Machine Learning for Acquiring Knowledge in Astro-Particle Physics

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Mirko Bunse

Dortmund

2022

Tag der mündlichen Prüfung:	17. November 2022
Dekan:	Prof. Dr.-Ing. Gernot A. Fink
Gutachter*innen:	Prof. Dr. Katharina Morik
	Dr. Fabrizio Sebastiani

Abstract

This thesis explores the fundamental aspects of machine learning, which are involved with acquiring knowledge in the research field of astro-particle physics. This research field substantially relies on machine learning methods, which reconstruct the properties of astro-particles from the raw data that specialized telescopes record. These methods are typically trained from resource-intensive simulations, which reflect the existing knowledge about the particles—knowledge that physicists strive to expand. We study three fundamental machine learning tasks, which emerge from this goal.

First, we address ordinal quantification, the task of estimating the prevalences of ordered classes in sets of unlabeled data. This task emerges from the need for testing the agreement of astro-physical theories with the class prevalences that a telescope observes. To this end, we unify existing methods on quantification, propose an alternative optimization process, and develop regularization techniques to address ordinality in quantification problems, both in and outside of astro-particle physics. These advancements provide more accurate reconstructions of the energy spectra of cosmic gamma ray sources and, hence, support physicists in drawing conclusions from their telescope data.

Second, we address learning under class-conditional label noise. More particularly, we focus on a novel setting, in which one of the class-wise noise rates is known and one is not. This setting emerges from a data acquisition protocol, through which astro-particle telescopes simultaneously observe a region of interest and several background regions. We enable learning under this type of label noise with algorithms for consistent, noise-aware decision thresholding. These algorithms yield binary classifiers, which outperform the existing state-of-the-art in gamma hadron classification with the FACT telescope. Moreover, unlike the state-of-the-art, our classifiers are entirely trained from the real telescope data and thus do not require any resource-intensive simulation.

Third, we address active class selection, the task of actively finding those proportions of classes which optimize the classification performance. In astro-particle physics, this task emerges from the simulation, which produces training data in any desired class proportions. We clarify the implications of this setting from two theoretical perspectives, one of which provides us with bounds of the resulting classification performance. We employ these bounds in a certificate of model robustness, which declares a set of class proportions for which the model is accurate with a high probability. We also employ these bounds in an active strategy for class-conditional data acquisition. Our strategy uniquely considers existing uncertainties about those class proportions that have to be handled during the deployment of the classifier, while being theoretically well-justified.

Acknowledgments

A project at the scale of a Ph.D. thesis cannot be realized without a sizeable amount of support. Therefore, I am deeply indebted to everyone who contributed to making this thesis a reality.

First and foremost, let me express my deep gratitude to my supervisor Katharina Morik, who established a thriving interdisciplinary environment between Computer Science and Astro-Particle Physics at TU Dortmund University. Without this environment, a thesis like this one would have been outright unimaginable.

Definitely, this thesis would also not have been possible without the generous hospitality of the members of *Istituto di Scienza e Tecnologie dell'Informazione* in Pisa. Especially, I am deeply grateful to Fabrizio Sebastiani and Alejandro Moreo, who invited me for the lovely research visit from which Chapter 4 emerged.

Who could write a Ph.D. thesis without having marvelous co-authors and colleagues? I certainly could not. Thanks to all of you for the enlightening discussions we had, for your instructive feedback on my ideas, and for all the fun we had during the last four years. Without Lukas Pfahler, Chapter 5 would not have come into existence. Maximilian Linhoff and Sebastian Buschjäger gave essential feedback on Chapters 1–3. Special thanks also to my closest collaborators Tim Ruhe, Jens Buß, Wolfgang Rhode, Amal Saadallah, Martin Senz, Dorina Weichert, Alexander Kister, and everyone from the working group of Katharina Morik.

Last but definitely not least, I am eternally grateful for the remarkable amount of support I have received from my friends and family. At times, your faith in this work was deeper than my own and, hence, an indispensable source of endurance. Jana Fingerhut, I cannot count how often you have backed me in withstanding the struggles that research has to offer. Thank you, my dear family, for placing the tracks that eventually led me to the point where I am at now. Special thanks to Mark Bunse for proof-reading several parts of this thesis and to Jonas Bonke for making invaluable comments on epistemology.

You all are amazing. Thank you.

Contents

1. Introduction

Astro-particle physics, a research field at the intersection between astronomy and particle physics, studies the characteristics of cosmic particles to advance the understanding of fundamental physics in the extreme cosmic environments where these particles are being accelerated [15]. This goal is pursued by observing cosmic particles with specialized telescopes that substantially rely on machine learning [16]. Without machines that learn from simulations, one could not reconstruct the characteristics of the cosmic particles from the raw and unlabeled data, which the telescopes take.

1.1. Challenges

In order to support the advancement of astro-particle research, machine learning methods must build on the existing knowledge of astro-particle physicists. A considerable amount of this knowledge is encoded in the simulations with which machine learning methods are trained. In fact, astro-particle simulations reflect a deep understanding of a large set of physical processes that contribute to the telescope recordings, including particle interactions, light propagation, and camera electronics. Moreover, the analysis outcomes have to be appropriately constrained in order to comply with the background knowledge of the physicist. Beyond these aspects of existing knowledge lie two prospects: either the discovery of a previously unknown physical effect or an undesired artifact that disturbs the analysis. Permitting discoveries while preventing disturbances is a truly interdisciplinary endeavor; it requires robust and reliable machine learning methods that are capable of supporting the advancement of astro-physical theories.

In this thesis, we investigate these machine learning requirements from a *computer science* perspective. Namely, we address the robustness of machine learning methods in the scope of three fundamental learning tasks:

Ordinal Quantification (OQ) is the task of estimating the marginal distribution of ordinal class labels from a set of unlabeled data. In astro-particle physics, this task emerges as a problem termed *deconvolution* [1] or *unfolding* [17], where the goal is to reconstruct energy spectra and sky maps from the telescope data. The challenge behind OQ is a shift in label probabilities between model training and deployment; OQ methods must be robust with respect to this shift in order to produce valid estimates of the marginal distributions.

Class-Conditional Label Noise Learning (CCN) is the task of learning with randomly flipped labels, where the flip probabilities exclusively depend on the true class

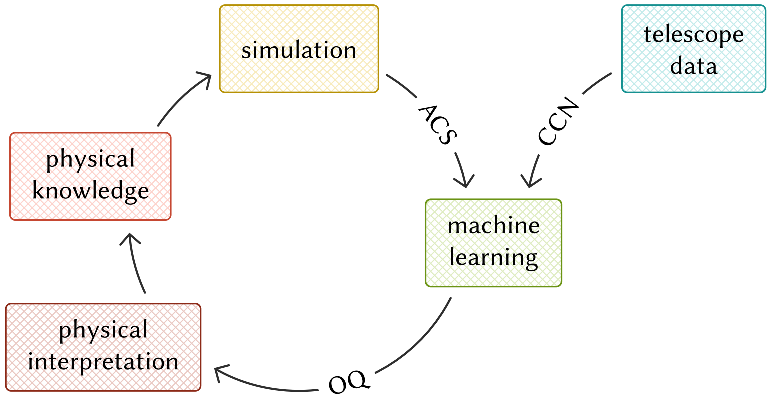

Figure 1.1.: The iterative progression of knowledge in astro-particle physics. Existing *physical knowledge* is encoded in the *simulation*, which provides *machine learning* methods with training data. The trained models are used to reconstruct the *telescope data*, in order to yield a *physical interpretation* and thereby advance the *physical knowledge*. In this illustration, the *machine learning* node subsumes all learning-related analysis steps, e.g. sampling, preprocessing, feature extraction, prediction, and validation. This figure expands the view of probabilistic rationalism [24, Fig. 7] by pointing out that the telescope data is an external source of the knowledge that is to be acquired.

of each instance. The central difficulty within this task lies in finding the decision threshold of a classifier when the noise rates are unknown [18]. The challenge of CCN in astro-particle physics is to leverage the fact that one noise rate is known and one is not [7].

Active Class Selection (ACS) is the task of finding those proportions of classes, with which the most cost-efficient training set can be acquired [19]. In astro-particle physics, this problem emerges from the fact that all training data is generated by simulations which take the class distribution as an input. Choosing a suitable class distribution for the simulation is challenging because the real distribution of astro-particle classes is extremely imbalanced and not known precisely.

These three tasks support the iterative expansion of scientific knowledge in several steps, which are displayed in Fig. 1.1. In this view, OQ facilitates the physical interpretation of the data, ACS provides cost-efficient simulations, and CCN reduces the simulation demand by enabling learning from real telescope data. Together, these tasks support the advancement of valid, scientific knowledge *under resource constraints*. Each of these task is *fundamental* because it can, by itself, also appear in other areas of application.

The non-captioned arrows in Fig. 1.1 lie beyond the scope of this thesis. Namely, we do not consider how to advance a physical theory with a physical interpretation and we do not consider how to improve the simulator with physical knowledge. These limitations reflect the fundamental computer science focus of this thesis. Regarding these topics, the interested reader can refer to the dissertations of astro-particle physicists with whom we have collaborated [20]–[23].

1.2. Impact

In this thesis, we focus on the machine learning aspects of the interdisciplinary endeavor, which faithful analyses of astro-particle data pose. Through this focus, we contribute, first and foremost, to the fundamental research of *computer science.* Our central contributions are as follows:

Concerning OQ, we unify existing methods on ordinal and non-ordinal quantification within a common framework. To this end, we demonstrate that unfolding methods from physics are in fact OQ methods, which are generally applicable, also to OQ tasks from other areas of application. Moreover, we propose regularization techniques for addressing ordinality in quantification algorithms that are, otherwise, non-ordinal. We further propose a soft-max objective for an effective, unconstrained solution of the quantification problem. This objective can be employed in several, widely-acknowledged ordinal and non-ordinal quantification methods. By revealing the equivalence of unfolding and OQ and by extending non-ordinal quantification algorithms with ordinality, we open the topic for strengthened interdisciplinary efforts on improving the reconstruction of the energy spectra of cosmic gamma ray sources.

In the scope of CCN, we discuss a novel setting of this learning task, in which one noise rate is known and one is not. This setting stems from astro-particle physics but can occur in any area of application where a similar data taking protocol is employed. We present a hypothesis test for this setting, to assess whether CCN learning is feasible. Moreover, we derive a heuristic performance metric, which only requires noisy labels, from this test. We employ this metric in finding optimal decision thresholds for any soft classifier and in learning entire decision tree classifiers from scratch. In large sets of closed telescope data, we demonstrate that CCN learning methods are even able to outperform the existing state-of-the-art in gamma hadron classification. This progress leads to more effective detections of cosmic gamma ray sources and it saves computational resources, which are otherwise required by simulations, through learning from *real* telescope data.

Regarding ACS, we argue that the free choice of class proportions constitutes a domain gap, the implications of which we discuss from two perspectives. Our first perspective, which stems from *information theory*, provides us with an explanation for typical behaviors of ACS methods. Our second perspective, which stems from *learning theory*, further allows for a quantitative assessment of model performance through probably and approximately correct (PAC) learning bounds. Based on these bounds, we propose a certificate for model robustness under prior probability shift, which declares a set of class proportions for which the model is accurate with a high probability. We also use our PAC bounds to propose an active strategy for data acquisition in ACS, which uniquely considers existing uncertainties about the class proportions that a classifier has to handle during its deployment. ACS tasks occur in any application area where a class-conditional data generator is employed for the production of labeled training data. In astro-particle physics, this setting emerges from class-conditional simulations.

1.3. Limitations

An essential characteristic of astro-particle analyses is that both the simulation and the real detector produce their observations $x \in \mathcal{X}$ from the ground-truth labels $y \in \mathcal{Y}$. In other words, we intend to *reconstruct* the particle properties y that have lead to an observation x, rather than forecasting the outcome of some given initial conditions. The characteristic of y leading to x is not a superficiality; indeed, it has profound implications on the feasibility of learning tasks. On the one hand, OQ would loose some of its difficulty and ACS would be ill-defined if y did not determine x. On the other hand, several learning tasks, which appear in simulation-based forecasting [25]–[28], are not applicable to the $\mathcal{Y} \rightarrow \mathcal{X}$ problems faced here.

For instance, an astro-particle simulation ($\mathcal{Y} \rightarrow \mathcal{X}$) is not capable of providing labels for existing unlabeled observations. Therefore, it cannot be employed as an *oracle* ($\mathcal{X} \rightarrow \mathcal{Y}$) in active learning frameworks, as it has been proposed for simulations in milling [25], in tunneling [26], and in molecular dynamics [27], [28]. These other simulations produce labels for unlabeled feature vectors; hence, they can be used as oracles in active learning. For the same reason, however, they cannot be optimized through ACS, which requires a simulation to produce feature vectors from labels. Despite these differences, active learning and ACS pursue the same goal: to reduce the amount of simulation resources that are needed to train a valid prediction model. In this thesis, we address this goal through ACS.

Due to the computer science focus of this thesis, we do not address the non-captioned arrows in Fig. 1.1. Namely, we do not consider how to advance a physical theory with a physical interpretation and we do not consider how to improve the simulator with physical knowledge. Note, however, that these topics are the subject of our on-going joint work with astro-particle physicists.

1.4. Outline

The fundamentals of machine learning and astro-particle physics are given in Chap. 2 and in Chap. 3. We then devote one chapter to each of the fundamental machine learning tasks. In particular, we address OQ in Chap. 4, CCN in Chap. 5, and ACS in Chap. 6. We draw the conclusions of this thesis in Chap. 7, where we also conceive several directions for future work. Appendix A provides a commented bibliography of our publications, which provide the foundation for this thesis.

2. Fundamentals of Machine Learning

In this thesis, we develop machine learning methods that address ordinal quantification, class-conditional label noise, and active class selection. Before we detail these methods in Chapters 4–6, we introduce their common, underlying concepts.

This chapter is structured as follows: based on the notation that we define in Sec. 2.1, we present the concepts of supervised machine learning in Sec. 2.2. These concepts provide the basis for all learning tasks that we address in this thesis. In ordinal quantification and in active class selection, we encounter prior probability shift, a complication that we detail in Sec. 2.3. The foundation of our experimental methodology is given in Sec. 2.4.

2.1. Notation

Throughout this thesis, the uppercase letters X and Y stand for random variables, where X denotes a vector-valued random variable that comprises all features and Y denotes the target variable. In supervised learning, X is the input of a model that is meant to accurately predict Y. The state spaces of these random variables, which contain all possible realizations thereof, are written as calligraphic letters $\mathcal{X}$ and $\mathcal{Y}$. Their realizations are denoted as lowercase letters x and y, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Throughout this thesis, the feature space $\mathcal{X} \subseteq \mathbb{R}^d$ is a multi-dimensional, real-valued space, where d is the number of features. The choice of the label space $\mathcal{Y}$ depends on the machine learning task.

The probability density of the event, in which X takes the value x, is written as $\mathbb{P}(X = x)$. We also use the term "probability density" for $\mathbb{P}(Y = y)$, even if $\mathcal{Y}$ is categorical. Joint densities are written as $\mathbb{P}(X = x \wedge Y = y)$ and conditional densities are written as $\mathbb{P}(X = x \mid Y = y)$. Expected values are denoted as $\mathbb{E}_{x \sim \mathbb{P}}(x)$.

If we encounter multiple realizations of the same random variable, we distinguish them with subscripts, x_i, where i is a positive integer. In contrast, we address the i-th component of a vector x with brackets and a subscript, $[x]_i$. This notation allows us to address the i-th component of some specific realization x_j as $[x_j]_i$.

A labeled data set $\mathcal{D}_{XY} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \,:\, 1 \leq i \leq m\} \sim \mathbb{P}^m$ is a sequence of m realizations of X and Y, as according to their joint density $\mathbb{P}(X = x \wedge Y = y)$. An un-labeled data set $\mathcal{D}_X = \{x_i \in \mathcal{X} \,:\, 1 \leq i \leq m\} \sim \mathbb{P}^m$ is a sequence of m realizations of X without their corresponding labels. Typically, $\mathcal{D}_{XY}$ is used for the training and the validation of supervised learning methods, while $\mathcal{D}_X$ requires a supervised model to predict the corresponding labels y_i during deployment.

2.2. Supervised Machine Learning

A frequent goal in machine learning is to predict some target variable Y from features X. This goal is motivated in countless applications, where Y is difficult, costly, or even impossible to measure, while X can be measured with a reasonable cost. A practitioner, who is able to acquire a labeled data set $\mathcal{D}_{XY}$, can use supervised learning to find a mapping that predicts Y for any future realization of X. With this mapping, the practitioner can omit future measurements of Y in favor of predicting Y from X.

Definition 2.1 (Supervised Learning). *Given a labeled data set $\mathcal{D}_{XY}$, the goal of supervised learning is to find a mapping $\mathcal{X} \rightarrow \mathcal{Y}$, from some family of mappings, such that some quality measure $\mathcal{Q} : (\mathcal{X} \rightarrow \mathcal{Y}) \rightarrow \mathbb{R}$ is maximized.*

We typically call the desired mapping a *prediction model* because it predicts the value of Y from any realization of X. Depending on the choice of the label space $\mathcal{Y}$, this definition comprises binary classification ($\mathcal{Y} = \{0, 1\}$), multi-class classification ($\mathcal{Y} = \{1, \ldots, C\}$, where C is the number of classes), ordinal classification ($\mathcal{Y} = \{1, \ldots, C\}$, where the semantic meaning of the classes follows a total order), regression ($\mathcal{Y} \subseteq \mathbb{R}$), and multi-task learning ($\mathcal{Y} \subseteq \mathbb{R}^t$, where t is the number of tasks).

An algorithm for supervised learning consists of the following components:

Model Family The family of permitted mappings $\mathcal{X} \rightarrow \mathcal{Y}$ has to be specified. With a given model family, which is parameterized by a vector β, supervised learning means to find a parameter vector β^* that maximizes $\mathcal{Q}$ for the data set $\mathcal{D}_{XY}$.

Quality Measure The quality measure $\mathcal{Q}$ has to reflect the criteria that characterize a "good" prediction model in the use case at hand. Typical criteria, in this regard, are low prediction errors and an accordance with prior assumptions about the prediction model. The measure $\mathcal{Q}$ estimates the quality of any candidate vector β through the data set $\mathcal{D}_{XY}$, so that a parameter vector β^*, which is optimal for $\mathcal{D}_{XY}$, can be identified.

Optimization Depending on the model family and the quality measure, a suitable optimization algorithm has to be chosen. This algorithm searches, in the space of parameter vectors, for the optimal parameter vector β^*.

In the following sub-sections, we discuss empirical risk minimization, thresholded decision functions, and decision trees. This selection of supervised learning algorithms has a particular relevance for the findings of this thesis.

2.2.1. Empirical Risk Minimization

Many supervised learning algorithms can be described in the framework of *empirical risk minimization* (ERM). In this framework, we are considering a hypothesis $h : \mathcal{X} \rightarrow \mathcal{Z}$ from some hypothesis class $\mathcal{H}$, where the output space $\mathcal{Z}$ depends on the learning task. In binary classification, for instance, $\mathcal{Z} = \mathbb{R}$ comprises real-valued *scores* for the positive class and a binary prediction $\widehat{y} \in \{0, 1\}$ is obtained by thresholding $h(x)$ at some scalar $\lambda \in \mathbb{R}$, i.e., $\widehat{y} = 1$ iff $h(x) > \lambda$. In multi-class classification, $\mathcal{Z} = \mathbb{R}^C$ comprises vectors of class-wise scores and a prediction $\widehat{y} \in \{1, \ldots, C\}$ is obtained by selecting the maximum score, $\widehat{y} = \arg\max_i [h(x)]_i$. In regression, $\mathcal{Z} = \mathcal{Y} = \mathbb{R}$ is identical to the label space $\mathcal{Y}$, so that predictions are directly given by $\widehat{y} = h(x)$. In Chap. 6, we use ERM to establish a probably approximately correct (PAC) learning theory on active class selection.

The ultimate goal of ERM is to minimize the *expected* risk

$$R(h; \ell) = \mathbb{E}_{(x,y)\sim\mathbb{P}}\big(\ell(h(x), y)\big) = \int_{\mathcal{X}\times\mathcal{Y}} \mathbb{P}\big(X = x \wedge Y = y\big) \cdot \ell\big(h(x), y\big) \,\mathrm{d}x\,\mathrm{d}y \tag{2.1}$$

of h under the loss function $\ell : (\mathcal{Z} \times \mathcal{Y}) \to \mathbb{R}$, which measures the error of each individual prediction. However, the true expected risk is not accessible when learning from a finite set of data. Therefore, the expected risk is approximated through the *empirical* risk

$$R_{\mathcal{D}_{XY}}(h; \ell) \;=\; \frac{1}{m} \sum_{i=1}^{m} \ell\big(h(x_i), y_i\big), \tag{2.2}$$

which is computed from the training set $\mathcal{D}_{XY}$. ERM learns an optimal prediction model h^* through minimizing this accessible, empirical risk instead of the inaccessible, expected risk from Eq. 2.1. In particular, ERM learns the model

$$h^* \;=\; \operatorname*{arg\,min}_{h\in\mathcal{H}} R_{\mathcal{D}_{XY}}(h; \ell) \tag{2.3}$$

from a fixed data set $\mathcal{D}_{XY}$. The larger the training set becomes, the more accurate will the estimation of R through $R_{\mathcal{D}_{XY}}$ be. Therefore, the minimizer of $R_{\mathcal{D}_{XY}}$ approaches the desired minimizer of R as the size m of $\mathcal{D}_{XY}$ increases.

To this end, the following bound measures the probability that the estimation error of the empirical risk, which is induced by the finite size m of $\mathcal{D}_{XY}$, is bounded above by some $\varepsilon > 0$. This bound is a standard bound from learning theory and it addresses learning tasks with identically and independently distributed (IID) data. This IID property means that the training set $\mathcal{D}_{XY}$ is drawn from the same probability density function that is also encountered during the deployment of the prediction model. In Sec. 2.3, we discuss a setting where the IID assumption does not hold. For now, however, the IID assumption establishes that an increase in m leads to a lower error ε, if the probability δ of the following bound is kept constant.

Proposition 2.1 (IID Bound [29]). For any $\varepsilon > 0$, any loss function $\ell : (\mathcal{Z} \times \mathcal{Y}) \to \mathbb{R}$, and any fixed $h \in \mathcal{H}$, it holds with probability at least $1 - \delta$, where $\delta = 2e^{-2m\varepsilon^2}$, that

$$\big|R_{\mathcal{D}_{XY}}(h; \ell) - R(h; \ell)\big| \;\leq\; \varepsilon.$$

Proof. We repeat the proof by Shalev-Shwartz & Ben-David [29, Sec. 4.2] for later reference. By letting $\theta_i = \ell(h(x_i), y_i)$ and $\mu = R(h; \ell)$, we apply Hoeffding's inequality, which states that, for $0 \leq \theta_i \leq 1$,

$$\mathbb{P}_{\mathcal{D} \sim \mathbb{P}^m} \left(\left| \frac{1}{m} \sum_{i=1}^{m} \theta_i - \mu \right| > \varepsilon \right) \; \leq \; 2e^{-2m\varepsilon^2} \; = \; \delta. \tag{2.4}$$

We see that the converse event, i.e., $\left| \frac{1}{m} \sum_{i_1}^{m} \theta_i - \mu \right| \leq \varepsilon$, holds with probability at least $1 - \delta$. Taking Eq. 2.4 for granted would therefore already yield our claim (□). For later reference, however, we take another step back and prove Eq. 2.4 from Hoeffding's Lemma, which states that, for every $\lambda > 0$ and for any random variable $X \in [a, b]$ with $\mathbb{E}_{x \sim \mathbb{P}}(x) = 0$,

$$\mathbb{E}_{x \sim \mathbb{P}}(e^{\lambda x}) \; \leq \; e^{\frac{\lambda^2 (b-a)^2}{8}}. \tag{2.5}$$

Letting $z_i = \theta_i - \mu$ and $\bar{z} = \frac{1}{m} \sum_{i=1}^{m} z_i$, we use *i)* monotonicity, *ii)* Markov's inequality, *iii)* independence, and *iv)* Eq. 2.5 with $a = 0$, $b = 1$, and $\lambda = 4m\varepsilon$ to see that

$$\mathbb{P}(\bar{z} \geq \varepsilon) \overset{i)}{=} \mathbb{P}(e^{\lambda \bar{z}} \geq e^{\lambda \varepsilon}) \overset{ii)}{\leq} e^{-\lambda \varepsilon} \, \mathbb{E}_{x \sim \mathbb{P}}(e^{\lambda \bar{z}}) \overset{iii)}{=} e^{-\lambda \varepsilon} \prod_{i=1}^{m} \mathbb{E}_{x \sim \mathbb{P}}(e^{\lambda z_i / m}) \overset{iv)}{\leq} e^{-2m\varepsilon^2}. \tag{2.6}$$

We apply Eq. 2.6 to $\bar{z}$ and to $-\bar{z}$. We yield Eq. 2.4 via the union bound. □

We substantiate our introduction of the ERM framework with the following example. This example defines two popular loss functions for different learning tasks.

Example 2.1 *(Logistic Loss and Squared Error).* Let $\mathcal{Y} = \{0, 1\}$. Examples of loss functions $\ell : (\mathcal{Z} \times \mathcal{Y}) \rightarrow \mathbb{R}$, which define the risk in ERM, are the logistic loss

$$\ell_{\text{logistic}}\big(h(x), y\big) \; = \; -y \cdot \ln\big(h(x)\big) - (1 - y) \cdot \ln\big(1 - h(x)\big),$$

which is also known as the cross-entropy loss, and the squared error loss

$$\ell_{\text{squared}}\big(h(x), y\big) \; = \; \big(h(x) - y\big)^2.$$

The former of these loss functions is frequently used in classification tasks, while the latter is more frequently used in regression tasks.

Any risk that is defined through a loss function ℓ can be directly optimized through ERM. However, some widely-acknowledged quality measures for binary classification cannot be specified in terms of example-wise losses, i.e., these measures cannot be decomposed in terms of Eqs. 2.1 and 2.2. Several measures of this kind, however, can be optimized through a combination of ERM with a separate optimization of the decision threshold, as we see in the following subsection.

2.2.2. Thresholded Decision Functions for Binary Classification

An widely-acknowledged family of *binary* classifiers learns some real-valued decision function $h : \mathcal{X} \rightarrow \mathbb{R}$, which is then thresholded at some $\lambda \in \mathbb{R}$ to yield a binary prediction

$$\widehat{y} \;=\; \mathbb{1}_{h(x) > \lambda} \;=\; \begin{cases} 1 & \text{if } h(x) > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

We call this family of classifiers *thresholded decision functions.* The practical relevance of this family stems from the fact that several important quality measures for binary classification, including accuracy and the F_β score, can indeed be optimized by classifiers of this kind. More specifically, several quality measures $\mathcal{Q} : (\mathcal{X} \rightarrow \mathcal{Y}) \rightarrow \mathbb{R}$ have the following property: if $h(x)$ is a consistent estimate of the posterior probability $\mathbb{P}(Y = 1 \mid X = x)$, there exists a threshold $\lambda_Q \in \mathbb{R}$ such that Eq. 2.7 approaches the Bayes-optimal classifier with respect to $\mathcal{Q}$ in the large sample limit $m \rightarrow \infty$. In other words: if $h(x)$ is consistent, we can optimize several quality measures $\mathcal{Q}$ merely by selecting a suitable threshold $\lambda_{\mathcal{Q}}$. We employ thresholded decision functions in Chap. 5, where we show that they are optimal even in spite of class-conditional label noise.

The requirement for consistent, binary hypotheses $h : \mathcal{X} \rightarrow \mathbb{R}$ is fulfilled by *proper scoring rules*, for which ERM yields a consistent estimate of $\mathbb{P}(Y = 1 \mid X = x)$ by Def. 2.2 and by Prop. 2.1. Therefore, we can use proper scoring rules to learn thresholded decision functions that are optimal with respect to the mentioned quality measures.

Definition 2.2 (Proper Scoring Rule). *We call a loss function $\ell : (\mathcal{Z} \times \mathcal{Y}) \rightarrow \mathbb{R}$ a* proper scoring rule*, if the* expected *loss $R(h; \ell)$, which is induced by ℓ, is minimized by the hypothesis $h^*(x) = \mathbb{P}(Y = 1 \mid X = x)$, i.e., if*

$$\operatorname*{arg\,min}_{h \in \mathcal{H}} R(h; \ell) \;=\; \mathbb{P}(Y = 1 \mid X).$$

Moreover, if h^* is unique, we call ℓ strictly proper.

Example 2.2 *(Logistic Loss and Squared Error—Revisited).* Let $\mathcal{Y} = \{0, 1\}$. Buja et al. [30] show that two widely adopted loss functions, the logistic loss and the squared error loss from Ex. 2.1 are indeed proper scoring rules. Therefore, these loss functions can be used for learning optimal, thresholded decision functions.

A non-exhaustive list of quality measures, which are optimized through thresholded decision functions, is given in Tab. 2.1. Some of these measures induce a fixed optimal threshold, e.g. accuracy has an optimal threshold of $\frac{1}{2}$. Other measures induce an optimal threshold that depends on the Bayes utility $\mathcal{Q}^* = \sup_{h \in \mathcal{H}} \mathcal{Q}(h)$, which is usually unknown. In these cases, it is possible to optimize the threshold on a hold-out set of training data. To this end, Alg. 2.1 learns a consistent decision function $h : \mathcal{X} \rightarrow \mathbb{R}$ on one part of the data and optimizes the decision threshold λ on another, disjunct part of the data. Even though this algorithm is fairly straightforward, it is indeed consistent in

Table 2.1.: The most important quality measures $\mathcal{Q} : (\mathcal{X} \to \mathcal{Y}) \to \mathbb{R}$ for which the Bayes-optimal classifier takes the form $h^*(x) = 1$ iff $\mathbb{P}(Y = 1 \mid X = x) > \lambda_\mathcal{Q}$, and $h^*(x) = 0$ otherwise, with an optimal threshold $\lambda_\mathcal{Q} \in \mathbb{R}$. The Bayes utility $\mathcal{Q}^* = \sup_{h \in \mathcal{H}} \mathcal{Q}(h)$ is the optimal value of the quality measure, which is attained by h^*. We further write TP, TN, FP, and FN for true positives, true negatives, false positives, and false negatives. We give reference to publications which explicitly state the optimal threshold $\lambda_\mathcal{Q}$; only in case of the G measure, no explicit threshold is given.

quality measure $\mathcal{Q}$	definition	optimal threshold $\lambda_\mathcal{Q}$	
accuracy	$\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$	$\frac{1}{2}$	[31], [32]
AM measure	$\frac{\text{TPR}+\text{TNR}}{2}$	$\mathbb{P}(Y = 1)$	[33]
F_β measure	$(1 + \beta^2) \cdot \left(\frac{\beta^2}{\text{Prec}} + \frac{1}{\text{TPR}}\right)^{-1}$	$\frac{\mathcal{Q}^*}{1+\beta^2}$	[34]
G measure	$\sqrt{\text{TPR} \cdot \text{Prec}}$		[32]
Jaccard similarity	$\frac{\text{TP}}{\text{TP}+\text{FP}+\text{FN}}$	$\frac{\mathcal{Q}^*}{1+\mathcal{Q}^*}$	[31]
precision / Prec	$\frac{\text{TP}}{\text{TP}+\text{FP}}$	$\mathcal{Q}^*$	[31]
recall / TPR	$\frac{\text{TP}}{\text{TP}+\text{FN}}$	0	[31]

terms of $\mathcal{Q}$. Its consistency is independently proven by Koyejo et al. [31, Theorem 10] and by Narasimhan et al. [32, Theorem 1]. Hence, we use Alg. 2.1 to optimize any of the quality measures of from Tab. 2.1.

Tab. 2.1 lists only the most important quality measures that are optimized through thresholded decision functions. Moreover, however, this family of measures also includes any other measure that is a ratio of linear combinations of the numbers of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [31]. Choi et al. [35] present a large collection of such measures.

Algorithm 2.1 Two-step expected utility maximization [31], [32].

```
Input: a training set $\mathcal{D}_{XY} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq m\}$, a consistent learning algorithm $\mathcal{A} : \bigcup_{m=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{H}$, and a quality measure $\mathcal{Q} : (\mathcal{X} \to \mathcal{Y}) \to \mathbb{R}$
Output: a binary classifier $\mathcal{X} \to \mathcal{Y}$
split $\mathcal{D}_{XY}$ into disjunct subsets $\mathcal{D}_h$ and $\mathcal{D}_\lambda$
estimate $h = \mathcal{A}(\mathcal{D}_h)$
optimize $\lambda^* = \arg\max_{0 \leq \lambda \leq 1} \mathcal{Q}(\mathbb{1}_{h(X) > \lambda})$ on $\mathcal{D}_\lambda$
return $\mathbb{1}_{h(X) > \lambda^*}$
```

2.2.3. Random Forests and Decision Trees

The random forest (RF) [36] is an ensemble method that aggregates the predictions from multiple, randomized decision trees [37]. It is a popular tool in astro-particle physics, where it is frequently used for classification [38], [39] and for regression tasks [40], [41]. In this thesis, we employ RFs in most of the experiments, due to their popularity and due to their high performance on the astro-particle data that we analyse. Another central advantage of RFs is that their out-of-bag (OOB) predictions can be used for consistent adjustments of the resulting classifier. We benefit from this property in Chap. 4, where we require OOB predictions for quantification-related adjustments, and in Chap. 5, where we require OOB predictions for the tuning of decision thresholds. After the following introduction of RFs, we also discuss boosted decision trees shortly.

Bagging Multiple methods for randomizing the trees in an RF have been proposed [36]. Here, we employ the classical *bagging* approach [42], which trains each ensemble member h_i, where $1 \leq i \leq B$ in an ensemble of size B, with a bootstrap sample $\mathcal{D}_i$ of the training set $\mathcal{D}_{XY}$. Each of the bootstrap samples is obtained by randomly drawing, with replacement, m training examples from $\mathcal{D}_{XY}$. Those examples that are not drawn, $\mathcal{D}_{XY} \setminus \mathcal{D}_i$, are called the OOB examples of h_i. An additional source of randomness in RFs is to consider, in each split of each tree, only a random subset of all features as splitting candidates; we detail this splitting in a moment.

The scikit-learn[1] implementation of RF classifiers, which we use throughout this thesis, aggregates the predictions of the trained ensemble members h_i by averaging the scores which they return for each class $1 \leq j \leq C$, i.e.,

$$\big[h(x)\big]_j \;=\; \frac{1}{B} \sum_{i=1}^{B} \big[h_i(x)\big]_j \,. \tag{2.8}$$

The OOB prediction of each training example $(x, y) \in \mathcal{D}_{XY}$ is averaged in a similar fashion, but only regarding those ensemble members h_i for which (x, y) is an OOB example and, hence, has not been used during the training of h_i. Thereby, OOB predictions are produced for all examples of the training set without being subject to overfitting. In contrast to Eq. 2.8, the original proposal by Breiman [36] aggregates the predictions in terms of voting, i.e., $[h(x)]_j = \frac{1}{B} \sum_{i=1}^{B} \mathbb{1}_{j = \arg\max_k [h_i(x)]_k}$, which, in our experience, produces less accurate results on the astro-particle data that we analyse.

Decision Tree Induction Each individual tree in an RF is induced through the CART algorithm [37], which creates an increasingly fine partition of the feature space. In particular, each iteration chooses a feature $j \in \mathcal{I}$ from an index set $\mathcal{I} \subseteq \{1, \ldots, d\}$ and a threshold $t \in \mathbb{R}$. In RFs, we choose the index set $\mathcal{I}$ as a random subset of size $\lfloor \sqrt{d} \rfloor$,

[1] `https://scikit-learn.org/`

which is drawn for each split individually. The combination of a feature j and a threshold t splits the data into the segments

$$\mathcal{D}_i^{(L)} = \big\{(x,y) \in \mathcal{D}_i : [x]_j \leq t\big\} \quad \text{and} \quad \mathcal{D}_i^{(R)} = \big\{(x,y) \in \mathcal{D}_i : [x]_j > t\big\}. \tag{2.9}$$

Each iteration of CART chooses the split (j, t) that minimizes the impurity of the resulting segments. In classification, we measure the impurity of a segment in terms of its entropy

$$\text{Entropy}(\mathcal{D}_i) \;=\; -\sum_{k=1}^{C} p_k \cdot \ln(p_k), \qquad \text{where} \quad p_k \;=\; \frac{1}{|\mathcal{D}_i|} \sum_{l=1}^{|\mathcal{D}_i|} \mathbb{1}_{y_l = k} \tag{2.10}$$

and we choose each split (j, t) such that it minimizes the weighted average

$$\frac{|\mathcal{D}_i^{(L)}|}{|\mathcal{D}_i|} \cdot \text{Entropy}\Big(\mathcal{D}_i^{(L)}\Big) \;+\; \frac{|\mathcal{D}_i^{(R)}|}{|\mathcal{D}_i|} \cdot \text{Entropy}\Big(\mathcal{D}_i^{(R)}\Big) \tag{2.11}$$

of the entropies that the resulting segments exhibit. The splitting according to Eq. 2.11 begins at the root node of the tree, where the entire bootstrap sample $\mathcal{D}_i$ is considered. It is then recursively continued on both sides $\mathcal{D}_i^{(L)}$ and $\mathcal{D}_i^{(R)}$ of every split, until a stopping criterion is reached. The result of this procedure is a randomized decision tree, which aims at a minimum entropy of its segments. Eq. 2.8 combines multiple trees in an RF.

Alternative: Boosting An alternative to combining multiple randomized trees via bagging is to train an ensemble of decision tree classifiers via *boosting*. In contrast to bagging, this approach does not train all trees in parallel, but sequentially: in each iteration, the training set is re-weighted in a way that focuses the next tree on patterns that the current ensemble does not yet regard. The popular AdaBoost algorithm [43] has been applied to the data of the HESS [44] and VERITAS [45] telescopes.

For the MAGIC telescope, a stratified variant of this algorithm, Ada^{2}Boost, has been considered [46], [47]. Its stratification weights the classes equally, which is desirable when the area under the receiver operating characteristic (AUROC) is the quality metric [48]. This metric, in turn, is appropriate when the class proportions, which the classifier has to handle during deployment, are *entirely unknown* [46]. In astro-particle physics, however, these class proportions can indeed be estimated [16], [22]. Despite existing uncertainties about this estimate, it is therefore not ideal to optimize AUROC; for any fixed threshold, another classifier might perform better. Moreover, an equal weighting of the classes is also achieved by simulating the classes in equal proportions. Due to these reasons, we focus our analyses on RFs, the standard learning method for the FACT telescope [49].

2.3. Domains and Prior Probability Shift

The supervised machine learning methods from Secs. 2.2.1–2.2.3 primarily address learning from identically and independently distributed (IID) data, which means that the training set $\mathcal{D}_{XY}$ is drawn from the same probability density that is also encountered during the deployment of the learned model. Due to the assumption of IID data, we can learn highly performant prediction models by maximizing a quality measure $\mathcal{Q}$ on $\mathcal{D}_{XY}$; if $\mathcal{D}_{XY}$ is sufficiently large, the IID property ensures that our maximization of $\mathcal{Q}$ is accurate. The IID assumption is, however, often inappropriate, particularly in the three learning tasks that we cover in this thesis. Fortunately, the research area of domain adaptation [50], [51] proposes weaker, more appropriate assumptions, which explicitly allow the training data and the deployment data to stem from different probability densities.

In this context, a *domain* is a joint probability density function $\mathbb{P}(X = x \wedge Y = y)$, which corresponds to some particular data-generating process; a different process would correspond to a different domain. To this end, let $\mathcal{S}$ be the *source* domain where a machine learning model is trained and let $\mathcal{T}$ be the *target* domain where the trained model is required to be valid. Domain adaptation is concerned with the impact of deviations $\mathcal{S} \neq \mathcal{T}$ on the performance in $\mathcal{T}$ and with improving this performance.

In quantification and in active class selection, which we cover in Chap. 4 and in Chap. 6, we assume that $\mathcal{S}$ and $\mathcal{T}$ differ *at most* in terms of their class proportions. In quantification, this assumption arises naturally because the class proportions are the object that is meant to be estimated. In active class selection, we intend to study the impact of a class-conditional data generator, which induces some class proportions in $\mathcal{S}$ while $\mathcal{T}$ is fixed. In both cases, we let all data, no matter if used for training or for testing, be generated by the same, random mechanism $\mathcal{Y} \to \mathcal{X}$. We define this assumption as follows:

Definition 2.3 (Identical Mechanism Assumption [8] a.k.a. Prior Probability Shift [52] or Target Shift [53]). *Assume that all data in the domains $\mathcal{S}$ and $\mathcal{T}$ is generated independently by the same class-conditional mechanism, i.e.,*

$$\mathbb{P}_{\mathcal{S}}(X = x \mid Y = y) \; = \; \mathbb{P}_{\mathcal{T}}(X = x \mid Y = y) \qquad \forall\, x \in \mathcal{X},\; \forall\, y \in \mathcal{Y}.$$

Due to this equality, any deviation $\mathcal{S} \neq \mathcal{T}$ must stem from diverging class proportions, where $\exists\, y \in \mathcal{Y} : \mathbb{P}_{\mathcal{S}}(Y = y) \neq \mathbb{P}_{\mathcal{T}}(Y = y)$.

In astro-particle physics, where supervised models are applied to real telescope data but are typically trained with simulated data, this assumption is a slight simplification. It is a simplification because the real data and the simulated data actually come from different mechanisms, i.e., from the real atmospheric particle interactions and from the simulation. It is, however, only *slight* because the simulation is a highly sophisticated and accurate model of these interactions. Therefore, we can assume that any deviation $\mathcal{S} \neq \mathcal{T}$ is at least *dominated* by diverging class proportions. We detail the real particle interactions in Sec. 3.2 and the simulation in Sec. 3.6.

2.4. Model Validation

For any trained machine learning model, the question is: how well does it perform? Finding an answer to this question requires some specific definition of the term “performance”, which has to match the requirements of the particular use case at hand. To this end, we have already introduced a set of quality measures in Tab. 2.1. Beyond this application-specific aspect of model validation, however, exist general *principles* of model validation, which are independent of the use case and the quality measure.

In the following sub-sections, we give an introduction to these principles, introduce critical difference diagrams as a way to summarize large sets of experiments, and define several divergence measures for probability density functions. These tools provide the foundation for the experiments that we present in Chapters 4–6.

2.4.1. Principles of Model Validation

A meaningful assessment of model performance has to address the *generalization* performance of the model in question. This performance is often in contrast to the performance exhibited on the data that are used for training; the difference between generalization performance and training performance stems from the fact that complex models have the capacity to memorize the training set instead of learning meaningful, underlying patterns [54]. This behavior is known as *overfitting* and a meaningful assessment of model performance has to be able to detect this behavior.

The most straightforward protocol for model validation is the *training test split.* In this protocol, we randomly split the labeled data $\mathcal{D}_{XY}$ into two disjunct subsets $\mathcal{D}_{\text{trn}} \subset \mathcal{D}_{XY}$ and $\mathcal{D}_{\text{tst}} \subset \mathcal{D}_{XY}$, where $\mathcal{D}_{\text{trn}} \cup \mathcal{D}_{\text{tst}} = \mathcal{D}_{XY}$ and $\mathcal{D}_{\text{trn}} \cap \mathcal{D}_{\text{tst}} = \emptyset$. We then use $\mathcal{D}_{\text{trn}}$ for training the model and $\mathcal{D}_{\text{tst}}$ for estimating its performance in terms of some quality measure. The splitting ensures that the performance assessment does not over-estimate the model performance in spite of overfitting. Unfortunately, however, omitting $\mathcal{D}_{\text{tst}}$ during training can decrease the performance of the model, as compared to training with all available data. Moreover, a single split produces only one performance value without exploring the distribution of this value, which stems from the random splitting.

These issues are addressed in *cross validation.* This alternative validation protocol randomly splits the data into K disjunct subsets $\mathcal{D}_i \subset \mathcal{D}_{XY}$ of an equal size, with $1 \leq i \leq K$, $\mathcal{D}_{XY} = \bigcup_{i=1}^{K} \mathcal{D}_i$, and $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset \; \forall \, i \neq j$. Each of its K iterations employs D_i as the test set and all remaining data, $\bigcup_{j \neq i} \mathcal{D}_j$, for training. Finally, the performance estimates of all iterations are averaged to obtain a robust, overall estimate. Through other statistics of the performance values, like their standard deviation, we can further assess their distribution. By choosing an appropriate value for K, e.g., $K = 10$, we can use most of the data for training, so that the model performance does not suffer from a reduced training set size. However, cross validation is K times more expensive, computationally, than a single training test split. In order to yield reliable results, we pay this price in all of our experiments.

Most machine learning methods have hyper-parameters that have to be selected. This purpose requires another performance assessment and, hence, another disjunct set of labeled data. This so-called *validation set* can stem from a single split or from cross-

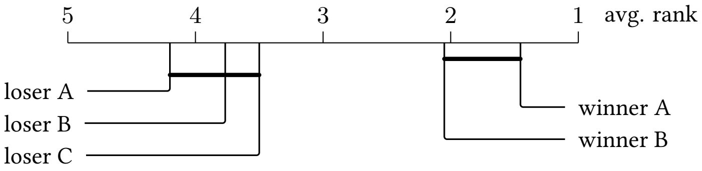

Figure 2.1.: An example of a CD diagram. Here, "winner A" and "winner B" are classification methods that are statistically significantly better than the other methods, in terms of a Holm-corrected Wilcoxon signed-rank test that considers some quality measure across multiple data sets. We can tell from the horizontal positions that "winner A" exhibits a lower average rank, and hence performs better on more data sets, than "winner B". However, since the two methods are not significantly different from each other, we cannot conclude that "winner A" is indeed better than "winner B".

validation splits, of which the performance estimates are averaged. In all experiments, we select hyper-parameters according to their validation performance.

2.4.2. Critical Difference Diagrams

Critical difference (CD) diagrams [55], [56] describe an evaluation process that is specifically designed for statistically valid comparisons of multiple machine learning methods across multiple data sets. In the resulting diagram, the horizontal axis plots the average rank (lower is better) of each method, in terms of some quality measure, across all data sets. From these ranks, a beholder can tell which method frequently wins against the others. Most importantly, however, a CD diagram connects, with horizontal bars, those methods that a Holm-corrected Wilcoxon signed-rank hypothesis test cannot significantly distinguish. Consequently, all *missing* connections indicate the presence of *statistically significant differences* between the competing methods. An example diagram, which illustrates these features, is shown in Fig. 2.1. We employ CD diagrams in Chapters 5–6, to evaluate the performance of multiple machine learning methods, relative to each other, across multiple data sets.

The expressive power of CD diagrams stems not only from their concise visualization, but also from the hypothesis tests that they involve. The process of generating a CD diagram is particularly designed with machine learning experiments in mind [55]. This process consists of the following steps:

Experimental Results The first step in generating a CD diagram is to produce a set of experimental results, i.e., to compute some performance metric for each combination of a method and a data set. Since CD diagrams do not account for the variances of these results, it is important to ensure robust performance estimates, preferably cross-validated, average performance values.

Friedman Test The second step is to conduct a Friedman test, which tells us whether the results exhibit any significant differences at all. If this test fails, we have not sufficient data to tell any of the methods apart and we must abort the generation of the CD diagram. If, however, the test successfully rejects this possibility, we can place each method at the position of its average rank and proceed with the evaluation.

Post-hoc Analysis In this third step, a Wilcoxon signed-rank test tells us whether each pair of methods exhibits a statistically significant difference across all data set-wise performance values; hence, it tells us whether one method is significantly better than the other. Since we are now testing many hypotheses at the same time, we must adjust each Wilcoxon test with Holm's method. This adjustment effectively reduces the p-value of each individual test so that the outcomes of all tests are valid simultaneously, within an overall p-value budget that is defined by the user. In the CD diagram, we then connect all methods that the Holm-corrected Wilcoxon test cannot distinguish.

The original proposal by Demšar was to use a Nemenyi test in the post-hoc analysis [55]. However, the argument by Benavoli et al. [56] establishes the Wilcoxon signed-rank test as a more appropriate choice.

We conclude that CD diagrams can concisely present the outcomes of large sets of experiments, in a statistically valid form. During the course of this thesis, we have implemented the above process in reusable open source software.[2]

2.4.3. Divergence Measures for Probability Densities

In the following, we present divergence measures which assess the dissimilarity between probability density functions P and Q, which are defined over the same support. We use these measures to study the error of quantification outcomes in Chap. 4 and to study the theoretical implications of active class selection in Chap. 6.

Definition 2.4 (Kullback-Leibler Divergence [57]). *Let $P : \mathcal{Z} \to \mathbb{R}$ and $Q : \mathcal{Z} \to \mathbb{R}$ be two probability density functions over the same support set $\mathcal{Z}$ with $P(z) > 0$ and $Q(z) > 0$ for all $z \in \mathcal{Z}$. The differential Kullback-Leibler (KL) divergence between P and Q is*

$$\mathrm{KL}_Z\,(P \,||\, Q) \;=\; \mathbb{E}_{z \sim P}\left(\ln \frac{P(z)}{Q(z)}\right) \;=\; \int_{\mathcal{Z}} P(z) \cdot \ln \frac{P(z)}{Q(z)} \,\mathrm{d}z.$$

For two joint densities $P : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and $Q : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we further define the conditional KL divergence, assuming that the corresponding conditional densities $P(y \mid x)$ and $Q(y \mid x)$ exist, as

$$\mathrm{KL}_{Y|X}\,(P \,||\, Q) \;=\; \mathbb{E}_{(x,y) \sim P}\left(\ln \frac{P(y \mid x)}{Q(y \mid x)}\right) \;=\; \int_{\mathcal{X} \times \mathcal{Y}} P(x, y) \cdot \ln \frac{P(y \mid x)}{Q(y \mid x)} \,\mathrm{d}x\,\mathrm{d}y.$$

[2] `https://github.com/mirkobunse/CriticalDifferenceDiagrams.jl`

A part of our theoretical analysis of active class selection in Chap. 6 is based on the following properties of KL divergences:

Proposition 2.2. Let P and Q be fixed. By omitting their explicit mentioning, we abbreviate $\mathrm{KL}_Z\left(P \;||\; Q\right)$ with KL_Z. Moreover, let $\mathrm{KL}_{X,Y}$ be the KL divergence between the joint densities of X and Y. KL divergences have the following properties [57]:

- **i)** $\mathrm{KL}_Z \geq 0$, where $\mathrm{KL}_Z = 0$ iff $P = Q$. The same property holds for $\mathrm{KL}_{Y|X}$.
- **ii)** $\mathrm{KL}_{X,Y} \;=\; \mathrm{KL}_{Y|X} + \mathrm{KL}_X \;=\; \mathrm{KL}_{X|Y} + \mathrm{KL}_Y$. This property is known as the chain rule of KL divergences.

