AStA Advances in Statistical Analysis (2024) 108:395–425
https://doi.org/10.1007/s10182-024-00502-5

ORIGINAL PAPER

Deducing neighborhoods of classes from a fitted model

Alexander Gerharz[1] · **Andreas Groll**[1] · **Gunther Schauberger**[2]

Received: 22 November 2022 / Accepted: 16 February 2024 / Published online: 8 May 2024

Abstract
In this article, a new kind of interpretable machine learning method is presented, which can help to understand the partition of the feature space into predicted classes in a classification model using quantile shifts, and this way make the underlying statistical or machine learning model more trustworthy. Basically, real data points (or specific points of interest) are used and the changes of the prediction after slightly raising or decreasing specific features are observed. By comparing the predictions before and after the shifts, under certain conditions the observed changes in the predictions can be interpreted as neighborhoods of the classes with regard to the shifted features. Chord diagrams are used to visualize the observed changes. For illustration, this quantile shift method (QSM) is applied to an artificial example with medical labels and a real data example.

Keywords Interpretable machine learning · Explainable artificial intelligence · Classification task · Feature space partition · Chord diagrams

1 Introduction

With the increasing demand for very complex models in the areas of data analysis and predictive modeling, the number of black box models is growing steadily. The problem with these models is that by raising the predictive power of a model or an algorithm by adding more complexity or flexibility to it, the loss of interpretability can be tremendous. While mostly it is fairly easy to understand the fitting algorithm, understanding the fitted prediction model is pretty hard. In a random forest with 500 trees, for example, it is easy to understand a single classification tree, but to completely understand the whole ensemble model it is necessary to look at every

✉ Alexander Gerharz
gerharz@statistik.tu-dortmund.de

[1] Chair of Statistical Methods for Big Data, Faculty of Statistics, TU Dortmund University, Dortmund, Germany

[2] Technical University of Munich, TUM School of Medicine and Health, Chair of Epidemiology, Munich, Germany

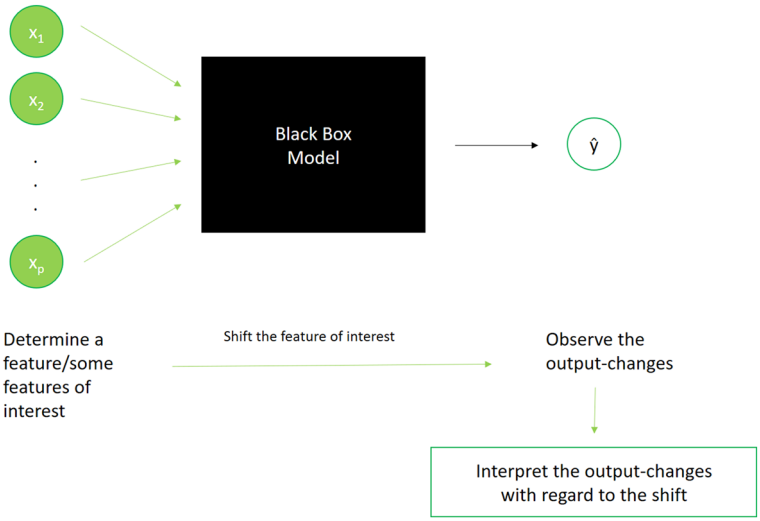


Fig. 1 Basic concept behind most interpretable machine learning methods/explainable artificial intelligence

split in every tree, which gets too expensive if the corresponding classification task was very huge and complex (Breiman 2001).

The world of interpretable machine learning (IML) methods tries to open a door to understand the internals of these complex models without having to understand every single internal detail of them. Altogether, this way IML methods try to increase the trustworthiness of such complex models. A famous IML method is the computation of the permutation feature importance as described by Breiman (2001). Here, the input is randomly permuted feature by feature and it is measured how much worse a model performs after permutation in order to determine the features' importance in the model. In contrast, the partial dependence plot (Friedman 2001), for example, does not calculate the importance of a feature in a model, but it is a well-known method to estimate the mean effect of a specific feature on the target value by shifting the inputs of some data and to observe how the output changes. A typical structure for this kind of IML methods is displayed in Fig. 1.

Another interpretable machine learning method that is based on this structure is the individual conditional expectations (ICE) plot, which, similar to the partial dependence plot, describes the effect of a specific feature on the target value, but instead of displaying a mean effect it presents the individual changes for every observation (Goldstein et al. 2015). Another completely different IML method is the usage of anchors (Ribeiro et al. 2018). Anchors are used to find specific features and their respective feature values that determine the prediction of an observation, while the other features could be randomly altered without affecting the prediction too much.

An overview of the still very limited range of IML methods can be found in the publicly available book of Molnar (2019), which lists even more IML methods and explains their usages on every day examples. Most of those methods are applicable

on both regression and classification tasks (or even more), while the method proposed in this article is specifically designed for classification tasks only.

The quantile shift method (QSM) presented in this work is based on the basic concept of IML (see Fig. 1) and is used to determine which classes are modeled as neighbors by a fitted model with regard to specific features of interest. QSM is used to determine, which small changes in the features lead to a substantial change in the predictions in the sense that the predicted class labels change. These changes can then be interpreted as neighborhoods for the different classes of an observation before and after the shift. In contrast, the anchors method is used to find the features and their respective values, which determine a specific prediction and interpret them as substantial for this specific prediction. While both methods observe whether slight changes in the features change a prediction, the interpretation is substantially different.

In an application setting, QSM can be used whenever the influence of a feature on the prediction of a target class within a fixed classification model is of interest. This could be useful, e.g., in medicine, when a physician uses a classification model for pain levels and wants to investigate what influence an increase of a feature, e.g., the heart rate, can have on the pain level of the patient. For some classic statistical models, this could be derived from the regression coefficients. However, if the model has many target classes and many features, the finally predicted label is hard to derive from the coefficients. Also, if the underlying model is a black box model with no easy way for interpretation, it is a challenge to investigate the influence of a specific feature. Here, QSM can help as it will indicate how the prediction will change if a specific feature (or even multiple features) of interest are modified.

The remainder of this article is structured as follows: In Sect. 2, we introduce the mathematical details of the method and derive the corresponding migration matrix containing these changes, which will later be presented as a chord diagram. Additionally, we illustrate the method's relevance with an artificially created example with labels from the field of medicine and also provide an in-depth discussion and explanation of how to generally interpret the method's results. In Sect. 4, a real data example is used to illustrate how the method works, and different ways how to use QSM are shown. Finally, Sect. 5 concludes and discusses advantages and disadvantages of the proposed method.

2 Methodology

In this section, we first set the mathematical background for QSM and explain how to interpret it. As there are certain conditions, which have to be kept in mind to assure a meaningful interpretation of the results, we will then explain some possible pitfalls and how they can be solved.

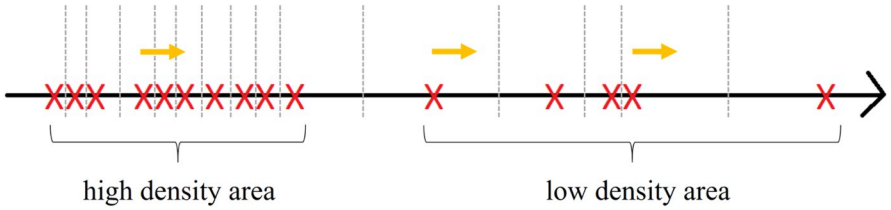


Fig. 2 Example for a feature with high- and low-density areas and possible decision boundaries; yellow arrows display a common possible shift size for all feature

2.1 Motivation

The general idea of the proposed method is to take the predictions of a model for a given data set, then increase or decrease the value of one or multiple feature(s) for each observation and interpret the changes in the predicted classes.

With the proposed method, it can be investigated, which classes of a categorical target are predicted next to each other based on the partition of the feature space by a specific classification model. This way, it can be learned which target classes differ with regard to the feature(s) of interest, but have very similar (or equal) values in their other features. It can also be investigated, which of the target classes have larger or lower values of the feature(s) of interest. Furthermore, this method can also be used to investigate whether there is an ordinal structure in the categorical target variable.

For this method, an intuitive way to conduct the shifts for a specific feature is to choose a fixed absolute amount and just add or subtract it to/from the feature for all observations. However, there are two major issues with this process, which we will describe in the following.

If the distribution of a feature is very complex and contains some low-density areas and some high-density areas, then a model can usually learn more decision boundaries in a high-density area than in a low-density area (Fig. 2). Here, it is already clear, why changing the feature values by a fixed amount is not always a good idea, because a small shift size might be enough to change the prediction in a high-density area, but not in a low-density area. If the shift size is chosen to be large, then it might be appropriate for the low-density area, but the shifts might be too big for the high-density area and some important neighbors might be skipped. The yellow arrows in this figure display a possible shift size. In the high-density area, the data points could already be shifted far enough to possibly skip two decision boundaries, while in the low-density area some data points do not even skip a single decision boundary.

Another issue is that by modifying the feature values the marginal distribution gets changed a lot, leading to values that have not existed before and might even be impossible to exist. This can easily lead to extrapolation, where the model is used to predict the class of observations with unlikely or even impossible feature values (see Hooker 2004), which will result in interpretations which are not meaningful.

The proposed method tackles also these issues by using the underlying data to dictate the size of the shift for each area of the range of a feature. This can be achieved by conducting the shifts based on the empirical cumulative distribution function. This way, smaller shifts will be conducted in high-density areas and larger shifts in low-density areas.

