Ordinal Prototype-Based Classifiers

Andre Burkovski*, Lyn-Rouven Schirra*, Florian Schmid, Ludwig Lausser†, Hans A. Kestler†

Abstract The identification of prototypical patterns is one of the major goals in the classification of microarray data. Prototype-based classifiers are of special interest in this context, since they allow a direct biological interpretation. In this work we present prototype-based classifiers that rely on ordinal-scaled data. Advantage of these ordinal-scaled signatures is their invariance to a wide range of data transformations. Standard prototype-based classifiers can be modified to this type of data by utilizing rank-distances and rank-aggregation procedures.

Andre Burkovski
Institute of Medical Systems Biology, Ulm University, 89069 Ulm, Germany,
✉ andre.burkovski@uni-ulm.de

Florian Schmid
Institute of Medical Systems Biology, Ulm University, 89069 Ulm, Germany,
✉ florian-1.schmid@uni-ulm.de

Lyn-Rouven Schirra
Institute of Medical Systems Biology and Institute of Number Theory and Probability Theory, Ulm University, 89069 Ulm, Germany,
✉ lyn-rouven.schirra@uni-ulm.de

Ludwig Lausser
Institute of Medical Systems Biology, Ulm University, 89069 Ulm, Germany,
✉ ludwig.lausser@uni-ulm.de

Hans A. Kestler
Head, Institute of Medical Systems Biology, Ulm University, 89069 Ulm, Germany,
✉ hans.kestler@uni-ulm.de

*equal contribution · † equally contributing senior authors

ARCHIVES OF DATA SCIENCE, SERIES A
(ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. 2, No. 1, 2017

DOI 10.5445/KSP/1000058749/29
ISSN 2363-9881

In this study, we compare the proposed methods with standard classifiers. They are examined in experiments with and without feature selection on a panel of publicly available microarray datasets. We show that the proposed techniques result in the construction of different signatures that improve classification performance.

1 Introduction

In life sciences, microarrays allow researchers to investigate processes in cells and tissues, e.g. with the goal of identifying prognostic markers or potential targets for therapies. Microarrays can measure gene expression levels for thousands of probes simultaneously by quantifying light intensities. Many steps between preparation of organic samples to postprocessing are influenced by different sources of noise so that the final data used for the classification may be distorted. One of the robust transformations that can help here, is the sample-wise transformation to an ordinal scale.
Figure 1 illustrates this transformation as well as the resulting class prototypes. In some cases, small effect differences can result in a wider separation per feature. Such feature-wise separation further can accumulate across features which in turn could produce ordinal-scaled prototypes profiles that are better suited to distinguish the individual classes. Furthermore, it is known that classifiers based on these rankings are invariant against feature-wise strictly monotone data transformations (Lausser et al, 2016).

Many microarray datasets describe the gene expression of single samples that are taken from a biological phenotype. The identification of a common characteristic of a phenotype can provide researchers with new insights (Little et al, 2009; Biehl, 2014). Hence, we focus on the prototype-based classifiers that essentially solve the tasks of computing the common characteristic for each class (Lausser et al, 2014). Furthermore, feature selection methods are applied in order to identify those features (genes) that contribute most to the accuracy of the classification in a robust way (Lausser et al, 2013; Schirra et al, 2016). When developing prototype-based classifiers on ordinal-scaled data or rankings we can apply additional distance metrics and methods to combine given input rankings, namely rank aggregation. An overview on rank aggregation methods can be found in Schalekamp and Zuylen (2009) and Dwork et al (2001) and a

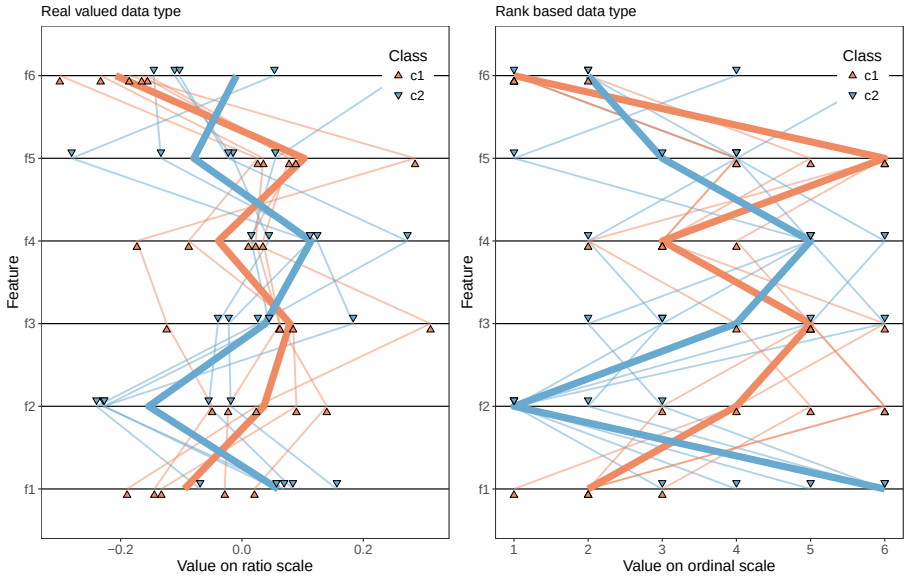

Fig. 1 Illustration of the difference between data types and computed prototypes. The y-axis in both plots denotes features in the data. In the left plot, the x-axis shows ratio scaled values of the features. In the right plot, the x-axis shows ordinal scaled values, i.e. the sample-wise rank transformation. A sample profile is indicated by a thin colored line. A prototype profile is shown as a thick line. For real-valued samples, the class' prototype corresponds to a centroid. For rank transformed samples, the class' prototype is computed using Borda rank aggregation. The color of lines and points indicates the class of samples or class of a prototype, respectively.

comparison of the methods for identifying signatures, i.e. sets of genes, can be found in Burkovski et al (2014); Kraus et al (2015).

In the following we will present the formal framework for the prototype-based classifiers, introduce the distance metrics, feature selection methods, and aggregation methods, describe our experimental setting, report the results, and finally discuss the performance of the rank-based classification methods.

2 Methods

Microarrays are often used for analysing gene expression in tissues or other biological samples. These arrays measure the expression of thousands of genes simultaneously. Together with a limited number of samples this results usually in data sets of a very high dimensionality and low cardinality. A sample, i.e. data point, is represented by a high-dimensional real valued vector $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T \in \mathscr{X} \subseteq \mathbb{R}^n$. In a classification setting, each sample has a biological phenotype, e.g. cancerous vs. normal tissue – a label $y \in \mathscr{Y} = \{-1, 1\}$ from a set of labels. A dataset is described by a set of $i = 1 \ldots k$ samples $\mathscr{S} = \{(\mathbf{x}_i, y_i)\}$ with their respective label. In order to simplify formal notation, samples can be divided into their respective class sets $\mathscr{T}^{\psi} = \{(\mathbf{x}, y) \in \mathscr{S} \,|\, \psi = y\}$. The goal of the classifier, a mapping function $c : \mathscr{X} \longrightarrow \mathscr{Y}$, is to predict labels of profiles $\mathbf{x} \in \mathscr{V} \subseteq \mathbb{R}^n$. Each classifier c is adapted beforehand to the data from the available training set $\mathscr{S} \supset \mathscr{T}$.

In this article we use the empirical error rate in order to compare the performance of the real- and ranked-based classifiers based on a test data set $\mathscr{V} = \mathscr{S} \setminus \mathscr{T}$:

$$R_{emp}(c, \mathscr{V}) = \frac{1}{|\mathscr{V}|} \sum_{(\mathbf{x}, y) \in \mathscr{V}} \mathbb{I}(c(\mathbf{x}) \neq y). \tag{1}$$

Here, we analyze the performance of original classifiers and their rank-based counterparts based on rank-transformed profiles. The rank transformation converts the profile's domain $\mathscr{X} \subseteq \mathbb{R}^n$ into a permutation of the set $\{1, \ldots, n\}^n$ and is derived by using the $\mathtt{rank}(\mathbf{x}) := (\mathtt{rk}_{\mathbf{x}}(x^{(1)}), \ldots, \mathtt{rk}_{\mathbf{x}}(x^{(n)}))^T$ function where $\mathtt{rk}_x$ is a feature-wise transformation of the values with regard to the vector $\mathbf{x}$:

$$\mathtt{rk}_{\mathbf{x}}(x^{(k)}) = \left| \{x^{(j)} | x^{(j)} > x^{(k)}, j = 1 \ldots n, \mathbf{x} = (x^{(1)}, \ldots, x^{(n)})\} \right| + 1. \tag{2}$$

We use π_j for the rank-transformed sample j as $\pi_j = \mathtt{rank}(\mathbf{x}_j)$. The rank-value of a feature i of a sample $x_j^{(i)}$ is denoted by $\pi_j^{(i)}$. For rank-based classification, the rank transformation is applied to both training samples $\mathscr{T}$ and test samples $\mathscr{V}$:

$$\mathscr{T}_{\text{rk}} = \left\{ \big(\mathtt{rank}(\mathbf{x}), y\big) \,|\, (\mathbf{x}, y) \in \mathscr{T} \right\} \quad \text{and} \quad \mathscr{V}_{\text{rk}} = \left\{ \big(\mathtt{rank}(\mathbf{x}), y\big) \,|\, (\mathbf{x}, y) \in \mathscr{V} \right\}.$$

Rank transformations allow to use other distances than the conventional that may be more suited for comparison of rankings as well as new methods for determining the centroids in prototype-based classifiers by utilizing rank aggregation methods.

2.1 Prototype-based classifiers

The basic idea of prototype-based classifiers is to find reference points $\mathscr{P}_{\mathscr{T}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathscr{P}_{\mathscr{T}}|}$, derived from the training sample set $\mathscr{T}$, that represent a class. The principle of the prototype-based classifier is to compute the pairwise distances between the reference points, or prototypes, and the new unclassified sample $\mathbf{v} \in \mathscr{V}$

$$D_{\mathbf{v}}^{\mathscr{P}_{\mathscr{T}}} = \left\{ d(\mathbf{v}, \mathbf{x}) \,|\, (\mathbf{x}, y) \in \mathscr{P}_{\mathscr{T}} \right\} \tag{3}$$

with $d(\cdot,\cdot)$ being the Euclidean distance in case of the original, real value-based classifiers, or distances listed in Table 2 for rank-based classifiers. The decision for the label $y \in \mathscr{Y}$ of the sample $\mathbf{v}$ is based on the neighbourhood $\mathrm{NN}_k(\mathbf{v}, \mathscr{P}_{\mathscr{T}})$ limited to k closest prototypes

$$\mathrm{NN}_k(\mathbf{v}, \mathscr{P}_{\mathscr{T}}) = \left\{ (\mathbf{x}, y) \in \mathscr{P}_{\mathscr{T}} \mid \left| \left\{ d_i \in D_{\mathbf{v}}^{\mathscr{P}_{\mathscr{T}}} \mid d_r \in D_{\mathbf{v}}^{\mathscr{P}_{\mathscr{T}}} : d_i > d_r \right\} \right| < k \right\}. \tag{4}$$

and typically the majority-vote is utilized for the label decision

$$c(\mathbf{v}, \mathscr{P}_{\mathscr{T}}, k, d(\cdot,\cdot)) = \underset{y \in \mathscr{Y}}{\operatorname{argmax}} \left| \{ (\mathbf{x}, y) \in \mathrm{NN}_k(\mathbf{v}, \mathscr{P}_{\mathscr{T}}) \} \right|. \tag{5}$$

The list of the analyzed prototype-based classifiers can be found in Table 1.

