Comput Stat manuscript No.
(will be inserted by the editor)

Benchmarking local classification methods

Bernd Bischl · Julia Schiffner · Claus Weihs

Received: date / Accepted: date

Abstract In recent years in the fields of statistics and machine learning an increasing amount of so called local classification methods has been developed. Local approaches to classification are not new, but have lately become popular. Well-known examples are the k nearest neighbors method and classification trees. However, in most publications on this topic the term "local" is used without further explanation of its particular meaning. Only little is known about the properties of local methods and the types of classification problems for which they may be beneficial. We explain the basic principles and introduce the most important variants of local methods. To our knowledge there are very few extensive studies in the literature that compare several types of local methods and global methods across many data sets. In order to assess their performance we conduct a benchmark study on real-world and synthetic tasks. We cluster data sets and considered learning algorithms with regard to the obtained performance structures and try to relate our theoretical considerations and intuitions to these results. We also address some general issues of benchmark studies and cover some pitfalls, extensions and improvements.

Keywords Local classification methods · Benchmark study · Machine learning · Model selection

1 Introduction

Recently, there has been a lot of interest in local approaches to classification. Their origins date back to the k nearest neighbors method (kNN) of Fix

Bernd Bischl and Julia Schiffner contributed equally to this work.

B. Bischl · J. Schiffner · C. Weihs
Department of Statistics, TU Dortmund University, 44221 Dortmund, Germany
Tel.: +49-231-755-{4355, 5409}
Fax: +49-231-755-4387
E-mail: {bischl, schiffner}@statistik.tu-dortmund.de

and Hodges (1951). Besides kNN, other local approaches such as recursive partitioning methods (e.g., Breiman et al 1984), radial basis function (RBF) neural networks as well as RBF support vector machines (SVMs) are widely used. Moreover, many of the recently proposed methods are localized versions of standard methods. Examples are localized variants of linear discriminant analysis (Czogiel et al 2007; Sugiyama 2007), logistic regression (Hand and Vinciotti 2003; Tutz and Binder 2005) as well as boosting (Zhang and Zhang 2008), local versions of SVMs (Segata and Blanzieri 2010a) and local neural networks (Alpaydin and Jordan 1996). The main idea behind local methods is as follows: Since it is often difficult to find a single classification rule that is suitable for the whole population, rather calculate several local rules that are only valid for certain regions of the predictor space. When predicting the class label of an observation x the prediction is mainly determined by rules based on local regions near x. There are various possible implementations of this idea leading to different types of local methods.

There are several reasons why we are interested in local classification methods. First, local methods seem to be increasingly popular and many authors claim that local methods have certain advantages, e. g., in computation time, interpretability or performance. Another very fundamental reason lies in the fact that new classification methods are steadily published, and a vast number of approaches is already available. In order to choose methods appropriate to a certain classification problem a deeper understanding of the relationship between properties of classifiers, characteristics of classification problems and expected performance would be very helpful. In light of this, it appears to be useful to investigate how the fact that a method is local affects its performance. Concrete questions of interest are: How do local methods perform in general? How do localized versions perform in comparison to the base method? Do local methods of the same type show similar performance? Can we identify classification problems where local methods are beneficial? Are different types of local methods appropriate for different classification problems?

Unfortunately, it is hardly possible to answer these questions from the available literature for several reasons. First, many authors when proposing a new (local) classification method illustrate its usefulness only on a small number of artificial or real-world data sets and often compare it only to few other methods – which are not necessarily the strongest competitors. Likewise, it is seldom addressed if observed differences in performance are actually significant or an artifact of the random nature of the experiment. To our knowledge there are no extensive studies that compare several types of local methods across many data sets and employ significance tests. Therefore, we conduct a benchmark study on real-world and synthetic data. We base this study on the analysis framework of Hothorn et al (2005) and the practical extension and implementation by Eugster et al (2008). Our goal is not, as it is often the case in comparison studies, to establish an overall-ranking of methods, but a differentiated analysis of the results with respect to the questions above.

The paper is organized as follows. In Sect. 2 an introduction to local approaches to classification is given and the local methods under consideration

are described. Subsequently, the basics of benchmark analysis are reviewed and the experimental setup is described in Sect. 3. The results of our study are presented and discussed in Sect. 4 and finally, in Sect. 5 a summary and an outlook to future work are given.

2 Local classification methods

2.1 Basic principles

In the literature the term "local classification method" is not clearly defined. One of the few existing formal characterizations is by Devroye et al (1996) who define a classification rule as k-local if it depends exclusively on the k nearest neighbors of the test observation x. Additionally, many rather informal characterizations exist that are to some extent contradictory and give rise to different types of local methods. For example, Hand and Vinciotti (2003) state that a local classification rule relies mainly on the training observations near the class boundary since this is the region where an accurate fit of the underlying model matters most. Sugiyama (2007) proposes a modified version of Fisher discriminant analysis that he terms local because the pairwise similarities of training observations are taken into account when fitting the model.

In contrast, global methods are characterized as follows: Hand and Vinciotti state that here "all aspects of the data and the distributions contribute to the estimate of goodness of fit". Cheng et al (2010) write that "global learning approaches attempt to build a complex model by using the global characteristics of the data". According to these characterizations global methods include for instance linear discriminant analysis, logistic regression and multilayer perceptrons with the usual sigmoid activation functions.

We now describe common characteristics of local methods and later on in Sect. 2.2 especially address the approaches we consider in our benchmark study. From the characterizations above we can see that it is usually assumed that in predictor space there are several local regions of interest. Terms like distance and similarity are often used to formally define these regions. In the examples given in the beginning of this section "local" refers either to closeness to a trial point, closeness to the class boundary or similarity between pairs of training observations. The latter one can be understood as a description of the local cluster structure in the data. Usually, observation weights are calculated from these similarities or distances that regulate the influence of observations during the training process. This is normally done by rescaling distances through a window function, so that observations in a local region of interest receive higher weights. For time efficiency often window functions with finite support are used. This results in fitting models on a subset of the training data. In case of a k-local classification rule a rectangle window is used. Its width equals the distance to the k-th nearest neighbor of the test observation x. A local classification method induces one or more local classifiers

and aggregates them if necessary. Aggregation is done in such a way that when making predictions the local classifiers trained on local regions near the trial point have largest influence on the prediction.

Since the terms distance and similarity are involved in local classification we now have a closer look at the relation to distance-based approaches. These methods "classify objects by dissimilarity between them as measured by a distance function" (Shen 2006). Many distance-based methods work by simply assigning a trial point to "the closest class" (Clarke et al 2009, Sect. 5.2, p. 235), but this does not necessarily result in a local approach. For example, measuring the distance to a class by the distance to the closest training observation in this class produces the 1 nearest neighbor method which clearly is a local method. Measuring the distance to a class by assessing the distance to its mean vector (cp. Shen 2006, Sect. 4.1, p. 72) results in a global method, a.k.a. the minimum distance classifier, since all training observations contribute equally to the fit. A further example is Fisher discriminant analysis that Clarke et al (2009, Sect. 5.2.1, p. 236) mention as one of the earliest distance-based classifiers. Here, the data are projected into a space with lower dimensionality and distances to the class means in this embedding space are used.