Principally, QSM can help to answer the following questions:

- *What happens to the predicted label, when a specific feature increases or decreases?* This question can be relevant in the field of Medicine. QSM could help a physician who is concerned what effect a specific pharmaceutical, which has an impact on a feature of the classification model, could have, e.g., on the pain level of a specific group of patients. Then, QSM can indicate the effect. Although the method is designed to work globally, a trained classification model can be used with focus on a very specific group of observations to explain the model locally. Researchers of other fields, e.g., Sports or Politics, could also be interested in this kind of question as often a new training method (Sports) or a new law or political guideline (Politics) can have an impact on a feature within a classification model. If one can estimate the impact on the features, then with QSM the impact on the prediction of the target classes can be investigated.
- *How are the target classes related?* This question can be relevant in, e.g., Social Sciences. If one models the impact of social-economic features on, e.g., the happiness in life (target variable - measured in categories), then QSM can help to investigate what kind of impact an increase in, e.g., the income or the living expenses has. Relying on quantiles to investigate the impact, as proposed in this method, might improve the interpretation.
- *Is there an ordinal structure within the target classes?* Here, there are two types of cases in which this can be relevant:
  - In the first case, if one assumes that there might be some kind of ordinal structure within the target classes based on a specific feature of interest, one could investigate if increasing or decreasing that feature confirms the direction of the assumed order. E.g., if there is a model for the relation of the physical fitness of a football player and the league the player is playing in, one could assume that faster players are playing in higher leagues. Of course, there might be some more complex structures that for some players being faster might not necessarily help to play in a higher league (goalkeepers?).
  - The second case is when the target actually has an ordinal structure, but the method used can only do classification without taking the ordinal structure into account. In Economics, the credit rating of a country or a company is usually indicated with ordinal categories. If the classification method used can only treat this target variable as categorical and not ordinal, then QSM can help to investigate this structure. If QSM indicates that slightly tweaking one of the features makes the predicted credit rating actually jump by multiple levels, then the fitted model might actually need further investigation. For this case, by no means does QSM indicate how the model needs to be modified, but it might reveal some pitfalls of the model at hand.

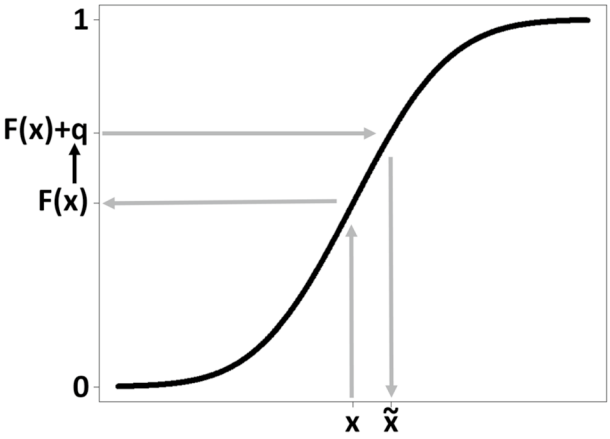


Fig. 3 Example of a feature shift for a single feature (continuous case)

2.2 Mathematical background

In the following, we will set the mathematical background for QSM. The aim is to slightly increase or decrease the value of the features of interest and observe the changes in the predicted classes.

Suppose $\hat{f}(\mathbf{x})$ is a final model fitted for a classification task on a sample of size n with K different classes, $K \geq 2$, and L be the set of all the features used for this classification with a specific set size $p = |L|$. Then, $\hat{\pi}_{\hat{f},k}(\mathbf{x})$ denotes the estimated probability by the model $\hat{f}(\cdot)$ for an observation $\mathbf{x}$ to belong to a specific class $k \in \{1, \ldots, K\}$. Next, we determine

$$k^*_{\hat{f}}(\mathbf{x}) = \arg\max_k \hat{\pi}_{\hat{f},k}(\mathbf{x})$$

such that $k^*_{\hat{f}}(\mathbf{x})$ is the class with the highest probability as estimated by the model $\hat{f}(\cdot)$ for the observation $\mathbf{x}$ (from here on we will always talk about the same fitted model, which is why we drop index $\hat{f}$ in the following for better readability).

Next, we choose a subset $M \subseteq L$ containing the feature(s) of interest. Mostly, the subset M has a size of $|M| = 1$, i.e., we focus on a single specific feature. Let now $\tilde{\mathbf{x}}_i$ represent the feature-vector for observation i, $i = 1, ..., n$, where those features from M each were shifted componentwisely by a small amount.

The shift is done by slightly increasing or decreasing the result of the empirical cumulative distribution function (ecdf) of the subset M containing the features of interest before applying the quantile-function (see Fig. 3). For this purpose, a small value q_l, the quantile shift size, is added componentwisely to $\hat{F}_l(\cdot)$ denoting the ecdf for all features $l = 1, ..., p$, with

$$q_l = \begin{cases} u_l, \; for \, L_l \in M, \; with \, u_l \in [-1, 1] \\ 0, \quad else. \end{cases}$$

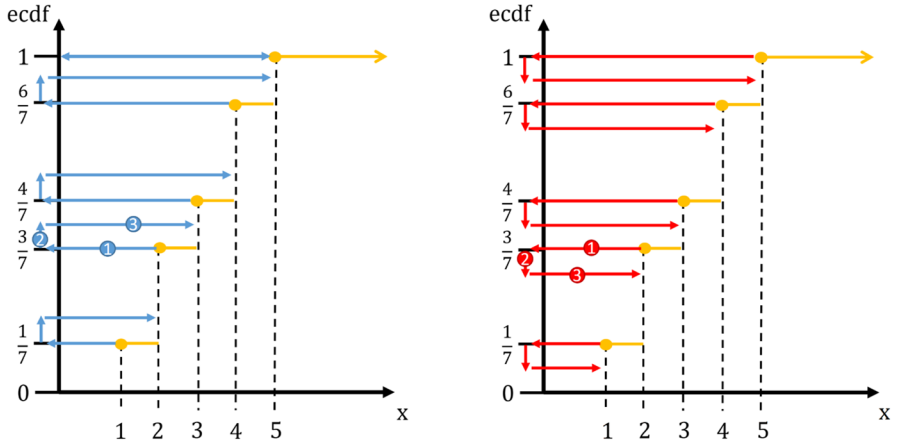


Fig. 4 Comparison of positive (left) and negative (right) shift with the same $|q_l|$

The empirical cumulative distribution function can be built either on the original data set, used for fitting the underlying model, or on a very specific data set which can be a subset of the original data or a completely new data set. Depending on the choice of the data used for interpretation, the user can choose between a global or a very targeted local explanation of the model. To prevent extrapolation in the quantile function $\hat{F}_l^{-1}(\alpha)$, α is limited to the interval $[0, 1]$. Then, for a positive shift with $q_l \in (0, 1]$, we define:

$$\begin{aligned} &\hat{F}_l(\tilde{x}_{i,l}) = min\{\hat{F}_l(x_{i,l}) + q_l, 1\} \\ \Longrightarrow\ &\tilde{x}_{i,l} = \hat{F}_l^{-1}(min\{\hat{F}_l(x_{i,l}) + q_l, 1\}). \end{aligned} \tag{1}$$

The modifying values q_l for each $l \in M$ are set by the user. As this is a crucial point for the method, in the following we provide some examples and recommendations for a reasonable choice of q.

The inverse of the ecdf $\hat{F}_l^{-1}$ does not necessarily exist, as $\hat{F}_l$ typically is not continuous. Hence, for a positive shift we have to define

$$\hat{F}_l^{-1}(\alpha) = inf\{x : \hat{F}_l(x) \geq \alpha\}. \tag{2}$$

Equation (2) determines each value of feature l after the shift as one out of the truly observed values of the respective feature, which were used to estimate the ecdf.

Due to the definition of the inverse of the ecdf as defined in Eq. (2), a positive shift is generally not comparable to a negative shift, if it is done the exact same way. While even a slight positive shift results in a change of the corresponding feature's values, slight negative shifts typically change nothing at all.

In Fig. 4, both a positive (left) and negative (right) shift are shown for a feature l with values $\mathbf{x}_l = (1, 2, 2, 3, 4, 4, 5)^T$. Now, QSM is applied with $0 < |q_l| < \frac{1}{7}$. In the ecdf as defined above, for the point $x_l = 2$ for example, we have $\hat{F}_l(2) = \frac{3}{7}$ (blue arrow ❶ in the left part of Fig. 4). If now q_l is added, this results in α where

$\frac{3}{7} < \alpha < \frac{4}{7}$ (blue arrow ❷). Due to the definition of Eq. (2), the positive shift results in $\hat{F}_l^{-1}(\alpha) = 3$ (blue arrow ❸), which means that a very small positive shift results in a change of the feature's value.

If an equally small, but negative shift is used for the same value $x_l = 2$, then $\hat{F}_l(2) = \frac{3}{7}$ (red arrow ❶ in the right part of Fig. 4), which is the same as for the positive shift. Next, the amount $|q_l|$ is subtracted, which results in α where $\frac{2}{7} < \alpha < \frac{3}{7}$ (red arrow ❷). When the shifted value is calculated now, due to the definition from Eq. (2), $\hat{F}_l^{-1}(\alpha) = 2$ (red arrow ❸), which means the value has not changed at all. In fact, in the present example not a single value would change when performing the negative shifts, which shows that negative and positive shifts of the same absolute amount q are not necessarily comparable as conducted right now. Hence, a negative shift has to be defined in another way.

A negative shift can be done by using a positive shift in the same manner as before after flipping the whole distribution of the corresponding feature. In particular, in a first step a whole feature l is flipped by multiplying all of its values with (-1). Next, the ecdf is constructed for the flipped feature l and the absolute amount of q_l is added to its values. Last, the resulting quantiles give the new values of the flipped feature l which have to be transformed back to get meaningful values on the original scale. Therefore, for any q_l from $q_l \in [-1, 0)$ Eq. (1) changes to

$$\tilde{x}_{i,l} = -\hat{F}_{-l}^{-1}(min\{\hat{F}_{-l}(-x_{i,l}) + |q_l|, 1\}), \tag{3}$$

with $\hat{F}_{-l}(\cdot)$ as the empirical cumulative distribution function of the flipped feature l and $\hat{F}_{-l}^{-1}(\cdot)$ being the corresponding quantile function. With this small difference in the process of the shifts the shift sizes become comparable in both directions.

Combining all equations from above, the shifts become well defined as

$$\tilde{x}_{i,l} = \begin{cases} \hat{F}_l^{-1}(min\{\hat{F}_l(x_{i,l}) + q_l, 1\}), & for\ q_l \in (0, 1] \\ -\hat{F}_{-l}^{-1}(min\{\hat{F}_{-l}(-x_{i,l}) + |q_l|, 1\}), & for\ q_l \in [-1, 0). \end{cases}$$

Now, $\tilde{\mathbf{x}}_i$ is the new shifted observation, which has the same value for those covariates from $L \backslash M$ as $\mathbf{x}_i$, but different values for the features from M. These features were increased or decreased by the value that corresponded to the component wise raise or reduction of the respective ecdf by the amount q_l, not exceeding the minimum or maximum of the empirical distribution of the features from M.

