Comparison of Classifiers Applied to Confocal Scanning Laser Ophthalmoscopy Data

W. Adler, A. Peters, B. Lausen
Department of Medical Informatics, Biometry and Epidemiology, Friedrich-Alexander-University Erlangen-Nuremberg, Erlangen, Germany

Summary

Objectives: Comparison of classification methods using data of one clinical study. The tuning of hyperparameters is assessed as part of the methods by nested-loop cross-validation.

Methods: We assess the ability of 18 statistical and machine learning classifiers to detect glaucoma. The training data set is one case-control study consisting of confocal scanning laser ophthalmoscopy measurement values from 98 glaucoma patients and 98 healthy controls. We compare bootstrap estimates of the classification error by the Wilcoxon signed rank test and box-plots of a bootstrap distribution of the estimate.

Results: The comparison of out-of-bag bootstrap estimators of classification errors is assessed by Spearman's rank correlation, Wilcoxon signed rank tests and box-plots of a bootstrap distribution of the estimate. The classification methods random forests 15.4%, support vector machines 15.9%, bundling 16.3% to 17.8%, and penalized discriminant analysis 16.8% show the best results.

Conclusions: Using nested-loop cross-validation we account for the tuning of hyperparameters and demonstrate the assessment of different classifiers. We recommend a block design of the bootstrap simulation to allow a statistical assessment of the bootstrap estimates of the misclassification error. The results depend on the data of the clinical study and the given size of the bootstrap sample.

Keywords

Classification, confocal scanning laser ophthalmoscopy, Glaucoma, R, machine learning by small data sets, parameter tuning

Methods Inf Med 2008; 47: 38–46

doi:10.3414/ME0348

1. Introduction

The application of machine learning methods in medicine for automated classification is common practice [1]. Considering different modern classification methods competing for this task it is not obvious how to compare the diagnostic performance. In clinical applications the definition of the examined disease is often complex and different examination methods are used. For illustration we use data of a clinical study on early detection of glaucoma [2].

Glaucoma is a neurodegenerative eye disease and the second most common reason for blindness worldwide [3]. The symptoms are progressive visual field loss and irreversible damage of the optic nerve. Early detection is essential for an efficient treatment. Changes in the optic nerve head precede loss of the visual field. Early detection should be based on examinations of the eye background. Confocal scanning laser ophthalmoscopy (CSLO), performed for example by the Heidelberg Retina Tomograph (HRT) [4], provides a 2.5-dimensional image of the optic nerve head (ONH). Such an image is used to assess the ONH morphology and allows the detection of glaucomatous changes at an early stage of the disease.

We compare 18 classification methods to detect glaucoma based on one clinical data set consisting of HRT measurement results. We use data of a case-control study drawn from the Erlangen Glaucoma Register, which consists of HRT measurement values of 98 glaucoma patients and 98 healthy controls [5]. Several articles discuss classification using visual field data [6], HRT data [7, 8] or both [2, 9]. Schwarzer et al. [10] compare classifier performance in general.

Often classification models involve hyperparameters, which enable the specific adjustment to different classification problems. To take advantage of this flexibility, tuning of the hyperparameters is required. Duin [11] points out the difficulties that arise if classification methods that require tuning, for example multilayer perceptrons, are compared to those classifiers that do not require tuning. When enough data is available, the data set can be split into a learning/tuning set and a test set [12]. The size of our clinical data set is too small to split the data. Instead, we estimate the classifier performance using bootstrapping and we consider the tuning process as part of the method. Our approach follows the recent suggestion to use nested-loop cross-validation by Markowetz and Spang [13].

Frequently, point estimates of the misclassification error are used to illustrate classification performance. We compare the classifiers by inspecting performance distributions rather than point estimates of expected performance. We assess the misclassification error using a bootstrap approach for a given size of the bootstrap sample. To test for an overall difference we use the Friedman test and for pairwise differences we use the Wilcoxon signed rank test. Additionally, we compute point estimates of the expectation of classification performance, namely the out-of-bag error (err^{oob}) and the .632+ bootstrap error ($err^{.632+}$), which can be seen as a bias reduced version of the err^{oob}. We do all our computations using the statistical programming environment **R** [14]. We give a brief description of the 18 used classifiers and describe their implementation in **R**. **R** also provides the procedures necessary for error rate estimation, statistical analysis and visualization of results.

Received: July 20, 2005; accepted: February 1, 2007

In Section 2 we describe the data of the Glaucoma study, we give an overview of the used classification methods and describe the suggested data analysis. The results are reported in Section 3 and discussed in Section 4.

2. Methods

2.1 Confocal Scanning Laser Ophthalmoscopy

Glaucoma is a slowly progressive and irreversible disease which affects the retinal nerve fiber layer. It is the second most frequent cause of blindness and generally occurs in the elderly. The diagnosis is mainly based on measurements of visual field [15, 16] and optic nerve head (ONH) morphology [6]. The intra-ocular pressure (IOP) defines normal- or hyper-tension glaucoma. Damage to the ONH precedes visual field defects, which are symptoms of glaucoma in an advanced stage.