Table 1 The table lists the prototype-based classifiers that are analyzed in our experiments.

classifier	**prototype set** $\mathscr{P}_{\mathscr{T}}$	k**-neighborhood**
k-Nearest Neighbor (k-NN) (Fix and Hodges, 1951)	$\mathscr{P}_{\mathscr{T}} = \mathscr{T}$	$k = 1,3,5,7$
Nearest Prototype Classifier (NPC)	$\mathscr{P}_{\mathscr{T}} = \{\mathbf{p}_\psi\}_{\psi \in \mathscr{Y}}$ $\mathbf{p}_\psi = \underset{(\mathbf{x}^*, y^*) \in \mathscr{T}\psi}{\operatorname{argmin}} \sum_{(\mathbf{x}, y) \in \mathscr{T}\psi} d(\mathbf{x}, \mathbf{x}^*)$	$k = 1$
Representative Prototype Sets (RPS) (Lausser et al, 2012)	$\mathscr{P}_{\mathscr{T}} = \underset{\substack{\mathscr{P}^* = \{(\mathbf{x}, \psi)\}_{\psi \in \mathscr{Y}} \\ \mathscr{P}^* \subset \mathscr{T}}}{\operatorname{argmin}} R_{emp}(c_{\mathscr{P}^*}, \mathscr{T})$	$k = 1$
Nearest Centroid Classifier (NCC)	$\mathscr{P}_{\mathscr{T}} = \{\mathbf{p}_\psi\}_{\psi \in \mathscr{Y}}$ $\mathbf{p}_\psi = \frac{1}{\lvert\mathscr{T}\psi\rvert} \sum_{(\mathbf{x}, y) \in \mathscr{T}\psi} \mathbf{x}$	$k = 1$

The rank-based counterpart to the real value-based classifiers is based on the ranked training set $\mathscr{T}_{\mathrm{rk}}$. With respect to the classifiers k-NN, RPS, and NPC only the distance computation changes. In both, training phase and prediction phase, any distance $d(\cdot,\cdot)$ can be utilized in order to determine the prototypes or to predict the label of the new sample, respectively Sect. 2.2. The rank aggregation

methods, introduced in Sect. 2.3, play a major role in the NCC. Here, instead of computing mean vectors, rank aggregation methods are used in order to compute a rank-based centroid – the consensus ranking. Hence the computation of the prototypes for NCC-ranked (NCCra) becomes:

$$\mathscr{P}_{\mathscr{T}}^{\mathrm{rk}} = \bigcup_{\psi \in Y} \mathscr{P}_{\mathscr{T}}^{\psi} \text{ with } \mathscr{P}_{\mathscr{T}}^{\psi} = \left\{ \mathtt{aggr}(\mathscr{T}_{\mathrm{rk}}^{\psi}) \right\}. \tag{6}$$

2.2 Distance functions

The prototype-based classifiers we consider here use distances between profiles in order to measure "closeness" to a prototype. Many distance metrics were developed for real valued vectors - but only some were analyzed in the context of rank-based data and classification. Cha (2007) suggests different families of distance functions, namely Minkowski, L_1, Intersection, and Fidelity, of which we chose two from each family (see Table 2). These distances can be applied to rankings by replacing $\mathbf{x}_i$ by their ranked counterpart $\mathtt{rank}(\mathbf{x}_i)$. It is worth noting, that values of rankings are always positive and thus Fidelity family distances become applicable.

Table 2 Distance functions for computing the distance between a sample and a prototype.

Minkowski family	L_1 **family**
Euclidean distance: $d_{euc}(\mathbf{x}_i,\mathbf{x}_j) = \sqrt{\sum_m (x_i^{(m)} - x_j^{(m)})^2}$	Bray-Curtis distance: $d_{bra}(\mathbf{x}_i,\mathbf{x}_j) = \frac{\sum_m \lvert x_i^{(m)} - x_j^{(m)} \rvert}{\sum_m x_i^{(m)} + x_j^{(m)}}$
Manhattan distance: $d_{man}(\mathbf{x}_i,\mathbf{x}_j) = \sum_m \lvert x_i^{(m)} - x_j^{(m)} \rvert$	Soergel distance: $d_{soe}(\mathbf{x}_i,\mathbf{x}_j) = \frac{\sum_m \lvert x_i^{(m)} - x_j^{(m)} \rvert}{\sum_m \max(x_i^{(m)}, x_j^{(m)})}$
Intersection family	**Fidelity family**
Motyka distance: $d_{mot}(\mathbf{x}_i,\mathbf{x}_j) = 1 - \frac{\sum_m \min(x_i^{(m)}, x_j^{(m)})}{\sum_m x_i^{(m)} + x_j^{(m)}}$	Hellinger distance: $d_{hel}(\mathbf{x}_i,\mathbf{x}_j) = \sqrt{\sum_m (\sqrt{x_i^{(m)}} - \sqrt{x_j^{(m)}})^2}$
Ruzicka distance: $d_{ruz}(\mathbf{x}_i,\mathbf{x}_j) = 1 - \frac{\sum_m \min(x_i^{(m)}, x_j^{(m)})}{\sum_m \max(x_i^{(m)}, x_j^{(m)})}$	Chord distance: $d_{cho}(\mathbf{x}_i,\mathbf{x}_j) = \sqrt{\sum_m \left(\frac{x_i^{(m)}}{\sqrt{\sum_l \left(x_i^{(l)}\right)^2}} - \frac{x_j^{(m)}}{\sqrt{\sum_l \left(x_j^{(l)}\right)^2}} \right)^2}$

Table 3 Panel A lists the rank aggregation methods that are analyzed in our experiments. Panel B lists the feature selection algorithms used for our experiments. We utilize following additional notation $\mathscr{X}_{\mathscr{T}}^{(i)} = \{x^{(i)} \,|\, (\mathbf{x}, y) \in \mathscr{T}\}$ and $\mathscr{Y}_{\mathscr{T}} = \{y \,|\, (\mathbf{x}, y) \in \mathscr{T}\}$ in this context.

A

Rank aggregation method	**consensus ranking**	**score**
Borda	$\boldsymbol{\sigma}_{\texttt{borda}} = \texttt{rank}(\mathbf{s}_{\texttt{borda}})$	$s_{\texttt{borda}}^{(i)} = \left(\frac{1}{\lvert\Pi\rvert} \sum_{\boldsymbol{\pi} \in \Pi} \boldsymbol{\pi}^{(i)} \right)$
Copeland	$\boldsymbol{\sigma}_{\texttt{cope}} = \texttt{rank}(-\mathbf{s}_{\texttt{cope}})$	$s_{\texttt{cope}}^{(i)} = \left(\sum_{j \neq i} \left(\mathbb{I}(v_{i,j} > l_{i,j}) + \frac{1}{2}\mathbb{I}(v_{i,j} = l_{i,j}) \right) \right)$ with $v_{i,j} = \sum_{\boldsymbol{\pi} \in \Pi} \mathbb{I}(\pi^{(i)} > \pi^{(j)})$ and $l_{i,j} = \sum_{\boldsymbol{\pi} \in \Pi} \mathbb{I}(\pi^{(i)} < \pi^{(j)})$
Robust Rank Aggregation	$\boldsymbol{\sigma}_{\texttt{rra}} = \texttt{rank}(\mathbf{s}_{\texttt{rra}})$	$s_{\texttt{rra}}^{(i)} = \left(\min_{j \in 1 \ldots \lvert\Pi\rvert} \left(\sum_{\gamma=j}^{\lvert\Pi\rvert} \binom{\lvert\Pi\rvert}{\gamma} \left(\hat{\pi}_{(j)}^{(i)}\right)^{\gamma} \left(1 - \hat{\pi}_{(j)}^{(i)}\right)^{\lvert\Pi\rvert - \gamma} \right) \right)$ with $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}/n$ and resorted $\forall j : \hat{\pi}_{(j)}^{(i)} \leq \hat{\pi}_{(j+1)}^{(i)}$
Spearman's footrule optimal Rank Aggregation	$\boldsymbol{\sigma}_{\texttt{spear}} = \underset{\boldsymbol{\sigma}}{\operatorname{argmin}} \sum_{i=1}^{n} \sum_{\boldsymbol{\pi} \in \Pi} \lvert\pi^{(i)} - \sigma^{(i)}\rvert$	-
Weighted Borda	$\boldsymbol{\sigma}_{\texttt{borda-w}} = \texttt{rank}(\mathbf{s}_{\texttt{borda-w}})$	$s_{\texttt{borda-w}}^{(i)} = \frac{1}{\lvert\Pi\rvert} \sum_{l=1}^{\lvert\Pi\rvert} w_l \cdot \pi_l^{(i)}$

B

Feature selection algorithm	**score**
Pearson correlation (COR)	$score_{\text{COR}}^{(\mathscr{T},i)} = \left\lvert \frac{\sum_{(\mathbf{x},y) \in \mathscr{T}} (x^{(i)} - \text{mn}(\mathscr{X}_{\mathscr{T}}^{(i)}))(y - \text{mn}(\mathscr{Y}_{\mathscr{T}}))}{\sqrt{\sum_{j=1}^{m} \left(x_j^{(i)} - \text{mn}(\mathscr{X}_{\mathscr{T}}^{(i)})\right)^2 \cdot \sum_{j=1}^{m} \left(y_j - \text{mn}(\mathscr{Y}_{\mathscr{T}})\right)^2}} \right\rvert$
Threshold number of misclassification (TNoM) (Ben-Dor et al, 2000)	$score_{\text{TNoM}}^{(\mathscr{T},i)} = \max_{\substack{d \in \{-1,+1\} \\ t \in \mathbb{R}}} \sum_{(x,y) \in \mathscr{T}} \mathbb{I}\left(y = \text{sign}\left(d \cdot \left(x^{(i)} - t\right)\right)\right)$
Signal-to-noise ratio (SNR) (Yeang et al, 2001; Cuperlovic-Culf et al, 2005)	$score_{\text{SNR}}^{(\mathscr{T},i)} = \left\lvert \frac{\text{mn}\left(\mathscr{X}_{\mathscr{T}-1}^{(i)}\right) - \text{mn}\left(\mathscr{X}_{\mathscr{T}+1}^{(i)}\right)}{\text{sd}\left(\mathscr{X}_{\mathscr{T}-1}^{(i)}\right) + \text{sd}\left(\mathscr{X}_{\mathscr{T}+1}^{(i)}\right)} \right\rvert$

2.3 Rank Aggregation Methods

The goal of rank aggregation methods is the computation of a consensus ranking $\boldsymbol{\sigma} \in \{1,\ldots,n\}^n$ which has least disagreements with input rankings. In our case, the input rankings correspond to the ranked profiles $\Pi = \{\texttt{rank}(\mathbf{x}) | (\mathbf{x}, y) \in \mathscr{T}_{\text{rk}}\}$ from the training set $\mathscr{T}_{\text{rk}}$. The disagreements are usually computed by the Kendall-τ correlation coefficient between $\boldsymbol{\sigma}$ and the rankings in $\mathscr{T}_{\text{rk}}$ (Kendall, 1938). However, finding an optimal $\boldsymbol{\sigma}$ using Kendall-τ is an NP-hard problem (Dwork et al, 2001) and thus many rank aggregation methods use heuristics in order to find a close to optimal consensus ranking. Here we consider a representative set of rank aggregation methods that utilize different heuristic approaches. These methods have in common that they compute a score for a each feature and the resulting consensus ranking is retrieved by ranking the score. To simplify the notation we denote the aggregation function as $\texttt{aggr} : \Pi \mapsto \{1,\ldots,n\}^n$ that computes the consensus ranking $\boldsymbol{\sigma}$ given some input rankings from $\Pi = \mathscr{T}_{\text{rk}}$. An overview of the utilized rank aggregation methods can be found in panel A of Table 3 on the aforementioned page 7.