Proof. These properties follow from Def. 2.4, with detail provided in [57, Theorem 2.6.3 and Theorem 2.5.3]. □

The following divergence measures are required in Chap. 4, where we compare quantification outcomes to ground-truth class prevalences. Since classes are categorical, we can perfectly represent all probability density functions, in this case, as vectors $p \in \mathbb{R}^C$ and $q \in \mathbb{R}^C$, where C is the number of classes. For conciseness, we therefore define the following divergence measures already in terms of vectors p and q, instead of defining them in terms of continuous probability density functions P and Q. To this end, we first define the space $\mathcal{P}^C$ of admissible vectors. This definition allows us to write $p \in \mathcal{P}^C$ and $q \in \mathcal{P}^C$ for requiring valid vector representations of probability density functions in Definitions. 2.6–2.8.

Definition 2.5 (Vector Representation of Probability Density Functions). *Any vector in the unit simplex*

$$\mathcal{P}^C \;=\; \left\{ p \in \mathbb{R}^C \;:\; [p]_i \geq 0 \;\forall\; 1 \leq i \leq C \;\;\wedge\;\; 1 = \sum_{i=1}^{C} [p]_i \right\}.$$

is a valid representation of a probability density function over the set $\mathcal{Y} = \{1, \ldots, C\}$.

These vectors can directly be interpreted as probabilities, i.e., $[p]_i = P(Y = i)$ and $[q]_i = Q(Y = i)$. We use this vector representation to compute the following divergence measures.

Definition 2.6 (Hellinger Distance). *Let $p \in \mathcal{P}^C$ and $q \in \mathcal{P}^C$ be the vector representation of two probability density functions over $\mathcal{Y} = \{1, \ldots, C\}$. The Hellinger distance between p and q is*

$$\mathrm{HD}(p, q) \;=\; \sqrt{\sum_{i=1}^{C} \left(\sqrt{[p]_i} - \sqrt{[q]_i} \right)^2}.$$

Definition 2.7 (Earth Mover's Distance [58], a.k.a. Match Distance [59] or Wasserstein Metric). *Let $p \in \mathcal{P}^C$ and $q \in \mathcal{P}^C$ be the vector representation of two probability density functions over $\mathcal{Y} = \{1, \dots, C\}$. The earth mover's distance (EMD) between p and q is*

$$\mathrm{EMD}(p, q) \;=\; \sum_{i=1}^{C-1} d_i \cdot \left| \sum_{j=1}^{i} [p]_i - [q]_i \right|,$$

where $d_i \in \mathbb{R}$ is the ground distance *between the consecutive vector components i and $i + 1$. Throughout this thesis, we set $d_i = 1 \;\forall\, 1 \leq i \leq C - 1$.*

At this point, we note that the EMD [58], the match distance [59] and the Wasserstein metric originally address different levels of generality. These distances coincide only if we consider one-dimensional histograms, as we do in Def. 2.7.

In order to constrain the EMD to interpretable values between zero and one, Sakai [60] proposes a normalized variant of this distance. This normalized distance, evaluated between a quantification outcome p and a ground-truth vector q, is the main performance measure in our experiments from Chap. 4.

Definition 2.8 (Normalized Match Distance [60]). *Let $p \in \mathcal{P}^C$ and $q \in \mathcal{P}^C$ be the vector representation of two probability density functions over $\mathcal{Y} = \{1, \dots, C\}$. The normalized match distance (NMD) between p and q is*

$$\mathrm{NMD}(p, q) \;=\; \frac{1}{C - 1} \cdot \mathrm{EMD}(p, q).$$

3. Acquiring Knowledge in Astro-Particle Physics

Modern astro-particle physics is a three-decade old research field at the intersection between astronomy and particle physics [15], [24]. In this field, scientific knowledge is continuously expanded with the analysis outcomes of telescope data. In this chapter, we introduce the research goals of astro-particle physics and the machine learning tasks that have to be solved in pursuing these goals.

Where appropriate, we illustrate this introduction with the specifics of the FACT telescope [49], a small gamma-ray telescope on the Canary Island of La Palma, Spain. This focus is motivated in the availability of a comprehensive, open data set[3] and an open analysis pipeline,[4] which we employ throughout the experiments of this thesis. In contrast, the data of most other telescopes is kept private by the collaborations which operate these telescopes. Since this policy hinders reproducibility, we focus on the FACT data.

Most notably, the FACT collaboration has published the *entire set of simulated and labeled data*, which their standard analysis employs. This data is continuously updated. A data set of this kind is indispensable not only for the training of machine learning models, but also for their validation. Hence, we use this data in all astro-particle experiments.

However, not all data of the FACT telescope is public. With the exception of an open sample of Crab nebula observations, no *real data* of the telescope is publicly available. Through our long-standing interdisciplinary collaboration with astro-particle physicists, however, we have been granted access to a considerable part of the closed data. Where appropriate, we leverage this opportunity in the experiments of this thesis. In particular, we use closed meta-data of the effective areas of the telescope in Chap. 4 and we use closed data samples of three gamma ray sources in Chap. 5. Since all of the closed data are unlabeled, however, their applicability is limited to these cases.

We begin the present chapter with Sec. 3.1, where we introduce the research questions of modern astro-particle physics. Sec. 3.2 presents the imaging atmospheric Cherenkov technique, through which ground-based telescopes like FACT detect cosmic gamma radiation to answer these questions. The Cherenkov technique poses several machine learning tasks, which are introduced in Secs. 3.3–3.5. Namely, the reconstruction of energy spectra is presented in Sec. 3.3, the detection of cosmic gamma ray sources in Sec. 3.4, and gamma hadron classification in Sec. 3.5. We detail the production of labeled training data

[3] https://factdata.app.tu-dortmund.de/

[4] https://github.com/fact-project/open_crab_sample_analysis

through simulations in Sec. 3.6 and the processing of the telescope data in Sec. 3.7. Our presentation is complemented with Sec. 3.8, where we discuss probabilistic rationalism, the epistemological background of acquiring knowledge in particle physics.

3.1. Scientific Relevance of Astro-Physical Particles

The astronomical observation of the sky dates back at least to the Stone Ages [61]. Galileo revolutionized this part of human culture in 1609, in being the first who used an optical telescope to observe objects that appear too small or too dim for an observation with the naked eye. In the following centuries, the increasing quality of optical telescopes allowed astronomers to explore the universe in more and more detail. However, optical telescopes are limited to observing the visible light, which only makes up a small part of the electromagnetic spectrum. This limitation was set aside in the 20th century, when additional wavelengths became observable through infra-red telescopes, ultra-violet telescopes, radio telescopes, X-ray telescopes, and sub-millimeter telescopes. In 1989, the first detection of a cosmic gamma ray source by a ground-based telescope was reported [62]. Together, all of these telescopes observe photons that range from low-energy radio waves to high-energy gamma particles. In addition to these photons, Earth is also bombarded by charged particles, which are called cosmic rays, and by neutrinos. In 2016, we further witnessed the first detection of gravitational waves [63] on Earth.

All of the aforementioned types of particles, except for cosmic rays, carry information about their origin. Therefore, they reveal much more about the universe than visible light alone could reveal. Not only can we discover the locations of celestial objects that are not visible otherwise, but also their properties. Even more fundamentally, we can learn about the physical processes that are at work throughout the universe and within the most extreme cosmic environments, which accelerate particles to energies that an artificial particle accelerator on Earth could never provide. Each type of cosmic particle bears its own secrets to be uncovered and its own challenges in being detected, hence the multitude of specialized telescopes.

This thesis is mainly focused on the detection and analysis of very-high energy gamma radiation. Observations of this radiation can help astro-particle physicists in addressing several scientific questions of fundamental physics [22], [64]:

What is the origin of cosmic rays? In 1912, Victor Hess discovered cosmic rays in a series of balloon flights [65]. His discovery was the first evidence for the existence of acceleration processes that exceed the energies explainable by the thermal emission processes that can be observed with optical telescopes. However, no such acceleration process has yet been identified with certainty [64], [66]. Energy spectra of gamma ray sources, see Sec. 3.3, have the potential to support a future identification of such a process.

How do extreme environments accelerate particles? Active galactic nuclei, gamma ray bursts, and binary systems can accelerate particles in directed jets of plasma, a process that is not yet fully understood [66]. Precise measurements of the gamma-ray outflows of these systems are required for improving our understanding of the acceleration processes at work [64].

What is the nature of dark matter? There is strong evidence about the existence of an unknown form of gravitating matter. However, the kinds of particles that make up this so-called dark matter are still to be discovered. High-energy gamma rays have the potential to uncover these particles from distortions [66] and excesses [64] in energy spectra.

As of today, ground-based gamma ray telescopes have detected over 220 sources of very-high energy gamma rays.[5] Precise measurements of these objects have already helped in constraining some of the competing theories about fundamental physical processes at the cosmic scale [61].

3.2. Imaging Atmospheric Cherenkov Telescopes

Imaging atmospheric Cherenkov telescopes (IACTs) are gamma ray telescopes that are placed on the surface of our planet. Since these ground-based telescopes can observe even the most energetic gamma radiation, they are important devices for addressing the scientific questions raised above. FACT [49] is a comparably small telescope of this kind, which is located on the Canary Island of La Palma, Spain. Other notable IACTs of today are HESS [68], [69], MAGIC [70], and VERITAS [71]. Soon, also the Cherenkov Telescope Array (CTA) [72] will become operational. This array of about hundred individual telescopes is expected to outmatch all of the preceding IACTs in terms of scale and sensitivity [64]; however, CTA employs the same detection mechanism as the other IACTs and, hence, poses the same machine learning tasks.

The detection mechanism of IACTs, the imaging atmospheric Cherenkov technique [62], critically depends on simulations and machine learning. The reason for this dependency is that IACTs detect gamma radiation only *indirectly*, through physical processes that are triggered by gamma rays. An analysis of the gamma radiation requires that the properties of the radiation are reconstructed from the processes that an IACT observes. Since no definite rules for this reconstruction are known a priori, it is necessary to learn these rules from data that is simulated. We detail the machine learning tasks, which IACTs pose in this regard, in the following sections.

The indirect detection mechanism of IACTs is sketched in Fig. 3.1. This detection mechanism begins with a primary particle, ideally a gamma ray that originates in a distant, cosmic gamma ray source. When this particle hits Earth's atmosphere, it interacts with the molecules therein, forming an extensive air shower (EAS). This EAS consists of secondary particles, which are products of particle interactions; these secondary particles interact further, adding even more secondary particles to the EAS. Some of the particles

[5] `http://tevcat.uchicago.edu/`, catalog version 3.400, by Wakely and Horan [67]

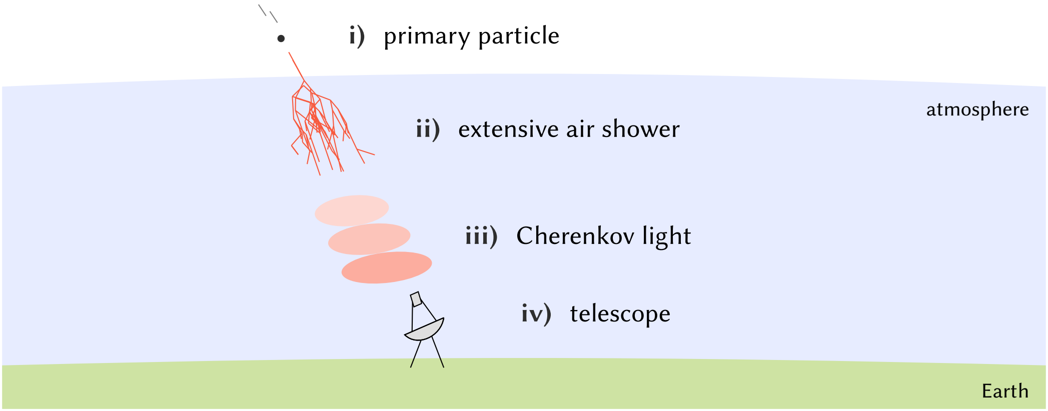

Figure 3.1.: A primary particle, such as a gamma ray or a cosmic ray, interacts in planet Earth's atmosphere. This interaction produces a cascade of secondary particles, the extensive air shower. Some of the secondary particles inside this shower emit Cherenkov light, which is recorded by an IACT. This figure is adapted from Bockermann et al. [16].

in an EAS travel faster than light travels through the atmosphere. Therefore, they emit light, due to the *Cherenkov effect*. The resulting light, the so-called Cherenkov light, can be recorded with specialized cameras. IACTs use these cameras together with large mirrors to leverage the Cherenkov effect in detecting cosmic gamma radiation. The Cherenkov light that an IACT records is a picture of the EAS, and the properties of the EAS are a picture of the primary gamma ray, in which astro-particle physicists are interested. In order to analyze the sources of cosmic gamma radiation, physicists have to reconstruct the properties of each primary particle from the resulting camera recordings.

Fortunately, the interaction processes within the atmosphere are well-understood. This knowledge about particle interactions is implemented in sophisticated simulations, which precisely mimic the real detection mechanism from Fig. 3.1. These simulations allow us to produce synthetic, high-quality IACT recordings, represented by features $x \in \mathcal{X}$, for any hypothetical input gamma ray with properties $y \in \mathcal{Y}$. We turn to the details of this production in Sec. 3.6.

Unfortunately, a simulation $\mathcal{Y} \rightarrow \mathcal{X}$ cannot reconstruct its input y from an observation x that is given by a real IACT. For this purpose, it is necessary to learn a prediction model $\mathcal{X} \rightarrow \mathcal{Y}$ from the simulated data. Operating an IACT hence requires not only simulations, but also machine learning techniques to translate the *fundamental* knowledge of the simulation into *practical* knowledge about the telescope recordings. This practical, learned knowledge aims at the reconstruction of energy spectra (see Sec. 3.3), at the detection of gamma ray sources (see Sec. 3.4), and at the separation of gamma rays from a hadronic background (see Sec. 3.5).

3.3. Reconstruction of Energy Spectra

The fundamental questions about the origin of cosmic rays, particle acceleration processes, and the nature of dark matter are closely connected to *energy spectra*, i.e., to the *distribution* of gamma ray energies that each gamma ray source exhibits. For instance, a strong hint for a gamma ray source being an origin of cosmic rays would be an energy distribution that can only be explained through a hadronic acceleration process [22]. Likewise, hints for dark matter can stem from distortions [66] and excesses [64] in the energy spectra of gamma rays. Answering these fundamental questions of astro-particle physics therefore requires precise measurements of the energy distributions that cosmic gamma ray sources exhibit.

A key difficulty in measuring an energy spectrum with an IACT is that an IACT cannot measure the energy of gamma rays directly. Instead, IACTs record Cherenkov light, which is correlated with the energy, but which does not allow for inferring the definite energy value of each gamma ray. Therefore, the desired energy spectrum has to be reconstructed from the set of Cherenkov light recordings that an IACT has taken for a gamma ray source under study. The correlation between energy and Cherenkov light, as learned from astro-particle simulations, provides the basis for this reconstruction.

Formally, we intend to find the probability density function $P(y) = \mathbb{P}(Y = y)$ of energy values $y \in \mathcal{Y}$. This density can then be scaled to the expected rate of particles per energy, per area, per time, and per solid angle. Typically, an analysis of this kind focuses on an interval $\mathcal{Y} = [E_{\text{min}}, E_{\text{max}})$ of gamma ray energies. However, an IACT measures, instead of y, a set of Cherenkov light recordings $\mathcal{D}_X = \{x_i \in \mathcal{X} : 1 \leq i \leq m\}$, as represented by a feature space $\mathcal{X}$, for the gamma ray source under study. Therefore, the reconstruction of P from $\mathcal{D}_X$ requires a mapping

$$(\cup_{m=1}^{\infty} \mathcal{X}^m) \;\rightarrow\; (\mathbb{R} \rightarrow \mathbb{R}), \tag{3.1}$$

which returns a probability density $P \in (\mathbb{R} \rightarrow \mathbb{R})$ for any input data set $\mathcal{D}_X \in \mathcal{X}^m$. In this formalization, we allow for any number m of Cherenkov light recordings in $\mathcal{D}_X$. The individual recordings $x \in \mathcal{D}_X$ are also called *events*.

The first step in defining the mapping from Eq. 3.1 is to represent $\mathcal{D}_X$ as a probability density $Q(x) = \mathbb{P}_{\mathcal{D}_X}(X = x)$ of the feature space. This representation leads to a probabilistic view-point, where the densities P and Q are related through the conditional probability density $M(x \mid y) = \mathbb{P}(X = x \mid Y = y)$ of measuring x when the unknown energy value of the gamma ray is y. In this view-point, the measured probability density Q is defined through the law of total probability,

$$Q(x) \;=\; \int_{\mathcal{Y}} M(x \mid y) \cdot P(y) \;\mathrm{d}y, \tag{3.2}$$

and the reconstruction of P consists in selecting the most likely P for a given Q and M. At this point, Q is obtained from the telescope recordings $\mathcal{D}_X$ and M is learned from labeled training data $\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq m'\}$, which is provided by a simulation.

The specification of some Q from $\mathcal{D}_X$, the learning of some M from training data, and the selection of P from Q and M require free parameters for Q, M, and P, which are chosen in accordance to the respective data set. In the reconstruction of energy spectra, a typical, approximate representation of probability densities is the histogram [73], [74]. Considering C bins for the gamma ray energy, hence, we limit our reconstructions to probability densities P that can be described as a histogram vector $p \in \mathbb{R}^C$ of bin-wise probabilities $[p]_i = \mathbb{P}(Y \equiv i)$. Likewise, a histogram of Q, with F bins, can be described as a vector $q \in \mathbb{R}^F$ with components $[q]_i = \mathbb{P}(X \equiv i)$. These approximations lead to a system of linear equations,

$$q \;=\; \mathrm{M} \cdot p, \tag{3.3}$$

in which each cell $[\mathrm{M}]_{ij}$ of the matrix M defines the conditional probability of measuring a gamma ray in the j-th bin of q when i is the true bin in p.

The physical interpretation of p requires a scaling from probabilities $[p]_i$ to the expected rate of particles per area, per time, and per solid angle. This scaling is taken out as a component-wise multiplication $a \odot p$, where $a \in \mathbb{R}^C$ is a vector of *acceptance factors*. These factors model the detection efficiency of the telescope for each energy bin. The almost linear appearance, in a log-log plot where both axes are logarithmic, of the corrected vector $a \odot p$ is typically leveraged in the regularization of p. We employ this regularization for ordinal quantification algorithms in Chap. 4. For the FACT data, we have computed the acceptance factors a from closed meta-data, to which we have been granted access through our long-standing collaboration with astro-particle physicists.

The histogram representation of P essentially maps the continuous target quantity Y to an ordered set of classes, where each bin represents one class. In principle, we could approach the estimation of p by learning a classifier, which predicts the energy bin of each event, from simulations; we could then count the number of predictions in each bin to estimate the histogram p for a data sample $\mathcal{D}_X$. While this simple counting of prediction is not Fisher-consistent [75], it connects the reconstruction of energy spectra to ordinal quantification, a machine learning task that we elaborate in Chap. 4.

An alternative way of estimating energy spectra is to pre-impose a parametric family of functions for P, typically with parameters that are interpretable in terms of physical theories about the spectrum. While interpretability is an advantage of this approach, the assumption of a parametric family of functions, and the theory they represent, poses a clear disadvantage. From this disadvantage stems a risk of overfitting: does a parameter value stem from a real physical phenomenon or do we witness a circular argument due to the pre-imposed theory? In physics literature, the parametric approach is called “Forward folding”, while the solution of Eq. 3.3 is called “unfolding”. In this work, we focus on the latter, due to its lower risk of overfitting, due to its importance for addressing fundamental questions of physics, and due to its close connection to ordinal quantification.

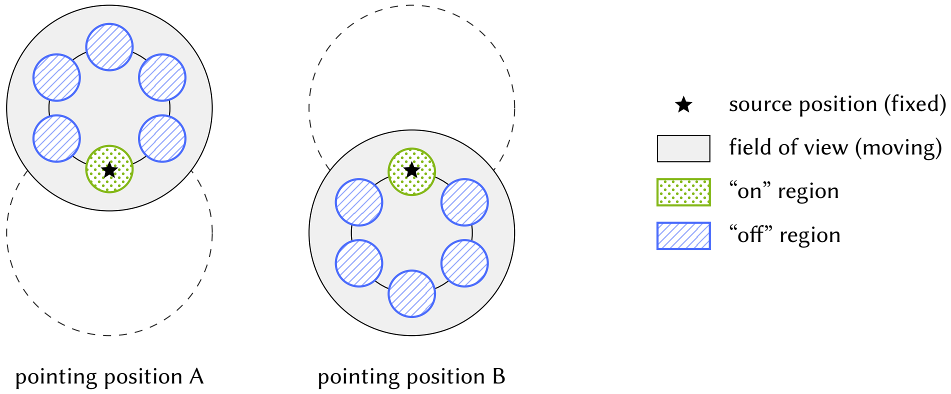

Figure 3.2.: Illustration of the wobble mode for source detection. We are displaying a movement of the camera, which alternates between two pointing positions A and B. Each pointing position places the assumed source position at a non-zero distance from the center of the field of view. The region around this position is observed as the "on" region. Other regions, at the same distance from the center, are simultaneously observed as "off" regions. After some time, the telescope moves to the other pointing position. The FACT telescope [49] typically employs two pointing positions and five background regions for observing the Crab nebula, as illustrated here, but other numbers of pointing positions and background regions are also feasible.

3.4. Source Detection

Before some particular source of gamma radiation can be analyzed, e.g., in terms of its energy spectrum, we need to establish a sufficiently certain *detection* of this source. This requirement translates to finding sufficient empirical support for the claim that the IACT is able to "see" the source in the data. This support is not yet established by merely recording some gamma ray candidates from the direction of the source because most of these candidates might not actually be gamma rays. In particular, an IACT can mistake other particles as gamma rays, as we detail in the next section. Due to these background events, we must show that the direction of an assumed gamma ray source exhibits a sufficiently large excess in gamma radiation over other directions, where no gamma ray source is assumed. Only then we can claim that the telescope has successfully detected a gamma ray source.

We refer to the region around the assumed gamma ray source as the "on" region and to any region where no gamma ray source is assumed as an "off" region. The IACT must record data from both types of regions to enable an assessment of the excess in gamma radiation that the "on" region exhibits. A simultaneous recording of both types is taken out, for instance, in the so-called *wobble mode* from Fig. 3.2. This mode of operating an IACT points the telescope such that the "on" region is placed at some distance from the center of the telescope's field of view. At the same distance, multiple "off" regions are placed. As time passes, the telescope moves between different pointing positions, each of which observes "on" and "off" regions simultaneously. This characteristic, alternating movement of the telescope is colloquially described as "wobbling", which coined the name of the wobble mode of operation.

Observing an assumed gamma ray source over an extended period of time provides us with N_{on} gamma ray candidates in the "on" region and N_{off} background gamma ray candidates in all "off" regions. A detection requires us to show that the *excess count* of gamma rays, which we have observed in the "on" region,

$$N_{\text{on}} \; - \; \alpha \cdot N_{\text{off}}, \tag{3.4}$$

is not just a random artifact but a statistically *significant* outcome of the observation. Here, the scaling factor

$$\alpha \; = \; \frac{A_{\text{on}}}{A_{\text{off}}}. \tag{3.5}$$

allows for a fair comparison between N_{on} and N_{off} even if the area A_{on} of the "on" region and the total area of all "off" regions, A_{off}, differ from each other. Such different areas are typical; in Fig. 3.2, for instance, we have $n = 5$ "off" regions, each spanning the same area as the "on" region. Hence, the total area of all "off" regions is $A_{\text{off}} = n \cdot A_{\text{on}}$ and a scaling in terms of α is necessary.

To establish the statistical significance of an observation, Li and Ma [76] have proposed a hypothesis test, which is, by now, the conventional way of realizing source detections with IACTs. This hypothesis test appropriately models the counts N_{on} and N_{off} as being Poisson-distributed with the rates λ_{on} and λ_{off}. The hypothesis test consists in checking whether these rates, scaled by α, are indeed significantly different from each other.

Definition 3.1 (Li & Ma hypothesis test for source detection [76]). *Let N_{on} be the number of gamma ray candidates in the "on" region and let N_{off} be the number of background gamma ray candidates in all "off" regions. Moreover, let $\mathcal{N} : \mathbb{R} \rightarrow [0, 1]$ be the cumulative density function of the standard normal distribution. The p-value for rejecting the null hypothesis $h_0 : \lambda_{\text{on}} \leq \alpha \cdot \lambda_{\text{off}}$ is*

$$\begin{aligned} p = \;& 1 - \mathcal{N}(f_\alpha(N_{\text{on}}, N_{\text{off}})), \\ f_\alpha(N_{\text{on}}, N_{\text{off}}) = \;& \Bigg[2N_{\text{on}} \cdot \ln\left(\frac{1+\alpha}{\alpha} \cdot \frac{N_{\text{on}}}{N_{\text{on}} + N_{\text{off}}} \right) \\ & \quad + 2N_{\text{off}} \cdot \ln\left((1+\alpha) \cdot \frac{N_{\text{off}}}{N_{\text{on}} + N_{\text{off}}} \right) \Bigg]^{1/2}. \end{aligned}$$

In gamma ray astronomy, a detection typically succeeds when the test statistic f_α exceeds a value of 5. We are then speaking of a "five sigma detection", which amounts to a p-value of $2.87 \cdot 10^{-7}$. At this immense level of required significance, it is more practical, and thus conventional in gamma ray astronomy, to report the values of f_α instead of reporting the

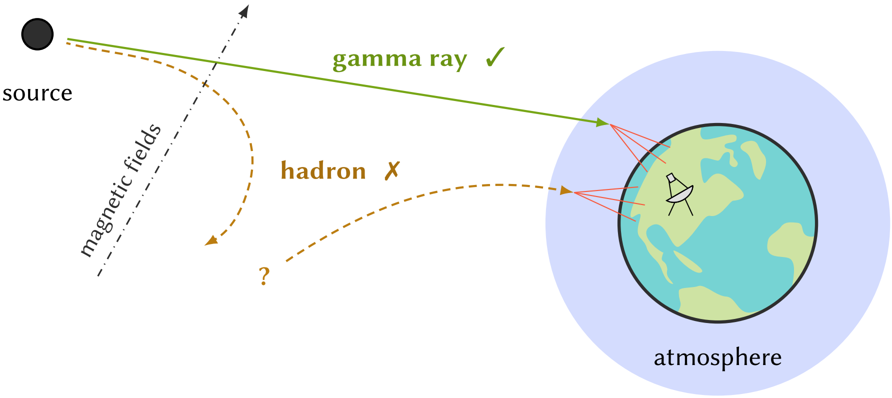

Figure 3.3.: A cosmic source emits gamma rays and hadronic particles. While gamma rays travel straight through space, hadrons are being deflected by inter-stellar magnetic fields. Therefore, we cannot know in which cosmic source a hadron originates. Unfortunately, both particle classes, gamma rays and hadronic particles, induce extensive air showers in Earth's atmosphere, which are then recorded by IACTs. To obtain a clean sample of gamma ray observations, we have to separate these two particle classes from each other.

corresponding p-values. In Chap. 5, we use f_α to evaluate classification models because a high significance of a detection is an indicator of an effective gamma hadron classifier. To this end, we employ closed data of three gamma ray sources.

3.5. Gamma Hadron Classification

Reconstructions of energy spectra and detections of gamma ray sources require IACT recordings of cosmic gamma rays, in particular. Unfortunately, the extensive air showers that IACTs record are not only triggered by gamma rays. Another class of primary particles, which trigger air showers, consists of *hadronic* particles. As a consequence, these particles end up as additional, undesired IACT recordings. An analysis of the gamma radiation, e.g., the reconstruction of an energy spectrum or the detection of a gamma ray source, therefore requires the separation of the relevant gamma ray recordings from the undesired hadron recordings.

The reason why hadronic particles are undesired is illustrated in Fig. 3.3. In fact, these particles can originate in the same sources as the relevant gamma rays. However, hadronic particles are *charged*. Therefore, they are deflected by inter-stellar magnetic fields, thus reaching Earth from directions that do not reveal the position of their origin. Even if all magnetic fields were known, the uncertainty in their energy estimates would render a position estimation infeasible [61]. Gamma rays, in contrast, are *not* deflected by magnetic fields; they travel in straight lines, thereby allowing us to reconstruct the positions of their origins from IACT recordings. This difference between gamma rays and hadronic particles makes a separation between the two particle classes indispensable.

The separation of gamma rays from hadronic particles amounts to a classical, binary classification task $\mathcal{X} \rightarrow \{\text{“gamma”}, \text{“hadron”}\}$, where $\mathcal{X}$ is again the feature space of IACT recordings. Typically, the binary classifiers for this task are trained with simulated data [16], [39], [77], for which the binary class label is known by design. The real telescope observations are never labeled in these terms. Alternatively, however, we see in Chap. 5 that *noisy*, weak labels from the *real* observations can be employed for training competitive gamma hadron classifiers without any simulated data.

3.6. Simulation of Telescope Data

The reconstruction of energy spectra, the detection of gamma ray sources, and the classification into gamma rays and hadrons require labeled training data to learn to predict particle properties $y \in \mathcal{Y}$ from unlabeled telescope recordings $x \in \mathcal{X}$. These particle properties comprise the particle class (gamma ray or hadron), the energy of the particle, and its direction of origin. For the real telescope data, these properties are never known because no artificial source of these particles exists for the energy regime that is probed by IACTs. Hence, no real, labeled data is available and simulations must be employed, instead. Fortunately, the FACT collaboration has published the entire set of simulated data that are used in their standard analysis.[4]

In order to produce realistic data, from which valid prediction models can be learned, it is necessary to simulate all processes that “generate” the data in the real world. As shown in Fig. 3.1, these processes include the formation of an extensive air shower (EAS), the production of Cherenkov light, and the data acquisition through an IACT. Due to the diversity of these processes, physicists have developed the simulation as a combination of several components, each of which is dedicated to some particular aspect of the real measurement process. These components are:

Particle Interactions The simulation of particle interactions begins in the moment in which a primary particle enters Earth’s atmosphere. From there on, the simulation covers the development of the entire EAS, as shown in steps i) and ii) of Fig. 3.1. For this purpose, the widely-used simulation software CORSIKA [78] employs separate models for electromagnetic interactions (e.g., EGS4 [79]), for hadronic interactions below 80 GeV (e.g., UrQMD [80]) and for hadronic interactions above 80 GeV (e.g., EPOS-LHC [81]) [22]. Each of these models comprises a plethora of physical processes, e.g., annihilation, scattering, bending, and particle production processes [82]. Hence, they require comprehensive knowledge of the involved, fundamental particle interactions. However, the small angles, at which some of these interactions occur, exceed the capacity of physical experiments, like the large hadron collider at CERN. Therefore, the involved processes have to be extrapolated, in terms of theoretical models, from lower energy regimes [78]. This extrapolation is the subject of continuous improvements of the simulation pipeline.

Cherenkov Light Production The sensitivity of IACTs requires an accurate simulation of the Cherenkov light that an EAS produces, see step iii) in Fig. 3.1. This requirement introduces a necessity to account for the density and the refraction of the atmosphere, which have to be measured or estimated for all relevant altitudes. In CORSIKA, the production of Cherenkov light is implemented through the iact/atmo extension [82].

Detector Response An IACT consists of a segmented reflector with multiple mirrors, of a specialized camera, and of read-out electronics. Each of these components has an influence on the telescope recordings that needs to be simulated. For this purpose, the simulation software CERES [83] models the absorption of photons, their reflection in the mirrors, and the behavior of the camera and the electronics, see step iv) in Fig. 3.1. The input of CERES is the position and arrival time of each Cherenkov photon that CORSIKA has simulated.

Many of the software components mentioned above are being continuously developed since decades. Most importantly, they are being continuously updated with the latest fundamental knowledge that particle physicists acquire about the interactions of the involved particles. So to speak, the simulated data comprises nothing less than the state-of-the-art of our understanding of particle interactions inside Earth's atmosphere. Therefore, the simulation is an ideal provider for labeled training data that agrees well with the data of a real IACT. However, the level of detail that the simulation exhibits poses a considerable challenge with respect to the resource footprint of IACT analyses. We address this challenge in Chap. 6.

The input of the simulation pipeline is the description of some primary particle [22], [78] in terms of particle properties $y \in \mathcal{Y}$ that an IACT cannot measure directly. For each input particle, the simulation produces a synthetic telescope observation $x \in \mathcal{X}$, similar to the detection process of a real IACT. The description of the primary particle includes:

Particle Energy One input of the simulation is the desired particle energy. This information is later used in the reconstruction of energy spectra.

Direction of Origin Another input of the simulation is the direction, in which the primary particle originates. This information is later used to predict whether a telescope observation comes from the "on" region or from an "off" region.

Particle Class Also the particle class, i.e., gamma ray or hadron, is specified as an input of the simulation. This information is later used as the class label for training gamma hadron classifiers.

Typically, the data are produced in batches, where the input either consists of fixed values or of distributions of particle properties. In the latter case, the simulation software samples all individual particles in the batch from the given distribution.

Since the simulation pipeline generates data only for specifically chosen classes, we can consider this pipeline as a *class-conditional* data generator $\mathcal{Y} \rightarrow \mathcal{X}$, for which we elaborate the employment of active class selection techniques in Chap. 6. One particular aspect that we cover in this regard is that the real-world distribution of particle classes is not precisely known; hence, uncertainties have to be addressed.

The investigation of complex systems through simulations finds applications not only in astro-particle physics, but also in several other areas of science and engineering. Recently, there is an increase of attention towards the employment of simulated data particularly for machine learning, an integration that is sometimes termed *simulation data mining* [84]–[87]. The goal of this integration is to reason about a real system under study, by learning from data which are generated by a simulation of that system. In fact, acquiring data from the real system is often costly or even impossible, e.g., if the actual system is still in the design phase and not yet deployed. In contrast, simulations have the potential to provide large volumes of data, only at the expense of their computation. However, the need for accurate simulations often leads to complex simulation models, which result in high resource demands.

3.7. Data Processing

The raw data from the telescope and the raw data from the simulation consist of small video sequences that picture the evolvements of extensive air showers over short time scales. The FACT telescope, for instance, records a sequence of 300 frames, which cover just 150 nanoseconds, where each frame contains 1440 pixels [22]. Each of the two MAGIC telescopes records a sequence of 50 frames, which cover just 30 nanoseconds, where each frame contains 1039 pixels [20]. Not only is this raw data extremely noisy; it is also biased by environmental conditions, such as temperature and background light. These conditions affect the hardware of the telescope and, hence, the data.

In order to facilitate effective machine learning for IACTs, it is necessary to de-noise and correct the raw data of the telescope and the raw data of the simulation. Further processing of the data additionally assists the learning algorithms in finding meaningful patterns. Hence, the data processing pipeline of IACTs is a central element in predicting the properties of the primary particles, which are indispensable in the reconstruction of energy spectra, in the detection of gamma ray sources, and in gamma hadron classification. In particular, a typical processing pipeline consists of the following steps:

Calibration and Correction Changing environmental conditions require the telescope to be re-calibrated frequently. For this purpose, the environmental conditions are tracked and their influence on the raw video sequences is corrected [88]. Also, the inherent noise of the data, e.g., sampling errors and spike voltages, is rectified. Some simulations allow us to omit some of these actions because the corresponding artifacts are not being simulated. For instance, CERES [83] does not simulate sampling errors and possible jumps in pixel intensities [22]. The real data always require this processing step.

Signal Extraction To reduce the complexity of the data, each of the calibrated video sequences is reduced to two images that represent the video sequence sufficiently well [22], [89]. The first image pictures the total, estimated number of photons that is recorded by each pixel over the entire video sequence. The second image pictures the mean, estimated arrival time of photons in each pixel.

Image Cleaning Typically, the majority of pixels does not picture the relevant EAS but background light or mere noise. For instance, the stars and the moon can easily, though undesirably, trigger the highly sensitive camera of an IACT. The removal of these pixels is often a custom procedure that accepts and discards individual pixels based on global and local thresholds [22], [89].

Feature Extraction EAS have an approximately elliptical shape, which allows for a representation of the data in terms of geometric features, such as width, size, and orientation of the shower, as it appears in the cleaned image. In addition to these so-called *Hillas parameters* [90], other features have been proposed, e.g., the percentage of light in the ellipse, the percentage of light in the brightest pixels, or the percentage of light in the outer ring of the camera frame [20], [22]. The standard analysis[4] of the FACT telescope, for instance, considers 20 features, while the extraction of more than 100 features is implemented [88].

This selection of features is motivated in a trade-off between models that accurately predict the simulations and models that generalize well from the simulated data to the real telescope recordings. To this end, astro-particle physicists have selected a set of features that permits effective learning while ensuring validity also in terms of the real telescope.

The data processing pipeline of the FACT telescope is implemented in the FACT-tools[6] software, which is based on the streams framework [88]. This implementation is able to handle the data acquisition rate of FACT, which amounts to up to 1 TB per night, in near real-time even on a small-scale desktop computer. Its extensions to large-scale, distributed systems further allow for a fast processing of historical, big data.

However, not all of the above processing steps are indispensable. Deep learning promises to not only learn a prediction model but also a suitable sequence of feature representations [91]. For the FACT data, this promise is demonstrated to be kept, at least with respect to the last two processing steps. In particular, competitive gamma hadron classifiers can be learned from the images that are produced by the signal extraction step [77].

In this work, all experiments with the FACT data are based on the feature representation of the standard analysis[4]. Moreover, we always employ random forest classifiers, which are also a part of this analysis pipeline. Thereby, we fully leverage the available domain knowledge of physicists with the goal of producing analysis outcomes that advance this

[6] `https://github.com/fact-project/fact-tools`

knowledge as far as machine learning can. Reusing the existing pipeline [22], [88] allows us to fully focus on the machine learning tasks, which evolve around this pipeline.

3.8. Probabilistic Rationalism

We now turn to the philosophical background of knowledge expansion in astro-particle physics. This background is provided through Wolfgang Rhode's *probabilistic rationalism* [24], a recent epistemological view that is based on Karl Popper's conception of a *critical rationalism* [92]. While Popper's view verifies scientific theories only in terms of dichotomous decisions, the central proposal of probabilistic rationalism is to verify theories in probabilistic terms.

Both forms of rationalism, critical and probabilistic, found the scientific advancement of knowledge on the falsification of theories through experiments. In this foundation, the goal of any experiment is to falsify a connection between measurable quantities that is predicted by a rationally conceived theory. If this falsification is successful, the theory must be improved or rejected. If not, the experiment has to be re-designed under more rigorous conditions. Any theory that cannot be falsified through an experiment is outright rejected for not being considered scientific. Whatever theory remains under these requirements, at present, is considered to be the current state-of-the-art [24].

The difference between critical and probabilistic rationalism lies in their conceptions of a successful falsification. In Popper's critical rationalism, falsification is dichotomous: a theory is either falsified or it is not. This dichotomy is suitable for theories that consist in a simple statement that can only be either true or false. In Rhode's probabilistic rationalism [24], however, a theory is allowed to exhibit any probability. This freedom is indispensable in modern astro-particle physics, where complex theories with large numbers of free parameters have to be tested and a dichotomous decision is not always appropriate. In fact, Popper's dichotomous falsification is not even intended for describing the actual progression of physics research [24]. The shift from dichotomous statements to probabilistic ones is also rooted in the physical world: processes such as quantum effects are not only probabilistic in our understanding, but probabilistic in their very own nature.

Probabilistic rationalism manifests in an iterative scientific process, which continuously adapts the probabilities of competing theories to the latest experimental findings. For astro-particle physics, this iterative process is illustrated in Fig. 1.1: existing, fundamental knowledge is encoded in a simulation, from which machine learning models are trained to reconstruct physically meaningful quantities from the experimental raw data. Improvements in the fundamental knowledge propagate to the analysis outcomes, which in turn contribute to the body of fundamental knowledge. In the following chapters, we elaborate the machine learning tasks that contribute to this iterative process of probabilistic knowledge expansion in astro-particle physics.

4. Ordinal Quantification

Quantification [93], [94] is the task of predicting the probability of each class in an unlabeled data sample. This supervised learning task is in contrast to “standard” classification learning, where predictions for individual data items, and not for samples of items, are desired to be accurate. It is also in contrast to simple counting, where the class of each data item is known and does not need to be predicted [95]. Applications of quantification arise in text sentiment analyses [96], in the social sciences [97], in technical support log analyses [93], and in several other areas. The task of quantification is also known as “class prior estimation” and “prevalence estimation”, among other names [94].

In astro-particle physics, the reconstruction of energy spectra, see Sec. 3.3, poses an *ordinal* quantification task because the energy bins, which we predict, are totally ordered. Reconstruction methods from the physics literature, going under the name of “unfolding” [74], [98], [99] or “deconvolution” [1], address the order of classes through regularization. Most other quantification research, however, is focused on the non-ordinal setting. Between these two areas of quantification research, we recognize an interdisciplinary gap. In fact, the literature from quantification research and the literature from unfolding research have developed in the unawareness of each other, despite substantial similarities in terms of their problem statements and their solutions. In this chapter, we contribute to quantification and unfolding research in several ways.

First, Sec. 4.1 unifies algorithms from quantification literature and algorithms from unfolding literature in a single, common framework [2]. This unification, which also addresses non-ordinal settings, validates our claim that unfolding and quantification are indeed the same mathematical problem. Our presentation includes a comparison of multiple, different multi-class extensions [3] to one of the most important methods in quantification, the Adjusted Classify and Count method [93]. Our common framework solves quantification tasks by the means of constrained, numerical optimization techniques.

In Sec. 4.2, we propose a novel, alternative solution, which circumvents constrained optimization techniques by embedding the constraints in the quantification model. As a result, the constraints become invariances, which allow us to solve quantification tasks effectively, by the means of *un-constrained* optimization techniques.

Drawing inspiration from unfolding methods, we propose novel, regularized variants of existing quantification methods in Sec. 4.3. Through the regularization, these variants address ordinal quantification in particular. We locate the general difficulty of solving quantification in Sec. 4.4 and we empirically evaluate our proposals against the existing quantification methods in Sec. 4.5.

4.1. Unification of Algorithms for Quantification and Unfolding

In this section, we propose a common framework for algorithms that stem from quantification literature and from unfolding literature [2]. This framework reveals several similarities between existing methods from the two research fields. Moreover, it paves the way for strengthening the interdisciplinary efforts on the subject. A similar attempt on unifying existing quantification methods is proposed by Firat [100], who recognizes that many quantification methods can be phrased in terms of constrained, numerical optimization tasks. However, he does not formally prove the correctness of his claims. We complete this attempt on unifying existing quantification methods in terms of i) taking unfolding algorithms from the physics literature into consideration and ii) giving formal proofs about the correctness of our framework. Our unification covers algorithms for ordinal and non-ordinal quantification.

Recall from Eq. 3.2 that our goal is to estimate the probability density P from the integral $Q(X = x) = \int_{\mathcal{Y}_c} M(X = x \mid Y = y) \cdot P(Y = y) \, \mathrm{d}y$. The estimation of P from data is enabled through the discretization of this integral, which yields the system of linear equations, $q = \mathrm{M} \cdot p$ with histograms $q \in \mathbb{R}^F$ and $p \in \mathbb{R}^C$, from Eq. 3.3. In case of a continuous energy interval $\mathcal{Y}_c = [E_{\text{min}}, E_{\text{max}})$, we first need to map each continuous energy value to a discrete class index $\mathcal{Y} = \{1, \ldots, C\}$. For instance, the estimation of an energy spectrum requires a binning of the interval $\mathcal{Y}_c$ into C bins [73], [74]. Due to the order of the bins, which are used as classes, we recognize that the reconstruction of energy spectra is an ordinal quantification task. Typically, the bin boundaries are chosen to be equi-distant on a logarithmic energy scale [22].

We proceed similarly with the feature space $\mathcal{X} \subseteq \mathbb{R}^d$, in mapping it to a discrete feature representation $f(x) \in \{1, \ldots, F\}$. This representation is still to be defined for each unfolding / quantification algorithm in particular. Possible choices are, for instance, classifier predictions [93], a k-means clustering [1], or leaf indices in a decision tree [101].

Outside of the reconstruction of energy spectra, the set of classes is typically fixed. For instance, a training set in sentiment quantification might define the classes {“positive”, “neutral”, “negative”} [96], or a marine plankton recognition system might deal with the prevalence of a single “plankton” class [102]. With such a fixed set, there is no need for defining the classes based on continuous energy values. The discretization of the feature space, however, is a necessary step in *any* algorithm for quantification / unfolding.

The discretization of y and x gives rise to a straightforward representation of distributions in terms of histograms and sample averages. Consider an unlabeled data sample $\mathcal{D}_X = \{x_i \in \mathcal{X} \times \mathcal{Y} : 1 \leq i \leq N\}$. Assuming that each data example has some un-observed ground-truth class label y_i, we estimate the quantities from Eq. 3.2 in terms of histograms p and sample averages q. In particular, letting

$$p = \frac{1}{N} \sum_{i=1}^{N} \delta_{y_i}, \qquad q = \frac{1}{N} \sum_{i=1}^{N} \delta_{f(x_i)}, \qquad [\delta_j]_k = \begin{cases} 1 & \text{if } j = k, \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

leads to the system of linear equations, $q = \mathrm{M} \cdot p$, from Eq. 3.3. Here, the transfer matrix $\mathrm{M} \in \mathbb{R}^{C \times F}$ is estimated by counting and normalizing the co-occurrences of labels y and transformed features $f(x)$ in a training set. Advanced algorithms are required to estimate p because a direct solution $\mathrm{M}^{-1}q$ typically leads to estimates that are no valid probability densities, are not physically plausible, or are not even defined. To ensure valid estimates p, we define the feasible set of solutions as follows.

Definition 4.1 (Feasible Solutions in Unfolding and Quantification). *With $C \geq 2$ classes, the goal of unfolding and quantification is to estimate a vector $p \in \mathcal{P}$ of class prevalences, where the set of feasible solutions is the unit simplex*

$$\mathcal{P} \;=\; \left\{ p \in \mathbb{R}^C \;:\; [p]_i \geq 0 \;\forall\; 1 \leq i \leq C \;\;\wedge\;\; 1 = \sum_{i=1}^{C} [p]_i \right\}.$$

All elements of this set are valid probability densities over the set of classes $\mathcal{Y} = \{1, \ldots, C\}$, as required by unfolding / quantification tasks.

This feasible set allows us to define an appropriate constraint in a common framework for unfolding and quantification algorithms. Our framework extends the one by Firat [100] with regularization functions $r(p)$.

Definition 4.2 (Common Framework for Unfolding and Quantification [2]). *Algorithms for unfolding and quantification solve the system of linear equations from Eq. 3.3, $q = \mathrm{M} \cdot p$, for the vector p of the class probabilities in a data set that is characterized by q. A general, regularized solution to this problem, with a regularization strength $\tau \geq 0$, is*

$$p^* \;=\; \operatorname*{arg\,min}_{p \in \mathcal{P}} \mathcal{L}(p\,;\, q, \mathrm{M}) + \tau \cdot r(p),$$

where the loss function $\mathcal{L} : \mathbb{R}^C \to \mathbb{R}$, the regularization function $r : \mathbb{R}^C \to \mathbb{R}$, and the feature transformation $f(x)$, which defines q according to Eq. 4.1, are still to be defined for each particular unfolding / quantification method.

The general solution from Def. 4.2 holds several opportunities for the development and the analysis of unfolding and quantification methods. Namely, we can

- employ different loss functions $\mathcal{L}$
- employ different regularization terms r
- employ different feature transformations $f(x)$
- follow different approaches to the numerical minimization of Def. 4.2

Adhering to this framework are the most important algorithms from unfolding and quantification, e.g., RUN [73], [98], SVD [99], IBU [74], [103], ACC [93], [97], PACC [104], ReadMe [97], HDx and HDy [105], CC [93], and PCC [104]. We detail these algorithms in the following sub-sections and develop additional methods for ordinal quantification in Sec. 4.3. An overview of all algorithms from our framework is given in Tab. 4.1.

Table 4.1.: Algorithms for unfolding and quantification within the framework of Def. 4.2. This table is adapted from one of our publications [2].

	loss function $\mathcal{L}$	regularizer r	feature transformation f
RUN [73], [98]	$\sum_{i=1}^{d} [\mathrm{M}\bar{p}]_i - \bar{q}_i \ln[\mathrm{M}\bar{p}]_i$	$\frac{1}{2}\left(\mathrm{T}p\right)^2$	any $f : \mathcal{X} \to \{1, \dots, F\}$
SVD [99]	$\left\lVert \frac{q - \mathrm{M}p}{w} \right\rVert_2^2$	$\frac{1}{2}\left(\mathrm{T}p\right)^2$	any $f : \mathcal{X} \to \{1, \dots, F\}$
IBU [74], [103]	expectation maximization	smoothing	any $f : \mathcal{X} \to \{1, \dots, F\}$
ACC [93], [97]	$\lVert q - \mathrm{M}p \rVert_2^2$	none	$\delta_{\arg\max_i [h(x)]_i}$
PACC [104]	$\lVert q - \mathrm{M}p \rVert_2^2$	none	$h(x)$
ReadMe [97]	$\lVert q - \mathrm{M}p \rVert_2^2$	none	$\delta_{x=(X_1,\dots,X_{2^d})}$
HDx [105]	$\frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M}p)$	none	$\left(\delta_{b(x;1)}, \dots, \delta_{b(x;d)}\right)$
HDy [105]	$\frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M}p)$	none	$\left(\delta_{b(h(x);1)}, \dots, \delta_{b(h(x);C)}\right)$
CC [93]	none (assume $\mathrm{M} = \mathbb{I}$)	none	$\delta_{\arg\max_i [h(x)]_i}$
PCC [104]	none (assume $\mathrm{M} = \mathbb{I}$)	none	$h(x)$
o-ACC (Sec. 4.3.1)	$\lVert q - \mathrm{M}p \rVert_2^2$	$\frac{1}{2}\left(\mathrm{T}p\right)^2$	$\delta_{\arg\max_i [h(x)]_i}$
o-PACC (Sec. 4.3.1)	$\lVert q - \mathrm{M}p \rVert_2^2$	$\frac{1}{2}\left(\mathrm{T}p\right)^2$	$h(x)$
o-HDx (Sec. 4.3.2)	$\frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M}p)$	$\frac{1}{2}\left(\mathrm{T}p\right)^2$	$\left(\delta_{b(x;1)}, \dots, \delta_{b(x;d)}\right)$
o-HDy (Sec. 4.3.2)	$\frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M}p)$	$\frac{1}{2}\left(\mathrm{T}p\right)^2$	$\left(\delta_{b(h(x);1)}, \dots, \delta_{b(h(x);C)}\right)$

4.1.1. Regularized Unfolding (RUN)

The RUN method by Blobel [73], [98] is among the most influential works on unfolding. In fact, all other methods that we consider reference Blobel's tech report from 1985 [98] for the ground it has set. In 2002, the method received an update [73], which is still widely used today [106], [107]. The foundation of RUN is a likelihood function, from which we derive a loss function $\mathcal{L}$, that is combined with a regularizer r.

Feature Transformation To model the likelihood function of RUN, we consider any feature transformation $f : \mathcal{X} \to \{1, \dots, F\}$ that partitions the feature space by mapping each data item to an integer. This partition can be induced, for instance, by a k-means clustering [1], by a decision tree [101], or by the predictions of a classifier. In case of a k-means clustering, we set F to the number of clusters; in case of a decision tree partition, we set F to the number of leaf nodes; and in case of classifier predictions, we set F to the number of classes. A feature transformation of this kind results in q being a histogram of the feature space $\mathcal{X}$ with bins that correspond to the partition indices $\{1, \dots, F\}$.