Principally, this modus operandi does not only work for metric features, but also for (ordered or nominal) categorical features of the form $x_l \in \{1, \ldots, c\}$, where c is the number of categories. For this kind of features, a shift from one group to another has to be chosen manually, e.g., switching from group r to another group s within a specific feature l, i.e., changing from $x_l = r$ to $x_l = s$.

Finally, for observation $i, i = 1, ..., n$, $M \subseteq L$ and $\mathbf{q} = (q_1, ..., q_p)^T$ (possibly including zeros if $M \subset L$) let $C_{\mathbf{q},M}(x_i)$ define the pair of the original and the (potentially) new class prediction resulting from this shift, i.e.,

$$\mathbf{x}_i = [\mathbf{x}_{i,L\backslash M}, \mathbf{x}_{i,M}],$$
$$\tilde{\mathbf{x}}_i = [\tilde{\mathbf{x}}_{i,L\backslash M}, \tilde{\mathbf{x}}_{i,M}],$$

$$C_{\mathbf{q},M}(\mathbf{x}_i) = \left(k^*(\mathbf{x}_i), k^*(\tilde{\mathbf{x}}_i)\right)$$
$$= (\hat{y}_{i,old}, \hat{y}_{i,new}),$$

with $\hat{y}_{i,old}$ being the predicted class for observation i before, and $\hat{y}_{i,new}$ the predicted class for observation i after the shift. This yields $\hat{y}_{i,old} = \hat{y}_{i,new}$, if the predicted class *has not changed* by shifting $\mathbf{x}_{i,M}$ and $\hat{y}_{i,old} \neq \hat{y}_{i,new}$ otherwise. Note that $\tilde{\mathbf{x}}_{i,L\backslash M} = \mathbf{x}_{i,L\backslash M}$ holds, as the features from $L\backslash M$ were not modified.

2.3 The choice of q for numeric features

In general, the choice of q for a numeric feature mostly depends on the research question at hand. If the research question requires to find the direct neighbors assuming the feature of interest would be just slightly larger or lower, then choosing a very small number $|q| > 0$ is sufficient. As the ecdf is a step function, even a very small value of q will lead to a change in the value of the feature (given the same data is used for QSM as for estimating the ecdf). In this ecdf step function, assuming only unique values, every step has the size of $\frac{1}{n}$, with n being the number of observations in the data set. If $0 < |q| < \frac{1}{n}$ is chosen, then every value gets mapped to the next unique larger or smaller value in the underlying empirical distribution, limited to the maximum and minimum of the range of the feature of interest.

Considering that due to multi-dimensionality one target class can have multiple neighbors in the same direction a slightly larger q is recommended. In this case, the number of observations in the data set (n), but also the number of unique values of the feature of interest takes a big role. We elaborate on this point in the following paragraph.

If a user has a very concrete quantile-based shift in mind, e.g., shifting the feature of interest by five values to one side or the other, then, principally, it is recommended to choose $q = \pm\frac{v}{n}$, in this example $v = 5$. In general, $v \in \{1, 2, ..., n\}$ is an integer. Assuming that there are no ties in the underlying data set and every value is unique, then v can be chosen as the number of ordered values by which all the observations of the respective feature of interest are shifted (limited to maximum and minimum). If the next larger or smaller value of a concrete value $x_{i,l}$ is not unique, then the next step in the ecdf has the size of the number of observations with this value divided by n. Let us assume that the next larger or smaller value occurs **ten** times. If $x_{i,l}$ should be shifted further than just to the next unique value, then q needs to be larger than this step of the step function, so $|q| > \frac{10}{n}$. If all values would be unique, then the choice of this magnitude would shift $x_{i,l}$ by more than ten values, but with ties it can be less.

In general, the number of observations has a huge influence in choosing q, considering the specific research question at hand, and can be illustrated with a concrete example. Let us assume a classification model f, which classifies what kind of sport

a kid is doing in its spare time based on basic characteristics like the body height. Now, one might want to investigate, what sport a kid would do if it would be a little bit taller, keeping all other features constant. Also assuming that 100 kids were part of this data set we can now rank them based on their height. Then, choosing $q = \frac{1}{100} = 0.01$ means that all kids are shifted to a height that ranks 1 position higher than themselves. With this shift, it is very unlikely that an intermediate class was skipped. If instead we now include 9900 more kids into our study, retrain the model and then choosing a similar shift of $q = \frac{1}{100} = 0.01 = \frac{100}{10{,}000}$, assuming a similar distribution, then, in fact, the increase in the height should be comparable, even though each kids height gets increased by 100 ranks. However, as there are now more data points in the data set, within these 100 ranks it is more likely that intermediate classes can occur within this shift that are skipped. Now, the user has to decide, if these intermediate classes should also be found. Then, q needs to be chosen smaller than 0.01, in the most extreme case $q = \frac{1}{10{,}000}$. If the user decides that the magnitude of the shift should still be similar to the magnitude in the example with the 100 kids, because otherwise the increase in height into potential intermediate classes is not substantial enough, then $q = 0.01$ might still be a good choice.

There is one thing about this kind of choice for q that should be kept in mind. If a tiny amount of q is added to the ecdf, numerical problems in the sense of rounding errors[1] might occur when determining the shifted feature value. Facing this problem, the shift resulting from the addition might get larger than intended.

To avoid this problem, the usage of $q = \pm\frac{v}{n+1}$ is highly recommended. Due to the definition in Eq. (2), this leads to the exact same result as $q = \pm\frac{v}{n}$, as

$$\frac{v-1}{n} < \frac{v}{n+1} < \frac{v}{n},$$

for all $v, n \in \mathbb{N}$ with $v < n$.

2.4 Presentation of results

The results could now be given in form of a migration matrix for all observations $i = 1, ..., n$, where the rows indicate the predicted classes of an observation before the shift of $\mathbf{x}_{i,M}$ and the columns indicate its predicted classes after the shift. The trace of this migration matrix counts the number of observations that have not changed classes by the shift, while off-diagonal elements aggregate the number of observations that have changed from the predicted class as indicated by the respective row into the predicted class as indicated by the respective column. An example of a migration matrix can be found in Table 1.

The off-diagonal elements of Table 1 can be interpreted as follows:

[1] The rounding errors are a consequence of the way real numbers are represented in a computer: as a signums-bit, a bit-sequence for the exponent and a bit-sequence for the significand.

Table 1 General structure of migration matrices for 2 classes

	A_{after}	B_{after}
A_{before}	$n_{A\to A}$	$n_{A\to B}$
B_{before}	$n_{B\to A}$	$n_{B\to B}$

Table 2 Exemplary migration matrix for 2 classes

	A_{after}	B_{after}
A_{before}	10	0
B_{before}	1	9

- if $n_{A\to B} > 0$, there exists an area in which class B is classified close to an area in which class A is classified with regard to the shift of the features from M
- if $n_{A\to B} = 0$, no area in which class B is classified is found next to an area in which class A is classified with regard to the shift of the features from M - but it could still exist! (maybe the shift was not substantial enough and the data points have not reached the other side of the border or class B was skipped because the shift was too strong)

Of course, $n_{B\to A}$ can be interpreted analogously.

Chord diagrams are a nice way to represent these migration matrices. If we have an exemplary migration matrix as defined in Table 1, for a two class example the migration matrix might look like Table 2.

The migration matrix in Table 2 can be visualized as a chord diagram as shown in Fig. 5. In the lower half of this figure, the ten observations, which belong to class A before the shift, are shown by the big red strang of chords starting on the scale between 0 and 10 and ending up on the same scale between 11 and 21 as indicated by the arrow. This shows that ten observations belong to class A before the shift and all ten observations are still in class A after the shift. In the upper half of this figure, the observations, which belong to class B before the shift, are shown by the big turquoise strang of chords starting on the scale between 0 and 10. From this strang of chords, a big part ends up on the upper halves scale between 10 and 19, which indicates that nine observations that belong to class B before the shift are still in class B after the shift, but a small part ends up in the lower half's scale between 10 and 11, which indicates that one observation is predicted as class B before the shift, but is predicted as class A after the shift. This is exactly, what the migration matrix in Table 2 indicates.

This shows that the migration matrix, which was a result of QSM with a specific data shift, indicates that in the direction of the data shift there is an area of class A modeled next to an area of class B.

All analytic graphics and analyses in this work have been performed in R Version 4.1.1 (R Core Team 2021). The chord diagrams, which are the main visualization tool for this method, were computed with the `circlize`-package in R (Gu et al. 2014).

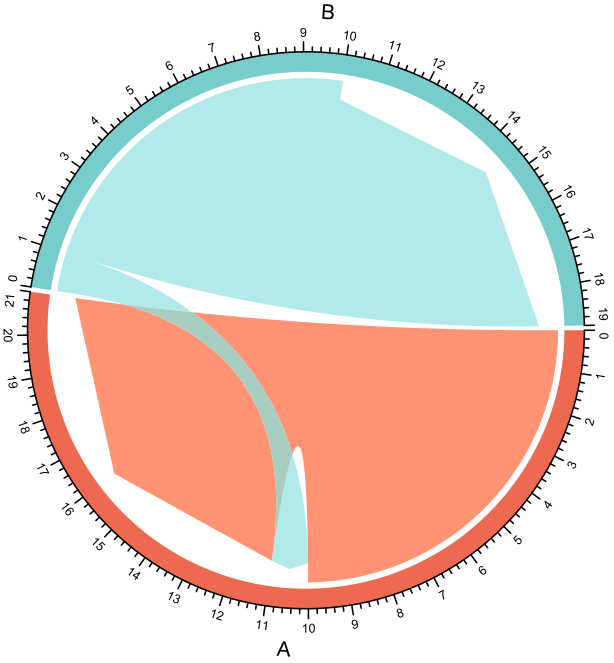


Fig. 5 Chord diagram for migration matrix in Table 2

3 Illustrative example

First, we create a simple artificial data example and assign specific labels to the classes in order to illustrate for which kind of research questions this method is applicable. In the same context, we will show why choosing quantiles as basis of the data shift should be preferred over choosing a specific number. For illustration purposes, the present example is rather simple and clearly structured, but especially in really complex data situations, in which there are too many features to look at in single graphics, using quantile-based shifts can be advantageous.