2.4 Feature selection

Gene expression data contains a large amount of measurements, i.e. genes or features, that often do not result in an improved classification error. One can reduce the high dimensionality by selecting a small subset of features that may be considered informative for the chosen classifier. Feature selection is a function that maps a training set $\mathscr{T}$ to a new training set $\mathscr{T}'$ with reduced dimensionality,

$$\begin{aligned} f: &\qquad \mathscr{T} \to \mathbf{i} \in \mathscr{I} = \{\mathbf{i} \in \mathbb{N}^{\hat{n} \leq n} | i_k < i_{k-1}, 1 \leq i_k \leq n\}. \\ s: &\quad (\mathscr{I}, \mathscr{T}) \to \mathscr{T}' \end{aligned} \tag{7}$$

Sample vector $\mathbf{x}$ will then be mapped to a lower-dimensional representation $\mathbf{x}^{(\mathbf{i})} = (x^{(i_1)}, \ldots, x^{(i_{\hat{n}})})^T$ by using the derived index vector $\mathbf{i} = (i_1, \ldots, i_{\hat{n}})^T$.

Feature selection methods can be categorized by the type of knowledge they utilize to select features. Purely *data-driven* algorithms only consider the knowledge derived from the data itself, i.e. the measurements and the associated class labels. *Model-based* methods, like wrapper and embedded methods (Kohavi and John, 1997; Ben-Dor et al, 2000), utilize the feedback from a classification

algorithm to assess the predictive utility of the features. In our experiments we apply purely data-driven methods. The different methods assign scores to each feature based on the correlation between the feature vector and the class labels (COR), the threshold number of misclassification (TNoM), or the signal-to-noise ratio (SNR) to assess their importance (Panel B of Table 3). The best features (highest score) are then selected based on the ranking of these scores.

3 Experimental Setup

Our goal is to test whether or not rank-based classifiers are able to outperform the real value-based classifiers on the microarray datasets. An overview of the datasets used to compare the performance is shown in Table 4.

To evaluate the performance we compute the empirical error rate $\mathscr{R}_{emp}$ in a cross-validation (Pierre A. Devijver, 1982) setting. We compare the rank-based and real value-based variants of four prototype-based classifiers k-NN, *RPS*, *NPC*, and *NCC*. For each classifier we use three different feature-selection methods listed in the previous section and for each method five different number of selected features (5, 50, 100, 250, 500). These threshold limits, in the context of microarray data, represent a number of biomarkers used to distinguish a class. If one wants to determine a set of highly informative biomarkers, a low number of features is preferable. On the other hand, a couple of hundreds of patterns are usually already enough to obtain accurate classification results due to Covers' theorem (Cover, 1965) and as shown in simulation studies (Schirra et al, 2016). After feature selection all ranks are re-ranked. For real value-based classifiers we use d_{euc} as a distance metric and for rank-based classifiers we utilize all distance metrics listed in Table 2. Rank-based classifiers allow the use of a different distance metric for training and prediction, thereby increasing the number of parameters. The *NCC* includes the parameter for the rank aggregation method and we included all five presented rank aggregation methods.

Here, we compute the error in a 10×10 cross-validation setting. In order to report on sampling independent error identical training and test set splits are used for all classifiers. Since we focus on the comparison of the rank-based (rk) and real value-based (re) classifiers, we additionally report the error difference between specific classifiers

$$E_{\text{diff}} = \text{ERR}^{\text{re}}_{10\times 10} - \text{ERR}^{\text{rk}}_{10\times 10} \tag{8}$$

which we are using in our graphics. In case the real value-based classifier outperforms the rank-based one the difference E_{diff} will be positive. All cross-validation experiments are simulated with TunePareto (Müssel et al, 2012).

Table 4 The table lists the datasets and their description that we use in order to compare real value and rank-based classifiers.

Id	Citation	GEO id	Class	Features	Samples	Class -1	Class 1
d1	Armstrong et al (2002)	-	Leukemia	12559	72	24	48
d2	Dyrskjøt et al (2003)	-	Bladder cancer	7071	40	20	20
d3	Kuner et al (2009)	GSE10245	Lung cancer	54613	58	40	18
d4	Badea et al (2008)	GSE15471	Pancreatic cancer	54613	78	39	39
d5	Sun and Goodison (2009)	GSE25136	Prostate cancer	22215	79	40	39
d6	Alter et al (2011)	GSE25507	Autism	54613	146	64	82
d7	Lu et al (2014)	GSE53890	Brain cells	54613	41	20	21

4 Results

The empirical error rates R_{emp} of the original classifiers on the real-value datasets are reported in Tables 5–7. Figure 2 shows the direct comparison of the real value-based and rank-based classifiers with respect to a feature selection method. For each pair of classifiers the rectangle summarizes the performance based on the sign of E_{diff}, i.e. compares the minimal error over all corresponding parameters for each classifier and feature selection method. Clearly, almost all real value-based classifiers outperform the rank-based classifiers when the number of selected features in the feature selection step is very low (5 selected features). However, with an increasing number of selected features, there are more and more cases where rank-based classifiers become better with regard to the empirical error rate. The tile plot in Fig. 2 reveals further specific information about the applicability of the rank-based classifiers.

Results for k-NN classifiers: The classification error for the **d1** dataset is generally very low, i.e. is close to zero (see also Table 5), and therefore any performance gain using rank-based classifiers is hardly achievable with regard to the *k*-NN classifier. However, rank-based *k*-NN classifier is able to slightly outperform the real value-based *NCC*, *NPC*, and *RPS* classifiers on **d1**. Ad-

ditionally, the rank-based k-NN classifier outperforms any real value-based prototype classifier on the **d5** and **d2** dataset with 250 features - even independent of the choice of the feature selection method. It has, however, lower performance when tested on the **d4** dataset when compared to the real value-based classifiers. However, in many cases the performance of the rank-based k-NN classifier is better than any of the real value-based *NCC*, *NPC*, and *RPS* classifiers.

Results for RPS classifiers: The rank-based *RPS* classifier, with any of the applied feature selection methods, outperforms any other real value-based classifier when only 100 features are selected on the **d2** and **d3** datasets. However, in many cases the real value-based k-NN classifier still has a better performance. Yet, the rank-based *RPS* can outperform real value-based *NCC*, *NPC*, and *RPS* classifiers in many cases.

Results for NPC classifiers: In rare cases, such as for dataset **d7** and 250 features selected by the TNoM method and for dataset **d5** and 250 features selected by the SNR method, the rank-based NCP classifier returns better results than any of the real value-based classifiers. But in the majority of simulations, either another rank-based or real value-based classifiers have relatively better performance.

Results for NCC classifiers: Similar to the other classifiers, the rank-based *NCC* classifier, has too cases where it outperforms any other real value-based classifier, like for **d4** and **d7** with 100 selected features. Furthermore, it has in many cases better performance than its real value-based counterpart.

Classification of datasets: Simulation results show that rank-based classification in many cases outperforms the real value-based classification methods on the **d5** dataset. In contrast, in the majority of cases real value-based classifiers have better performance on the **d3** and **d4** datasets, but parameter settings exist for which a rank-based classifier can get an edge over the real value-based variants.

Parameter space: Figure 3 shows the performance of the rank-based *NCC* classifier with regard to the parameters chosen in the simulation. We chose to show the *NCC* parameters (Figures. for the other methods can be found in the supplement) since it includes the choice of the aggregation method and the distance metric that reveals many influences to classification performance. In the figure we show the classification performance for 500 features.

The distance metric d_{hel} has the lowest classification error when combined with the Borda-based aggregation methods on the dataset **d4**. Even when combined with Copeland or Spearman rank aggregation, the usage of the d_{hel} metric

is preferred over the other options. In contrast, the $d_{bra}, d_{man}, d_{mot}, d_{ruz}$, and d_{soe} distance metrics in combination with the Spearman aggregation method have the lowest classification error on the **d5** dataset. Similarly, the choice of one of these distance metrics has a positive influence on the classification error despite the choice of the aggregation method. Lastly, d_{euc} and d_{cho} have the best performance in the **d6** dataset when combined with the Copeland or Spearman aggregation method. Notably, both d_{euc} and d_{cho} seem to be unsuitable when used in combination with the Robust Rank Aggregation method.

Table 5 Error rates for the k-NN classifier on real valued datasets. The parameter combination of k and feature selection method that results in the best error rate for individual feature number threshold per dataset is shown in parenthesis.

k-**NN**	Feature number threshold				
Dataset	5	50	100	250	500
d1	0.015 (cor,k=1,3,5)	0.000 (tnom,k=1,3,5,7)	0.001 (snr/tnom,k=1)	0.001 (snr/tnom,k=1)	0.001 (cor/snr/tnom,k=1)
d2	0.078 (cor/snr,k=3)	0.055 (snr,k=1,3)	0.052 (snr,k=3)	0.06 (snr/tnom,k=1,3)	0.058 (snr,k=3)
d3	0.067 (tnom,k=5)	0.064 (tnom,k=1)	0.06 (tnom,k=1)	0.047 (cor,k=1)	0.045 (snr,k=1)
d4	0.091 (snr,k=3)	0.091 (cor/tnom,k=5,7)	0.091 (cor/snr/tnom,k=7)	0.09 (cor/snr/tnom,k=7)	0.09 (cor/tnom,k=7)
d5	0.284 (snr,k=5)	0.305 (cor,k=5)	0.295 (cor,k=3)	0.316 (cor,k=5)	0.333 (snr,k=7)
d6	0.401 (tnom,k=7)	0.323 (cor,k=7)	0.294 (snr,k=3)	0.321 (cor,k=3)	0.326 (snr,k=5)
d7	0.102 (cor,k=7)	0.107 (tnom,k=3)	0.117 (cor,k=3)	0.122 (tnom,k=3)	0.154 (cor/snr/tnom,k=1,3)

5 Discussion and Conclusion

Simulation results suggest that rank-based classification methods are able to outperform real value-based classification methods. The difference may be very small and is not always significant. It is, however, surprising that for most parameter combinations rank-based classification methods have a comparable performance in our experimental setting. In this setting we focus on microarray datasets – data that represents measurements of the gene expression level via light intensity ratios. Such data-in every phase from processing biological samples to its final numerical form is influenced by noise.

