Adv Data Anal Classif (2018) 12:917–936
https://doi.org/10.1007/s11634-016-0277-3

REGULAR ARTICLE

Rank-based classifiers for extremely high-dimensional gene expression data

Ludwig Lausser[1] · Florian Schmid[1] · Lyn-Rouven Schirra[1,3] · Adalbert F. X. Wilhelm[4] · Hans A. Kestler[1,2]

Received: 15 December 2014 / Revised: 21 November 2016 / Accepted: 28 November 2016 /
Published online: 19 December 2016

Abstract Predicting phenotypes on the basis of gene expression profiles is a classification task that is becoming increasingly important in the field of precision medicine. Although these expression signals are real-valued, it is questionable if they can be analyzed on an interval scale. As with many biological signals their influence on e.g. protein levels is usually non-linear and thus can be misinterpreted. In this article we study gene expression profiles with up to 54,000 dimensions. We analyze these measurements on an ordinal scale by replacing the real-valued profiles by their ranks. This type of rank transformation can be used for the construction of invariant classifiers that are not affected by noise induced by data transformations which can occur in the measurement setup. Our 10×10 fold cross-validation experiments on 86 different data sets and 19 different classification models indicate that classifiers largely benefit from this transformation. Especially random forests and support vector machines achieve improved classification results on a significant majority of datasets.

Keywords Rank-based classification · Invariance · High-dimensional data · Gene expression data

L. Lausser, F. Schmid, and L.-R. Schirra contributed equally.

Electronic supplementary material The online version of this article (doi:10.1007/s11634-016-0277-3) contains supplementary material, which is available to authorized users.

✉ Hans A. Kestler
hans.kestler@uni-ulm.de; hans.kestler@leibniz-fli.de

1 Institute of Medical Systems Biology, Ulm University, 89069 Ulm, Germany

2 Leibniz Institute on Aging–Fritz Lipmann Institute, 07745 Jena, Germany

3 Institute of Number Theory and Probability Theory, Ulm University, 89069 Ulm, Germany

4 Department of Psychology and Methods, Jacobs University, 28759 Bremen, Germany

Mathematics Subject Classification 62H30 Classification and discrimination; cluster analysis · 68T10 Pattern recognition, speech recognition · 92C40 Biochemistry, molecular biology

1 Introduction

Utilizing high-throughput technologies for measuring gene expression levels has become a widely used approach in life sciences. Generating profiles of tens of thousands of simultaneous measurements, these techniques allow to characterize a biological object (e.g. a tissue) by a more or less detailed snapshot of its transcriptomic activities (see Fig. 1). From a machine learning point of view, the analysis of gene expression profiles is a typical example of learning from high-dimensional datasets that only comprise small numbers of objects. Nevertheless dataset collections exist that allow a large-scale evaluation of new approaches, see Table 1 for such a collection.

In this work we analyze the effects of rank transformation on classifiers that were originally designed for the analysis of real-valued data. Unlike in other approaches that reduce the information content mainly via feature selection (Breiman et al. 1984; Breiman 2001; Ben-Dor et al. 2000; Thomas et al. 2001; Guyon and Elisseeff 2003; Saeys et al. 2007), we apply a rank transformation to the individual feature profiles. The rank-based counterparts of these classifiers thus only receive the ordinal structure of the gene expression profiles and cannot access the absolute expression levels. Although these algorithms receive a reduced amount of information, the rank-based algorithms gain beneficial properties. As a consequence of this transformation the classifier gains an invariance with respect to feature-wise strictly monotonically increasing data transformations. The predictions of these classifier cannot be affected by misleading data transformations of this type. In line with other investigations such as from Jamain and Hand (2009) we test the performance of the modified classifiers on a large collection of 86 microarray datasets.

Rank transformation is not the only technique for inducing invariances and classification not the only problem/methodological field (Bavaud 2009). An overview

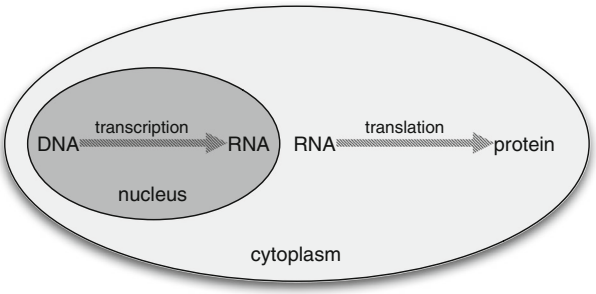

Fig. 1 Principle of information processing in the cell. Genomic information on the DNA is transcribed into RNA which is translated into proteins. DNA and RNA mainly consist of four different chemical compounds (bases). Proteins are chemically more complex and consist of amino acids. Due to the more uniform chemical structure, large DNA and RNA screens, the latter being called gene expression profiles, are simpler to measure in tissue or blood than different proteins. See Table 1 for a list of utilized gene expression profiles measured with microarrays

Table 1 List of utilized datasets

	Authors	GEOid	Journal	No. of Features	Classes 0/1
d1	Alcalay et al. (2005)	GSE34860	Blood	22215	21/57
d2	Alhopuro et al. (2012)	GSE24514	Int J Cancer	22215	15/34
d3	Alter et al. (2011)	GSE25507	PLoS One	54613	64/82
d4	Armstrong et al. (2002)		Nat Genet	12559	24/48
d5	Badea et al. (2008)	GSE15471	Hepatology Gastroenterol	54613	39/39
d6	Boersma et al. (2008)	GSE5847	Int J Cancer	22215	69/26
d7	Bowen et al. (2009)	GSE14407	BMC Med Genomics	54613	12/12
d8	Creighton et al. (2009)	GSE7515	Proc Natl Acad Sci	54613	15/11
d9	D'Errico et al. (2009)	GSE13911	Eur J Cancer	54613	31/38
d10	Desmedt et al. (2007)	GSE7390	Clin Cancer Res	22215	154/36
d11	de Sousa E Melo et al. (2011)	GSE33113	Cell Stem Cell	54613	71/18
d12	Dom et al. (2012)	GSE33630	Br J Cancer	54613	45/60
d13	Dyrskjoet et al. (2003)		Nat Genet	7071	20/20
d14	Fountzilas et al. (2013)	GSE27020	PLoS One	22215	75/34
d15	Gadd et al. (2012)	GSE31403	Neoplasia	22215	163/61
d16	Hou et al. (2010)	GSE19188	PLoS One	54613	65/91
d17	Hummel et al. (2006)	GSE4475	New Engl J Med	22215	129/48
d18					48/44
d19					129/44
d20	Ivshina et al. (2006)	GSE4922	Cancer Res	44792	103/30
d21	Iwamoto et al. (2004)	GSE12654	Mol Psychiatr	12558	15/11
d22					15/11
d23					15/13
d24	Jelinsky et al. (2011)	GSE26051	BMC Musculoskelet Disord	54613	23/23
d25	Jones et al. (2005)	GSE15641	Clin Cancer Res	22215	23/69
d26	Koth et al. (2011)	GSE19314	Am J Respir Crit Care Med	54613	20/38
d27	Kuner et al. (2009)	GSE10245	Lung Cancer	54613	40/18
d28	LaBreche et al. (2011)	GSE27567	BMC Med Genomics	45037	28/65
d29	Landi et al. (2008)	GSE10072	PLoS One	22215	49/58
d30	Liang et al (2008)	GSE5281	Proc Natl Acad Sci	54613	74/87
d31	Li et al. (2011)	GSE23025	Cancer Cell	54613	78/46
d32	Loi et al. (2010)	GSE6532	Proc Natl Acad Sci	44792	91/21
d33	Lu et al. (2010)	GSE19804	Cancer Epidem Biomar	54613	60/60
d34	Lu et al. (2014)	GSE53890	Nature	54613	20/21