Figure 6 shows a 2-dimensional feature space in which three different pain levels are predicted by some statistical model (e.g., a decision tree) for which we assume that it is able to describe the relationship between x_1, x_2 and the target y very well. The model maps patients with features x_1 and x_2 to a 3-class target y. All patients with a low value of x_1 are assigned to the class *medium pain*. However, if x_1 exceeds a certain threshold, patients with a very low x_2 are assigned to class *no pain*, patients with a medium value of x_2 are assigned to class *medium pain* and patients with a large value of x_2 are assigned to class *high pain*.

Suppose that this model is provided to a physician, who plans to increase x_1 to ease the pain of a patient (e.g., if x_1 is the heart rate, the physician might advise the patient to become physically more active). If at the same time also x_2 is small, the model agrees with the physician's assumption and the patient in fact might get better. However, in contrast to linear relationships as modeled, e.g., by standard

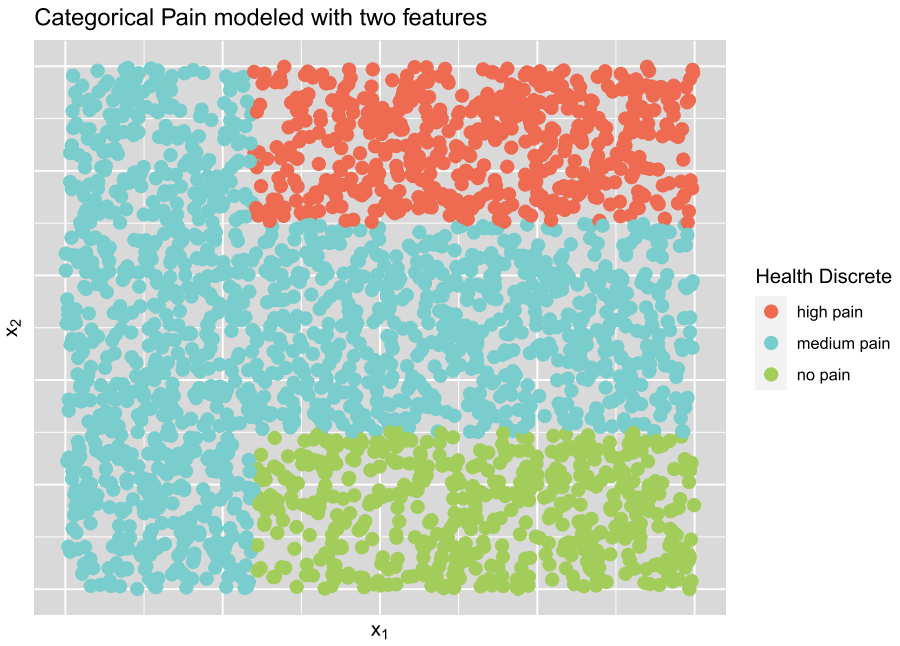


Fig. 6 Predicted pain levels by two metrical features

Table 3 Migration matrix for prediction changes when raising x_1 by $q_1 = \frac{250}{2501}$ (and leaving x_2 unchanged)

	High pain	Medium pain	No pain
High pain	534	0	0
Medium pain	73	1286	72
No pain	0	0	535

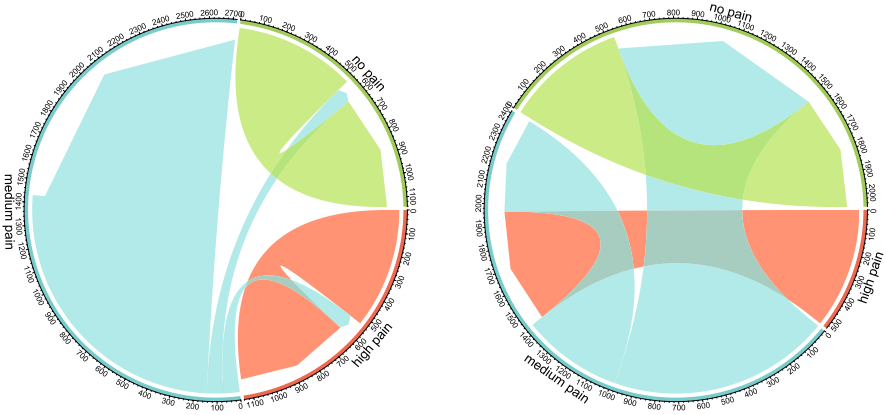
linear regression, increasing x_1 would not always lead to improvement in this more complex model. For a patient with a rather large value of x_2, an increase of x_1 might even result in a migration from the *medium pain* level to *high pain*. In this case, a detailed analysis of the migration matrix for increasing x_1 would reveal the more complex, nonlinear relationships (see Table 3).

This migration matrix is presented as a chord diagram in Fig. 7a, which is built up from all the observations in the migration matrix. Each observation is presented as a single chord, starting from the class in which it was predicted before and ending up in the class it was predicted after the feature shift. Multiple observations that have the same starting and ending point in the chord diagram are combined together as a big strang of chords, which is the wider the more observations have the same starting and ending points.

In the present example, it is easy to see that after raising x_1 by just a small amount some *medium pain* patients get better and move to the *no pain* category, while some get worse and end up in the *high pain* category.

Table 4 Migration matrix for prediction changes when raising x_1 and at the same time decreasing x_2 by $q_1 = \frac{750}{2501}$ and $q_2 = -\frac{750}{2501}$, respectively

	High pain	Medium pain	No pain
High pain	0	534	0
Medium pain	0	445	986
No pain	0	0	535



(a) Changes in prediction when raising x_1 with $q_1 = \frac{250}{2501}$ and x_2 by $q_2 = 0$

(b) Changes in prediction when raising x_1 and decreasing x_2 with $q_1 = \frac{750}{2501}$ and $q_2 = -\frac{750}{2501}$

Fig. 7 Artificially created medical data predicted with different data shifts

Furthermore, in Fig. 6 it is displayed that by decreasing x_2 only, a patient with a *high pain* or *medium pain* level could actually get better and by decreasing x_1 only, *high level* patients could get better. Furthermore, by decreasing x_2 and simultaneously increasing x_1 more substantially (e.g., in case the physician has a very effective drug or any other method for the considerable manipulation of these features) every patient classified in the *high pain* level class by the model is then assigned to a *medium pain* level and more than half of the *medium pain* level predictions get now predicted in the *no pain* level category (see Table 4 and Fig. 7b).

In this case, if one would increase x_1 and decrease x_2 by a small amount, one can easily imagine that in the upper third of Fig. 6 some of the medium pain level patients could get worse and become high pain level patients. For the chord diagram in Fig. 7b, the corresponding value of q was chosen so large that the graph would not show this type of transition. Actually, the used shifts were too large such that the class of high pain level patients is skipped. As mentioned in the remarks below, this method does not proof that there is no neighborhood between the two classes for the regarded shift. The chord diagram simply displays to which classes the patients would migrate, when the shift of the features was that large. Here, we should not consider the relationship that was found as some kind of "direct neighborhood", but more as a "general closeness".

Note that in Fig. 6, features x_1 and x_2 were intentionally presented without a scale. The intention is to illustrate by this artificial data example why using quantile-based feature shifts can be advantageous compared to using plain (rather arbitrary) numbers. The main reason is that it is often hard to determine what size a "slight" increase or decrease might be in the typical case when the exact distribution of the features is unknown. This is particularly relevant when the task is to find direct neighborhoods. Moreover, if many predictors are present, a fast and automatic method for the computation of the corresponding chord diagrams is essential, instead of determining for every feature and observation manually, what a slight shift might be. Different features usually have different scales and, depending on their location in the feature space, a "slight" shift could have a different meaning for different observations, especially if a feature of interest has a very complex distribution (e.g., a multimodal distribution). To avoid the problem of different scales, it could be a good idea to determine the shift sizes based on the feature's standard deviation, but especially for features with high-density and low-density areas, this still does not solve the problem (see Fig. 2). All of these cases can be handled by using small amounts on the quantile scale, which are comparable for all metric features.

3.1 Remarks about the method's interpretation

In the following, we give some important remarks regarding the interpretation of the results.

1. If we define the preference order
   $A \succ B :=$ class A is directly (or generally) next to class B in the direction of the shift,

   then due to the fact that particularly complex models can produce also complex partitions of the feature space, it follows

   $$A \succ B \wedge B \succ C \nRightarrow A \succ C.$$

   This expresses that the results of QSM can not be interpreted transitively. In particular, a rather complex model could classify a specific class spotwise in the feature space, in which case one could get results that seem transitive, but in fact are not (for more details, see Sect. 3.2).
2. If the classification method used for modeling is only designed to estimate (linear) monotonous effects, then migrations between two classes can only be found in one direction, but not the other. If, instead, the underlying method can also estimate more complicated effects, then migrations from a certain class A into another class B, but also in the other direction, from B into A, can be found at the same time. This indicates that the values of the original feature of interest, but also the values of the covariates, can be a deciding factor on what is happening if the feature of interest is increased. In these cases, a local application of QSM can help to improve the understanding on what might happen if the feature of interest is increased.