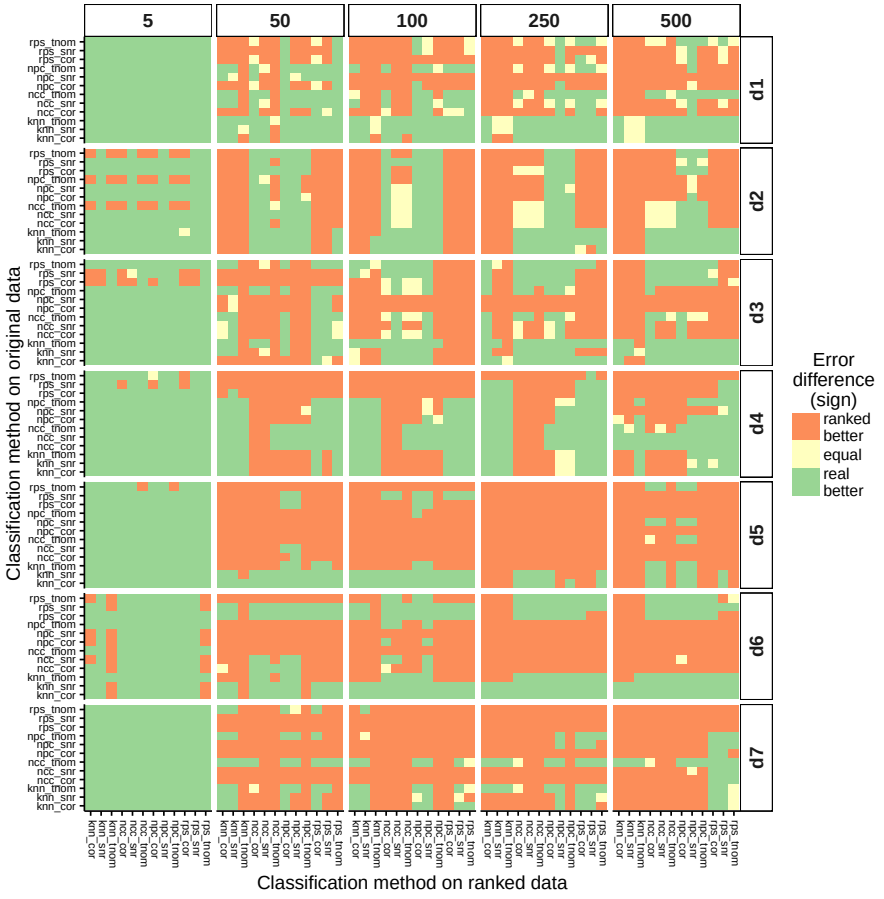

Fig. 2 Summary of the simulation results regarding the sign of E_{diff}. Each row shows the summary for a single dataset and each column shows the results for the selected number of features. Each tile in the plot represents the difference of the best error between the real valued (y-axis) and rank-based (x-axis) classification method combined with a specific feature selection method over all remaining parameter combinations. A positive difference (orange/black) means that the method on the x-axis has lower empirical error than the method on the y-axis. A negative difference (green/white) means that the method on the y-axis has lower empirical error than the method on the x-axis.

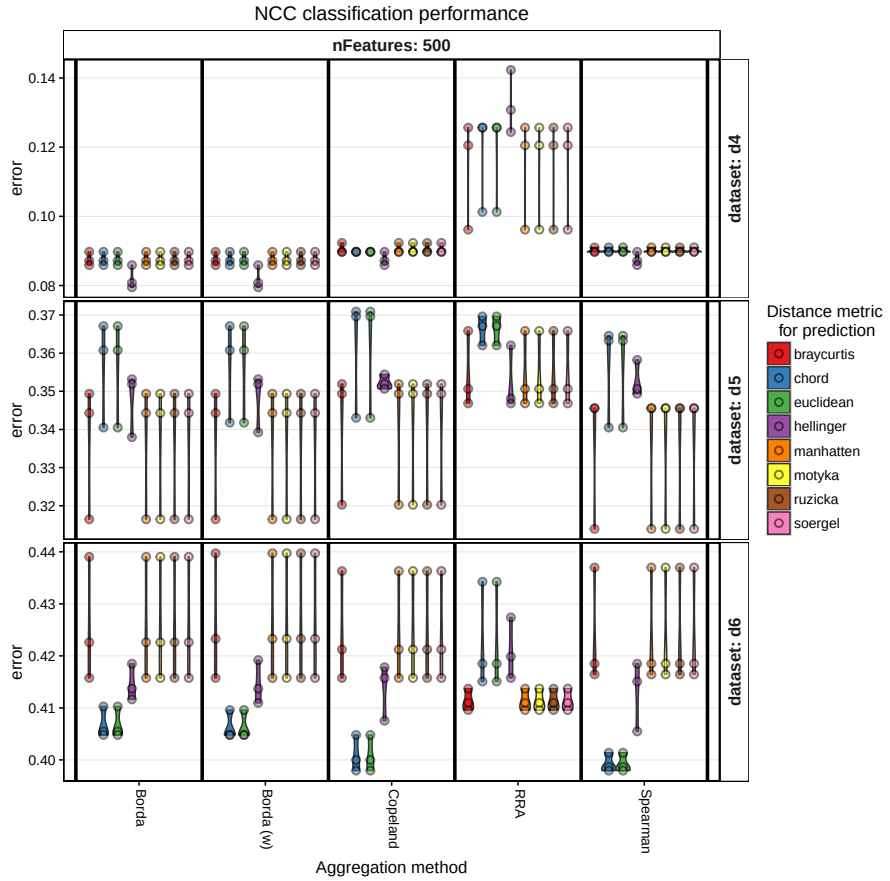

Fig. 3 Empirical error for the rank-based *NCC* classifier. Each row shows the summary for a single dataset (500 features selected). The error is shown on the y-axis and is scaled to its range. The *NCC* uses three parameters, namely the aggregation method (x-axis), the distance metric used to measure the distance between the consensus ranking (centroid) and the sample ranking (ordered violin plots), and a feature selection method (points in the violin plot on which the width of the violin plot is drawn).

In this regard, rank-transformation of the data values is a *robust* way to preserve the order of the values and the limit the effect size when analyzing it.

The performance of the real value and ranked-based classification methods depends on the dataset and the chosen parameters. Therefore no clear "winner" method can be named. However, the rank transformation of the data allows the application of additional distance metrics, such as rank aggregation methods for centroid calculation of the *NCC* classifier. Again, there is up to now no way

Table 6 Error rates for the NCC classifier on real valued datasets. The feature selection method that results in the best error rate for individual feature number threshold per dataset is shown in parenthesis.

NCC	Feature number threshold				
Dataset	5	50	100	250	500
d1	0.021 (cor)	0.004 (tnom)	0.011 (snr)	0.012 (tnom)	0.012 (tnom)
d2	0.085 (cor)	0.082 (snr)	0.1 (cor/snr/tnom)	0.1 (cor/snr/tnom)	0.1 (cor/snr/tnom)
d3	0.069 (snr)	0.069 (tnom)	0.069 (cor/tnom)	0.069 (tnom)	0.069 (tnom)
d4	0.09 (snr)	0.077 (snr)	0.077 (snr)	0.083 (snr)	0.074 (cor/snr)
d5	0.291 (snr)	0.356 (snr)	0.362 (cor)	0.353 (tnom)	0.346 (tnom)
d6	0.408 (tnom)	0.342 (cor)	0.375 (snr)	0.402 (snr)	0.407 (snr)
d7	0.105 (cor/snr)	0.1 (tnom)	0.122 (tnom)	0.122 (tnom)	0.127 (tnom)

Table 7 Error rates for the NPC and RPS classifier on real valued datasets. The feature selection method that results in the best error rate for individual feature number threshold per dataset is shown in parenthesis.

NPC	Feature number threshold					**RPS**	Feature number threshold				
Dataset	5	50	100	250	500		5	50	100	250	500
d1	0.031 (cor)	0.006 (tnom)	0.011 (tnom)	0.014 (tnom)	0.031 (tnom)		0.024 (cor)	0.015 (cor/tnom)	0.015 (snr/tnom)	0.014 (tnom)	0.014 (tnom)
d2	0.088 (cor)	0.095 (cor)	0.1 (cor/snr)	0.125 (cor)	0.148 (cor)		0.095 (snr)	0.08 (cor)	0.09 (snr)	0.1 (cor)	0.128 (cor)
d3	0.067 (snr)	0.066 (tnom)	0.069 (tnom)	0.069 (tnom)	0.078 (tnom)		0.071 (tnom)	0.067 (tnom)	0.064 (tnom)	0.057 (tnom)	0.062 (cor)
d4	0.094 (cor)	0.086 (snr)	0.087 (cor)	0.088 (cor/snr)	0.088 (cor)		0.103 (cor)	0.103 (cor)	0.106 (tnom)	0.095 (snr)	0.095 (snr)
d5	0.304 (snr)	0.371 (tnom)	0.357 (tnom)	0.357 (tnom)	0.343 (snr)		0.332 (snr)	0.353 (cor)	0.323 (snr)	0.352 (snr)	0.339 (tnom)
d6	0.393 (tnom)	0.369 (snr)	0.375 (tnom)	0.404 (cor)	0.41 (snr)		0.402 (cor)	0.316 (snr)	0.308 (snr)	0.347 (snr)	0.373 (snr)
d7	0.093 (cor/snr)	0.115 (tnom)	0.137 (tnom)	0.141 (tnom)	0.149 (tnom)		0.107 (cor/snr)	0.115 (tnom)	0.132 (tnom)	0.159 (tnom)	0.18 (tnom)

to tell which is the best aggregation method, but in our simulation study we can say, that the Robust Rank Aggregation method is not competitive when compared to the other aggregation methods.

We used feature selection methods in order to reduce the dimensionality of the data and also to better observe their effects on the rank-based classification methods. First, the number of features used plays an important role in rank aggregation methods. The Copeland and Spearman aggregation methods have

quadratic and cubic run-time complexity, respectively, which renders them less applicable when the number of features used in a dataset is high. Second, the number of features should not be too low, as can be seen in Fig. 2. Due to the rank-transformation, the number of different rankings is $d!$ where d is the number of selected features. Thus, when using only 5 features for classification, only 120 different prototypes or samples can be compared and therefore rank-based methods have a relatively high empirical error. Furthermore, the relatively accurate performance of the k-NN classifier is also due to the comparatively low number of features used for classification.

Simulation results also reveal, that for some choice of distance function the classification performance does not change. This is due to the fact that several distance functions produce the same neighborhood topology and the prediction of the label for a sample point only depends on the topology and not the scale of the distance value itself. Thus, distance functions can be divided in distance "classes" thus reducing the parameter space by allowing to use an exemplary distance function from a distance class.