2.2 Local classification methods under consideration

In the following we will introduce the local classification methods we consider in our benchmark study. Due to space limitations we cannot go into much detail here, but we will explain the basic ideas and point to further references. The methods are listed according to the underlying meaning of the term "local" which we think is a natural way of presentation. The question if local methods of the same type show similar performance is investigated in the second part of this paper.

2.2.1 Observation-specific methods: k nearest neighbors and localized linear discriminant analysis

Observation-specific methods or *instance-based* approaches are by far the best-known type of local methods. A review paper is by Atkeson et al (1997). The term "local" refers to the environment of a test observation. For each test observation a local classification rule is built based on nearby training observations. In our study we consider the k nearest neighbors method (kNN, Fix and Hodges 1951) where a test observation x is assigned the most frequent class among the k nearest training observations. Technically speaking the predictor space is partitioned into a large number of local regions, the Voronoi cells corresponding to the different possible sets of k nearest neighbors. For each Voronoi cell a local classification rule is built on the k nearest training observations. For a test observation x it is assessed in which Voronoi cell it falls and the corresponding local classifier is used to predict the class label. In

case of kNN the local classifiers are constant functions in the predictor variables. Generally, also more complex functions can be used. For example, Tutz and Binder (2005) use penalized logistic regression to create local classifiers, Blanzieri and Melgani (2006); Zhang et al (2006) propose to use support vector machines and Kotsiantis and Pintelas (2004) describe an observation-specific version of boosting. In our benchmark study we also consider a combination of kNN with linear discriminant analysis proposed by Czogiel et al (2007). For each test observation an individual linear discriminant analysis model is fitted based on the k nearest neighbors.

2.2.2 Multiple prototype methods: Learning vector quantization and mixture discriminant analysis

These methods represent the training data by a reduced number of prototypes. Each prototype has a class label and normally there are multiple prototypes per class. Usually, an object is assigned the class of its closest prototype. That is, the predictor space is partitioned into Voronoi cells that correspond to individual prototypes. Again, there is one local classifier per Voronoi cell which is a constant function in x.

This group of methods is closely related to observation-specific approaches. If every training observation is itself a prototype and x is assigned the class of the closest prototype the 1NN method results. If the most frequent class among the k nearest prototypes is chosen we obtain kNN. Since the number of prototypes is usually smaller than the number of training observations prototype methods help to reduce the memory amount required by the kNN method.

Although kNN can be regarded as a prototype method the locality concepts behind kNN (as well as combinations of kNN with other classifiers) and the prototype methods described in the following are different. While for kNN 'local' refers to closeness to the test observation for the methods mentioned below 'local' relates to similarity between training observations.

For example a clustering method can be applied to each class and the cluster centers are used as prototypes (e. g., K-means classification, Hastie et al 2009, Sect. 13.2.1, p. 460). In our benchmark study we consider learning vector quantization (Kohonen 1989). The LVQ1 algorithm works as follows: First, an initial set of prototypes for every class is chosen by random sampling from the training data. In an iterative procedure training points are sampled one at a time. For each training point the class label of the nearest prototype is determined. If both, the training point and the prototype, are in the same class the prototype is moved towards the training point. Otherwise the prototype is moved in the opposite direction. In each iteration step the learning rate which controls how far the prototypes are moved is decreased. The procedure is stopped when the learning rate reaches zero.

It is also possible to employ model-based clustering methods, i. e., to fit mixture models. In our benchmark study we use mixture discriminant analysis (Hastie and Tibshirani 1996). Each class conditional distribution is modeled

by a mixture of normal distributions with different means and equal covariance matrices. The mean vectors can be regarded as prototypes. For a test observation x the class with largest posterior probability is predicted. The distribution parameters are estimated by maximizing the likelihood via the EM algorithm.

2.2.3 Local kernel methods: RBF support vector machine

Segata and Blanzieri (2010b) mention SVMs with *local kernels* as an alternative to observation-specific support vector machines. Local kernels are defined by the property that their value becomes constant when the distance between two points tends to infinity. In contrast, the value of a global kernel can also be influenced by observations that are far away from each other (Smits and Jordaan 2002). A typical local kernel is the Gaussian or RBF kernel. Linear, polynomial and sigmoidal kernels are global kernels. Due to the form of the SVM discriminant function $f(x) = \sum_{n=1}^{N} \beta_n K(x, x_n)$ only support vectors near the trial point x have an influence on the predicted label of x if K is a local kernel. SVMs with local kernels can be regarded as a compromise between global and observation-specific approaches. In a purely observation-specific approach the coefficients β_n are estimated individually for each trial point x. When using local kernels the optimization problem is solved globally, but the pairwise similarities of training observations are taken into account.

SVMs with global kernels are usually regarded as global classification methods. Although the fitted model only depends on a subset of the data (the support vectors) this is only a result of the induced sparseness by the Hinge loss and not because any local structure is reflected. Also note that simply changing the Hinge loss to the L2 loss (least-squares SVMs) leads to a non-sparse solution.

There is still ongoing research on the properties and on the combination of local and global kernels in SVM classification and regression (Brailovsky et al 1999; Smits and Jordaan 2002; Zhu et al 2005; Segata and Blanzieri 2010b). In our benchmark study we consider SVMs with polynomial kernel where the degree of the polynomial is tuned and SVMs with Gaussian kernel.

Another possibility to localize SVMs via kernel functions is proposed by Gönen and Alpaydin (2008) by extending the multiple kernel approach. While in multiple kernel learning weighted sums of different kernels with constant weights are constructed, Gönen and Alpaydin (2008) allow the weights to depend on the location of the test observation x.

2.2.4 Recursive partitioning methods: Classification and regression trees and random forests

These approaches partition the predictor space into a set of, normally, hyper-rectangles and then fit a local classifier in each one. In case of classification trees (Breiman et al 1984) the local classifiers are constant functions in x and the hyper-rectangles are chosen such that they are as pure as possible with

respect to the class label. Partitioning is done in a greedy, recursive manner and purity is usually measured by the Gini index. Recently, there has been interest in using more complex local classifiers like linear discriminant analysis (Kim and Loh 2003), logistic regression (Landwehr et al 2005) or naive Bayes (Seewald et al 2001). This is mainly motivated by the fact that constant fits can lead to large and unstable trees. Zeileis et al (2008) provide a general framework to incorporate parametric models into recursive partitioning. Here, not the impurity, but the instability of the model parameter estimates is used as splitting criterion. In our benchmark study we consider classification trees (Breiman et al 1984). Moreover, we use random forests (Breiman 2001). Random forests are bagged decision trees and were developed to reduce the high variance of decision trees.

2.2.5 Discriminant-adaptive approaches: Localized logistic regression

Since all models we fit to the data are hardly ever correct there will be parts of the population where the fit of a model is quite accurate and others where it is not. As for discrimination an accurate fit is required especially near the class boundary. Therefore, discriminant-adaptive methods give higher weight to training observations near the class boundary in the training process. In our benchmark study we consider a localized form of logistic regression proposed by Hand and Vinciotti (2003). An iterative procedure is used to identify observations near the decision boundary: First, an unweighted logistic regression model is fitted. Based on the distances between the estimated class posterior probabilities and the threshold used for classification, observation weights can be calculated. A weighted model is fitted by introducing these weights into the log-likelihood. Subsequently, model fitting and calculation of weights is done in turn until a fixed number of iterations is reached. We use a rather simple weighting scheme. The k nearest neighbors of the decision boundary receive weight 1 and all other training observations are given weight 0.