Table 1 continued

	Authors	GEOid	Journal	No. of Features	Classes 0/1
d35	Maenhaut et al. (2011)	GSE29265	Clin Oncol	54613	20/29
d36	Maycox et al. (2009)	GSE17612	Mol Psychiatr	54613	23/28
d37	Meyer et al. (2011)	GSE13576	Cancer Cell	54613	157/40
d38	Miesner et al. (2010)	GSE21261	Blood	54613	24/16
d39					56/24
d40					16/56
d41	Miller et al. (2005)	GSE3494	Proc Natl Acad Sci	44792	176/60
d42	Nair et al. (2009)	GSE13355	Nat Genet	54613	64/58
d43					64/58
d44					58/58
d45	Noordhuis et al. (2011)	GSE26511	Clin Cancer Res	54613	20/19
d46	Oricchio et al. (2014)	GSE37088	J Exp Med	22215	14/23
d47					14/35
d48					23/35
d49	Orsmark-Pietras et al. (2013)	GSE27011	Eur Respir J	32321	18/19
d50					18/17
d51					19/17
d52	Pawitan et al. (2005)	GSE1456	Br Cancer Res	44792	130/22
d53	Pei et al. (2009)	GSE16515	Cancer Cell	54613	16/36
d54	Peña-Llopis et al. (2012)	GSE36895	Nat Genet	54613	29/23
d55	Pomeroy et al. (2002)		Nature	7071	25/9
d56	Rowe et al. (2007)	GSE5666	J Neurosci	15866	29/49
d57	Salaverria et al. (2011)	GSE22470	Blood	22215	228/43
d58	Sanchez-Palencia et al. (2011)	GSE18842	Int J Cancer	54613	45/46
d59	Scherzer et al. (2007)	GSE6613	Proc Natl Acad Sci	22215	72/50
d60					22/33
d61					33/50
d62	Schmidt et al. (2008)	GSE11121	Cancer Res	22215	154/28
d63	Shipp et al. (2002)		Nat Med	7071	58/19
d64	Singh et al. (2002)		Cancer Cell	12558	50/52
d65	Singh et al. (2011)	GSE22148	Thorax	54613	71/72
d66	Stanke et al. (2013)	GSE15568	Eur J Hum Genet	22215	13/16
d67	Stirewalt et al. (2008)	GSE1159	Gene Chromosome Cancer	22215	116/177
d68	Stirewalt et al. (2009)	GSE13496	Leukemia	22215	12/18
d69	Sun and Goodison (2009)	GSE25136	Prostate	22215	40/39
d70	Suresh et al. (2014)	GSE48060	J Mol Cell Cardiol	54613	21/31
d71	Vathipadiekal et al. (2012)	GSE33874	PLoS One	54613	10/10
d72	Wang et al. (2005)	GSE2034	Lancet	22215	183/93
d73	Wang et al. (2012)	GSE19826	Med Oncol	54613	15/12
d74	Wang et al. (2013)	GSE29272	PLoS One	22215	134/134

Table 1 continued

	Authors	GEOid	Journal	No. of Features	Classes 0/1
d75					62/72
d76					134/62
d77					134/72
d78	Wong et al. (2010)	GSE12413	Crit Care Med	45037	41/43
d79	Wu et al. (2013)	GSE35809	Gut	54613	15/26
d80					29/26
d81					29/15
d82	Xu et al. (2008)	GSE8401	Mol Cancer Res	22215	31/52
d83	Yeoh et al. (2002)		Cancer Cell	12558	205/43
d84	Zhang et al. (2009)	GSE12093	Br Cancer Res Tr	22215	116/20
d85	Zhang et al. (2010)	GSE14245	Gastroenterology	54613	12/12
d86	Zhang et al. (2013)	GSE28735	Clin Cancer Res	32321	45/45

References are given in the supplementary information

for inducing invariances can be found in Haasdonk and Burkhardt (2007). Certain classes of concepts and functions only comprise invariant classifiers (Schmid et al. 2014). Invariances or tolerances against data transformations can also be enforced by modifying existing training procedures (Wood 1996). They can be adapted by extending the training sets by virtual examples (Schölkopf et al. 1996; Niyogi et al. 1998). Additionally, modified cost functions for optimizing invariant classifiers were for example proposed for neural networks (Simard et al. 2012) and support vector machines (Schölkopf et al. 1998; Tsuda 1999).

2 Methods

In the context of classifying gene expression data, an object (e.g. cell or tissue) is represented by a high-dimensional profile of n measurements, a real-valued vector $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T \in \mathcal{X} \subseteq \mathbb{R}^n$. $\mathcal{X}$ is called the feature space of the classification task. We will additionally utilize the notation $\hat{\mathbf{x}} = \{x^{(i)}\}_{i\in[n]}$ to denote the unsorted set of measurements in $\mathbf{x}$, where $[n] = \{1, \ldots, n\}$. The class label of interest $y \in \mathcal{Y}$ arises from a biomedical context. We restrict ourselves to binary classification tasks $|\mathcal{Y}| = 2$ in the following. The prediction of the class label is modeled as a function $c : \mathcal{X} \longrightarrow \mathcal{Y}$, a so-called classifier.

For a particular classification task, a suitable classifier has to be selected in an initial learning phase l. Here, the classifier is adapted to a known training set of m labeled feature profiles $\mathcal{T} = \{(\mathbf{x}_j, y_j)\}_{j\in[m]}$. It is typically selected from a concept class $\mathcal{C}$, a class of functions or concepts, that characterizes the structural (data-independent) properties of a model type

$$l : \mathcal{T} \times \mathcal{C} \longrightarrow c_{\mathcal{T}}. \tag{1}$$

We will drop the subscript $\mathcal{T}$, if the training set is clear from the context or irrelevant.

The most important characteristic of a trained classifier is its performance in predicting the class labels of new unseen feature profiles. In the case of unknown class distributions, it can be estimated on a finite validation set of m' labeled feature profiles $\mathcal{V} = \{(\mathbf{x}'_j, y'_j)\}_{j\in[m']}$. Typical measures for the performance of a classifier are its empirical error rate

$$R_{emp}(c, \mathcal{V}) = \frac{1}{|\mathcal{V}|} \sum_{(\mathbf{x},y)\in\mathcal{V}} \mathbb{I}_{[c(\mathbf{x})\neq y]}. \tag{2}$$

or the empirical accuracy $A_{emp}(c, \mathcal{V}) = 1 - R_{emp}(c, \mathcal{V})$.

2.1 Invariance properties

The invariances of classifiers or concept classes define the conditions under which the classifiers' predictions are guaranteed to be unaffected by a certain kind of data transformation. Of special interest are structural properties that can be used a priori for the design of a training algorithm. Slightly different definitions of invariance exist (Haasdonk and Burkhardt 2007). In the following, we will use the definition from Schmid et al. (2014):

Definition 1 A classifier $c : \mathcal{X} \to \mathcal{Y}$ is called *invariant* with respect to a parameterized class of data transformations $\mathcal{F} = \{f_\theta \mid \theta \in \Theta\}$ if

$$\forall\, \theta \in \Theta, \forall\, \mathbf{x} \in \mathcal{X} : \quad c(f_\theta(\mathbf{x})) = c(\mathbf{x}). \tag{3}$$

Here Θ denotes the set of parameters. A concept class $\mathcal{C}$ is called invariant with respect to $\mathcal{F}$ if each $c \in \mathcal{C}$ is invariant with respect to $\mathcal{F}$.

Definition 1 states that the predictions of a chosen classifier have to be unaffected by the data transformations for all possible input values $\mathbf{x} \in \mathcal{X}$. It does not cover classifiers that are invariant only for subset of $\mathcal{X}$. It additionally states that a classifier must be invariant for each possible parameter $\theta \in \Theta$. For choosing a suitable invariant classification model, only the class of transformations $\mathcal{F}$ must be known a priori. Structural properties that induce an invariance or invariant concept classes can be selected to constrain the training of a classifier.