In conclusion, rank-based classification methods allow the usage of additional parameters which can be determined in the parameter tuning phase. Additionally, they allow the computation of centroids in a new way. In our experiments we applied the prototype-based classification methods – ranked and real value-based variants – on microarray datasets and compared their performance. In some cases, the rank-based methods outperformed the real value-based classifiers in terms of empirical error rate. Our experimental results suggest that the application of rank-based prototype classifiers should be included when dealing with microarray datasets.

Acknowledgements The research leading to these results has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 602783 (to HAK), the German Research Foundation (DFG, SFB 1074 project Z1 to HAK), and the Federal Ministry of Education and Research (BMBF, Gerontosys II, Forschungskern SySta R, project ID 0315894A and e:Med, SYMBOL-HF, ID 01ZX1407A to HAK).

References

Alter M, Kharkar R, Ramsey K, Craig D, Melmed R, Grebe T, Bay RC, Ober-Reynolds S, Kirwan J, Jones JJ, Turner JB, Hen R, Stephan D (2011) Autism and increased paternal age related changes in global levels of gene expression regulation. PLoS ONE 6(2):e16,715, DOI 10.1371/journal.pone.0016715

Armstrong S, Staunton J, Silverman L, Pieters R, den Boer M, Minden M, Sallan S, Lander E, Golub T, Korsmeyer S (2002) MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genetics 30(1):41–47, DOI 10.1038/ng765

Badea L, Herlea V, Dima S, Dumitrascu T, Popescu I (2008) Combined gene expression analysis of whole-tissue and microdissected pancreatic ductal adenocarcinoma identifies genes specifically overexpressed in tumor epithelia. Hepatogastroenterology 55(88):2016–2027

Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z (2000) Tissue classification with gene expression profiles. Journal of Computational Biology 7:559–583, DOI 10.1089/106652700750050943

Biehl M (2014) Prototype-Based Classifiers and Their Application in the Life Sciences, vol 295, Springer International Publishing, chap Advances in Self-Organizing Maps and Learning Vector Quantization, pp 121–121

Burkovski A, Lausser L, Kraus JM, Kestler HA (2014) Rank aggregation for candidate gene identification. In: Spiliopoulou M, Schmidt-Thieme L, Janning R (eds) Data Analysis, Machine Learning and Knowledge Discovery, Springer, pp 285–293

Cha SH (2007) Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences 4:300–307

Cover TM (1965) Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers EC-14:326–334, DOI 10.1109/PGEC.1965.264137

Cuperlovic-Culf M, Belacel N, Ouellette RJ (2005) Determination of tumour marker genes from gene expression data. Drug Discovery Today 10:429–437, DOI 10.1016/S1359-6446(05)03393-3

Dwork C, Kumar R, Naor M, Sivakumar D (2001) Rank aggregation revisited. Systems Research 13:86–93

Dyrskjøt L, Thykjaer T, Kruhøffer M, Jensen J, Marcussen N, Hamilton-Dutoit S, Wolf H, Orntoft T (2003) Identifying distinct classes of bladder carcinoma

using microarrays. Nature Genetics 33(1):90–96, DOI 10.1038/ng1061

Fix E, Hodges JL (1951) Discriminatory analysis: Nonparametric discrimination: Consistency properties. Tech. rep., USAF School of Aviation Medicine, Randolf Field, Texas

Kendall M (1938) A new measure of rank correlation. Biometrika 30:81–89, DOI 10.2307/2332226

Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial Intelligence 97:273–324, DOI 10.1016/S0004-3702(97)00043-X

Kraus J, Lausser L, Kestler HA (2015) Exhaustive k-nearest-neighbour subspace clustering. Journal of Statistical Computation and Simulation 85:30–46, DOI 10.1080/00949655.2014.933222

Kuner R, Muley T, Meister M, Ruschhaupt M, Buness A, Xu E, Schnabel P, Warth A, Poustka A, Sültmann H, Hoffmann H (2009) Global gene expression analysis reveals specific patterns of cell junctions in non-small cell lung cancer subtypes. Lung Cancer 63(1):32–38, DOI 10.1016/j.lungcan.2008.03.033

Lausser L, Müssel C, Kestler HA (2012) Representative prototype sets for data characterization and classification. In: Mana N, Schwenker F, Trentin E (eds) Artificial Neural Networks in Pattern Recognition, Lecture Notes in Computer Science, vol 7477, Springer Berlin Heidelberg, pp 36–47

Lausser L, Müssel C, Maucher M, Kestler HA (2013) Measuring and visualizing the stability of biomarker selection techniques. Computational Statistics 28:51–65, DOI 10.1007/s00180-011-0284-y

Lausser L, Müssel C, Melkozerov A, Kestler HA (2014) Identifying predictive hubs to condense the training set of k-nearest neighbour classifiers. Computational Statistics 29:81–95, DOI 10.1007/s00180-012-0379-0

Lausser L, Schmid F, Schirra LR, Wilhelm A, Kestler H (2016) Rank-based classifiers for extremely high-dimensional gene expression data. Advances in Data Analysis and Classification pp 1–20, DOI 10.1007/s11634-016-0277-3

Little S, Colatino S, Salvetti O, Perner P (2009) Can prototype-based classification be a good method for biomedical applications? Transactions on Machine Learning and Data Mining 2:44–61

Lu T, Aron L, Zullo J, Pan Y, Kim H, Chen Y, Yang TH, Kim HM, Drake D, Liu X, Bennett D, Colaiacovo M, Yankner B (2014) REST and stress resistance in ageing and alzheimer's disease. Nature 507(7493):448–454, DOI 10.1038/nature13163

Müssel C, Lausser L, Maucher M, Kestler HA (2012) Multi-objective parameter selection for classifiers. Journal of Statistical Software 46:1378–1380, DOI 10.18637/jss.v046.i05

Pierre A Devijver JK (1982) Pattern Recognition: A Statistical Approach. Prentice Hall

Schalekamp F, Zuylen A (2009) Rank aggregation: Together we're strong. In: Proceedings of the 11th Workshop on Algorithm Engineering and Experiments, SIAM, pp 38–51

Schirra LR, Lausser L, Kestler HA (2016) Selection stability as a means of biomarker discovery in classification. In: Wilhelm AFX, Kestler HA (eds) Analysis of Large and Complex Data, Springer, pp 79–89

Sun Y, Goodison S (2009) Optimizing molecular signatures for predicting prostate cancer recurrence. The Prostate 69(10):1119–1127, DOI 10.1002/pros.20961

Yeang CH, Ramaswamy S, Tamayo P, Mukherjee S, Rifkin RM, Angelo M, Reich M, Lander E, Mesirov J, Golub T (2001) Molecular classification of multiple tumor types. Bioinformatics 17:S316–S322, DOI 10.1093/bioinformatics/17.suppl_1.S316

Ordinal prototype-based classifiers

Supplementary Information

Andre Burkovski[2,3,5], Lyn-Rouven Schirra[2,4], Florian Schmid[2], Ludwig Lausser[1], Hans A. Kestler[1,2,*]

1 Leibniz Institute on Aging, Fritz-Lipmann Institute, 07745 Jena, Germany
2 Medical Systems Biology, Ulm University, 89069 Ulm, Germany
3 Institute for Neural Information Processing, Ulm University, 89069 Ulm, Germany
4 Institute of Number Theory and Probability Theory, Ulm University, 89069 Ulm, Germany
5 International Graduate School in Molecular Medicine, Ulm University, 89069 Ulm, Germany
∗ corresponding authors email: hans.kestler@fli-leibniz.de

Comparison of classification methods

Figures 4–7 show the error distribution based on the parameter combination. The plots are grouped by the number of features selected via a feature selection method (columns) and dataset (rows). The y-axis shows the error and the x-axis shows the according ranked-based classification method. The red line shows the *best error* of real value-based classifier (achievable by selection of the best suiting parameter). The violin plots indicate the density of parameters that are associated with the classification error.

Similarly, Figs. 8–11 show the error distribution based on the parameter combination for all rank-based classifier. The plots are grouped by the number of features selected via a feature selection method (columns) and dataset (rows). Thy y-axis shows the error and the x-axis shows the according all ranked-based classification methods described in the main text. The red line shows the *best error* of real value-based classifier (achievable by selection of the best suiting parameter). The violin plots indicate the density of parameters that are associated with the classification error.

Figures 12–15 show the classification error for different parameters of the NCC classifier. Here, the classification error for all datasets and 500 of selected features is shown.

Figure 16 shows the actual error difference between rank-based and real value-based. The plots are grouped by the number of features selected via a feature selection method (columns) and dataset (rows). Thy y-axis denotes the real value-based classification methods and the x-axis shows the according ranked-based in combination with a specific feature selection method, respectively. Each tile corresponds the error difference between the best errors for the respective classifiers. Positive difference mean that the real value-based classifier has a lower error than the rank-based classifier. Negative difference mean that the rank-based classifier has a lower error than the real value-based classifiers.

Figure 17 shows a summary of all parameter combination comparisons per feature number (columns) and dataset (rows). The bar plot shows the number of parameter combinations divided in three categories. “Positive” category represents the number of parameter combinations where the real value-based classifiers have a lower error than the rank-based classifiers. Similarly, “negative” category represents the number of parameter combinations where the rank-based classifiers have a lower error than the real value-based classifiers.

The “zero” category represents the number of parameter combinations where real value and ranked-based classifiers have the same error.

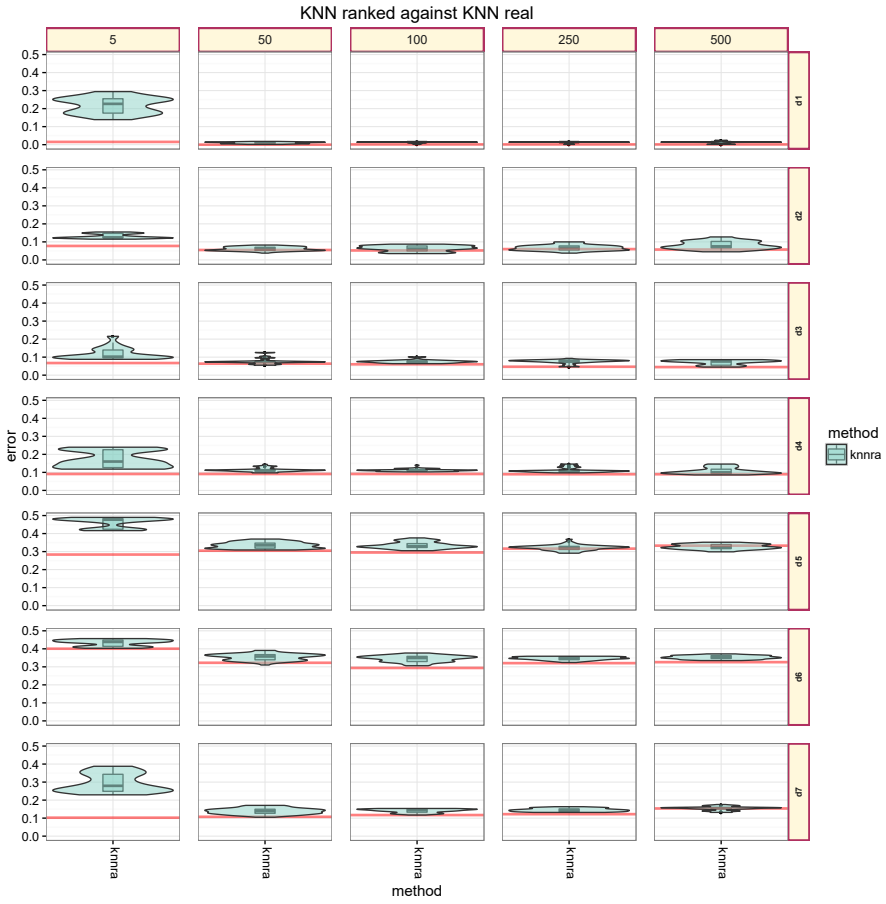

Fig. 4 Error distribution of the rank-based k-NN classifier with regard to the best real value-based counterpart.