In the literature some more approaches concerning the selection of class boundary patterns or border patterns are reported for example by Foody (1999); Lyhyaoui et al (1999); Chen and Burrell (2001); Shin and Cho (2002). In Foody (1999) for each training pattern the Mahalanobis distance to each class center is calculated. The class with the smallest Mahalanobis distance is the most likely class. Thus the "borderness" of an observation is measured as the difference between the two smallest Mahalanobis distances. Shin and Cho (2002) measure the proximity of a training point to the decision boundary in terms of the entropy among its k nearest neighbors.

2.2.6 Concluding remarks

Model assumptions in local classification are less stringent, as they need only be valid for subsets of the population. For this reason most of these methods exhibit increased flexibility in comparison with their global counterparts and are expected to give good results in case of complex problems with irregularly

shaped class boundaries. One special situation which is addressed by some local methods is multimodality of class conditional distributions. This is not too unlikely to occur since many classification problems are inverse problems. That is, for one value of the class variable there may exist multiple explanations that correspond to very different values of the predictors. Formally, we can say that local classification methods exhibit a lower bias, but also carry the risk of having a higher variance. But note that localization is only one way to obtain flexible classifiers because this effect can also be achieved by considering, e. g., higher order interactions of features in the model.

3 Benchmarking and experimental setup

Benchmarking experiments concern the evaluation of a set of candidate algorithms for one or a number of tasks in order to assess their absolute and relative performances. The choice of an appropriate performance measure has to be made by the user in advance. Here, we will assume that only one main performance criterion is of interest, but would like to point out that many problems in machine learning could naturally be considered as multi-criteria ones (e. g., consider trade-offs between predictive power, time to fit a model and number of features used). Of special interest is the generation of a ranking of these candidate algorithms w.r.t. their individual performances and sometimes the identification of the best performing algorithm(s). If pseudo-random computer experiments are used, these performance values will be stochastic and statistical significance tests should be employed to compare the candidates. Still, many papers compare the algorithm(s) under consideration only to few alternatives (and these are often of the same type, e. g., only variants of neural networks) and on few data sets whose selection is never made clear. Testing for significant differences is frequently ignored.

3.1 General framework

We employ the framework for benchmark experiments by Hothorn et al (2005) to compare the discriminating power of different classification algorithms. By applying a resampling strategy like bootstrapping or subsampling (Simon 2007) one independently generates training sets from a given data set, uses a machine learning algorithm to fit models on these, predicts the out-of-bag test samples and assesses their performance according to an appropriate measure. This generates a finite sequence of performance values for every algorithm, which now can be compared by standard statistical inference methodology. We make the following choices: We use subsampling (sometimes called Monte Carlo cross-validation) with 100 iterations and 4/5 splits to generate training and test sets. The dependence structure of ordinary cross-validation is problematic for subsequent significance tests as shown by Nadeau and Bengio (2003). Bootstrapping on the other hand can produce biased results if combined with tuning due to its sampling with repetitions. Binder and Schumacher

(2008) demonstrate this in an extensive study concerning boosting. Training and test sets are equal for each data set for all learning algorithms (paired design) to reduce variance. For hyperparameter tuning we employ proper nested resampling with 5-fold cross-validation and grid search in the inner loop. As performance measure we only consider the misclassification error.

3.2 Classification methods

We employed eleven classification methods in our benchmark study, among them eight local methods of four different types and three global methods. As global methods we apply two linear methods, linear discriminant analysis (*lda*) and multinomial logistic regression (*mnr*), and as a more flexible method a polynomial support vector machine (*p-svm*) where the degree of the polynomial is tuned. Moreover, we use localized logistic regression *llr* which is a discriminant-adaptive method. The nearest neighbors of the decision boundary are used for fitting the model. Their number (parameter k) is tuned. As recursive partitioning methods we apply classification trees (*rp*) and random forests (*rf*). From the group of prototype methods we consider mixture discriminant analysis (*mda*) as well as learning vector quantization (*lvq1*). In *mda* each class is modeled as a mixture of subclasses whose number has to be specified. In order to reduce tuning parameters we assume that the this number is equal for all classes. In *lvq1* each class is represented by multiple prototypes. Their total number (size) as well as the learning rate (alpha) are tuned. From the group of observation-specific methods we consider k nearest neighbors (*knn*) and localized linear discriminant analysis (*llda*). As local kernel method we use the RBF support vector machine (*r-svm*). In *llda* we use the k nearest neighbors of each trial point to fit the local *lda* models. The parameter k is tuned. Localized logistic regression and support vector machines can only handle binary classification problems. For multiclass-classification the all-pairs approach is used. Table 1 gives an overview of the employed classification methods and their tuning parameters.

3.3 Data

In our study we consider 49 data sets, both artificial and real-world. Table 2 provides a brief overview together with basic data characteristics. By far the most data sets are taken from the UCI machine learning repository (Frank et al 2010). Moreover, the *circle*, *cuboids*, *ringnorm*, *twonorm*, *threenorm*, *spirals*, *waveform* and *xor* data are taken from the *mlbench* R-package (Leisch and Dimitriadou 2010). The *crystal* and *encoded* data sets are described in more detail in Szepannek et al (2008), the *mixture* data set is taken from Bishop (2006). The *hvdata* set is an artificial data set described in Hand and Vinciotti (2003) and the *orange* and South-African-heart-disease (*SAheart*) data are taken from Hastie et al (2009). The *crabs* data are available in the *MASS*

Table 1 Survey of classification methods and tuned parameters.

type	method	R-package	parameters	range
global	lda	*MASS* (Venables and Ripley 2002)		
	mnr	*nnet* (Venables and Ripley 2002)		
	p-svm	*e1071* (Dimitriadou et al 2010)	cost	$2^{-5}, 2^{-4}, \ldots, 2^4, 2^5$
			degree	$1, 2, 3$
observation-specific	knn	*kknn* (Schliep and Hechenbichler 2010)	k	1,2,3,10,25,50
	llda	*klaR* (Weihs et al 2005)	k	$[(0.1, 0.2, \ldots, 0.5) \cdot \#\text{obs}]$
local kernels	r-svm	*e1071* (Dimitriadou et al 2010)	cost	$2^{-5}, 2^{-4}, \ldots, 2^5$
			gamma	$2^{-5}, 2^{-4}, \ldots, 2^5$
prototype	mda	*mda* (S original by Hastie et al 2009)	subclasses	$1, 2, \ldots, 5$
	lvq1	*class* (Venables and Ripley 2002)	size	$(1, 2, \ldots, 5) \cdot \#\text{class}$
			alpha	0.01, 0.03, 0.05
recursive partitioning	rp	*rpart* (Therneau et al 2010)	cp	$0.005, 0.01, 0.015, \ldots, 0.05$
			minsplit	$10, 12, 14, \ldots, 30$
	rf	*randomForest* (Liaw and Wiener 2002)	mtry	$[(0.4, 0.6, \ldots, 1.6) \cdot \sqrt{\dim}]$
			ntree	100, 500, 1000, 2000
discriminant-adaptive	llr		k	$[(0.2, 0.3, \ldots, 0.5) \cdot \frac{2\#\text{obs}}{\#\text{class}})]$

R-package (Venables and Ripley 2002) and the *texture* data are taken from the ELENA data base (`http://www.dice.ucl.ac.be/mlg/?page=Elena`).