Loss Function RUN's likelihood function, from which we derive the loss, is defined over the absolute counts $\bar{q} \in \mathbb{N}^F$ of data items in each bin of the feature space,

$$[\bar{q}]_i \;=\; N \cdot [q]_i \;=\; \sum_{k=1}^{N} \delta_{f(x_k)} \quad \forall\, 1 \leq i \leq F, \tag{4.2}$$

and over the unknown, absolute counts $\bar{p} \in \mathbb{N}^C$ of data items in each class,

$$[\bar{p}]_j \;=\; N \cdot [p]_j \;=\; \sum_{k=1}^{N} \delta_{y_k} \quad \forall\, 1 \leq j \leq C. \tag{4.3}$$

The assumptions behind the likelihood function of RUN are i) that the components of $\bar{q}$ are independent from each other and ii) that the count $[\bar{q}]_i$ of data items, which is observed in each component, is Poisson-distributed with the rate

$$\lambda_i \;=\; \mathrm{M}_i^\top \bar{p}, \tag{4.4}$$

where M_i is the i-th column vector of M. More compactly, we can also represent a vector of these Poisson rates through the matrix product $\mathrm{M}\bar{p}$.

With the probability density function of the Poisson distribution,

$$\mathbb{P}(Z = z) \;=\; \frac{\lambda^z e^{-\lambda}}{z!}, \tag{4.5}$$

we define the loss function of RUN as the negative log-likelihood

$$\mathcal{L}^{\mathrm{RUN}}(p\,;\,q, \mathrm{M}) \;=\; -\ln \prod_{i=1}^{F} \mathbb{P}(Z_i = [\bar{q}]_i) \;=\; \sum_{i=1}^{F} [\mathrm{M}\bar{p}]_i - \bar{q}_i \ln[\mathrm{M}\bar{p}]_i. \tag{4.6}$$

Regularization The goal of RUN's regularization term is to penalize candidate solutions p, in which neighboring classes exhibit large differences in terms of their probabilities $[p]_i$, $[p]_{i+1}$. The desire for a penalty of this kind is rooted in the ordinal nature of unfolding: any solution with large differences between neighboring classes is considered "un-physical" in the sense of not being plausible against the background of what we expect from a physical processes. Moreover, if the set of classes was not totally ordered, there would not even be a well-defined concept of class neighborhood. Well-behaved candidate solutions p, where the differences between neighboring classes are sufficiently small, are called "smooth solutions". In RUN, smooth solutions are ensured through Tikhonov regularization, i.e., through a regularization term $\frac{1}{2}\left(\mathrm{T}p\right)^2$. Here, the Tikhonov matrix $\mathrm{T} \in \mathbb{R}^{C \times C}$ is defined such that

$$\frac{1}{2}\left(\mathrm{T}p\right)^2 \;=\; \frac{1}{2}\sum_{i=2}^{C-1}\left([p]_{i-1} - 2[p]_i + [p]_{i+1}\right)^2 . \tag{4.7}$$

The energy spectra of cosmic gamma ray sources exhibit their particular smoothness *only in acceptance-corrected log-log plots*, as we have noted in Sec. 3.3. Tikhonov regularization is therefore most appropriate when applied to a logarithmic and acceptance-corrected unfolding / quantification result. To leverage the full potential of Tikhonov regularization, we hence regularize with respect to $\log_{10}(a \odot p)$ instead of p, where $\odot$ denotes an element-wise multiplication of the acceptance factors a and the unfolding / quantification estimate p. Under this domain-specific transformation, the regularization function of RUN becomes

$$r^{\mathrm{RUN}}(p;\, a) \;=\; \frac{1}{2}\big(\mathrm{T} \cdot \log_{10}(a \odot p)\big)^2, \tag{4.8}$$

where the acceptance factors $a \in \mathbb{R}^C$ are computed from closed meta-data, to which we have access through our long-standing collaboration with astro-particle physicists. We emphasize that other transformations of p might be more suitable in other areas of application. These transformations are just as easily fed into the standard Tikhonov regularization term from Eq. 4.7.

Having defined the loss function and the regularization term of RUN, we now relate our unified notation to the original notation of the algorithm. This relation proves that RUN is indeed an instance of our unified framework of unfolding / quantification.

Proposition 4.1. The RUN algorithm [73], [98] is defined over the loss function from Eq. 4.6 and the regularization term from Eq. 4.7. Through these definitions, RUN qualifies as an instance of our common framework from Def. 4.2.

Proof. The loss function we present in Eq. 4.6 is a verbatim statement by Blobel [98, Eqs. (2.29), and (2.26)]. The original algorithm from 1985 [98] treats the elements of p as B-spline coefficients; however, the 2002 version by the same author [73] employs histograms, which are consistent with our Eq. 4.1. Due to this change "the second derivative in bin j is proportional to $x_{j-1} - 2x_j + x_{j+1}$" [73], where Blobel's x_i corresponds to our $[p]_i$. This derivative defines the regularization term from Eq. 4.7. □

Numerical Optimization The original RUN algorithm minimizes $\mathcal{L}^{\mathrm{RUN}}(p\,; q, R) + \tau \cdot r^{\mathrm{RUN}}(p;\, a)$ with the Newton-Raphson method, a standard second-order optimization technique, which is detailed, for instance, by Wright and Nocedal [108]. However, the original RUN algorithm does not employ the vanilla version of this method. Instead, Blobel [98] proposes a customized variant of Newton-Raphson, which allows him to fix the *effective number of degrees of freedom* of the solution, n_{df}. A fixed value of n_{df} also defines the value of τ, the regularization strength in our Def. 4.2. Therefore, a user can specify the impact of the regularization in terms of n_{df}, which can be interpreted as the number of free latent parameters that defines the solution, instead of specifying a less interpretable τ value. The interested reader can refer to our presentation of the original RUN algorithm [1], which covers this aspect in detail.

A considerable disadvantage of fixing the value of n_{df} is that the resulting value of τ must change in every iteration. Therefore, the objective function changes in every iteration, so that the original RUN algorithm cannot employ off-the-shelf optimization tools. Moreover, a fixed value of n_{df} does not prevent the need for cross validation: a practitioner would still have to select the value of n_{df} based on some performance metric, just as he would select a value of τ directly. In this sense, the merit of specifying the regularization impact through n_{df} is limited.

Moreover, the custom Newton-Raphson variant of RUN does not properly constrain its solutions to the feasible set $\mathcal{P}$ from Def. 4.1. Therefore, the original RUN frequently returns solutions with negative components and with sums that are unequal to one. These deficiencies have to be corrected after the optimization, through clipping and normalization. The impact of these operations on the accuracy of the solution is unclear; in particular, it is not guaranteed that the clipped and normalized values indeed minimize the regularized loss function within $\mathcal{P}$.

We correct this deficiency in two steps: first, we do *not* fix the value of n_{df}. Instead, we fix the value of τ, which is the usual practice in machine learning and in numerical optimization. A fixed value of τ allows us to use any off-the-shelf method for numerical optimization. Second, we leverage this freedom of choice in using a constrained optimization technique to minimize $\mathcal{L}^{\mathrm{RUN}}(p\,; q, R) + \tau \cdot r^{\mathrm{RUN}}(p)$ over the feasible set $\mathcal{P}$ from Def. 4.1. In particular, our implementation employs a primal-dual interior-point algorithm with a filter line search [109]. The outcome of our corrections is a variant of RUN which always returns valid probability densities. We compare this improved variant to the original RUN in Sec. 4.5.

4.1.2. Unfolding via Singular Value Decomposition (SVD)

The SVD-based unfolding by Hoecker and Kartvelishvili [99] is designed to be a computationally more effective alternative to the original RUN algorithm. In this regard, SVD employs the same feature transformation and the same regularization term as RUN. The computational effectiveness of SVD stems from a least squares loss

$$\mathcal{L}^{\mathrm{SVD}}(p\,;\,q, \mathrm{M}) \;=\; \left\| \frac{q - \mathrm{M}p}{w} \right\|_2^2 , \tag{4.9}$$

with a weight vector $w \in \mathbb{R}^F$. In principle, this loss can be minimized analytically, i.e., without the need for iterative, numerical optimization techniques.

An analytic solution of Eq. 4.9, however, necessarily disregards the feasible set $\mathcal{P}$ from Def. 4.1, just like the original RUN algorithm does. Therefore, SVD is prone to returning solutions with negative components and with sums that are unequal to one, which have to be improperly corrected through clipping and normalization. In our implementation, we do *not* use the proposed analytic solution but a numerical optimization technique that is properly constrained to $\mathcal{P}$. In our experiments, we compare this variant to the original SVD.

The weight vector in Eq. 4.9 is typically given by the standard deviations $w = \sqrt{\bar{q}}$ of component-wise Poisson distributions. Therefore, the SVD-based approach is similar to RUN in several regards. In particular, SVD and RUN

- employ the same feature transformation
- employ the same regularization term
- assume independence between the components of q
- assume a Poisson distribution for each component of q

The difference between SVD and RUN, however, is that the Poisson rates in SVD are assumed to be $\bar{q}$ instead of $\mathrm{M}\bar{p}$. This difference allows one to replace the likelihood of RUN with the least squares loss of SVD. However, this difference is actually a less appropriate simplification of the RUN likelihood.

Proposition 4.2. The SVD-based unfolding [99] is defined over the loss function from Eq. 4.9 and the regularization term from Eq. 4.7. Through these definitions, SVD qualifies as an instance of our common framework from Def. 4.2.

Proof. The loss function we present in Eq. 4.9 and the regularization term from Eq. 4.7 are verbatim statements by Hoecker and Kartvelishvili [99, Eqs. (29), (37), and (38)]. □

4.1.3. Iterative Bayesian Unfolding (IBU)

The IBU method by D'Agostini [74], [103] follows a probabilistic approach to solving the system of linear equations from Eq. 3.3, $q = \mathrm{M} \cdot p$, for p. This probabilistic approach evolves around an expectation maximization process, which repeatedly applies Bayes' theorem in order to update the solution vector of each previous iteration. IBU is still popular today [110]; recently, physicists have even proposed to apply IBU to the correction of read-out noise in quantum computers [111].

Expectation Maximization The optimization process starts from a user-defined prior $p^{(0)}$, which is typically set to a uniform distribution $[p^{(0)}]_i = \frac{1}{C} \; \forall \; 1 \leq i \leq C$. The expectation (E) step and the maximization (M) step of this process can be written as a single, combined update rule, which revises each previous estimate $p^{(k-1)}$ according to Bayes' theorem,

$$[p^{(k)}]_i \; = \; \sum_{j=1}^{F} \frac{[\mathrm{M}]_{ij} [p^{(k-1)}]_i}{\sum_{i'=1}^{C} [\mathrm{M}]_{i'j} [p^{(k-1)}]_{i'}} [q]_j . \tag{4.10}$$

Here, each previous estimate $p^{(k-1)}$ takes the role of the Bayes prior. Due to the application of Bayes' theorem, we acknowledge that the conditional "direction" of each entry in M, i.e., $[\mathrm{M}]_{ij} = \mathbb{P}(f(X) = j \mid Y = i)$ is appropriately considered. However, the loss function of IBU is only implicit due to the expectation maximization process; we cannot easily define a loss function $\mathcal{L}$ in parametric terms.

Regularization IBU implements regularization toward ordinal solutions in two ad-hoc manners, by smoothing its intermediate estimates and by early stopping.

A smoothing of intermediate estimates means to replace the actual prior $p^{(k-1)}$ of each iteration, see Eq. 4.10, with a transformed, more smooth variant of this prior. In particular, D'Agostini [74] has proposed to fit a low-order polynomial to each $p^{(k-1)}$; due to its low order, the fitted polynomial will exhibit a high degree of smoothness, which the user can control by specifying the order. This polynomial is then used, instead of $p^{(k-1)}$, as the Bayes prior of the current iteration.

In our experience [112], a complete replacement of $p^{(k-1)}$ with a low-order polynomial does *not* improve the quality of the unfolding / quantification result. Therefore, we propose a more general alternative, which employs linear interpolation instead of complete replacements. In particular, we suggest to replace the actual prior $p^{(k-1)}$ with

$$(1 - \lambda) \cdot p^{(k-1)} \; + \; \lambda \cdot p_o, \tag{4.11}$$

where $0 \leq \lambda \leq 1$ is the interpolation parameter and p_o is a polynomial of order $o \in \mathbb{N}$, which is fitted to $p^{(k-1)}$ and evaluated at the indices of $p^{(k-1)}$. This linear interpolation allows us to control the impact of the smoothing. For instance, a complete replacement can be implemented by setting $\lambda = 1$ and smoothing can be entirely de-activated by setting $\lambda = 0$.

Since the energy spectra of cosmic gamma ray sources exhibit their particular smoothness only in acceptance-corrected log-log plots, we have to apply the smoothing to a logarithmic and acceptance-corrected unfolding / quantification result. Hence, we smooth and replace $\log_{10}(a \odot p)$ instead of p, similar to the argument we have detailed in Sec. 4.1.1.

The second technique for regularization, early stopping, requires a smooth prior, like the default uniform prior $[p^{(0)}]_i = \frac{1}{C} \ \forall \, 1 \leq i \leq C$. Stopping before the repeated application of Eq. 4.10 converges will maintain the smoothness of $p^{(0)}$ to some degree. Therefore, early stopping can introduce an inductive bias toward smooth solutions. However, due to the more fine-grained control that linear interpolations with low-order polynomials provide, we omit early stopping from our experiments.

Despite the fact that the expectation maximization process optimizes a loss function that is only implicit, we consider IBU an instance of our common unfolding / quantification framework. We now prove the correctness of our notation.

Proposition 4.3. IBU [74], [103] maximizes an implicit loss function via expectation maximization, as according to the update rule from Eq. 4.10. Moreover, it implements regularization through smoothing. With these properties, IBU qualifies as an instance of our common framework from Def. 4.2.

Proof. D'Agostini [74, Eqs. (3), and (4)] estimates $[p^{(k)}]_i$ as

$$\frac{1}{\epsilon_i} \sum_{j=1}^{n_E} n(E_j) \cdot \frac{P(E_j \mid C_i) \cdot P_0(C_i)}{\sum_{l=1}^{n_C} P(E_j \mid C_l) \cdot P_0(C_l)},$$

where we identify our notation as $F = n_E$, $C = n_C$, $\mathrm{M}_{ij} = P(E_j \mid C_i)$, and $[p^{(k-1)}]_i = P_0(C_i)$. In the original algorithm, $n(E_j) \in \mathbb{N}$ is the count observed in the j-th bin, i.e., $n(E_j) = N \cdot [q]_j$. Moreover, $\epsilon_i > 0$ is an acceptance factor, which models the probability that an existing instance of class i is indeed part of the sample—and not hidden due to measurement complications. Setting $\epsilon_i = N$, we obtain $[q]_j = \frac{n(E_j)}{\epsilon_i}$, which is consistent with our Eq. 4.10.

For regularization, D'Agostini [74] proposes to "smooth the results of the unfolding before feeding them in the next step", for instance "by a polynomial fit of 3$^{\text{rd}}$ degree" or by another low-order polynomial. □

4.1.4. Other Unfolding Methods

Other physics-inspired algorithms are based on similar concepts as RUN, SVD, and IBU. We focus on these methods due to their long-standing popularity within physics research. In fact, they are among the first methods that have been proposed in this field, and are still widely adopted today, in astro-particle physics [106], [107], high-energy physics [110], and more recently in quantum computing [111]. Moreover, they already cover the most important aspects of unfolding methods with respect to ordinal quantification.

We have already seen that SVD employs a simplification of the Poisson rates that RUN uses. Two other methods, MRX [113] and TUnfold [114] employ the same simplification as SVD, but regularize in different ways. To this end, MRX regularizes with respect to the deviation from a prior, instead of regularizing with respect to ordinal plausibility; we thus do not perceive this method as a true method for ordinal quantification. TUnfold adds to the RUN regularization a second term, which penalizes estimates that do not sum up to one. In our implementation of RUN, there is no need for this penalty because the $\mathcal{P}$ constraint already guarantees that all estimates sum up to one.

Another line of work evolves around DSEA [115] and its extension DSEA$^+$ [1]. We perceive this algorithm to lie outside the scope of ordinal quantification because it does not address the order of classes, like the other unfolding methods do. Moreover, the algorithm was shown to exhibit a performance comparable to, but not better than, RUN and IBU [1].

4.1.5. Adjusted Classify and Count (ACC)

We now leave the realm of "unfolding" methods, which stem from physics literature, and turn to methods that go under the name of *quantification*. One of the most widely acknowledged methods for quantification is ACC [93], a method that was initially proposed for *binary* quantification in particular. For the multi-class setting, there exist four different extensions [3], which we detail in the following.

The original, binary variant of ACC adjusts the fraction $q \in \mathbb{R}$ of positive predictions with the true positive and false positive rates of the classifier $h : \mathcal{X} \to \mathbb{R}$, i.e.,

$$p = \frac{q - \text{FPR}}{\text{TPR} - \text{FPR}} \quad \text{where} \quad q = \frac{1}{N} \sum_{k=1}^{N} \delta_{h(x_k)>0} \,. \tag{4.12}$$

This adjustment has desirable properties. Most importantly, p is a Fisher consistent estimator of $\mathbb{P}(Y = +1)$ even under prior probability shift [75]. Moreover, the method is computationally efficient: a prediction requires only a single pass over the data sample to compute q; so does the computation of FPR and TPR during training.

Hence, multi-class extensions to the binary ACC are a promising topic for quantification research. In this regard, we are aware of four extensions to the binary variant of ACC: one-versus-all decomposition [93], matrix inversion [116], [117], pseudo-inversion [118], and constrained least squares [97], [2], [100]. We argue that the constrained least squares variant is the most appropriate extension because it explicitly constrains its solutions to the feasible set $\mathcal{P}$ from Def. 4.1. Therefore, we select this variant for being included in our common unfolding / quantification framework. For completeness, we also detail the other multi-class variants of ACC. We delay a comparison of their performances to Sec. 4.2, where we develop yet another, more effective multi-class extension.

Table 4.2.: Adjustments in multi-class ACC extensions. Our proposal, the soft-max adjustment, is detailed later, in Sec. 4.2. This table is adapted from one of our publications [3].

adjustment	premise	loss function	constraints	optimization
one-vs-rest (Eq. 4.14)	TPR_i, FPR_i	–	–	–
inverse (Eq. 4.16)	M	–	–	–
pseudo-inverse (Eq. 4.19)	M	least squares	min. norm	–
constrained (Eq. 4.20)	M	least squares	$\mathcal{P}$	constrained
soft-max (ours; Eq. 4.28)	M	least squares	$\mathcal{P}$	unconstrained

A summary of the conceptual properties of the four existing multi-class variants, and our "soft-max" variant from Sec. 4.2, is displayed in Tab. 4.2. Their common ground is an adjustment of the fractions of predictions,

$$[q]_i \;=\; \frac{1}{N}\sum_{k=1}^{N} \delta_{\arg\max_i [h(x_k)]_i} \quad \forall\; 1 \leq i \leq C, \tag{4.13}$$

through performance estimates of the classifier, e.g., $\mathrm{TPR}_i \in \mathbb{R}$, $\mathrm{FPR}_i \in \mathbb{R}$, or $\mathrm{M} \in \mathbb{R}^{C \times C}$. Note that Eq. 4.13 is in line with our general definition of q, see Eq. 4.1, if the feature transformation $f(x) = \arg\max_i [h(x)]_i$ is employed.

One-Versus-Rest Decomposition

The most straightforward extension of binary ACC decomposes the multi-class quantification problem into C one-versus-rest tasks [93]. Each of these tasks requires a binary quantification of one class versus all others. Hence, we can use the binary adjustment rule from Eq. 4.12 in each of the tasks separately. The resulting estimate is a vector with components

$$[p^{(\mathrm{one-vs-rest})}]_i \;=\; \frac{[q]_i - \mathrm{FPR}_i}{\mathrm{TPR}_i - \mathrm{FPR}_i} \tag{4.14}$$

where TPR_i and FPR_i are the true positive rate and the false positive rate of the classifier h when class i is classified against all other classes.

Like binary ACC, the estimate from Eq. 4.14 requires clipping and normalization to ensure that the solution is a valid probability density. Unfortunately, these ad-hoc corrections can lead to estimation errors if the data sets are not sufficiently large to accurately estimate q, TPR_i, and FPR_i.

Matrix Inversion

The one-versus-rest adjustment ignores the fact that mis-classifications can occur between any pair of classes. It is therefore more appropriate to account not only for the class-wise TPR_i and FPR_i but for the full confusion matrix $\text{M} \in \mathbb{R}^{C \times C}$ of the classifier h. This matrix comprises all mis-classification probabilities

$$[\text{M}]_{ij} \;=\; \mathbb{P}\big(\arg\max_k\, [h(X)]_k \;=\; i \mid Y = j\big) \tag{4.15}$$

and it is well-acknowledged [97], [116], [117] that the prediction outcome q is determined by the system of linear equations, $q = \text{M} \cdot p$, from Eq. 3.3 if $p \in \mathcal{P}$ is the true class distribution.

Consequently, we can recover an estimate of the true p with an estimate of the true confusion matrix M. The most straightforward attempt [116], [117] in this direction is to invert M to yield the estimate

$$p^{(\text{inverse})} \;=\; \text{M}^{-1} \cdot q, \tag{4.16}$$

which has to be clipped and normalized to represent a valid probability density.

For instance, this matrix inversion estimate is implemented in the current release[7] of QuaPy [119]. Since QuaPy is likely the most complete and usable software package for quantification, this choice has established $p^{(\text{inverse})}$ as the “quasi-standard” multi-class extension of ACC and PACC.

However, the inverse of an estimated matrix M is not guaranteed to exist. Therefore, the estimator is sometimes undefined. QuaPy deals with this issue by falling back to returning the un-adjusted q if M is not invertible.

Pseudo-Inversion

A robust alternative to matrix inversion is to replace the actual inverse M^{-1} with the Moore-Penrose pseudo-inverse

$$\text{M}^{\dagger} \;=\; \text{VS}^{\dagger}\text{U}^{\top}, \tag{4.17}$$

where the matrices V, S, and U are defined through the singular value decomposition

$$\begin{aligned} \text{M} \;&=\; \text{USV}^{\top}, \\ \text{S} \;&=\; \text{diag}(s_1, \dots, s_k, 0, \dots, 0), \\ \text{S}^{\dagger} \;&=\; \text{diag}(s_1^{-1}, \dots, s_k^{-1}, 0, \dots, 0), \end{aligned} \tag{4.18}$$

and $s_1, \dots, s_k$ are the non-zero singular values of M in decreasing order. Replacing the true inverse M^{-1} with the pseudo-inverse $\text{M}^{\dagger}$ leads to the quantification estimate

$$p^{(\text{pseudo}-\text{inverse})} \;=\; \text{M}^{\dagger} \cdot q, \tag{4.19}$$

which is always defined because $\text{M}^{\dagger}$ is always guaranteed to exist. Fortunately, $\text{M}^{\dagger}$ is equal to M^{-1} if M^{-1} exists. Therefore, the replacement does not reduce the quality of

[7] `https://github.com/HLT-ISTI/QuaPy/releases/tag/0.1.6`, release v0.1.6 of QuaPy [119]

the estimate but it gains robustness because no fallback to a completely un-adjusted q is necessary if M is not invertible.

The pseudo-inverse estimator is proven to be a least squares estimate of the true p, which is constrained to a minimum norm solution [118, Th. 4.1]. This constraint has the advantage that $p^{(\text{pseudo}-\text{inverse})}$ is unique while an unconstrained least squares estimate, in general, is not. However, a minimum norm constraint lacks motivation from a practical perspective; in fact, a minimum norm is unrelated to the actual feasible set $\mathcal{P}$ from Def. 4.1. Like in the above variants, clipping and normalization are therefore necessary for returning a valid probability density.

Constrained Least Squares

Both matrix inversion techniques, $p^{(\text{inverse})}$ and $p^{(\text{pseudo}-\text{inverse})}$, suffer from not being constrained to the feasible set $\mathcal{P}$ from Def. 4.1. In fact, both techniques tend to produce estimates that i) do not sum to one and ii) have components that are below zero. This deficiency is typically addressed through clipping and normalization, an ad-hoc correction that can easily lead to estimation errors.

Loss Function A more appropriate approach to multi-class ACC, as presented by Hopkins and King [97], is implemented as a constrained optimization task

$$p^{(\text{constrained})} = \operatorname*{arg\,min}_{p \in \mathcal{P}} \left\| q - \mathrm{M} \cdot p \right\|_2^2, \tag{4.20}$$

which explicitly constrains the estimate to the feasible set $\mathcal{P}$ from Eq. 4.1. Within this set, the most accurate estimate according to the squared L_2 norm is searched for. Accordingly, $p^{(\text{constrained})}$ employs the same loss function as $p^{(\text{pseudo}-\text{inverse})}$, but appropriately constrains its solution to $\mathcal{P}$ instead of minimum norm solutions.

Note that Hopkins and King have developed their method independently of Forman's binary ACC. However, the basis of their work is precisely the binary ACC adjustment from Eq. 4.12, as can be seen in Hopkins and King [97, Eq. (3)]. Therefore, we understand their method as a multi-class extension to ACC. Due to the specific $\mathcal{P}$ constraint in Eq. 4.20, we select $p^{(\text{constrained})}$ as the most appropriate multi-class extension to ACC, which we include in our common unfolding / quantification framework.

Proposition 4.4. The constrained least squares extension to binary ACC [93], as proposed by Hopkins and King [97], is defined over the loss function from Eq. 4.20. Therefore, this extension qualifies as an instance of our common framework from Def. 4.2. The extension does not employ regularization.

Proof. Hopkins and King [97, Eq. (4)] marginalize over the true labels $D \in \{1, \dots, J\}$ to yield the probability density of class predictions $\widehat{D}$ as

$$P(\widehat{D} = j) \; = \; \sum_{j'=1}^{J} P(\widehat{D} = j \mid D = j')P(D = j).$$

The authors note that "this expression represents a set of J equations [...] that can be solved for the J elements in $P(D)$". Accordingly, we identify our notation as $p = P(D)$, $q = P(\widehat{D})$, and $\mathrm{M} = P(\widehat{D} \mid D)$ in their presentation. To solve this set of equations, the authors propose a "standard constrained least squares to ensure that elements of $P(D)$ are each in [0,1] and collectively sum up to 1". This proposal defines the least squares loss from Eq. 4.20 and matches our constraints with respect to $\mathcal{P}$. □

Numerical Optimization Unfortunately, Hopkins and King [97] do not propose a specific algorithm to solve Eq. 4.20. While an analytic solution exists for the unit sum constraint, $1 = \sum_{i=1}^{C} [p]_i$ [120, Chap. 1.4], we are not aware of an analytic solution that considers the inequality constraints, $[p]_i \geq 0 \;\; \forall \;\; 1 \leq i \leq C$, from Def. 4.1.

Consequently, the optimization of Eq. 4.20 requires iterative, numerical optimization techniques. In our implementation, we employ a primal-dual interior-point algorithm with a filter line search [109]. However, other numerical methods are conceivable at this point. For instance, Firat [100] employs a sequential quadratic programming technique [108, Chap. 18] to solve Eq. 4.20. We have to leave a comparison of numerical optimization techniques in quantification to future work.

4.1.6. Probabilistic ACC (PACC)

The essential proposal of binary PACC [104] is to replace the crisp predictions $\delta_{h(x)>0}$, which are employed in ACC, with probabilistic classification outcomes $h(x) \in [0, 1]$. Despite this replacement, binary PACC employs exactly the adjustment rule of binary ACC, which we have introduced in Eq. 4.12.

Since PACC differs from ACC only in this single aspect, we can extend the binary PACC to the multi-class setting in the same variety of multi-class extensions that we have just discussed for binary ACC: one-versus-rest decomposition, matrix inversion, pseudo-inversion, and the constrained least squares estimate. All of these extensions have in common the adjustment of predictions in terms of performance estimates of the classifier. Unlike ACC, these predictions are now given in probabilistic terms, i.e.,

$$[q]_i \; = \; \frac{1}{N} \sum_{k=1}^{N} [h(x_k)]_i \quad \forall \;\; 1 \leq i \leq C, \tag{4.21}$$

where $h : \mathcal{X} \rightarrow \mathcal{P}$ is a probabilistic multi-class classifier.

Like in multi-class ACC, we opt for the constrained least squares estimate as the most appropriate multi-class extension to binary PACC. Therefore, the same loss function is employed.

Proposition 4.5. The constrained least squares extension to binary PACC [104], as proposed by Hopkins and King [97], is defined over the loss function from Eq. 4.20. Therefore, this extension qualifies as an instance of our common framework from Def. 4.2. The extension does not employ regularization.

Proof. The essential proposal by Bella et al. [104] is to replace the crisp classification outcomes $\delta_{h(x)>0}$ with probabilistic ones $h(x) \in [0, 1]$; their adjustment is the same as in binary ACC. Taking this proposal to multi-class ACC [97], we obtain multi-class PACC with the loss from Eq. 4.20. □

4.1.7. ReadMe

Building on the multi-class version of ACC, ReadMe [97] employs the loss function from Eq. 4.20. However, ReadMe transforms the features in a unique way that is motivated in text mining. In this application area, data items x are often represented as bags of words, i.e., as sparse indicator vectors $\{0, 1\}^d$ for a vocabulary of size d. In ReadMe, q is a histogram over all 2^d possible incarnations X_i of these indicator vectors, i.e.

$$f^{(\text{ReadMe})}(x) \;=\; \delta_{x=(X_1,\dots,X_{2^d})}, \;\; \text{where} \;\; [\delta_{a=(a_1,\dots,a_n)}]_i \;=\; \begin{cases} 1 & \text{if } a = a_i, \\ 0 & \text{otherwise} \end{cases}. \tag{4.22}$$

Since such a representation is only feasible with small d, ReadMe produces multiple estimates, each of which employs a different and small, random selection of words. Finally, all of these individual estimates are averaged.

Proposition 4.6. ReadMe [97] is defined over the loss function from Eq. 4.20. Therefore, it qualifies as an instance of our common framework from Def. 4.2. ReadMe does not employ regularization.

Proof. Building on their multi-class design of ACC, Hopkins and King [97, Eq. (6)] set up a matrix equation $P(\mathbf{S}) = P(\mathbf{S} \mid D)P(D)$, which maps to our notation as $q = P(\mathbf{S}) \in \mathbb{R}^{2^d}$, $\mathrm{M} = P(\mathbf{S} \mid D) \in \mathbb{R}^{2^d \times C}$, and $p = P(D) \in \mathbb{R}^C$. The authors note that "$P(\mathbf{S})$ is the probability of each of the 2^K possible word stem profiles" with $K = d$ being the number of word stems. To estimate this probability, "we merely compute the proportion of documents observed with each pattern of word profiles". This computation leads to a histogram

$$q = \frac{1}{N} \sum_{i=1}^{N} \delta_{x_i=(X_1,\dots,X_{2^d})},$$

which is consistent with our Eqs. 4.1 and 4.22. □

In astro-particle physics, we are not aware of any suitable bag-of-words representation of the data. Therefore, we omit ReadMe from our experiments.

4.1.8. HDx

HDx [105] builds on the conception that a least-squares loss is not a well-motivated measure for divergences between probability densities; in fact, other measures are more appropriate for this purpose. One such appropriate measure is the Hellinger distance (HD) from Def. 2.6, which inspires the name of HDx.

In HDx, each feature is separately binned. A data sample is then represented as a concatenation of all feature-wise histograms,

$$f(x) \;=\; (\delta_{b(x;1)}, \ldots, \delta_{b(x;d)}), \tag{4.23}$$

where $b(x;i)$ is a binning function, which maps the feature value $[x]_i$ to the corresponding bin index $\{1, \ldots, B_i\}$.

The loss of a candidate solution p is measured as the average of all feature-wise Hellinger distances,

$$\begin{aligned} \mathcal{L}(p\,;\,\mathrm{M}, q) \;&=\; \frac{1}{d}\sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M}p), \\ \text{where}\quad \mathrm{HD}_i(q,\, \mathrm{M}p) \;&=\; \sqrt{\sum_{j=1+\sum_{k=1}^{i-1} B_k}^{\sum_{k=1}^{i} B_k} \left(\sqrt{[q]_j} - \sqrt{[\mathrm{M}p]_j}\right)^2}\,. \end{aligned} \tag{4.24}$$

Here, we let the bin indices range from $1 + \sum_{k=1}^{i-1} B_k$ to $\sum_{k=1}^{i} B_k$. This choice is made in order to define a Hellinger distance HD_i which only considers the bins of the i-th feature.

Proposition 4.7. HDx [105] is defined over the loss function from Eq. 4.24. Therefore, it qualifies as an instance of our common framework from Def. 4.2. HDx does not employ regularization.

Proof. González-Castro et al. [105, Eq. (9)] minimize the average of feature-wise Hellinger distances, as we have stated in Eq. 4.24. They present the distance with respect to a single feature j, binned into b bins, as

$$\sqrt{\sum_{i=1}^{b} \left(\sqrt{\frac{|V_{j,i}|}{|V|}} - \sqrt{\frac{|U_{j,i}|}{|U|}}\right)^2},$$

where $|U|$ is the total number of instances and $|U_{j,i}|$ is the number of instances whose feature j is mapped to the i-th bin [105, Eq. (10)]. $|V|$ and $|V_{j,i}|$ are the numbers of instances that are to be expected under class prevalences p, hence

$$\frac{|V_{j,i}|}{|V|} = [\mathrm{M}p]_{i+\sum_{k=1}^{j-1} B_k},$$

where $\sum_{k=1}^{j-1} B_k$ is the offset of the histogram of feature j within our concatenation of feature-wise histograms. Using the multi-class product $\mathrm{M}p$ at this point is consistent with the binary conception of González-Castro et al. [105, Eq. (12)]. □

The original proposal by González-Castro et al. [105] is to find the minimum of Eq. 4.24 through an extensive grid search. In our implementation, we prefer to stick to constrained, numerical optimization process in terms of a primal-dual interior-point algorithm with a filter line search [109].

4.1.9. HDy

HDy [105] minimizes Hellinger distances, just like HDx. However, the Hellinger distances are now measured in terms of classification outcomes instead of feature binnings.

Originally, HDy was proposed for binary quantification in particular. However, we can easily extend the method to the multi-class setting, borrowing the treatment of multiple features that HDx employs. In particular, a multi-class HDy can be realized by separately binning the class-wise classification outcomes, just like features are separately binned in HDx. Hence, this multi-class extension employs the loss function from Eq. 4.24 but replaces the feature binning $b(x; i)$ in Eq. 4.23 with $b(h(x), i)$.

Proposition 4.8. Our multi-class extension of HDy [105] employs the loss function from Eq. 4.24. Therefore, it qualifies as an instance of our common framework from Def. 4.2. HDy does not employ regularization.

Proof. The original HDy [105, Eqs. (13) and (14)] only addresses binary quantification. For this case, however, the only change with respect to HDx is that HDy employs soft classifier outputs $h(x)$ instead of features x. A straightforward extension to the multi-class setting is therefore to bin the class-wise outputs $[h(x)]_i$ separately, as HDx does in case of features and as we propose in our multi-class extension of HDy. □

4.1.10. Un-Adjusted Variants of Classify and Count

We also conceive the un-adjusted methods Classify and Count (CC) [93] and Probabilistic Classify and Count (PCC) [104] as instances of our framework. Instead of adjusting the counts q of predictions in terms of the mis-prediction probabilities that are expressed in the confusion matrix M, these methods simply return q as their estimate for p.

Therefore, strictly speaking, CC and PCC do not require any minimization of a loss function. More loosely speaking, however, their disregard of M can be understood as the assumption of a perfect classifier, which is expressed by $\mathrm{M} = \mathbb{I}$ being the identity matrix. Under this assumption, the least squares loss from Eq. 4.20 actually leads to the estimate $p^{(\mathrm{CC})} = q$ and we can understand this estimate as an instance of our common unfolding / quantification framework.

Regarding the representation $f(x)$, CC employs the feature transformation of ACC, i.e., $\delta_{\arg\max_i [h(x)]_i}$, and PCC employs the feature transformation of PACC, i.e., $h(x)$.

Proposition 4.9. Assume a perfect classifier and, hence, $\mathrm{M} = \mathbb{I}$. Under this assumption, we can claim that the un-adjusted methods CC [93] and PCC [104] employ the loss function from Eq. 4.20. In this limited sense, they qualify as instances of our common framework from Def. 4.2. CC and PCC do not employ regularization.

Proof. Let $\mathrm{M} = \mathbb{I}$. Recognize that the global minimum of the least squares loss,

$$\min_p \|q - \mathrm{M}p\|_2^2 \; = \; 0,$$

is now attained if and only if $p = q$. Therefore, under the assumption $\mathrm{M} = \mathbb{I}$, the unique minimizer of the least squares loss is q. In this sense, PCC and CC are proper instances of our framework. □

4.2. Solving Quantification Through Unconstrained Optimization

Our common framework for unfolding and quantification algorithms, see Def. 4.2, minimizes an objective function, $\mathcal{L}(p\,;\,q, \mathrm{M}) + \tau \cdot r(p)$, over the feasible set $\mathcal{P}$ from Def. 4.1. In the previous section, we have seen that many algorithms from unfolding and quantification indeed adhere to this framework.

Constrained, numerical optimization techniques provide us with the opportunity of implementing this constrained optimization task precisely in the way it is formalized. These techniques ensure, through dedicated efforts, that every intermediate solution candidate p is in the feasible set $\mathcal{P}$. Thereby, they indeed find a solution that minimizes the objective function over $\mathcal{P}$. However, their dedicated efforts for constraining the solution candidates can come at a computational price.

Therefore, we propose an alternative optimization task for implementing any unfolding / quantification algorithm from our framework. This alternative task is *un-constrained*, and thus can lead to more effective solutions. We enable un-constrained optimization through a soft-max "trick" [3], which embeds the constraints as a part of the unfolding / quantification model. Thereby, this trick ensures that all solutions are valid solutions that lie in $\mathcal{P}$, without requiring a numerical optimization technique to ensure this constraint.

In the following, we detail the method that stems from employing our soft-max trick in unfolding / quantification. Then, we evaluate our proposal in the scope of the multi-class ACC variants from Sec. 4.1.5.

4.2.1. The Soft-Max Trick

Our alternative optimization task does no longer consider the components of p to be the free parameters over which the objective function is minimized. Instead, we minimize over *latent* parameters $l \in \mathbb{R}^C$, which we transform to valid probability densities through a soft-max layer $\text{softmax}(l) \in \mathcal{P}$, where

$$[\text{softmax}(l)]_i \;=\; \frac{\exp([l]_i)}{\sum_{j=1}^{C} \exp([l]_j)}. \tag{4.25}$$

Since $\text{softmax}(l)$ is always in $\mathcal{P}$, we use this transformation i) as our estimate of p, ii) as the input of the loss function, and iii) as the input of the regularization term [3]. Thereby, we obtain the estimate

$$p^* \;=\; \text{softmax}(l^*) \tag{4.26}$$

through the alternative, un-constrained optimization task

$$l^* \;=\; \underset{l \in \mathbb{R}^C}{\arg\min}\; \mathcal{L}\big(\text{softmax}(l)\,;\, q, \mathrm{M}\big) \;+\; \tau \cdot r\big(\text{softmax}(l)\big) \;+\; \lambda \cdot \|l\,\|_2^2, \tag{4.27}$$

where $\lambda \cdot \|l\,\|_2^2$ is an additional regularization term. This term, however, is only a technical detail that ensures all $\exp([l]_i)$ to be finite within floating point precision. In our experiments, we fix $\lambda = 10^{-6}$, a value that is meant to not influence the final estimate p^*.

We can re-formulate any algorithm from our unified framework in terms of the unconstrained optimization task from Eq. 4.27. As two concrete examples, we instantiate the soft-max version of ACC and PACC,

$$l^{(\text{ACC})} \;=\; \underset{l \in \mathbb{R}^C}{\arg\min}\, \big\|\, q - \mathrm{M} \cdot \text{softmax}(l)\, \big\|_2^2 \;+\; \lambda \cdot \big\|l\,\big\|_2^2\,, \tag{4.28}$$

and the soft-max version of RUN,

$$\begin{aligned} l^{(\text{RUN})} \;=\; \underset{l \in \mathbb{R}^C}{\arg\min} & \sum_{i=1}^{F} \,[\mathrm{M} \cdot N \cdot \text{softmax}(l)]_i - \bar{q}_i \ln[\mathrm{M} \cdot N \cdot \text{softmax}(l)]_i \\ & + \frac{\tau}{2}\big(\mathrm{T} \cdot \log_{10}(a \odot \text{softmax}(l))\big)^2 \\ & + \lambda \cdot \big\|l\,\big\|_2^2\,, \end{aligned} \tag{4.29}$$

simply by replacing p in Eqs. 4.20, 4.6, and 4.8 with $\text{softmax}(l)$. In earlier work [3], we have taken out this replacement only in the scope of multi-class ACC.

Table 4.3.: Test set performance of the different multi-class adjustments, for ACC and PACC and in terms of AE and RAE. The performance of the best adjustment in each setting is printed in boldface. This table is adapted from one of our publications [3].

adjustment	ACC		PACC	
	AE	RAE	AE	RAE
un-adjusted (Eq. 4.13 / Eq. 4.21)	0.0254	2.5532	0.0246	2.6771
one-vs-rest (Eq. 4.14)	0.0262	4.1484	0.0262	4.1484
inverse (Eq. 4.16)	0.0222	1.7224	0.0195	1.5288
pseudo-inverse (Eq. 4.19)	0.0177	1.7224	0.0195	1.5288
constrained (Eq. 4.20)	0.0158	1.2826	0.0123	**0.9908**
soft-max (Eq. 4.28)	**0.0130**	**1.2633**	**0.0106**	1.0886

4.2.2. Empirical Validation

In the following, we intend to uncover the merits of our soft-max trick in the scope of multi-class ACC and PACC. To this end, we evaluate the performance of each method on the public data set [121] of the LeQua2022 competition [122].

The LeQua2022 dataset is designed to constitute a gold-standard benchmark, both for binary text quantification and for multi-class text quantification. The multi-class problem in this competition features 28 classes, 20 000 training items and 1 000 validation samples. Each of the validation samples consists of 1 000 data items that are drawn according to varying class prevalences. The ground-truth class prevalence vectors, with which quantification performances are measured, are generated with the artificial prevalence protocol we introduce in Sec. 4.5.1. We employ the vectorial representation of the data and a logistic regression classifier, which obtained the highest performance on this representation during the competition [4]. We optimize the regularization parameter of this classifier on the validation set and over the grid $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$, to obtain the best performance for each quantification method. The selection of the best regularization parameter is either in terms of the average absolute error (AE) or in terms of the average relative absolute error (RAE). After selecting the best hyper-parameters for each method, we report the results, in terms of both metrics, on the test set.

All multi-class extensions of ACC require the estimation of the confusion matrix M (or at least the rates TPR_i and FPR_i) on hold-out data. In order to use all labeled data for classifier training and for the adjustments, we use a bagging ensemble of size 100. We estimate M, TPR_i, and FPR_i on the out-of-bag predictions of this ensemble.

During the hyper-parameter optimization on the validation set, almost all methods succeeded in producing estimates for the class prevalences. An exception to this outcome is the matrix inversion from Eq. 4.16 in ACC. This method failed to produce prevalence estimates for the values 10^{-3} and 10^{-2} of the classifier's regularization parameter because these values led to confusion matrices M that were *not* invertible.

Discussion The results from Tab. 4.3 demonstrate that the different multi-class adjustments exhibit quite different performances. The lowest errors are achieved by the constrained estimator from Eq. 4.20 (in terms of RAE in PACC) and by our unconstrained soft-max estimator from Eq. 4.28 (in terms of all other configurations). The margins of improvement over all other adjustments are considerable: for instance, the constrained PACC achieves an RAE that is 35% smaller than the RAE of the pseudo-inverse PACC (last column, 0.9908 vs 1.5288); our soft-max PACC achieves an AE that is 46% smaller than the AE of the pseudo-inverse PACC (third column, 0.0106 vs 0.0195).

4.3. Addressing Ordinality Through Regularization

In the following we develop algorithms which extend ACC, PACC, HDx, HDy, and the Saerens-Latinne-Decaestecker (SLD) method [123] with the regularizers from RUN and IBU. This extension yields o-ACC, o-PACC, o-HDx, o-HDy, and o-SLD [5], the OQ counterparts of these well-known, non-ordinal quantification algorithms. Our ordinal extensions demonstrate the practical value of our common framework from Def. 4.2: having revealed that the existing methods differ from each other mainly in terms of their loss functions, their regularization terms, and their feature transformations, we are ready to re-combine these characteristics in order to develop novel quantification methods.

Our ordinal extensions, which employ the existing regularizers from RUN and IBU, preserve the general characteristics of the original, non-ordinal ACC, PACC, HDx, HDy, and SLD. In particular, our extensions do not alter the loss functions and feature transformations of these original methods. Therefore, our extensions are "minimal", in the sense that they directly address ordinality without introducing any other, undesired side-effects into the original, well-known quantification methods.

In the following, we detail our extensions. We delay an empirical validation of our proposals to Sec. 4.5.

4.3.1. o-ACC and o-PACC

Our ordinal extensions of ACC and PACC employ the same least squares loss that ACC and PACC employ. To address ordinality, we include the regularization term from RUN and SVD, which is defined in Eq. 4.8. Through this re-combination, we obtain the constrained optimization task

$$p^{(\text{o-ACC / o-PACC})} = \operatorname*{arg\,min}_{p \in \mathcal{P}} \; \| \, q - \mathrm{M} \cdot p \, \|_2^2 + \frac{1}{2} \big(\mathrm{T} \cdot \log_{10}(a \odot p) \big)^2 , \tag{4.30}$$

where q is defined either by Eq. 4.13 in case of o-ACC or by Eq. 4.21 in case of o-PACC.

To implement this constrained optimization task effectively, we can re-use the soft-max trick from Sec. 4.2. This trick yields an un-constrained optimization task, through which we find the latent parameters

$$\begin{aligned} l^* \;=\; \underset{l \in \mathbb{R}^C}{\arg\min} \; & \left\| q - \mathrm{M} \cdot \mathrm{softmax}(l) \right\|_2^2 \\ & + \frac{\tau}{2} \big(\mathrm{T} \cdot \log_{10}(a \odot \mathrm{softmax}(l)) \big)^2 \\ & + \lambda \cdot \left\| l \right\|_2^2 , \end{aligned} \tag{4.31}$$

where, again, q is defined either by Eq. 4.13 or by Eq. 4.21 and we set $\lambda = 10^{-6}$. According to Eq. 4.26, we recover the estimate $p^{(\text{o-ACC / o-PACC})} = \mathrm{softmax}(l^*)$ from the latent parameter vector l^*.

4.3.2. o-HDx and o-HDy

Our ordinal extensions of HDx and HDy employ the same loss that HDx and HDy employ. Like in o-ACC and o-PACC, we address ordinality by including the regularization term from RUN and SVD, which is defined in Eq. 4.8. Through this re-combination, we obtain the constrained optimization task

$$p^{(\text{o-HDx / o-HDy})} \;=\; \underset{p \in \mathcal{P}}{\arg\min} \; \frac{1}{d} \sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M}p) \;+\; \frac{1}{2} \big(\mathrm{T} \cdot \log_{10}(a \odot p) \big)^2 \tag{4.32}$$

where HD_i is defined by Eq. 4.24 and q is defined either by a feature binning $b(x; i)$ in case of o-HDx or by a classifier output binning $b(h(x); i))$ in case of o-HDy.

Like in o-ACC and o-PACC, we implement the above task with the soft-max trick from Sec. 4.2. This trick yields an un-constrained optimization task

$$\begin{aligned} l^* \;=\; \underset{l \in \mathbb{R}^C}{\arg\min} \; & \frac{1}{d} \sum_{i=1}^{d} \mathrm{HD}_i(q,\, \mathrm{M} \cdot \mathrm{softmax}(l)) \\ & + \frac{\tau}{2} \big(\mathrm{T} \cdot \log_{10}(a \odot \mathrm{softmax}(l)) \big)^2 \\ & + \lambda \cdot \left\| l \right\|_2^2 , \end{aligned} \tag{4.33}$$

where, again, we recover the estimate $p^{(\text{o-HDx / o-HDy})} = \mathrm{softmax}(l^*)$ from the latent parameter vector l^*.

4.3.3. o-SLD

We now develop the OQ counterpart of SLD [123], a well-known, non-ordinal quantification method. This counterpart solves the quantification task in the same way as the original SLD, but introduces the regularization technique of IBU. Since SLD does not compute a sample average q, we have not included this method in our common unfolding /

quantification framework. Therefore, we start by presenting the original SLD before we develop our ordinal extension thereof.

Original SLD The algorithm by Saerens, Latinne, and Decaestecker [123], also known as EM-based quantification (EMQ), follows an expectation maximization approach. This approach is similar, although not equal, to the approach of IBU, which is introduced in Sec. 4.1.3. In particular, SLD and IBU leverage Bayes' theorem in the E-step and update their estimates in the M-step, see Eq. 4.10. However, SLD does not represent the sample in terms of a histogram $q = \frac{1}{N}\sum_{i=1}^{N} \delta_{f(x_i)}$, see Eq. 4.1, like IBU does. Instead, SLD maintains all individual predictions $h(x)$ of the classifier.

We combine the E-step and the M-step of SLD in a single update rule

$$[p^{(k)}]_i \; = \; \frac{1}{N} \sum_{j=1}^{N} \frac{\frac{[h(x_j)]_i}{[p^{(0)}]_i} \cdot [p^{(k-1)}]_i}{\sum_{i'=1}^{C} \frac{[h(x_j)]_{i'}}{[p^{(0)}]_{i'}} \cdot [p^{(k-1)}]_{i'}} \, , \tag{4.34}$$

which is applied until the estimates converge. The prior $p^{(0)}$ is typically set to the class distribution of the training set. This default prior is another difference from IBU, which, instead, uses a uniform prior by default.

Regularization Towards Ordinal Solutions o-SLD addresses ordinality through the regularization technique of IBU, a smoothing of intermediate estimates. In particular, o-SLD does not use the latest estimate $p^{(k-1)}$ as the prior of the next iteration, but the linear interpolation $(1 - \lambda) \cdot p^{(k-1)} \, + \, \lambda \cdot p_o$ from Eq. 4.11. Here, $0 \, \leq \, \lambda \, \leq \, 1$ is again the interpolation parameter and p_o is again a polynomial of order $o \in \mathbb{N}$, which is fitted to $p^{(k-1)}$ and evaluated at the indices of $p^{(k-1)}$. Like in IBU, we consider λ and o as hyper-parameters, through which the strength of the regularization is controlled.

4.3.4. Related Work: Ordinal Quantification Without Regularization

Not all methods for ordinal quantification address ordinality through regularization. We briefly describe these methods in the following.

Ordinal Quantification Tree The OQT method by Da San Martino et al. [124] creates a binary tree of classes, which quantifies each split in terms of binary PCC. The induction of this tree is informed about the order of classes, which is maintained in the binary splits of classes, and splits are chosen based on the KL divergence of the resulting binary quantification outcomes. Thereby, OQT is specifically aimed at ordinal quantification, but does not introduce an ordinal regularization.