Since we want to examine methods that allow for the early detection of glaucoma, we concentrate on confocal scanning laser ophthalmoscopy (CSLO) by the Heidelberg Retina Tomograph. CSLO is a method to obtain 2.5-dimensional topography images of the optic nerve head. The HRT creates 32 images of the eye background. These images cover a range of about 0.5 to 4.0 millimeters in depth. The depth information that is stored in the image series can be transformed to grey values that lie between 0 (black = near the HRT) and 255 (white = far from the HRT). Figure 1 shows typical normal and glaucomatous topography images. The excavation of the ONH, which is identifiable as the bright area in the topography image is larger in the glaucomatous image which indicates the loss of retinal nerve fibers, i.e. the pathological changes of the ONH.

However, the difference between normal and glaucomatous images is often less obvious. Therefore, features that enable the automated discrimination of the two classes have to be computed.

The HRT software [4] is used to extract features from the topography images. These features are based on manual outlining of the papilla in the reflectivity image performed by a clinical expert. Like the topography image, the reflectivity image is generated from the image series, but instead of depth values, the eye background's reflectivity is shown like a photograph. Figure 2 shows a reflectivity image and the corresponding topography image. The manually plotted so-called contour-line has been automatically transferred to the topography image.

A reference plane that marks a certain height in the ONH is computed based on the contour-line. The papilla is divided into four sectors (temporal, superior, nasal and inferior; see Fig. 3, cf. Fig. 20 of [17]). Three-dimensional landmarks are established from the contour-line and reference plane

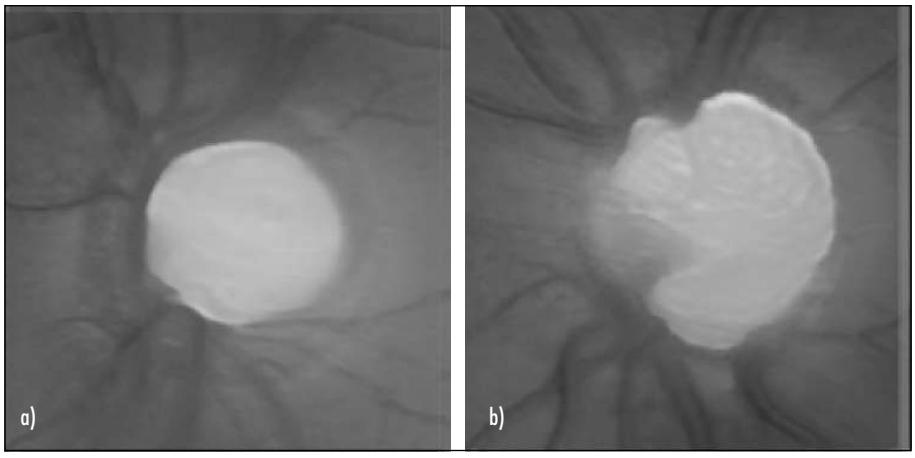

Fig. 1 Normal (left) and glaucomatous (right) topography image of the ONH

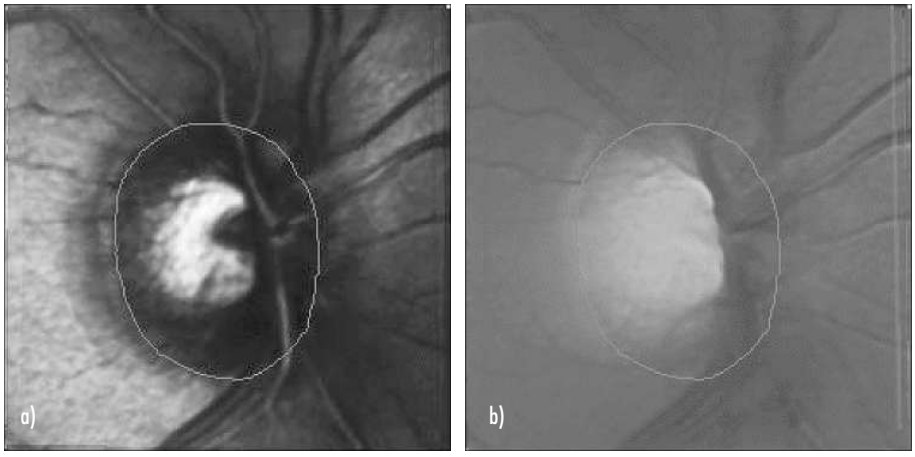

Fig. 2 Reflectivity image with manually outlined papilla (left) and corresponding topography image with automatically transferred outlining (right)

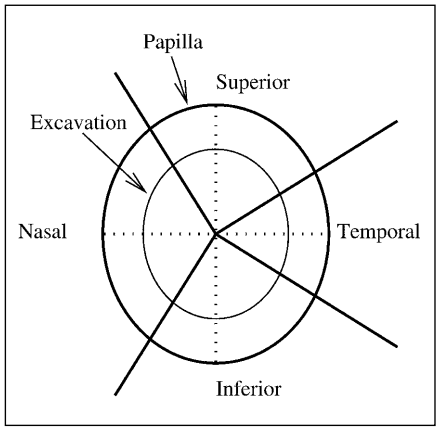

Fig. 3 The papilla is divided into four sectors (example: left eye): The bisectors of the superior and inferior right-angled sectors lie 13° temporal to the vertical optic disc axis. The angle of the temporal sector has 64°, the angle of the nasal sector has 116°.