3. As indicated, QSM is built to find neighborhoods as described by the model, but not to proof that there is no neighborhood between two classes regarding the shift of $\mathbf{x}_M$. If the goal is to *proof* that a certain class has no direct neighborhoods within the fitted model, then one would have to fill the complete modeled space of the class of interest with data points and then had to shift $\mathbf{x}_M$ with infinitely small steps from the starting points until the limits of the feature range. If one also considers the model for extrapolation, then these infinitely small steps would need to be done either until $+\infty$ or $-\infty$, respectively, in direction of the shift, or until all points have switched classes.
4. The shift of $\mathbf{x}_M$ could be too big, such that an intermediate class was skipped, and consequently, no direct neighborhood was found. Hence, the "neighborhoods" from above should be regarded more generally as an "exists above" (if the shift was done by raising $\mathbf{x}_M$) or as an "exists below" (if the shift was done by decreasing $\mathbf{x}_M$). To find *direct* neighborhoods one would have to start with a very small shift of $\mathbf{x}_M$ and increase (or decrease) it continuously. In contrast to this, if one has a specific shift in mind, one could just use this specific shift and then the resulting migration matrix shows the corresponding class changes (if any).
5. If specific features are shifted and a neighborhood is found between two classes with respect to the magnitude of the shift, this neighborhood can be interpreted as a neighborhood between the original class and the class after the shift, if the task at hand is to find out, how the *model describes* the neighborhoods (of course, keeping in mind that an intermediate class could have been skipped). But if the task at hand is to find realistic and practical neighborhoods between modeled classes, then these neighborhoods should always be investigated in two ways. If a neighborhood is found by the intended shift between two classes, this means that there exist data points on one side near the border between these two classes. This does not necessarily mean that on the other side of this border data points can also exist. If similar shifts are carried out in the opposite direction and this neighborhood is not confirmed, then this might mean that due to the shift impossible feature combinations have been created and, thus, the found neighborhood has no practical use. Consider, e.g., a classification model for the position of football players containing *shots on target* and *scored goals* as features. If the player had a single shot on target, which resulted in a goal, then increasing the amount of goals leads to impossible data points, but a classification border, and thus a neighbor, could still be found if the model does extrapolation in this area of the feature space. A reason for this might be that the model simply extrapolates into this area of the feature space (general problem of extrapolation, which can lead to unreasonable interpretations; Hooker 2004).
6. In many cases, multiple data points could occur with equal values of a possible feature of interest. If those points are directly at a border between two classes, different problems can be observed, as shifting all data points changes the underlying distribution not just at the edges. For most applications, this is typically not a problem, but if the actual number of observations changing class is investigated, then having ties might lead to misleading results. In cases with ties, a bulk of data points tied in the feature of interest could find a neighborhood, while if the data

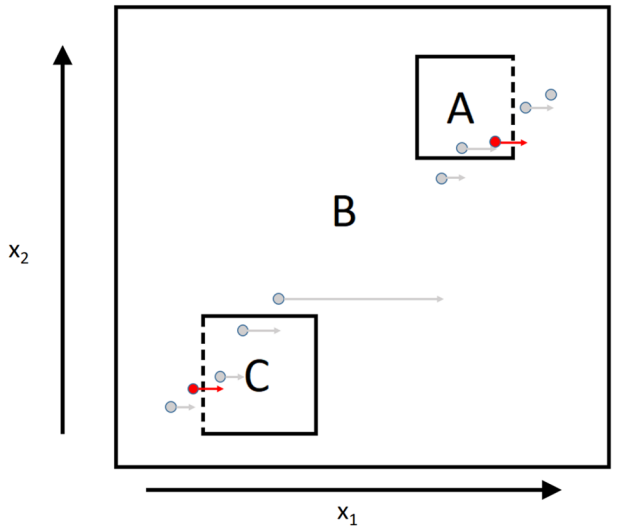


Fig. 8 Example for transitivity problem in a 2-dimensional feature space

points would not be tied, then just a small amount of data points would find this neighborhood. A detailed discussion can be found in Sect. 3.3.

7. Finally, a rather straightforward and fundamental remark: If the model at hand is rather bad and inappropriate, the found neighborhoods between the classes are correct for describing and understanding the model, but would not reflect the reality. Hence, it is important to properly evaluate the model first, before it is interpreted.

3.2 Transitive interpretations

To show possible problems regarding a transitive interpretation of this method, here is a small artificial data example. In Fig. 8, a feature space with two features x_1 and x_2 is shown. A model now labels most of this feature space as class **B**, while a small area with a low value of both features is labeled as class **C** and a small area with high values of both features is labeled as class **A**. In addition to that there are 10 red and gray data points, which are used to describe the partition of the feature space with QSM.

Now, the feature space partition should be determined by choosing $q_{x_1} = \frac{1}{11}$ and $q_{x_2} = 0$ (keeping x_2 constant). Hence, only neighborhoods with regard to x_1 are looked for. With 10 different data points, this means that holding x_2 constant each data point gets assigned the next larger value of x_1 contained in the data set. The data point with the highest x_1 does not change, as it is already at the upper bound of the range of x_1. This shift results in the migration matrix shown in Table 5.

In Fig. 8, the two points, which change their prediction, are marked in red and switch the modeled class through the dashed lines. These are the two points shown

Table 5 Migration matrix for QSM with the shift of $q = (\frac{1}{11}, 0)$ from Fig. 8

	A_{after}	B_{after}	C_{after}
A_{before}	1	1	0
B_{before}	0	5	1
C_{before}	0	0	2

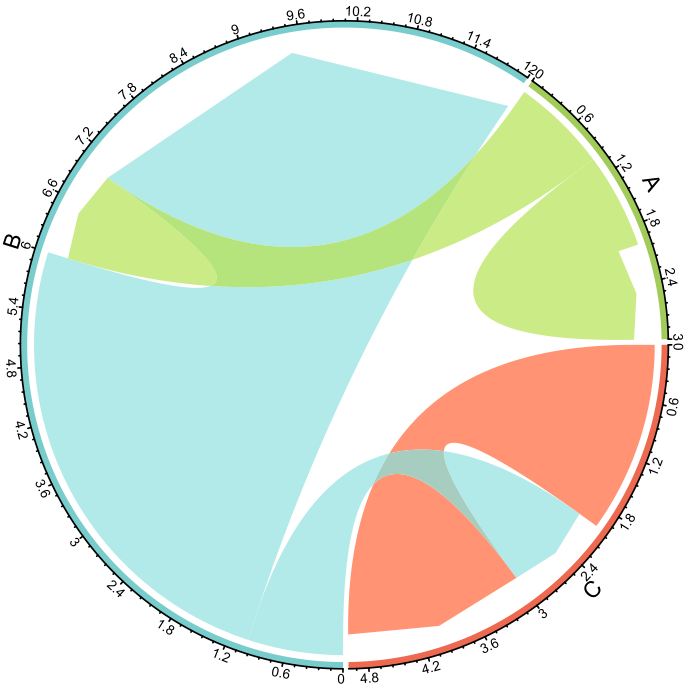


Fig. 9 Chord diagram for migration matrix in Table 5

in the migration matrix in Table 5 as one has switched from class **A** to class **B** and the other one from class **B** to class **C**. As there is an area of class **B** modeled in the direction of the shift above an area of class **A**, and then there is an area of class **C** modeled in the direction of the shift above an area of class **B**, based on the corresponding migration matrix one could conclude that in the direction of the feature shift there is some kind of "hierarchy". Particularly, here one might conclude that along the direction of this shift class **A** is below class **B**, which itself is below class **C**, but as shown in Fig. 8 this is not the case. Even if the dimensionality would be too large to graphically visualize it, just by checking the respective feature of interest for the groups separately would likely confirm the non-transitivity for this example.

In particular, the respective chord diagram as shown in Fig. 9 can (wrongly) suggest this already mentioned kind of "hierarchy". The chord diagram shows the

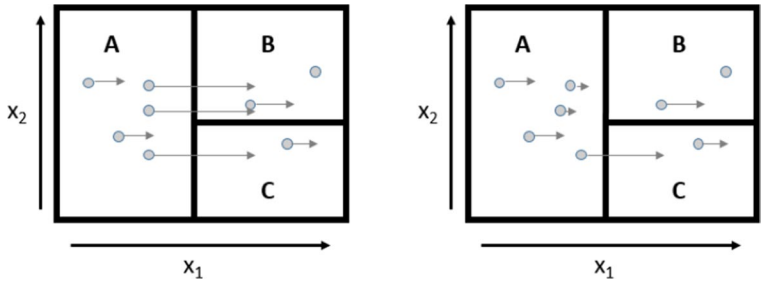


Fig. 10 QSM with $q_{x_1} = \frac{1}{n+1}$ with some equal (left) and slightly jittered (right) data points

migration of one observation from class **A** to class **B** and the migration of one observation from class **B** to class **C**, which looks like a hierarchical structure that actually does not exist. To conclude, QSM results should not be interpreted with regard to transitivity.

In this specific case, QSM just describes that there is in fact an area of class **B** modeled in the direction of the shift above an area of class **A** and there is an area of class **C** modeled in the direction of the shift above an area of class **B**. Thus, with this specific shift these two neighborhoods were found. When using other shifts in other directions, then other neighborhoods could be found.

3.3 Multiple data points with the same value

When multiple data points with the same value occur in a data set and QSM is used, some unexpected problems can occur. In the left graph of Fig. 10, there are 3 data points with an equal value for x_1. When choosing the shift size $q_{x_1} = \frac{1}{n+1}$, so the smallest possible number that allows the data points to shift, then all 3 data points would change their predictions. In this case a neighborhood between class **A** and class **B** and another neighborhood between class **A** and class **C** would be found. As shown in the right graph of Fig. 10, if the same three data points would not be exactly equal but just slightly different, only one of these data points would change its prediction and the neighborhood between class **A** and class **B** would not even be found in this case, which is another problem.

If the shift size $q_{x_1} = \frac{2}{n+1}$ is chosen, such that every data point is shifted by 2 unique values of x_1, then both neighborhoods would be found even in the situation of the right graph in Fig. 10, when the data points are not exactly equal, as one of the data points changes its prediction from class **A** to class **B**.

For any continuous (and random) feature, the probability for a specific value is zero. Hence, multiple data points with the same covariate value theoretically should never occur. But in real world applications, for example due to rounding, equal covariate values are possible and could substantially change the interpretation of QSM (see Fig. 10).

If some observations have exactly the same value for a feature of interest (e.g., due to rounding), then the marginal distribution of the corresponding feature gets changed substantially, which is not desired. To avoid this problem, there are some ways to adjust QSM to treat observations with equal values more fairly compared to those, which have unique values for a specific feature.