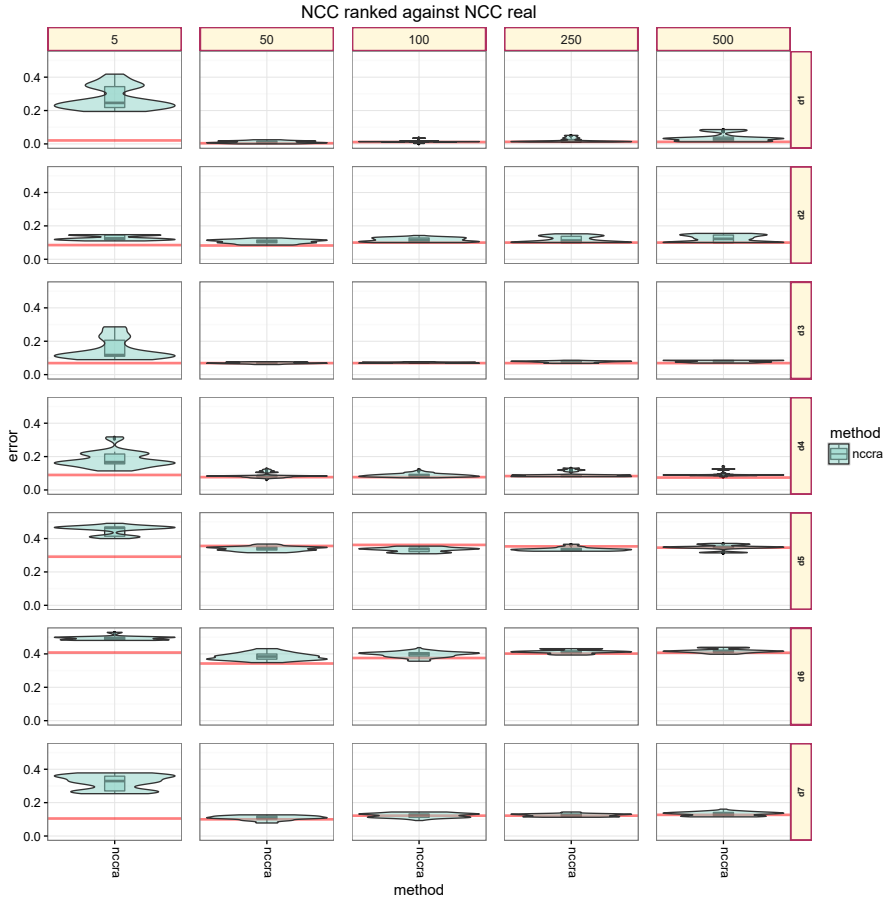

Fig. 5 Error distribution of the rank-based NCC classifier with regard to the best real value-based counterpart.

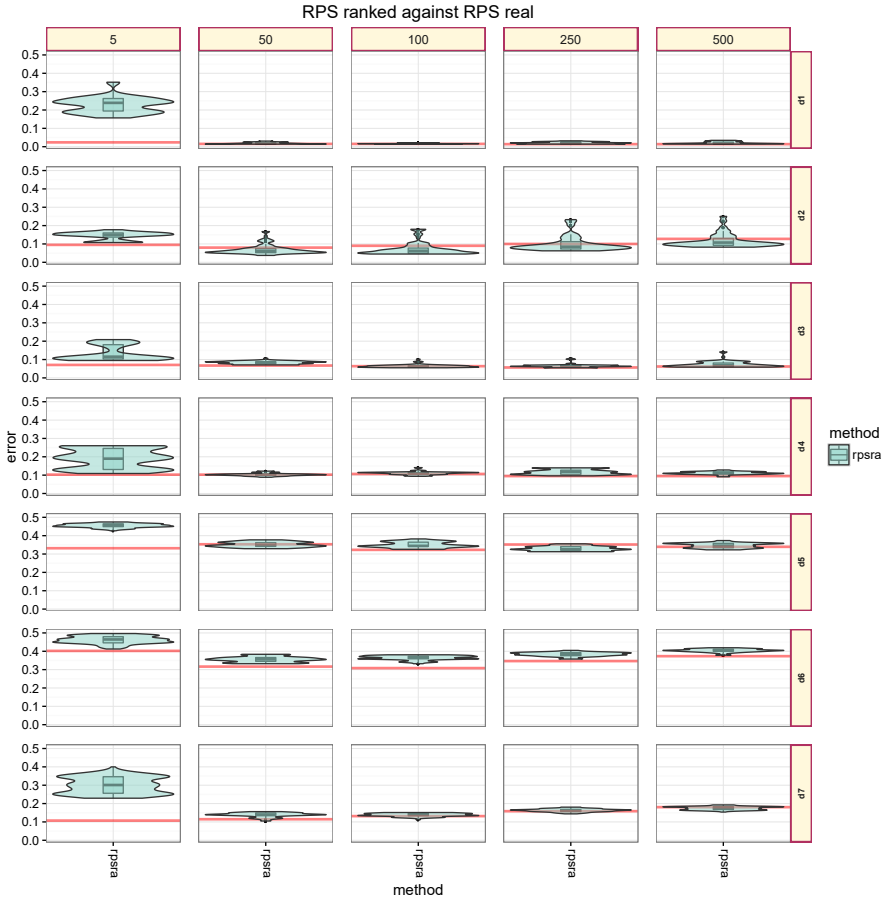

Fig. 6 Error distribution of the rank-based RPS classifier with regard to the best real value-based counterpart.

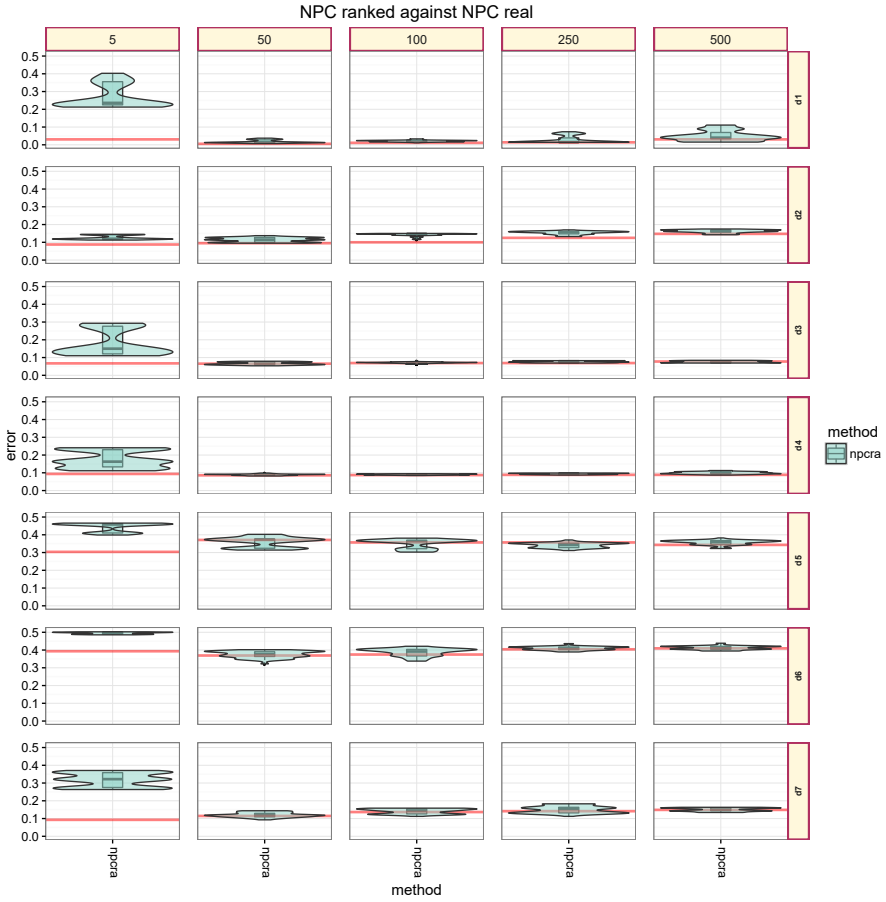

Fig. 7 Error distribution of the rank-based NPC classifier with regard to the best real value-based counterpart.

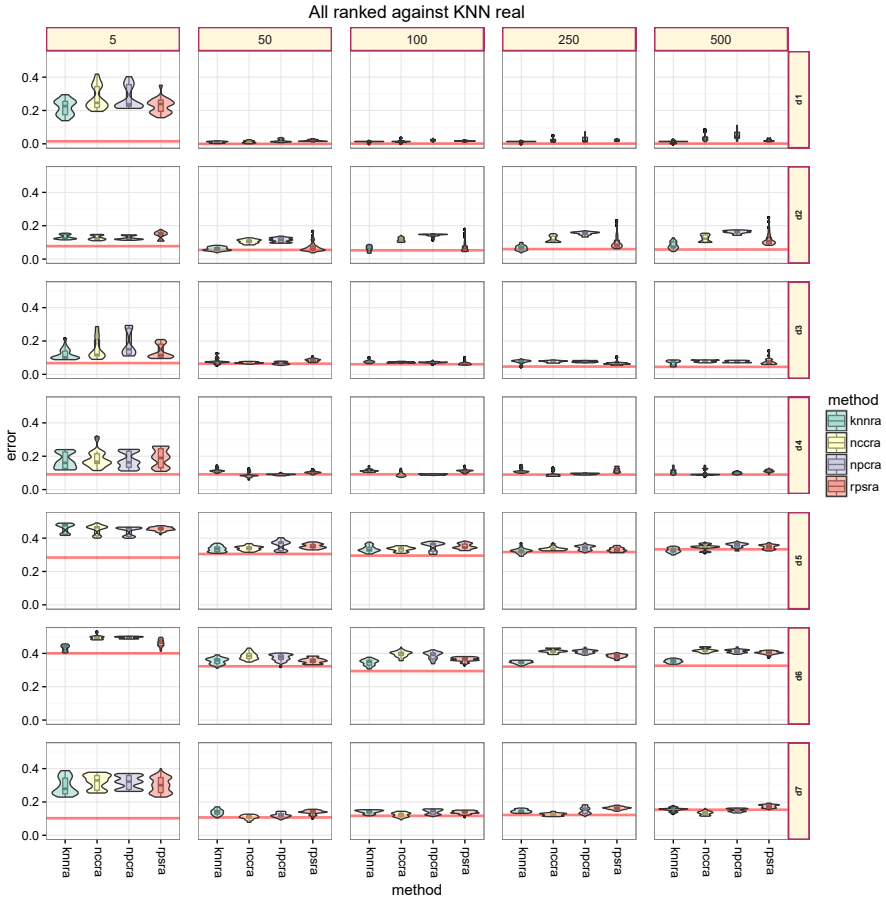

Fig. 8 Error distribution of all rank-based classifier with regard to the best real value-based k-NN classifier.