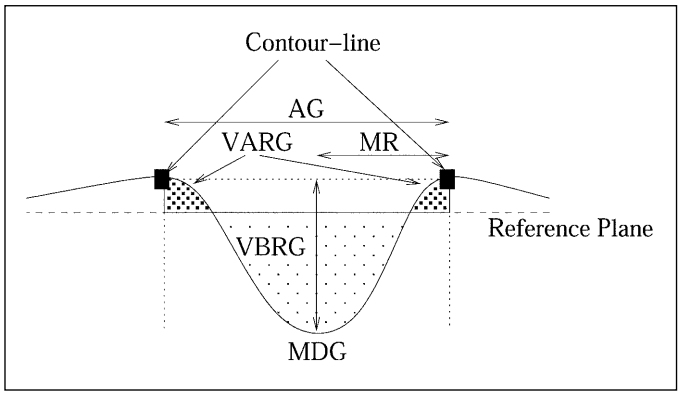

Fig. 4 Profile of the optic nerve head with the features mean radius (MR), volume above reference global (VARG), volume below reference global (VBRG), maximum depth global (MDG) and area global (AG)

and enable the extraction of 62 features, among which are volumes and two-dimensional measurements of the individual sectors and of the global papilla. A list of the features is given in Table 1, and selected features are illustrated in Figure 4.

Table 1 HRT variables (variables indicated by (G,T,S,N,I) are given for several locations in the image): G = global, T = temporal, S = superior, N = nasal, I = inferior.

Variable no.	Variable name
1–5	A – Area (G,T,S,N,I)
6–10	EA – Effective Area (G,T,S,N,I)
11–15	ABR – Area below reference (G,T,S,N,I)
16	HIC – Height in contour
17–21	MHC – Mean height of contour (G,T,S,N,I)
22–26	PHC – Peak height of contour (G,T,S,N,I)
27	HVC – Height variation contour
28–32	VBS – Volume below surface (G,T,S,N,I)
33–37	VAS – Volume above surface (G,T,S,N,I)
38–42	VBR – Volume below reference (G,T,S,N,I)
43–47	VAR – Volume above reference (G,T,S,N,I)
48–52	MD – Maximum depth (G,T,S,N,I)
53–57	TM – Third Moment (G,T,S,N,I)
58	MR – Mean radius
59	RNF – Retinal nerve fiber layer thickness
60	MDIC – Mean depth in contour
61	EMD – Effective mean depth
62	MV – Mean variability

2.2 Classification Methods

In the following we introduce the used classification methods. The methods include established statistical methods and modern machine-learning classifiers.

2.2.1 Linear Discriminant Analysis (LDA)

LDA tries to find a class-separating hyperplane that maximizes the ratio of between-class to within-class variance [18]. The **R** function `lda()` is available in the package `MASS` [19].

2.2.2 Stabilized Linear Discriminant Analysis (sLDA)

The stability of LDA suffers from high-dimensional data. A solution is presented with the stabilized linear discriminant analysis (sLDA) [20, 21]. sLDA performs a LDA on q-dimensional variables $\tilde{\mathbf{x}}^T := \mathbf{x}^T\mathbf{D}_q$, where $\mathbf{x}$ denotes the p-dimensional observation and $\mathbf{D}_q$ is a $p \times q$ projection matrix from the p-dimensional input space into the reduced q-dimensional feature space. The function `slda()` is available in the package `ipred` [22].

2.2.3 Flexible Discriminant Analysis (FDA)

A classification problem can be solved via regression by optimal scoring [23]. LDA then can be described as linear regression. Flexible discriminant analysis (FDA) allows regression with more advanced methods, for example multivariate additive regression splines (MARS) [24]. Multivariate additive regression splines are given by a function f:

$$f(\mathbf{x}) = \alpha + \sum_{m=1}^{M} \beta_m \prod_{k=1}^{K_m} \phi_{km}(x_{v(k,m)}),$$

where $x_{v(k,m)}$, $v(k,m) \in \{1,...,p\}$, is the predictor used in the k-th term of the m-th product. The basis function $\phi(x)$ is defined as $\phi_{km}(x) = \max((x - t_{km}),0)$, $\phi_{k,m+1}(x) = \max((t_{km} - x),0)$ and t_{km} is one of the values of $x_{v(k,m)}$. M is the number of knots of the spline, K_m is the degree of interaction between the variables and α and β_m are the parameters that have to be optimized. We used MARS with degree 1 and 2 for an additive model and a model with pairwise interaction, respectively. In **R**, FDA using MARS is performed by calling `fda(method=mars, degree, nk)`, where `degree` is the degree of interaction, i.e. K_m, and `nk` specifies M, the maximum number of model terms. The function `fda()` is available in the package `mda`.

2.2.4 Penalized Discriminant Analysis (PDA)

The approach of PDA [25] is similar to that of FDA. The classification problem is transformed into a regression problem by optimal scoring, but instead of regression by complex functions like MARS, generalized ridge regression is performed. Consequently, this classification method is called `fda(method=gen.ridge)`. The idea of penalization allows PDA to deal with high-dimensional and highly correlated data.