An invariance property of a classifier can be inherited to more complex classifier systems. An ensemble of invariant base classifiers $\mathcal{E} = \{c_e\}_{e=1}^{|\mathcal{E}|}$ can be combined to an invariant ensemble classifier via a fusion architecture $h : \mathcal{Y}^{|\mathcal{E}|} \longrightarrow \mathcal{Y}'$,

$$h_{\mathcal{E}}\Big(c_1\big(f_{\theta_1}(\mathbf{x})\big), \ldots, c_{|\mathcal{E}|}\big(f_{\theta_{|\mathcal{E}|}}(\mathbf{x})\big)\Big) = h_{\mathcal{E}}\big(c_1(\mathbf{x}), \ldots, c_{|\mathcal{E}|}(\mathbf{x})\big). \tag{4}$$

In this context, the final label space $\mathcal{Y}'$ must not be identical to the label space of the base classifiers. The construction principle can therefore be used for generating invariant multi classifier systems as well as multiclass classifiers ($|\mathcal{Y}'| > 2$). The base classifiers of the ensemble can potentially cope simultaneously with different parameter values $\theta_1, \ldots, \theta_{|\mathcal{E}|} \in \Theta$ of the data transformation. This can be used for constructing new invariances for example when the base classifiers operate on different feature subsets.

Although invariances can remove the effect of distracting data transformations, their application does not guarantee an improved classification performance. Invariant classifiers neglect a certain kind of information that cannot be used for the prediction of a class label. For example, scale invariant classifiers can neither rely on the original scale nor on a (possibly) transformed scale. The benefit of incorporating an invariance into a classification model depends on the type of data and its inherent noise.

2.2 Rank-based classifiers

Here, the focus lies on classifiers designed for real-valued data that operate on rank-transformed profiles. We will call these classifiers *rank-based* in the following. They will be denoted by c_{rk}. The rank transformation assigns a number $\mathrm{rk}_{\hat{\mathbf{x}}}(x)$ in the interval $[1\,,\,n]$ to each element $x \in \hat{\mathbf{x}}$. It indicates the position of x in $\hat{\mathbf{x}}$ according to an ordering relation. In our experiments we utilize the "$<$" relation. The smallest number is assigned to the smallest element. Formally the rank of an element $x \in \hat{\mathbf{x}}$ is determined by

$$\mathrm{rk}_{\hat{\mathbf{x}}}(x) = |\mathcal{L}_x| + \frac{1}{|\mathcal{I}_x|}\sum_{i=1}^{|\mathcal{I}_x|} i. \tag{5}$$

In this context the symbols $\mathcal{L}_x$ and $\mathcal{I}_x$ denote the sets of elements that are smaller than, or equal to x elements

$$\mathcal{L}_x = \left\{x' \in \hat{\mathbf{x}} \,|\, x' < x\right\} \quad \text{and} \quad \mathcal{I}_x = \left\{x' \in \hat{\mathbf{x}} \,|\, x' = x\right\}. \tag{6}$$

The function $\mathrm{rk}_{\hat{\mathbf{x}}}(x)$ will assign the same rank to each element $x' \in \mathcal{I}_x$. The rank transformation of a vector $\mathbf{x} = (x^{(1)}, \ldots, x^{(n)})^T$ can then be formulated as

$$\mathrm{rk}(\mathbf{x}) = \begin{pmatrix} \mathrm{rk}_{\hat{\mathbf{x}}}(x^{(1)}) \\ \vdots \\ \mathrm{rk}_{\hat{\mathbf{x}}}(x^{(n)}) \end{pmatrix}. \tag{7}$$

We apply the rank transformation individually on the feature profiles of each object. A rank-based classifier c_{rk} will operate on modified versions of the training set $\mathcal{T}$ and the test set $\mathcal{V}$

$$\mathcal{T}_{\mathrm{rk}} = \left\{\big(\mathrm{rk}(\mathbf{x}), y\big) \,|\, (\mathbf{x}, y) \in \mathcal{T}\right\} \quad \text{and} \quad \mathcal{V}_{\mathrm{rk}} = \left\{\big(\mathrm{rk}(\mathbf{x}), y\big) \,|\, (\mathbf{x}, y) \in \mathcal{V}\right\}. \tag{8}$$

The inner training routines of the real-valued classifiers are not modified. They will handle a ranking as a real-valued profile. Especially training algorithms for sparse classifiers that only operate on a small selection of $n' < n$ features receive the ranks calculated on the full n-dimensional feature profiles. The rank-based versions of sparse real-valued classifiers operating on $[1, n]^{n'}$ can therefore no longer be seen as sparse rank-based classifiers which operate on k-dimensional rankings $\left[1, n'\right]^{n'}$.

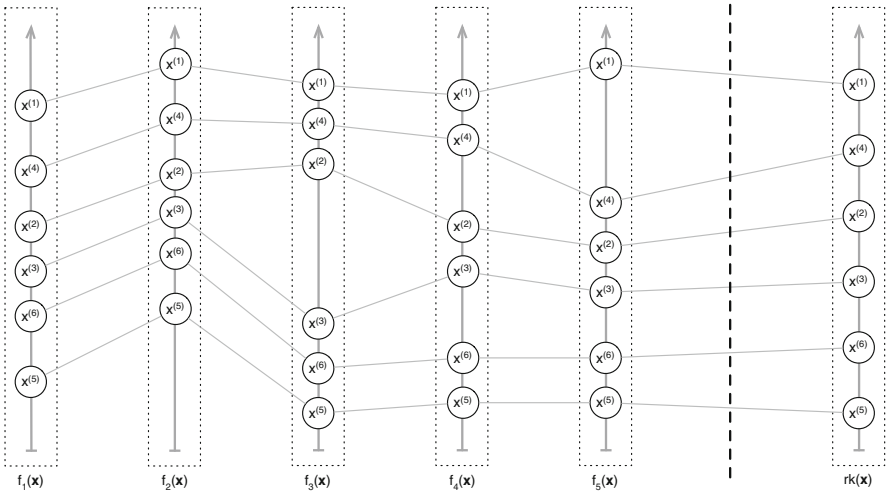

Fig. 2 Effects of the class of feature-wise strictly monotonically transformations $\mathcal{F}_\mathcal{G}$ on the feature profile of an object $\mathbf{x}$. The figure shows five examples for the effects of feature-wise strictly monotonically increasing data transformations $f_1, \ldots, f_5 \in \mathcal{F}_\mathcal{G}$. Feature-wise strictly monotonically increasing functions can affect the form and range of the feature profile of $\mathbf{x}$, but not the ordering of the measured values. This is reflected by the the rank transformation $\mathrm{rk}(\mathbf{x})$

Rank-based classifiers share an invariance against feature-wise strictly monotone data transformations (see also supplementary information).