Only little preprocessing was done. Since handling of missing values is beyond the scope of this paper observations containing them were omitted for the sake of simplicity. Therefore, the number of observations given in Table 2 may be lower than the number reported on the UCI web site.

Since we want the differences between methods to be revealed we included problems that are known to be easily linearly separable like *iris* and *wine*, problems with quadratic decision boundaries like *balance-scale* or *circle*, as well as multimodal problems (*subclasses* and *subclasses2*). Moreover, we chose problems where some local classification methods are reported to yield good results, for example in Hastie and Tibshirani (1996) *mda* is reported to exhibit superior performance compared to *lda*, quadratic discriminant analysis and classification trees on the *waveform* data set. The *hvdata* set (Hand and Vinciotti 2003) is especially designed for *llr*. Originally, it is a two dimensional problem where the contour lines of the class posterior distribution are not parallel which leads to a poor fit of a global logistic regression model. We use a modified version where the contour lines are not placed symmetrically around the true decision boundary and which has been augmented to six dimensions.

We are aware of all the problems connected with using a large amount of data from repositories like UCI, e. g., see Soares (2003) and Saitta and Neri (1998) for a discussion of this topic. On the other hand we want to point out the following facts: Not many alternatives exist if one wants to assess classification

algorithms under various difficult and "realistic" conditions. And we are not interested in proving that one of our considered algorithms is "the best one for all tasks". Rather we want to study advantages and drawbacks of algorithmic variants and groups of algorithms.

Table 2 Survey of the 49 data sets used in the benchmark study: Number of observations, number of classes, dimensionality, number of numeric and number of categorical predictors

data set	#obs	#class	dim	#num	#categ	data set	#obs	#class	dim	#num	#categ
balance-scale	625	3	4	4	0	orange	1000	2	10	10	0
breast-cancer	277	2	9	0	9	page-blocks	5473	5	10	10	0
breast-w	683	2	9	0	9	pima	768	2	8	8	0
car	1728	4	6	0	6	ringnorm	1000	2	4	4	0
circle	1000	2	4	4	0	SAheart	462	2	9	8	1
cmc	1473	3	9	2	7	segment	2310	7	16	16	0
crabs	200	2	5	5	0	sonar	208	2	60	60	0
credit-a	653	2	15	6	9	soybean	562	15	35	0	35
credit-g	1000	2	20	7	13	spect	267	2	22	0	22
crystal	2746	3	37	37	0	spirals	1000	2	2	2	0
cuboids	1002	4	3	3	0	splice	3190	3	60	0	60
dermatology	358	6	34	1	33	subclasses	300	2	2	2	0
encoded	794	2	6	6	0	subclasses2	990	3	5	5	0
glass	214	6	9	9	0	tae	151	3	5	3	2
haberman	306	2	3	2	1	texture	5500	11	40	40	0
heart-c	296	2	13	6	7	threenorm	1000	2	4	4	0
heart-statlog	270	2	13	13	0	tic-tac-toe	958	2	9	0	9
hvdata	1000	2	6	6	0	twonorm	1000	2	4	4	0
ionosphere	351	2	33	33	0	vehicle	846	4	18	18	0
iris	150	3	4	4	0	vote	232	2	16	0	16
kdd-synth-control	600	6	60	60	0	vowel	990	11	9	9	0
liver-disorders	345	2	6	6	0	waveform	1000	3	21	21	0
mfeat-fourier	2000	10	76	76	0	wine	178	3	13	13	0
mixture	200	2	2	2	0	xor	1000	8	4	4	0
optdigits	5620	10	62	62	0						

3.4 Testing generalization performance

Different parametric and non-parametric alternatives exist to test the performance value sets for differences in location. We use a linear mixed effects model proposed by Eugster et al (2008). The observed performance value p_{ij} of a learner i on test set j is modeled as

$$p_{ij} = \gamma_i + \delta_j + \varepsilon_{ij} \ .$$

The above equation includes a fixed effect γ_i for the underlying true performance of learner i and a random effect δ_j for the test set. The latter blocking factor encodes the fact that all algorithms share the same training and test samples. As we are interested in comparing all algorithms with each other, Tukey contrasts are used to test for significant pairwise differences between the γ_i and pairwise simultaneous confidence intervals are calculated. We always use a family-wise significance level of $\alpha = 0.05$ in the following analysis.

3.5 Preference orderings

Using the mixed effects model of the previous section we produce an order relation $\prec$ on the learning algorithms for each data set where $a \prec b$ means that the performance of learner a is significantly smaller than the one of b. We visualize these order relations as graphs, see Fig. 1 for four examples. In Eugster et al (2008) it is proposed to further transform the ordering to a ranking between equivalence classes of algorithms if possible. In many of our results this is not valid, e. g., consider the result for the vowel data set in Fig. 1: This relation is not negative transitive, as *r-svm* $\prec$ *knn* and *knn* $\prec$ *p-svm* both do not hold, but *r-svm* $\prec$ *p-svm* does. Topologically sorting this graph results in *knn* $\approx$ *r-svm* $\prec$ *p-svm* $\prec \ldots$ which implies that there is a significant difference between *knn* and *p-svm* which is not true. Instead, we propose to always directly work with the order relation and to only derive properties which are independent of any additional assumptions about the relation structure:

For a data set D we call an algorithm a

- dominated, if another algorithm b exists so that $b \prec a$,
- (relatively) optimal, if a is not dominated,
- (relatively) worst, if a is dominated, but does not dominate any other algorithm.

3.6 Software and execution of experiments

All our experiments are conducted within the statistical programming language R (R Development Core Team 2010). A large number of different packages are needed to provide the implementations of all used learners, we have used the *mlr* package (Bischl 2010) to interface these. This package also provides most standard resampling techniques of machine learning, any nested combination of them and the possibility to perform hyperparameter tuning. For subsequent analysis the benchmark results are passed to the *benchmark* package (Eugster et al 2008). Although it contains many tools for visualization, these are not really feasible here due to the large number of data sets and *benchmark*'s current requirement to create rankings of equivalence classes. We mainly use it to perform significance tests regarding differences in performance.

3.7 The problem of algorithmic failures

If one conducts experiments of a larger scale some algorithmic runs often fail (e. g., due to numerical problems). The real problem stems from the fact that these failures might only occur on a subset of resampling iterations and one does not want to discard the results of the whole experiment. Simply omitting these "missing values" or imputing them by, e. g., the mean of the remaining performance values favors the algorithm unfairly: The resampled data where it failed might have been more difficult and the competing algorithms might

have scored relatively badly on them, but producing no result at all should be considered even worse. On the other hand, imputing them by the worst performance value possible will likely result in a multimodal distribution and hindering subsequent tests.

We decided to use an ad hoc rule here: If the algorithm fails on more than 20% of all resampling iterations we consider its behavior as too unreliable for the current data set and therefore set all its performance values to the worst possible - those achieved by the data-independent rule on the 100 subsamples. If it is less than 20% we impute by sampling from an estimated normal distribution of the remaining values in order to not violate the normality assumption of the mixed effects model in Sect. 3.4. In any case we report the percentage of failures in the node labels of the preference relations (see Fig. 1).

4 Results and discussion

In this section we present all results obtained from our benchmark experiments and discuss their relevance w.r.t. the questions asked in the introduction.