2.2.5 Mixture Discriminant Analysis (MDA)

The difference between LDA and MDA [26] is the way, the class-conditional densities $p(\mathbf{x}|c_j)$ are modelled. Rather than a single Gaussian distribution, MDA assumes a mixture of Gaussian distributions, i.e. classes are divided into subclasses. MDA is available in the package `mda` and its function call is `mda(subclasses, iter)`, where `iter` limits the total number of iterations.

2.2.6 Logistic Discrimination (LD)

Given two classes, we model the difference of the logarithms of the class-conditional densities as linear in $\mathbf{x}$:

$\log(p(\mathbf{x}|c_1)/p(\mathbf{x}|c_2)) = \beta_0 + \beta^T\mathbf{x}$ [27]. Patterns are assigned to classes in the following way: Let $\beta'_0 = \beta_0 + \log(p(c_1)/p(c_2))$, where $p(c_j)$ is the unconditional class probability of class j, $j = \{1,2\}$. Then $\mathbf{x}$ is assigned to c_1, if $\beta'_0 + \beta^T\mathbf{x} > 0$, else to c_2. Logistic discrimination shows higher accuracy and robustness with respect to small sample size and measurement errors compared to LDA. The **R** function call is `multinom()`. This function is distributed with the package `nnet`.

2.2.7 Multilayer Perceptron (MLP)

Multilayer perceptrons are often used for classification problems in medicine [28, 29]. The MLP is a layered neural network, i.e. its processing units, so-called neurons, are arranged in layers 1 to n, where layer 1 denotes the input layer and layer n denotes the output layer. Layers 2 to $(n - 1)$ are called hidden layers. Multilayer perceptrons are feed-forward networks, i.e. a neuron in layer a can send a signal to a neuron in layer b only if $b > a$. The connections between the neurons are weighted. The computational flexibility of MLPs results from the variability of these weights. Training is done by changing the weights in a way that the desired output for a given input is reached. Back-propagation is an efficient training method for MLPs [30]. This is a gradient-descent method, where the derivative of the neurons' output is needed. Therefore, the threshold function that was used in single-layer perceptrons is not appropriate. A similar function, which is continuously differentiable is given with the logistic function $f(z) = (1 + e^{-z})^{-1}$. The limits of this function are 0 and 1 for $z \to -\infty$ and $z \to \infty$, respectively. To prevent overfitting of MLPs, weight decay can be used. With this variant of the learning algorithm, large weights are avoided and decision boundaries are smoothed. MLPs can separate nonlinearly separable data.

In **R**, a MLP with one hidden layer consisting of a variable number of neurons is implemented. Skip-layer connections are possible, i.e. the input-layer can be directly connected to the output-layer. The use of weight decay is supported. The function is called `nnet(size, decay)`, where the parameter `size` determines the size of the hidden layer, i.e. the number of hidden neurons, and decay is the weight decay. `nnet()` comes with the package `nnet`.

2.2.8 Support Vector Machines (SVM)

The idea of SVMs [31, 32] is to separate two classes by a maximal margin hyperplane. To achieve this goal, data that are probably non-linearly separable in the input space are transformed to a higher dimensional feature space, where linear separability is assumed. This transformation is performed by a so-called kernel function $k(\mathbf{x}_k, \mathbf{x})$, where $\mathbf{x}_k$ denotes the k-th support vector. The hyperplane is defined by these support vectors. The kernel function is composed of non-linear transformations $\mathbf{\Phi}(\mathbf{x}_k)^T\mathbf{\Phi}(\mathbf{x})$. Common kernel functions are the polynomial, the sigmoid or the radial basis function. For the two-class problem, the sign of the output of the SVM determines the class of the input. We use the radial basis function $e^{-\gamma|\mathbf{x}-\mathbf{x}_k|^2}$ as the kernel in our experiments. The **R** function, that implements SVMs is available in the package `e1071` and is called `svm()` [33]. The regularization parameter C, which penalizes the training errors, can be passed to the function via the parameter `cost`.

2.2.9 *k* Nearest Neighbor (kNN)

To assign a new observation to one of the classes, the Euclidean distances between the observation and the data points in the training set are measured [18]. The class is determined by the class membership of the majority of the k nearest data points. kNN classifiers can be applied to linearly non-separable data. We used the **R** function `knn(k)`, available in the package `class`.

2.2.10 Learning Vector Quantization (LVQ)

LVQ is a neural network with one active layer of n neurons, where n is the number of so-called codebook entries. This codebook consists of vectors that represent the different classes. The codebook vectors are used as weight vectors of the neurons. Input vectors $\mathbf{x}$ are compared to each neuron's weight by measurement of the Euclidean distance. The neuron with the weight that is most similar to the input is called the winner neuron. It "fires" and defines the class membership of the input. Several methods exist to assess the vectors of the codebook. We apply OLVQ1 followed by LVQ3 [34]. OLVQ1 performs an initial guess of the codebook vectors and LVQ3 performs fine-tuning to achieve better entries. The **R** functions that are needed to run this classification method, `lvqinit()`, `olvq1()`, `lvq3()` and `lvqtest()`, are available in the package `class`.

