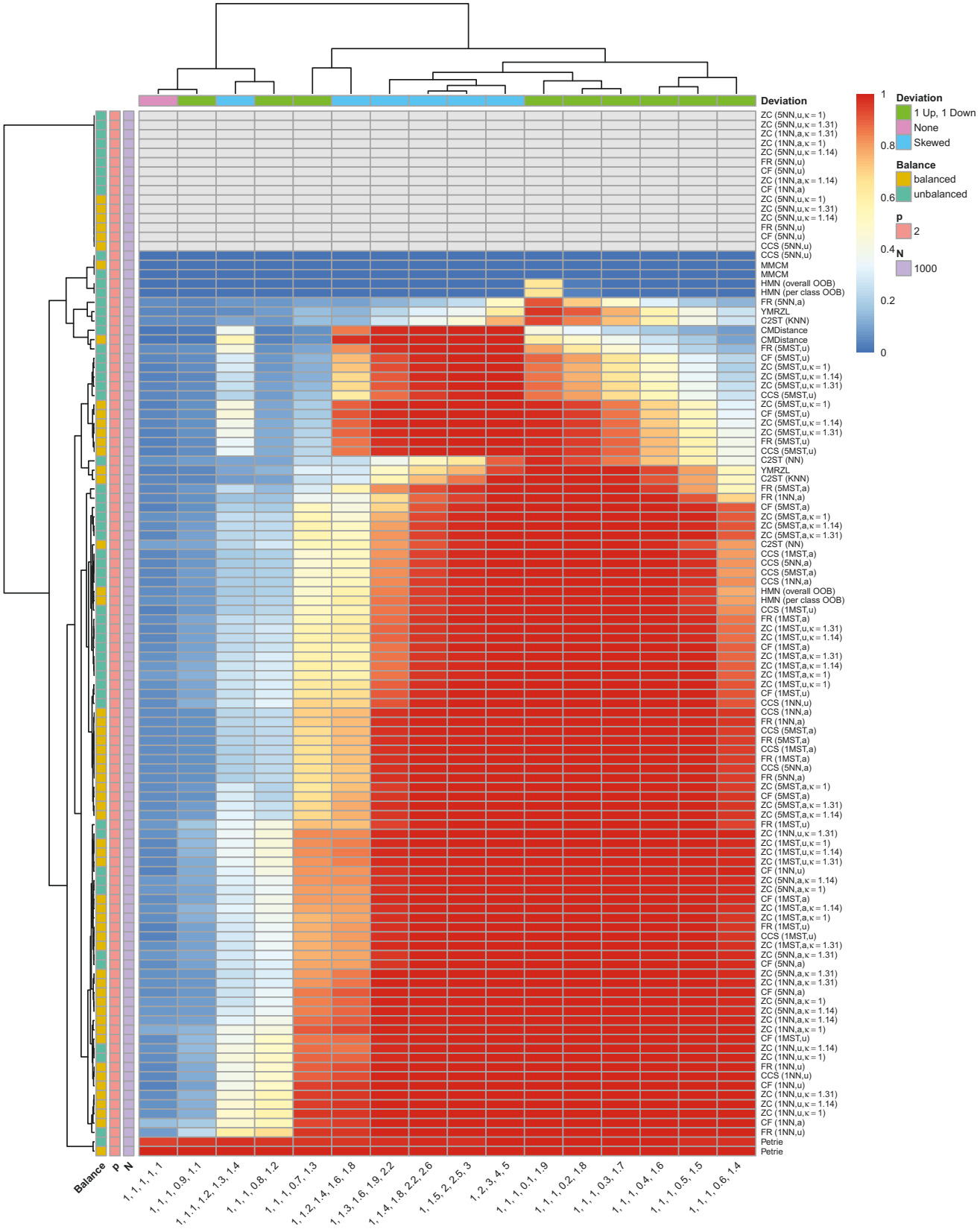


Figure 96: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 1000$ and $p = 2$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