4.1 Basic results, preference orderings and comparison to selected results from the literature

First, Table 3 presents the mean error rates of all considered classifiers on all data sets. From the underlying distributions we have calculated 49 learner preference orderings – one for each data set. Because of space limitations we cannot present all of them graphically in this article, but instead display exemplary relations for four data sets in Fig. 1.

Looking at the preference orderings we make the following observations: Contrary to Hastie and Tibshirani (1996) we arrive at different results concerning the *waveform* data set. They report superior performance of *mda* compared to *lda* and classification trees. Our results indicate that ordinary multinomial logistic regression significantly outperforms *mda*. The remaining linear methods *lda* and *llr* perform equally well. The *hvdata* set is especially constructed to meet the special conditions *llr* is designed for, that is, a linear decision boundary, but contour lines of the class posterior distribution that are not parallel. Indeed, the fit of *llr* in terms of deviance is better than the fit of *mnr*, and *llr* achieves the lowest error rate. However, this is not significantly lower than the error rates of almost all other methods that also lie near the noise level of about 26%.

Table 3 Mean error rates obtained on the 49 data sets for all eleven classifiers in the benchmark study.

	knn	lda	llda	llr	lvq1	mda	mnr	p-svm	r-svm	rf	rp
balance-scale	0.11	0.13	0.08	0.09	0.15	0.10	0.11	0.01	0.04	0.11	0.22
breast-cancer	0.25	0.41	0.41	0.31	0.41	0.41	0.29	0.26	0.24	0.25	0.26
breast-w	0.03	0.04	0.46	0.08	0.04	0.46	0.08	0.04	0.03	0.46	0.05
car	0.06	0.11	0.06	0.39	0.46	0.46	0.07	0.01	0.01	0.02	0.05
circle	0.12	0.51	0.25	0.43	0.36	0.28	0.51	0.02	0.03	0.10	0.18
cmc	0.50	0.48	0.46	0.49	0.50	0.65	0.50	0.44	0.45	0.46	0.45
crabs	0.06	0.00	0.00	0.00	0.29	0.00	0.00	0.00	0.00	0.11	0.13
credit-a	0.14	0.14	0.50	0.15	0.50	0.49	0.14	0.14	0.14	0.13	0.15
credit-g	0.25	0.25	0.42	0.26	0.42	0.42	0.25	0.24	0.23	0.24	0.27
crystal	0.21	0.21	0.19	0.20	0.30	0.21	0.21	0.19	0.19	0.20	0.26
cuboids	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
dermatology	0.04	0.80	0.80	0.18	0.23	0.80	0.08	0.04	0.04	0.02	0.06
encoded	0.12	0.13	0.10	0.10	0.13	0.10	0.09	0.10	0.10	0.09	0.10
glass	0.30	0.39	0.30	0.40	0.36	0.32	0.37	0.31	0.32	0.22	0.31
haberman	0.26	0.25	0.28	0.27	0.25	0.39	0.26	0.27	0.28	0.27	0.27
heart-c	0.17	0.16	0.50	0.17	0.50	0.50	0.16	0.17	0.17	0.17	0.22
heart-statlog	0.17	0.17	0.18	0.17	0.36	0.16	0.16	0.16	0.16	0.17	0.22
hvdata	0.27	0.27	0.27	0.26	0.29	0.27	0.26	0.27	0.26	0.28	0.28
ionosphere	0.14	0.13	0.46	0.13	0.15	0.12	0.13	0.10	0.05	0.07	0.12
iris	0.05	0.02	0.03	0.03	0.04	0.02	0.04	0.05	0.03	0.05	0.05
kdd-synth	0.03	0.03	0.03	0.08	0.03	0.01	0.17	0.01	0.01	0.01	0.12
liver-disorders	0.35	0.33	0.29	0.32	0.36	0.34	0.32	0.29	0.29	0.26	0.34
mfeat-fourier	0.19	0.19	0.16	0.20	0.20	0.17	0.22	0.16	0.18	0.17	0.27
mixture	0.26	0.26	0.25	0.30	0.26	0.26	0.27	0.26	0.26	0.27	0.28
optdigits	0.02	0.05	0.90	0.03	0.05	0.04	0.04	0.01	0.07	0.02	0.20
orange	0.27	0.49	0.33	0.48	0.31	0.24	0.49	0.06	0.06	0.10	0.16
page-blocks	0.03	0.05	0.04	0.04	0.09	0.05	0.04	0.03	0.03	0.03	0.03
pima	0.25	0.23	0.23	0.23	0.30	0.23	0.23	0.24	0.23	0.24	0.25
ringnorm	0.19	0.37	0.20	0.34	0.28	0.22	0.37	0.17	0.17	0.19	0.21
SAheart	0.30	0.27	0.29	0.29	0.45	0.45	0.27	0.28	0.28	0.31	0.30
segment	0.09	0.16	0.14	0.10	0.21	0.16	0.11	0.09	0.08	0.06	0.12
sonar	0.15	0.26	0.25	0.29	0.31	0.24	0.26	0.15	0.13	0.17	0.28
soybean	0.11	0.89	0.90	0.19	0.12	0.90	0.10	0.08	0.08	0.08	0.13
spect	0.19	0.18	0.20	0.21	0.19	0.33	0.19	0.18	0.18	0.17	0.18
spirals	0.00	0.34	0.35	0.34	0.40	0.30	0.34	0.34	0.00	0.02	0.05
splice	0.13	0.61	0.61	0.09	0.61	0.61	0.09	0.05	0.04	0.03	0.05
subclasses	0.09	0.34	0.09	0.39	0.10	0.08	0.33	0.09	0.09	0.09	0.08
subclasses2	0.28	0.42	0.29	0.39	0.34	0.28	0.42	0.29	0.28	0.30	0.34
tae	0.43	0.48	0.67	0.54	0.62	0.66	0.49	0.47	0.48	0.39	0.51
texture	0.01	0.00	0.91	0.00	0.08	0.91	0.01	0.00	0.00	0.02	0.15
threenorm	0.12	0.15	0.12	0.15	0.13	0.12	0.15	0.12	0.11	0.12	0.15
tic-tac-toe	0.02	0.02	0.02	0.02	0.45	0.45	0.02	0.00	0.00	0.01	0.07
twonorm	0.03	0.02	0.02	0.03	0.03	0.02	0.02	0.03	0.03	0.03	0.06
vehicle	0.29	0.22	0.16	0.22	0.45	0.18	0.21	0.15	0.16	0.25	0.31
vote	0.08	0.03	0.50	0.07	0.49	0.49	0.07	0.04	0.03	0.04	0.03
vowel	0.02	0.46	0.12	0.29	0.36	0.19	0.42	0.03	0.01	0.05	0.36
waveform	0.16	0.15	0.16	0.15	0.16	0.15	0.14	0.14	0.14	0.14	0.26
wine	0.03	0.01	0.03	0.05	0.30	0.02	0.06	0.02	0.02	0.02	0.10
xor	0.11	0.81	0.11	0.68	0.21	0.06	0.82	0.08	0.08	0.01	0.04