Before our study on ordinal quantification [5], OQT was only evaluated in the ordinal quantification task of the SemEval 2016 competition on sentiment analysis in Twitter [125]. While OQT was the best performer in this task, we must note that the evaluation featured only a single testing sample, which is insufficient for a reliable performance assessment. Since we found OQT to perform inferior in our previous experiments [5], which we revisit in 4.5.3, we exclude OQT from our main evaluation in Sec. 4.5.

Adjusted Regress And Count The ARC method by Esuli [126] creates a binary tree of classes, similar to OQT. However, ARC chooses each split such that both sides are maximally balanced. Moreover, ARC quantifies each split in terms of binary ACC.

Before our study, ARC was only evaluated in the SemEval 2016 competition [125]. Like OQT, the method then performed inferior in our own, previous experiments [5]. Therefore, we exclude ARC from our main evaluation in Sec. 4.5.

4.4. Spotting the Difficulty of Quantification

We investigate more deeply the pseudo-inverse estimator $\mathrm{M}^\dagger q$ from Eq. 4.19. This estimator effectively transforms the linear system of equations $q = \mathrm{M}p$, which we intend to solve for p, to the surrogate problem

$$q' = \mathrm{S}p', \tag{4.35}$$

where $q' = \mathrm{U}^\top q$ and $p' = \mathrm{V}^\top p$. Here, the matrices S, U, and V are defined through the singular value decomposition from Eq. 4.18. Note that the surrogate problem is solved by a simple, element-wise multiplication $p' = \mathrm{S}^\dagger q'$ and that we can recover a solution to the original problem as $p = (\mathrm{V}^\top)^{-1} p'$.

Now imagine the last singular values of M to be extraordinarily small, as compared to the first singular values. In this case, the inverse of the last singular values, which is collected in $\mathrm{S}^\dagger = \mathrm{diag}(s_1^{-1}, \dots, s_k^{-1}, 0, \dots, 0)$, is extraordinarily large. Consequently, their corresponding components in q' contribute a lot to the surrogate estimate p'. Small changes in these large components can result in huge changes in the solution; put differently, the pseudo-inverse estimator exhibits a considerable amount of *variance* if the singular values of M vary. This variance occurs despite the minimum-norm constraint of this estimator.

In this sense, we can characterize the difficulty of solving $q = \mathrm{M}p$ through the condition number $\kappa_\mathrm{M} = \frac{s_1}{s_k}$. This standard characteristic of a matrix expresses the range of its non-zero singular values. Physics-inspired OQ methods like RUN and IBU aim at reducing the variance of their estimates through regularization. A high condition number of M requires a strong regularization, which can produce considerably biased results.

However, a considerable reduction of the variance can also be achieved by suitable feature transformations f, namely by those transformations which lead to matrices M with low condition numbers. To this end, Fig. 4.1 compares a k-means clustering [1], a decision tree partition [101], and an equi-distant binning of the single, most predictive feature [98].

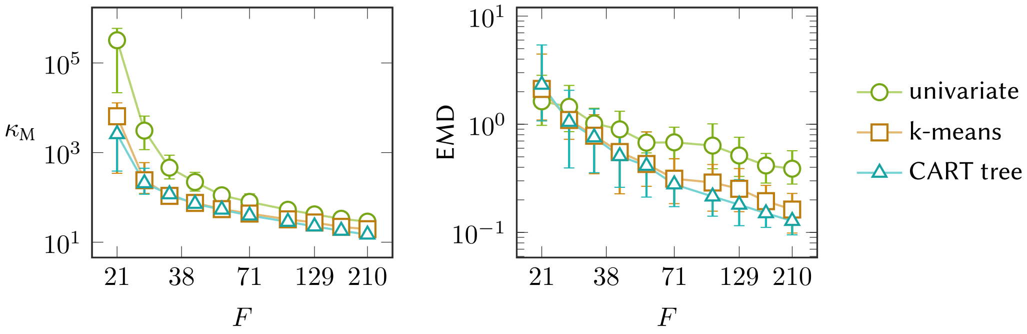

Figure 4.1.: The condition number κ_M (left) of the confusion matrix $M \in \mathbb{R}^{C \times F}$ decreases with an increasing number F of categories used in the feature transformation $f : \mathcal{X} \rightarrow \{1, \dots, F\}$ that defines $q \in \mathbb{R}^F$. The CART decision tree produces the least difficult problems, as according to κ_M, which also leads to lower errors in terms of the Earth Mover's Distance (EMD; right). This figure is taken from one of our publications [1].

Due to the strong performance of the decision tree partition, we employ this approach in all unfolding algorithms.

4.5. Evaluation

Our empirical evaluation of unfolding / quantification algorithms focuses on the reconstruction of energy spectra, the problem in astro-particle physics that we have detailed in Sec. 3.3. The goal of our experiments is to answer the following questions:

Q1 How suitable are the existing methods from quantification and unfolding, which we have introduced in Sec. 4.1, for the reconstruction of energy spectra?

Q2 How does our soft-max trick from Sec. 4.2 compare to implementations that employ standard, constrained optimization?

Q3 How do our regularized OQ methods from Sec. 4.3 compare to the existing methods?

Our focus on energy spectra manifests in an evaluation that employs the simulated data of the FACT telescope. To produce labeled test samples for quantification, we draw these samples from the simulated data. The sampling is taken out in three different, experimental protocols, which allow for a broad discussion of the relative merits of each method. We describe these protocols in the following sub-section.

Our focus also manifests in the choice of the evaluation metric. Since the reconstruction of energy spectra is a problem of *ordinal* quantification, we deem the normalized match distance (NMD) from Def. 2.8 most suitable. This distance is directly aimed at ordinal probability vectors but does, by itself, not consider the fact that the estimates need to be accurate in the *logarithmic* and *acceptance-corrected* plots that physicists typically inspect.

Therefore, we compute this distance with respect to logarithmic acceptance-corrected spectra

$$\log_{10}(a \odot p), \tag{4.36}$$

which are the subject of the physical analyses, as detailed in Sec. 3.3. Namely, we compute the quality of an estimate p as

$$\mathrm{NMD}\left(\frac{\log_{10}(a \odot p)}{\sum_{i=1}^{C} [\log_{10}(a \odot p)]_i}, \; \frac{\log_{10}(a \odot p^*)}{\sum_{i=1}^{C} [\log_{10}(a \odot p^*)]_i} \right), \tag{4.37}$$

where p^* is the vector of ground-truth class prevalences in the current sample. Effectively, the logarithm and the acceptance correction weight the NMD appropriately, in order to reflect the fact that a physicist typically inspects $\log_{10}(a \odot p)$ instead of p.

In some of our publications [3], [5], we have also evaluated a part of the presented algorithms on data sets that do not stem from astro-particle physics, but from text quantification. Accordingly, we have taken out these evaluations in terms of standard performance metrics. In these experiments, we have found that our ordinal methods o-PACC, o-ACC, and o-SLD perform well in ordinal settings, in terms of the standard NMD metric without acceptance correction and the logarithm [5]. Moreover, our soft-max implementation indeed outperforms constrained optimization in terms of the absolute error and the relative absolute error [3]. The domain-specific performance according to Eq. 4.37, however, has not been employed in these works. It is focused in the following experiments to allow for an evaluation that is even more targeted at the precise needs of knowledge acquisition in astro-particle physics. We revisit our previous results [3], [5] in Sec. 4.5.3.

4.5.1. Data Sampling Protocols

Our evaluation requires a large set of test samples $(\mathcal{D}_X, p^*)$ with known ground-truth class prevalences $p^* \in \mathbb{R}^C$. Each quantification method receives each sample $\mathcal{D}_X = \{x_i \in \mathcal{X} : 1 \leq i \leq m\}$ as an input and returns the corresponding estimate p. We measure the quality of each method in terms of its average performance in terms of Eq. 4.37. To generate samples with known ground-truth prevalences, we draw all samples from the labeled data of the FACT simulation.

The general process through which we draw the samples is illustrated in Fig. 4.2. First, we draw a labeled training set $\mathcal{D}_{XY}$ at random. All remaining data items are split into one pool for the validation of hyper-parameters and one pool for testing. Due to this split, there is no leakage between the data items that are used for training, for hyper-parameter optimization, and for testing. From each of the two pools, we generate the individual samples $(\mathcal{D}_X, p^*)$ by first drawing some class prevalence vector p^*. This draw is taken out according to one of the specific protocols, which we introduce in the next paragraphs. Then, we draw a set of data items $\mathcal{D}_X$, which exhibits the class proportions p^*, from the respective pool. We draw a number of samples $(\mathcal{D}_X, p^*)$ from each pool separately, to obtain a set of validation samples and a set of testing samples, which are disjunct from each other.

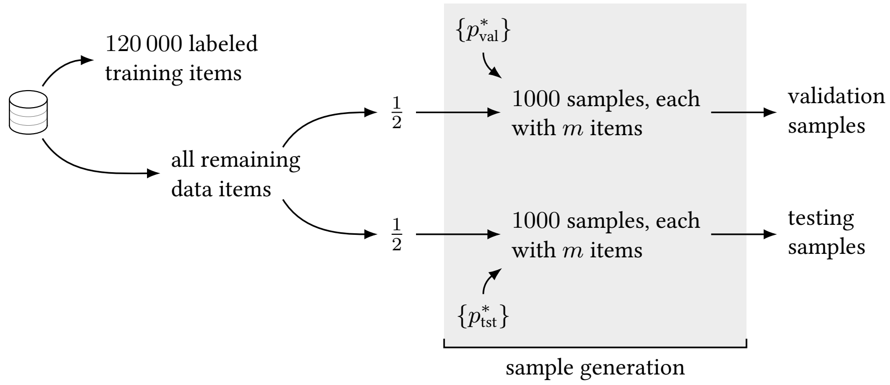

Figure 4.2.: The general process of sample generation for the evaluation of quantification methods. We first draw a training set and split the remaining data into a validation pool and a training pool. In each pool, we draw data samples according to ground-truth class prevalence vectors p^*_{val} and p^*_{tst}. Our experiments feature three different protocols for the determination of these prevalence vectors, as well as two values for m, the number of items in each sample.

The specific protocols, which implement this process, differ in terms of the prevalence vectors p^*, which they generate. Through their differences, they allow us to explore different spaces of quantification problems. We introduce multiple variants of each protocol before we select, from each protocol, the most suitable variant for our evaluation.

Natural Prevalence Protocol

The natural prevalence protocol (NPP) keeps the class proportions p^* fixed to those proportions that occur naturally in the data. Typically, these proportions match the proportions of the training set, so that no prior probability shift occurs in the NPP samples, i.e., $\mathbb{P}_{\mathcal{T}}(Y) = \mathbb{P}_{\mathcal{S}}(Y)$. Therefore, the typical NPP is not appropriate for assessing the required robustness of quantification methods against prior probability shift.

Here, we consider two variants of NPP, which stem from the prior probability shift that already exists between the simulated data and the real telescope data. In the data of the real telescope, the highest energy levels of gamma rays occur only extremely rarely. However, physicists consider the highest energies to be the most interesting. Moreover, the extreme imbalance between low-energy and high-energy events typically hinders effective machine learning, so that physicists have decided to artificially over-sample the interesting high-energy events in their simulations.

The natural prevalences, as according to the real data and as according to the simulated data, are plotted in Fig. 4.3. These prevalences give rise to the two NPP variants we consider here, “NPP (simulation)” and “NPP (Crab)”. The first of these variants introduces no prior probability shift, and hence, is not appropriate for assessing the required robustness of quantification methods against prior probability shift. The second of these variants

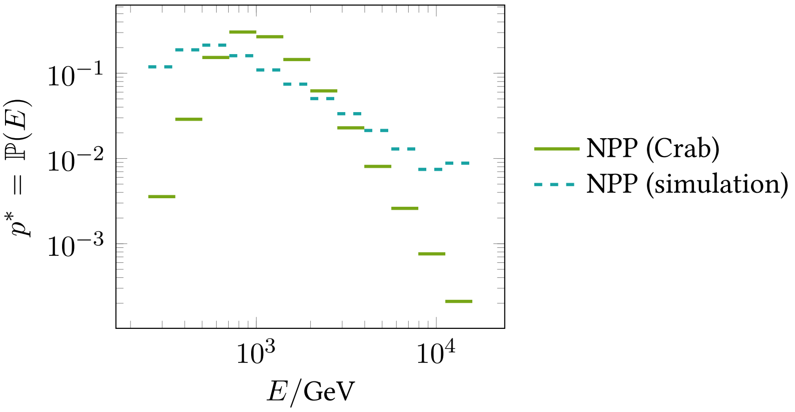

Figure 4.3.: The fixed class prevalences p^* that are used in the two variants of the natural prevalence protocol. Here, the horizontal axis plots the particle energy in Giga-electron-Volt (GeV) and the vertical axis plots the probabilities of the energy levels, i.e., the probabilities of the ordinal classes. The simulation exhibits an artificial oversampling of high energies, due to their importance and due to their extreme rarity in the real telescope data.

does introduce a prior probability shift, but only exactly one particular shift instance, which stems from the fixed, underlying prevalence vector.

The prevalence vector of the real telescope data, which yields the "NPP (Crab)" protocol, specifically models the expected energy density of gamma rays from the Crab nebula, a bright gamma ray source. This modelling is achieved in two steps. First, a parametric model of the actual flux of the Crab nebula [127],

$$\Phi(E) \;=\; \frac{3.23 \cdot 10^{-11}}{\text{TeV} \cdot \text{cm}^2 \cdot \text{s}} \cdot \left(\frac{E}{\text{TeV}}\right)^{-2.47 - 0.24 \cdot \log_{10}(E/\text{TeV})}, \tag{4.38}$$

is evaluated at the centers of the energy bins, which define our ordinal classes $\mathcal{Y}$. The resulting values, $p^*_{\text{Crab}} \in \mathbb{R}^C$, are then divided by the acceptance factors $a \in \mathbb{R}^C$ of the FACT telescope, to simulate the limited and energy-dependent detection efficiency of FACT. The outcome of this division,

$$[p^*]_i \;=\; \frac{[p^*_{\text{Crab}}]_i}{[a]_i} \quad \forall\, 1 \leq i \leq C, \tag{4.39}$$

defines the single, fixed prevalence vector of the "NPP (Crab)" variant of NPP. This vector is domain-specific, thus well suited for the evaluation of quantification for energy spectra, and it introduces prior probability shift for an assessment of quantification robustness. We compute the required acceptance factors a from closed meta-data of the FACT telescope.

Poisson Protocol

A disadvantage of "NPP (Crab)" is that only a single, fixed prevalence vector is used. Therefore, we propose another evaluation protocol, "Poisson (Crab)", that introduces a domain-specific variability of the class prevalence vectors. Like "NPP (Crab)", this protocol is based on a parametric model of the Crab nebula flux [127], divided by the acceptance factors of the FACT telescope. Therefore, the foundation of "Poisson (Crab)" is again Eq. 4.39.

To generate a domain-specific variability from this single prevalence vector, we interpret each number of examples, $m \cdot \frac{[p^*_{\text{Crab}}]_i}{[a]_i}$, as the rate of a Poisson distribution. From this distribution, we draw an actual, observed number $[m^*]_i$, which we normalize to obtain a probability density

$$[p^*]_i \;=\; \frac{[m^*]_i}{\sum_{j=1}^{C} [m^*]_j} \quad \forall\, 1 \leq i \leq C. \tag{4.40}$$

Through this protocol, we thus obtain prevalence vectors that reflect i) a proper model of the Crab nebula flux [127], ii) the acceptance of the FACT telescope, and iii) the random variations that physicists expect trough class-wise Poisson distributions. Indeed, *physicists have confirmed to us that this protocol accurately reflects the task of reconstructing energy spectra.* Due to these qualities, we consider "Poisson (Crab)" as our gold-standard protocol for the evaluation of quantification methods with respect to the reconstruction of energy spectra from IACT recordings.

Artificial Prevalence Protocol

The artificial prevalence protocol (APP) produces class prevalence vectors that uniformly populate the entire unit simplex of feasible solutions $\mathcal{P}$ from Def. 4.1. Therefore, APP explores every possible instance of prior probability shift and is hence well suited for assessing the robustness of quantification methods against prior probability shift.

Typically, however, not all possible prevalence vectors are equally likely in a given application. Therefore, most prevalence vectors of APP are often not representative for the application at hand. Unfortunately, this statement certainly holds for the reconstruction of energy spectra from IACT recordings. Averaging the performance over many unrepresentative samples can lead to overly conservative results, which consider only shift robustness but not the application-related merits of quantification methods.

Filtering Artificial Prevalences for Ordinal Plausibility

For this reason, we have developed APP-OQ [5], a variant of APP which filters candidate vectors p^* in terms of their plausibility in ordinal quantification. We measure plausibility in terms of their smoothness, as according to the Tikhonov regularization term $\frac{1}{2}\big(\mathrm{T} \cdot \log_{10}(a \odot p^*)\big)^2$ from Eq. 4.8. In particular, we maintain only those candidate vectors p^* which exhibit the lowest values of this term and we explore different fractions (i.e.,

Table 4.4.: Average smoothness $\frac{1}{2}\big(\mathrm{T} \cdot \log_{10}(a \odot p^*)\big)^2$ of the logarithmic acceptance-corrected ground-truth spectra p^*, which are sampled according to each of the protocols. We test two samples sizes m. Low values indicate smooth spectra.

m	protocol	percentile					average
		5th	25th	50th	75th	95th	
1000	APP	.00895	.01733	.03098	.05466	.14605	.04740
	APP-OQ (20%)	.00631	.01139	.01824	.03022	.05846	.02388
	APP-OQ (5%)	.00479	.00881	.01390	.02296	.04184	.01774
	APP-OQ (1%)	.00380	.00717	.01133	.01860	.03418	.01433
	NPP (Crab)	.00014	.00014	.00014	.00014	.00014	.00014
	NPP (simulation)	.00203	.00205	.00205	.00205	.00207	.00205
	Poisson (Crab)	.00018	.00032	.00052	.00091	.00233	.00081
10000	APP	.00592	.01144	.01829	.02969	.06612	.02506
	APP-OQ (20%)	.00381	.00642	.00896	.01243	.02127	.01019
	APP-OQ (5%)	.00272	.00437	.00601	.00821	.01368	.00682
	APP-OQ (1%)	.00183	.00304	.00421	.00586	.00976	.00482
	NPP (Crab)	.00002	.00002	.00002	.00002	.00002	.00002
	NPP (simulation)	.00051	.00051	.00051	.00051	.00051	.00051
	Poisson (Crab)	.00003	.00005	.00008	.00013	.00023	.00010

Table 4.5.: Average NMD, as according to Eq. 4.37, between the logarithmic acceptance-corrected training distribution p_0 and the logarithmic acceptance-corrected ground-truth spectrum p^*, where p^* is sampled according to each of the protocols. Low values indicate small magnitudes of prior probability shift.

m	protocol	percentile					average
		5th	25th	50th	75th	95th	
1000	APP	.02830	.04343	.05491	.06965	.12802	.06196
	APP-OQ (20%)	.02680	.04111	.05254	.06668	.12870	.05983
	APP-OQ (5%)	.02629	.04060	.05177	.06502	.11800	.05805
	APP-OQ (1%)	.02610	.04026	.05100	.06370	.11433	.05689
	NPP (Crab)	.10942	.10948	.10950	.10951	.10958	.10950
	NPP (simulation)	.00050	.00055	.00056	.00058	.00067	.00060
	Poisson (Crab)	.07895	.09321	.10602	.11143	.11780	.10210
10000	APP	.03362	.05105	.06624	.08692	.13733	.07290
	APP-OQ (20%)	.03234	.04944	.06438	.08541	.13702	.07139
	APP-OQ (5%)	.03235	.04881	.06380	.08424	.13323	.07035
	APP-OQ (1%)	.03162	.04839	.06290	.08296	.13319	.06957
	NPP (Crab)	.07801	.07801	.07801	.07801	.07801	.07801
	NPP (simulation)	.00005	.00006	.00006	.00006	.00011	.00007
	Poisson (Crab)	.07087	.07469	.07782	.08132	.08681	.07821

20%, 5%, and 1%) to be maintained. All other candidates are discarded and, hence, are not used to generate samples $\mathcal{D}_X$. In order to have 1000 samples, like all other protocols, we over-produce candidate vectors so that, after discarding, 1000 vectors remain.

While APP-OQ clearly produces class prevalence vectors that are plausible for ordinal quantification, we might ask: which of the APP-OQ fractions is the most appropriate for our use case? In fact, this question has motivated us to come up with our gold-standard "Poisson (Crab)" protocol, with which we address this question in the following.

Selection of Protocols

We intend to explore different spaces of quantification problems through different protocols, but we also want to focus our evaluation on the most suitable variants for reconstructing energy spectra in astro-particle physics. In particular, we are looking for protocol variants which are allowed to behave a little different from the "Poisson (Crab)" protocol, without being completely alien to our use case.

To compare the variants introduced above, we evaluate the sets of class prevalence vectors, which they produce, in two regards. First, we evaluate the vectors in terms of their logarithmic, acceptance-corrected smoothness, i.e., in terms of $\frac{1}{2}(\mathrm{T} \cdot \log_{10}(a \odot p^*))^2$. The average values of this metric, for each protocol, are given in Tab. 4.4. Second, we evaluate the class prevalence vectors in terms of their prior probability shift. To this end, we measure the distance between each generated class prevalence vector p^* and the training set prevalences p_0 through Eq. 4.37; later, we measure the quality of quantification estimates in the same way. The average shift values, for each protocol, are given in Tab. 4.5.

Based on the results from Tab. 4.4 and Tab. 4.5, we select "NPP (Crab)" and "APP-OQ (1%)" as our evaluation protocols, in addition to our gold-standard "Poisson (Crab)" protocol. We select "NPP (Crab)" because it behaves similarly to "Poisson (Crab)" in terms of smoothness and in terms of prior probability shift. The only exception to this observation is the smoothness for $m = 1000$, where "NPP (Crab)" produces much smoother and more constant prevalence vectors than "Poisson (Crab)". We select "APP-OQ (1%)" because it is, among all APP variants, the variant that is closest to "Poisson (Crab)", despite behaving quite differently. In particular, "APP-OQ (1%)" leads to a considerable variability in terms of smoothness and prior probability shift. Therefore, this protocol represents a more conservative assessment of quantification performance, i.e., an assessment with a stronger focus on robustness to prior probability shift, than the "Poisson (Crab)" protocol.

4.5.2. Comparison

Our implementations of all methods and experiments are available online.[8] Tab. 4.6 and Tab. 4.7 display the results of our comparison, as according to the selected protocols and the evaluation metric from Eq. 4.37. Due to the protocols and the evaluation metric,

[8] `https://github.com/mirkobunse/QUnfold.jl`

Table 4.6.: Average NMD (see Eq. 4.37, lower is better) with standard deviations of the logarithmic acceptance-corrected estimate p. Here, we set $m = 1000$ and print the best performance in boldface. All methods that perform worse than the sum of the best average and its standard deviation are printed in gray.

method	APP-OQ (1%)	NPP (Crab)	Poisson (Crab)
RUN (original)	.0539±.0565	.0963±.0461	.0937±.0449
SVD (original)	.0264±.0363	.0921±.0439	.0993±.0474
IBU	.0344±.0364	.0173±.0056	.0171±.0070
ACC (constrained)	.1963±.0945	.1225±.0292	.1287±.0327
PACC (constrained)	.0579±.0558	.0816±.0385	.0810±.0390
HDx (constrained)	.0767±.0846	.0742±.0486	.0733±.0496
HDy (constrained)	.1420±.1014	.1269±.0367	.1335±.0387
RUN (softmax)	**.0192**±**.0203**	.0134±.0049	.0205±.0123
SVD (softmax)	.0260±.0261	.0216±.0130	.0196±.0121
o-ACC (softmax)	.0275±.0249	.0093±.0053	**.0143**±**.0099**
o-PACC (softmax)	.0326±.0301	**.0080**±**.0036**	.0148±.0106
o-HDx (softmax)	.0320±.0318	.0166±.0091	.0166±.0098
o-HDy (softmax)	.0235±.0228	.0163±.0082	.0159±.0086
o-SLD	.0301±.0340	.0195±.0187	.0182±.0085
SLD	.0262±.0287	.0696±.0360	.0664±.0353

Table 4.7.: Average NMD (see Tab. 4.6) with $m = 10\,000$.

method	APP-OQ (1%)	NPP (Crab)	Poisson (Crab)
RUN (original)	.0221±.0256	.0616±.0232	.0622±.0245
SVD (original)	.0166±.0201	.0613±.0267	.0610±.0278
IBU	.0243±.0143	.0044±.0029	.0068±.0043
ACC (constrained)	.1840±.0428	.0983±.0092	.0989±.0104
PACC (constrained)	.0224±.0194	.0346±.0266	.0335±.0252
HDx (constrained)	.0530±.0566	.0493±.0309	.0494±.0314
HDy (constrained)	.1201±.0653	.1266±.0116	.1260±.0141
RUN (softmax)	**.0120**±**.0109**	**.0031**±**.0015**	**.0061**±**.0038**
SVD (softmax)	.0197±.0148	.0057±.0025	.0076±.0044
o-ACC (softmax)	.0217±.0154	.0041±.0014	.0069±.0039
o-PACC (softmax)	.0253±.0192	.0051±.0012	.0075±.0040
o-HDx (softmax)	.0204±.0172	.0037±.0020	.0064±.0043
o-HDy (softmax)	.0169±.0141	.0036±.0016	.0062±.0039
o-SLD	.0201±.0114	.0035±.0018	.0063±.0038
SLD	.0169±.0129	.0655±.0149	.0651±.0162

these results are highly domain-specific, despite the fact that all methods can also be applied to quantification tasks outside of astro-particle physics. We have set the domain-specific focus to address **Q1–Q3**, the questions we have brought up in the beginning of this section. From Tab. 4.6 and Tab. 4.7, we draw the following conclusions.

Q1: Suitability of Existing Methods

First, we assess how suitable the existing quantification / unfolding methods from Sec. 4.1 are. In Tab. 4.6 and in Tab. 4.7, these methods are listed as "RUN (original)", "SVD (original)", "IBU", "ACC (constrained)", "PACC (constrained)", "HDx (constrained)", "HDy (constrained)", and "SLD". The best methods among this collection are "SLD", in terms of the "APP-OQ (1%)" protocol, and "IBU", in terms of the other protocols. However, our soft-max and regularization proposals outperform the existing "IBU" and "SLD".

In terms of quantification performance, a minimum requirement for any quantification method is to achieve some average performance values that are well below the prior probability shift values from Tab. 4.5. If a quantification algorithm would not meet this requirement, it would be advisable to simply return the mistaken training class prevalences p_0, instead of running this algorithm. Fortunately, however, we see from Tab. 4.6 and Tab. 4.7 that most quantifiers easily meet this requirement. An exception of this observation are "ACC (constrained)" and "HDy (constrained)", which are, on average, worse than the training class prevalences in all three protocols. Therefore, all methods, except for "ACC (constrained)" and "HDy (constrained)", are indeed capable of counteracting prior probability shift in the quantification tasks that are posed by the reconstruction of energy spectra.

We also see that all methods improve between Tab. 4.6, where each sample comprises $m = 1000$ data items, and Tab. 4.7, where each sample comprises $m = 10\,000$ data items. This behavior is expected and desired.

Q2: Impact of our Soft-Max Trick

We have already seen in Tab. 4.3 that our soft-max trick from Sec. 4.2 improves the performance of the existing methods in terms of the absolute error and in terms of the relative absolute error. Here, we see that RUN and SVD achieve huge improvements in terms of the performance metric from Eq. 4.37. For instance, "RUN (softmax)" outperforms "RUN (original)" in the "Poisson (Crab)" protocol by 78% with $m = 1000$ (0.0205 versus 0.0937) and by 90% with $m = 10\,000$ (0.0061 versus 0.0622).

Q3: Impact of Ordinal Regularization

We now take a closer look at the regularized methods "o-ACC (softmax)", "o-PACC (softmax)", "o-HDx (softmax)", "o-HDy (softmax)", and "o-SLD", which we have developed in

Sec. 4.3. These methods draw inspiration from RUN, SVD, and IBU, which are regularized through the same techniques.

We recognize that the best performances, according to all protocols and sample sizes, are regularized methods. In particular, the best performances across all protocols and sample sizes are “RUN (softmax)”, “o-ACC (softmax)”, and “o-PACC (softmax)”. Moreover, none of the non-regularized methods achieves a performance that is within one standard deviation of the best method, except for SLD and PACC in the “APP-OQ (1%)” protocol. At this point, we emphasize that the best regularization parameters of all regularized methods are selected on the basis of the average validation performance in each protocol, while Tab. 4.4 reveals a variability of the smoothness, which the individual samples within each protocol exhibit. Hence, the regularization parameter has to fit all samples. We conceive further room for improvements through a regularization that targets each individual sample.

In this experiment, the winning performances of o-ACC and o-PACC and the high performances of o-HDx, o-HDy, and o-SLD are particularly noteworthy for two reasons. First, these algorithms do not originate in astro-particle physics, unlike RUN, SVD, and IBU. Second, our experiment is specifically targeted at an astro-particle use case, in which RUN, SVD, and IBU might be expected to exhibit the best performances. Therefore, the high performance of the other methods strongly supports our claim that quantification methods from outside of unfolding literature, if they are properly regularized and optimized, can indeed lead to accurate reconstructions of energy spectra.

4.5.3. Previous Results

A comparison similar to the one above already appears in one of our publications [5]. To broaden our discussion towards other applications of quantification, outside of astro-particle physics, we revisit the results of this comparison in Tabs. 4.8 and 4.9.

The differences between Tabs. 4.8 and 4.9 and the results presented in the section above are as follows. First, we have evaluated our methods on data from text quantification, where the goal is to quantify five ordinal classes that represent a scale of five stars in product reviews. Second, we have evaluated in terms of a standard NMD metric, see Def. 2.8, instead of the domain-specific performance measure from Eq. 4.37; we have further evaluated in terms of RNOD [60], another evaluation metric for ordinal quantification. Third, we have regularized in terms of the “plain” Tikhonov regularizer $(\mathrm{T}p)^2$ from Eq. 4.7, instead of using the logarithmic and acceptance-corrected regularizer from Eq. 4.8. Fourth, we have employed a regular APP protocol and the variant “APP-OQ (20%)”, instead of the more domain-specific evaluation protocols, which we have used above. Fifth, we have employed a slightly different selection of quantification methods. Due to these differences, we must be cautious with drawing conclusions for the reconstruction of energy spectra in particular. However, these differences allow us to broaden our discussion to an application of text quantification and to more conservative evaluation protocols, “APP” and “APP-OQ (20%)”, which emphasize robustness against prior probability shift.

Table 4.8.: Average NMD (see Def. 2.8, lower is better), of the plain estimate p, without a logarithm and without the acceptance correction. Here, we set $m = 1000$. The methods "o-ACC", "o-PACC" and "RUN" correspond to the constrained versions thereof, without our soft-max proposal. This table is adapted from one of our publications [5].

method	Amazon (RoBERTa)		Amazon (TFIDF)		FACT	
	APP	APP-OQ	APP	APP-OQ	APP	APP-OQ
CC	.0526±.019	.0344±.013	.0867±.034	.0683±.031	.0534±.012	.0494±.011
PCC	.0629±.022	.0440±.017	.1082±.044	.0950±.048	.0651±.017	.0621±.017
ACC	.0229±.009	.0193±.007	.0353±.015	.0333±.014	.0582±.028	.0575±.028
PACC	.0209±.008	.0176±.007	.0301±.015	.0310±.015	.0791±.048	.0816±.049
SLD	**.0172±.007**	.0154±.006	.0477±.018	.0381±.012	.0373±.010	.0355±.009
OQT	.0775±.026	.0587±.027	.1583±.065	.1539±.072	.0746±.019	.0731±.020
ARC	.0641±.023	.0477±.015	.0989±.037	.0855±.038	.0566±.014	.0568±.016
IBU	.0253±.010	.0197±.007	.0596±.023	.0454±.020	**.0213±.005**	.0187±.004
RUN	.0252±.010	.0198±.007	.0594±.023	.0452±.020	.0222±.006	.0194±.005
o-ACC	.0229±.009	.0188±.007	.0347±.017	.0227±.009	.0274±.007	.0230±.006
o-PACC	.0209±.008	.0174±.007	**.0276±.014**	**.0194±.007**	.0230±.006	**.0178±.004**
o-SLD	.0173±.007	**.0152±.006**	.0477±.018	.0363±.011	.0327±.008	.0289±.007

Table 4.9.: Average RNOD [60] (lower is better) in analogy to Tab. 4.8. This table is adapted from the supplementary material of one of our publications [5].

method	Amazon (RoBERTa)		Amazon (TFIDF)		FACT	
	APP	APP-OQ	APP	APP-OQ	APP	APP-OQ
CC	.1151±.048	.0606±.020	.1555±.062	.0953±.033	.1319±.036	.1071±.027
PCC	.1360±.054	.0758±.025	.1807±.063	.1244±.045	.1372±.034	.1096±.026
ACC	.0487±.024	.0374±.016	.0786±.039	.0735±.035	.1563±.040	.1375±.030
PACC	.0419±.019	.0327±.014	.0681±.037	.0708±.037	.1750±.056	.1719±.047
SLD	**.0363±.017**	.0302±.014	.1073±.051	.0814±.027	.0890±.029	.0767±.021
OQT	.1542±.064	.0960±.032	.2168±.071	.1659±.058	.1456±.035	.1225±.032
ARC	.1303±.056	.0770±.027	.1698±.065	.1123±.035	.1242±.032	.0973±.022
IBU	.0534±.025	.0357±.014	.1186±.052	.0678±.022	**.0822±.028**	.0649±.018
RUN	.0531±.025	.0361±.014	.1185±.053	.0675±.022	.0869±.029	.0685±.019
o-ACC	.0487±.024	.0353±.014	.0777±.038	.0465±.020	.1032±.033	.0754±.016
o-PACC	.0419±.019	.0316±.012	**.0624±.034**	**.0399±.017**	.0914±.029	**.0625±.016**
o-SLD	**.0365±.017**	**.0296±.013**	.0973±.036	.0688±.017	.0857±.027	.0658±.015

The columns in Tabs. 4.8 and 4.9 refer to the two evaluation protocols and to the three data sets we have employed. Here, the “FACT” data is the same data we have used throughout this thesis and the two columns “Amazon (RoBERTa)” and “Amazon (TFIDF)” refer to two text representations of a product review dataset [128]. To this end, we employ either RoBERTa [129] embeddings of the reviews or simple TFIDF features thereof. The classifier for classification-based quantification methods is a logistic regression [54], which performs highly competitive on both representations of the text data.

The results from Tabs. 4.8 and 4.9 suggest that ordinal regularization particularly benefits quantification methods if the quantification task is indeed ordinal. We draw this conclusion from the observation that regularized methods frequently win in the “APP-OQ (20%)” protocol, but not in the non-ordinal “APP” protocol. However, since this variant of APP-OQ maintains 20% of all samples, the improvement is not as pronounced as it is in Sec. 4.5.2, where APP-OQ maintains only the smoothest 1% of all samples. Moreover, Tabs. 4.8 and 4.9 suggest that quantification methods provide the largest benefit when the underlying classifier is weak. We draw this second conclusion from the observation that the powerful RoBERTa embedding allows all methods to achieve accurate results, while the weak TFIDF representation pronounces the performance differences between the quantification methods. Finally, we find that NMD and RNOD yield the same general conclusions in this experiment. In general, these conclusions are in line with our domain-specific conclusions from Sec. 4.5.2.

5. Class-Conditional Label Noise Learning

In this chapter, we explore the potential of weak supervision for gamma hadron classification, the essential prediction task for gamma ray astronomy that we have introduced in Sec. 3.5. By learning directly from the data that a *real* IACT produces, we solve this prediction task without the need for costly simulations. In particular, we learn a gamma hadron classifier by adopting the "on" and "off" annotations of real telescope recordings as *weak labels* with class-conditional label noise (CCN). These annotations are provided for each individual event, through the wobble mode from Fig. 3.2.

The potential of using "on" and "off" annotations as weak labels emerges from data that are taken while the telescope is pointed at a *known* gamma ray source, like the Crab nebula. By knowing that a gamma ray source indeed exists in the "on" region, we also know that a gamma ray *excess* must occur in this region. In particular, we know that $N_{\text{on}} > \alpha \cdot N_{\text{off}}$ must hold. Therefore, the "on" and "off" annotations $\widehat{y} \in \{\text{"on"}, \text{"off"}\}$, which the wobble mode provides for each *real* and individual telescope recording $x \in \mathcal{X}$, loosely correspond to the *true* class labels $y \in \{\text{"gamma"}, \text{"hadron"}\}$. In particular, the probability of observing a gamma ray in the "on" region is higher than observing a hadron in this region,

$$\mathbb{P}\Big(\widehat{Y} = \text{"on"} \;\Big|\; Y = \text{"gamma"}\Big) \;>\; \mathbb{P}\Big(\widehat{Y} = \text{"on"} \;\Big|\; Y = \text{"hadron"}\Big). \tag{5.1}$$

However, we cannot simply treat the "on" and "off" annotations as a direct replacement for the ground-truth "gamma" and "hadron" labels because there are non-zero probabilities

$$\begin{aligned} p_+ &= \mathbb{P}\big(\widehat{Y} = \text{"off"} \mid Y = \text{"gamma"}\big), \\ p_- &= \mathbb{P}\big(\widehat{Y} = \text{"on"} \mid Y = \text{"hadron"}\big), \end{aligned} \tag{5.2}$$

which characterize the class-conditional noise of the "on" and "off" weak labels. The mapping from the true "gamma" and "hadron" labels y to "on" and "off" annotations $\widehat{y}$, in terms of p_+ and p_-, is further illustrated in Fig. 5.1.

Theoretic results on learning under CCN [18], [130]–[132] suggest that Eq. 5.1 suffices for learning accurate gamma hadron classifiers from "on" and "off" annotations. In this chapter, we explore this potential to find that gamma hadron classification poses an original CCN problem in which the label noise is *partially known.* In particular, we show that the wobble mode ensures that exactly one of the class-wise noise rates is known and hence does not need to be estimated. Existing work on CCN, to the best of our knowledge, has either assumed the full knowledge of all noise rates [130], [133] or the complete ignorance

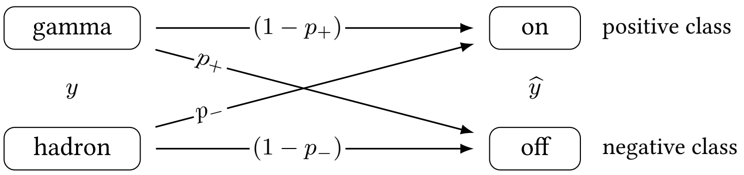

Figure 5.1.: CCN in gamma hadron classification. The goal is to distinguish gamma rays from hadrons, using the "on" and "off" annotations of real telescope recordings. This figure is adapted from one of our publications [7].

thereof [133], [134]. We demonstrate that ignoring the available, partial knowledge of the noise rates can lead to sub-optimal results.

We address partial knowledge in CCN through an objective function that a learning algorithm can optimize, be it in gamma hadron classification or on any other binary classification task under CCN. This objective function combines several qualities:

versatile: our objective can be used to optimize the decision threshold of any existing decision function $h : \mathcal{X} \to \mathbb{R}$ or to learn a new classifier from scratch. It can also be used to evaluate classification models in the typical case where no clean data is available for testing.

These merits require that one of the two class-wise noise rates is known. The wobble mode of IACTs meets this assumption by design.

interpretable: our objective is based on a hypothesis test, which we propose to assess whether learning is feasible with a given set of noisy labels. If clean ground-truth labels were available, we could employ this test directly. In gamma hadron classification, we employ an effective heuristic about this test.

straightforward: our objective function is easy to implement with off-the-shelf optimization tools.

practical: partial knowledge of the class-wise noise rates, as proposed by us, is relevant in real use cases. We present gamma hadron classification as one example.

This chapter is structured as follows: we first discuss class-conditional label noise, the properties of which are inherited by partially-known label noise, in Sec. 5.1. In Sec. 5.2, we then rephrase Li & Ma's hypothesis test from Def. 3.1 as a general test of CCN learnability. Sec. 5.3 introduces partial knowledge of CCN label noise and proposes algorithms to handle this setting. We empirically validate our proposals in Sec. 5.4 before we apply our proposals to a detection of the Crab nebula in Sec. 5.5.

5.1. Binary Classification Under Class-Conditional Label Noise

In binary classification under random label noise, we only have access to training labels $\widehat{y} \in \{+1, -1\}$ that are *noisy* in the sense of being randomly flipped versions of the *clean* ground-truth labels $y \in \{+1, -1\}$. We focus on the class-conditional random label noise model (CCN) [18], [130]–[132] in particular, which states that the labels are flipped according to the noise rates p_+ and p_-. These rates depend on the true class y but are independent of the features. Formally, they are defined as the label flip probabilities

$$p_y \;=\; \mathbb{P}\Big(\widehat{Y} = -y \mid Y = y\Big), \tag{5.3}$$

which might be known or unknown.

Remark 5.1 *(Compact notation).* Our subscript notation p_y with $y \in \{+1, -1\}$ and $N_{\widehat{y}}$ with $\widehat{y} \in \{+1, -1\}$ compactly represents the rates $p_+ = p_{+1}$ and $p_- = p_{-1}$ and the counts $N_+ = N_{+1}$ and $N_- = N_{-1}$. In other words, we omit the number "1" in subscripts when the meaning is clear from the context.

Conventionally, the feasibility of learning under class-conditional label noise is established by assuming a correspondence between noisy and true labels. Namely, the truly-positive class is required to exhibit a chance of positive noisy labels that is higher than the chance of positive noisy labels in the truly-negative class,

$$\mathbb{P}\Big(\widehat{Y} = +1 \mid Y = +1\Big) \;>\; \mathbb{P}\Big(\widehat{Y} = +1 \mid Y = -1\Big). \tag{5.4}$$

This assumption can be written equivalently as $p_+ + p_- < 1$. The connection between Eq. 5.4 and this compact form is established by the rearrangement

$$\begin{aligned} \mathbb{P}\big(\widehat{Y} = +1 \mid Y = +1\big) \;&>\; \mathbb{P}\big(\widehat{Y} = +1 \mid Y = -1\big) \\ \Leftrightarrow \quad 1 - \mathbb{P}\big(\widehat{Y} = -1 \mid Y = +1\big) \;&>\; \mathbb{P}\big(\widehat{Y} = +1 \mid Y = -1\big) \\ \Leftrightarrow \quad 1 \;&>\; p_+ + p_-. \end{aligned}$$

Based on this assumption, we formally define the CCN noise model for binary classification. In this definition, the random draw of some $\omega \in [0, 1)$ realizes label flips that occur randomly with probabilities p_+ and p_-.

Definition 5.1 (Binary CCN). *Let ω be the draw of a random variable that is uniformly distributed over $[0, 1)$. Further, let $p_+ + p_- < 1$ be the noise rates of the true classes $y \in \{+1, -1\}$. Binary CCN noisy labels $\widehat{y} \in \{+1, -1\}$ are defined by*

$$\widehat{y} \;=\; CCN\big(y\,;\omega,\, p_+,\, p_-\big) \;=\; \begin{cases} -y & \textit{if } \omega \leq p_y, \\ y & \textit{else} \end{cases}$$

We now discuss the general properties of binary CCN before we discuss CCN settings, which differ in whether p_+ and p_- are known or unknown.

5.1.1. Finding Optimal Decision Thresholds Under CCN

Theoretic studies have show that CCN noisy labels $\widehat{y}$ permit us to learn classifiers which are *optimal* even with respect to the clean ground-truth labels y [18], [130]–[132]. For the family of threshold-optimal performance metrics, the only difficulty lies in finding an optimal decision threshold $\theta \in \mathbb{R}$; a decision function $h : \mathcal{X} \to \mathbb{R}$, which yields the optimal hard classifier $h_\theta = \text{sign}(h(x) - \theta)$, can be learned directly from the noisy labels. This family of performance metrics, which is detailed in Sec. 2.2.2, includes weighted accuracy, the F_β score, and other relevant metrics.

Before proving this statement formally, let us illustrate the immediate implications of binary CCN in terms of the clean prediction function $\mathbb{P}(Y = +1 \mid X = x)$ and the noisy prediction function $\mathbb{P}(\widehat{Y} = +1 \mid X = x)$. Indeed, many learning algorithms produce decision functions h which are estimates of the clean $\mathbb{P}(Y = +1 \mid X = x)$. However, a CCN-unaware learning algorithm, when being trained only on noisily labeled data, would instead estimate the noisy variant $\mathbb{P}(\widehat{Y} = +1 \mid X = x)$ thereof.

Luckily, the following result states that the clean and noisy prediction functions are strictly monotone transformations of each other. Due to their strict monotonicity, there exists a unique one-to-one mapping between their probability values. A carefully chosen threshold for noisy probabilities $\mathbb{P}(\widehat{Y} = +1 \mid X = x)$ can therefore produce optimal predictions also with regard to the clean classes. Proposition 5.1 goes back to Natarajan et al. [130, supplementary material, proof of Lemma 7], who employ this result within a proof that addresses optimal thresholds for the 0-1 loss in particular.

Proposition 5.1. Assume binary CCN label noise according to Def. 5.1. There exists a strictly monotone transformation, which does not depend on x, between the true clean probabilities $\mathbb{P}(Y = +1 \mid X = x)$ and the true noisy probabilities $\mathbb{P}(\widehat{Y} = +1 \mid X = x)$. Namely, $\exists\ a > 0, b \in \mathbb{R}$, such that, $\forall\, x \in \mathcal{X} : \mathbb{P}(X = x) > 0$,

$$\mathbb{P}(\widehat{Y} = +1 \mid X = x) \ = \ a \cdot \mathbb{P}(Y = +1 \mid X = x) + b.$$

In particular, $a = 1 - p_+ - p_-$ and $b = p_-$.

Proof. The following rearrangement employs i) the law of total probability and ii) the fact that any CCN noisy label $\widehat{y}$ only depends on the clean label y; more specifically, any $\widehat{y}$

is conditionally independent of the features x given y. Moreover, we employ iii) the fact that $\mathbb{P}(Y=-1 \mid X=x) = 1 - \mathbb{P}(Y=+1 \mid X=x)$. For all $x \in \mathcal{X} : \mathbb{P}(X=x) > 0$,

$$
\begin{aligned}
&\mathbb{P}(\widehat{Y} = +1 \mid X = x) \\
&= \frac{1}{\mathbb{P}(X=x)} \mathbb{P}(\widehat{Y} = +1 \cap X = x) \\
&\overset{\text{i)}}{=} \frac{1}{\mathbb{P}(X=x)} \sum_{y \in \{+1,-1\}} \mathbb{P}(Y = y \cap \widehat{Y} = +1 \cap X = x) \\
&= \frac{1}{\mathbb{P}(X=x)} \sum_{y \in \{+1,-1\}} \mathbb{P}(\widehat{Y} = +1 \mid Y = y \cap X = x) \cdot \mathbb{P}(Y = y \cap X = x) \\
&\overset{\text{ii)}}{=} \frac{1}{\mathbb{P}(X=x)} \sum_{y \in \{+1,-1\}} \mathbb{P}(\widehat{Y} = +1 \mid Y = y) \cdot \mathbb{P}(Y = y \cap X = x) \\
&= \sum_{y \in \{+1,-1\}} \mathbb{P}(\widehat{Y} = +1 \mid Y = y) \cdot \mathbb{P}(Y = y \mid X = x) \\
&\overset{\text{iii)}}{=} (1-p_+) \cdot \mathbb{P}(Y = +1 \mid X = x) \, + \, p_- \cdot \big(1 - \mathbb{P}(Y = +1 \mid X = x)\big) \\
&= a \cdot \mathbb{P}(Y = +1 \mid X = x) \, + \, b
\end{aligned}
$$

Moreover, $a > 0$ due to the assumption $p_+ + p_- < 1$ from Def. 5.1. □

Remark 5.2. Menon et al. [18, Proposition 5] and Scott et al. [131, Proposition 1] present similar one-to-one correspondences between noisy and clean prediction functions in the context of binary classification with mutually contaminated distributions. While this context might appear to be different from CCN learning, they further show [18, Eq. 3] that both settings are in fact equivalent. In particular, both settings allow us to learn a prediction function from noisily labeled data and tune the decision threshold of the resulting model in order to optimally predict the clean classes.

To further prove the existence of an optimal threshold, we need a formal concept of optimality. To this end, we employ the performance metrics from Sec. 2.2.2, which are optimized by thresholded decision functions.

Proposition 5.2 (CCN consistency). Consider a performance metric $\mathcal{Q} : \mathcal{H} \to \mathbb{R}$ for which the Bayes-optimal classifier is of the form $h^*(x) = \operatorname{sign}(\mathbb{P}(Y=+1 \mid X=x) - \theta^*)$ with an optimal threshold $\theta^* \in \mathbb{R}$. Moreover, let the learning algorithm $\mathcal{A} : \cup_{m=1}^{\infty}(\mathcal{X} \times \mathcal{Y})^m \to \mathcal{H}$ be a consistent estimator of the scoring function, i.e., let $\forall x \in \mathcal{X} : \mathbb{E}_{(\mathcal{X}\times\mathcal{Y})^m}\, h_{\mathcal{A}}(x) \to \mathbb{P}(\widehat{Y} = +1 \mid X = x)$ for a noisy sample of size $m \to \infty$. Then, for $m \to \infty$, the classifier

$$h_{\lambda^*}(x) = \operatorname{sign}(h_{\mathcal{A}}(x) - \lambda^*)$$

is Bayes-optimal for $\mathcal{Q}$ with a threshold $\lambda^* = (1 - p_+ - p_-)\theta^* + p_-$.

Proof. Let $a = 1 - p_+ - p_- > 0$ and $b = p_-$. The following rearrangement employs i) Proposition 5.1 and ii) $m \to \infty$ with the fact that $\mathcal{A}$ is consistent. The Bayes-optimal classifier for $\mathcal{Q}$ is

$$\begin{aligned} h^*(x) &= \operatorname{sign}(\mathbb{P}(Y = +1 \mid X = x) - \theta^*) \\ &= \operatorname{sign}(a \cdot \mathbb{P}(Y = +1 \mid X = x) + b - a \cdot \theta^* - b) \\ &\overset{\text{i)}}{=} \operatorname{sign}(\mathbb{P}(\widehat{Y} = +1 \mid X = x) - (a \cdot \theta^* + b)) \\ &\overset{\text{ii)}}{=} \operatorname{sign}(h_{\mathcal{A}}(x) - (a \cdot \theta^* + b)), \end{aligned}$$

which yields the claim for $\lambda^* = a \cdot \theta^* + b$. □

Remark 5.3. Proposition 5.2 makes two assumptions: first, a performance metric for which the Bayes-optimal classifier is of the form $h^*(x) = \operatorname{sign}(\mathbb{P}(Y = +1 \mid X = x) - \theta^*)$; second, a consistent learning algorithm for the decision function $h_{\mathcal{A}}(x)$. The assumed family of performance metrics contains several important measures, like weighted accuracy and the F_β score, and the consistency assumption holds for any learning algorithm which evolves around empirical risk minimization of a proper loss [30], [33], like the logistic loss and the squared error. Therefore, the proposition is applicable to many performance metrics and learning algorithms, which are introduced in Sec. 2.2.2.