Definition 2 The class of feature-wise strictly monotonically increasing functions is defined as $\mathcal{F}_\mathcal{G} = \left\{ f_g \mid g \in \mathcal{G} \right\}$, $f_g : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ with

$$f_g(\mathbf{x}) : \begin{pmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{pmatrix} \mapsto \begin{pmatrix} g(x^{(1)}) \\ \vdots \\ g(x^{(n)}) \end{pmatrix}. \tag{9}$$

Here $\mathcal{G}$ denotes the class of strictly monotonically increasing function $g : \mathbb{R} \to \mathbb{R}$

$$\forall x^{(i)}, x^{(i')} \in \mathbb{R} : g(x^{(i)}) < g(x^{(i')}) \Longleftrightarrow x^{(i)} < x^{(i')}. \tag{10}$$

The influence of the rank transformation on a randomly chosen feature profile $\mathbf{x}$ is illustrated in Fig. 2. While a feature-wise strictly monotonically increasing function can affect the form of the empirical density function of $\mathbf{x}$ and range of the values in $\mathbf{x}$, it cannot affect the ordering of the values in $\mathbf{x}$. The ordering within the feature profile of an object $\mathbf{x}$ is the only information reflected by the rank-transformed profile $\mathrm{rk}(\mathbf{x})$. A rank-based classifier therefore neglects the effects of strictly monotonically increasing data transformations. As a consequence information about the overall scale or density of untransformed feature profile can also not be taken into account for training a rank-based classifier. The rank-based classifier uses only the relative position of the measurements within the profile. The rank transformation additionally affects the

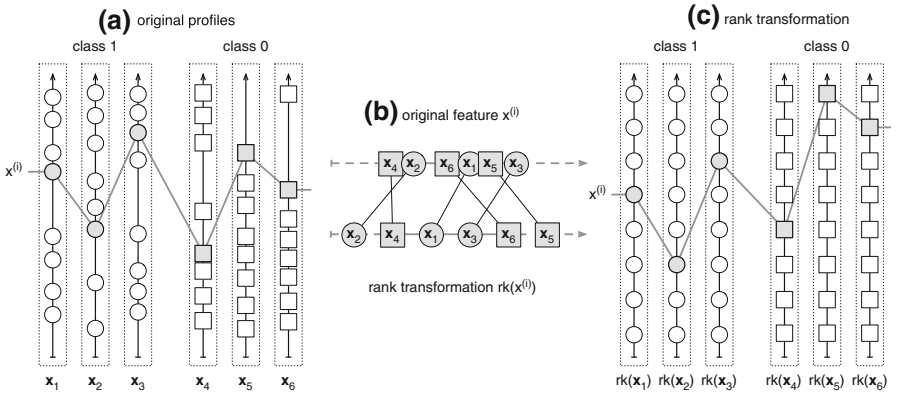

Fig. 3 Effects of the rank transformations of feature profiles on the univariate distributions of feature values. *Panel* **a** shows a collection of m = 6 real-valued profiles $\mathcal{S}\{(\mathbf{x}_j, y_j)\}_{j=1}^m$ of $n = 8$ features. The are separated in two classes $y \in \{0, 1\}$ denoted by the *symbols* $\{\bigcirc, \square\}$. *Panel* **c** shows the corresponding rank-transformed feature profiles $\mathcal{S}_{\text{rk}} = \{(\text{rk}(\mathbf{x}_j), y_j)\}_{j=1}^m$. In both *panels* a selected feature $x^{(i)}$ is highlighted. The distributions of feature values of $x^{(i)}$ and $\text{rk}_{\hat{\mathbf{x}}}(x^{(i)})$ are shown in *Panel* **b**. It can be observed that the rank transformation of the feature profiles has changed the ordering and the distribution of $x^{(i)}$

empirical distributions of the feature values (Fig. 3). Here, the ordering of the objects can be perturbed.

3 Experiments

We used two experimental setups for investigating the benefit of the rank transformation on the classification of gene expression profiles. Correlation (Sect. 3.1) and classification (Sect. 3.2) experiments were performed on a collection of 86 microarray datasets. As stated before, we will focus on binary classification problems $|\mathcal{Y}| = 2$. An overview on the analyzed datasets is given in Table 1 (see also the supplemental material). All datasets were normalized via robust multi-array average (RMA) normalization (Irizarry et al. 2003).

3.1 Correlation analyses

In the first experiment, we investigate the influence of the rank transformation on the correlations within a dataset. The pairwise correlation between features and the correlation between features and class labels is analyzed. The absolute value of Spearman's correlation coefficient is chosen for this experiment

$$\rho(\mathbf{a}, \mathbf{b}) = \frac{\left|\sum_{j=1}^m (\text{rk}_{\hat{\mathbf{a}}}(a_j) - \overline{\text{rk}})(\text{rk}_{\hat{\mathbf{b}}}(b_j) - \overline{\text{rk}})\right|}{\sqrt{\sum_{j=1}^m (\text{rk}_{\hat{\mathbf{a}}}(a_j) - \overline{\text{rk}})^2}\sqrt{\sum_{j=1}^m (\text{rk}_{\hat{\mathbf{b}}}(b_j) - \overline{\text{rk}})^2}}. \tag{11}$$

In this context, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ denote two vectors of size m and $\overline{\text{rk}} = \frac{m+1}{2}$ denotes their common mean rank.

For each dataset $\mathcal{S} = \{(\mathbf{x}_j, y_j)\}_{j=1}^m$ of m objects, we analyze the differences in pairwise correlations among two features

$$\boldsymbol{\rho}_{ff} = \left\{ \rho(\mathbf{X}^{(i)}, \mathbf{X}^{(i')}) - \rho(\mathbf{R}^{(i)}, \mathbf{R}^{(i')}) \right\}_{\substack{i,i' \in [n] \\ i<i'}}. \tag{12}$$

Here $\mathbf{X}^{(i)}$ denotes the vector of the values in the ith feature of the real-valued dataset $\mathcal{S}$ and $\mathbf{R}^{(i)}$ the equivalent of the rank-transformed dataset $\mathcal{S}_{rk}$

$$\mathbf{X}^{(i)} = \left(x_1^{(i)}, \ldots, x_m^{(i)}\right)^T \quad \text{and} \quad \mathbf{R}^{(i)} = \left(rk_{\hat{\mathbf{x}}_1}(x_1^{(i)}), \ldots, rk_{\hat{\mathbf{x}}_m}(x_m^{(i)})\right)^T. \tag{13}$$

The experiment is repeated for the differences in the correlations between the single features and the class labels

$$\boldsymbol{\rho}_{fc} = \left\{ \rho(\mathbf{X}^{(i)}, Y) - \rho(\mathbf{R}^{(i)}, Y) \right\}_{i \in [n]}, \tag{14}$$

where $\mathbf{Y} = (y_1, \ldots, y_m)^T$ denotes the vector of class labels.

3.2 Classification

In the second experiment, we evaluate the performance of rank-based classifiers in series of $R \times P$ cross-validation classification experiments (Bishop 2006). For a single P-fold cross-validation, the set of all labeled feature profiles $\mathcal{S} = \{(\mathbf{x}_j, y_j)\}_{j=1}^m$ is partitioned into P mutually exclusive parts of approximately equal size $\mathcal{P}_1, \ldots, \mathcal{P}_P \subseteq \mathcal{S}$. These subsets are utilized to form training sets $\mathcal{T}$ and validation sets $\mathcal{V}$ for P single experiments,

$$\mathcal{T}_p = \bigcup_{o \in [P] \setminus \{p\}} \mathcal{P}_o, \quad \mathcal{V}_p = \mathcal{P}_p, \quad \forall p \in [P]. \tag{15}$$

While one set $\mathcal{P}_p$ is excluded for subsequent validation, the remaining labeled feature profiles are used for training a classification model.

The P-fold cross-validation is typically repeated on R independent permutations of $\mathcal{S}$ resulting in $R \cdot P$ training-test splits denoted by $\mathcal{T}_{r,p}$ and $\mathcal{V}_{r,p}$. The final estimate of a classifier's error probability is given by

$$E_{R \times P} = \frac{1}{R|\mathcal{S}|} \sum_{r=1}^{R} \sum_{p=1}^{P} \sum_{(\mathbf{x},y) \in \mathcal{V}_{r,p}} \mathbb{I}_{\left[c_{\mathcal{T}_{r,p}}(\mathbf{x}) \neq y\right]}, \tag{16}$$

where $c_{\mathcal{T}_{r,p}}(\mathbf{x})$ denotes the classifier trained on $\mathcal{T}_{r,p}$. For our experiments, we have chosen a 10×10 cross-validation. All classifiers and all modifications were evaluated on identical training- and testsets.