4.2 Dominance relations between algorithms and "Do local variants outperform global counterparts?"

Since we cannot address all preference relations in detail we aggregate our results in the following way. In Table 4 it is shown how often one algorithm dominates a competing candidate and in Table 5 how often each algorithm is (relatively) optimal or worst for a data set, according to our terminology from Sect. 3.5. Quite a few useful observations can already be made: Both the global and local variants of the SVM perform very well, as does the random forest. Interestingly, the SVMs and the random forest seem to compliment each other in the sense that if one does not perform well on a given data set, the other method is a good option. *lvq1* is among the worst methods for nearly half of all considered data sets. Also, while the local versions of *lda* (*llda* and *mda*) are able to significantly outperform *lda* in quite a few instances, their performance is significantly worse about the same number of times. The same conclusion

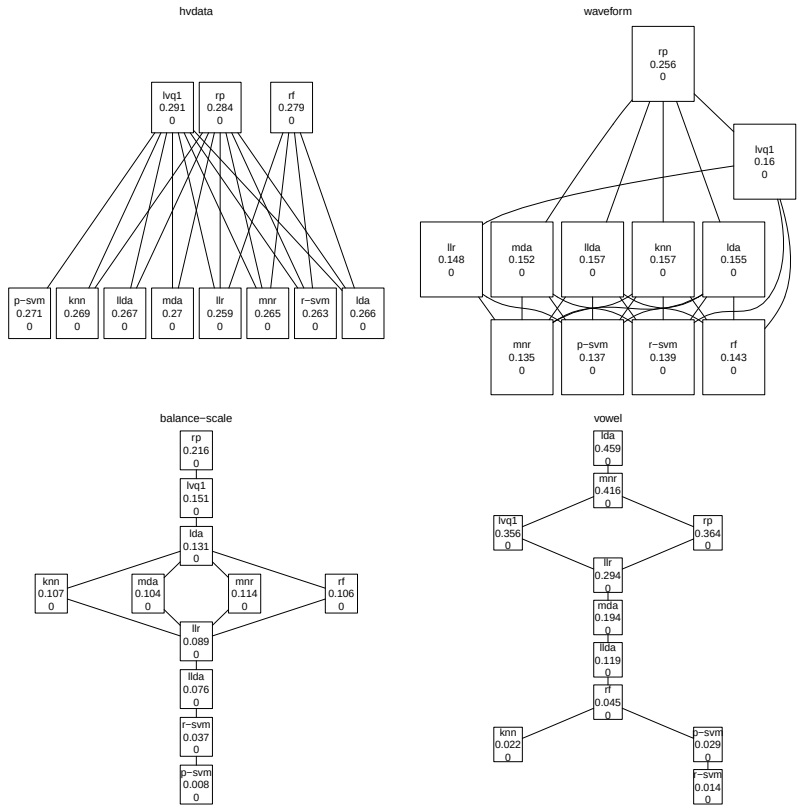

Fig. 1 Exemplary preference orderings for four data sets. Nodes refer to learners and we also show their mean misclassification error for reference. A learner A significantly outperforms learner B if only an upward leading path of edges connects A with B.

can be drawn for *mnr* and its local counterpart *llr*. This means one cannot always simply use the locals variants instead of the base method and trust in achieving a better or equal result. While their larger flexibility offers indeed the possibility for significant improvements in some scenarios, they also carry the risk of worsening the result because of their increased variance (Schiffner et al 2012).

4.3 On which data sets are local methods beneficial?

Again, using the concept of dominating algorithms from Sect. 3.5, we will now structure the data sets into different subgroups depending on which algorithms perform well on them. For subsequent analysis in further experiments we think it is worthwhile to have some knowledge available on which data sets local methods might actually perform well and which are therefore suitable first candidates for further comparison experiments.

Table 4 Counts of how often one algorithm (row) dominates another one (column) out of the 49 data sets.

	knn	lda	llda	mda	lvq1	p-svm	r-svm	rf	rp	mnr	llr
knn	0	22	20	24	37	2	2	6	28	19	24
lda	10	0	12	13	26	1	1	7	21	7	11
llda	9	19	0	15	26	0	1	5	18	17	19
mda	9	15	7	0	28	3	3	4	18	15	16
lvq1	0	10	8	8	0	1	1	2	9	11	12
p-svm	23	30	26	31	42	0	3	15	33	30	33
r-svm	24	31	26	31	42	3	0	18	35	31	33
rf	24	30	26	29	40	11	11	0	35	29	29
rp	8	18	19	23	28	2	1	2	0	17	18
llr	11	15	15	21	29	1	1	8	20	10	0
mnr	11	11	16	22	30	0	0	6	19	0	11

Table 5 Counts of how often one algorithm is optimal and worst out of the 49 data sets.

	knn	lda	llda	llr	lvq1	mda	mnr	p-svm	r-svm	rf	rp
optimal	18	14	16	12	4	14	13	35	36	29	8
worst	2	13	13	6	24	17	8	1	0	3	9

We define our data set groups in the following way: If one of the models which construct linear decision boundaries (*lda*, *mnr*, *llr* or a *p-svm* where a degree of 1 is selected in tuning most of the time) is (relatively) optimal, we call this data set "linear". Quite a large number of 19 data sets fall into this category, namely:

crabs, *cuboids*, *encoded*, *haberman*, *heart-c*, *heart-statlog*, *hvdata*, *iris*, *kdd-synthetic-control*, *mixture*, *optdigits*, *pima*, *SAheart*, *spect*, *texture*, *twonorm*, *vote*, *waveform*, *wine*.

If only local classifiers (including *llr*) are optimal we call this data set "local", as an algorithm of this type is required to achieve superior results. We can identify 12 such data sets:

credit-a, *dermatology*, *glass*, *ionosphere*, *liver-disorders*, *page-blocks*, *segment*, *spirals*, *splice*, *tae*, *vowel*, *xor*.

Using the symmetric difference as a distance measure between the algorithm orderings per data set, we apply a hierarchical clustering approach (complete linkage), see Fig. 2. We observe that the more difficult data sets, which either require local models or a flexible global one, cluster in the left part of the plot (in the latter case the *p-svm* is mainly the only optimal global model and a higher degree for the polynomial is selected during tuning). In the middle of the plot a large cluster of mostly linear data sets can be located.

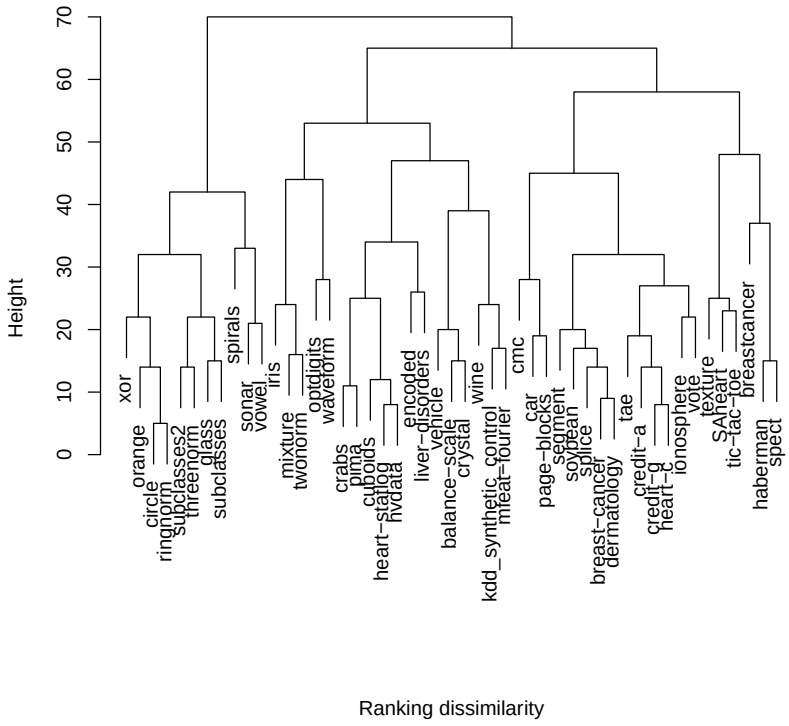

Fig. 2 Hierarchical clustering (complete linkage) of the 49 data sets based on the symmetric difference of the algorithm ordering. Adjacent data sets have a similar performance ordering of algorithms.