In practice, this result allows us to use the noisily labeled training data to fit a decision function h and to obtain an estimate $\widehat{\lambda} \in \mathbb{R}$ of the noisy-optimal threshold λ^*. From this estimate, we can estimate the clean-optimal threshold θ^* as

$$\widehat{\theta} = \frac{\widehat{\lambda} - p_-}{1 - p_+ - p_-}. \tag{5.5}$$

The remaining difficulty of binary classification under CCN is to estimate p_+ or p_- from the noisy data if at least one of these quantities is unknown.

Example 5.1 *(Accuracy).* The optimal threshold for this performance metric is known to be $\lambda^*_{\text{Acc}} = \frac{1}{2}$, see Tab. 2.1. Hence, we do not need to estimate this threshold from data but, due to CCN, we have to acknowledge that this threshold is only optimal for the *noisy* labels. As according to Eq. 5.5, an optimal *clean* threshold is $\theta^*_{\text{Acc}} = (\frac{1}{2} - p_-) \cdot (1 - p_+ - p_-)^{-1}$ [130, Theorem 9], which requires the precise knowledge or the estimation of p_+ and p_-.

So far, we have focused on the performance metrics from Sec. 2.2.2, which are optimized by thresholded decision functions. For a family of other performance metrics, a similar approach evolves around empirical risk minimization with CCN-aware loss weights [130], [133]. In the following, we keep focused on thresholded decision functions in order to study the effects of known and unknown noise rates in particular.

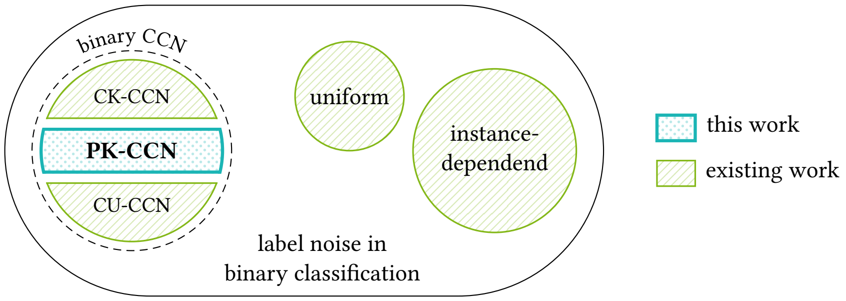

Figure 5.2.: Set diagram of label noise models. Partially-known class-conditional noise (PK-CCN; addressed in this work) is a subset of binary CCN, but distinct from completely known (CK-CCN) and completely unknown (CU-CCN) class-conditional noise, from uniform and purely instance-dependent label noise.

5.1.2. Completely Known and Completely Unknown Label Noise

The results of Propositions 5.1 and 5.2 hold independently of whether p_+ and p_- are known or unknown. If they are known precisely, we can already adapt the threshold of a decision function that is trained on noisy labels, as according to Eq. 5.5. Typically, however, p_+ and p_- are not known, so that an adaptation of the decision threshold requires additional efforts, like estimating these rates from noisy data [18], [131]. Without access to clean labels, this estimation requires additional assumptions, e.g. the existence of clean labels at a point in feature space where the other clean class has zero probability [18]. Class imbalance further complicates the estimation of the noise rates [134].

Due to the fundamental difference between handling CCN with known noise rates and handling CCN without known rates, we distinguish the two scenarios explicitly, as according to the following definitions. In Sec. 5.3, we introduce an additional setting, which appears in gamma hadron separation, where one noise rate is known and the other is not. Fig. 5.2 displays a set diagram of these different settings.

Definition 5.2 (CK-CCN). *Consider a training set with binary CCN noisy labels $\widehat{y}$ defined by Def. 5.1. Furthermore, let p_+ and p_- be* precisely known. *We call this setting* completely known *CCN or CK-CCN for short.*

Definition 5.3 (CU-CCN). *Consider a training set with binary CCN noisy labels $\widehat{y}$ defined by Def. 5.1. Furthermore, let p_+ and p_- be* unknown. *We call this setting* completely unknown *CCN or CU-CCN for short.*

Note that, if we had a small, cleanly labeled data set, we could tune the decision threshold directly on this set [132]. Such a cleanly labeled set of data might exist independently of whether p_+ and p_- are known. However, since gamma rays and hadronic particles are never ground-truth labeled, we do not discuss this possibility further.

5.1.3. Other Types of Label Noise

Beyond the CCN noise model, other types of label noise have been discussed in the scientific literature. For instance, the label noise is called *uniform* [133] if each label has the same chance of being flipped, independent of the class and of the instance. Otherwise, if the chance of being flipped does not depend on the true class, but on the features, we are speaking of *purely instance-dependent* label noise [135]. Learning is feasible under each of these noise models, given that dedicated assumptions about the data and the learning method hold. For gamma hadron separation, CCN is the most appropriate model.

Recently, multi-class settings are moving into the focus of CCN learning [136]–[140]. These settings require not only the estimation (or the knowledge) of two noise rates, but the estimation (or the knowledge) of a dense *transition matrix* between all noisy and true classes [137], [141]. This matrix is applied at inference time to compute the estimated clean class probabilities from the predictions for the noisy classes. In the context of deep learning, methods for the simultaneous training of the prediction model and the transition matrix have been proposed [142].

All of these multi-class methods either assume full knowledge of all noise rates or full ignorance of the rates. In this chapter, we focus on binary classification because CCN with partial knowledge of the noise rates—our focus topic—has a practical use case in the binary classification task that is posed by gamma hadron separation; an extension of partial knowledge to the multi-class setting shall remain open for future work.

We further recognize that the above mentioned, recent works on CCN [136]–[140] primarily optimize the accuracy metric, which is inadequate for imbalanced classification tasks [143]. The problem of class imbalance in CCN unfortunately has remained largely unexplored, with the notable exception of Mithal et al. [134]. In gamma hadron separation, due to its extreme and inherent class imbalance, we have to address this issue.

5.1.4. When is Class-Conditional Label Noise Unproblematic?

CCN label noise does not *always* affect the clean label performance of a model that is trained from noisy data. In fact, there are performance metrics and noise settings for which an optimal clean classifier can be trained from noisy data and without the need to handle CCN. At this point, however, we acknowledge these settings only for completeness. Gamma hadron separation, which aims at maximizing the significance of detection, is *not* among the settings where CCN is unproblematic. Therefore, we need to handle CCN in order to find accurate binary classifiers for our use case.

One performance measure that is robust to CCN is the AM metric, the arithmetic mean of the true positive rate and the true negative rate. This performance measure is optimized through a thresholded decision function with a decision threshold at the base rate $\mathbb{P}(Y = +1)$ [33], as introduced in Tab. 2.1. Accordingly, the optimal decision threshold for the noisy data, under a noisy decision function, is $\mathbb{P}(\widehat{Y} = +1)$. It might appear that the Bayes-optimal classifier for the clean data

$$\operatorname*{arg\,max}_{f : \mathcal{X} \to \mathcal{Y}} \mathrm{AM}(f; Y) \;=\; \mathrm{sign}\big(\mathbb{P}(Y = +1 \mid X = x) - \mathbb{P}(Y = +1)\big) \tag{5.6}$$

and the Bayes-optimal classifier for the noisy data

$$\operatorname*{arg\,max}_{f : \mathcal{X} \to \mathcal{Y}} \mathrm{AM}(f; \widehat{Y}) \;=\; \mathrm{sign}\big(\mathbb{P}(\widehat{Y} = +1 \mid X = x) - \mathbb{P}(\widehat{Y} = +1)\big) \tag{5.7}$$

are different. However, Menon et al. [18, Proposition 1] have proven a linear relationship between $\mathrm{AM}(f; Y)$ and $\mathrm{AM}(f; \widehat{Y})$. Due to this relationship, the maximizers from Eq. 5.6 and Eq. 5.7 are indeed *identical.* Also in terms of the area under the receiver operating characteristic, there is a linear relationship between the noisy and the clean metric [18, Corollary 3].

Therefore, when the the performance metric is either the AM measure or the area under the receiver operating characteristic, we do not need to address CCN at all. We can treat the noisy labels as if they were clean: no noise rates need to be estimated and the decision threshold does not need to be adapted because learning from noisy data is already consistent with respect to the clean data if the fundamental CCN assumption $p_+ + p_- < 1$ holds. However, we still pay a price for learning under label noise: the sample complexity, i.e., the number of data items needed to train an accurate model, increases due to CCN [18, Proposition 2 and Corollary 4].

For the ERM framework, Ghosh et al. [133] present additional conditions under which learning is robust to CCN. Their symmetry condition of the loss function is $\ell(h(x), +1) + \ell(h(x), -1) = c$ for some $c \in \mathbb{R}$ and all $x \in \mathcal{X}$. This condition is fulfilled, for instance, by the zero-one loss and by the sigmoid loss. ERM is robust to CCN if, under such a loss function, either i) the Bayes risk of the clean data is zero or ii) the label noise is uniform with $p_+ = p_-$ [133, Theorem 1]. Unfortunately, for gamma hadron separation, we cannot safely assume any of these two settings.

5.2. Testing CCN Learnability

In order to establish optimal CCN learning, we intend to validate the fundamental CCN assumption, i.e., $p_+ + p_- < 1$, through a hypothesis test. To this end, we formulate the null hypothesis

$$h_0 : \;\; p_+ + p_- \geq 1, \tag{5.8}$$

which states that this assumption was violated. Hence, we can validate the fundamental CCN assumption through a statistically significant rejection of h_0.

It turns out that we can approach the rejection of h_0 with the Li & Ma hypothesis test from Def. 3.1, which was originally proposed for source detection in particular. For general CCN learning, we now rephrase this test in terms of a CCN noise rate p_- and in terms of the true positives that are noisily labeled as positives and as negatives. We then prove i) that this rephrasing is indeed identical to the original hypothesis test from Def. 3.1, and ii) that this rephrasing indeed tests the null hypothesis from Eq. 5.8.

Definition 5.4 (Li & Ma hypothesis test for CCN). *Let $N_{\widehat{y}}$ be the number of true positives with a noisy label $\widehat{y} \in \{+,-\}$ and let $\alpha = \frac{p_-}{1-p_-}$. Moreover, let $\mathcal{N} : \mathbb{R} \to [0,1]$ be the cumulative density function of the standard normal distribution. The p-value for rejecting the null hypothesis $h_0 : p_+ + p_- \geq 1$ is*

$$\begin{aligned} p = & \; 1 - \mathcal{N}(f_\alpha(N_+, N_-)), \\ f_\alpha(N_+, N_-) = & \left[2N_+ \cdot \ln\left(\frac{1+\alpha}{\alpha} \cdot \frac{N_+}{N_+ + N_-} \right) \right. \\ & \left. + 2N_- \cdot \ln\left((1+\alpha) \cdot \frac{N_-}{N_+ + N_-} \right) \right]^{1/2}. \end{aligned}$$

To show that this test is indeed the Li & Ma hypothesis test from Def. 3.1, we proceed in two steps. First, we connect our general CCN definition of $\alpha = \frac{p_-}{1-p_-}$ to its domain-specific, original definition from source detection in astronomy. Second, we prove that our general, CCN-related null hypothesis is equivalent to the original null hypothesis from source detection.

The connection of the two α definitions stems from a particular characteristic of the “hadron” class. This class is expected to occur uniformly, i.e., at a constant rate, across all sky positions. Consequently, we know that $p_- = \mathbb{P}(\widehat{Y} = \text{“on”} \mid Y = \text{“hadron”})$ must exclusively depend on the areas of observation, A_{on} and A_{off}. More specifically, we know that $p_- = \frac{A_{\text{on}}}{A_{\text{on}}+A_{\text{off}}}$ must hold.

Theorem 5.1. The following definitions of the scaling factor $\alpha \in \mathbb{R}$ are equivalent:

1. (CCN learning; Def. 5.4) $\alpha = \dfrac{p_-}{1-p_-}$
2. (source detection; Def. 3.1) $\alpha = \dfrac{A_{\text{on}}}{A_{\text{off}}}$ with $p_- = \dfrac{A_{\text{on}}}{A_{\text{on}} + A_{\text{off}}}$

Proof. The second definition of α is recovered by plugging $p_- = \frac{A_{\text{on}}}{A_{\text{on}}+A_{\text{off}}}$ into the first definition of α. □

The equivalence of the hypothesis tests from Def. 5.4 and from Def. 3.1 is established through the following theorem. In particular, we show that the null hypotheses of both tests are equivalent, so that the original hypothesis test from Def. 3.1 is applicable to the null hypothesis from Def. 5.4.

Theorem 5.2. Let $\lambda_{\widehat{y}}$ be the rate of a Poisson distribution which models the number of true positives in the noisy class $\widehat{y} \in \{+, -\}$ and let $\alpha = \frac{p_-}{1-p_-}$. The null hypotheses from Def. 5.4 and Def. 3.1 are then equivalent, i.e.,

$$p_+ + p_- \geq 1 \quad \Leftrightarrow \quad \alpha\lambda_- \geq \lambda_+$$

Proof. Let N^+ be the number of true positives in the training set, so that the Poisson rates are given by $\lambda_- = p_+ N^+$ and by $\lambda_+ = (1 - p_+)N^+$.

$$\begin{array}{rlcl} & p_+ + p_- & \geq & 1 \\ \Leftrightarrow & p_+ p_- & \geq & 1 - p_- - p_+ + p_+ p_- \\ & & = & (1 - p_-)(1 - p_+) \\ \Leftrightarrow & p_+ \dfrac{p_-}{1 - p_-} & \geq & 1 - p_+ \\ \Leftrightarrow & \alpha p_+ N^+ & \geq & (1 - p_+)N^+ \\ \Leftrightarrow & \alpha\lambda_- & \geq & \lambda_+ \end{array}$$

Hence, the null hypotheses of both tests are equivalent. □

Due to Th. 5.1 and Th. 5.2, we can employ the test statistic f_α from the original Li & Ma hypothesis test to test the feasibility of CCN learning in terms of the fundamental CCN assumption, $p_+ + p_- < 1$. If the test from Def. 5.4 fails to reject, we can conclude that optimal, binary CCN learning is not sufficiently feasible with the given noisy labels.

We emphasize that the counts $N_{\widehat{y}}$ from Def. 5.4 require access to a set of data that is labeled *both* in terms of clean ground-truth labels y and in terms of noisy labels $\widehat{y}$. With a small data set of this kind and with a known rate p_-, we can use this hypothesis test to check whether the noisy labels allow for optimal CCN learning; if learning appears feasible, we can then use a larger set of noisily labeled data for training, without requiring clean ground-truth labels. However, the real telescope data is never cleanly labeled. Therefore, we cannot use the hypothesis test directly. In the next section, we develop an effective heuristic for this case, which employs an approximation of this test.

5.3. Partially-Known Class-Conditional Label Noise

Th. 5.1 has just revealed that the noise rate p_- is precisely known in gamma hadron classification. Hence, it does not need to be estimated. Existing work on CCN learning, to the best of our knowledge, has, instead, either assumed the full knowledge of all noise rates [130], [133] or the complete ignorance thereof [133], [134]. Motivated by gamma hadron classification, we now focus on a CCN setting where one of the class-wise noise rates is known but the other is not. We refer to this novel setting as *partially-known* CCN (PK-CCN) in binary classification.

Definition 5.5 (PK-CCN). *Consider a training set with binary CCN noisy labels $\widehat{y}$ defined according to Def. 5.1. Further let p_- be precisely known but let p_+ be unknown. We call this setting* partially known *CCN or PK-CCN for short.*

Like in general CCN, which does not make any assumption about whether p_- and p_+ are known, we can find a classification rule that is optimal with respect to the clean labels by optimizing the decision threshold of a decision function that is trained with noisy labels. In contrast to CK-CCN and CU-CCN, we employ a known p_- rate, but not the unknown p_+ rate, in the threshold optimization.

Our first goal is to choose a decision threshold that i) takes into account the partial knowledge of p_- and ii) maximally aligns with the foundational CCN assumption, i.e., with $p_+ + p_- < 1$. To this end, we employ a heuristic that evolves around our hypothesis test from Def. 5.4. Namely, we choose this threshold such that the outcome of the hypothesis test maximally agrees with the foundational CCN assumption. Our second goal is to learn not only a decision threshold, but a complete classifier that maximally aligns with this assumption.

5.3.1. Decision Threshold Optimization

So far, the counts $N_{\widehat{y}}$ from Def. 5.4 describe the numbers of *true* positives with a noisy label $\widehat{y}$. Thus, they require a set of data that is both ground-truth labeled and noisily labeled. In order to drop this impractical requirement, we heuristically replace the $N_{\widehat{y}}$ counts with counts of *predicted* positives. This replacement allows us to tune the decision threshold which leads to the predicted positives, so that the tuned threshold maximally aligns with the foundational CCN assumption $p_+ + p_- < 1$.

More specifically, we compute the function f_α from Def. 5.4 over $N^\theta_{\widehat{y}}$, the numbers of predicted positives according to a decision threshold θ. We choose θ such that f_α becomes maximal, i.e., for noisy labels $\widehat{y} \in \{+1, -1\}$, a decision function $h : \mathcal{X} \to \mathbb{R}$, and a noisily labeled data set $\{(x_i, \widehat{y}_i) : 1 \geq i \geq N\}$, we choose

$$\begin{aligned} \theta^* \;&=\; \underset{\theta}{\arg\max}\; f_\alpha(N^\theta_+, N^\theta_-), \\ \text{where}\;\; N^\theta_{\widehat{y}} \;&=\; \textstyle\sum_{i=1}^N \mathbb{1}_{\widehat{y}h(x_i) > \widehat{y}\theta}\,, \end{aligned} \tag{5.9}$$

so that f_α becomes the objective function with which we optimize the threshold θ.

Effectively, we have just replaced the true counts $N_{\widehat{y}}$ from Def. 5.4 with predicted counts $N^\theta_{\widehat{y}}$. This heuristic replacement has the following implications:

no need for clean labels: the optimization in Eq. 5.9 does not require any ground-truth labels; it only needs the counts of predicted positives in both noisy classes, which is easily obtained from the noisy data.

partial knowledge of noise rates: the optimization in Eq. 5.9 needs to know the rate p_-, so that $\alpha = \frac{p_-}{1-p_-}$ can be computed. However, we do not need to know the p_+ rate. Hence, our method is a true PK-CCN method.

model agnosticism: the optimization in Eq. 5.9 works with any decision function $h : \mathcal{X} \to \mathbb{R}$, like SVMs, decision trees, deep neural networks, and many more.

hypothesis testing: can we still test PK-CCN learnability if $N_{\widehat{y}}$ is replaced with $N^{\theta}_{\widehat{y}}$? Unfortunately, there is no guarantee that PK-CCN learnability can be tested with $N^{\theta}_{\widehat{y}}$. More specifically, we cannot conclude that PK-CCN is indeed feasible if the optimization from Eq. 5.9 leads to a successful rejection according to Def. 5.4. This conclusion would only be justified if the classifier was perfect, and hence $N_{\widehat{y}} = N^{\theta}_{\widehat{y}}$, which we cannot assume in general.

Therefore, despite being motivated through a hypothesis test, the optimization from Eq. 5.9 is only heuristic. However, this heuristic yields powerful classifiers, as we empirically demonstrate in Sec. 5.4. We can also use the heuristic performance metric $f_\alpha(N^{\theta}_{+}, N^{\theta}_{-})$ to evaluate classifiers without the need for clean ground-truth labels.

Our threshold optimization technique for PK-CCN is summarized in Alg. 5.1. If our heuristic does not indicate a successful rejection of the null hypothesis, we inform the user through an error message in line 6.

Algorithm 5.1 PK-CCN decision threshold tuning [7].

Input: A scoring function $h : \mathcal{X} \to \mathbb{R}$, a desired p-value $p > 0$, a noise rate $0 < p_- < 1$, and N noisily labeled instances $\{(x_i, \widehat{y}_i) : 1 \geq i \geq N\}$

Output: A decision threshold $\theta^* \in \mathbb{R}$

```
1: $\alpha \leftarrow \frac{p_-}{(1-p_-)}$
2: $\theta^* \leftarrow \arg\max_{\theta \in \mathbb{R}} f_\alpha(N^{\theta}_{+}, N^{\theta}_{-})$, see Eq. 5.9
3: if $p > 1 - \mathcal{N}(f_\alpha(N^{\theta^*}_{+}, N^{\theta^*}_{-}))$ then
4:     return $\theta^*$
5: else
6:     failure PK-CCN learning does not appear feasible
7: end if
```

The idea of tuning the decision threshold of a gamma hadron classifier by maximizing f_α is not new. In fact, this approach is frequently applied to gamma hadron classifiers that are trained from *simulated* data [144]. However, physicists have not yet discussed this approach in the context of CCN, which has motivated us to fill this gap. Our contribution is the proposal of learning directly from the *real* telescope data and to tune the threshold of the resulting *noisy* classifier by maximizing f_α. The potential of this proposal, to learn optimal classifiers without any simulated data, is supported by CCN theory. Moreover, this proposal opens the opportunity of applying Alg. 5.1 also to other use cases beyond gamma hadron classification.

5.3.2. Decision Tree Induction

The threshold tuning from Alg. 5.1 is particularly advantageous if some decision function h already exists. For the case in which no h exists, we now develop an algorithm which maximizes f_α over an entire model space, not only over the decision threshold θ.

To this end, we propose a decision tree induction algorithm which learns by maximizing f_α directly, i.e., in every split. Like classic decision trees, our algorithm recursively partitions the (noisy) training set in a greedy fashion: at each node, we split the data by thresholding a single feature. To find the best split, we evaluate f_α for both sides of the partition. We stop at a user-defined maximum tree depth or when no split can further improve f_α. Namely, each iteration of our tree induction algorithm selects a feature j and a feature threshold t, such that

$$\begin{aligned}
j^*,\, t^* \;&=\; \operatorname*{arg\,max}_{j,t}\; \max\left\{ f_\alpha^{X_j \leq t},\, f_\alpha^{X_j > t} \right\}, \\
\text{where} \quad f_\alpha^{X_j \leq t} \;&=\; f_\alpha\Big(N_+^{X_j \leq t},\, N_-^{X_j \leq t} \Big), \\
f_\alpha^{X_j > t} \;&=\; f_\alpha\Big(N_+^{X_j > t},\, N_-^{X_j > t} \Big), \\
N_{\widehat{y}}^{X_j \leq t} \;&=\; \textstyle\sum_{i=1}^N \mathbb{1}_{[x_i]_j \,\leq\, t \,\wedge\, \widehat{y}_i = \widehat{y}}\,, \\
N_{\widehat{y}}^{X_j > t} \;&=\; \textstyle\sum_{i=1}^N \mathbb{1}_{[x_i]_j \,>\, t \,\wedge\, \widehat{y}_i = \widehat{y}}\,,
\end{aligned} \tag{5.10}$$

The best split according to Eq. 5.10 can be evaluated efficiently by sorting the data according to each of the features. Considering only the maximum of $f_\alpha^{X_j \leq t}$ and $f_\alpha^{X_j > t}$ amounts to the fact that a greedy maximization of f_α does not require balanced splits: if the other side of a split results in a low value of f_α, we can either discard this side without harming the overall f_α or we can split this side further in a later step. As f_α increases with larger counts, the greedy approach is not tempted to split off individual examples. We have summarized our method in Alg. 5.2.

As with classic random forests [36], we choose a bagging approach [42] to build ensembles from Alg. 5.2. Namely, we use a bootstrapped sample of the noisy instances for each tree and draw a random feature subset of size $\lfloor \sqrt{d} \rfloor$ for each split. We combine the predictions of all trees by averaging their classification outputs and we tune the final decision threshold of the ensemble with Alg. 5.1 on out-of-bag, noisily labeled data.

5.3.3. F1 Score Optimization

Menon et al. [18] propose a technique to estimate several performance metrics under binary CCN, including the F_1 score. This general technique builds on a CCN-aware estimation of the clean positive rate and the clean negative rate of a classifier [131], which we detail in the following. This estimation allows us to optimize the clean F_1 score with data that is only noisily labeled. While Menon et al. have already conceived this optimization, it has not yet been evaluated empirically, to the best of our knowledge.

Algorithm 5.2 PK-CCN decision tree induction [7].

Input: A maximum depth $d \geq 0$, a noise rate $0 < p_- < 1$, and N noisily labeled instances $\{(x_i, \widehat{y}_i) : 1 \geq i \geq N\}$

Output: A trained decision tree

```
if $d > 0$ then
    $j^*, t^* \leftarrow \arg\max_{j,t} \max\{f_\alpha^{X_j \leq t}, f_\alpha^{X_j > t}\}$, see Eq. 5.10
    Split $L \leftarrow \{x_i : [x_i]_{j^*} \leq t^*\}$, $R \leftarrow \{x_i : [x_i]_{j^*} > t^*\}$
    Construct sub-trees on $L$ and $R$ with depth $d - 1$
    return a tree with a split $j^*, t^*$ and both sub-trees
else
    return a leaf node with the output $\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\widehat{y}_i = +1}$
end if
```

In particular, Menon et al. [18] estimate the clean F_1 score as

$$F_1(\theta;\, h, p_-, p_+) \;=\; \frac{2 \cdot \pi \cdot \mathrm{TPR}(\theta; p_-, p_+)}{\pi(1 + \mathrm{TPR}(\theta; p_-, p_+)) + (1 - \pi)(1 - \mathrm{TNR}(\theta; p_-, p_+))}, \tag{5.11}$$

where the clean true positive rate and the clean true negative rate are estimated as

$$\begin{aligned} \mathrm{TPR}(\theta; p_-, p_+) \;&=\; 1 - \frac{a(1 - \widehat{\mathrm{TNR}}(\theta)) + (1 - b)(1 - \widehat{\mathrm{TPR}}(\theta)) - a}{1 - a - b}, \\ \mathrm{TNR}(\theta; p_-, p_+) \;&=\; 1 - \frac{(1 - a)(1 - \widehat{\mathrm{TNR}}(\theta)) + b(1 - \widehat{\mathrm{TPR}}(\theta)) - b}{1 - a - b}. \end{aligned} \tag{5.12}$$

Here, $\widehat{\mathrm{TPR}}(\theta) = \mathbb{P}(h(x) > \theta \mid \widehat{Y} = +1)$ and $\widehat{\mathrm{TNR}}(\theta) = \mathbb{P}(h(x) \leq \theta \mid \widehat{Y} = -1)$ are the *noisy* rates of true positives and true negatives and $\widehat{\pi} = \mathbb{P}(\widehat{Y} = +1)$ is the noisy class prior. All of these probabilities can easily be estimated from data that is only noisily labeled. Moreover,

$$a \;=\; \frac{p_-(1 - p_+ - \widehat{\pi})}{\widehat{\pi}(1 - p_+ - p_-)}, \quad b \;=\; \frac{p_+(\widehat{\pi} - p_-)}{(1 - \widehat{\pi})(1 - p_+ - p_-)}, \quad \pi \;=\; \frac{\widehat{\pi} - p_-}{1 - p_+ - p_-}, \tag{5.13}$$

are constants which facilitate the notation in the equations above.

Recall from Sec. 2.2.2 that the F_1 score cannot be optimized analytically. However, it can be optimized through thresholded decision functions, e.g., through the two-step expected utility maximization from Alg. 2.1 [31], [32]. By itself, this algorithm is un-aware of CCN. However, the strict monotonicity of noisy and clean posteriors, see Prop. 5.1, allows us to adapt this algorithm to binary CCN: first, we learn a noisy classifier from noisily labeled training data; second, we choose a decision threshold that is optimal for the estimate of the clean F_1 score. We present this adaptation in Alg. 5.3.

Note that lines 1 and 2 of Alg. 5.3 estimate the noise rates p_- and p_+ under the anchor point assumption [141]. Therefore, the vanilla version of this algorithm addresses the CU-CCN setting from Def. 5.3, where none of the noise rates is known.

Algorithm 5.3 Two-step expected utility maximization [31], [32] of the F_1 score in spite of binary CCN [18].

```
Input: A scoring function $h : \mathcal{X} \to [0, 1]$ and $N$ noisily labeled instances $\{(x_i, \widehat{y}_i) : 1 \geq i \geq N\}$
Output: A decision threshold $\theta^* \in \mathbb{R}$
$\widehat{p}_- \leftarrow \min_{1 \geq i \geq N} h(x_i)$
$\widehat{p}_+ \leftarrow 1 - \max_{1 \geq i \geq N} h(x_i)$
$\theta^* \leftarrow \arg\max_{\theta \in \mathbb{R}} F_1(\theta;\, h, \widehat{p}_-, \widehat{p}_+)$, see Eq. 5.11
return $\theta^*$
```

If these rates are (partially) known, we can replace their estimates $\widehat{p}_-$ and $\widehat{p}_+$ with their known, ground-truth values. In this case, Alg. 5.3 becomes either a PK-CCN algorithm (if p_- is known; see Def. 5.5) or a CK-CCN algorithm (if both rates are known; see Def. 5.2). In order to assess the merits of our proposed PK-CCN setting, we evaluate all versions of Alg. 5.3: the vanilla CU-CCN version, the PK-CCN version with the ground-truth value for p_-, and the CK-CCN version with the ground-truth values for p_- and p_+.

Further note that Alg. 5.3 is a blueprint for the CCN-aware optimization of *any* performance metric that is optimized by a thresholded decision function. An overview of these performance metrics, all of which can be estimated under CCN according to Menon et al. [18], is given in Tab. 2.1. For imbalanced classification, we perceive the widely acknowledged F_1 score to be the most appropriate measure.

5.4. Validation on Conventional Imbalanced Data Sets

Our goal is to evaluate the merits of PK-CCN over the existing settings CU-CCN and CK-CCN. In this regard, we start with an extensive evaluation of CCN learning techniques on 27 conventional, imbalanced data sets. We load these data sets from the *imbalanced-learn* library[9] [145] and artificially inject different levels of CCN, which are listed in Fig. 5.3, in Tab. 5.1, and in Tab. 5.2. The first two noise configurations are designed by ourselves and the remaining four are by Natarajan et al. [130]. Our implementations of all methods and experiments is available online.[10]

5.4.1. Methodology

We estimate the performance of each CCN method in terms of the F_1 score and in terms of the f_α score from Def. 5.4. Each score is averaged over 20 repetitions of a 10-fold stratified cross validation, where we observe small standard deviations among all repetitions. We emphasize that our evaluation in terms of the F_1 score is only possible because we maintain some of the *clean* labels for evaluation. If no clean labels were available, a metric like our f_α score, which operates on *noisy* labels, was indispensable. We also emphasize

[9] `https://imbalanced-learn.org/stable/datasets/`

[10] `https://github.com/mirkobunse/pkccn`

that we do *not* validate in terms of the area under the receiver operating characteristic or in terms of balanced accuracy because these metrics are anyway *immune* to CCN [18] and therefore not informative. Due to the class imbalance, we do also not validate in terms of the standard, un-balanced accuracy.

In total, our results comprise 226 800 classification models. We employ random forest classifiers because they have a high predictive power and they allow us to tune decision thresholds *consistently*, on out-of-bag noisily labeled data.

5.4.2. Results

Fig. 5.3 compares the CCN learning performances in critical difference (CD) diagrams [55], [56], which we have introduced in Sec. 2.4.2. Recall that a CD diagram compares multiple machine learning methods across multiple data sets and that all *missing* connections indicate the presence of *statistically significant differences* between the competitors. Here, we display multiple CD diagrams, one per row, in a single plot.

The CD diagrams from Fig. 5.3 are complemented by Tab. 5.1, which reports the average F_1 scores across all data sets and repetitions. While these average values, by themselves, do not reflect how methods compare to each other, they complete the picture that is given by Fig. 5.3 in showing the magnitudes of performance differences.

The average f_α score of each method is displayed in Tab. 5.2. Unlike the F_1 score, this metric can be evaluated directly on the *noisy* labels if p_- is known. Fig. 5.4 further presents CD diagrams for this metric.

Discussion The perspectives from Fig. 5.3 and Tab. 5.1 demonstrate the merits of our proposed setting, PK-CCN: the advantage of PK-CCN methods, which consider a known p_-, over CU-CCN methods, which ignore this knowledge, is *statistically significant* and has a *considerable magnitude.* For instance, the PK-CCN version of the threshold by Menon et al. [18] achieves an overall average F_1 score of 0.425, which is way beyond the average CU-CCN score of 0.310.

The improvement of CK-CCN over PK-CCN has a much smaller magnitude (Menon: $F_1 = 0.433$ in CK-CCN versus $F_1 = 0.425$ in PK-CCN) and this improvement is *not* statistically significant in several noise configurations, namely in $(p_-, p_+) \in \{(0.5, 0.1)$, $(0.5, 0.25)$, $(0.4, 0.4)\}$. Hence, knowing at least one noise rate is of great importance for learning.

We recognize that our methods from Alg. 5.1 and Alg. 5.2 significantly loose against the Menon PK-CCN method in the noise configurations $(0.2, 0.2)$ and $(0.1, 0.3)$. We attribute this observation to the fact that Fig. 5.3 and Tab. 5.1 present an evaluation in terms of the F_1 score, which the Menon technique from Alg. 5.3 optimizes directly. If we replace the F_1 evaluation with an f_α evaluation, as presented in Tab. 5.2 and in Fig. 5.4, we see that our methods frequently win against the baselines. Evaluating in terms of the f_α score has

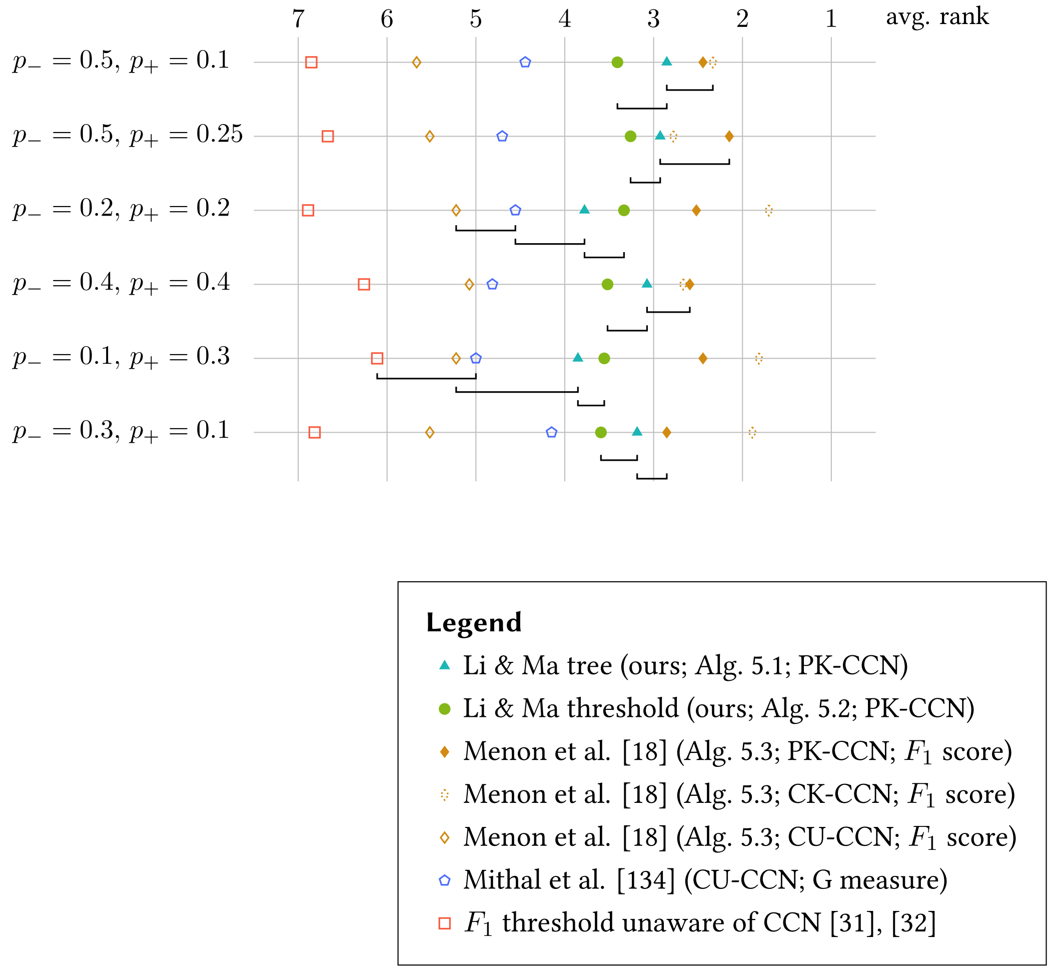

Figure 5.3.: Each row in this plot displays one critical difference diagram, see Sec. 2.4.2, which corresponds to one noise configuration. This overview summarizes a total of 226 800 random forest classifiers, a collection which consists of 7 CCN learning methods × 6 noise configurations × 27 data sets × 10 cross validation folds × 20 repetitions with random initialization. The x-coordinates indicate average ranks (lower is better), i.e., how often a method looses against its competitors. CCN methods are connected with horizontal bars if and only if a Holm-corrected Wilcoxon signed-rank test cannot significantly distinguish their pairwise performances, in terms of their cross-validated F_1 score, at a confidence level of 95%. Hence, all *missing* connections indicate statistically significant differences.

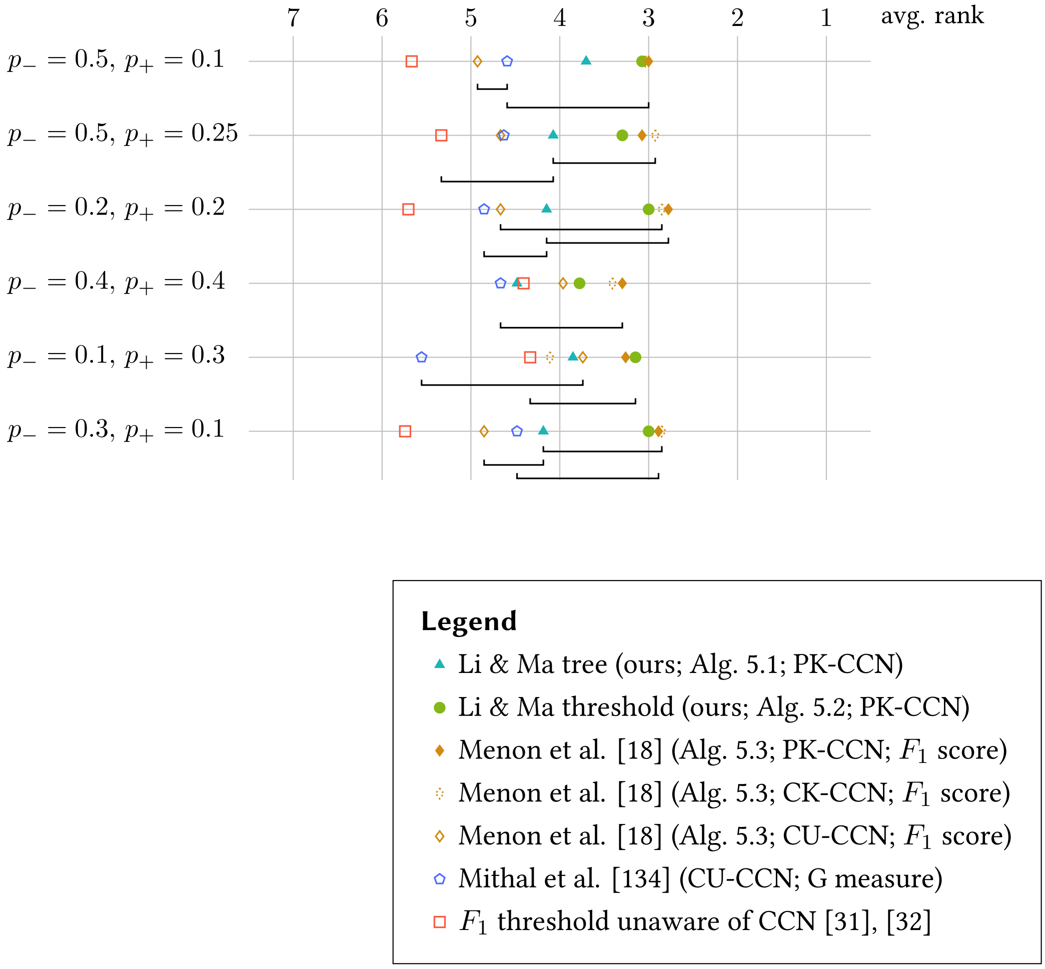

Figure 5.4.: Critical difference diagrams in terms of cross-validated f_α scores. This overview of our results is analogous to Fig. 5.3, but employs the f_α score instead of the F_1 score to determine the performance of each method.

Table 5.1.: Average F_1 scores (higher is better) over all 27 data sets and over 20 repetitions of a 10-fold cross validation.

method	$p_- = 0.5, p_+ = 0.1$	$p_- = 0.5, p_+ = 0.25$	$p_- = 0.2, p_+ = 0.2$	$p_- = 0.4, p_+ = 0.4$	$p_- = 0.1, p_+ = 0.3$	$p_- = 0.3, p_+ = 0.1$	overall average
Li&Ma tree	.414±.049	.330±.071	.482±.037	.281±.061	.497±.041	.482±.039	.414±.050
Li&Ma threshold	.408±.050	.327±.059	.485±.044	.280±.067	.499±.039	.480±.046	.413±.051
Menon PK/F_1	.417±.051	.336±.066	.498±.041	.294±.069	.512±.034	.490±.041	.425±.050
Menon CK/F_1	**.424±.049**	**.339±.062**	**.510±.035**	**.298±.068**	**.525±.032**	**.501±.037**	**.433±.047**
Menon CU/F_1	.262±.046	.184±.037	.408±.044	.172±.039	.466±.043	.371±.042	.310±.042
Mithal CU/G	.333±.051	.234±.057	.429±.051	.185±.047	.461±.052	.405±.045	.341±.050
CCN-unaware F_1	.112±.000	.112±.000	.169±.036	.112±.000	.402±.045	.114±.002	.170±.014

Table 5.2.: Average f_α scores (higher is better, see Def. 5.4) over all 27 data sets and over 20 repetitions of a 10-fold cross validation.

method	$p_- = 0.5, p_+ = 0.1$	$p_- = 0.5, p_+ = 0.25$	$p_- = 0.2, p_+ = 0.2$	$p_- = 0.4, p_+ = 0.4$	$p_- = 0.1, p_+ = 0.3$	$p_- = 0.3, p_+ = 0.1$	overall average
Li&Ma tree	6.99±1.12	3.54±1.17	12.44±1.16	2.68±0.97	14.75±1.21	11.54±0.97	8.66±1.10
Li&Ma threshold	**7.16±1.07**	3.85±1.17	**12.79±1.13**	2.90±1.15	**15.06±1.20**	**11.91±1.09**	**8.95±1.13**
Menon PK/F_1	6.99±1.09	3.83±1.20	12.59±1.19	2.90±1.17	14.83±1.20	11.66±1.13	8.80±1.16
Menon CK/F_1	7.16±1.09	**3.88±1.16**	12.76±1.09	**2.94±1.12**	14.90±1.21	11.89±1.05	8.92±1.12
Menon CU/F_1	5.57±1.27	2.70±1.07	12.01±1.23	2.27±1.16	14.74±1.23	10.66±1.21	7.99±1.20
Mithal CU/G	6.61±1.26	3.15±1.24	12.25±1.34	2.25±1.25	14.60±1.29	11.27±1.23	8.36±1.27
CCN-unaware F_1	2.79±0.90	1.75±0.86	7.02±1.43	1.65±1.01	12.87±1.38	4.88±1.10	5.16±1.11

the advantage that *no clean labels are required* during the evaluation. For some use cases of CCN learning, this advantage is critical.

5.5. Case Study: Detection of Cosmic Gamma Ray Sources

We now apply CCN learning methods to the detection of gamma ray sources in the data of the FACT telescope. For this purpose, as we have motivated in Sec. 3.4 and in Sec. 3.5, we have to predict the true "gamma" and "hadron" classes, only having access to the noisy "on" and "off" labels. Our goal is to achieve high f_α scores, see Def. 5.4, which indicate a high classification performance in terms of gamma ray source detections. In the following, we compare the same CCN learning methods that we have compared in the previous section, but have to exclude the privileged CK-CCN baseline because we have no knowledge of p_+. The standard operation mode of the FACT telescope [49] chooses observation areas $A_{\text{off}} = 5 \cdot A_{\text{on}}$, which yields $\alpha = 1/5$ or, due to the equivalence from Th. 5.1, $p_- = 1/6$.

We compare the CCN thresholding methods on random forests that are trained from two different data sources: first, on noisily labeled data; second, on labels that we obtain from the computationally expensive simulation from Sec. 3.6. The second data source yields the state-of-the-art classifier[4] for the FACT data. Notably, physicists already tune the decision threshold of this simulation-trained state-of-the-art classifier through Alg. 5.1 [144]. However, they do not use the noisy "on" and "off" annotations for the training of the underlying random forest.

We argue that doing so, however, would be preferable for two reasons. The first reason is the high computational cost of the simulation, which can be circumvented by learning directly from the real telescope. The second one is the fact that the simulated labels are not necessarily clean. In particular, any domain gap between the simulated and real telescope data would correspond to some form of label noise, which is hard to characterize. In contrast, the CCN noise of "on" and "off" annotations has the clear theoretical implication that optimal learning, solely from these annotations, is indeed feasible. Still, these potentials of CCN learning have not yet been considered, to the best of our knowledge, in the context of gamma hadron classification.

Due to the lack of ground-truth labels, we can report the real-world performance of our models only in terms of the test statistic f_α from Def. 5.4, not in terms of supervised measures like accuracy or the F_1 score. However, f_α is an appropriate and conventional performance metric for gamma hadron classification. Higher values indicate that the analysis pipeline is able to make a successful detection also with less data; therefore, the f_α values, for any fixed data set, allow us to compare the source detection *efficiency* of CCN learning methods. Hence, a comparison of f_α values is meaningful even beyond a mere reject / accept decision of the corresponding hypothesis test from Def. 5.4.

5.5.1. Data Selection and Methodology

We employ four data sets of the FACT telescope, which correspond to different gamma ray sources and observation periods. Of these four data sets, only the open Crab sample[3] is publicly available. The other three data sets are closed data, which we access through our interdisciplinary collaboration with astro-particle physicists.

Open Crab Sample Due to its public availability, this data sample allows us to conduct experiments that are completely reproducible with the code we publish. This public sample covers 17.7 hours of observation time between November 1st 2013 and November 6th 2013.

We detect the Crab nebula in this sample because we intend to produce f_α values that are comparable to other studies and to the outcome of the standard analysis[4] of the FACT telescope. This outcome, which we use as a considerably strong baseline, amounts to a value of $f_\alpha = 26.63$. We emphasize that this outcome is already the result of applying Alg. 5.1 with "on" and "off" annotations, despite using simulated data for the training of the underlying random forest.

Large Closed Crab Sample We also employ another, larger sample of Crab nebula observations of FACT. This sample amounts to 91.1 hours of observation time between October 1st 2013 and February 5th 2014 [22]. From this range, we exclude all data between November 1st 2013 and November 6th 2013, to obtain a set of data that is disjunct from the open Crab sample.

The purpose of this larger, disjunct set is to have a big, noisily labeled training set for our CCN learning methods. In particular, we train our methods on this noisy set and apply them to the three other samples. Unfortunately, the large Crab sample is not publicly available.

Markarian 421 Sample Due to the wobble mode from Fig. 3.2, physicists have ensured that no systematic bias towards any particular gamma ray source has to be expected from the data. Still, we intend to verify this expectation for CCN learning methods by applying these methods also to gamma ray sources other than the Crab nebula. To this end, we detect Markarian 421 in a closed data sample that covers observations between October 1st 2013 and February 5th 2014. Markarian 421 is a bright blazar, i.e., an active galactic nucleus that emits a jet of high-energy particles in the direction of planet Earth.

Markarian 501 Sample This source is another bright blazar, which we employ for the same reasons for which we also employ Markarian 421. This sample, which is also closed, covers observations between July 1st 2013 and December 31st 2013.

With these data sets, we conduct four experiments. First, we train CCN learning methods on the large, closed Crab sample and we apply the resulting gamma hadron classifiers to the open Crab sample. Hence, we obtain performance values that are comparable to the standard analysis baseline of $f_\alpha = 26.63$. Second and third, we train CCN learning

Table 5.3.: Average f_α scores (higher is better, see Def. 5.4) with standard deviations for the open Crab sample of the FACT telescope. We evaluate the threshold tuning methods with classifiers that are either trained with CCN noisy labels (left column) or with simulated labels (right column). In this table, all scores are achieved in 20 repetitions of a 6-fold cross validation. Thresholds are separately fitted to each training fold.

method	CCN labels (open; CV)	simulated labels (SOTA; CV)
Li & Ma tree (ours; Alg. 5.2; PK-CCN)	$\mathbf{23.91 \pm 0.34}$	–
Li & Ma threshold (ours; Alg. 5.1; PK-CCN)	25.39 ± 0.36	$\mathbf{26.09 \pm 0.23}$
Menon et al. [18] (Alg. 5.3; PK-CCN; F_1 score)	$\mathbf{25.72 \pm 0.38}$	25.14 ± 0.06
Menon et al. [18] (Alg. 5.3; CU-CCN; F_1 score)	$\mathbf{24.75 \pm 0.33}$	16.22 ± 0.04
Mithal et al. [134] (CU-CCN; G measure)	25.31 ± 0.42	$\mathbf{26.29 \pm 0.10}$
F_1 threshold unaware of CCN [31], [32]	14.39 ± 0.30	$\mathbf{16.37 \pm 0.07}$

methods again on the large, closed Crab sample, but we apply the resulting classifiers to the samples of Markarian 421 and Markarian 501. These experiments provide us with performance values that assess the generalization capabilities of CCN learning methods in terms of different gamma ray sources.

Fourth, we conduct an experiment that evaluates CCN learning performances solely on the open Crab sample, through cross validation. In this experiment, we employ each of the six nights as one cross validation fold, so that differences between nights, e.g., different atmospheric conditions, are appropriately handled. Here, the CCN decision thresholds are fitted in each fold separately. Therefore, the performance values must be expected to be lower than the state-of-the-art values, which stem from a threshold that is fitted once, to the entire open Crab sample. Still, this reproducible experiment provides us with a proper assessment of the general feasibility of CCN learning for source detection. All experiments are repeated 20 times.

5.5.2. Results

We begin our discussion with the fourth experiment, which employs solely the open Crab sample through cross validation. The results of this experiment are displayed in Tab. 5.3. We tune the decision thresholds of two random forests, one trained from "on" and "off" annotations (left column) and one trained from simulated labels (right column). Our Li & Ma tree from Alg. 5.2 is missing from the right column because this model has to be trained on noisy labels with $p_- > 0$; hence, we cannot use the simulated labels for training. Due to the cross validation, where thresholds are fitted to each training fold separately, the scores are generally below the state-of-the-art value of $f_\alpha = 26.63$. However, we see from this comparison that CCN approaches, in general, perform similar to the costly simulation-trained approach. We also see that a CCN-aware thresholding is necessary because a CCN-*unaware* thresholding (last row) exhibits inferior performances.