The main focus of our experiments is the detection of pairwise differences between the error rates of real-valued classifiers and their rank-based counterparts:

$$E_{diff} = E^{re}_{10\times 10} - E^{rk}_{10\times 10}. \tag{17}$$

A positive value indicates that the rank-based classifier has outperformed the real-valued classifier in this experiment. All cross-validation experiments were performed within the TunePareto software framework (Müssel et al. 2012).

3.3 List of classifiers

The influence of the rank transformation of the features profiles is analyzed for three categories of classifiers, *linear classifiers*, *prototype-based classifiers* and *hierarchical classifiers*.

3.3.1 Linear classifiers

Linear classifiers utilize linear hyperplanes as decision boundaries for their predictions

$$c(\mathbf{x}) = \begin{cases} +1 & \text{if } \mathbf{w}^T\mathbf{x} \geq t \\ -1 & \text{otherwise.} \end{cases} \tag{18}$$

Here, the parameter $\mathbf{w} \in \mathbb{R}^n$, $\|\mathbf{w}\|_2 = 1$ determines the orientation of the hyperplane. The threshold $t \in \mathbb{R}$ determines the distance between the hyperplane and the origin. For our experiments, we have chosen three types of linear support vector machines (SVM) and two types of logistic regression (LOGR) (Fan et al. 2008). As a common cost parameter $C = 1$ was chosen.

L_1-Loss Support Vector Machine (L1-SVM) The basic idea of an SVM is the construction of a large margin classifier, which maximizes the distance between the training profiles and the decision boundary. This is achieved by minimizing the L_2-norm of $\mathbf{w}$,

$$\min_{\mathbf{w},t} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{(\mathbf{x},y)\in\mathcal{T}} \left(\max\left(0, 1 - y(\mathbf{w}^T\mathbf{x} - t)\right)\right), \tag{19}$$

with the threshold $t \in \mathbb{R}$ determining the distance of the hyperplane to the origin. The L1-SVM minimizes the hinge loss (second term), which takes into account the distance between a misclassified profile and the margin.

L_2-Loss Support Vector Machine (L2-SVM) In contrast to the L1-SVM, this training algorithm minimizes the squared hinge-loss

$$\min_{\mathbf{w},t} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{(\mathbf{x},y)\in\mathcal{T}} \left(\max\left(0, 1 - y(\mathbf{w}^T\mathbf{x} - t)\right)\right)^2. \tag{20}$$

In this version feature profiles with a large hinge loss (>1) have more influence on the optimization process than feature profiles with a smaller hinge-loss (<1).

L_1-regularized Support Vector Machine (R1-SVM) The R1-SVM is a feature selecting version of the support vector machine

$$\min_{\mathbf{w},t} \quad \|\mathbf{w}\|_1 + C \sum_{(\mathbf{x},y)\in\mathcal{T}} \left(\max\left(0, 1 - y(\mathbf{w}^T\mathbf{x} - t)\right)\right)^2. \tag{21}$$

By regularizing by L_1-norm, small weights ($|w^{(i)}| < 1$) are forced towards zero. The corresponding features will not influence the final decision boundary.

Logistic Regression (LOGR) This training algorithm is based on the minimization of the logistic loss function

$$\min_{\mathbf{w},t} \quad \frac{1}{r}\|\mathbf{w}\|_r^r + C \sum_{(\mathbf{x},y)\in\mathcal{T}} \log\left(1 + e^{-y(\mathbf{w}^T\mathbf{x}-t)}\right). \tag{22}$$

Similar to the SVM, an L_1-regularized ($r = 1$, R1-LOGR) and an L_2-regularized ($r = 2$, R2-LOGR) version exist.

3.3.2 Prototype-based classifiers

The decision rule of these classifiers is based on a set of labeled reference points $\mathcal{Z} = \{(\mathbf{z}_j, y_j)\}_{j=1}^{|\mathcal{Z}|}$, $\mathbf{z}_j \in \mathbb{R}^n$, so-called prototypes. The label of a new unseen feature profile $\mathbf{v} \in \mathbb{R}^n$ is predicted by evaluating its distances (or similarities) to the prototypes

$$c(\mathbf{v}) = \underset{y\in\mathcal{Y}}{\operatorname{argmax}} \left|\{(\mathbf{z}, y) \in \mathrm{NN}_k(\mathbf{v}, \mathcal{Z})\}\right|. \tag{23}$$

Here, $\mathrm{NN}_k(\mathbf{v}, \mathcal{Z})$ denotes the k nearest neighborhood of $\mathbf{v}$ in the set of prototypes

$$\mathrm{NN}_k(\mathbf{v}, \mathcal{Z}) = \left\{(\mathbf{z}, y) \in \mathcal{Z} \mid \mathrm{rk}_{D_\mathbf{v}}\left(d(\mathbf{v}, \mathbf{z})\right) \leq k\right\} \tag{24}$$

and $\mathrm{rk}_{D_\mathbf{v}}$ denotes the rank function with respect to the set of pairwise distances between $\mathbf{v}$ and the members of $\mathcal{Z}$

$$D_\mathbf{v} = \left\{d(\mathbf{v}, \mathbf{z}) \mid (\mathbf{z}, y) \in \mathcal{Z}\right\}. \tag{25}$$

In this context, $d(.,.)$ is an arbitrary chosen but fixed distance measure. Our experiments comprise four training algorithms for prototype-based classifiers. They are commonly based on the Euclidean distance.

k-Nearest-neighbor (k-NN) The k-Nearest-Neighbor classifier (k-NN) (Fix and Hodges 1951) can be seen as an extreme case of an prototype-based approach. It

utilizes all available training profiles as potential prototypes, $\mathcal{Z} = \mathcal{T}$. It is parameterized by k, which determines the size of the k nearest neighborhood. In this work, we have conducted experiments for $k \in \{1, 3, 5, 7\}$.

Nearest centroid classifier (NCC) The NCC is an example for an algorithm based on center-type prototypes. It utilizes the class-wise centroids as reference points $\mathcal{Z} = \{(\bar{\mathbf{x}}_y, y)\}_{y \in \mathcal{Y}}$.

Nearest Shrunken centroid classifier (NSC) The NSC can be seen as a feature selecting version of the NCC (Tibshirani et al. 2002). In its training process, this classifier shrinks the class-wise centroids towards the global centroid (centroid over all feature profiles in $\mathcal{T}$). A feature does not influence prediction of NSC, if the corresponding distance between the class-wise centroids and the global centroid is equal to zero.

Representative prototype sets (RPS) The RPS classifier (Lausser et al. 2012) is based on a set of prototypes which is directly selected from the available training profiles $\mathcal{Z} \subseteq \mathcal{T}$. For each class, one prototype is selected from its training samples $\mathcal{T}_y$. From all sets fulfilling these properties $\mathcal{ZR} = \{\mathcal{Z}' \subseteq \mathcal{T} \,|\, |\mathcal{Z}'| = |\mathcal{Y}|, \forall y \in \mathcal{Y} : \mathcal{T}_y \cap \mathcal{Z}' \neq \emptyset\}$ the set of prototypes minimizing the error rate on the available training profiles is selected

$$\mathcal{Z} = \underset{\mathcal{Z}' \in \mathcal{ZR}}{\operatorname{argmin}} \sum_{(\mathbf{x}, y) \in \mathcal{T}} \mathbb{I}_{[c_{\mathcal{Z}'}(\mathbf{x}) \neq y]}. \tag{26}$$

3.3.3 Hierarchical classifiers

Single threshold classifier (STC) In contrast to the other classifiers in our panel, the single threshold classifier (Kestler et al. 2011) (decision stump or single ray classifier) only evaluates a single feature $x^{(i)}$, $i \in [n]$ of the feature profile. It determines the class label of an object according to a single threshold $t \in \mathbb{R}$

$$c(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad s(x^{(i)} - t) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{27}$$

Here $s = \pm 1$ indicates whether the objects above or below the threshold should be assigned to class 1 or class 0. We have utilized two criteria for training STCs. The classifiers were either optimized for minimizing the unweighted training error or the class-wise weighted training error.