2.2.11 Classification Trees (RPART)

At each node of a classification tree, a feature value of the observation to be classified is compared to a threshold until a leaf of the tree is reached. Leafs are labelled with classes and an observation is assigned to the class corresponding to the reached leaf. Hence, a classification tree is a set of splits corresponding to selected variables with a certain ordering. The **R** function to compute classification trees is `rpart()`, available in the package `rpart`.

2.2.12 Bagging

Classification trees are instable in the way that small changes in the training data, for example increasing or decreasing the number of observations, can lead to extremely different trees. Breiman [35] suggests creating several trees by bootstrap sampling. He obtains the final decision by majority voting which results in a stabilized tree-based classifier. We created trees for 200 bootstrap samples. The **R** function is called `bagging()` and it is implemented in the package `ipred`.

2.2.13 Random Forests

Random forests [36] are similar to bagged classification trees. The special feature of random forests is the way the trees (in our case 200) are created. At every split point of the tree, the features which are used to describe the split are drawn from a randomly selected subset of all variables. We fix the subset size to $\text{round}(\log_2(62)) + 1 = 7$, as proposed by Breiman [36]. This method leads to stability against noise and better

Table 2 Mean values (standard deviations) of variables of the used data set (abbreviations see Table 1)

Variable	Normal	Glaucoma
AG	2.608 (0.764)	2.605 (0.539)
ABRG	0.976 (0.786)	1.608 (0.647)
MHCG	0.066 (0.066)	0.122 (0.060)
VBSG	0.497 (0.400)	0.770 (0.372)
VASG	0.066 (0.079)	0.034 (0.025)
VBRG	0.292 (0.435)	0.559 (0.356)
VARG	0.414 (0.197)	0.179 (0.117)
MDG	0.636 (0.203)	0.735 (0.193)
TMG	–0.152 (0.093)	–0.034 (0.088)
MR	0.903 (0.130)	0.907 (0.092)

performance. Again, the final classification result is accomplished by majority voting of all trees. The function call in **R** is called `randomForest()`. It is available in the package `randomForest` [37].

2.2.14 Bundling

Bundling [38] is a bootstrap-based classifier that combines bagging and an arbitrary number of additional classifiers. Classification trees constructed within this algorithm are based on an extended set of features. These additional features are calculated using the out-of-bag sample, i.e. those observations not included in the bootstrap sample. The out-of-bag constitutes an independent sample and enables us to combine a set of classification techniques with bootstrap-aggregated classification trees. As we used bootstrapping for error estimation in this work, it is important not to confuse the out-of-bag sample, which is used to train additional classifiers (oobT) with the out-of-bag sample, which is used for error rate estimation (oobE). Sample oobT is part of the bootstrap sample used for error rate estimation: bundling performs a bootstrapping in the bootstrap sample.

Benchmark results show that bundling performs comparable to the classifier that is best suited for the given data set and distribution. In our experiments, we combine LDA, sLDA, LD and the combination of LD, sLDA, 5-NN and 10-NN with bagging. In **R**, bundling is treated as a special case of the function `bagging()` and is performed by calling this function with the additional parameter `comb`, where `comb` describes the list of additional classifiers that are trained with the out-of-bag data.

Note that the construction of this classifier combined with a bootstrap-based error estimator is computationally demanding. For example, if we estimate the error rate of bundling combined with LDA, we draw 100 bootstrap samples for the construction of the error estimator. For each sample, 200 bootstrap samples are drawn by bundling. A LDA and a classification tree are constructed for each sample. This gives a total of $200 \times 100 = 20{,}000$ calculated LDAs and additionally 20,000 classification trees that are computed to estimate the misclassification error.

2.3 Case-control Study

We compare the classifiers by data of one case-control study drawn from the Erlangen Glaucoma Register [5]. The study consists of 98 normal and 98 glaucomatous observations, matched by age (normal: 54.7 ± 9.3 years, glaucomatous: 54.7 ± 9.3 years) and sex. The glaucoma group contains primary open-angle glaucoma and normal pressure glaucoma. The controls were members of the administrative staff of the hospital or persons who came to exclude a glaucomatous disease. Ocular hypertension was not an exclusion criterion for normal subjects. Eyes with a myopic refractive error <–8 diopters were not included, because of a differing optic disc morphology. Only one eye per person was selected. In the glaucoma group, the more advanced glaucomatous eye was chosen and in the control group, the eye with the better visual field performance was selected. The diagnosis was based on clinical examination, visual field evaluation and optic nerve head analysis. Summary statistics are reported in Table 2.