The relative merits of CCN-trained classifiers are more pronounced in the closed data experiments from Tabs. 5.4–5.6. In these experiments, each repetition and method fits a single threshold directly to the respective sample that is analysed, which is the typical approach in gamma hadron classification [144]. Hence, we are able to reproduce the state-of-the-art performance for the open Crab sample, $f_\alpha = 26.63$, in the combination of "simulated labels" and the "Li & Ma threshold" of Tab. 5.4. Despite this strong baseline, however, our CCN learning approach with the large Crab sample achieves an even better performance, with a value of $f_\alpha = 27.08$. Hence, an improved Crab nebula detection is possible with CCN-trained classifiers, which do not employ any simulated data. In Tab. 5.6, which reports the results for the Markarian 501 sample, our CCN methods even achieve a performance of $f_\alpha = 28.24$, which is considerably above the state-of-the-art performance of $f_\alpha = 26.34$ for this data sample. Generally, throughout Tabs. 5.4–5.6, all CCN classifiers outperform the corresponding simulation-trained classifiers. Not only verifies this finding our claim that CCN learning is indeed feasible for the detection of cosmic gamma ray sources, but it also suggests that CCN learning can even yield gamma hadron classifiers which are more effective source detectors than the simulation-trained state-of-the-art.

We attribute this finding to a combination of two aspects of gamma ray source detection: first, CCN theory states that optimal learning from "on" and "off" annotations is feasible; the only difficulty is to find an appropriate decision threshold. Astro-particle physicists, however, have already addressed this difficulty through Alg. 5.1 [144]. In contrast, any domain gap between the simulation and the real telescope can result in simulated data that exhibit some unknown form of label noise, for which the theoretical implications, in terms of learning, are not as clear as they are for CCN "on" and "off" annotations. In other words: we cannot rule out the possibility that the performance of a simulation-trained classifier is indeed limited by the quality of the simulation. The second aspect, which we witness particularly in Tabs. 5.4–5.6, is that large samples of training data particularly benefit the performances of CCN methods. To this end, CCN theory predicts that label noise increases the sample complexity. Our large, closed sample of the Crab nebula, however, appropriately addresses this issue.

In summary, our experiments demonstrate that highly performant gamma hadron classifiers can be learned solely from the "on" and "off" annotations of the real telescope data. No simulated data is required to train these classifiers, which enables astro-particle physicists to produce less simulated data in order to reduce the resource demands of their analyses. While small samples of "on" and "off" annotations already yield classifiers which perform on par with the state-of-the-art (see Tab. 5.3), larger samples can even outperform the state-of-the-art classifiers (see Tabs. 5.4–5.6). We have shown that "on" and "off" annotations are PK-CCN noisy labels with a strong theoretical foundation.

Table 5.4.: Average f_α scores (higher is better, see Def. 5.4) with standard deviations for the open Crab sample of the FACT telescope. These scores are achieved when a larger, closed, and disjunct sample of the Crab nebula [22] is used for the training of the CCN methods. Each threshold is fitted only once in each of 20 repetitions.

method	CCN labels (closed)	simulated labels (SOTA)
Li & Ma tree (ours; Alg. 5.2; PK-CCN)	$\mathbf{24.71 \pm 0.36}$	–
Li & Ma threshold (ours; Alg. 5.1; PK-CCN)	$\mathbf{27.08 \pm 0.22}$	26.63 ± 0.01
Menon et al. [18] (Alg. 5.3; PK-CCN; F_1 score)	$\mathbf{27.07 \pm 0.24}$	26.07 ± 0.73
Menon et al. [18] (Alg. 5.3; CU-CCN; F_1 score)	$\mathbf{26.88 \pm 0.23}$	16.32 ± 0.02
Mithal et al. [134] (CU-CCN; G measure)	$\mathbf{26.99 \pm 0.25}$	26.63 ± 0.03
F_1 threshold unaware of CCN [31], [32]	$\mathbf{19.56 \pm 1.01}$	16.35 ± 0.11

Table 5.5.: Average f_α scores (see Tab. 5.4) for the Markarian 421 data set.

method	CCN labels (closed)	simulated labels (SOTA)
Li & Ma tree (ours; Alg. 5.2; PK-CCN)	$\mathbf{22.37 \pm 0.31}$	–
Li & Ma threshold (ours; Alg. 5.1; PK-CCN)	$\mathbf{24.15 \pm 0.14}$	23.55 ± 0.03
Menon et al. [18] (Alg. 5.3; PK-CCN; F_1 score)	$\mathbf{24.00 \pm 0.18}$	22.51 ± 0.28
Menon et al. [18] (Alg. 5.3; CU-CCN; F_1 score)	$\mathbf{24.09 \pm 0.18}$	12.64 ± 0.11
Mithal et al. [134] (CU-CCN; G measure)	$\mathbf{24.14 \pm 0.17}$	23.53 ± 0.05
F_1 threshold unaware of CCN [31], [32]	$\mathbf{13.20 \pm 0.66}$	12.63 ± 0.08

Table 5.6.: Average f_α scores (see Tab. 5.4) for the Markarian 501 data set.

method	CCN labels (closed)	simulated labels (SOTA)
Li & Ma tree (ours; Alg. 5.2; PK-CCN)	$\mathbf{24.70 \pm 0.35}$	–
Li & Ma threshold (ours; Alg. 5.1; PK-CCN)	$\mathbf{28.24 \pm 0.12}$	26.34 ± 0.02
Menon et al. [18] (Alg. 5.3; PK-CCN; F_1 score)	$\mathbf{28.17 \pm 0.16}$	25.76 ± 0.11
Menon et al. [18] (Alg. 5.3; CU-CCN; F_1 score)	$\mathbf{27.79 \pm 0.26}$	14.23 ± 0.20
Mithal et al. [134] (CU-CCN; G measure)	$\mathbf{28.20 \pm 0.16}$	26.36 ± 0.00
F_1 threshold unaware of CCN [31], [32]	13.59 ± 0.26	$\mathbf{14.29 \pm 0.10}$

Table 5.7.: Average f_α scores for the open Crab sample with artificially removed "on" instances. Due to the removal, small values are desirable. In this table, all scores results from 20 repetitions of a 6-fold cross-validation of the open Crab sample, similar to Tab. 5.3.

method	CCN labels (open; CV)	simulated labels (SOTA; CV)
Li & Ma tree (ours; Alg. 5.2; PK-CCN)	0.55 ± 0.75	–
Li & Ma threshold (ours; Alg. 5.1; PK-CCN)	0.75 ± 0.76	0.00 ± 0.00
Menon et al. [18] (Alg. 5.3; PK-CCN; F_1 score)	0.95 ± 0.72	0.00 ± 0.00
Menon et al. [18] (Alg. 5.3; CU-CCN; F_1 score)	0.00 ± 0.00	0.00 ± 0.00
Mithal et al. [134] (CU-CCN; G measure)	0.11 ± 0.20	0.59 ± 0.25
F_1 threshold unaware of CCN [31], [32]	0.00 ± 0.00	0.00 ± 0.00

5.5.3. Interpretability

We have clarified in Sec. 5.3.1 that CCN learnability can only be tested through Def. 5.4 if we can access a data set that is both ground-truth labeled and noisily labeled. In astro-particle physics, no ground-truth labels exist for the real telescope data, so that interpreting f_α as an actual test statistic, strictly speaking, remains a heuristic assessment. Astro-particle physicists practically cope with this limitation by requiring huge certainties of $f_\alpha = 5$ and above, which corresponds to p-values of $2.87 \cdot 10^{-7}$ and below.

In the following experiment, we intend to assess the degree to which f_α can be interpreted as a test statistic about CCN learnability, in spite of its heuristic nature. In this experiment, we artificially remove all "on"-labeled instances from the data set. We then incorrectly label some of the "off" instances as being "on" instances. These fake re-assignments are meant to break the correspondence between clean and noisy labels. Hence, they should render CCN learning infeasible. In this situation, any proper hypothesis test on CCN learnability should *not* reject the null hypothesis, i.e., it should not falsely suggest that CCN learning was feasible.

As desired, Tab. 5.7 demonstrates that all CCN learning methods indeed produce f_α values close to zero. Practitioners of PK-CCN learning can tell from these values that no strong classifier can be expected to be learned from the given noisy labels. The fact that our Alg. 5.1 and Alg. 5.2 directly aim at maximizing the value of f_α does not result in a false outcome of the hypothesis test.

6. Active Class Selection

A class-conditional data generator, like the astro-particle simulation we have introduced in Sec. 3.6, produces labeled data in arbitrary proportions of classes. Active class selection (ACS) [19], [146] builds on the idea that this level of freedom can be leveraged to produce more cost-efficient training sets through an *active* acquisition of the data. This active acquisition consists in the production of small batches of data, with each batch being produced according to those proportions of classes that are expected to provide the largest performance improvement of a classifier that is trained on all previous batches. This idea is similar to the idea of active learning [147], but with a class-conditional data generator instead of an oracle.

The data of imaging atmospheric Cherenkov telescopes (IACTs), which we have introduced in Sec. 3.2, is never ground-truth labeled. Therefore, all labeled data is produced in sophisticated simulations of the detection mechanism that IACTs leverage. This mechanism is class-conditional: its input, be it in the real world or in the simulation, is a primary particle, which is characterized, among other properties, by its class. A user of the simulation must decide for the class proportions of the data before any data can be simulated. Other applications, where a training set with features and labels is not given but features can be acquired depending on the class proportions, are, for instance, the calibration of gas sensor arrays [19] and brain-computer interaction [148]–[150].

The goal of ACS is to optimize the acquisition of training data in such applications. This goal is pursued in a sequence of multiple acquisition steps, which are sketched in Fig. 6.1. In each step, a classifier is trained and evaluated on the data that have been acquired so far, starting from a small initial data set (i). Based on the classifier's performance, a data acquisition *strategy* is then allowed to choose the class proportions of the next acquisition step (ii). The class-conditional data generator realizes these proportions, i.e., it produces a batch of labeled data according to the choice of the strategy (iii). This batch is added to the labeled set of data, from which the classifier is trained and evaluated in all subsequent iterations (iv). The promise of such a sequential and informed data acquisition process is that the classifier can benefit in terms of data acquisition cost and performance, as compared to being trained with some predetermined proportions of classes that are not necessarily optimal.

However, despite these potential benefits, a free choice of training class proportions also puts the practical value of any ACS-trained model into question: is this model really appropriate for the class proportions that are handled during deployment? What if the deployment class proportions are uncertain or change over time? Indeed, astro-particle

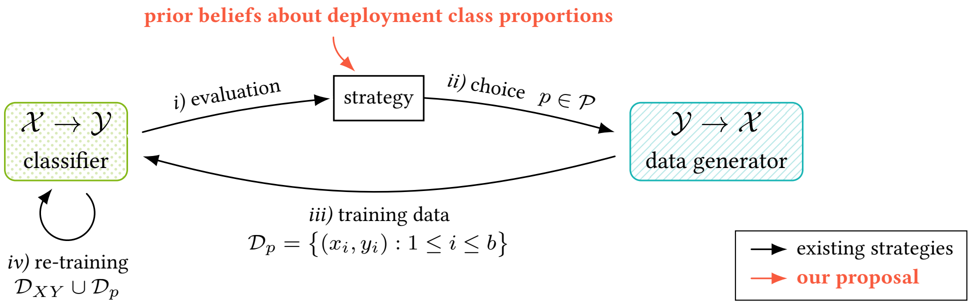
prior beliefs about deployment class proportions
i) evaluation
strategy
ii) choice $p \in \mathcal{P}$
$\mathcal{X} \rightarrow \mathcal{Y}$
classifier
$\mathcal{Y} \rightarrow \mathcal{X}$
data generator
iii) training data
$\mathcal{D}_p = \{(x_i, y_i) : 1 \leq i \leq b\}$
iv) re-training
$\mathcal{D}_{XY} \cup \mathcal{D}_p$
existing strategies
our proposal

Figure 6.1.: Active class selection optimizes class-conditioned data acquisition. Strategies for active class selection choose the label proportions of newly acquired training data. They are allowed to base their decisions on the performance of a classifier that is trained with all previously acquired data. We propose to incorporate prior beliefs, which can be uncertain, into the decision making. This figure is adapted from one of our publications [9].

physicists can estimate the ratio between gamma rays and hadrons only roughly, as being approximately $1 : 10^3$ or even $1 : 10^4$ [16].

We address these questions of model validity through a *certification* of ACS-trained classifiers. Specifically, our certificate declares a set of class proportions for which the certified model induces a performance gap between training and deployment that is small with a high probability. This declaration is theoretically justified by PAC bounds.

Our certificate also provides the basis for an ACS strategy that we develop. Most importantly, this strategy is able to incorporate uncertainties about the deployment class proportions. This incorporation aims at learning classifiers that are valid in all deployment scenarios that might have to be addressed.

This chapter is structured as follows: In Sec. 6.1, we introduce the existing heuristic strategies for data acquisition in ACS, before we survey existing experimental results in Sec. 6.2 to manage our expectations on the topic. With these foundations, we turn to two theoretical perspectives from which we examine ACS. The first perspective, to which we turn in Sec. 6.3, is from information theory and provides us with valuable intuitions about the quality of ACS strategies. The second perspective, to which we turn in Sec. 6.4, is from learning theory and provides us with a quantitative assessment of ACS in terms of PAC bounds. From these bounds, we develop our certificate of model robustness in ACS. In Sec. 6.5, we develop our strategy for uncertainty-aware data acquisition in ACS, which is also based on our PAC bounds.

6.1. Existing Heuristics for Active Class Selection

The term “active class selection” was coined in 2007, when Lomasky et al. [19] introduced five heuristics for a class-dependent data acquisition with a gas sensor array. Before their seminal work, some general rules of thumb were already known for the determination of training class proportions. We introduce these rules of thumb and the active heuristic strategies in the following sub-sections.

6.1.1. Rules of Thumb

Before Lomasky et al. proposed an active, class-conditional acquisition of training data, systematic empirical studies [48], [151]–[153] confirmed two rules of thumb for data acquisition in binary classification: first, if the classifier performance is measured in accuracy, sampling in the natural proportions of classes leads to high performance scores. Second, if the performance metric is the area under the receiver operating characteristic (AUROC), sampling with balanced classes is more advantageous. The second observation can be explained by the fact that AUROC is akin to an average performance over all possible class weights. In contrast to Lomasky's proposal of an active and informed data acquisition, these prior works do not attempt to acquire data in multiple iterations, except for the beam search heuristic by Weiss and Provost [48].

6.1.2. Active Strategies

Lomasky et al. [19] consider five heuristics for the active acquisition of data from a class-conditional data generator. Their heuristics are based on a utility measure, which the current classifier $h : \mathcal{X} \to \mathbb{R}^C$ exhibits for each class $y \in \mathcal{Y}$. In particular, each of the five heuristics samples the class y according to the weight

$$w(y) \;=\; \frac{u(y)}{\sum_{y' \subseteq \mathcal{Y}} u(y')}, \tag{6.1}$$

where $u : \mathcal{Y} \to \mathbb{R}$ is a utility heuristic that might or might not depend on the current prediction model h. The five heuristics by Lomasky et al. are the following:

Uniform Assign the same utility $u(y) = 1$ to all classes.

Proportional Sample with the class proportions that also occur during deployment, i.e., $u(y) = \mathbb{P}_{\mathcal{T}}(Y = y)$. While the proportional strategy usually performs highly competitively in terms of accuracy, it is sometimes beaten by one of the other heuristics.

Inverse Sample by the inverse class-wise accuracy, $u(y) = \text{Accuracy}_h(y)^{-1}$. This strategy is motivated by the assumption that a low accuracy is exclusively caused by an insufficient amount of training data.

Improvement Similar to the "inverse" method, but based on the improvement of class-wise accuracy, $u(y) = (\text{Accuracy}_h(y) - \text{LastAccuracy}_h(y))^{-1}$, during the last iteration. This strategy assumes that classes with a "stable" accuracy will remain stable in future iterations.

Redistriction Count the number of examples $u(y) = m_h^y$, for which the prediction changed during the last iteration. The idea behind this strategy is that examples close to a decision boundary may be more informative than others. Classes for which predictions frequently change are assumed to contain many examples close to decision boundaries.

Several authors experimented with these heuristics. In brain-computer interaction, for instance, the plain use of the above heuristics [148], [154] and their combination with other learning tasks, i.e., with collaborative filtering [149] and with instance weighting [150], was considered. These studies report that the combinations of tasks outperform a plain use of ACS heuristics in their areas of application.

The most notable experiments have been conducted by Kottke et al. [146], who proposed another, original heuristic for ACS. This "ACS-PAL" heuristic embeds strategies for active learning in the ACS process. This embedding aggregates the class-wise utilities,

$$u(y) \;=\; \frac{1}{m_y} \sum_{i=1}^{m_y} u_{\mathrm{AL}}(x_i), \tag{6.2}$$

from utility assessments $u_{\mathrm{AL}}(x_i)$ that an active learning strategy evaluates for individual pseudo-examples x_i. These pseudo-examples are produced by some generative model $\mathcal{Y} \rightarrow \mathcal{X}$ that is trained, together with the classifier, from ACS-generated data.

Our theoretical analysis of ACS, which we detail in Sections 6.3 and 6.4, reveals that the "proportional" strategy is actually more than a heuristic; this strategy is indeed *optimal* in the large sample limit. However, it requires precise knowledge of the deployment class proportions $\mathbb{P}_{\mathcal{T}}(Y = y)$, which practitioners might not be able to provide. Contrastingly, all other strategies are entirely oblivious to the deployment proportions; they solely focus on different perceptions of class-wise difficulties. We are not aware of an ACS strategy that allows for uncertain deployment class proportions; therefore, we develop a novel strategy for this setting in Sec. 6.5.

6.2. What Can We Expect from Active Class Selection?

Before we turn to a theoretical analysis of ACS, we intend to manage our expectations about the potential benefits of the existing ACS methods. To this end, let us narrow the question that is raised by the title of this section: given that new training data can be acquired from a class-conditional data generator, can we expect ACS to make the optimal use of a limited data generation budget?

In the following sub-sections, we address this question from two angles: first, we survey the existing experiments on ACS to assess the relative merits of the proposed ACS strategies. Second, we compare the performance of ACS to the performance of active learning, a different but related learning task.

Table 6.1.: The winning ACS strategies for each evaluated data set. Methods that clearly win against their competitors, in terms of a subjective inspection of the published plots and tables, are denoted with a check mark (✓). Methods that clearly lose are denoted with a cross mark (✗). In some experiments, the "proportional" and "uniform" strategies are equivalent (≡) due to a uniform class distribution in the test set. Multiple values for the number of classes and the number of examples indicate sub-sampling. An infinity sign (∞) indicates synthetic data, from which arbitrary amounts can be sampled. This table is an extended version of a survey that we have published earlier [10].

data set	no. classes	no. examples	PAL-ACS [146]	beam search [48]	redistricting [19]	improvement [19]	inverse [19]	uniform [19]	proportional [19]
3clusters [146]	3	60 / ∞	✓		✗		✗	✗	≡
spirals [146]	3	120 / ∞	✓		✗		✗	✗	≡
bars [146]	3	120 / ∞	✓		✓		✓	✓	≡
vehicle [155]	4	80 / 946	✓		✗		✗	✓	≡
vertebral [155]	3	60 / 310	✓		✗		✗	✓	≡
yeast [155]	5 / 8	60 / 1150	✓		✗		✗	✓	≡
land cover [156]	11	≈ 28 000			✓	✗	✗	✗	✗
artificial nose [19]	8	≈ 1 250			✗	✗	✗	✗	✓
phone [48]	2	652 557		✓					✗
blackjack [157]	2	15 000		✓					✗
weather [157]	2	5597		✓					✓
adult [155]	2	48 842		✓					✗
kr-vs-kp [155]	2	3196		✓					✗
covertype [155]	2 / 7	581 102		✓					✓
letter-recognition [155]	2 / 26	20 000		✓					✓

6.2.1. Survey of Existing Experimental Results

Tab. 6.1 summarizes the results that have been reported for the existing heuristic methods, which we have presented in Sec. 6.1. The columns of this table indicate whether a method clearly outperforms its competitors (✓) or not (✗). Missing values indicate that a method has not been evaluated for the particular data set. Unfortunately, the qualification of a "clear winner" must remain somewhat subjective because the precise values of the results are not published. In particular, we declare a method as a winner when the plots in the corresponding paper allow for a clear, visual distinction of this method from its competitors. We declare multiple methods as winners whenever a single winner cannot be made out from the published plots and tables.

A central observation to make from Tab. 6.1 is that the random strategies "proportional" and "uniform" perform highly competitive. In this survey, they win on 8 out of 15 data sets. Computationally, these strategies come for free, whereas the competing *active* strategies imply a certain computational overhead. Hence, the practical merits of the existing *active* ACS heuristics appear to be limited. Beyond these experimental findings, more general concerns about the applicability of active sampling are being discussed [158]. We note, however, that the exact class proportions, which are required by the "proportional" strategy, are often not known at training time. In astro-particle physics, these proportions can only be estimated with considerable uncertainties, which no existing strategy can take into consideration.

6.2.2. Active Class Selection Versus Active Learning

Active learning (AL) is based on the assumption of an *oracle*, which can assign labels to arbitrarily chosen data examples. Its goal, to acquire a cost-efficient training set for machine learning, is the same goal that ACS pursues through class-conditional data acquisition. In AL, this goal is pursued differently, through an active choice of examples that are to be labeled by the oracle. Since every labeling induces some cost, AL has to decide which individual examples exhibit the largest utility for supervised learning.

Conceptually, ACS and AL differ in two aspects. First, their assumptions differ: while ACS assumes a class-conditional data generator $\mathcal{Y} \rightarrow \mathcal{X}$, the assumption of AL is an oracle $\mathcal{X} \rightarrow \mathcal{Y}$, which produces labeled data in the other conditional direction. Second, ACS requires some utility metric $u_{\mathrm{ACS}} : \mathcal{Y} \rightarrow \mathbb{R}$, which supports the choice of *classes* for which additional data is to be generated. We have introduced the existing utility metrics for ACS in Sec. 6.1. AL requires a different utility function $u_{\mathrm{AL}} : \mathcal{X} \rightarrow \mathbb{R}$, which supports the choice of data *examples* that are to be labeled. Therefore, we cannot apply AL techniques in ACS and vice versa [11]. At least, an embedding of AL techniques in ACS is complicated by the requirement of a generative model, as proposed in the "PAL-ACS" strategy from Eq. 6.2.

In the context of simulation data mining, ACS and AL correspond to different simulation strategies. Either, a simulation may generate feature vectors from input labels ($\mathcal{Y} \rightarrow \mathcal{X}$) [16], [159] or it may generate labels from input features ($\mathcal{X} \rightarrow \mathcal{Y}$) [25], [26], [84], [85], [87], [160]. Here, the first strategy establishes the simulation as a class-conditional data generator for ACS and the second one establishes the simulation as an oracle for AL. Both settings aim at minimizing the computational resources that are required by the simulation. However, we cannot employ an ACS strategy to optimize an $\mathcal{X} \rightarrow \mathcal{Y}$ simulation and we cannot employ an AL strategy to optimize a $\mathcal{Y} \rightarrow \mathcal{X}$ simulation [11].

Despite these fundamental, conceptual differences, we intend to assess how the empirical performances of ACS and AL compare to each other. To this end, we extend some of the experiments by Kottke et al. [146] with an evaluation of an AL strategy, the probabilistic active learning (PAL) [161] that is also embedded in "PAL-ACS".

As another extension of these experiments, we consider a strategy that is optimal for the artificial "spirals" data set. Its optimality stems from background knowledge about the data generating process of this data, which allows us to achieve 100% accuracy on one of

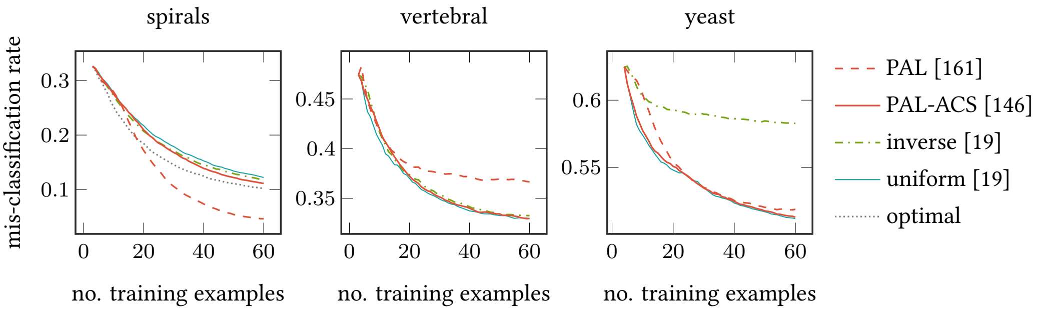

Figure 6.2.: The learning curves of ACS strategies and the AL strategy "PAL". This figure is adapted from one of our publications [10].

the classes just by sampling a single instance of this class. In fact, the artificial "spirals" data set contains one "easy" class that is linearly separable from the other two classes. In between these other classes, however, a complex decision boundary is defined, such that many examples of these two classes have to be generated. Even though this optimal strategy does not generalize to any other data set, it shows how well an ACS strategy could potentially perform on the "spirals" data.

Fig. 6.2 presents the results of these extensions, specifically the average error among 500 trials. The optimal strategy for the "spirals" data indicates that there is still room for improving ACS methods. However, PAL is even better than that, indicating that even the best ACS strategy is not guaranteed to outperform AL strategies. These observations cannot be made with the two UCI data sets, where no ACS strategy is a clear winner. We deem this finding consistent with the original experiments. Notably, the AL strategy performs worse than ACS on the "vertebral" data set. We conclude that the identification of relevant examples (AL) is not necessarily easier than, but instead considerably different from the identification of relevant classes (ACS).

6.3. Active Class Selection Constitutes a Domain Gap

In Tab. 6.1, we have seen that active strategies for ACS are often just as good as sampling randomly with respect to the natural class proportions. Still, research on ACS tries hard to beat this simple baseline. Seeking to explain this phenomenon, we examine the theoretical background of ACS from the viewpoint of information theory. Due to this viewpoint, our analysis is limited to the large sample limit $m \to \infty$, which, however, is continuously approached by ACS through the iterative acquisition of data. We drop this limitation in Sec. 6.4, where we establish a more sophisticated analysis that is based on learning theory and on PAC bounds.

The information-theoretic viewpoint taken here yields an upper bound, which depends on the class proportions chosen by an ACS strategy, of the error that a consistent Bayesian classifier induces. Namely, this bound depends i) on the Kullback-Leibler (KL) divergence between the deployment class proportions and the class proportions that are chosen by an ACS strategy; and ii) on the error of the generative part of the classifier. This generative error diminishes as the size of the training set increases. Therefore, the "proportional" strategy becomes increasingly superior to other ACS strategies while more and more training data is acquired. Hence, these theoretical findings enrich the common observation of a competitive "proportional" strategy by revealing that the relative performance of this strategy has the tendency to *increase with the size of the training set.*

In the following, we employ the prior probability shift assumption from Def. 2.3. Under this assumption, the training set is drawn from $\mathbb{P}(X \mid Y) \cdot \mathbb{P}_{\mathcal{S}}(Y)$, where $\mathbb{P}_{\mathcal{S}}(Y)$ is the class distribution that is chosen by an ACS strategy, and the deployment data is drawn from $\mathbb{P}(X \mid Y) \cdot \mathbb{P}_{\mathcal{T}}(Y)$, where $\mathbb{P}_{\mathcal{T}}(Y)$ can be different from $\mathbb{P}_{\mathcal{S}}(Y)$. In this section, our concept of optimality is meant to study the impact of ACS in a theoretically sound, yet simple way. To this end, we focus on classifiers that intend to find the most accurate estimate $\mathbb{P}_{\mathcal{D}}(Y \mid X)$, learned from an ACS-generated training set $\mathcal{D}_{XY}$, of the true underlying conditional density $\mathbb{P}_{\mathcal{T}}(Y \mid X)$. We formalize this intent as a minimization of the KL divergence

$$\mathrm{KL}^{\mathcal{D}}_{Y|X} \;=\; \mathrm{KL}\left(\mathbb{P}_{\mathcal{T}}(Y \mid X) \mid\mid \mathbb{P}_{\mathcal{D}}(Y \mid X)\right) \;\overset{!}{\to}\; \min, \tag{6.3}$$

where the KL divergence between probability density functions is defined in Def. 2.4. The central question of the following analysis is how ACS affects $\mathrm{KL}^{\mathcal{D}}_{Y|X}$.

We approach this question in two steps. First, we study the effect of ACS on the distribution of the training data, ignoring the potential effects of estimating $\mathbb{P}_{\mathcal{T}}(Y \mid X)$ from a finite amount of data. These effects are then introduced in the second step, where we incorporate a Bayesian classifier into our analysis. For now, the limitation to Bayesian classifiers yields a simple analysis in terms of KL divergences. In Sec. 6.4, we drop this narrow limitation in favor of a more general analysis.

6.3.1. Effects on the Training Data Distribution

In this first step of our analysis, we study how ACS affects the distribution $\mathbb{P}_{\mathcal{S}}(Y \mid X)$ of the training set. To this end, we assess a KL divergence

$$\mathrm{KL}^{\mathcal{S}}_{Y|X} \;=\; \mathrm{KL}\left(\mathbb{P}_{\mathcal{T}}(Y \mid X) \mid\mid \mathbb{P}_{\mathcal{S}}(Y \mid X)\right) \tag{6.4}$$

where we emphasize that $\mathrm{KL}^{\mathcal{S}}_{Y|X}$ is different from Eq. 6.3 because it considers only the training set distribution $\mathbb{P}_{\mathcal{S}}$ but no estimation from a finite data set $\mathcal{D}_{XY}$. The importance of studying $\mathrm{KL}^{\mathcal{S}}_{Y|X}$ stems from the existence of *consistent* estimators of $\mathbb{P}_{\mathcal{S}}(Y \mid X)$, e.g., the proper scoring rules from Def. 2.2. Only if $\mathrm{KL}^{\mathcal{S}}_{Y|X} = 0$ will a learning algorithm, which is consistent with respect to $\mathbb{P}_{\mathcal{S}}(Y \mid X)$, be consistent also with respect to $\mathbb{P}_{\mathcal{T}}(Y \mid X)$. Otherwise, standard techniques for consistent machine learning can fail as a consequence of the $\mathrm{KL}^{\mathcal{S}}_{Y|X}$ divergence that ACS has induced.

In ACS, we allow some strategy to actively choose $\mathbb{P}_{\mathcal{S}}(Y)$, the class proportions of the training set. Through this choice, ACS can induce a discrepancy between $\mathbb{P}_{\mathcal{S}}(Y)$ and the class proportions of the deployment data, $\mathbb{P}_{\mathcal{T}}(Y)$. We assess this discrepancy, like before, in terms of a KL divergence

$$\mathrm{KL}_Y^{\mathcal{S}} \;=\; \mathrm{KL}\left(\mathbb{P}_{\mathcal{T}}(Y) \;||\; \mathbb{P}_{\mathcal{S}}(Y)\right).$$

The following theorem bounds the training set divergence $\mathrm{KL}_{Y|X}^{\mathcal{S}}$, which is induced by ACS through $\mathrm{KL}_Y^{\mathcal{S}}$, in terms of the divergences $\mathrm{KL}_Y^{\mathcal{S}}$ and $\mathrm{KL}_X^{\mathcal{S}}$.

Theorem 6.1. [8] The KL divergence $\mathrm{KL}_{Y|X}^{\mathcal{S}}$, which is exhibited by an ACS-generated training set, is bounded above by the divergence $\mathrm{KL}_Y^{\mathcal{S}}$ between the ACS-chosen class proportions and the deployment class proportions, i.e.,

$$\mathrm{KL}_{Y|X}^{\mathcal{S}} \;=\; \mathrm{KL}_Y^{\mathcal{S}} - \mathrm{KL}_X^{\mathcal{S}} \;\leq\; \mathrm{KL}_Y^{\mathcal{S}}.$$

Proof. We apply the chain rule from Prop. 2.2.ii) to see that

$$\begin{aligned} \mathrm{KL}_{X,Y}^{\mathcal{S}} \;&=\; \mathrm{KL}_{Y|X}^{\mathcal{S}} + \mathrm{KL}_X^{\mathcal{S}} \;=\; \mathrm{KL}_{X|Y}^{\mathcal{S}} + \mathrm{KL}_Y^{\mathcal{S}} \\ \Rightarrow \quad \mathrm{KL}_{Y|X}^{\mathcal{S}} \;&=\; \mathrm{KL}_{X|Y}^{\mathcal{S}} + \mathrm{KL}_Y^{\mathcal{S}} - \mathrm{KL}_X^{\mathcal{S}}. \end{aligned}$$

Further making the assumption from Def. 2.3, i.e., $\mathbb{P}_{\mathcal{S}}(X \mid Y) = \mathbb{P}(X \mid Y)$, means under Prop. 2.2.i) that

$$\mathrm{KL}_{X|Y}^{\mathcal{S}} \;=\; \mathrm{KL}\left(\mathbb{P}(X \mid Y) \;||\; \mathbb{P}_{\mathcal{S}}(X \mid Y)\right) \;=\; 0,$$

which yields the claim. □

Consequently, if we want to minimize $\mathrm{KL}_{Y|X}^{\mathcal{S}}$, we can instead minimize its upper bound $\mathrm{KL}_Y^{\mathcal{S}}$. This approach is precisely the idea of the "proportional" strategy, which always acquires data according to $\mathbb{P}_{\mathcal{T}}(Y)$. Due to the theoretical finding from Th. 6.1, this strategy is more than a heuristic: it is indeed optimal in the large sample limit.

Quite interestingly, Th. 6.1 further implies that an increase in $\mathrm{KL}_X^{\mathcal{S}}$ can decrease $\mathrm{KL}_{Y|X}^{\mathcal{S}}$. While this finding might appear counter-intuitive at first, it can be traced back to the correlation between X and Y. For instance, the edge case $\mathrm{KL}_{Y|X}^{\mathcal{S}} = \mathrm{KL}_Y^{\mathcal{S}}$ only occurs if $\mathrm{KL}_X^{\mathcal{S}} = 0$, which in turn can only happen if a change in $\mathbb{P}_{\mathcal{S}}(Y)$ has no effect on $\mathbb{P}_{\mathcal{S}}(X)$, or if $\mathrm{KL}_Y^{\mathcal{S}} = 0$ anyways. If changing $\mathbb{P}_{\mathcal{S}}(Y)$ has no effect $\mathbb{P}_{\mathcal{S}}(X)$, then X and Y must be uncorrelated and we can not expect to learn any meaningful relation between them. Conversely, if $\mathrm{KL}_X^{\mathcal{S}} = \mathrm{KL}_Y^{\mathcal{S}}$, so that we obtain $\mathrm{KL}_{Y|X}^{\mathcal{S}} = 0$, we can conclude that X and Y must be fully correlated. Generally speaking, the higher the correlation is between X and Y, the easier is the classification task, even despite mistaken class proportions. We also recognize the following:

Corollary 6.1. [8] The divergence of an ACS-generated training set with respect to X is always lower than or equal to its divergence with respect to Y.

$$\mathrm{KL}_X^{\mathcal{S}} \;\leq\; \mathrm{KL}_Y^{\mathcal{S}}$$

Table 6.2.: The data sets from Fig.6.3.

dataset	$\mathbb{P}_{\mathcal{T}}(Y)$	no. features	no. NCA components	no. examples
3clusters [146]	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	2	*	†
spirals [146]	$(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$	2	*	†
vertebral [155]	$(.32, .19, .48)$	6	2	310
yeast [155]	$(.44, .41, .15)$	6	2	1055
vehicle [155]	$(.26, .24, .26, .25)$	18	3	846

*no NCA is performed
†arbitrary number of examples (synthetic data set; we use 1200)

Proof. This finding follows directly from Theorem 6.1 and from $\mathrm{KL}^{\mathcal{S}}_{Y|X} \geq 0$, which is stated by Prop. 2.2.i). □

In order to assess the tightness of the upper bound of Th. 6.1, we now evaluate $\mathrm{KL}^{\mathcal{S}}_{Y}$ and $\mathrm{KL}^{\mathcal{S}}_{X}$ empirically. In this regard, we alter the underlying class proportions $\mathbb{P}_{\mathcal{S}}(Y)$ of the training set by changing the probability of each class separately; for each class y, we scan through several probabilities $\mathbb{P}_{\mathcal{S}}(Y = y) \in \{0.05, 0.1, \ldots 0.95\}$ and scale the probabilities of the other classes $y' \neq y$ so that $\mathbb{P}_{\mathcal{S}}(Y)$ remains a valid density.

To compute $\mathrm{KL}^{\mathcal{S}}_{X}$, we must be able to evaluate $\mathbb{P}_{\mathcal{S}}(X)$ and $\mathbb{P}_{\mathcal{T}}(X)$ at arbitrary positions $x \in \mathcal{X}$. We do so with a kernel density estimate (KDE), which we fit for each class on the full data set. In line with the prior probability shift assumption from Def. 2.3, we use the same KDEs for both distributions, $\mathbb{P}_{\mathcal{S}}$ and $\mathbb{P}_{\mathcal{T}}$. This way, we can compute $\mathbb{P}_{\mathcal{S}}(X)$ and $\mathbb{P}_{\mathcal{T}}(X)$ through a marginalization over $\mathcal{Y}$. On the UCI data sets, which we employ here, we enable proper KDE estimations through a neighborhood component analysis (NCA) [162], a technique for dimensionality reduction. An overview of the experimental data and the cross-validated number of NCA components is given in Tab. 6.2

Fig. 6.3 shows the results of this experiment. We see that a considerable range of mistaken training class proportions induces a tight bound $\mathrm{KL}^{\mathcal{S}}_{X} \approx \mathrm{KL}^{\mathcal{S}}_{Y}$, which leads to negligible divergences $\mathrm{KL}^{\mathcal{S}}_{Y|X} \approx 0$. In this range, an optimal classifier can still be learned despite mistaken class proportions, which an ACS strategy other than the "proportional" strategy has produced. The width of this range depends on the data set, which is characterized here as $\mathbb{P}(X \mid Y)$.

6.3.2. Effects on Bayesian Classifier Induction

In order to analyze the effect of an ACS-chosen $\mathbb{P}_{\mathcal{S}}(Y)$ on $\mathrm{KL}^{\mathcal{D}}_{Y|X}$, we must relate the conditional direction $X \to Y$, which is modeled by the classifier, to the converse conditional direction $Y \to X$, which is implemented by the data generating process. Since such a

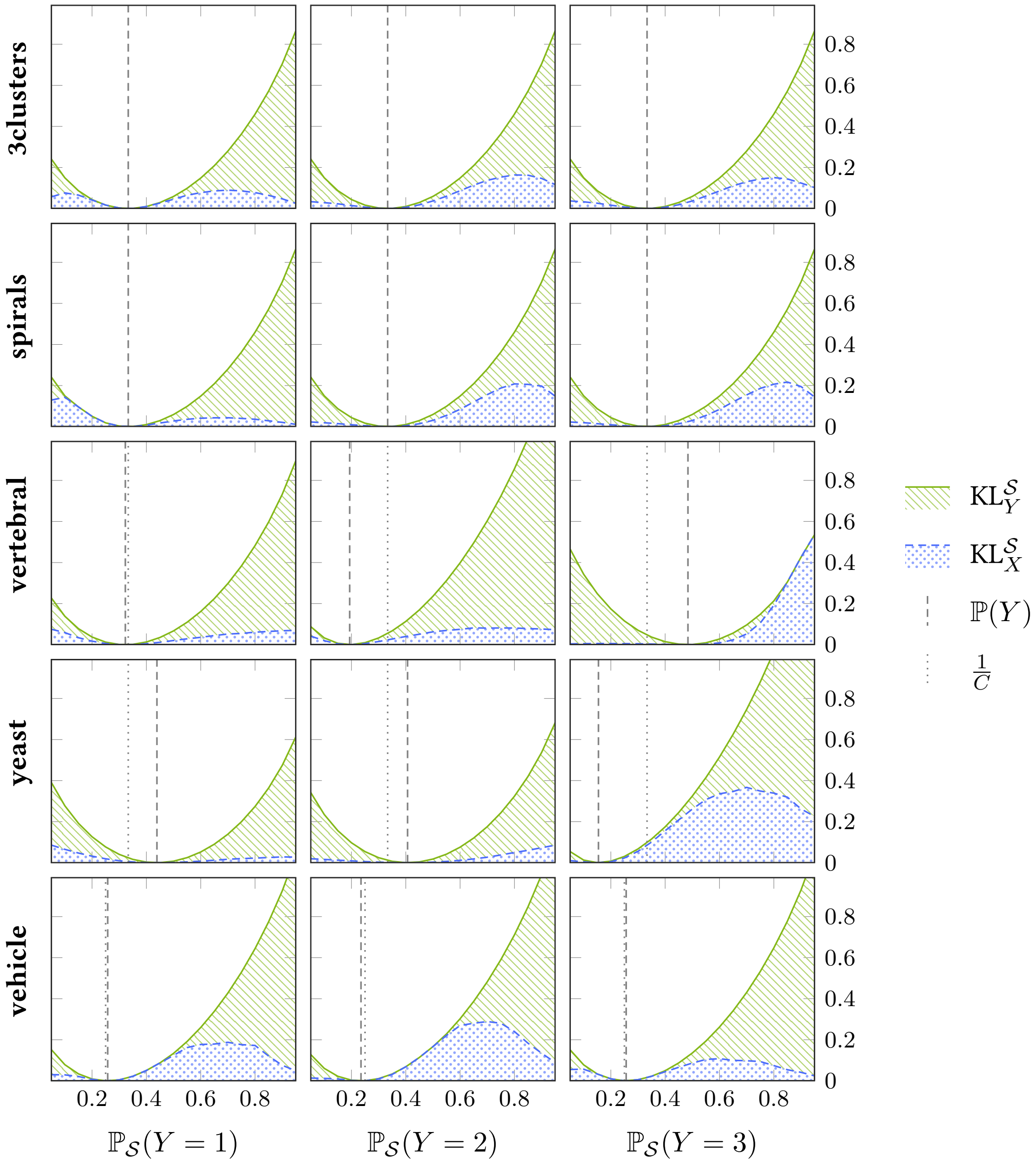

Figure 6.3.: The divergence $\mathrm{KL}^{\mathcal{S}}_{Y|X} = \mathrm{KL}^{\mathcal{S}}_{Y} - \mathrm{KL}^{\mathcal{S}}_{X}$ for several data sets, i.e., for different $\mathbb{P}(X \mid Y)$. According to Th. 6.1, regions with $\mathrm{KL}^{\mathcal{S}}_{X} \approx \mathrm{KL}^{\mathcal{S}}_{Y}$ indicate that the corresponding class proportions will not impair the performance of the classifier. Each row in this group plot presents one data set and each column presents another class $y \in \{1, 2, 3\}$, for which the probability $\mathbb{P}_{\mathcal{S}}(Y = y)$ is altered. In each cell, the values of $\mathrm{KL}^{\mathcal{S}}_{X}$ and $\mathrm{KL}^{\mathcal{S}}_{Y}$ are plotted on the vertical axis. They stem from the altered probability values $\mathbb{P}_{\mathcal{S}}(Y = y)$, which are displayed on the horizontal axis of each cell. As a reference, the natural class proportions and the uniform class proportions are displayed in vertical lines. This figure is adapted from one of our publications [8].

relation is prevalent in Bayesian classifiers, we will focus our analysis on this particular family of classification models. These classifiers make probabilistic predictions

$$\mathbb{P}_{\mathcal{D}}(Y \mid X) \;=\; \frac{\mathbb{P}_{\mathcal{D}}(X \mid Y) \cdot \widehat{\mathbb{P}}(Y)}{\int_{\mathcal{Y}} \mathbb{P}_{\mathcal{D}}(X \mid Y = y) \cdot \widehat{\mathbb{P}}(Y = y) \; \mathrm{d}y}$$

as according to Bayes' rule. We see that Bayesian classifiers learn a generative model $\mathbb{P}_{\mathcal{D}}(X \mid Y)$, which is weighted by some prior $\widehat{\mathbb{P}}(Y)$. Assuming consistency of the generative part, we obtain the following result:

Theorem 6.2. [8] Consider a Bayesian classifier that employs a consistent generative model, i.e., let there be some $N \in \mathbb{N}$ such that

$$\mathrm{KL}^{\mathcal{D}}_{X|Y} \;=\; \mathrm{KL}\left(\mathbb{P}_{\mathcal{S}}(X \mid Y) \;||\; \mathbb{P}_{\mathcal{D}}(X \mid Y)\right) \;\leq\; \varepsilon_{X|Y} \quad \forall\, m > N,$$

where $\mathbb{P}_{\mathcal{D}}(X \mid Y)$ is the model that is learned from a training set $\mathcal{D}_{XY}$ with m examples. The error of such a classifier is bounded above by the error of its assumed class proportions,

$$\mathrm{KL}_{\widehat{Y}} \;=\; \mathrm{KL}\left(\mathbb{P}_{\mathcal{T}}(Y) \;\Big|\Big|\; \widehat{\mathbb{P}}(Y)\right),$$

and by the error $\varepsilon_{X|Y}$ of its generative model, i.e.,

$$\mathrm{KL}^{\mathcal{D}}_{Y|X} \;\leq\; \mathrm{KL}_{\widehat{Y}} \,+\, \varepsilon_{X|Y}\,.$$

Proof. We use the chain rule from Prop. 2.2.ii),

$$\mathrm{KL}^{\mathcal{D}}_{Y|X} \;=\; \mathrm{KL}_{\widehat{Y}} + \mathrm{KL}^{\mathcal{D}}_{X|Y} - \mathrm{KL}^{\mathcal{D}}_{X} \;\leq\; \mathrm{KL}_{\widehat{Y}} + \mathrm{KL}^{\mathcal{D}}_{X|Y} \;\leq\; \mathrm{KL}_{\widehat{Y}} \,+\, \varepsilon_{X|Y},$$

where the last step employs our consistency assumption. □

The prior $\widehat{\mathbb{P}}(Y)$ from Eq. 6.3.2 does not have to match the ACS-generated class proportions $\mathbb{P}_{\mathcal{S}}(Y)$. In this sense, $\mathrm{KL}_{\widehat{Y}}$ is independent of the ACS-generated training set; however, if some informed prior existed, we might as well have employed it in ACS.

The error of the generative model, $\varepsilon_{X|Y}$, does depend on the training data, but only indirectly on the ACS-chosen class proportions. In particular, we assume through Def. 2.3 that $\mathbb{P}_{\mathcal{S}}(X \mid Y) = \mathbb{P}(X \mid Y)$, so that any consistent estimator can learn the true $\mathbb{P}(X \mid Y)$ from the training data, at least up to a divergence of $\varepsilon_{X|Y}$. As long as the ACS strategy assigns a non-zero probability to each class, we will reduce $\varepsilon_{X|Y}$ eventually by sampling from any class proportion; what matters for this eventual reduction is the sheer amount of training data, which continuously increases during the data acquisition process.

We must, however, recognize that the rate with which $\varepsilon_{X|Y}$ decreases can indeed depend on the class proportions. In fact, the idea that the natural class proportions $\mathbb{P}_{\mathcal{T}}(Y)$ might not yield an optimal *rate* of decrease has motivated previous work on ACS [19], [146] and it may motivate future work as well. However, Th. 6.2 strongly suggests that this potential disadvantage of acquiring data in terms of $\mathbb{P}_{\mathcal{T}}(Y)$ is eventually ruled out as the amount of training data increases. In other words, the error term $\varepsilon_{X|Y}$ might benefit from a non-proportional ACS heuristic, but only at the beginning of the data acquisition process or under extreme class imbalances.

The following experiment explores the transition from an early acquisition stage, where active ACS heuristics can still benefit learning, to a late acquisition stage, where we expect the largest benefit to come from the "proportional" strategy, which always acquires data in terms of $\mathbb{P}_{\mathcal{T}}(Y)$. To this end, we measure the performance of $\mathbb{P}_{\mathcal{T}}(Y)$ relative to all possible, alternative class proportions that a different ACS heuristic might have chosen. This measurement is taken out as follows:

Measuring the Performance of a Training Set Our evaluation is similar to the setup proposed by Kottke et al. [146]. Namely, we stick to their Bayesian classifier, a Parzen window classifier [163] with a constant KDE bandwidth of 0.05, scale each feature to the unit interval, and measure performance in terms of test set accuracy. Additionally, we also employ a support vector classifier and a decision tree as non-Bayesian alternatives. Unlike Kottke et al., we reduce the dimensionality of the data with NCA [162] because the number of features in some data sets would otherwise result in poor KDE estimates. For testing, we sample $\frac{1}{3}$ of all examples in each data set in their natural class proportions. This pipeline provides us with a performance score for arbitrary training sets.

Measuring Relative Performance We apply the above pipeline to several class proportion candidates $p \in \mathbb{R}^C$, which are evenly distributed over the entire unit simplex, and we rank all candidates according to their performance values. The natural class proportions are one of these candidates. Their position in this ranking tells us, how many of the competing class proportions are strictly better: the natural proportions are optimal if there is no better candidate; the number of better competitors further tells us how close to the optimum the natural class proportions are. The ranks we report here are relative, i.e., they describe the *percentage* of other class proportions that perform strictly better. For instance, a value of 0.5 means that one half of the other candidates outperform the natural proportions. Note that a rank relates multiple class proportions to each other; it does not show the trivial improvement in accuracy due to a larger training set, but the improvement *over other strategies* as the training set grows.

We perform the ranking for several training set sizes to see the effect of an increasing amount of training data. For each step, the ranking is computed separately in each of 1000 repetitions, from which the mean relative ranks are reported together with their lower and upper quartiles.

The results of this experiment, which are illustrated in Fig. 6.4, exhibit a downward trend of the ranks for all classifiers. Thus, the natural class proportions become more competitive as the size of the training set increases. One can, however, also observe that several candidates outperform the natural proportions. In this regard, we must note that a relative rank > 0 only tells us about the existence of *proportions* that outperform the natural

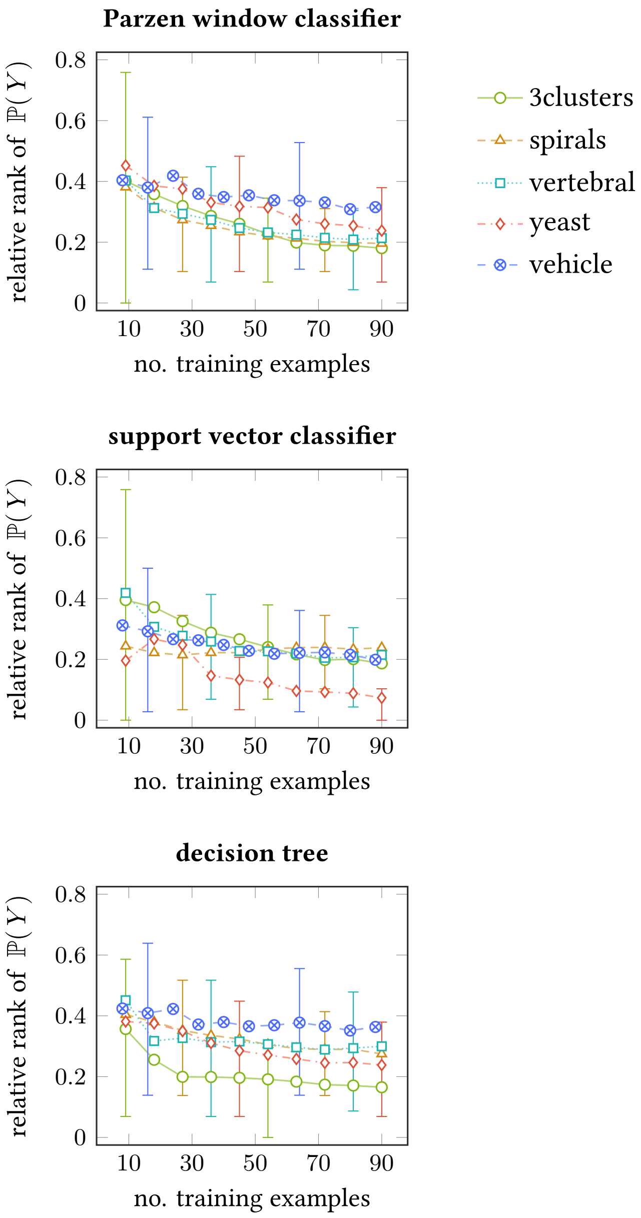

Figure 6.4.: The rank of the natural class proportions, relative to alternative proportions, is presented for different sizes of the training set. A lower rank translates to a (relatively) better performance measured in terms of accuracy. Error bars present the lower and upper quartiles over 1000 trials. We see that the natural proportions gain superiority over their competitors while the number of training examples increases. This figure is adapted from one of our publications [8].

candidate, but not about the existence of an ACS strategy that actually finds them. Indeed, the performance gained by two different acquisitions with the same class proportions can deviate substantially between trials. As a consequence, it is extremely difficult to actively decide for future class proportions based on performances exhibited in the past, which is a general difficulty of the ACS task. Our findings, both theoretically and experimentally, imply that the goal of outperforming natural proportions is becoming more difficult while more training data is being acquired.