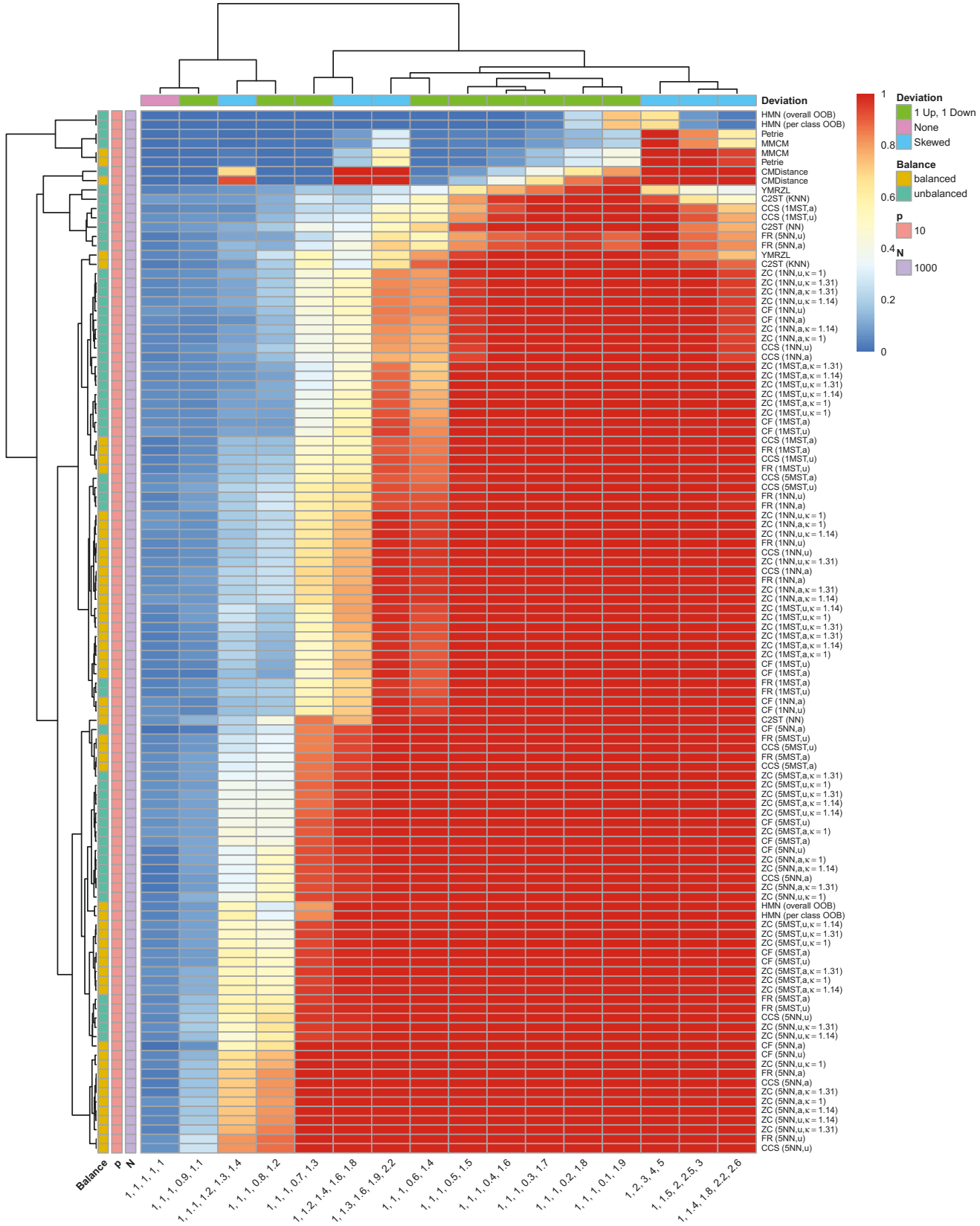


Figure 97: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 1000$ and $p = 10$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