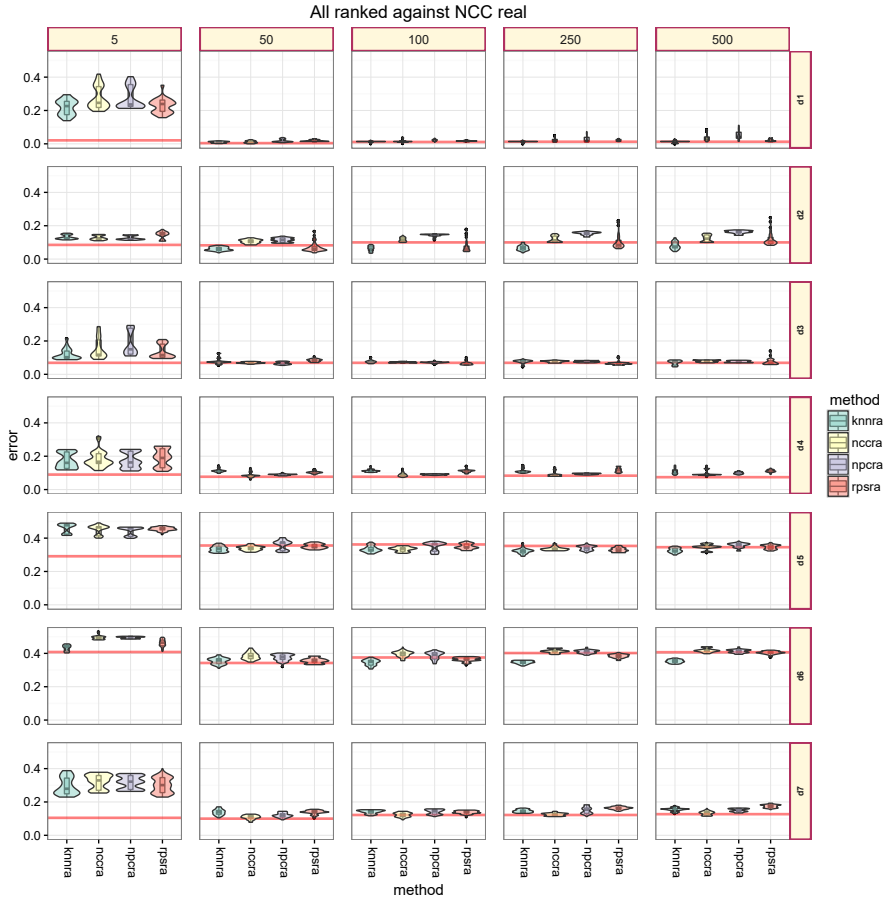

Fig. 9 Error distribution of all rank-based classifier with regard to the best real value-based NCC classifier.

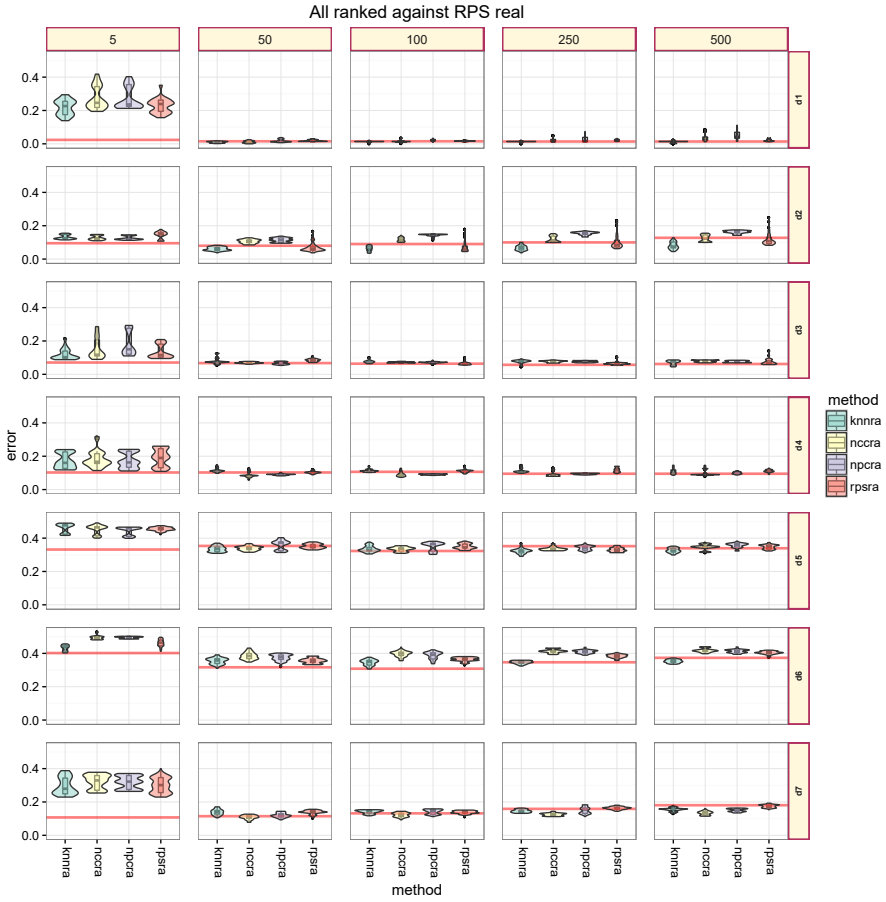

Fig. 10 Error distribution of all rank-based classifier with regard to the best real value-based RPS classifier.

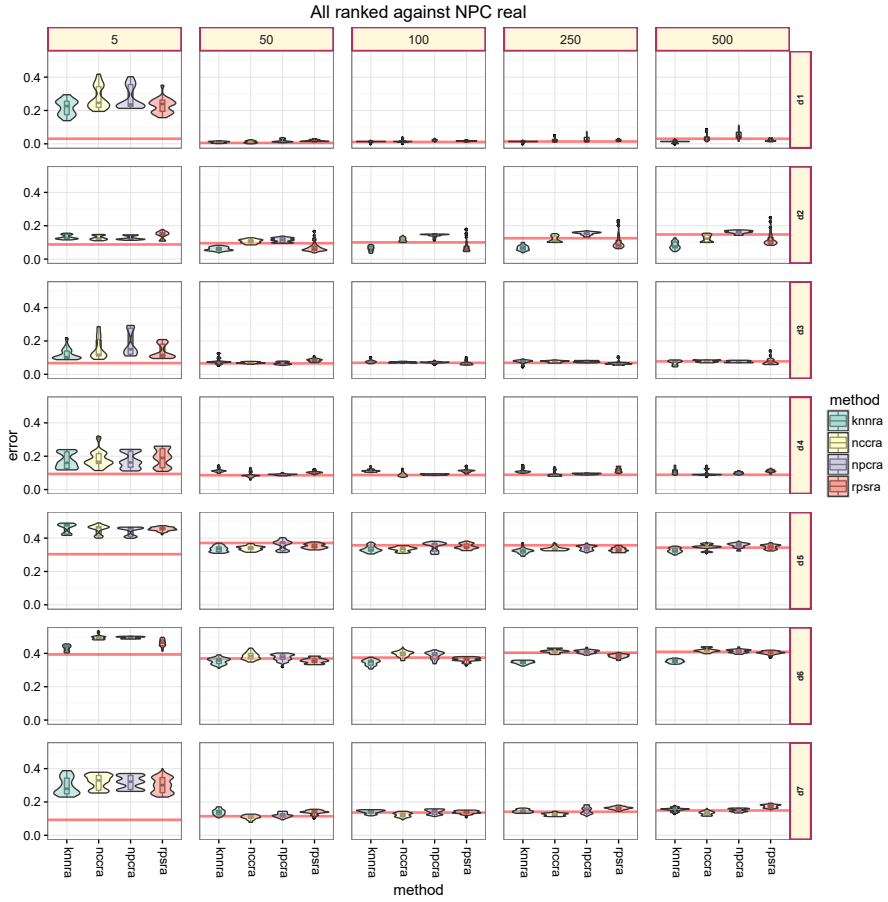

Fig. 11 Error distribution of all rank-based classifier with regard to the best real value-based NPC classifier.

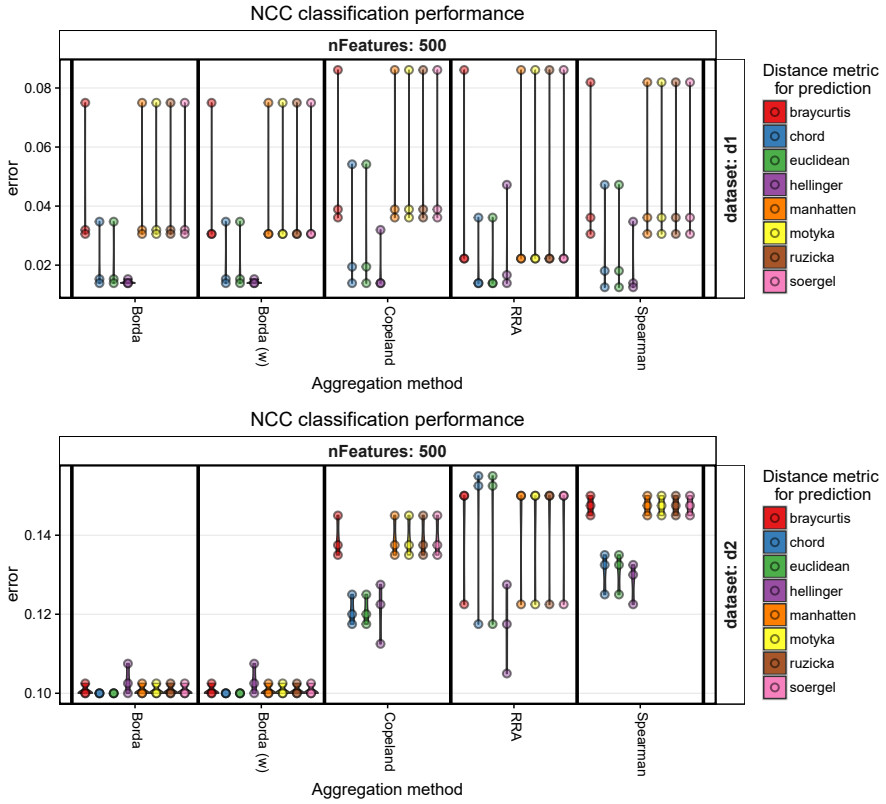

Fig. 12 Influence of different parameters to the classification error of the NCC classifier.