2.4 Tuning of Hyperparameters

To obtain comparable estimates for the misclassification error, we define the tuning of hyperparameters as part of the methods. Consequently, we include tuning in the

Table 3 Misclassification rates, sensitivities and specificities for tested classifiers

Classifier	err^{oob}	$err^{.632+}$	Sensitivity	Specificity
Random Forests	15.4%	11.0%	82.79%	86.54%
Bundling (RPART, LD, sLDA, 5-NN, 10-NN)	16.3%	11.7%	82.82%	84.97%
Bundling (RPART, LD)	16.3%	11.7%	82.98%	84.70%
Bundling (RPART, sLDA)	17.3%	12.5%	81.22%	84.38%
Bundling (RPART, LDA)	17.8%	13.0%	80.86%	83.67%
SVM	15.9%	13.4%	82.65%	85.70%
PDA	16.8%	15.3%	83.74%	83.02%
MLP	17.2%	14.5%	81.92%	83.92%
Bagging	17.9%	13.0%	80.91%	83.54%
sLDA	18.2%	15.8%	80.63%	83.02%
FDA/MARS (degree=1)	20.0%	17.5%	78.17%	82.05%
FDA/MARS (degree=2)	20.2%	18.6%	77.58%	82.07%
Logistic Discrimination	21.7%	16.9%	76.80%	79.80%
LVQ	22.1%	18.4%	77.13%	78.72%
RPART	22.9%	19.7%	75.48%	78.66%
kNN	24.0%	21.1%	75.26%	76.85%
LDA	27.7%	22.8%	71.84%	72.46%
MDA	27.9%	22.8%	72.49%	71.51%

Table 4 Spearman's rank correlation in % (lower triangular matrix) and significance of Wilcoxon signed rank tests for the given number of bootstrap samples $B = 100$ ("*" indicates a p-value < 0.0003; "+" indicates a p-value < 0.001) between performances (upper triangular matrix).[F1] FDA / MARS (degree 1), [F2] FDA / MARS (degree 2), [B1] Bundling (LDA), [B2] Bundling (sLDA), [B3] Bundling (LD), [B4] Bundling (LD, sLDA, 5NN, 10NN)

	MDA	LDA	kNN	RPART	LVQ	LD	FDA[F1]	FDA[F2]	sLDA	Bagging	Bundling[B1]	Bundling[B2]	MLP	PDA	Bundling[B3]	Bundling[B4]	SVM	Random Forests
MDA			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
LDA	43		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
kNN	4	–2			*	+	*	*	*	*	*	*	*	*	*	*	*	*
RPART	11	27	29				*	*	*	*	*	*	*	*	*	*	*	*
LVQ	11	19	58	32			*	*	*	*	*	*	*	*	*	*	*	*
LD	32	37	–2	–2	5				*	*	*	*	*	*	*	*	*	*
FDA[F1]	10	23	17	19	26	10				*	*	*	*	*	*	*	*	*
FDA[F2]	12	36	14	28	30	–1	60		*	*	*	*	*	*	*	*	*	*
sLDA	27	27	7	12	21	23	20	31						+	*	*	*	*
Bagging	10	25	21	34	14	3	46	40	8						*	*	*	*
Bundling[B1]	6	22	28	33	20	–3	46	42	6	89					*	*	*	*
Bundling[B2]	2	21	29	31	24	0	62	47	6	87	88				*	*		*
MLP	37	34	28	29	27	29	34	43	35	40	35	38						+
PDA	18	28	37	27	43	26	38	40	48	42	41	45	55					
Bundling[B3]	9	29	27	30	23	10	46	41	21	71	71	77	40	51				
Bundling[B4]	11	31	28	30	26	13	46	47	22	70	69	77	45	55	82			
SVM	25	28	31	29	33	23	39	46	35	38	39	48	45	50	42	48		
Random Forests	11	22	30	27	40	–7	61	53	18	64	64	72	42	53	61	67	42	

training process for each bootstrap sample. We perform a threefold cross-validation of the bootstrap sample and the classifier is modeled with the best combination of hyperparameters. This procedure is repeated for each bootstrap sample. Our computations are performed using the **R** functions `errorest()` for error rate estimation and `tune()` [39] for tuning.

We determined the hyperparameters for SVM, MLP, kNN, FDA and MDA. For the SVM, we tried several values for the kernel parameter γ (2^{-10}, 2^{-9},..., 2^4, 2^5), and for C ($2^{-5}, 2^{-4}, ..., 2^{11}, 2^{12}$). The hyperparameters for MLP are the number of neurons in the hidden layer (1, 2,..., 6) and the weight decay (0.025, 0.05, 0.075, 0.1, 0.2, 0.4, 0.6). The hyperparameter k of the kNN was tuned with the values (1, 3, 5, 7, 9, 11, 13, 15). When computing FDA, we varied the number of knots (5, 15, 45, 75, 125) for both degrees of MARS (1 and 2). The MDA was evaluated with a varying number of subclasses (2, 3, 4,..., 10) and different maximum numbers of iteration (3, 5, 10, 20).

2.5 Statistical Analysis

We compare classification methods by using the bootstrap distribution of the estimate of misclassification error. We use the same set of B different learning samples $L^1, \ldots, L^B$ (in our case: $B = 100$) for each classifier. The variance, which is induced by the different bootstrap samples is reduced by subtraction of the mean over all classification methods per sample:

$$err_{ij,aligned} = err_{ij} - \frac{1}{n}\sum_{k=1}^{n} err_{ik},$$

$i = 1,\ldots,B, \qquad j = 1,\ldots,n$

err_{ij} denotes the j-th classifier's classification error using the ith bootstrap sample, $err_{ij,\,aligned}$ denotes the aligned misclassification rates. We additionally compute the out-of-bag estimation of the misclassification error and the bias corrected .632+ estimation [36, 40, 41]. We estimate the sensitivity and the specificity by estimates defined by the classification results in all out-of-bag samples. To illustrate the significance of these results in the given setting, the Friedman test is performed and the pairwise differences between classification methods are tested using the Wilcoxon signed rank test.