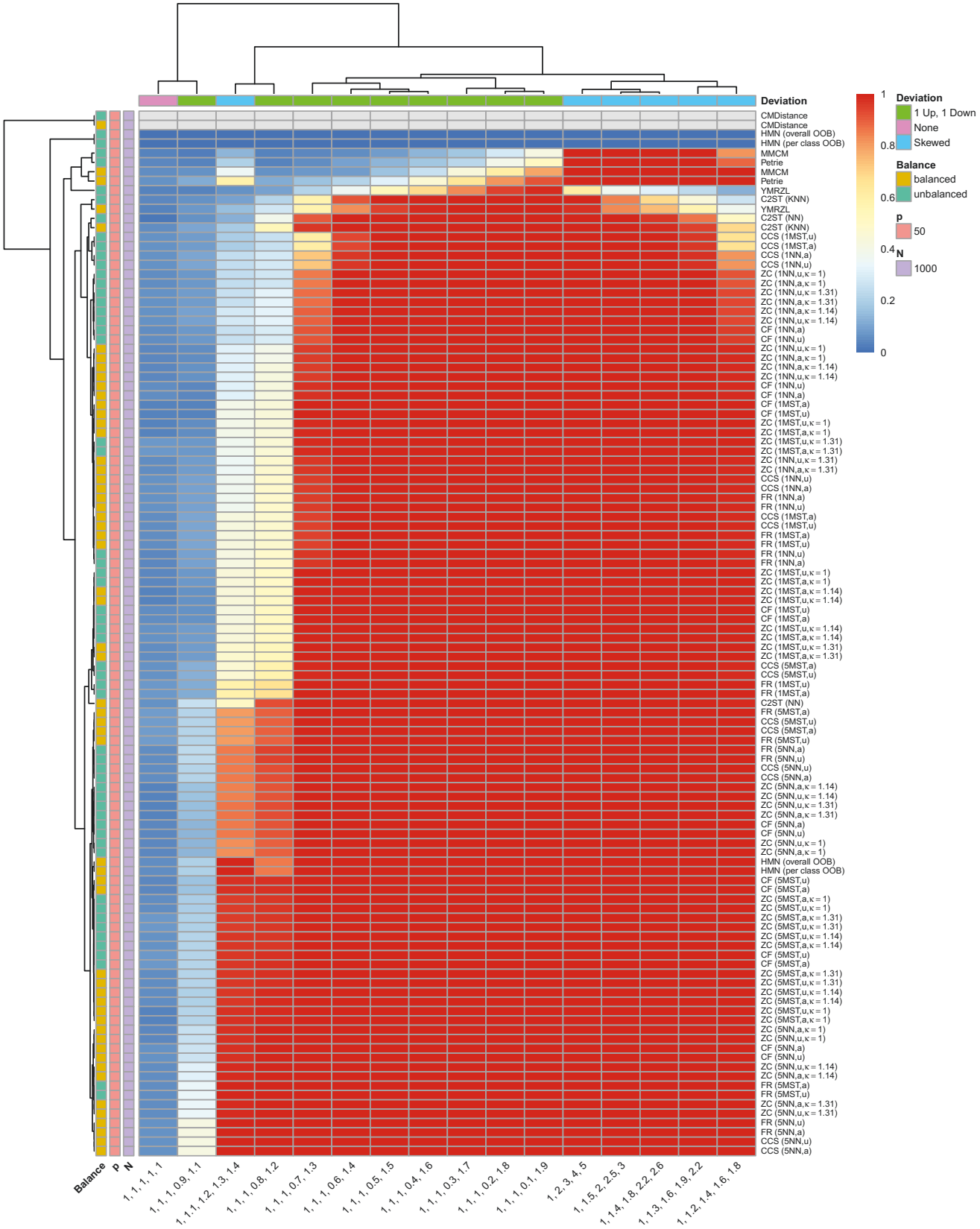


Figure 98: Clustering of PESR values per deviation (x-axis) and per method and sample size balance (y-axis) for two multinomial datasets with $N = 1000$ and $p = 50$. The values on the x-axis give the weight vector (unnormalized class probabilities) of the first deviating dataset.