Classification and regression trees (CART) A classification tree can be seen as hierarchical system of decision stumps (Breiman et al. 1984). Instead of directly predicting the class label of an object, the top level classifier passes the feature profile $\mathbf{x}$ to subsequent classifiers $c_l(\mathbf{x})$ or $c_r(\mathbf{x})$ which either pass the profile theirselves (inner node) or predict a class label (leave)

$$c(\mathbf{x}) = \begin{cases} c_l(\mathbf{x}) & \text{if} \quad s(x^{(i)} - t) \geq 0 \\ c_r(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{28}$$

In its training process, the tree classifiers recursively split the training data $\mathcal{T}$ into more and more homogeneous subsets. A new subsequent classifier is adapted only on the basis of the data in one of the available subsets. This procedure is recursively repeated until the cardinality of a subset falls below a minimum number of training examples ($m_{\min} = 5$) or training examples are correctly classified.

Random forest (RF) The random forest (Breiman 2001) can be seen as an ensemble classifier of $|\mathcal{E}|$ classification trees $\mathcal{E} = \{c_{T_e}\}_{e=1}^{|\mathcal{E}|}$. It predicts the class label of a new unseen object $\mathbf{x}$ according to a majority vote

$$c(\mathbf{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \left| \{c_T(\mathbf{x}) = y \mid c_T \in \mathcal{E}\} \right|. \tag{29}$$

Each tree classifier is trained on an independent bootstrap resample $\mathcal{T}_e^*$ of $\mathcal{T}$. In this process, the tree classifiers are restricted in their selection of suitable features. For each split, they are allowed to selected one out of $\sqrt{n}$ randomly selected features for a axis-parallel separation. In our experiments we have utilized RFs of size $|\mathcal{E}| \in \{50, 100, 150, 200\}$.

4 Results

The pairwise differences between the correlations of real-valued and rank-transformed datasets can be found in Fig. 4. Panel a summarizes the pairwise differences of the feature-feature correlations over all analyzed datasets. The rank transformation can lead to both increased and decreased absolute correlations. Over all datasets, the median maximal difference lies at 0.6498. The corresponding median minimal value lies at -0.6240. In median, the first and the third quartile lies at -0.0372 and 0.0806. Looking at the median over all datesets, more than 59% of all differences are positive.

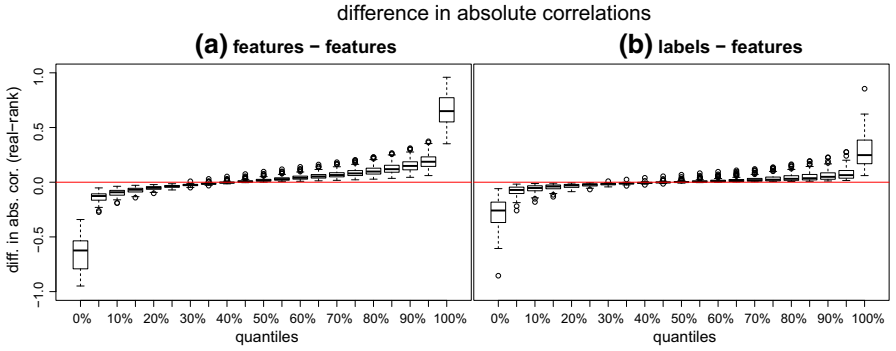

Fig. 4 Pairwise differences in correlations (absolute Spearman's rank correlation) between real-valued and rank-transformed datasets. For each dataset, the corresponding distribution of differences is summarized in quantiles. The *boxplots* show the distributions of these quantiles over all datasets. *Panel* **a** displays the distributions of pairwise correlations between features. *Panel* **b** shows the distributions for correlations between features and class labels

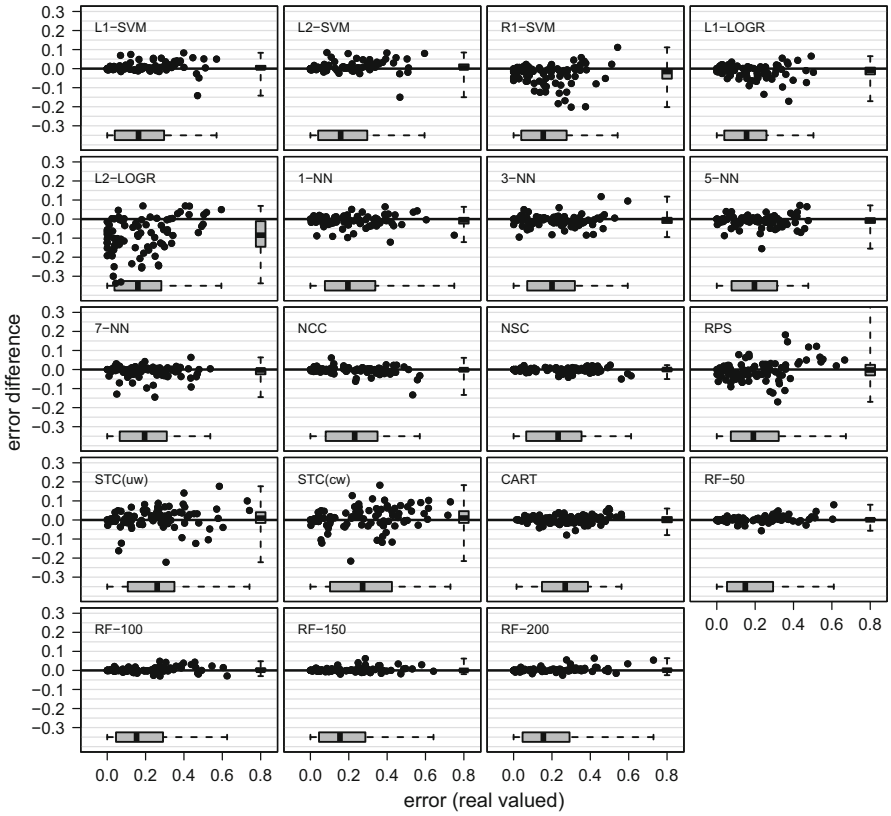

Fig. 5 Real-valued classifiers vs. rank-based classifiers: the figure summarizes the error rates of the 10×10 cross validation experiments. *Each dot* corresponds to the error rates achieved on one of the datasets listed in Table 1. The *horizontal axis* shows the error rates of the real-valued classifier. The *vertical axis* displays the differences in error rates between a real-valued classifier and its rank-based counterpart. A difference value indicates that the real-valued classifier shows a higher error rate than the rank-based classifier

Smaller absolute differences can be observed for the correlations between features and class labels (Panel b). The median differences range from 0.2478 to −0.2586. In median, the first and the third quartile lie by −0.0221 and 0.0232 respectively. About 53% of all differences are positive.