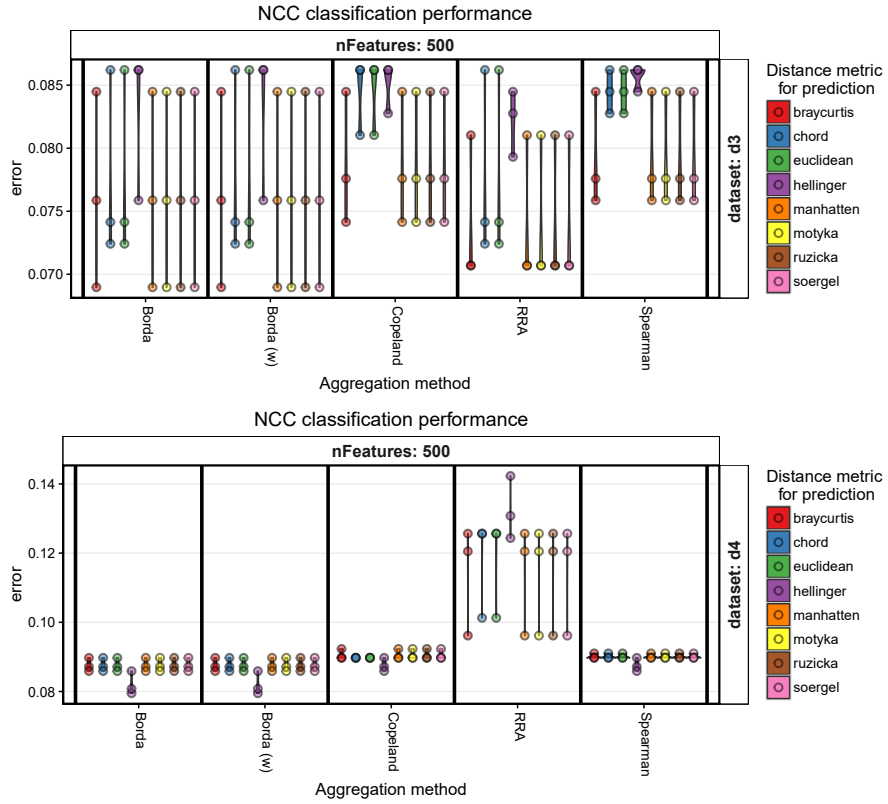

Fig. 13 Influence of different parameters to the classification error of the NCC classifier.

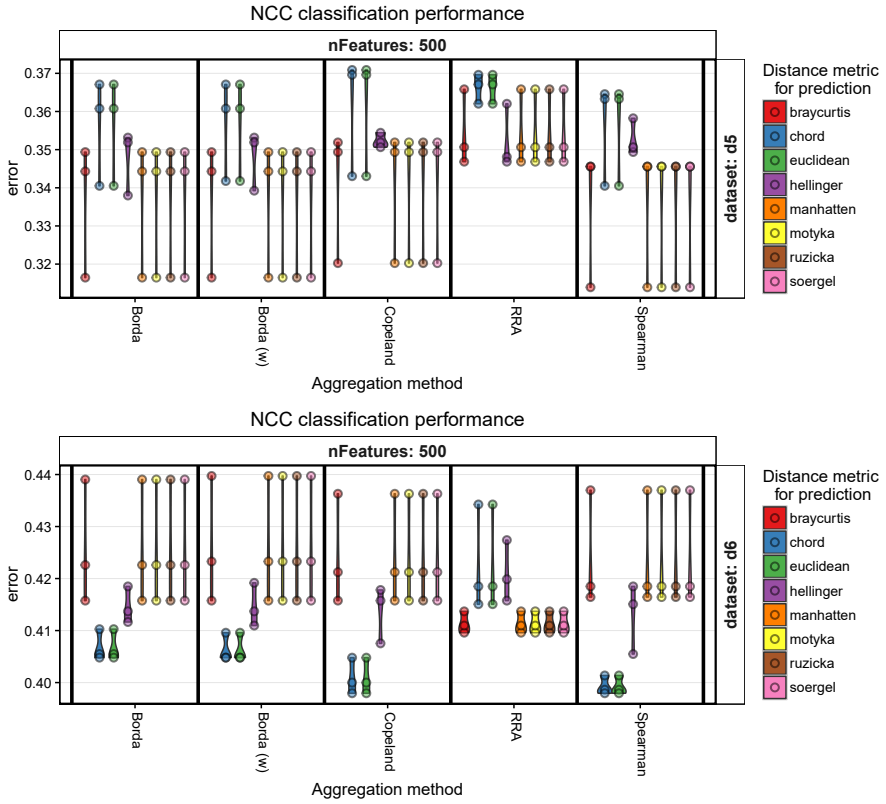

Fig. 14 Influence of different parameters to the classification error of the NCC classifier.

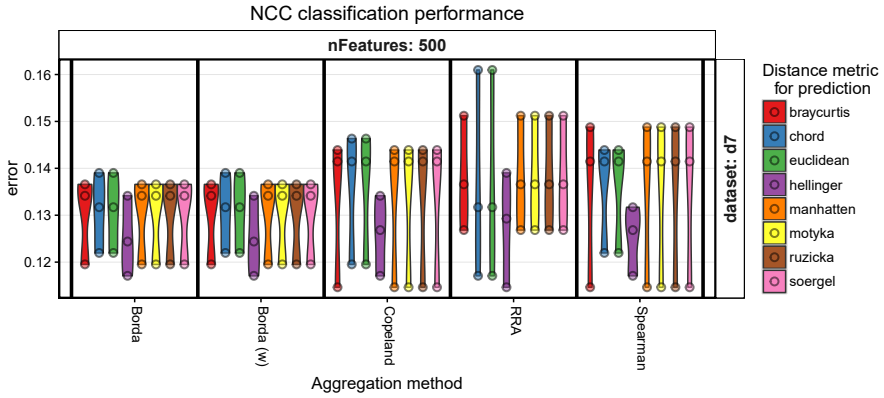

Fig. 15 Influence of different parameters to the classification error of the NCC classifier.

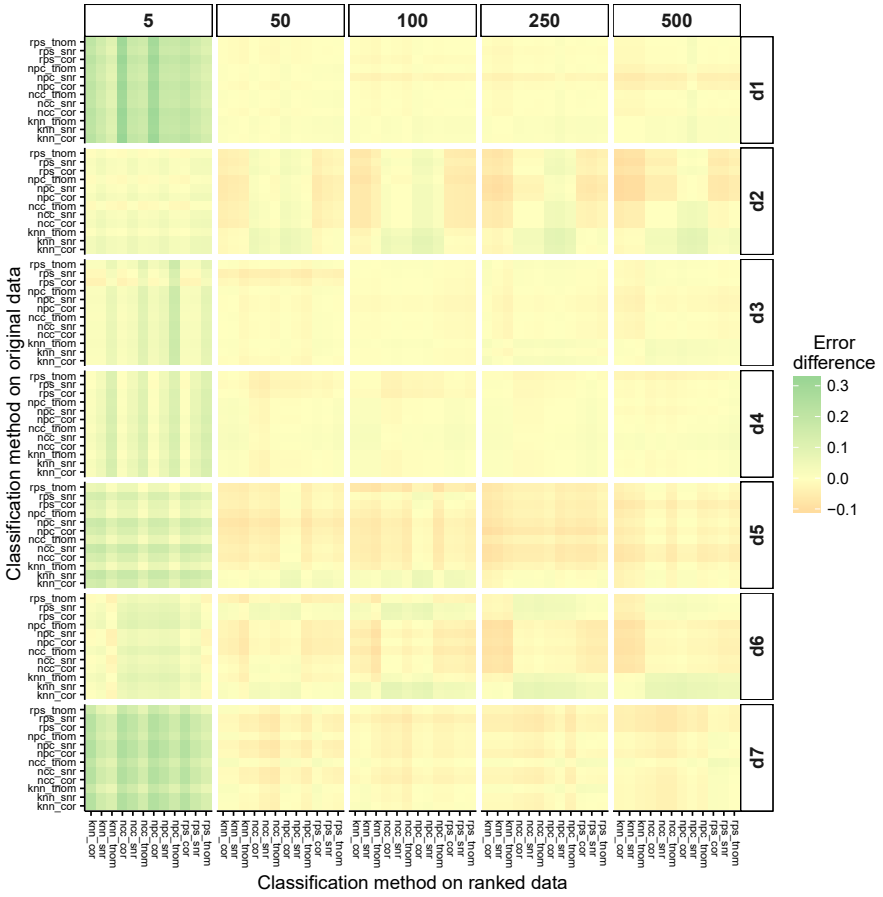

Fig. 16 Error difference between the rank-based and real value-based classifiers.

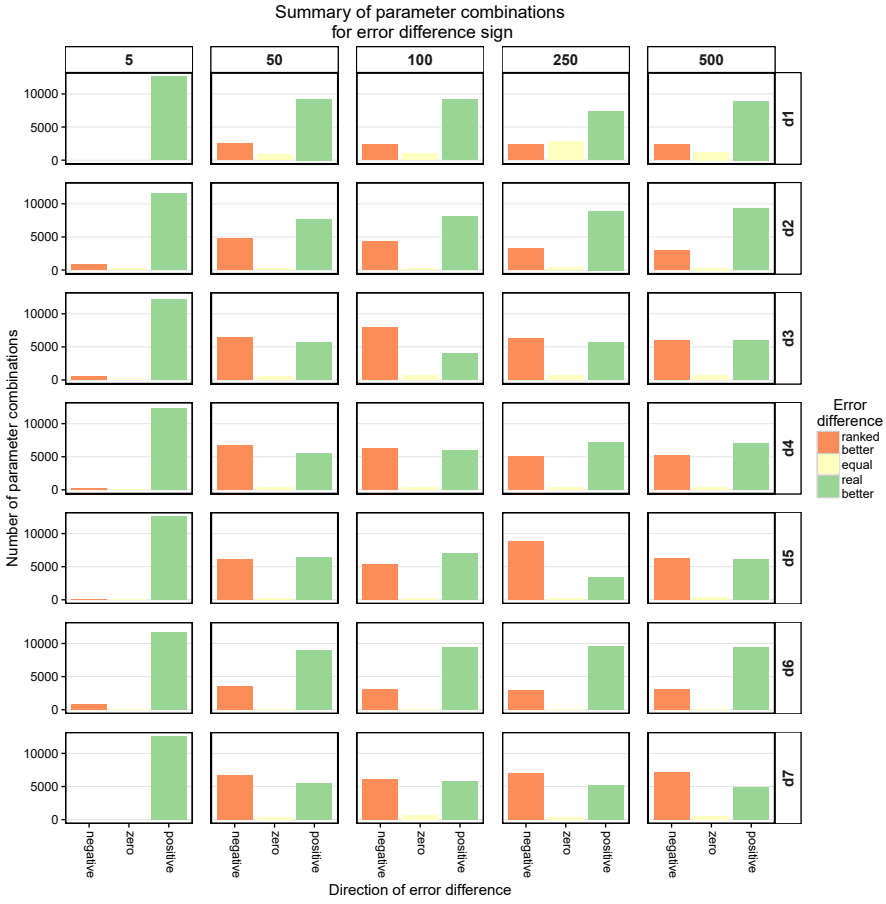

Fig. 17 Summary of all parameter combination comparisons.