4.4 Do local methods or different types of local methods perform in a similar way?

In this section we will take a look at whether local methods, in general or different types, actually perform well on the same data sets. Based on the mean ranks of the eleven classifiers per data set we conduct a hierarchical clustering (complete linkage). The resulting dendrogram is shown in Fig. 3.

On the left hand side a cluster is formed by four very flexible methods (*r-svm*, *p-svm*, *rf* and *knn*). Interestingly, *llda* is not contained in this group, although it is a combination of k nearest neighbors and linear discriminant analysis. On the right hand side there is a cluster consisting of linear methods (*mnr*, *llr* and *lda*). The most similar methods with respect to their performance are the discriminant-adaptive version of logistic regression, *llr*, and its counterpart *mnr*. This is not surprising because both methods produce linear

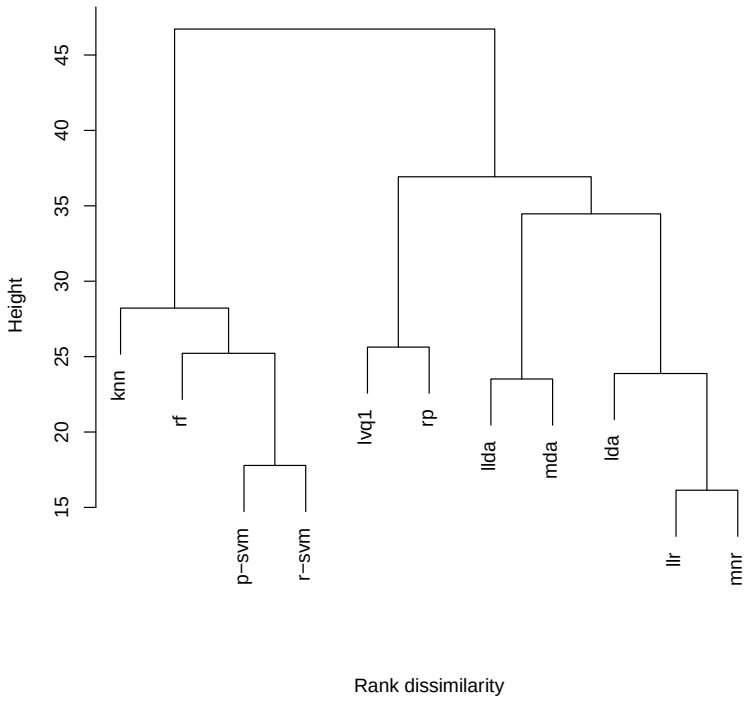

Fig. 3 Hierarchical clustering (complete linkage) of classification methods. The dendrogram was calculated based on the distances between the mean ranks of the classifiers per data set.

decision boundaries and the only effect of *llr* is a translation and rotation of the separating hyperplanes.

In the introduction we raised the questions if local methods or different types of local methods show similar behavior. The different types of local methods are not reflected by the dendrogram. Besides the flexibility of the methods the underlying base method seems to have a large impact on the performance. As already mentioned *mnr* and *llr* that are based on logistic regression behave very similar. The same holds for the SVM variants *p-svm* and *r-svm*. Moreover, *llda* and *mda* which are both localized variants of *lda* form a cluster and, additionally, the group containing these two methods is close to the cluster that consists of linear methods and thus includes *lda*.

4.5 Identification of an optimal, reduced set of algorithms

The general problem of reducing the burden of model selection is a still prevalent topic in machine learning. For every algorithm there might be a special data situation where it performs best, but the resulting option to try out every available learner for a given task is highly impractical. Even worse, algorithms have to be tuned for optimal performance and other, uncertain choices (for example in preprocessing) can dramatically influence the outcome, too. We therefore think that it is useful to identify few algorithms, which complement each other, e. g., which perform well if the other ones fail and vice versa. Therefore, we now construct a minimal set of classifiers S so that for each data set D we have at least one algorithm from the top scoring group on D in S. In our case, by considering only *r-svm* and *rf* one would have achieved optimal results for all but two data sets. These are the *balance-scale* data set, where only the *p-svm* scores best and the *haberman* data set, where *knn*, *lda*, *llr*, *lvq1*, *mnr* and *rp* all score best. Such a reduced algorithm set now provides a useful starting point for further experiments where computation time is limited and one cannot try out every available learner.

5 Conclusion and outlook

We present a benchmark study for eleven global and local classification methods on a large number of artificial and real-world data sets. Significance tests are used to determine whether there are substantial differences in predictive performance between algorithms. We define the notions of a classifier being relatively optimal or worst for a given task and provide a novel way to select a minimal, optimal set of complementary methods. Based on the performance of the classification methods we determine different types of classification problems. It turns out that for many data sets in the benchmark study simple linear methods are sufficient to achieve satisfactory results. We identify twelve data sets where only local methods yield optimal performance. Among them are many real-world data sets. The two artificial data sets in this group, *xor* and *spirals*, are rather complicated decision problems, meeting our expectation that local methods perform well on complex tasks. It is difficult to make general statements about the performance of local classification methods. This has been clear in advance since we consider distinct types of local methods. Moreover, even local approaches of the same type are found to show different behavior. What seems to be more important is the general flexibility of the learner and the base method from which the local variant is derived. Local versions do not always perform at least as good as their global counterparts. Although in some instances significant improvements can be achieved, on other occasions worse results are obtained. We suspect that this is due to overfitting in cases with noise where the underlying true model is quite simple. Gaining deeper insight into the underlying bias-variance trade-offs could be a subject for future work. We could not identify any (practically useful)

relations between data characteristics and performance differences between various algorithms or groups. The only criterion we used for evaluation is the misclassification rate. Further work may also take into account computation time and/or interpretability of the resulting models.