1. Shift all ties: One possibility is to shift all of the data points that share a specific value of a feature of interest and observe the changes in the predictions (left graph in Fig. 10). In this example, this guarantees that all data points are shifted, and neighborhoods can be found more easily. As mentioned above, this method might lead to disproportionately large shifts compared to observations with unique values. Also, the marginal distribution of the feature of interest gets changed a lot by this very fast, which is a high indicator of creating improbable covariate combinations.
2. Repeatedly shift ties randomly: Another alternative in the case of multiple observations with equal values for a specific feature of interest, is to repeatedly shift the observations after determining an artificial random order. As shown in the right graph of Fig. 10, when the observations were just slightly jittered, in this example two observations were changed by almost no amount and one by a larger amount. As a consequence, these observations are treated more fair compared to the other observations than in the *shift all ties* method, but very unequally among themselves. For the artificial example, the prediction of just one of these three observations changes. The *repeatedly shift ties randomly* approach does exactly that, but instead of jittering and thus adding some blurredness to the data, it repeatedly executes QSM and randomly determines an order of the tied observations. While all observations with unique values of the feature of interest are shifted in the same way for all the repeated shifts, the observations sharing their value of the feature of interest with other observations are shifted in just a fraction of the repetitions by the full shift as in the *shift all ties* approach. Hence, observations with equal values of the specific feature of interest are in most repetitions shifted less, and if $|q|$ is rather small, they might not be shifted at all. Thus, in comparison to the *shift all ties* method, this method treats the observations with the same values of the specific feature of interest more fair compared to the observations, which have unique values of this feature of interest. Additionally, the parameter q is more sensitive as a smaller change of q has more impact on the output of the method. Thus, it can be used more flexible by the user.

The differences of these methods are illustrated in Sect. 4 in a real data example.

4 Real data application

In this section, QSM is applied to a real data set in order to provide some insight on the method's potential. For this purpose, a data set for classifying the location of proteins in yeast is used. First, the data set is used to show how QSM should be applied and how to interpret the results. Next, the two proposed approaches to handle ties from Sect. 3.3 are compared.

4.1 Application and interpretation of QSM

The chosen data set is the `yeast` data set (Horton and Nakai 1996), which was taken from the OpenML database (Vanschoren et al. 2013). The data set contains proteins from yeast and the target of the classification task is to localize the protein into one of ten possible locations within the yeast. The ten locations are further described by Horton and Nakai (1996) and contain cytoplasmic, including cytoskeletal (CYT); nuclear (NUC); vacuolar (VAC); mitochondrial (MIT); peroxisomal (POX); extracellular, including those localized to the cell wall (EXC); proteins localized to the lumen of the endoplasmatic reticulum (ERL); membrane proteins with a cleaved signal (ME1); membrane proteins with an uncleaved signal (ME2); and membrane proteins with no N-terminal signal (ME3), where ME1, ME2 and ME3 proteins may be localized to the plasma membrane, the endoplasmatic reticulum membrane or the membrane of a Golgi body.

The data set also contains eight different features, which are used to classify the target variable. These features are also further described in Horton and Nakai (1996) and in the UCI Repository (Dua and Graff 2017) and represent:

- *mit*: score of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins.
- *erl*: presence of "HDEL" substring (thought to act as a signal for retention in the endoplasmic reticulum lumen); binary attribute.
- *pox*: peroxisomal targeting signal in the C-terminus.
- *vac*: score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins.
- *nuc*: score of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins.
- *mcg*: McGeoch's method for signal sequence recognition.
- *gvh*: von Heijne's method for signal sequence recognition.
- *alm*: score of the ALOM membrane spanning region prediction program.

For this classification task, a multinomial regression model using the logit link function is now computed with the `nnet` package (Venables and Ripley 2002). When using a multinomial regression model we obtain coefficients as well as an intercept for every class of the target except of the chosen reference class for every feature. Overall, this results in 81 different coefficients for this exemplary data set.

As interpreting a multinomial logistic regression model can be very complex, there exist techniques to help with the interpretation. One method is the visualization with Effect Stars (Tutz and Schauberger 2012). Effect Stars condense the potentially huge amount of coefficients into one very clearly arranged graphic. The aim is to give a nice overview of the coefficients for all the target classes into one single star plot per feature. This overview still has its limitations as the coefficients of every feature are visualized without considering the complex relations between all the features including the intercept. Another way to assist in the interpretation of a multinomial logistic regression model is the use of Effect graphics (Fox and Weisberg, 2019; Fox and Hong, 2009). Effect graphics display how the predicted probabilities for all the target classes change for a single specific feature or a small amount of features at the same time. In this process, all other features are fixed to specific values, while the features of interest are varied over their range of values. This does not consider the complex structure between the features as this process can produce feature combinations that are very unlikely to exist or are even impossible. Also, as the additional features are fixed to specific values, the graphic only visualizes the behavior of the model in a very narrow range of the feature space ignoring everything around it. Consider a 2-dimensional example as, e.g., in Sect. 3. If the effect of x_1 should be investigated, then x_2 is fixed to its mean (default), and the effect of x_1 is investigated on a thin horizontal line in the middle of Fig. 6. All the other areas are not considered for investigating the effect. Imbalanced target classes may also lead to problems in the interpretation of both methods. Effect Stars are visualized without linkage between all the features, but especially the different intercepts for the imbalanced target classes can have a huge impact on whether a class has a high likelihood to be predicted. The Effect visualizations may also suggest that a specific small target class would not be predicted for any value of the feature(s) of interest, but the cause for this could be the fixed values of the remaining features.

QSM has the advantage that the interdependence between the features is considered as it uses real data points. Also, the imbalance of classes does not affect QSM, as real data points with feature combinations of the smaller classes are used instead of one fixed combination like in the Effect graphics. For the multinomial logistic regression model, QSM can help to visualize which classes will realistically be predicted if a specific feature or a small amount of features of the empirical data would have slightly larger or lower values based on the underlying model.

In the following, QSM is applied to this multinomial model to learn which target classes are modeled closely and similar to each other with slight differences concerning a specific feature. For exemplary purpose, we choose to use just the single feature *mit*, but also other single features or even multiple features at the same time could be used. Here, it is investigated how the predicted target class would change if the computed *mit* score would be slightly larger. This helps to grasp the relations between the predicted target classes regarding this *mit* score.

An important reason why this data set is used here is that it contains a lot of observations with equal feature values. Even though the data set contains $n = 1484$ observations, x_{mit} contains just 78 different values. For the fitted model,

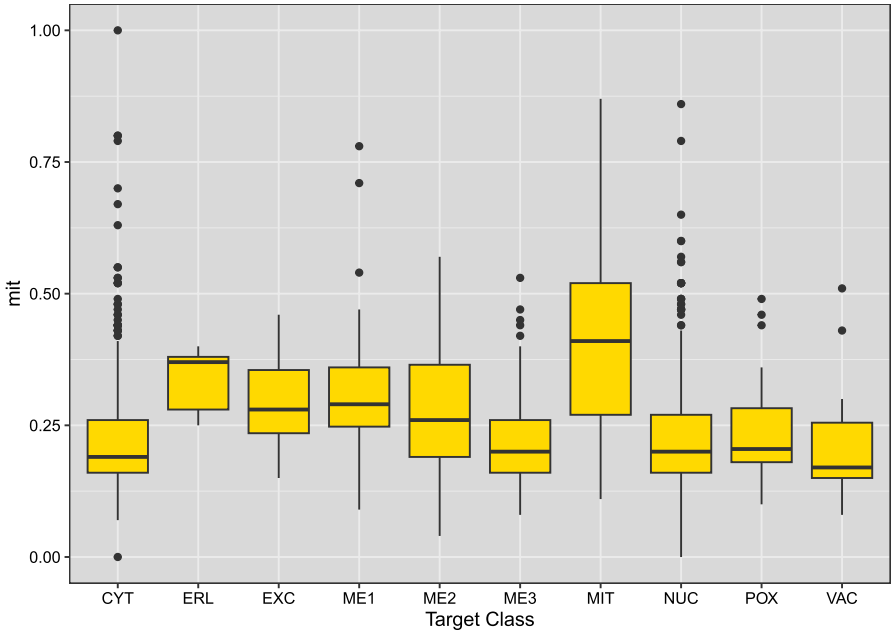


Fig. 11 Marginal distributions of x_{mit} for each of the target classes

the quantile shift size of $q_{mit} = \frac{75}{1484+1} \approx 0.05$ is chosen and $(q_{mcg}, q_{gvh}, q_{alm}, q_{erl}, q_{pox}, q_{vac}, q_{nuc})^\top = \mathbf{0}$. This means that all feature values of x_{mit} are increased by a slight amount. By this choice we simulate a data set, where the score of the respective discriminant analysis underlying the *mit* value would have been slightly larger.

The marginal distribution of the feature x_{mit} is shown in Fig. 11. This indicates that larger values might lead to a classification of target class *MIT*. Recall that feature x_{mit} is representing a score of a discriminant analysis of the amino acid content of the N-terminal region (20 residues long) to divide mitochondrial and non-mitochondrial proteins. The coefficients of the multinomial model can be used to check if according to the model a larger value of x_{mit} does indeed result in a larger probability of being predicted as target class *MIT*. QSM can now add the information, which other target classes might be similar to the target class *MIT* regarding the unchanged other features, but with a lower value of x_{mit} according to the model. Of course, depending on the structure of the data set and the complexity of the model, even the target class *MIT* might have another target class close to it in which observations might migrate if x_{mit} is increased. Also, it is globally investigated if other target classes are modeled similar to each other regarding the other features, but with slightly different values in x_{mit}. Two target classes that are found to be similar in the described manner are what we refer to as a neighbors.