F.11 Memory and Runtime for $k = 2$

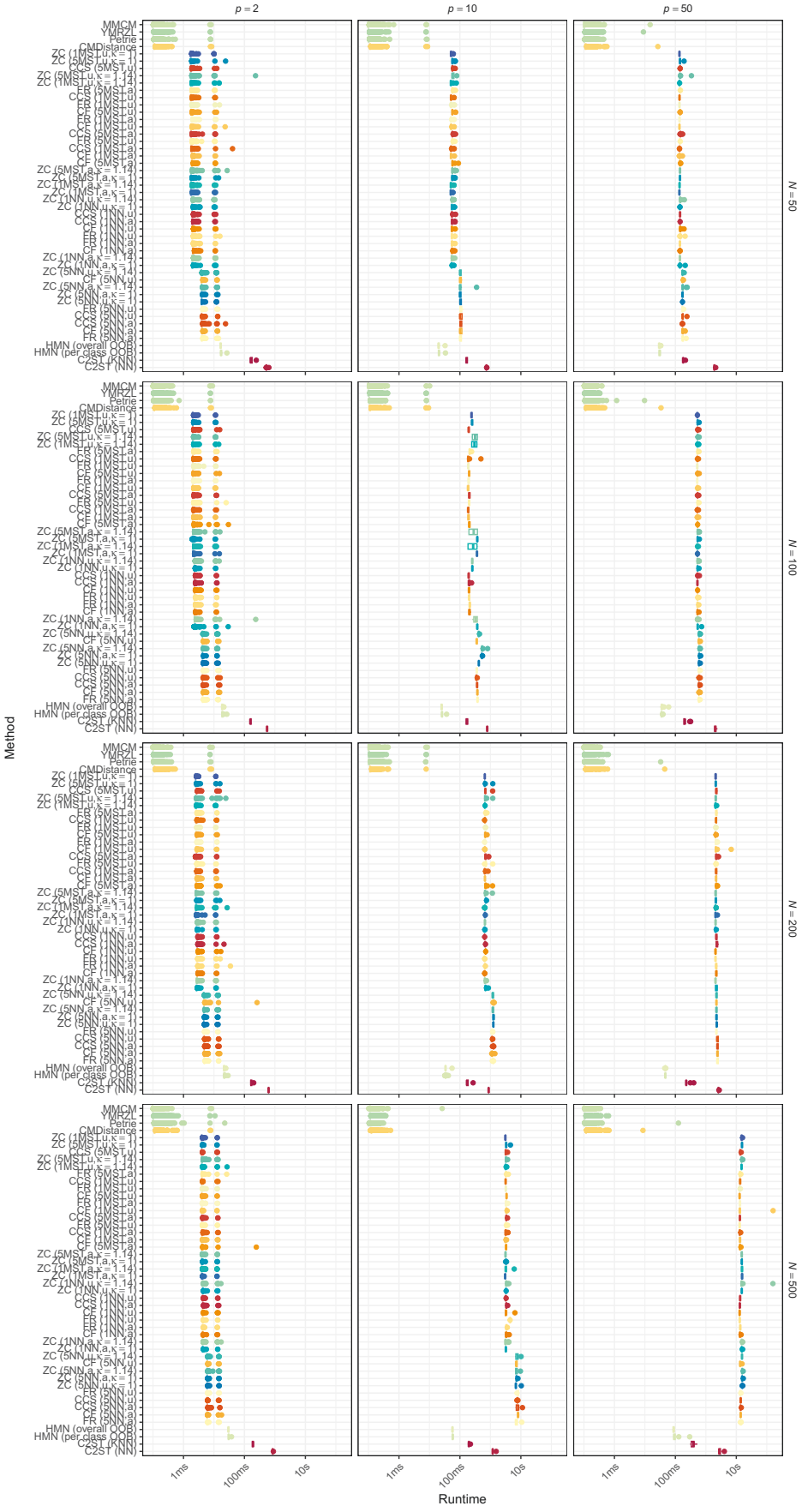


Figure 99: Runtime comparison for the scenario with two binary datasets with balanced class probabilities and equal sample sizes. Ten repetitions were performed for each method.