References

Alpaydin E, Jordan M (1996) Local linear perceptrons for classification. IEEE Transactions on Neural Networks 7(3):788–792

Atkeson C, Moore A, Schaal S (1997) Locally weighted learning. Artificial Intelligence Review 11:11–73

Binder H, Schumacher M (2008) Adapting prediction error estimates for biased complexity selection in high-dimensional bootstrap samples. Statistical Applications in Genetics and Molecular Biology 7(1):Article 12

Bischl B (2010) mlr: Machine learning in R. URL `http://mlr.r-forge.r-project.org/`

Bishop CM (2006) Pattern Recognition and Machine Learning. Information Science and Statistics, Springer, New York

Blanzieri E, Melgani F (2006) An adaptive SVM nearest neighbor classifier for remotely sensed imagery. In: Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS-2006), pp 3931–3934

Brailovsky V, Barzilay O, Shahave R (1999) On global, local, mixed and neighborhood kernels for support vector machines. Pattern Recognition Letters 20:1183–1190

Breiman L (2001) Random forests. Machine Learning 45(1):5–32

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and Regression Trees. Wadsworth, Belmont

Chen D, Burrell P (2001) On decision-boundary-based approaches for structure selection of multilayered neural nets. In: Proceedings of the International Conferences on Info-tech and Info-net 2001, ICII 2001, vol 3, pp 486–490

Cheng H, Tan PN, Jin R (2010) Efficient algorithm for localized support vector machine. IEEE Transactions on Knowledge and Data Engineering 22(4):537–549

Clarke B, Fokoué E, Zhang H (2009) Principles and Theory for Data Mining and Machine Learning. Springer Series in Statistics, Springer, New York

Czogiel I, Luebke K, Zentgraf M, Weihs C (2007) Localized linear discriminant analysis. In: Decker R, Lenz H (eds) Advances in Data Analysis, Springer, Berlin Heidelberg, Studies in Classification, Data Analysis, and Knowledge Organization, vol 34, pp 133–140

Devroye L, Györfi L, Lugosi G (1996) A Probabilistic Theory of Pattern Recognition. Applications of Mathematics, Springer, New York

Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A (2010) e1071: Misc functions of the Department of Statistics (e1071), TU Wien

Eugster M, Hothorn T, Leisch F (2008) Exploratory and inferential analysis of benchmark experiments. Tech. Rep. 30, Department of Statistics, LMU München

Fix E, Hodges J (1951) Discriminatory analysis – nonparametric discrimination: Consistency properties. Report 4, U.S. Airforce School of Aviation Medicine, Randolph Field, Texas

Foody G (1999) The significance of border training patterns in classification by a feedforward neural network using back propagation learning. International Journal of Remote Sensing 20(18):3549–3562

Frank A, Asuncion A, University of California, Irvine, School of Information and Computer Sciences (2010) UCI machine learning repository

Gönen M, Alpaydin E (2008) Localized multiple kernel learning. In: Proceedings of the 25th International Conference on Machine learning (ICML '08), pp 352–359

Hand DJ, Vinciotti V (2003) Local versus global models for classification problems: Fitting models where it matters. The American Statistician 57(2):124–131

Hastie T, Tibshirani R (1996) Discriminant adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(6):607–616

Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Springer Series in Statistics, Springer, New York
S original by Hastie T, Tibshirani R, Original R port by Leisch F, Hornik K, Ripley B (2009) mda: Mixture and flexible discriminant analysis
Hothorn T, Leisch F, Zeileis A, Hornik K (2005) The design and analysis of benchmark experiments. Journal of Computational and Graphical Statistics 14:675–699
Kim H, Loh WY (2003) Classification trees with bivariate linear discriminant node models. Journal of Computational and Graphical Statistics 12:512–530
Kohonen T (1989) Self-Organization and Associative Memory. Springer, Berlin
Kotsiantis S, Pintelas P (2004) Local boosting of weak classifiers. In: Proceedings of the IEEE 4th International Conference on Intelligent Systems Design and Applications (ISDA 2004), pp 175–180
Landwehr N, Hall M, Frank E (2005) Logistic model trees. Machine Learning 59:161–205
Leisch F, Dimitriadou E (2010) mlbench: Machine learning benchmark problems. R package version 2.0-0
Liaw A, Wiener M (2002) Classification and regression by randomForest. R News 2(3):18–22
Lyhyaoui A, Martinez M, Mora I, Vaquez M, Sancho J, Figueiras-Vidal A (1999) Sample selection via clustering to construct support vector-like classifiers. IEEE Transactioons on Neural Networks 10(6):1474–1481
Nadeau C, Bengio Y (2003) Inference for the generalization error. Machine Learning 52(3):239–281
R Development Core Team (2010) R: A language and environment for statistical computing
Saitta L, Neri F (1998) Learning in the ''real world''. Machine Learning 30(2-3):133–163
Schiffner J, Bischl B, Weihs C (2012) Bias-variance analysis of local classification methods. In: Gaul WA, Geyer-Schulz A, Schmidt-Thieme L, Kunze J (eds) Challenges at the Interface of Data Analysis, Computer Science, and Optimization, Springer, Berlin Heidelberg, Studies in Classification, Data Analysis, and Knowledge Organization, vol 43, pp 49–57
Schliep K, Hechenbichler K (2010) kknn: Weighted k-nearest neighbors
Seewald AK, Petrak J, Widmer G (2001) Hybrid decision tree learners with alternative leaf classifiers: An empirical study. In: Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, pp 407–411
Segata N, Blanzieri E (2010a) Fast and scalable local kernel machines. Journal of Machine Learning Research 11:1883–1926
Segata N, Blanzieri E (2010b) Operators for transforming kernels into quasi-local kernels that improve SVM accuracy. Journal of Intelligent Information Systems pp 1–32
Shen Z (2006) Classification: Distance-based algorithms. In: Berry M, Browne M (eds) Lecture Notes in Data Mining, World Scientific Publishing, Singapore
Shin H, Cho S (2002) Pattern selection for support vector classifiers. In: Yin H, Allinson N, Freeman R, Keane J, Hubbard S (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2002, Springer, Berlin Heidelberg, Lecture Notes in Computer Science, vol 2412, pp 97–103
Simon R (2007) Fundamentals of Data Mining in Genomics and Proteomics, Springer, US, chap Resampling Strategies for Model Assessment and Selection, pp 173–186
Smits G, Jordaan E (2002) Improved SVM regression using mixtures of kernels. In: Proceedings of the International Joint Conference on Neural Networks, vol 3, pp 2785–2790
Soares C (2003) Is the UCI repository useful for data mining? Progress in Artificial Intelligence 2902:209–223
Sugiyama M (2007) Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. Journal of Machine Learning Research 8:1027–1061
Szepannek G, Schiffner J, Wilson J, Weihs C (2008) Local modelling in classification. In: Perner P (ed) Advances in Data Mining. Medical Applications, E-Commerce, Marketing, and Theoretical Aspects, Springer, Berlin Heidelberg, Lecture Notes in Computer Science, vol 5077, pp 153–164
Therneau T, Atkinson B, R port by Ripley B (2010) rpart: Recursive partitioning
Tutz G, Binder H (2005) Localized classification. Statistics and Computing 15:155–166
Venables WN, Ripley BD (2002) Modern Applied Statistics with S. Springer, New York

Weihs C, Ligges U, Luebke K, Raabe N (2005) klaR analyzing german business cycles. In: Baier D, Decker R, Schmidt-Thieme L (eds) Data Analysis and Decision Support, Springer, Berlin Heidelberg, Studies in Classification, Data Analysis, and Knowledge Organization, vol 30, pp 335–343

Zeileis A, Hothorn T, Hornik K (2008) Model-based recursive partitioning. Journal of Computational and Graphical Statistics 17(2):492–514

Zhang CX, Zhang JS (2008) A local boosting algorithm for solving classification problems. Computational Statistics & Data Analysis 52:1928–1941

Zhang H, Berg A, Maire M, Malik J (2006) SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol 2

Zhu Yf, Tian Lf, Mao Zy, Wei LfT (2005) Mixtures of kernels for SVM modeling. In: Wang L, Chen K, Ong Y (eds) Advances in Natural Computation, Springer, Berlin Heidelberg, Lecture Notes in Computer Science, vol 3610, pp 601–607