Table 6 Migration matrix for prediction changes when raising x_{mit} by $q_{mit} = \frac{75}{1485}$ with the *shift all ties* approach; the observed class changes are in bold font

	CYT	ERL	EXC	ME1	ME2	ME3	MIT	NUC	POX	VAC
CYT	608	0	0	0	0	0	**21**	**1**	0	0
ERL	0	5	0	0	0	0	0	0	0	0
EXC	0	0	27	0	0	0	**4**	0	0	0
ME1	0	0	0	42	0	0	0	0	0	0
ME2	0	0	0	0	41	0	**6**	0	0	0
ME3	0	0	0	0	**2**	170	**2**	0	0	0
MIT	0	0	0	0	0	0	227	0	0	0
NUC	0	0	0	0	0	0	**21**	293	0	0
POX	0	0	0	0	0	0	0	0	13	0
VAC	0	0	0	0	0	0	**1**	0	0	0

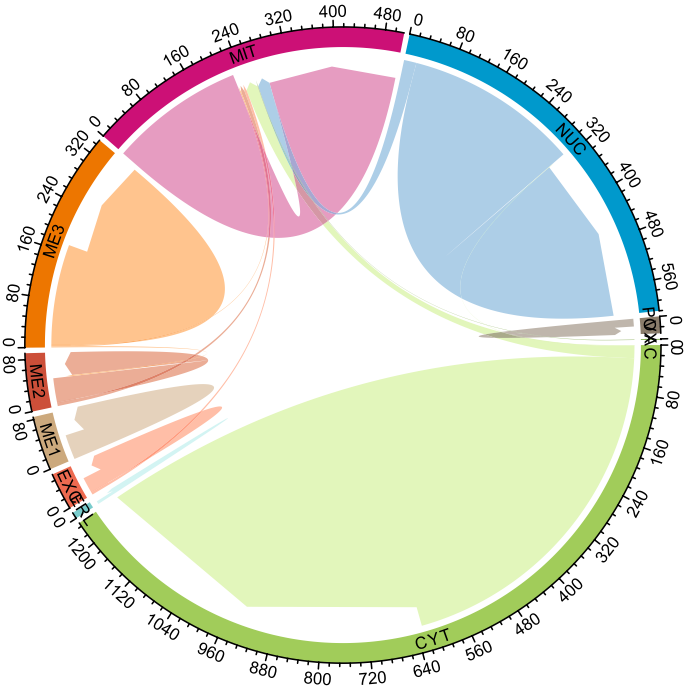


Fig. 12 Changes in prediction when increasing x_{mit} with $q_{mit} = \frac{75}{1485}$; all observations portrayed

In Table 6, it is shown that by using the *shift all ties* approach a lot of observations change their predicted class and indicate neighborhoods. These neighborhoods are visualized in the chord diagrams in Figs. 12 and 13. We find that 21 observations change their prediction from the target class *CYT* to *MIT*. This means that for 21 of the proteins the model predicts them to be mitochondrial

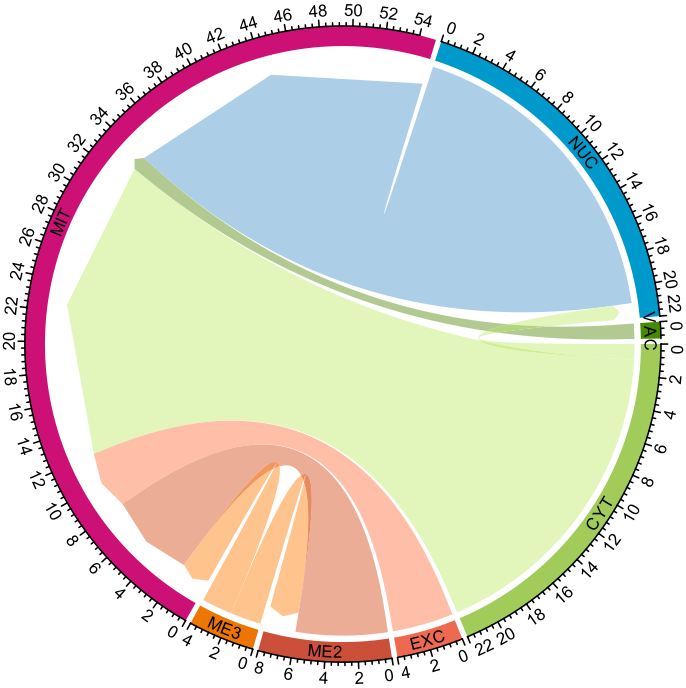


Fig. 13 Changes in prediction when increasing x_{mit} with $q_{mit} = \frac{75}{1485}$: only observations that switch their prediction portrayed

proteins instead of a cytoplasmic proteins if the score of the discriminant analysis for a mitochondrial protein was slightly larger. This absolutely makes sense as this specific score is a high indicator to have a mitochondrial protein. More technically speaking, the underlying model classifies the two classes *CYT* and *MIT* neighboring to each other with the class *MIT* having a larger value of the feature x_{mit} than class *CYT*, while the other features are the same. Here, a neighborhood between these two classes considering the feature *mit* was found.

Next, another 21 observations from the predicted class *NUC*, six observations from the predicted class *ME2*, four observations from the predicted class *EXC*, two observations from the predicted class *ME3* and one observation from the predicted class *VAC* change their prediction to target class *MIT*. This means that for 34 further proteins the model would have predicted them to be mitochondrial proteins if their respective discrimination score would have been slightly larger. More technically speaking, the underlying model classifies target class *NUC* as well as target class *ME2*, *EXC*, *ME3* and *VAC* neighboring to target class *MIT* with target class *MIT* having a larger value of feature x_{mit} than classes *NUC*, *ME2*, *EXC*, *ME3* and *VAC*, while the other features are the same. If the method's underlying model allows for very complex relations, then a single target class can be predicted at different locations within the feature space (see Sect. 3.2). This means that it does not necessarily have to be the same area where target class *MIT* is predicted and where all of the

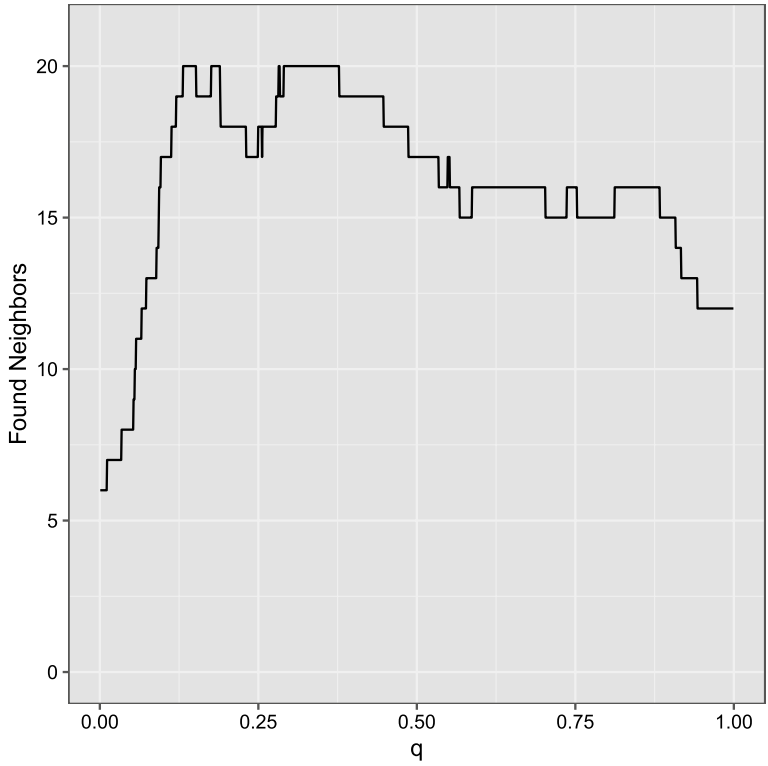


Fig. 14 Number of found neighbors for $q = \frac{v}{1485}$ with $v \in \{1, \ldots, 1484\}$

shifted observations that we see in the migration matrix end up in. In general, if the model is very complex and the target classes are scattered spotwise in the feature space, then some of these migrations might have occurred elsewhere than others. Even if two classes *A* and *B* are both neighbors with another class *C*, this does not necessarily mean that the classes *A* and *B* are also similar or even neighbors.

Also, we find that one observation changes its prediction from target class *CYT* to target class *NUC*, and two observations change their prediction from target class *ME3* to target class *ME2*. This indicates two further neighborhoods, which we have not expected, that were found by these feature shifts between target class *CYT* and *NUC*, where target class *NUC* has slightly larger values of feature x_{mit}, and between target class *ME3* and *ME2*, where target class *ME2* has slightly larger values of feature x_{mit}.

As in this example we have quite a lot of observations compared to the number of data points which ultimately switch their prediction, we propose two ways of producing the respective chord diagram. In Fig. 12, all the data points including the ones that have not changed their prediction are portrayed. Here, one can see that in general a lot of observations do not change their prediction after a

small increase of x_{mit}. Also, the only switches that can easily be made out are the switches between the target classes where multiple data points have switched their prediction. This is a good way to grasp the most important neighborhoods as they are found by multiple data points.

The changes between target classes where only a few data points have switched their prediction is harder to detect in Fig. 12. For this, we propose to also look at the chord diagram without all the data points which have not switched their prediction. This is shown in Fig. 13, where one could focus on the found neighborhoods and disregard the overall target class proportions.

Another topic to discuss is the influence of the choice of q regarding the found neighborhoods. In Fig. 14, the amount of found neighborhoods is illustrated against the size of q. Starting with a small $q > 0$, directly 6 neighborhoods are found. When q is increased, until $q \approx 0.13$ the number of found neighborhoods increases monotonously until it reaches a peak of 20 different found neighborhoods. This indicates that with a too small magnitude of the shift, not all direct neighborhoods are found. When q is further increased, the number of found neighborhoods can decrease again, indicating that some observations that were shifted to a direct neighbor are already overshooting these areas and, thus, do not indicate direct neighborhoods anymore.

To the best of our knowledge, the only other IML approach which is methodologically similar is the *anchors* method (see Ribeiro et al. 2018). In the *anchors* method, the focus would be to interpret the observations, which have not changed its prediction additionally to the specific shift sizes of the feature values. In this way, it could be determined which values of the corresponding features "anchor" the prediction of an observation.

In conclusion, by using QSM one could learn, which locations are classified by the underlying model very closely to each other with just slightly different values in one single feature. In general, it has been learnt what influence a change of the feature of interest might have on the predicted classes.