3. Results

Using nested-loop cross-validation we account for the tuning of hyperparameters. We demonstrate the comparison of the point estimates of 18 methods on data of one clinical study. The resampling estimates are as-

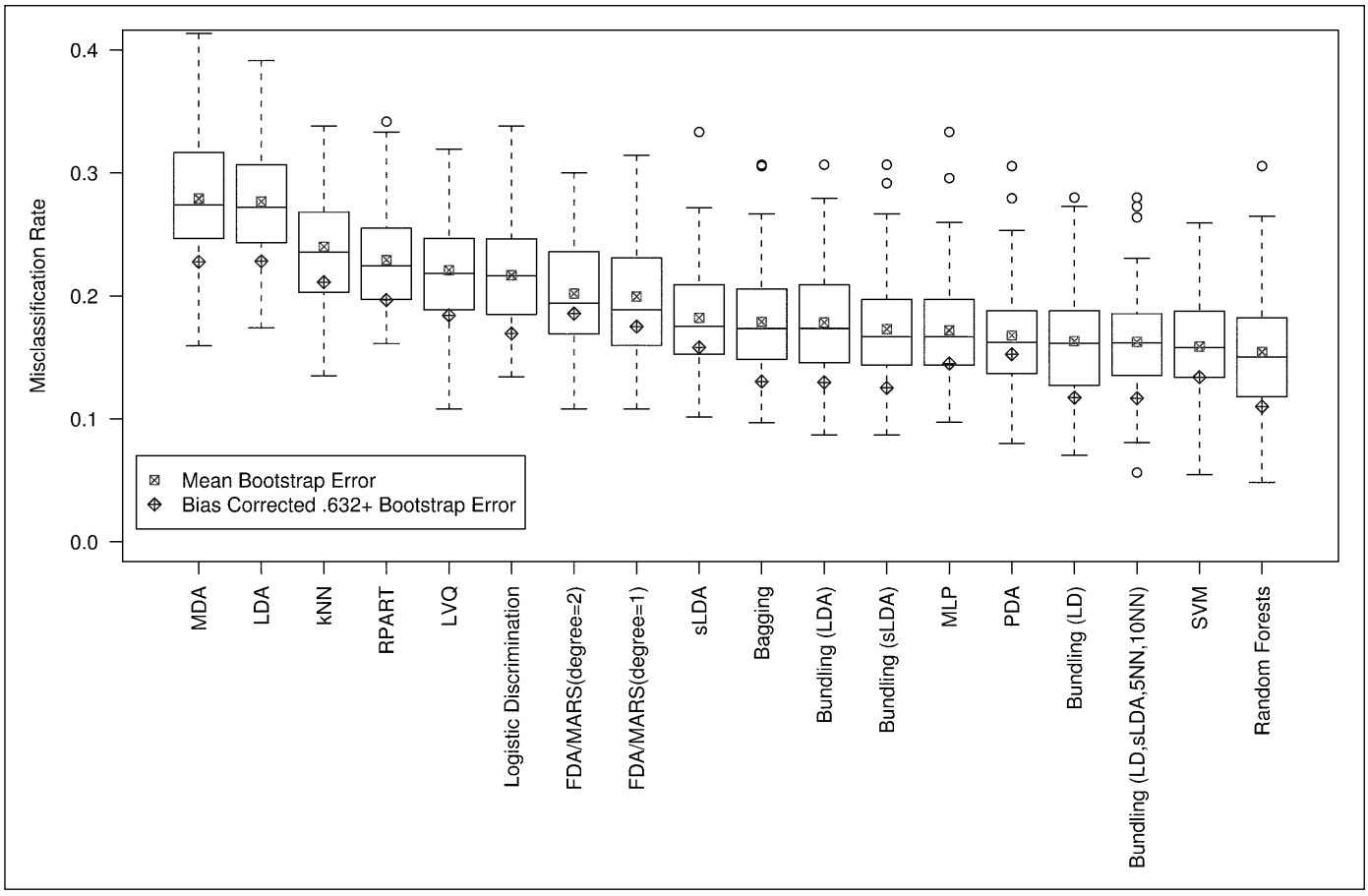

Fig. 5 Aligned misclassification rates

sessed by Spearman's rank correlation, Wilcoxon signed rank tests and naive bootstrap confidence intervals. Table 3 shows the point estimates out-of-bag error (err^{oob}), .632+ bootstrap error ($err^{.632+}$) and estimates of the sensitivity and specificity for the 18 analyzed methods.

Friedman's test shows that the classifier performances differ significantly ($p < 0.001$). To illustrate the meaning of these values in our experimental setup ($B = 100$), significant results of the 144 pairwise comparisons using the Wilcoxon signed rank test are listed in Table 4 (Bonferroni adjusted level of significance: 0.05/144 = 0.0003).