6.4. Certification of Model Robustness in Active Class Selection

In the previous section, we have taken a viewpoint from information theory to study the implications of ACS qualitatively, in the large sample limit. In the following, we take out a more quantitative assessment of ACS by establishing a different viewpoint from learning theory. This viewpoint provides us with a PAC bound, from which we develop a *certificate* for the robustness of ACS-trained classifiers. In particular, this certificate declares the set of class proportions for which a classifier is valid with a high probability. Our PAC bound also motivates a novel strategy for ACS, which we develop in Sec. 6.5.

In the following sub-sections, we begin with proving our PAC bound for ACS. Based on this bound, we provide a quantitative assessment of the ACS-induced error and we develop our certificate of model robustness. Finally, we validate our certificate empirically and we certify actual gamma hadron classifiers from astro-particle physics.

6.4.1. PAC Bound for Active Class Selection

The common foundation of our certificate and our ACS strategy is a PAC bound for ACS-trained classifiers. This bound adapts the standard IID bound from learning theory, which we have introduced in Th. 2.1. This standard bound quantifies the probability that the estimation error for the source domain risk $R_{\mathcal{S}}(h;\ell)$, induced by the finite amount m of data in a data set $\mathcal{D}_{XY} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \,:\, 1 \leq i \leq m\} \sim \mathbb{P}_{\mathcal{S}}^m$, is bounded above by some $\varepsilon > 0$.

The first step of adapting the standard bound from Th. 2.1 to ACS is to extract an asymmetric variant thereof. This variant, which still addresses IID data, states the following:

Corollary 6.2 (Asymmetric IID Bound [12]). For any $\varepsilon^{(l)}, \varepsilon^{(u)} > 0$, for any loss function $\ell : (\mathcal{Z} \times \mathcal{Y}) \rightarrow \mathbb{R}$, and for any fixed $h \in \mathcal{H}$, each of the following bounds holds with probability at least $1 - \delta^{(i)}$ respectively, where $\delta^{(i)} = e^{-2m(\varepsilon^{(i)})^2}$ and $i \in \{l, u\}$:

i) $R_{\mathcal{D}}(h;\ell) - R_{\mathcal{S}}(h;\ell) \;\leq\; \varepsilon^{(l)}$

ii) $R_{\mathcal{S}}(h;\ell) - R_{\mathcal{D}}(h;\ell) \;\leq\; \varepsilon^{(u)}$

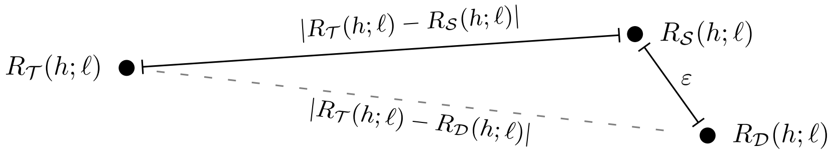
$|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)|$
$R_{\mathcal{T}}(h;\ell)$
$R_{\mathcal{S}}(h;\ell)$
ε
$|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{D}}(h;\ell)|$
$R_{\mathcal{D}}(h;\ell)$

Figure 6.5.: Illustration of the proofs of Theorems 6.3 and 6.4. Keeping $\delta > 0$ fixed, we can choose an estimation error $\varepsilon \to 0$ as the training set size $m \to \infty$. What remains is the *inter-domain gap* $|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)|$. This figure is adapted from one of our publications [12].

Proof. The claim follows from applying Eq. 2.6 to $\bar{z}$ and $-\bar{z}$, just like we do in the proof of Prop. 2.1. This time, however, we use two different $\varepsilon^{(l)}$, $\varepsilon^{(u)}$ for the two sides of the bound and we do not combine them via the union bound. □

To study the ACS problem, we now replace the IID assumption above with the identical mechanism / prior probability shift assumption from Def. 2.3. The result of this replacement is Th. 6.3, in which the factor 4 in δ, as compared to the factor 2 in the δ of Lemma 2.1, stems from the fact that either the upper bound or the lower bound might be violated, each time with at most the same probability. The idea of our proof, which builds on the triangle inequality, is illustrated in Fig. 6.5.

Theorem 6.3 (Identical Mechanism Bound [12]). For any $\varepsilon > 0$, for any loss function $\ell : (\mathcal{Z} \times \mathcal{Y}) \to \mathbb{R}$, and for any fixed $h \in \mathcal{H}$, it holds with probability at least $1 - \delta$, where $\delta = 4e^{-2m\varepsilon^2}$, that

$$|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| - \varepsilon \;\leq\; |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{D}}(h;\ell)| \;\leq\; |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| + \varepsilon\,.$$

Proof. We employ Prop. 2.1 through the triangle inequality (see Fig. 6.5):

$$\begin{aligned} |R_{\mathcal{D}}(h;\ell) - R_{\mathcal{T}}(h;\ell)| \;&\leq\; |R_{\mathcal{D}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \;+\; |R_{\mathcal{S}}(h;\ell) - R_{\mathcal{T}}(h;\ell)| \\ &\leq\; \varepsilon \;+\; |R_{\mathcal{S}}(h;\ell) - R_{\mathcal{T}}(h;\ell)|, \end{aligned}$$

where the second inequality holds with probability at least $1 - 2e^{-2m\varepsilon^2}$.

Likewise, and with the same probability, we use the triangle inequality for the other side of the claim:

$$\begin{aligned} |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \;&\leq\; |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{D}}(h;\ell)| \;+\; |R_{\mathcal{D}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \\ \Leftrightarrow |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{D}}(h;\ell)| \;&\geq\; |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \;-\; |R_{\mathcal{D}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \\ &\geq\; |R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \;-\; \varepsilon \end{aligned}$$ □

The above bound addresses a single, fixed hypothesis $h \in \mathcal{H}$, which suits our goal of certifying any given prediction model. For completeness, however, let us also mention that Th. 6.3 can be extended to entire hypothesis classes $\mathcal{H}$. A bound of this kind is useful for analyzing algorithms that return a prediction model as a function of the training data. As an example, we obtain the following result for finite hypothesis classes.

Theorem 6.4 (Identical Mechanism Bound; Finite Hypothesis Class [12]). Consider a finite hypothesis class $\mathcal{H}$, i.e., $|\mathcal{H}| < \infty$. With probability at least $1 - \delta$, where $\delta = 4|\mathcal{H}|e^{-2m\varepsilon^2}$, the upper and lower bounds from Theorem 6.3 hold for all $h \in \mathcal{H}$.

Proof. We draw a training set $\mathcal{D}$ of size m, where each individual example is drawn from $\mathcal{X} \times \mathcal{Y}$, according to $\mathbb{P}_\mathcal{S}$. Consequently, the full training set is drawn from $(\mathcal{X} \times \mathcal{Y})^m$, according to the probability density $\mathbb{P}_\mathcal{S}^m$. We are now interested in the probability that $\mathbb{P}_\mathcal{S}^m$ assigns to the event that all $h \in \mathcal{H}$ admit to the identical mechanism bound:

$$\begin{aligned}\mathbb{P}_\mathcal{S}^m\Big(\big\{\mathcal{D} : \forall h \in \mathcal{H},\ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ - \ \varepsilon \ &\leq |R_\mathcal{T}(h) - R_\mathcal{D}(h)| \\ &\leq |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ + \ \varepsilon\big\}\Big)\end{aligned}$$

We estimate the above probability from the probability of the converse event; if the above probability is p, then the following must be $1 - p$:

$$\begin{aligned}\mathbb{P}_\mathcal{S}^m\Big(\big\{\mathcal{D} : \exists h \in \mathcal{H},\ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ - \ \varepsilon \ > \ |R_\mathcal{T}(h) - R_\mathcal{D}(h)| \\ \wedge \quad |R_\mathcal{T}(h) - R_\mathcal{D}(h)| \ > \ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ + \ \varepsilon\big\}\Big)\end{aligned}$$

We now apply the union bound twice. This bound states that $\mathbb{P}(A \wedge B) \leq \mathbb{P}(A) + \mathbb{P}(B)$ for any two events A and B:

$$\begin{aligned}\ldots \quad &\leq \mathbb{P}_\mathcal{S}^m\Big(\big\{\mathcal{D} : \exists h \in \mathcal{H},\ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ - \ \varepsilon \ > \ |R_\mathcal{T}(h) - R_\mathcal{D}(h)|\big\}\Big) \\ &\quad + \mathbb{P}_\mathcal{S}^m\Big(\big\{\mathcal{D} : \exists h \in \mathcal{H},\ |R_\mathcal{T}(h) - R_\mathcal{D}(h)| \ > \ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ + \ \varepsilon\big\}\Big) \\ &\leq \sum_{h \in \mathcal{H}} \mathbb{P}_\mathcal{S}^m\Big(\big\{\mathcal{D} : |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ - \ \varepsilon \ > \ |R_\mathcal{T}(h) - R_\mathcal{D}(h)|\big\}\Big) \\ &\quad + \sum_{h \in \mathcal{H}} \mathbb{P}_\mathcal{S}^m\Big(\big\{\mathcal{D} : |R_\mathcal{T}(h) - R_\mathcal{D}(h)| \ > \ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ + \ \varepsilon\big\}\Big)\end{aligned}$$

We have thus reduced the probability of the identical mechanism bound with respect to an entire hypothesis class $\mathcal{H}$ to a sum of probabilities for single hypotheses $h \in \mathcal{H}$. The single-hypothesis case has already been proven in Th. 6.3. Let us rephrase this result here to clarify the connection: Each of the following statements describes a violation of the Th. 6.3 bound, each having a probability of at most $2e^{-2m\varepsilon^2}$:

- $|R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ - \ \varepsilon \ > \ |R_\mathcal{T}(h) - R_\mathcal{D}(h)|$
- $|R_\mathcal{T}(h) - R_\mathcal{D}(h)| \ > \ |R_\mathcal{T}(h) - R_\mathcal{S}(h)| \ + \ \varepsilon$

These two events, together with their probabilities, can be plugged into the above transformation, which proves the claim:

$$\ldots \quad \leq \quad \sum_{h\in\mathcal{H}} 2e^{-2m\varepsilon^2} + \sum_{h\in\mathcal{H}} 2e^{-2m\varepsilon^2} \quad = \quad 4|\mathcal{H}|e^{-2m\varepsilon^2} \qquad \square$$

The lower and upper bounds in Theorems 6.3 and 6.4 quantitatively asses the way in which the total error $|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{D}}(h;\ell)|$ approaches the *inter-domain gap* $|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)|$, depending on the interplay between ε, δ, m, ℓ, and $\mathcal{H}$. It is therefore a quantitative and thus more nuanced equivalent of Prop. 6.2. The inter-domain gap is constant with respect to the random draw of the training sample $\mathcal{D}_{XY} \sim \mathbb{P}_{\mathcal{S}}^m$ and is therefore independent of ε, of δ, and of m. Consequently, it remains even with an infinite amount of training data. Depending on the choice of $\mathcal{H}$ and ℓ, and depending on the data distribution, it may be large or negligible. In the following, we will therefore study this error in more detail.

6.4.2. Quantitative Assessment of the Domain Gap

We begin our study of the inter-domain gap $|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)|$ by factorizing the total expected risk $R(h;\ell)$ from Def. 2.1 into label-dependent risks $\ell_X(h,y)$ that are marginalized over the entire feature space $\mathcal{X}$. These risks only depend on the hypothesis h and on the label y and are, under the identical mechanism assumption from Def. 2.3, identical among $\mathcal{S}$ and $\mathcal{T}$.

$$\begin{aligned} R(h;\ell) \quad &= \quad \int_{\mathcal{Y}} \int_{\mathcal{X}} \mathbb{P}(X = x \wedge Y = y) \cdot \ell(h(x), y) \ \mathrm{d}x \ \mathrm{d}y \\ &= \quad \int_{\mathcal{Y}} \mathbb{P}(Y = y) \cdot \underbrace{\int_{\mathcal{X}} \mathbb{P}(X = x \mid Y = y) \cdot \ell(h(x), y) \ \mathrm{d}x}_{= \ell_X(h,y)} \ \mathrm{d}y \end{aligned}$$

Plugging $\ell_X(h,y)$ into the domain gap $|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)|$ from the Theorems 6.3 and 6.4 allows us to marginalize the class-wise risks over the label space:

$$|R_{\mathcal{T}}(h;\ell) - R_{\mathcal{S}}(h;\ell)| \quad = \quad \left| \int_{\mathcal{Y}} \Big(\mathbb{P}_{\mathcal{T}}(Y = y) - \mathbb{P}_{\mathcal{S}}(Y = y) \Big) \cdot \ell_X(h,y) \ \mathrm{d}y \right|$$

For classification tasks, i.e., for $\mathcal{Y} = \{1, \ldots, C\}$ with $C \geq 2$, we define the vectors $p_{\mathcal{S}}, p_{\mathcal{T}} \in \mathbb{R}^C$ through $[p_\bullet]_i = \mathbb{P}_\bullet(Y = i)$, i.e., through the label probabilities in the domains $\mathcal{S}$ and $\mathcal{T}$. Furthermore, we define a vector $\ell_h \in \mathbb{R}^C$ of class-wise risks through $[\ell_h]_i = \ell_X(h,i)$. The computation of the ACS-induced domain gap then simplifies to an absolute difference between scalar products $R_\bullet(h;\ell) = \sum_{i\in\mathcal{Y}} [p_\bullet]_i [\ell_h]_i = \langle p_\bullet, \ell_h \rangle$. Namely, for classification tasks:

$$|R_{\mathcal{T}}^{\mathrm{clf}}(h;\ell) - R_{\mathcal{S}}^{\mathrm{clf}}(h;\ell)| \quad = \quad \big| \langle p_{\mathcal{T}}, \ell_h \rangle - \langle p_{\mathcal{S}}, \ell_h \rangle \big| \qquad (6.5)$$

Before we move on to a theorem about the practical implications of Eq. 6.5, let us build an intuition about these implications in a more simple setting, binary classification.

Example 6.1 *(Binary classification).* In binary classification, the situation from Eq. 6.5 simplifies to $\mathcal{Y} = \{1, 2\}$ with $\mathbb{P}_\bullet(Y = 1) = p_\bullet$ and $\mathbb{P}_\bullet(Y = 2) = 1 - p_\bullet$. Let $\Delta p = |p_\mathcal{T} - p_\mathcal{S}|$ be be the absolute difference of the binary class proportions between the two domains and let $\Delta \ell_X = |\ell_X(h, 2) - \ell_X(h, 1)|$ be the absolute difference between the class-wise risks. The difference $\Delta \ell_X$ is independent of the class proportions and can be defined over any loss function ℓ. Rearranging Eq. 6.5 for binary classification, we obtain

$$\begin{aligned} & |R^{\text{bin}}_\mathcal{T}(h; \ell) - R^{\text{bin}}_\mathcal{S}(h; \ell)| \\ = \; & \big|\big(p_\mathcal{T} \ell_X(h, 2) + (1 - p_\mathcal{T}) \ell_X(h, 1)\big) \; - \; \big(p_\mathcal{S} \ell_X(h, 2) + (1 - p_\mathcal{S}) \ell_X(h, 1)\big)\big| \\ = \; & \big|\big(p_\mathcal{T} - p_\mathcal{S}\big) \cdot \big(\ell_X(h, 2) - \ell_X(h, 1)\big)\big| \\ = \; & \Delta p \cdot \Delta \ell_X, \end{aligned} \tag{6.6}$$

from which we see that in binary classification, for any loss function, the domain gap induced by ACS is simply the product of the class proportion difference Δp and the (true) class-wise risk difference $\Delta \ell_X$. If one of these terms is zero, so is the inter-domain gap. If one of these terms is non-zero but fixed, the domain gap will grow linearly with the other term.

Example 6.2 *(Binary classification with zero-one loss).* Let us illustrate Eq. 6.6 a little further. The zero-one loss is defined by $\ell(h(x), y) = 0$ if the prediction is correct, i.e., if $y = \mathbb{1}_{h(x) > 0.5}$, and $\ell(h(x), y) = 1$ otherwise. Consequently, $\ell_X(h, 2)$ is the true rate of false positives and $\ell_X(h, 1)$ is the true rate of false negatives. The more similar these rates are, the smaller will the inter-domain gap be for any distribution of classes in the target domain. Supposing that balanced training sets tend to balance $\ell_X(h, 2)$ and $\ell_X(h, 1)$, we can argue that balanced training sets (supposedly) maximize the range of feasible target domains with respect to the zero-one loss.

Example 6.3 *(Cost-sensitive learning).* The situation is quite different if the binary zero-one loss is weighted by the class, i.e., if mis-predictions are penalized by $\ell(h(x), y) = w_y$. Such a weighting is common in cost-sensitive and imbalanced classification [143]. Here, Eq. 6.6 illustrates how counteracting class imbalance with weights can increase the robustness of the model: balancing $\ell_X(h, 2)$ and $\ell_X(h, 1)$ will increase the range of target domains that are feasible under the class-based weighting.

For completeness, we extend a part of this intuition from binary classification to classification tasks with an arbitrary number of classes:

Theorem 6.5. [12] In classification, the inter-domain gap $|R^{\text{clf}}_\mathcal{T}(h; \ell) - R^{\text{clf}}_\mathcal{S}(h; \ell)|$ from Theorem 6.3 is equal to zero if one of the following conditions holds:

i) $p_\mathcal{S} = p_\mathcal{T}$

ii) $\ell_X(h, i) = \ell_X(h, j) \;\; \forall \, i, j \in \mathcal{Y}$

Proof. Condition i) trivially yields the claim through Eq. 6.5. Condition ii) means that $\Delta\ell = |\ell_X(h,i) - \ell_X(h,j)| = 0$ for every binary sub-task in a one-vs-one decomposition of the label set $\mathcal{Y}$. The domain gap of each binary sub-task, and therefore the total domain gap, is then zero according to Eq. 6.6. □

Despite the fact that condition 6.5.ii) yields a domain gap of zero, one should not prematurely jump to the conclusion that a learning algorithm should enforce this condition necessarily. Recall that Theorem 6.5 addresses the domain gap, but not the deployment risk which we actually want to minimize; if enforcing condition 6.5.ii) results in a high source domain error, all domain robustness will not help to find an accurate target domain model. We therefore advise practitioners to carefully weigh out the source domain error with the domain robustness of the model, depending on the requirements of the use case at hand. Bayesian classifiers, which allow practitioners to mimic arbitrary $p_\mathcal{S}$ even after training, can prove useful in this regard.

6.4.3. Certification of Binary Classifiers

We now develop a certificate which declares a set of class proportions $\mathcal{P}$, for which an ACS-trained classifier is valid with a high probability. This certificate is theoretically justified by the PAC bounds and the quantitative assessment of the domain gap, which we have presented above. While the general concept of our certificate is applicable to any classification task, we substantiate our proposal here in terms of a certificate for binary classifiers, in particular.

Model certification, in the broad sense of model performance reports [164]–[166] and formal proofs of robustness [167]–[170], has motivated us to study model robustness for the particular setting of ACS. Our certificate is only a single component in the more comprehensive reports that are conceived in the literature; yet, the certification of feasible class proportions is a central and trust-critical issue in ACS, due to the arbitrary choice of the training class proportions that a class-conditional data generator offers. Moreover, the concept of feasible class proportions is easily understandable and can be relevant in any other setting where prior probability shift occurs.

In particular, our certificate declares a set of class proportions, to which a fixed hypothesis h, trained in the ACS-induced domain $\mathcal{S}$, is safely applicable. By "safely", we mean that during the deployment on $\mathcal{T}$, h induces only a small domain-induced error with a high probability.

Definition 6.1 (Certified Hypothesis [12]). *A hypothesis $h \in \mathcal{H}$ is (ε, δ)-certified for all class proportions in the set $\mathcal{P} \subseteq \mathbb{R}^C$ if with probability at least $1 - \delta$ and $\varepsilon, \delta > 0$:*

$$|R_\mathcal{T}(h;\ell) - R_\mathcal{S}(h;\ell)| \;\leq\; \varepsilon \quad \forall\, p_\mathcal{T} \in \mathcal{P}$$

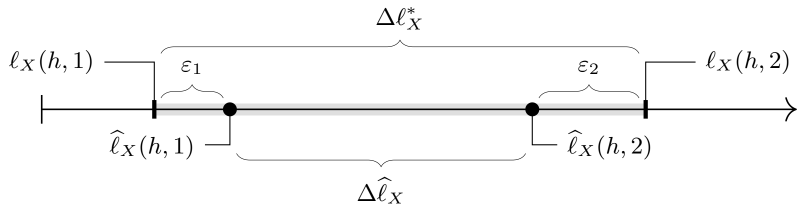

Figure 6.6.: Estimation of the minimum upper bound $\Delta\ell^*_X$ from data. We have to choose ε_1 and ε_2 as small as possible, within a given, total probability budget δ. Adding ε_2 to the estimate $\widehat{\ell}_X(h,2)$ and subtracting ε_1 from the estimate $\widehat{\ell}_X(h,1)$ yields the desired, minimum upper bound. This figure is taken from one of our publications [12].

For simplicity, we limit our presentation to binary classification, i.e., $C = 2$ (see Ex. 6.1). In this case, $\mathcal{P}$ is simply a scalar range $[p^{\min}_{\mathcal{T}},\, p^{\max}_{\mathcal{T}}]$ of the positive class proportion $\mathbb{P}_{\mathcal{T}}(Y=1)$. According to Eq. 6.6, this range is defined by the largest Δp^* for which

$$\Delta p \cdot \Delta\ell_X \;\leq\; \varepsilon \quad \forall\, \Delta p \;\leq\; \Delta p^*. \tag{6.7}$$

Keep in mind that $\Delta\ell_X$ is defined over the *true* class-wise risks. If we knew them, we could simply rearrange Eq. 6.7 to find the largest Δp for a given ε; the equation would then hold with probability one. However, we do not know the true class-wise risks; instead, we estimate an upper bound that is only exceeded by the true $\Delta\ell_X$ with a small probability of at most $\delta > 0$. Particularly, to maximize Δp^*, we find the *smallest* upper bound $\Delta\ell^*_X$ among all such upper bounds.

An empirical estimate $\Delta\widehat{\ell}_X$ of the true $\Delta\ell_X$ is given by the empirical class-wise risks $\widehat{\ell}_X(h,y)$ observed in an ACS-generated validation sample $\mathcal{D}$:

$$\Delta\widehat{\ell}_X \;=\; \left|\widehat{\ell}_X(h,1) - \widehat{\ell}_X(h,2)\right|, \quad \text{where} \quad \widehat{\ell}_X(h,y) \;=\; \tfrac{1}{m_y} \textstyle\sum_{i\,:\,y_i=y} \ell(y, h(x_i))$$

Here, each $\widehat{\ell}_X(h,y)$ can be associated with maximum lower and upper errors $\varepsilon^{(l)}_y, \varepsilon^{(u)}_y > 0$, which are not exceeded with probabilities at least $1-\delta^{(l)}_y$ and $1-\delta^{(u)}_y$. By choosing $\varepsilon^{(l)}_y, \varepsilon^{(u)}_y$ for both classes, we can thus find all upper bounds of the true $\Delta\ell_X$ that hold with at least the desired probability $1-\delta$.

Fig. 6.6 sketches our estimation of the *smallest* upper bound $\Delta\ell^*_X$. For simplicity, we assume that $\widehat{\ell}_X(h,2) \geq \widehat{\ell}_X(h,1)$. This assumption comes without a loss of generality because we can otherwise simply switch the labels to make the assumption hold. Now, $\widehat{\ell}_X(h,1)$ shrinks at most by ε_1 and $\widehat{\ell}_X(h,2)$ grows at most by ε_2. Minimizing ε_1 and ε_2 simultaneously, within a user-specified probability budget δ, yields the desired minimum upper bound $\Delta\ell_X^*$ which the true $\Delta\ell_X$ only exceeds with probability at most $\delta = \delta_1 + \delta_2 - \delta_1\delta_2$. We find the values of δ_1 and δ_2 through Corollary 6.2, letting

$$\begin{aligned} &-\underbrace{(\widehat{\ell}_X(h,2) - \widehat{\ell}_X(h,1) + \varepsilon_1)}_{=\,\varepsilon_2^{(l)}} \;\leq\; \ell_X(h,2) - \widehat{\ell}_X(h,2) \;\leq\; \underbrace{\varepsilon_2}_{=\,\varepsilon_2^{(u)}} \\ \text{and} \quad &-\underbrace{(\widehat{\ell}_X(h,2) - \widehat{\ell}_X(h,1) + \varepsilon_2)}_{=\,\varepsilon_1^{(u)}} \;\leq\; \widehat{\ell}_X(h,1) - \ell_X(h,1) \;\leq\; \underbrace{\varepsilon_1}_{=\,\varepsilon_1^{(l)}}, \end{aligned}$$

so that $\delta_y = \delta_y^{(l)} + \delta_y^{(u)} - \delta_y^{(l)}\delta_y^{(u)}$ and $\delta_y^{(i)} = e^{-2m_y(\varepsilon_y^{(i)})^2}$.

During the optimization, strict inequalities are realized through non-strict inequalities with some sufficiently small $\tau > 0$:

$$\min_{\varepsilon_1,\,\varepsilon_2 \in \mathbb{R}} \;\varepsilon_2 + \varepsilon_1, \quad \text{s.t.} \;\begin{cases} \varepsilon_1,\, \varepsilon_2 & \geq\; \tau \\ \delta - (\delta_1 + \delta_2 - \delta_1\delta_2) & \geq\; 0 \end{cases} \tag{6.8}$$

The minimizer $(\varepsilon_1^*, \varepsilon_2^*)$ of this optimization problem defines the smallest upper bound $\Delta\ell_X^* = (\widehat{\ell}_X(h,2) + \varepsilon_2^*) - (\widehat{\ell}_X(h,1) - \varepsilon_1^*)$ that is not exceeded by the true $\Delta\ell_X$ with probability at least $1 - \delta$. Choosing $\Delta p^* = \frac{\varepsilon}{\Delta\ell_X^*}$ will make Eq. 6.7 hold with the same probability, so that the range $[p_\mathcal{S}-\Delta p^*,\, p_\mathcal{S}+\Delta p^*]$ of binary deployment class proportions $p_\mathcal{T}$ is (ε, δ)-certified according to Def. 6.1.

If only small data volumes are available, it can happen that ϵ_1 must exceed $\widehat{\ell}_X(h,1)$ to stay within the user-specified probability budget δ. This situation would mean that the lower bound $\ell_X(h,1) = \widehat{\ell}_X(h,1) - \varepsilon_1$ is below zero, which does not reflect the basic loss property $\ell(h,y) \geq 0$. If the estimation of $\Delta\ell_X^*$ fails in this way, we fall back to a more simple, one-sided estimation. Namely, we only minimize the two upper bounds $\varepsilon_y^{(u)}$ that depend only on ε_2 and fix the two lower bounds to $\varepsilon_y^{(l)} = 0$. Doing so allows us to estimate a valid upper bound $\Delta\ell_X^*$ also for arbitrarily small data sets.

6.4.4. Empirical Validation of Binary Certificates

In the following, we show that an (ε, δ) certified class proportion set $\mathcal{P}$ indeed characterizes an upper bound of the inter-domain gap. Our experiments even demonstrate that our certificate, being estimated only with source domain data, is very close to bounds that are obtained with labeled target domain data and are therefore not accessible in practice. It is therefore highly beneficial when only ACS-generated data is available and no labeled data from the target domain can be accessed.

We randomly sub-sample the data to generate different deployment class proportions $p_\mathcal{T}$ while keeping $\mathbb{P}(X \mid Y)$ fixed, in accordance to Def. 2.3. We compare two ways of estimating the target domain risk:

a) Our baseline is an empirical estimate $\widehat{R}_\mathcal{T}$ of the target domain risk that is computed with actual target domain data. Data of this kind is typically unavailable in practice.

b) We predict the target domain risk $R_\mathcal{T}$ from an (ε, δ) certificate by adding the domain gap parameter ε to the empirical source domain risk $\widehat{R}_\mathcal{S}$. We always choose the certificates such that they cover the class proportion difference $\Delta p = |p_\mathcal{T} - p_\mathcal{S}|$; in fact, we consider ε as a function of Δp in this experiment.

The certificate is *correct* if $\widehat{R}_\mathcal{S} + \varepsilon \geq \widehat{R}_\mathcal{T}$ holds, i.e., if ε indeed characterizes an upper bound of the inter-domain gap. If the two values are close to each other, i.e., if $\widehat{R}_\mathcal{S} + \varepsilon \approx \widehat{R}_\mathcal{T}$, we speak of a *tight* upper bound.

Correctness: Our experiments cover a repeated three-fold cross validation on eight imbalanced data sets, eight loss functions, and three learning algorithms, to represent a broad range of scenarios. Of all 9000 certificates, only 4.5% fail the test of ensuring $\widehat{R}_\mathcal{S} + \varepsilon \geq \widehat{R}_\mathcal{T}$. Since we have used $\delta = 0.05$ in these experiments, this amount of failures is actually foreseen by the statistical nature of our certificate: if it holds with probability at least $1 - \delta$, it is allowed to fail in 5% of all tests. This margin is almost completely used but not exceeded. Consequently, our certificate is correct in the sense of indeed characterizing an upper bound ε of the inter-domain gap with probability at least $1 - \delta$.

Tightness: A *fair* comparison between our certificate and our baseline $\widehat{R}_\mathcal{T}$ requires us to take the estimation error $\varepsilon_\mathcal{T}$ of the baseline into account. This necessity stems from the fact that $\widehat{R}_\mathcal{T}$ is also just an estimate from a finite amount of data. Having access to labeled target domain data will thus yield an upper bound $\widehat{R}_\mathcal{T} + \varepsilon_\mathcal{T}$ of the true target domain error $R_\mathcal{T}$, according to Prop. 2.1; this upper bound is then compared to our certificate, which has only seen the source domain data.

Fig. 6.7 presents this comparison for a random selection of our experiments. For most target domains $p_\mathcal{T}$, the two predictions (■ and ▲) are almost indistinguishable from each other. This observation means that the certificate, which is based only on source domain data, is as accurate as estimating the target domain risk with a privileged access to labeled target domain data. Over all 9000 certificates, we find a mean absolute difference between the two predictions of merely 0.049; in fact, all supplementary plots [12] look highly similar to those displayed in Fig. 6.7, despite covering many other data sets, loss functions, and learning methods. The margin to the left of each vertical line appears because our certificate covers an absolute inter-domain gap rather than a signed value.

By choosing ε as a function of Δp, we have "turned the certificate around"; in a usual application, a user would rather fix the ε value and look for a certified range Δp of feasible class proportions. Therefore, we provide an excerpt of the certified Δp values in Tab. 6.3, where the certified target domain ranges $[p_\mathcal{S} - \Delta p^*,\ p_\mathcal{S} + \Delta p^*]$ induce a domain gap of at most $\varepsilon = 0.01$ with a probability of at least $1 - \delta = 0.95$. Since the domain gap is at most 0.01, we can expect a target domain risk of at most $R_\mathcal{S}(h; \ell) + 0.01$.

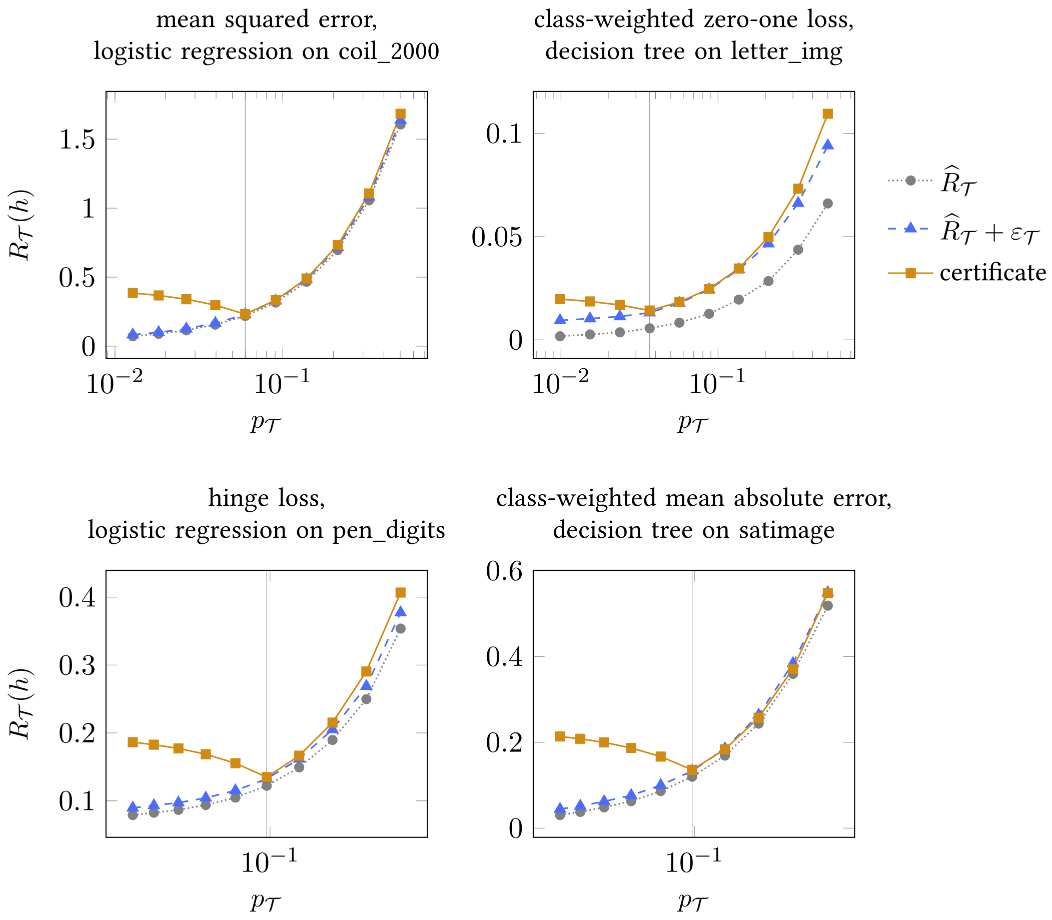

Figure 6.7.: The target domain risk $R_\mathcal{T}(h;\ell)$ is upper-bounded by our (ε,δ) certificate and a baseline $\widehat{R}_\mathcal{T} + \varepsilon_\mathcal{T}$ with privileged access to target domain data. Each of the above plots displays a different combination of loss function, learning method, and data set. The class proportions $p_\mathcal{T}$ of the target domain are varied over the x axis with a thin vertical line indicating the training proportions $p_\mathcal{S}$. This figure is adapted from the supplementary material of one of our publications [12].

Table 6.3.: Feasible class proportions Δp^*, according to (ε, δ) certificates that are computed for a class-weighted zero-one loss with $\varepsilon = 0.01$ and $\delta = 0.05$. This table is taken from one of our publications [12].

data set	classifier	$R_{\mathcal{S}}(h; \ell)$	$p_{\mathcal{S}}$	Δp^*
coil_2000	logistic regression	0.0722	0.0597	0.0109
coil_2000	decision tree	0.0778	0.0597	0.0107
letter_img	logistic regression	0.0179	0.0367	0.0463
letter_img	decision tree	0.0139	0.0367	0.0504
optical_digits	logistic regression	0.0406	0.0986	0.0437
optical_digits	decision tree	0.0463	0.0986	0.0309
pen_digits	logistic regression	0.038	0.096	0.044
pen_digits	decision tree	0.0216	0.096	0.0695
protein_homo	logistic regression	0.0056	0.0089	0.036
protein_homo	decision tree	0.006	0.0089	0.0291
satimage	logistic regression	0.1205	0.0973	0.0118
satimage	decision tree	0.0763	0.0973	0.018

Table 6.4.: The parameters of an (ε, δ) certificate that covers the extreme class proportions $p_{\mathcal{T}} = 10^{-4}$ in astro-particle physics. This table is adapted from one of our publications [12].

$R_{\mathcal{S}}(h)$	δ	ϵ_δ
0.058 ± 0.015	0.01	0.0315
	0.025	0.0314
	0.05	0.0314
	0.1	0.0313

6.4.5. Binary Certificates in Astro-Particle Physics

We now apply our certification scheme to the data of the FACT telescope. In particular, we apply the default analysis pipeline, fix δ to a small value (0.01 or 0.1), and select ε such that the resulting (ε, δ) certificate covers the anticipated maximum class proportion difference $\Delta p = |p_{\mathcal{T}} - p_{\mathcal{S}}|$ between the simulated and the observed domain. For both δ values, we obtain similar ε values under the zero-one loss, namely $\varepsilon_{(\delta=0.01)} = 0.0315$ and $\varepsilon_{(\delta=0.1)} = 0.0313$. We conclude that the ACS-induced zero-one loss of the FACT pipeline is at most 3.15% with probability at least 99%, and at most 3.13% with probability at least 90%. The pipeline is trustworthy within these specific ranges and improvements to these certified values can be achieved by improving the performance of the pipeline.

Tab. 6.4 presents the results of our astro-particle experiment. The fact that all ε values are close to each other stems from the large amount of source domain data (24000 examples) we use to certify the model. We note, however, that the zero-one loss is not a particularly informative measure for the performance of gamma hadron classifiers. Future research on ACS certification has to address quality measures that cannot be decomposed in terms of Eqs. 2.1 and 2.2, e.g., the F_1 score or the Li & Ma statistic from Def. 3.1.

Despite these current limitations, *physicists have confirmed to us the great value of certificates of this kind*: until now, they were not able to properly assess the performance implications of the mistaken class proportions of their simulated training data. Our certificate addresses this need in declaring an upper bound for the loss, which stems from the mistaken class proportions and holds with a precisely determined probability. Confidence assessments of this kind are of utter importance for drawing trustworthy physical interpretations from the telescope data.

6.5. A Strategy for Uncertain Deployment Class Proportions

Our certificate is motivated by the uncertainties that a practitioner can have about the class proportions which need to be handled during deployment. It addresses this uncertainty by building trust in a classifier after this classifier has been trained through ACS. However, it has no immediate implications on how to acquire data, in terms of an ACS strategy, if the deployment class proportions are uncertain.

In the following, we evolve the theoretical basis of our certificate towards a data acquisition strategy for ACS [9]. Unlike existing strategies, which solely focus on the perceived difficulties of the classes, our strategy uniquely combines the following qualities:

- It naturally supports uncertainty about the deployment class proportions. A practitioner expresses this uncertainty through a prior distribution. For binary classification, we propose a Beta prior, for instance.

- Our strategy is theoretically justified by the PAC learning bounds, which we have developed for the ACS setting in Sec. 6.4.

The goal of our strategy is to decrease the inter-domain gap $\Delta p \cdot \Delta \ell$ from Theorem 6.3 as much as possible, as according to a prior distribution $\widehat{\mathbb{P}}$ of the deployment class proportions $p_{\mathcal{T}}$. This decrease will allow us to learn accurate predictions for the target domain, as according to the prior beliefs of a domain expert.

Formally, we assume a prior $\widehat{\mathbb{P}} : [0,1] \rightarrow [0,1]$ of the positive class prevalence $p_{\mathcal{T}} \in [0,1]$ to be given. We incorporate $\widehat{\mathbb{P}}$ by marginalizing the inter-domain gap over this prior, as according to Eq. 6.9. Since we do not know the true $\Delta \ell_X$, we are using the estimated upper bound $\Delta \ell^*_X$ from Eq. 6.8 instead. Consequently, the marginalization according to $\Delta \ell^*_X$ is an upper bound, with probability $1 - \delta$, of the marginalization according to the true $\Delta \ell_X$.

$$\varepsilon^* \; = \; \int_0^1 \widehat{\mathbb{P}}(p_{\mathcal{T}} = p) \, \cdot \, \underbrace{|p_{\mathcal{S}} - p|}_{= \Delta p} \cdot \Delta \ell^*_X \;\; \mathrm{d}p \tag{6.9}$$

In each ACS iteration, we are free to alter the class proportions $p_{\mathcal{S}}$ of the ACS-generated training set to some degree, depending on how much data we acquire in each batch and on how much data we already have acquired. In fact, we can understand

$$p_{\mathcal{S}} = \frac{m_2}{(m_1 + m_2)} \tag{6.10}$$

as a function of the class-wise numbers of samples m_1 and m_2. The upper bound $\Delta \ell^*_X$ also lends itself for being interpreted as a function of sample sizes: the more data is acquired in both classes, the tighter will our estimation of this quantity be. Ultimately, we consider ε^* to be a function of m_1 and m_2, so that we can minimize ε^* via an optimal choice of m_1 and m_2 in each data acquisition batch.

6.5.1. Minimizing the Marginalized Error

Our strategy decreases ε^* in the direction of its steepest descent, i.e., it takes a simple gradient step with respect to the acquisition vector $m = (m_1, m_2)$. The gradient which defines the steepest descent is computed via the product rule:

$$\begin{aligned} \nabla_m \, \varepsilon^* \; &= \; \nabla_m f \cdot \Delta \ell^*_X \; + \; f \cdot \nabla_m \Delta \ell^*_X \\ \text{where } \; f(m) \; &= \; \int_0^1 \widehat{\mathbb{P}}(p_{\mathcal{T}} = p) \, \cdot \, |p_{\mathcal{S}}(m) - p| \;\; \mathrm{d}p \end{aligned} \tag{6.11}$$

We will come back to the function f shortly. For now, we plug $\Delta \ell^*_X$ and $\nabla_m \Delta \ell^*_X$ into the equation above. These functions are defined by

$$\begin{aligned} \Delta \ell^*_X(m) \; &= \; \widehat{\ell}_X(h;2) + \sqrt{\frac{\ln \delta_2}{-2[m]_2}} - \widehat{\ell}_X(h;1) + \sqrt{\frac{\ln \delta_1}{-2[m]_1}}, \\ [\nabla_m \Delta \ell^*_X(m)]_y \; &= \; \left(-\frac{\ln \delta_y}{[m]_y} \right)^{\frac{3}{2}} \cdot (2\sqrt{2} \ln \delta_y)^{-1}, \end{aligned} \tag{6.12}$$

where the δ_y are probabilities of violations of $\Delta\ell^*_X$ that occur from either one of the class-wise risks $\ell_X(h, y)$ in $\Delta\ell$. In fact, we have seen in Sec. 6.4 that finding a suitable assignment of δ_y values within a given probability budget $\delta = \delta_1 + \delta_2 - \delta_1\delta_2$ is the central difficulty in model certification; there, the sample size m is fixed, so that $\Delta\ell^*_X$ can be optimized over this assignment. Here, we keep the δ_y fixed instead, to values that are obtained with a certificate from previous ACS acquisitions. This change allows us to optimize $\Delta\ell^*_X$ over m to acquire new data and it guarantees that $\Delta\ell^*_X$ remains an upper bound of the true $\Delta\ell$ also in the next batch, at least with probability $1 - \delta$. The class-wise estimates $\widehat{\ell}_X(h, y)$ in Eq. 6.12 are the average values of risks in the training data; they are also part of our certificate.

6.5.2. A Beta Prior for Binary Class Proportions

Now we turn to the value and the gradient of the function f in Eq. 6.11. Plugging a parametric prior $\widehat{\mathbb{P}}$ into this function can allow us to compute these terms efficiently, in closed forms. To this end, a Beta(α, β) prior is suitable for binary classification because the Beta distribution is a conjugate prior of the Bernoulli distribution, which in turn is a suitable model for the prevalence of binary class labels. As a matter of convenience, the parameters $\alpha > 0$ and $\beta > 0$ can be chosen such that the resulting distribution has some predetermined mean and standard deviation; we believe that domain experts can often express their prior beliefs in terms of these properties.

Plugging a Beta prior into the f function from Eq. 6.11 yields the following components, where I is the regularized incomplete Beta function:

$$\begin{aligned} f_{\alpha,\beta}(m) &= \frac{2p_{\mathcal{S}}(m)^{\alpha}(1 - p_{\mathcal{S}}(m))^{\beta}}{(\alpha+\beta)B(\alpha,\beta)} + \Big(p_{\mathcal{S}}(m) - \frac{\alpha}{\alpha+\beta}\Big)\Big(2I_{p_{\mathcal{S}}(m)}(\alpha,\beta) - 1\Big) \\ \nabla_m f_{\alpha,\beta} &= \frac{2I_{p_{\mathcal{S}}(m)}(\alpha,\beta) - 1}{([m]_1 + [m]_2)^2} \cdot \begin{pmatrix} [m]_2 \\ -[m]_1 \end{pmatrix} \end{aligned} \tag{6.13}$$

Plugging Eq. 6.12 and 6.13 into Eq. 6.11 provides us with a gradient that we can compute analytically from a certificate with a δ_y assignment, from sample sizes $[m]_1$ and $[m]_2$ and from the prior parameters α and β. The negative gradient $-\nabla_m \varepsilon^*$ of the marginalized error ε^* defines the class-wise numbers of samples that our strategy acquires in the next data acquisition batch.

With small data volumes or with highly imbalanced classes, our strategy is dominated by the $\Delta\ell^*_X$ component; small classes need additional data until this upper bound holds with some desired probability $1 - \delta$. Contrastingly, when the total data volume is large, our strategy is dominated by the f component; to this end, a Beta prior favors class proportions that are close to its mean $\frac{\alpha}{\alpha+\beta}$. We have already seen in our information-theoretic examination of ACS, see Sec. 6.3, that a transition between these two behaviors is desirable because non-natural class proportions can facilitate learning only at the early acquisition stage, while the late stage benefits from the natural class proportions. The precise turn-ing point between these two behaviors is well-founded in the PAC learning theory from Sec. 6.4, which underlies the estimation of $\Delta\ell^*_X$.

6.5.3. Experimental Validation

We have parameterized the Beta prior of our strategy with a predetermined mean and standard deviation, both set to the value of $p_\mathcal{T}$. Accordingly, the mean of the prior is well aligned with the true class proportions of the deployment data; the uncertainty, however, which is expressed by the standard deviation, is as large as possible.

In accordance to a reliable evaluation methodology of active data acquisition strategies [171], we present pairwise differences between ACS strategies in terms of their statistical significance. A comprehensive way of plotting such differences is through critical difference (CD) diagrams [55], [56], which we have introduced in Sec. 2.4.2. We employ accuracy as the underlying performance metric and we conduct multiple trials to obtain an average performance value for each combination of strategy and data set.

We define the trials via five repetitions of a three-fold cross validation. From the *imbalanced-learn* package [145], we retrieve 13 data sets[9] that have at least 150 minority class samples (to facilitate sampling) and at most 100 features (to facilitate learning). We ensure comparability between all strategies by employing the same classifier in all experiments, a logistic regression with default meta-parameters. The data acquisition happens in up to 8 batches, each of which acquires 50 new training examples. However, not all strategies reach the last batch on all data sets; we stop each trial as soon as the strategy exhausts one of the classes. We opted for this early stopping criterion to focus on "realistic" acquisitions that happen due to free choices and not due to the fact that our experiment only simulates class-dependent data acquisition with finite pools of data. For the same reason, and due to weak performances on imbalanced data, we did not evaluate the *uniform* strategy here. Due to the early stopping, it becomes increasingly harder to detect significant differences; while the batches three and four can be evaluated on all data sets, only 9 data sets remain for batch eight. The implementation of our configurable experiments is available online.[11]

Fig. 6.8 presents the CD diagrams, as according to our evaluation methodology. We see that our method, with access to an uncertain prior of $p_\mathcal{T}$, performs as well as "proportional" sampling, the privileged strategy that knows $p_\mathcal{T}$ precisely. Moreover, our method outperforms all existing strategies which are oblivious to $p_\mathcal{T}$.

Fig. 6.9 traces this success back to the acquisition behavior that each strategy exhibits. Our own strategy quickly approaches the true proportions $p_\mathcal{T}$ of classes, due to the perfect alignment between the mean of the prior and $p_\mathcal{T}$. For the particular case of a Beta prior, this behavior is a reason for concern: if the mean of this prior was not well aligned with $p_\mathcal{T}$, we might have acquired data in mistaken class proportions; only if the mean of the Beta prior is sufficiently accurate, we can expect the competitive behavior that Fig. 6.8

[11] `https://github.com/mirkobunse/AcsCertificates.jl`

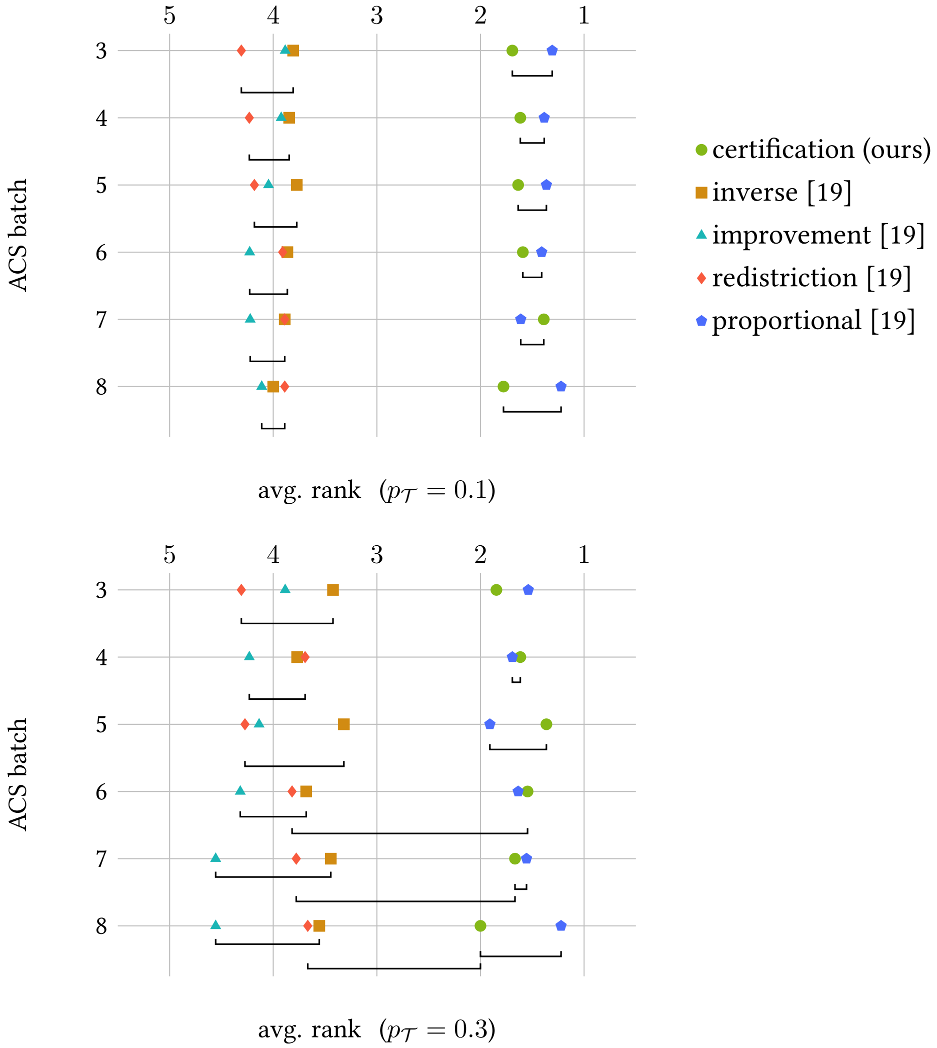

Figure 6.8.: Critical difference (CD) diagrams, see Sec. 2.4.2, evaluate our ACS strategy (●) against existing ACS strategies, one of which has privileged access to the true class proportions $p_{\mathcal{T}}$ (⬟). The two plots present different values of $p_{\mathcal{T}}$. Each position on the vertical axes corresponds to one CD diagram for one batch in the ACS data acquisition loop. Horizontal positions correspond to the average ranks of strategies across multiple data sets, as according to the average accuracy in multiple trials; lower ranks are better. Horizontal connections between two or more strategies indicate that a Wilcoxon signed-rank test is not able to detect significant differences between these methods from the performances they exhibit. This figure is taken from one of our publications [9].