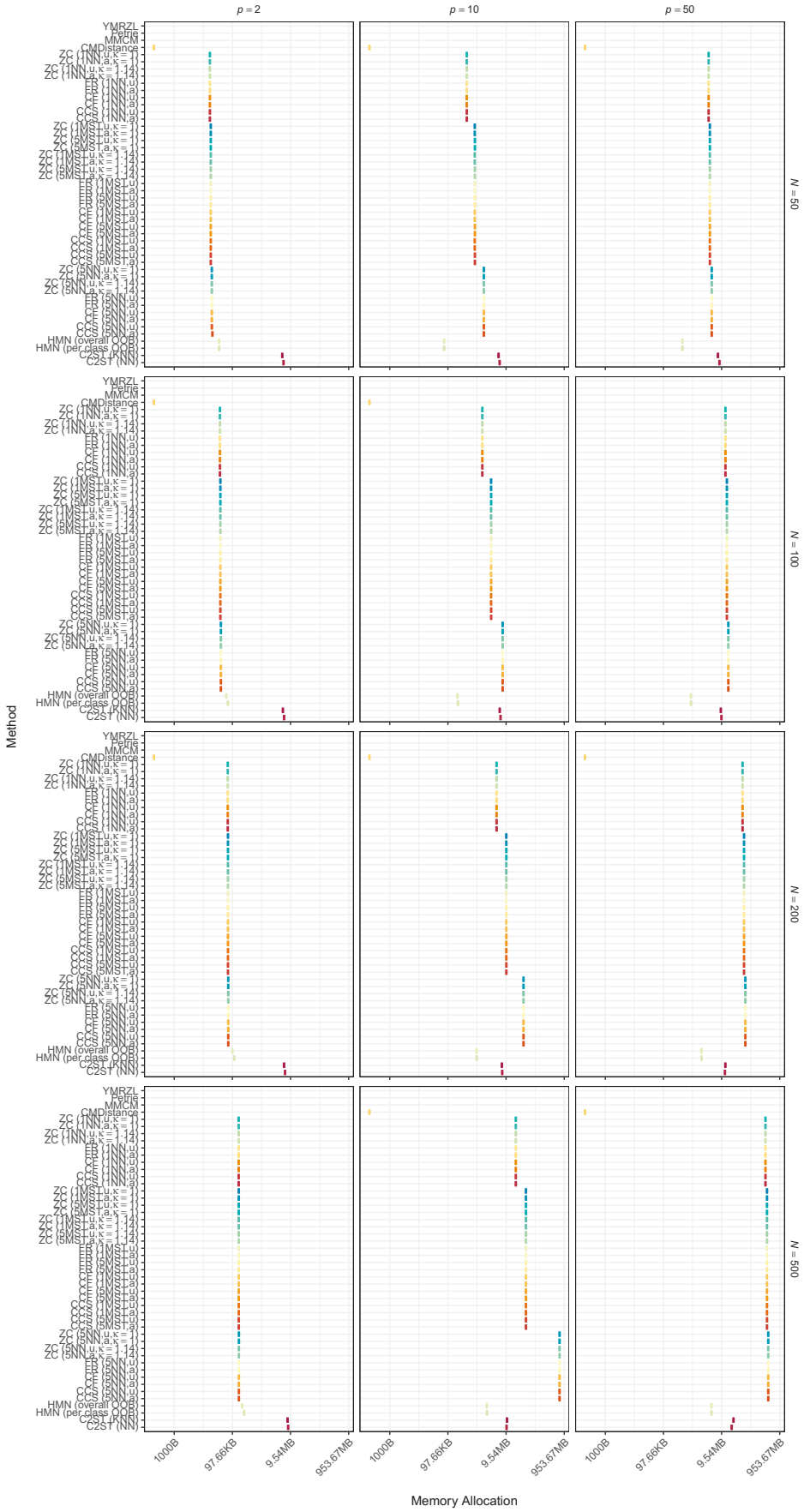


Figure 100: Memory consumption comparison for the scenario with two binary datasets with balanced class probabilities and equal sample sizes. One repetition was performed for each method.