A comparison of the cross-validation results of real-valued classifiers and rank-based classifiers is given in Fig. 5. Each panel shows the results for one classifier. For each dataset, a dot indicates the error rate of the real-valued classifier (x-axis) and its difference towards the error rate of the corresponding rank-based classifier (y-axis). The median error rates over all datasets for the real-valued classifiers ($Q^{re}_{0.5}$) and the rank-based classifiers ($Q^{rk}_{0.5}$) as well as the median (pairwise) error rate difference ($Q^{diff}_{0.5}$) are listed in Table 2. For the real-valued classifiers the best median error rates are achieved by random forests ($Q^{re}_{0.5} \in [0.149, 0.155]$) and support vector machines ($Q^{re}_{0.5} \in [0.155, 0.163]$). The prototype-based classifiers achieve a higher median error

Table 2 Error rates of the 10×10 cross-validation experiments with real-valued classifiers ($E^{re}_{10\times10}$) and rank-based classifiers ($E^{rk}_{10\times10}$)

	$E^{re}_{10\times10}$			$E^{rk}_{10\times10}$			E_{diff}		
	$Q^{re}_{0.25}$	$Q^{re}_{0.5}$	$Q^{re}_{0.75}$	$Q^{rk}_{0.25}$	$Q^{rk}_{0.5}$	$Q^{rk}_{0.75}$	$Q^{diff}_{0.25}$	$Q^{diff}_{0.5}$	$Q^{diff}_{0.75}$
L1-SVM	0.040	0.163	0.293	0.035	0.155	0.279	−0.002	0.004	0.014
L2-SVM	0.041	0.158	0.296	0.035	0.155	0.279	−0.001	0.004	0.021
R1-SVM	0.042	0.155	0.274	0.062	0.184	0.307	−0.053	−0.015	−0.002
L1-LOGR	0.041	0.155	0.258	0.047	0.168	0.290	−0.029	−0.009	0.005
L2-LOGR	0.039	0.160	0.281	0.159	0.249	0.378	−0.144	−0.085	−0.012
1-NN	0.077	0.195	0.336	0.092	0.214	0.345	−0.024	−0.004	0.007
3-NN	0.071	0.200	0.319	0.084	0.201	0.332	−0.021	−0.003	0.005
5-NN	0.077	0.196	0.312	0.086	0.194	0.349	−0.020	−0.006	0.004
7-NN	0.070	0.195	0.311	0.097	0.202	0.329	−0.025	−0.002	0.003
NCC	0.080	0.231	0.349	0.073	0.237	0.356	−0.009	−0.001	0.004
NSC	0.066	0.232	0.354	0.071	0.239	0.353	−0.007	0.000	0.006
RPS	0.076	0.191	0.322	0.091	0.197	0.325	−0.030	−0.002	0.024
STC(uw)	0.109	0.260	0.350	0.107	0.219	0.334	−0.015	0.010	0.041
STC(cw)	0.102	0.272	0.420	0.113	0.249	0.426	−0.015	0.010	0.046
CART	0.149	0.269	0.385	0.162	0.268	0.390	−0.012	0.002	0.015
RF-50	0.053	0.149	0.294	0.051	0.153	0.288	−0.002	0.001	0.008
RF-100	0.048	0.153	0.290	0.046	0.155	0.281	−0.002	0.001	0.009
RF-150	0.046	0.154	0.287	0.047	0.154	0.283	−0.002	0.000	0.005
RF-200	0.048	0.155	0.290	0.047	0.155	0.286	−0.001	0.001	0.006

The first, second and third quartile over all tested datasets are shown. Additionally the paired differences of the error rates (E_{diff}) of a real-valued classifier and its rank-based classifier are reported

rate $Q^{re}_{0.5} \in [0.191, 0.232]$. The highest median error rates are gained by the STC and CART $Q^{re}_{0.5} \in [0.260, 0.272]$.

For the rank-based classifiers, only the support vector machines with L_2 regularization achieve low median error rate $Q^{rk}_{0.5} = 0.155$. The L_1 regularized R1-SVM and the two versions of the logistic regression achieve a higher median error rate $Q^{rk}_{0.5} = 0.184$ and $Q^{rk}_{0.5} \in [0.168, 0.249]$, respectively. The random forests gain median error rates $Q^{rk}_{0.5} \in [0.153, 0.155]$. The prototype-based classification models show in general slightly increasing median error rates $Q^{rk}_{0.5} \in [0.194, 0.239]$. The median error rates of the CART classifier and the STCs lie in $Q^{rk}_{0.5} \in [0.219, 0.268]$. We found significant positive median differences in error rates for the L_2 regularized SVMs, the STCs and the random forests RF-50, RF-100 and RF-200 (one-sided and paired Wilcoxon rank sum test, FDR correction all $q < 0.05$).

The interquartile range of the error rate differences ($IQR = Q^{diff}_{0.75} - Q^{diff}_{0.25}$) varies among the training algorithms. The smallest variation is gained by the random forests, which achieved an $IQR < 0.012$ for all tested parameters. They are followed by

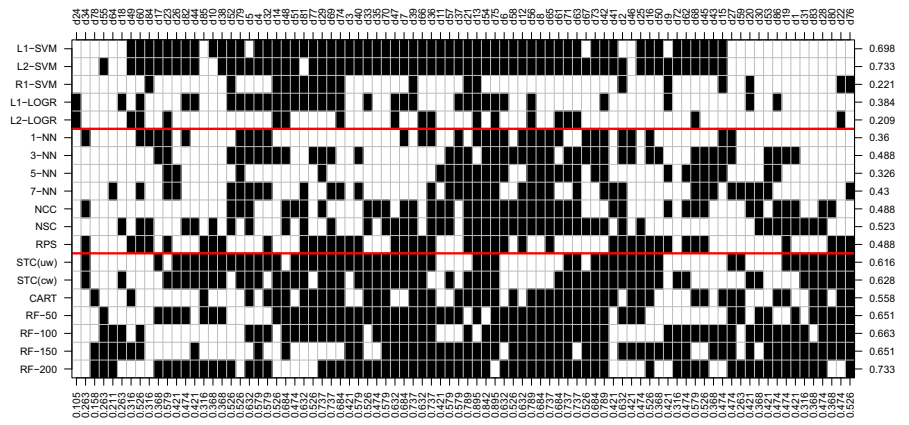

Fig. 6 Comparison of real-valued and rank-based classifiers: The figure expands a grid of all tested classifiers (*rows*) and datasets (*columns*). A cell is *colored black*, if the cross-validation error of the real-valued classifier is larger than or equal to the cross-validation error of the corresponding rank-based classifier (847 of the 1634 cells). For each classifier or dataset, the percentage of *black* cells is given

the centroid based algorithms NCC ($IQR = 0.012$) and NSC ($IQR = 0.013$). The large margin classifiers achieved an $IQR = 0.016$ (L1-SVM) and $IQR = 0.022$ (L2-SVM). For different values of k, the k-NN classifier gains *IQR*s between 0.024 and 0.030. CART shows an *IQR* of 0.027. The highest variations are achieved for the L1-LOGR ($IQR = 0.034$), the R1-SVM ($IQR = 0.051$), the RPS ($IQR = 0.055$), the STCs ($IQR \leq 0.061$) and the L2-LOGR ($IQR = 0.132$).

Figure 6 shows a bitmap of all pairwise comparisons between real-valued classifiers and rank-based classifiers. For each dataset, a black square denotes a better or equal performance of the rank-based approach. The numbers give the percentages of black squares per dataset or classifier.

As an optimum, 89.5% of all tested rank-based classifiers perform comparably or better than their real-valued equivalents (datasets $d13$, $d75$). The scaling of these datasets may be uninformative or even misleading for the analysed categorizations. There also exist datasets for which the rank-based methods only rarely reach the cross-validation errors of real-valued ones in most of the cases. For example, the rank-based classifiers only outperform their real-valued counterparts in less than 15.8% of all experiments for the datasets $d24$ and $d78$. For these scenarios, the relative ordering of the gene expression levels seem to miss important information.