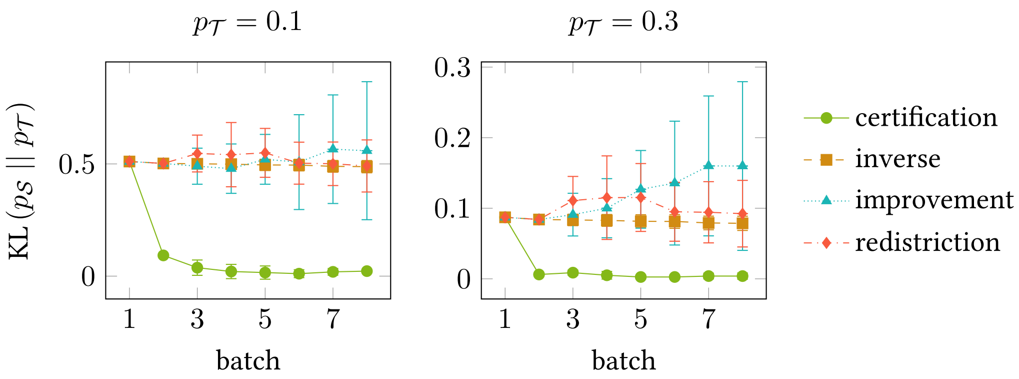

Figure 6.9.: Our ACS strategy (●) quickly approaches the true proportions $p_\mathcal{T}$ of classes in terms of the Kullback-Leibler divergence from Def. 2.4. Due to the uncertainty of the prior, however, this divergence always remains above zero. The standard deviations, which are displayed as error bars, increase considerably with the other strategies. This figure is taken from one of our publications [9].

suggests. Future research down this lane, e.g., with other types of prior distributions, is needed.

Fig. 6.9 further reveals two explanations for the poor performances of the existing strategies: first, all of these strategies exhibit a central tendency of staying close to the class proportions of the initial training set; second, each of these strategies prefers class proportions of an increasingly large variability. Both of these behaviors are due to the sole focus of these strategies on the perceived difficulties of classes, which can differ considerably between the data sets. Our theoretical analyses from Sections 6.3 and 6.4 suggest that a behavior of this kind is not desirable. Our theoretically justified strategy should be preferred over these heuristic strategies.

7. Conclusions

The results of this thesis allow us to draw several conclusions about the machine learning aspects of acquiring knowledge in astro-particle physics. Our scope is defined by the iterative process of knowledge expansion from Fig. 1.1, by the fundamental physics questions from Sec. 3.1, and by the physics analysis tasks from Secs. 3.3–3.5.

The core of this thesis is concerned with the three machine learning tasks, which we have addressed in Chapters 4–6: ordinal quantification, class-conditional label noise learning, and active class selection. Despite their practical relevance for astro-particle physics, our insights into these tasks contribute, first and foremost, to the *fundamental research* of computer science. This contribution stems from the fundamental nature of the three machine learning tasks, which also appear in other areas of application, e.g., in text quantification [96], in the social sciences [97], in support log analyses [93], in brain computer interaction [149], in the calibration of gas sensor arrays [19], and more.

The following Secs. 7.1–7.3 discuss our findings with respect to each of the three machine learning tasks. In Sec. 7.4, we conceive several topics for future work.

7.1. Ordinal Quantification

Our unification of algorithms from literature on quantification and from literature on unfolding demonstrates that quantification and unfolding are indeed the same mathematical problem. Consequently, we can employ quantification algorithms also in astro-particle physics and we can employ unfolding algorithms also in applications outside of astro-particle physics. This claim is supported by our empirical evaluation, where our proposed methods o-ACC and o-PACC, which are ordinally regularized variants of existing quantification methods, provide excellent performances in the reconstruction of energy spectra from the data of the FACT telescope.

Our unification also includes a discussion of multiple multi-class extensions for the binary ACC method, a central method in quantification literature. These extensions differ in terms of the constraints they employ and in terms of their resulting performances. Among the existing multi-class extensions, we consider a numerical optimization of a least squares objective with a unit simplex constraint to be the most appropriate one.

Beyond the existing proposals for numerically solving multi-class quantification, we have proposed a novel, unconstrained optimization task that employs a soft-max layer to circumvent constraints in the optimization procedure. Through this layer, the necessary unit simplex constraint becomes an invariant part of the quantification model, which the optimization algorithm does not have to address, anymore. This soft-max "trick" can be employed in all numerically optimized quantification methods, including also the unfolding methods RUN and SVD. Our experiments suggest that our novel "RUN (softmax)" method, which stems from the soft-max trick, outperforms the original RUN algorithm by a considerable margin. Also, the quantification methods ACC and PACC are shown to benefit from our soft-max proposal.

Unfolding methods employ regularization to promote smooth solutions. We argue that this inductive bias addresses ordinal quantification in general because neighboring classes are typically similar and hence should exhibit similar prevalences. Moreover, smoothness would not be properly defined if the classes were not totally ordered. Due to this conception, we have proposed to introduce ordinality into non-ordinal quantification methods by regularizing their solutions in the same ways in which unfolding methods regularize. To this end, we have proposed o-ACC, o-PACC, o-HDx, o-HDy, and o-SLD, the ordinal counterparts of several widely-acknowledged quantification methods. In our experiments with ordinal data, we have demonstrated that these ordinal counterparts indeed outperform the non-ordinal, original methods in ordinal settings.

These fundamental findings about quantification practically benefit astro-particle physics as a use-case. In particular, astro-particle physicists require precise estimates of the energy spectra of cosmic gamma ray sources to interpret the telescope data in terms of physical theories. Through our advancements of ordinal quantification methods, we have paved the way for more accurate estimations of these energy spectra. Indeed, physicists have confirmed to us that our soft-max improvement of the RUN algorithm and our regularized quantification methods o-ACC, o-PACC, o-HDx, o-HDy, and o-SLD have a great potential of supporting the future expansion of scientific knowledge in their field.

7.2. Class-Conditional Label Noise Learning

We have successfully employed "on" and "off" annotations as a form of weak supervision for learning strong gamma hadron classifiers. In particular, our weakly supervised classifiers even outperform the strongly supervised state-of-the-art in gamma hadron classification in terms of their source detection efficiency, if they are trained on large samples of the real telescope data. With smaller samples, our weakly supervised classifiers already perform on par with the existing state-of-the-art.

While pursuing the goal of supervising models with the "on" and "off" annotations of real telescope data, we have discovered a novel setting of learning under class-conditional label noise, PK-CCN, where one of the class-wise noise rates is known and the other is not. Beyond this assumption, PK-CCN inherits the theoretical properties of the general class-conditional label noise setting. One of these properties is that, indeed, optimal classifiers can be learned from the noisy labels.

In order to establish the feasibility of learning under class-conditional label noise, we have proven that the hypothesis test by Li and Ma [76] is applicable to testing the fundamental assumption of class-conditional label noise learning in general. A practical limitation of this test is, however, that it requires data that is both cleanly and noisily labeled.

We circumvent this limitation with a heuristic that uses the test statistic by Li and Ma as a quality metric. Unlike supervised measures, such as accuracy or the F_1 score, this quality metric does not require clean ground-truth labels; noisy labels suffice for an assessment of model performance. Therefore, we have proposed to employ this heuristic quality metric as an objective function in consistent, noise-aware decision thresholding. Moreover, we have shown that a decision tree induction algorithm can even optimize this metric over the entire model space, without being limited to an optimization of the decision threshold. Both of our algorithms are heuristic, but demonstrate competitive performances on standard imbalanced data sets and in gamma hadron classification.

In demonstrating the feasibility of class-conditional label noise learning from “on” and “off” annotations, we have shown that accurate gamma hadron classifiers can be learned even without any simulated data. For astro-particle physics, this finding translates to a reduced resource footprint of the analyses because the simulations, otherwise, contribute considerably to this footprint. Moreover, by learning directly from the real telescope data, we can *circumvent the domain gap* that otherwise occurs between the real telescope and the simulation. Hence, we can outperform the existing state-of-the-art in gamma hadron classification. Our findings allow for more effective detections of cosmic gamma ray sources. Moreover, the existing theory on class-conditional label noise learning allows us to optimize a large set of quality measures, e.g., the F_1 score or the G measure, through the cheap, noisily labeled data from the real telescope.

7.3. Active Class Selection

We have argued that the free choice of class proportions in active class selection leads to a domain gap in terms of prior probability shift. The severity of this gap depends on the data set and on the amount of data that is generated. In particular, we have seen that small domain gaps have almost no influence on the performance of a classifier, but that ensuring a small domain gap is becoming increasingly essential as the size of the data set increases. In active class selection, this increase of the data set size happens naturally because the whole purpose of this learning task is to generate additional training data. In these terms, we have provided a comprehensive picture of the behavior that can be expected from active class selection techniques.

By developing PAC bounds for active class selection, we have evolved this general picture into a quantitative assessment of prediction performance. This assessment is also applicable in any other learning task that suffers from prior probability shift. Our PAC bounds give rise to a certificate of model robustness, which provides faithful assessments of the model performance by declaring a set of class proportions for which the model is probably and approximately correct. These faithful assessments of model performance are capable of building trust in prediction models, particularly if the models are trained with a free choice of class proportions. Astro-particle physicists have confirmed to us the great practical value of these assessments.

Our PAC bounds also give rise to a novel data acquisition strategy for active class selection. Unlike other strategies, which are mostly heuristic, our strategy is theoretically well-justified. Moreover, it is capable of considering existing uncertainties about the class proportions, which have to be handled during deployment. These uncertainties are ubiquitous in astro-particle physics, making their consideration an essential requirement for any suitable data acquisition strategy.

In astro-particle physics, active class selection has the potential to optimize the acquisition of data from a simulation. Since the simulation is indeed a class-conditional data generator, some decision for the class proportions always has to be made before any data can be generated. Today, physicists employ fixed ad-hoc class proportions that do not match the deployment data, due to the extreme class imbalances that are faced during deployment. Our data acquisition strategy paves the way for a theoretically grounded choice of class proportions, in spite of class imbalances, in spite of uncertainties, and in spite of data acquisition costs.

7.4. Outlook

Our findings pave the way for strengthened and persisted interdisciplinary efforts between computer science and astro-particle physics. In particular, we conceive the following topics for future work.

Regularizations for Other Quantification Tasks We have seen how regularization towards smooth solutions addresses ordinality in quantification. A natural question to ask is whether other regularization schemes can be developed, which address other types of quantification tasks. For instance, an L_1 regularizer could promote sparse solutions in multi-label quantification [172], assuming that only few labels contribute to the quantification outcomes. Another regularizer could penalize some distance to the class prevalences of the training set [113] to address the assumption that only some limited amount of prior probability shift occurs in a particular quantification setting.

Background Subtraction in Quantification The estimation of energy spectra is complicated by the same background events, which also complicate the detection of gamma ray sources [98]. These background events distort the spectrum if they are not properly handled by the quantification method. Since these background events are difficult to synthesize in experiments, we have, for now, not yet considered this issue in our experiments and methods. We aim to fill this gap in the near future, in order to obtain quantification results with proper physical interpretations.

Regarding the methods of quantification, considering background events only requires a slight adaptation of the existing loss functions. In particular, the loss functions have to be extended with fixed terms that represent a measurement of the background.

Multi-Class Certification and Data Acquisition Our certificate of model robustness and our data acquisition strategy for active class selection are currently limited to binary classification. Therefore, we cannot employ them in multi-class settings, although the general ideas of model certification and uncertainty-based data acquisition have a certain potential for being applied to multi-class classification tasks.

As a first step in this direction, we have developed a multi-class generalization of our certificate, which builds on Hölder's inequality [13]. Since other generalizations are conceivable, we are continuing our work on this topic. Recently, we have supervised a Master's thesis, which explores multiple generalizations of this kind.

Non-Decomposable Quality Measures in Active Class Selection Our certificate of model robustness and our data acquisition strategy for active class selection are currently limited to quality measures that can be decomposed into example-wise losses. However, several non-decomposable measures, see Tab. 2.1, can be optimized through an appropriately chosen decision threshold. We expect that these other measures also lend themselves for being optimized through active class selection because the choice of a decision threshold bears strong similarity to a weighting of the classes in terms of actively chosen class proportions. Therefore, we intend to extend our certificate and our acquisition strategy to the non-decomposable quality measures from Tab. 2.1.

Unsupervised Domain Adaptation Our identical mechanism assumption is a slight simplification because the real telescope and the simulation are actually different data generating processes. One of our ongoing efforts is to address this simplification through unsupervised domain adaptation [50], [173], [174], a learning task that handles settings of this kind. So far, we have not been able to considerably improve our prediction models through domain adaptation techniques. On the one hand, this momentary inability actually supports our claim that the identical mechanism assumption is sufficiently accurate for our use case. On the other hand, we are continuing our work on this topic to ensure that we leverage domain adaptation to its full potential.

A. Covered Publications

The foundation of this thesis are several scientific publications by us. The following, commented bibliography gives an overview of these publications. If a publication emerged from the collaboration with other students, we detail the contributions of all students that were involved.

Ordinal Quantification

We have covered our publications on the unification of quantification algorithms [2], on the unification of unfolding algorithms [1], [6], on multi-class extensions of Adjusted Classify and Count [3], and on regularization for ordinal quantification [5].

Our early works on the unification of unfolding algorithms [1], [6] emerged as extensions of Mirko Bunse's Master's thesis, who was thus the main author of these papers. The other authors contributed their feedback to the manuscripts.

Our work on the unification of quantification algorithms [2] was taken out on a research visit of Mirko in Pisa, Italy. Martin Senz implemented a part of the experiments.

Regarding our submission to the LeQua competition [4], Mirko suggested to participate with o-SLD, a quantification method that Mirko already had implemented. Martin Senz implemented all experiments for the competition, all experiments for the paper, and he wrote the manuscript. Mirko gave feedback on these matters.

[1] M. Bunse, N. Piatkowski, K. Morik, T. Ruhe, and W. Rhode, "Unification of deconvolution algorithms for Cherenkov astronomy," in *Int. Conf. on Data Sci. and Adv. Analyt.*, 2018, pp. 21–30. DOI: `10.1109/DSAA.2018.00012`.

[2] M. Bunse, "Unification of algorithms for quantification and unfolding," in *Worksh. on Mach. Learn. for Astropart. Phys. and Astron.*, 2022, pp. 459–468. DOI: `10.18420/INF2022_37`.

[3] M. Bunse, "On multi-class extensions of adjusted classify and count," in *Int. Worksh. on Learn. to Quantify: Meth. and Appl.*, 2022, pp. 43–50.

[4] M. Senz and M. Bunse, "DortmundAI at LeQua 2022: Regularized SLD," in *Conf. and Labs of the Eval. Forum*, 2022, pp. 1911–1915.

[5] M. Bunse, A. Moreo, F. Sebastiani, and M. Senz, "Ordinal quantification through regularization," in *Europ. Conf. on Mach. Learn. and Knowl. Discov. in Databases*, Accepted for publication, 2022.

[6] M. Bunse, N. Piatkowski, and K. Morik, “Towards a unifying view on deconvolution in Cherenkov astronomy,” in *Lernen, Wissen, Daten, Analysen*, 2018, pp. 73–77.

Class-Conditional Label Noise Learning

Our chapter on class-conditional label noise is based on a manuscript that is currently under review. An earlier version of this manuscript [7] is already published on the arXiv pre-print repository.

Our manuscript is based on Lukas Pfahler’s initial idea of using the Li&Ma significance as a learning criterion for machine learning models. Our joint discussions established that the “on” and “off” annotations, which the approach employs, effectively serve as class-conditional noisy labels. Due to this finding, Mirko Bunse developed our baselines, which use existing approaches for class-conditional label noise learning. Mirko also developed our theoretical re-framing of the Li&Ma test, which establishes this test as a hypothesis test for the feasibility of class-conditional label noise learning in general. Our implementations of the experiments and methods are joint work.

[7] L. Pfahler, M. Bunse, and K. Morik, “Noisy labels for weakly supervised gamma hadron classification,” *CoRR*, 2021.

Active Class Selection

We have covered our publications on a meta-study of existing experiments [10], on a distinction between active class selection and active learning [11], on our information-theoretic investigation [8], on the certification of model robustness [12], on an uncertain strategy for data acquisition [9], and on a multi-class extension of our certificate [13]. These works are preceded by an early, empirical study on data acquisition strategies in astro-particle physics [14].

Our distinction between active class selection and active learning [11] evolved when Amal Saadallah and Mirko Bunse tried to bring together their previous work on machine learning from simulated data. In our joint discussions, we found that our respective areas of application use entirely different simulation mechanisms, which amount to the difference between active learning and active class selection. The contributions by Amal and Mirko are equal.

The initial plan of the collaboration between Dorina Weichert and Mirko was to evaluate Bayesian optimization techniques for active class selection. However, we were not able to achieve convincing results from this conception. As a consequence, Mirko developed the information-theoretic analysis and the experiments, which became the basis of our joint publication [8]. Dorina, Mirko, and Alexander Kister intensively discussed the implications of this analysis, answered potential questions, and clarified the story line of our paper. Dorina Weichert contributed an extensive literature survey to the manuscript.

Our work on multi-class certification [13] stems from the Master's thesis of Martin Senz, who wrote the manuscript and implemented all experiments on this topic. Mirko contributed the seminal suggestion of employing Hölder's inequality as a foundation of our multi-class extension.

[8] M. Bunse, D. Weichert, A. Kister, and K. Morik, "Optimal probabilistic classification in active class selection," in *Int. Conf. on Data Mining*, 2020, pp. 942–947. DOI: 10.1109/ICDM50108.2020.00106.

[9] M. Bunse and K. Morik, "Active class selection with uncertain deployment class proportions," in *Worksh. on Interact. Adapt. Learn.*, 2021, pp. 70–79.

[10] M. Bunse and K. Morik, "What can we expect from active class selection?" In *Lernen, Wissen, Daten, Analysen*, 2019, pp. 79–83.

[11] M. Bunse, A. Saadallah, and K. Morik, "Towards active simulation data mining," in *Int. Tutorial and Worksh. on Interact. Adapt. Learn.*, 2019, pp. 104–107.

[12] M. Bunse and K. Morik, "Certification of model robustness in active class selection," in *Europ. Conf. on Mach. Learn. and Knowl. Discov. in Databases*, 2021, pp. 266–281.

[13] M. Senz, M. Bunse, and K. Morik, "Certifiable active class selection in multi-class classification," in *Worksh. on Interact. Adapt. Learn.*, 2022, pp. 68–76.

[14] M. Bunse, C. Bockermann, J. Buss, K. Morik, W. Rhode, and T. Ruhe, "Smart control of monte carlo simulations for astroparticle physics," in *Astron. Data Analys. Softw. and Syst.*, Astron. Society of the Pacific, 2017, pp. 417–420.

List of Figures

List of Tables

List of Footnotes and Online Resources

Bibliography

[15] B. Falkenburg and W. Rhode, *From Ultra Rays to Astroparticles: A Historical Introduction to Astroparticle Physics*. Springer, 2012. DOI: 10.1007/978-94-007-5422-5.

[16] C. Bockermann, K. Brügge, J. Buss, *et al.*, "Online analysis of high-volume data streams in astroparticle physics," in *Europ. Conf. on Mach. Learn. and Knowl. Discov. in Databases*, 2015, pp. 100–115. DOI: 10.1007/978-3-319-23461-8_7.

[17] S. Schmitt, "Data unfolding methods in high energy physics," in *EPJ Web of Conf.*, EDP Sci., 2017.

[18] A. K. Menon, B. van Rooyen, C. S. Ong, and R. C. Williamson, "Learning from corrupted binary labels via class-probability estimation," in *Int. Conf. on Mach. Learn.*, 2015, pp. 125–134.

[19] R. Lomasky, C. E. Brodley, M. Aernecke, D. Walt, and M. A. Friedl, "Active class selection," in *Europ. Conf. on Mach. Learn.*, 2007, pp. 640–647. DOI: 10.1007/978-3-540-74958-5_63.

[20] L. M. Linhoff, "Multiwavelength analysis of the TeV-radio galaxy 3C 84/NGC 1275," Ph.D. dissertation, TU Dortmund Univ., 2021. DOI: 10.17877/DE290R-22408.

[21] J. B. Buß, "Bad moon rising? Studies on the performance of the first G-APD Cherenkov telescope under bright light conditions using SiPMs for gamma-ray observations," Ph.D. dissertation, TU Dortmund Univ., 2020. DOI: 10.17877/DE290R-21918.

[22] M. Nöthe, "Monitoring the high energy universe," Ph.D. dissertation, TU Dortmund Univ., 2020. DOI: 10.17877/DE290R-21143.

[23] K. Brügge, "Unmasking the gamma-ray sky: Comprehensive and reproducible analysis for Cherenkov telescopes," Ph.D. dissertation, TU Dortmund Univ., 2019. DOI: 10.17877/DE290R-20440.

[24] W. Rhode, "On probabilistic rationalism," TU Dortmund Univ., Tech. Rep., 2020.

[25] A. Saadallah, F. Finkeldey, K. Morik, and P. Wiederkehr, "Stability prediction in milling processes using a simulation-based machine learning approach," in *CIRP Conf. on Manuf. Syst.*, 2018. DOI: 10.1016/j.procir.2018.03.062.

[26] A. Saadallah, A. Egorov, B.-T. Cao, S. Freitag, K. Morik, and G. Meschke, "Active learning for accurate settlement prediction using numerical simulations in mechanized tunneling," *Procedia CIRP*, pp. 1052–1058, 2019.

[27] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," *Comput. Materials Sci.*, pp. 171–180, 2017.

[28] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," *Nature Commun.*, pp. 1–10, 2018.

[29] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning - From Theory to Algorithms.* Cambridge Univ. Press, 2014, ISBN: 978-1-10-705713-5.

[30] A. Buja, W. Stuetzle, and Y. Shen, "Loss functions for binary class probability estimation and classification: Structure and applications," Univ. of Pennsylvania, Tech. Rep., 2005.

[31] O. Koyejo, N. Natarajan, P. Ravikumar, and I. S. Dhillon, "Consistent binary classification with generalized performance metrics," in *Adv. in Neur. Inform. Process. Syst.*, 2014, pp. 2744–2752.

[32] H. Narasimhan, R. Vaish, and S. Agarwal, "On the statistical consistency of plug-in classifiers for non-decomposable performance measures," in *Adv. in Neur. Inform. Process. Syst.*, 2014, pp. 1493–1501.

[33] A. K. Menon, H. Narasimhan, S. Agarwal, and S. Chawla, "On the statistical consistency of algorithms for binary classification under class imbalance," in *Int. Conf. on Mach. Learn.*, 2013, pp. 603–611.

[34] N. Ye, K. M. A. Chai, W. S. Lee, and H. L. Chieu, "Optimizing F-measure: A tale of two approaches," in *Int. Conf. on Mach. Learn.*, 2012.

[35] S.-S. Choi, S.-H. Cha, and C. C. Tappert, "A survey of binary similarity and distance measures," *J. of Systemics, Cybernetics and Informatics*, pp. 43–48, 2010.

[36] L. Breiman, "Random forests," *Mach. Learn.*, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.

[37] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* Wadsworth, 1984, ISBN: 0-534-98053-8.

[38] M. Aartsen, M. Ackermann, J. Adams, *et al.*, "Development of a general analysis and unfolding scheme and its application to measure the energy spectrum of atmospheric neutrinos with IceCube," *Europ. Phys. J. C*, 2015. DOI: 10.1140/epjc/s10052-015-3330-z.

[39] R. Bock, A. Chilingarian, M. Gaug, *et al.*, "Methods for multidimensional event classification: A case study using images from a Cherenkov gamma-ray telescope," *Nucl. Inst. and Meth. in Phys. Res. Sec. A*, pp. 511–528, 2004. DOI: 10.1016/j.nima.2003.08.157.

[40] F. Temme, M. Noethe, R. Walter, *et al.*, "FACT – first energy spectrum from a SiPM Cherenkov telescope," in *Int. Cosmic Ray Conf.*, 2016. DOI: 10.22323/1.236.0707.

[41] K. Berger, T. Bretz, D. Dorner, D. Hoehne, and B. Riegel, "A robust way of estimating the energy of a gamma ray shower detected by the MAGIC telescope," in *Int. Cosmic Ray Conf.*, 2005, pp. 100–104.

[42] L. Breiman, "Bagging predictors," *Mach. Learn.*, pp. 123–140, 1996. DOI: 10.1007/BF00058655.

[43] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, pp. 119–139, 1997. DOI: 10.1006/jcss.1997.1504.

[44] B. Khelifi, A. Djannati-Ataï, L. Jouvin, *et al.*, "HAP-Fr, a pipeline of data analysis for the HESS-II experiment," in *Int. Cosmic Ray Conf.*, 2016. DOI: 10.22323/1.236.0837.

[45] M. Krause, E. Pueschel, and G. Maier, "Improved γ/hadron separation for the detection of faint γ-ray sources using boosted decision trees," *Astropart. Phys.*, pp. 1–9, 2017.

[46] M. Helf, "Gamma-Hadron-Separation im MAGIC-Experiment durch verteilungsgestütztes Sampling," M.S. thesis, TU Dortmund Univ., 2011.

[47] M. Scholz, "Scalable and accurate knowledge discovery in real-world databases," Ph.D. dissertation, TU Dortmund Univ., 2007.

[48] G. M. Weiss and F. J. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *J. Artif. Intell. Res.*, pp. 315–354, 2003. DOI: 10.1613/jair.1199.

[49] H. Anderhub, M. Backes, A. Biland, *et al.*, "Design and operation of FACT–the first G-APD Cherenkov telescope," *J. Inst.*, 2013. DOI: 10.1088/1748-0221/8/06/p06008.

[50] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomput.*, pp. 135–153, 2018. DOI: 10.1016/j.neucom.2018.05.083.

[51] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, pp. 1345–1359, 2010. DOI: 10.1109/TKDE.2009.191.

[52] J. G. Moreno-Torres, T. Raeder, R. Alaíz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognit.*, pp. 521–530, 2012. DOI: 10.1016/j.patcog.2011.06.019.

[53] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *Int. Conf. on Mach. Learn.*, 2013, pp. 819–827.

[54] T. Hastie, J. H. Friedman, and R. Tibshirani, *The elements of statistical learning: Data mining, inference, and prediction.* Springer, 2001, ISBN: 978-1-4899-0519-2. DOI: 10.1007/978-0-387-21606-5.

[55] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, pp. 1–30, 2006.

[56] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?" *J. Mach. Learn. Res.*, 5:1–5:10, 2016.

[57] T. M. Cover and J. A. Thomas, *Elements of information theory (2. ed.)* Wiley, 2006, ISBN: 978-0-471-24195-9.

[58] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Int. Conf. on Comput. Vis.*, 1998, pp. 59–66. DOI: 10.1109/ICCV.1998.710701.

[59] M. Werman, S. Peleg, and A. Rosenfeld, "A distance metric for multidimensional histograms," *Comput. Vis. Graph. Image Process.*, pp. 328–336, 1985. DOI: 10.1016/0734-189X(85)90055-6.

[60] T. Sakai, "Comparing two binned probability distributions for information access evaluation," in *Ann. Int. ACM SIGIR Conf. on Res. and Dev. in Inform. Retr.*, 2018, pp. 1073–1076. DOI: 10.1145/3209978.3210073.

[61] E. Lorenz and R. Wagner, "Very-high energy gamma-ray astronomy: A 23-year success story in astroparticle physics," in *From Ultra Rays to Astroparticles: A Historical Introduction to Astroparticle Physics.* 2012, ch. 6, pp. 143–185. DOI: 10.1007/978-94-007-5422-5.

[62] T. C. Weekes, M. F. Cawley, D. Fegan, *et al.*, "Observation of TeV gamma-rays from the Crab nebula using the atmospheric Cherenkov imaging technique," *Astrophys. J.*, pp. 379–395, 1989.

[63] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, *et al.*, "Observation of gravitational waves from a binary black hole merger," *Phys. Rev. Lett.*, 2016. DOI: 10.1103/physrevlett.116.061102.

[64] B. S. Acharya, I. Agudo, I. Al Samarai, R. Alfaro, J. Alfaro, C. Alispach, *et al.*, *Science with the Cherenkov telescope array.* World Sci., 2018.

[65] V. F. Hess, "Über Beobachtungen der durchdringenden Strahlung bei sieben Freiballonfahrten," *Physikalische Zeitschrift*, pp. 1084–1091, 1912.

[66] S. Funk, "Ground- and space-based gamma-ray astronomy," *Ann. Rev. of Nucl. and Part. Sci.*, pp. 245–277, 2015. DOI: 10.1146/annurev-nucl-102014-022036.

[67] S. P. Wakely and D. Horan, "TeVCat: An online catalog for very high energy gamma-ray astronomy," in *Int. Cosmic Ray Conf.*, 2008, pp. 1341–1344.

[68] M. de Naurois, "H.E.S.S.-II - gamma ray astronomy from 20 GeV to hundreds of TeV´s," in *Int. Conf. on Astopart. Phys.*, 2017. DOI: 10.1051/epjconf/201713603001.

[69] J. Hinton, "The status of the HESS project," *New Astron. Rev.*, pp. 331–337, 2004. DOI: 10.1016/j.newar.2003.12.004.

[70] D. B. Tridon, T. Schweizer, F. Goebel, R. Mirzoyan, M. Teshima, *et al.*, "The MAGIC-II gamma-ray stereoscopic telescope system," *Nucl. Inst. and Meth. in Phys. Res. Sec. A*, pp. 437–439, 2010. DOI: 10.1016/j.nima.2010.03.028.

[71] J. Holder, "VERITAS: Status and highlights," in *Int. Cosmic Ray Conf.*, The DOI for this reference is currently broken (10.7529/ICRC2011/V12/H11), 2011.

[72] M. Actis, G. Agnetta, F. Aharonian, *et al.*, "Design concepts for the Cherenkov telescope array CTA: An advanced facility for ground-based high-energy gamma-ray astronomy," *Exper. Astron.*, pp. 193–316, 2011. DOI: 10.1007/s10686-011-9247-0.

[73] V. Blobel, "An unfolding method for high energy physics experiments," in *Adv. Stat. Tech. in Part. Phys.*, 2002, pp. 258–267.

[74] G. D'Agostini, "A multidimensional unfolding method based on Bayes' theorem," *Nucl. Instr. and Meth. in Phys. Res. Sect. A*, pp. 487–498, 1995.

[75] D. Tasche, "Fisher consistency for prior probability shift," *CoRR*, 2017.

[76] T.-P. Li and Y.-Q. Ma, "Analysis methods for results in gamma-ray astronomy," *Astrophys. J.*, pp. 317–324, 1983.

[77] S. Buschjäger, L. Pfahler, J. Buß, K. Morik, and W. Rhode, "On-site gamma-hadron separation with deep learning on FPGAs," in *Europ. Conf. on Mach. Learn. and Knowl. Discov. in Databases: Appl. Data Sci. Track*, 2020, pp. 478–493. DOI: 10.1007/978-3-030-67667-4_29.

[78] D. Heck, G. Schatz, T. Thouw, J. Knapp, and J. N. Capdevielle, "CORSIKA: A monte carlo code to simulate extensive air showers," Forschungszentrum Karlsruhe, Tech. Rep., 1998.

[79] W. Nelson and Y. Namito, "The EGS4 code system: Solution of gamma-ray and electron transport problems," Stanford Linear Accelerator Center, Tech. Rep., 1990.

[80] S. A. Bass, M. Belkacem, M. Bleicher, *et al.*, "Microscopic models for ultrarelativistic heavy ion collisions," *Progress in Part. and Nucl. Phys.*, pp. 255–369, 1998. DOI: 10.1016/S0146-6410(98)00058-1.

[81] T. Pierog, I. Karpenko, J. M. Katzy, E. Yatsenko, and K. Werner, "EPOS LHC: Test of collective hadronization with data measured at the CERN large hadron collider," *Phys. Rev. C*, 2015. DOI: 10.1103/physrevc.92.034906.

[82] K. Bernlöhr, "Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_telarray," *Astropart. Phys.*, pp. 149–158, 2008. DOI: 10.1016/j.astropartphys.2008.07.009.

[83] T. Bretz and D. Dorner, "MARS-CheObs goes monte carlo," in *Int. Cosm. Ray Conf.*, 2009.

[84] Y. Shao, Y. Liu, X. Ye, and S. Zhang, "A machine learning based global simulation data mining approach for efficient design changes," *Adv. in Engin. Softw.*, pp. 22–41, 2018. DOI: 10.1016/j.advengsoft.2018.07.002.

[85] T. F. Brady and E. Yellig, "Simulation data mining: A new form of computer simulation output," in *Winter Simulat. Conf.*, 2005, pp. 285–289. DOI: 10.1109/WSC.2005.1574262.

[86] S. Burrows, B. Stein, J. Frochte, D. Wiesner, and K. Müller, "Simulation data mining for supporting bridge design," in *Austral. Data Mining Conf.*, 2011, pp. 163–170.

[87] H. Trittenbach, M. Gauch, K. Böhm, and K. Schulz, "Towards simulation-data science – a case study on material failures," in *Int. Conf. on Data Sci. and Adv. Analyt.*, 2018, pp. 450–459. DOI: 10.1109/DSAA.2018.00058.

[88] C. Bockermann, "Mining big data streams for multiple concepts," Ph.D. dissertation, TU Dortmund Univ., 2015. DOI: 10.17877/DE290R-16437.

[89] J. Aleksić, S. Ansoldi, L. Antonelli, P. Antoranz, A. Babic, *et al.*, "The major upgrade of the MAGIC telescopes, part II: A performance study using observations of the Crab nebula," *Astropart. Phys.*, pp. 76–94, 2016. DOI: 10.1016/j.astropartphys.2015.02.005.

[90] A. M. Hillas, "Cerenkov light images of EAS produced by primary gamma rays and by nuclei," in *Int. Cosm. Ray Conf.*, 1985.

[91] I. J. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning*. MIT Press, 2016, ISBN: 978-0-262-03561-3.

[92] K. R. Popper, "The logic of scientific discovery," 1934.

[93] G. Forman, "Quantifying counts and costs via classification," *Data Mining and Knowl. Discov.*, pp. 164–206, 2008. DOI: 10.1007/s10618-008-0097-y.

[94] P. González, A. Castaño, N. V. Chawla, and J. J. del Coz, "A review on quantification learning," *ACM Comput. Surv.*, 74:1–74:40, 2017. DOI: 10.1145/3117807.

[95] Y. Yin, X. Wang, Q. Li, P. Shang, and F. Hou, "Quantifying interdependence using the missing joint ordinal patterns," *Chaos: An Interdisc. J. of Nonlin. Sci.*, Jul. 2019. DOI: 10.1063/1.5084034.

[96] W. Gao and F. Sebastiani, "From classification to quantification in tweet sentiment analysis," *Soc. Netw. Analys. and Mining*, pp. 1–22, 2016.

[97] D. J. Hopkins and G. King, "A method of automated nonparametric content analysis for social science," *Amer. J. of Polit. Sci.*, pp. 229–247, 2010. DOI: 10.1111/j.1540-5907.2009.00428.x.

[98] V. Blobel, "Unfolding methods in high-energy physics experiments," CERN, Tech. Rep., 1985. DOI: 10.5170/CERN-1985-009.88.

[99] A. Hoecker and V. Kartvelishvili, "SVD approach to data unfolding," *Nucl. Instr. and Meth. in Phys. Res. Sect. A*, pp. 469–481, 1996.

[100] A. Firat, "Unified framework for quantification," *CoRR*, 2016.

[101] M. Börner, T. Hoinka, M. Meier, T. Menne, W. Rhode, and K. Morik, "Measurement/simulation mismatches and multivariate data discretization in the machine learning era," in *Astron. Data Analys. Softw. and Syst.*, Astron. Society of the Pacific, 2017, pp. 431–434.

[102] P. González, A. Castaño, E. E. Peacock, J. Díez, J. J. Del Coz, and H. M. Sosik, "Automatic plankton quantification using deep features," *J. of Plankton Res.*, pp. 449–463, 2019, ISSN: 0142-7873. DOI: 10.1093/plankt/fbz023.

[103] G. D'Agostini, "Improved iterative Bayesian unfolding," *CoRR*, 2010.

[104] A. Bella, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana, "Quantification via probability estimators," in *Int. Conf. on Data Mining*, 2010, pp. 737–742. DOI: 10.1109/ICDM.2010.75.

[105] V. González-Castro, R. Alaíz-Rodríguez, and E. Alegre, "Class distribution estimation based on the Hellinger distance," *Inform. Sci.*, pp. 146–164, 2013. DOI: 10.1016/j.ins.2012.05.028.

[106] M. G. Aartsen, M. Ackermann, J. Adams, *et al.*, "Measurement of the ν_μ energy spectrum with IceCube-79," *Europ. Phys. J. C*, 2017. DOI: 10.1140/epjc/s10052-017-5261-3.

[107] M. Nöthe, J. Adam, M. L. Ahnen, *et al.*, "FACT – performance of the first Cherenkov telescope observing with SiPMs," in *Int. Cosmic Ray Conf.*, 2018. DOI: https://doi.org/10.3929/ethz-b-000315180.

[108] S. J. Wright and J. Nocedal, *Numerical optimization*, 2nd ed. Springer, 2006.

[109] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Programming*, pp. 25–57, 2006.

[110] G. Aad, B. Abbott, D. C. Abbott, *et al.*, "Measurements of the inclusive and differential production cross sections of a top-quark–antiquark pair in association with a Z boson at $\sqrt{s} = 13$ TeV with the ATLAS detector," *Europ. Phys. J. C*, 2021. DOI: 10.1140/epjc/s10052-021-09439-4.

[111] B. Nachman, M. Urbanek, W. A. de Jong, and C. W. Bauer, "Unfolding quantum computer readout noise," *npj Quantum Inform.*, 2020. DOI: 10.1038/s41534-020-00309-7.

[112] M. Bunse, "DSEA rock-solid – regularization and comparison with other deconvolution algorithms," Master's thesis, TU Dortmund Univ., 2018.

[113] M. Schmelling, "The method of reduced cross-entropy – a general approach to unfold probability distributions," *Nucl. Instr. and Meth. in Phys. Res. Sec. A*, pp. 400–412, 1994.

[114] S. Schmitt, "TUnfold, an algorithm for correcting migration effects in high energy physics," *J. Inst.*, 2012. DOI: `10.1088/1748-0221/7/10/t10003`.

[115] T. Ruhe, M. Schmitz, T. Voigt, and M. Wornowizki, "DSEA: A data mining approach to unfolding," in *Int. Cosmic Ray Conf.*, 2013.

[116] S. Vucetic and Z. Obradovic, "Classification on data with biased class distribution," in *Europ. Conf. on Mach. Learn.*, 2001, pp. 527–538. DOI: `10.1007/3-540-44795-4_45`.

[117] G. J. McLachlan, *Discriminant analysis and statistical pattern recognition.* Wiley, 1992, ISBN: 0-471-69115-1.

[118] J. L. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications.* SIAM, 2012, ISBN: 978-1-61197-233-7. DOI: `10.1137/1.9781611972344`.

[119] A. Moreo, A. Esuli, and F. Sebastiani, "QuaPy: A publicly available python-based software library for quantification," in *Worksh. of the Int. Conf. on Inform. and Knowl. Management*, 2021.

[120] T. Amemiya, *Advanced econometrics.* Blackwell, 1985, ISBN: 9780631133452.

[121] A. Esuli, A. Moreo, and F. Sebastiani, *Learning to quantify: LeQua 2022 datasets*, 2021. DOI: `10.5281/zenodo.6546188`.

[122] A. Esuli, A. Moreo, and F. Sebastiani, "LeQua@CLEF2022: Learning to quantify," in *Europ. Conf. on Inform. Retr.*, to appear, 2022.

[123] M. Saerens, P. Latinne, and C. Decaestecker, "Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure," *Neural Comput.*, pp. 21–41, 2002. DOI: `10.1162/089976602753284446`.

[124] G. Da San Martino, W. Gao, and F. Sebastiani, "Ordinal text quantification," in *Ann. Int. ACM SIGIR Conf. on Res. and Dev. in Inform. Retr.*, 2016, pp. 937–940. DOI: `10.1145/2911451.2914749`.

[125] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "SemEval-2016 task 4: Sentiment analysis in twitter," in *Int. Worksh. on Semantic Eval.*, 2016, pp. 1–18. DOI: `10.18653/v1/s16-1001`.

[126] A. Esuli, "ISTI-CNR at semeval-2016 task 4: Quantification on an ordinal scale," in *Int. Worksh. on Semantic Eval.*, 2016, pp. 92–95. DOI: `10.18653/v1/s16-1011`.

[127] J. Aleksić *et al.*, "Measurement of the crab nebula spectrum over three decades in energy with the MAGIC telescopes," *J. of High Energy Astrophys.*, pp. 30–38, 2015. DOI: `10.1016/j.jheap.2015.01.002`.

[128] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Ann. Int. ACM SIGIR Conf. on Res. and Dev. in Inform. Retr.*, 2015, pp. 43–52. DOI: `10.1145/2766462.2767755`.

[129] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, 2019.

[130] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Adv. in Neur. Inform. Process. Syst.*, 2013, pp. 1196–1204.

[131] C. Scott, G. Blanchard, and G. Handy, "Classification with asymmetric label noise: Consistency and maximal denoising," in *Annual Conf. on Learn. Theory*, 2013, pp. 489–511.

[132] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *Conf. on Comput. Learn. Theory*, 1998, pp. 92–100. DOI: 10.1145/279943.279962.

[133] A. Ghosh, N. Manwani, and P. S. Sastry, "Making risk minimization tolerant to label noise," *Neurocomput.*, pp. 93–107, 2015. DOI: 10.1016/j.neucom.2014.09.081.

[134] V. Mithal, G. Nayak, A. Khandelwal, V. Kumar, N. C. Oza, and R. R. Nemani, "RAPT: rare class prediction in absence of true labels," *IEEE Trans. Knowl. Data Eng.*, pp. 2484–2497, 2017. DOI: 10.1109/TKDE.2017.2739739.

[135] A. K. Menon, B. van Rooyen, and N. Natarajan, "Learning from binary labels with instance-dependent noise," *Mach. Learn.*, pp. 1561–1595, 2018. DOI: 10.1007/s10994-018-5715-3.

[136] C. G. Northcutt, L. Jiang, and I. L. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *J. Artif. Intell. Res.*, pp. 1373–1411, 2021. DOI: 10.1613/jair.1.12125.

[137] Y. Yao, T. Liu, B. Han, *et al.*, "Dual T: Reducing estimation error for transition matrix in label-noise learning," in *Adv. in Neur. Inform. Process. Syst.*, 2020.

[138] X. Ma, H. Huang, Y. Wang, S. Romano, S. M. Erfani, and J. Bailey, "Normalized loss functions for deep learning with noisy labels," in *Int. Conf. on Mach. Learn.*, 2020, pp. 6543–6553.

[139] B. Han, Q. Yao, X. Yu, *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Adv. in Neur. Inform. Process. Syst.*, 2018, pp. 8536–8546.

[140] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Conf. on Comput. Vis. and Pattern Recogn.*, 2017, pp. 2233–2241. DOI: 10.1109/CVPR.2017.240.

[141] X. Xia, T. Liu, N. Wang, *et al.*, "Are anchor points really indispensable in label-noise learning?" In *Adv. in Neur. Inform. Process. Syst.*, 2019, pp. 6835–6846.

[142] X. Li, T. Liu, B. Han, G. Niu, and M. Sugiyama, "Provably end-to-end label-noise learning without anchor points," in *Int. Conf. on Mach. Learn.*, 2021, pp. 6403–6413.

[143] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Springer, 2018, ISBN: 978-3-319-98073-7. DOI: 10.1007/978-3-319-98074-4.

[144] F. Aharonian, A. Akhperjanian, A. Bazer-Bachi, *et al.*, "Observations of the Crab nebula with HESS," *Astron. & Astrophys.*, pp. 899–915, 2006.

[145] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, 17:1–17:5, 2017.

[146] D. Kottke, G. Krempl, M. Stecklina, *et al.*, "Probabilistic active learning for active class selection," in *Proc. of the NeurIPS Worksh. on the Future of Interact. Learn. Mach.*, 2016.

[147] B. Settles, *Active Learning*. Morgan & Claypool Publishers, 2012. DOI: 10.2200/S00429ED1V01Y201207AIM018.

[148] T. D. Parsons and J. L. Reinebold, "Adaptive virtual environments for neuropsychological assessment in serious games," *IEEE Trans. Consumer Electron.*, pp. 197–204, 2012. DOI: 10.1109/TCE.2012.6227413.

[149] D. Wu, B. J. Lance, and T. D. Parsons, "Collaborative filtering for brain-computer interaction using transfer learning and active class selection," *PloS one*, 2013. DOI: 10.1371/journal.pone.0056624.

[150] I. Hossain, A. Khosravi, and S. Nahavandi, "Weighted informative inverse active class selection for motor imagery brain computer interface," in *Canad. Conf. on Electr. and Comput. Engin.*, 2017, pp. 1–5. DOI: 10.1109/CCECE.2017.7946613.

[151] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *SIGKDD Explor.*, pp. 40–49, 2004. DOI: 10.1145/1007730.1007737.

[152] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Analys.*, pp. 429–449, 2002.

[153] G. Weiss and F. Provost, "The effect of class distribution on classifier learning: An empirical study," Dept. of Comput. Sci., Rutgers Univ., Tech. Rep., 2001.

[154] M. Cakmak and A. L. Thomaz, "Designing robot learners that ask good questions," in *Int. Conf. on Human-Robot Interact.*, 2012, pp. 17–24. DOI: 10.1145/2157689.2157693.

[155] D. Dua and C. Graff, *UCI machine learning repository*, 2019.

[156] C. Brodley and M. Friedl, "Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data," in *Int. Geosci. and Remote Sensing Symposium*, Citeseer, 1996, pp. 1382–1384.

[157] W. W. Cohen and Y. Singer, "A simple, fast, and effective rule learner," in *National Conf. on Artif. Intell.*, 1999, p. 3.

[158] J. Attenberg and F. J. Provost, "Inactive learning? difficulties employing active learning in practice," *SIGKDD Explor.*, pp. 36–41, 2010. DOI: 10.1145/1964897.1964906.

[159] A. Mendizabal, T. Fountoukidou, J. Hermann, R. Sznitman, and S. Cotin, "A combined simulation and machine learning approach for image-based force classification during robotized intravitreal injections," in *Med. Image Comput. and Comput. Assisted Intervention*, 2018, pp. 12–20. DOI: 10.1007/978-3-030-00937-3_2.

[160] L. Wang and M. Marek-Sadowska, "Machine learning in simulation-based analysis," in *Int. Symposium on Phys. Design*, 2015, pp. 57–64. DOI: 10.1145/2717764.2717786.

[161] D. Kottke, G. Krempl, D. Lang, J. Teschner, and M. Spiliopoulou, "Multi-class probabilistic active learning," in *Europ. Conf. on Artif. Intell.*, 2016, pp. 586–594. DOI: 10.3233/978-1-61499-672-9-586.

[162] J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Adv. in Neur. Inform. Process. Syst.*, 2004, pp. 513–520.

[163] O. Chapelle, "Active learning for parzen window classifier," in *Int. Worksh. on Artif. Intell. and Stat.*, 2005.

[164] M. Mitchell, S. Wu, A. Zaldivar, *et al.*, "Model cards for model reporting," in *Conf. on Fairness, Accountability, and Transparency*, 2019, pp. 220–229. DOI: 10.1145/3287560.3287596.

[165] M. Arnold, R. K. E. Bellamy, M. Hind, *et al.*, "Factsheets: Increasing trust in AI services through supplier's declarations of conformity," *IBM J. Res. Dev.*, 6:1–6:13, 2019. DOI: 10.1147/JRD.2019.2942288.

[166] I. D. Raji, A. Smart, R. N. White, *et al.*, "Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing," in *Conf. on Fairness, Accountability, and Transparency*, 2020, pp. 33–44. DOI: 10.1145/3351095.3372873.

[167] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Adv. in Neur. Inform. Process. Syst.*, 2006, pp. 601–608.

[168] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," in *Adv. in Neur. Inform. Process. Syst.*, 2020.

[169] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. T. Vechev, "Fast and effective robustness certification," in *Adv. in Neur. Inform. Process. Syst.*, 2018, pp. 10 825–10 836.

[170] D. Zhang, M. Ye, C. Gong, Z. Zhu, and Q. Liu, "Black-box certification with randomized smoothing: A functional optimization based framework," in *Adv. in Neur. Inform. Process. Syst.*, 2020.

[171] D. Kottke, A. Calma, D. Huseljic, G. Krempl, and B. Sick, "Challenges of reliable, realistic and comparable active learning evaluation," in *Worksh. and Tutorial on Interact. Adapt. Learn.*, 2017, pp. 2–14.

[172] R. Levin and H. Roitman, "Enhanced probabilistic classify and count methods for multi-label text quantification," in *Int. Conf. on Theory of Inform. Retr.*, 2017, pp. 229–232. DOI: 10.1145/3121050.3121083.

[173] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. on Mach. Learn.*, 2015, pp. 1180–1189.

[174] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," in *Worksh. Proc. of the Int. Conf. on Learn. Representations*, 2017.