4.2 Comparison of two approaches in handling ties

When using the *shift all ties* approach, all of the observations with the same original value are increased to the exact same larger or equal value. When using the *repeatedly shift ties randomly* approach the observations with the same original value could end up at different values after the shift in each repetition. For the latter method, ten repetitions were utilized in this example.

To achieve a cleaner way of interpretation, the absolute counted number of switched predictions should be divided by the number of repetitions to get an average per repetition. Otherwise the number of switched observations might be larger than the overall number of observations in the original data set, which might

confuse the user. The corresponding results are shown in Table 7. Here, the exact same neighborhoods are found with the *repeatedly shift ties randomly* approach compared to the *shift all ties* approach. The only difference is that by using the *repeatedly shift ties randomly* approach, the influence of q is indicated a bit more precisely. This can be seen as the number of observations which have switched their predictions tend to be lower than when using the *shift all ties* approach.

The biggest difference between the two approaches of handling ties is the influence of q. When all ties are always shifted together, then even the slightest shift might lead to a lot of observations changing its prediction. If just some of them are shifting in each repetition, this process is slowed down. Parameter q should give the user the option to smoothly adjust the step size. If this is important to the user, then the *repeatedly shift ties randomly* should be preferred. If it is not important, then the *shift all ties* approach suffices.

In this example, it is shown here that using the different methods changes the resulting migration matrix, even though in this example the difference is just in the number of observations, which change their prediction. As argued above, the *repeatedly shift ties randomly* method treats the observations more fairly and additionally does not change the marginal distribution to much, which is why it generally should be preferred.

To show the difference between the *shift all ties* approach and the *repeatedly shift ties randomly* approach regarding sensibility, a small simulation study with the yeast data set is conducted. Here, the quantile shift size q_{mit} is increased in small steps and the absolute amount of switched predictions is noted. As the *repeatedly shift ties randomly* approach is conducted ten times, it is expected that this results in about ten times the amount of shifted values. Hence, for better comparison the absolute amount of shifted predictions is divided by the number of repetitions here.

Figure 15 indicates the number of predictions that change their class, when the shift size q_{mit} is steadily increased. The first major difference can be seen at the very start of the two curves, where the *shift all ties* approach produces more switched predictions even for a very small amount of q. This shows that even a very tiny value

Table 7 Migration matrix for prediction changes when raising x_{mit} by $q_{mit} = \frac{75}{1485}$ with the *repeatedly shift ties randomly* approach 10 times. In this table, the absolute amounts of the shifts after all the repetitions are divided by the number of repetitions; the observed class changes are in bold font

	CYT	ERL	EXC	ME1	ME2	ME3	MIT	NUC	POX	VAC
CYT	609.4	0	0	0	0	0	**19.7**	**0.9**	0	0
ERL	0	5.0	0	0	0	0	0	0	0	0
EXC	0	0	27.0	0	0	0	**4.0**	0	0	0
ME1	0	0	0	42.0	0	0	0	0	0	0
ME2	0	0	0	0	42.2	0	**4.8**	0	0	0
ME3	0	0	0	0	**2.0**	170.6	**1.4**	0	0	0
MIT	0	0	0	0	0	0	227.0	0	0	0
NUC	0	0	0	0	0	0	**18.9**	295.1	0	0
POX	0	0	0	0	0	0	0	0	13.0	0
VAC	0	0	0	0	0	0	**1.0**	0	0	0

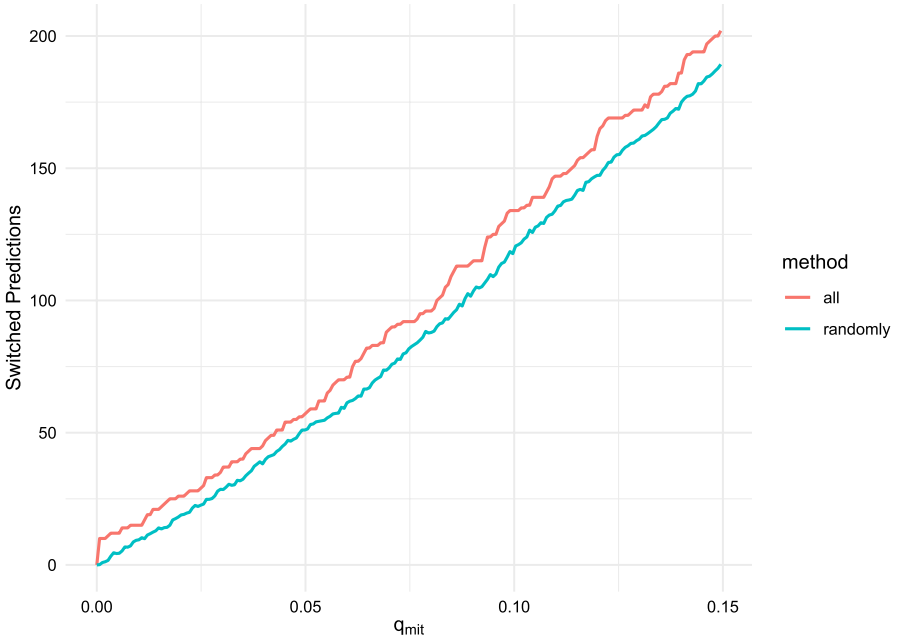


Fig. 15 Smoothness comparison between the *shift all ties* and *repeatedly shift ties randomly* approach

of $q > 0$ is sufficient to find neighborhoods. It is also obvious that the *repeatedly shift ties randomly* approach shows a much smoother curve than the *shift all ties* approach. As discussed before, this indicates again that the *repeatedly shift ties randomly* approach reacts generally more sensible to changes of q than the *shift all ties* approach.

Overall, both methods work well and reveal the same neighborhoods, and even in complex data situations they should tend to do so, if the number of performed repetitions is chosen appropriately. The only difference is that the *repeatedly shift ties randomly* approach is more sensitive to changes in q compared to the *shift all ties* approach.

5 Conclusion

In this work, a method to determine neighborhoods between predicted classes in a fitted model is presented, accompanied by examples illustrating the purpose of the method and how to correctly interpret the corresponding results. The method improves the understanding of the partition of the feature space of a statistical classification model and can be simply visualized and interpreted with the aid of chord diagrams. One of the main advantages is to illustrate the influence of the features on the prediction of the target classes. This can help to understand the fitted black box model, even though the model might be hardly interpretable

for practitioners, and thus can improve the trustworthiness of the model. Another advantage is that the method helps to gain insight about the partition of the feature space of the corresponding classification model at hand. It can help to illustrate, which classes are modeled similar by the underlying classification model, but differ within the feature of interest.

Similar to most statistical tools, the method is an approximation of the reality, which becomes more meaningful the better the model at hand performs. However, the real additional value of the proposed method is its wide and unrestricted applicability to any kind of classification model. In general, the method can also be applied in very high-dimensional and complex settings. The only conditions are that the fitted model at hand builds upon a feature space and returns a categorical output or predicted probabilities for the different response categories.

The greatest risks when using this method are probably false interpretations of results, as the pitfalls in this regard are manifold. Of course, the method only describes the underlying model, and, hence, heavily relies on its goodness-of-fit and adequacy. The usage of a weak model, which badly represents reality, will almost certainly lead to unpractical interpretations by the proposed method (as any other model-describing methods would do as well). Another source for potential misinterpretation could occur, if the feature shifts are too strong or too weak, such that some neighboring classes are skipped or simply not found.

Nevertheless, the examples in this work show the variety of fields this method could be applied to. First of all, QSM can help to determine which classes are modeled close to each other by the model at hand and can show in which order they are modeled next to each other with respect to the shift. Even more, this method shows to which class predictions generally tend to switch if one or multiple specific feature(s) of interest are changed, which can be relevant, e.g., in clinical usage when a specific medicine should be used to alter the features.

Since extrapolation typically is a problem in many statistical modeling tasks and typically gets worse when the model complexity rises, care has to be taken when large feature shifts are used. These might force the model to predict new classes in a region of the feature space where data points are rather unlikely or even impossible. In order to avoid this problem, it is recommended to start with rather small feature shifts, assuming that very small shifts do not create impossible observations. As it is typically hard to determine what generally defines a "likely" observation, we recommend that this should be decided manually by the user. Altogether, this manuscript aims at providing sufficient advice to enable practitioners to safely use this tool for meaningful interpretations.

Additionally, it should be investigated in future work how different types of outliers (e.g., "trivial" and "non-trivial" outliers, see Keller et al. 2012) might influence the outcome and interpretation of the proposed method and how possible issues in this regard could be avoided.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare there is no conflict of interest. The code used for the examples can be found in a public github repository: https://github.com/habbeda1/QSM.

References

Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
Dua, D., Graff, C.: UCI machine learning repository (2017)
Fox, J., Hong, J.: Effect displays in r for multinomial and proportional-odds logit models: extensions to the effects package. J. Stat. Softw. **32**(1), 1–24 (2009)
Fox, J., Weisberg, S.: An R Companion to Applied Regression, 3rd edn. Sage, Thousand Oaks (2019)
Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. J. Comput. Graph. Stat. **24**(1), 44–65 (2015)
Gu, Z., Gu, L., Eils, R., Schlesner, M., Brors, B.: circlize implements and enhances circular visualization in r. Bioinformatics **30**, 2811–2812 (2014)
Hooker, G.: Diagnostics and extrapolation in machine learning. Ph.D. thesis, Stanford, CA, USA (2004)
Horton, P., Nakai, K.: A probablistic classification system for predicting the cellular localization sites of proteins. Intell. Syst. Mol. Biol. **8**, 109–115 (1996)
Keller, F., Muller, E., Bohm, K.: HiCS: high contrast subspaces for density-based outlier ranking. IEEE (2012)
Molnar, C.: Interpretable machine learning: a guide for making black box models explainable (2019)
R Core Team.: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna (2021)
Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: high-precision model-agnostic explanations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32(1) (2018)
Tutz, G., Schauberger, G.: Visualization of categorical response models: from data glyphs to parameter glyphs. J. Comput. Graph. Stat. **22**(1), 156–177 (2012)
Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: networked science in machine learning. SIGKDD Explor. **15**(2), 49–60 (2013)
Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S, 4th edn. Springer, New York (2002)