Classifiers exist for which the rank-based version shows an equal or better performance than the real-valued one with a frequency $>50\%$ (one-sided binomial tests, FDR correction). The rank-based random forests perform comparable or better than their real-valued counterparts on at least 65% (all $p \leq 0.011$) of the tested datasets. Similar observations can be made for the rank-based large margin classifiers. The L_2 regularized versions of the SVM work better or equal than the real-valued ones in 69.8% (L1-SVM, $p = 1 \cdot 10^{-3}$) and 73.3% (L2-SVM, $p = 9 \cdot 10^{-5}$) of all cases. The single threshold classifiers show increased frequencies of 61.6% (STC(uw), $p = 0.047$) and 62.8% (STC(cw), $p = 0.031$). For the prototype-based classifiers, the rank-based

versions perform better or equal in less than 52.3% of the experiments. Rank-based logistic regression performs comparable or better in 38.4% (L1) and 20.9% (L2) of the cases. The rank-based version of the R1-SVM is only comparable to its real-valued counterpart on 22.1% of all tested datasets.

5 Conclusion

Rank transformation is a general way of incorporating invariances into standard classification models for real-valued data. The resulting rank-based classifiers are guaranteed to be invariant against all feature-wise strictly monotonically increasing data transformations. This might be a beneficial property in multi-center or multi-platform studies (Patil et al. 2015) or other multi-array analyses (McCall et al. 2010). We also observed a decorrelating effect for the analyzed gene expression profiles. They therefore show an increased diversity in contrast to their real-valued counterparts (Hariharan et al. 2012). Nevertheless the restriction to certain aspects of a dataset can also be limiting. For example, the global scaling of a feature profile cannot be utilized for the categorization of an object.

In our study we have analyzed the usefulness of 19 rank-based classifiers for the binary classification of gene expression profiles on a panel of 86 microarray datasets. Our results show that classifiers exist for which an initial rank transformation can be applied beneficially and with a low risk of increasing error rates. Independent from the chosen training algorithms, hierarchical classifiers showed stable improvements ($\geq$55.8%). The tested random forests showed the most stable results. Incorporating a rank transformation has only marginal effects on the performance of these classifiers. The random forests can be made invariant against the discussed misleading data transformations without changing the behavior on unaffected test objects. This is an interesting result as the real-valued and rank-based single threshold classifiers, which are the main ingredients of a random forest, show large performance differences. While the rank-based single threshold classifiers show a better or equal performance than the real-valued ones in at least 61.6% of all experiments, these classifiers can also lead to an error rate which is increased up to 22.2%. Other base classifiers might be more beneficial or more stable in this context.

The largest impact of the chosen training algorithm was observed for the concept class of linear classifiers. Strong and frequent improvement could be observed for the L_2 regularized support vector machines. These classifiers performed equal or better than their real-valued counterparts in up to 73.3% of our experiments. Nevertheless other classifiers exist for which rank-transformed data seems to be misleading. The rank-based version of the L_1 regularized support vector machine was outperformed by the real-valued version in 77.9% of all cases. Both logistic regressions were improved in less than 38.4%, which might be referred to the logistic loss. The real-valued versions of tested prototype-based classifiers outperformed the rank-based versions in up to 67.4% of all experiments. The rank-transformed k-nearest neighbor classifiers were outperformed more frequently than the centroid-based algorithms. This might be an effect of the chosen Euclidean distance. Other distance metrics which are designed for ordinal data might be more suitable for the rank-based versions.

Generally, our results indicate a superiority of the rank transformation for the SVM and random forest classifiers. Over all classifiers and datasets we find a slight dominance using the rank versions. This indicates that for categorisation purposes not the absolute values of the gene expression measurements, but their relative order to each other is the determining characteristic of the profiles; and that the rank transformation removes deteriorating noise induced e.g. by varying experimental conditions from the data.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/20072013) under Grant Agreement No. 602783, the German Research Foundation (DFG, SFB 1074 project Z1 to HAK), and the Federal Ministry of Education and Research (BMBF, Gerontosys II, Forschungskern SyStaR, project ID 0315894A and e:Med, SYMBOL-HF, Grant ID 01ZX1407A) all to HAK.

References

Bavaud F (2009) Aggregation invariance in general clustering approaches. Adv Data Anal Classif 3(3):205–225

Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z (2000) Tissue classification with gene expression profiles. J Comput Biol 7(3–4):559–583

Bishop CM (2006) Pattern recognition and machine learning (information science and statistics). Springer, New York

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. The Wadsworth statistics/probability series. Chapman & Hall/CRC, Boca Raton

Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: a library for large linear classification. J Mach Learn Res 9:1871–1874

Fix E, Hodges JL (1951) Discriminatory analysis: nonparametric discrimination: consistency properties. Tech. Rep. Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, Randolf Field, Texas

Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Haasdonk B, Burkhardt H (2007) Invariant kernel functions for pattern analysis and machine learning. Mach Learn 68(1):35–61

Hariharan B, Malik J, Ramanan D (2012) Discriminative decorrelation for clustering and classification. In: Fitzgibbon AW, Lazebnik S, Perona P, Sato Y, Schmid C (eds) Computer Vision–ECCV 2012, Springer, Lecture notes in computer science 7575:459–472

Irizarry R, Hobbs B, Collin F, Beazer-Barclay Y, Antonellis K, Scherf U, Speed T (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. Biostatistics 4(2):249–264

Jamain A, Hand D (2009) Where are the large and difficult datasets? Adv Data Anal Classif 3(1):25–38

Kestler HA, Lausser L, Lindner W, Palm G (2011) On the fusion of threshold classifiers for categorization and dimensionality reduction. Comput Stat 26(2):321–340

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 (ANNPR12), Lecture notes in artificial intelligence, Springer, Heidelberg 7477:36–47

McCall M, Bolstad B, Irizarry R (2010) Frozen robust multiarray analysis (fRMA). Biostatistics 11(2):242n++253

Müssel C, Lausser L, Maucher M, Kestler HA (2012) Multi-objective parameter selection for classifiers. J Stat Softw 46(5):1–27

Niyogi P, Poggio T, Girosi F (1998) Incorporating prior information in machine learning by creating virtual examples. IEEE Proc Intell Signal Process 86(11):2196–2209

Patil P, Bachant-Winner PO, Haibe-Kains B, Leek J (2015) Test set bias affects reproducibility of gene signatures. Bioinformatics 31(14):2318–2323

Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):2507–2517

Schmid F, Lausser L, Kestler HA (2014) Linear contrast classifiers in high-dimensional spaces. In: Gayar NE, Schwenker F, Suen C (eds) Artificial neural networks in pattern recognition (ANNPR14), Springer, Heidelberg, Lecture notes in artificial intelligence 8774:141–152

Schölkopf B, Burges C, Vapnik V (1996) Incorporating invariances in support vector learning machines. In: von der Malsburg C, von Seelen W, Vorbrüggen J, Sendhoff S (eds) Artificial neural networks—ICANN'96, Springer, Lecture Notes in Computer Science, 1112:47–52

Schölkopf B, Smola A, Müller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319

Simard PY, LeCun YA, Denker JS, Victorri B (2012) Transformation invariance in pattern recognition—tangent distance and tangent propagation. In: Orr G, Müller KR (eds) Neural networks: tricks of the trade, vol 7700, 2nd edn., Lecture notes in computer scienceSpringer, Heidelberg, pp 239–274

Thomas J, Olson J, Tapscott S, Zhao L (2001) An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. Genome Res 11(7):1227–1236

Tibshirani R, Hastie T, Narasimhan B, Chu G (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc Natl Acad Sci USA 99(10):6567–6572

Tsuda K (1999) Support vector classifier with asymmetric kernel functions. In: Verleysen M (ed) Proceedings of ESANN'99 - European symposium on artificial neural networks, D-Facto public, Brussels, pp 183–188

Wood J (1996) Invariant pattern recognition: a review. Pattern Recogn 29(1):1–17