The misclassification rates of the classifier performing best in our setup, namely random forests ($err^{oob} = 15.4\%$, $err^{.632+} = 11.0\%$), differ significantly from those of all other methods, except bundling (best combination: $err^{oob} = 16.3\%$, $err^{.632+} = 11.7\%$), SVM ($err^{oob} = 15.9\%$, $err^{.632+} = 13.4\%$) and PDA ($err^{oob} = 16.8\%$, $err^{.632+} = 15.3\%$). MDA ($err^{oob} = 27.9\%$, $err^{.632+} = 22.8\%$) and LDA ($err^{oob} = 27.7\%$, $err^{.632+} = 22.8\%$) mark the other end of the spectrum. The misclassification rates of these methods also differ significantly from those of all other methods.

Table 4 also shows the computed correlations between the misclassification rates of all classifiers. Similar concepts of classification methods may differ only in the classification of few observations, which results in correlation of estimated misclassification rates.

The aligned misclassification rates are visualized in Figure 5. As the number of bootstrap iterations is increased, the order of the boxplots of the different classification methods converges.

4. Discussion

We introduce several classification methods provided by the statistical programming environment **R** and demonstrate their application. A randomized block design is used to compare the bootstrap estimates of the misclassification rates of the classifiers. Bias resulting from tuning the hyperparameters of several classification methods is avoided by incorporation of an inner cross-validation in the bootstrap process [13]. Thus, classification performances of several different classifiers with and without the use of hyperparameters can be compared using a rather small clinical dataset. Bootstrap-based evaluation is done for a chosen size of the bootstrap samples B. We provide a statistical evaluation of our bootstrap estimation ($B = 100$), which shows sig-

nificant differences of classifier performance. The derived significances depend on the used dataset and the chosen number of bootstrap replications. The results illustrated by the glaucoma study are that ensemble methods [42] perform comparably to the best non-ensemble methods, namely SVM, PDA, MLP and sLDA. Table 3 shows the misclassification rates, sensitivities and specificities obtained with the different classification methods. Misclassification rates are lowest for random forests, closely followed by bundling. SVM and PDA are the only non-ensemble methods that are not significantly worse than random forests. LDA, which is often used in literature on classification of CSLO data [43-46] is outperformed by every other classifier under test except MDA. It is well known that LDA is unstable for high-dimensional data. We did not analyze the reduction of dimensionality by variable selection, which is an important task in medical classification to characterize prognostic factors. We included in our analysis two methods based on LDA which deal with high-dimensional data sets without parameter selection by mapping the data to a lower dimensional feature space (sLDA) or by regularization (PDA).

5. Conclusions

We recommend bootstrapping to assess the differences of misclassification results given one data set of a clinical study and a chosen size of the bootstrap sample. The estimated sensitivities and specificities depend on the Erlangen glaucoma study and the patients selected. Our work illustrates, that the classification method double-bagging, which was used by Mardin et al. [2], results in similar classification characteristics as the best methods. In our paper we do not aim to publish a new glaucoma classification rule for clinical application, but we illustrate a strategy to compare different classifiers using data of one clinical study. We use inner cross-validation for the tuning of hyperparameters as proposed by Markowetz and Spang [13] by incorporating the tuning process into the training process. Thus, the comparability of the classification methods and the flexibility of classifiers using hyperparameters are preserved.

References

1. Kononenko I. Machine learning for medical diagnosis: history, state of the art and perspective. Artificial Intelligence in Medicine 2001; 23: 89-109.
2. Mardin CY, Peters A, Horn FK, Jünemann AG, Lausen B. Improving glaucoma diagnosis by the combination of perimetry and HRT measurements. Journal of Glaucoma 2006; 15 (4): 299-305.
3. Quigley HA. Number of people with glaucoma worldwide. British Journal of Ophthalmology 1996; 80: 389-393.
4. Heidelberg-Engineering. Heidelberg Retina Tomograph: Bedienungsanleitung Software Version 2.01. Heidelberg Engineering GmbH, Heidelberg. 1997.
5. Mardin CY, Hothorn T, Peters A, Jünemann AG, Nguyen NX, Lausen B. New glaucoma classification method based on standard HRT parameters by bagging classification trees. Journal of Glaucoma 2003; 12: 340-346.
6. Chan K, Lee T, Sample PA, Goldbaum MH, Weinreb RN. Comparison of machine learning and traditional classifiers in glaucoma diagnosis. IEEE Transactions on Biomedical Engineering 2002; 49 (9): 963-974.
7. Adler W, Hothorn T, Lausen B. Simulation based analysis of classification of medical images. Methods Inf Med 2004; 43 (2): 150-155.
8. Bowd C, Chan K, Zangwill LM, Goldbaum MH, Lee T, Sejnowski TJ, et al. Comparing neural networks and linear discriminant functions for glaucoma detection using confocal scanning laser ophthalmoscopy of the optic disc. Investigative Ophthalmology and Visual Science 2002; 43 (11): 3444-3454.
9. Peters A, Lausen B, Michelson G, Gefeller O. Diagnosis of glaucoma by indirect classifiers. Methods Inf Med 2003; 42 (1): 99-103.
10. Schwarzer G, Nagata T, D. M, Schmelzeisen D, Schumacher M. Comparison of fuzzy inference, logistic regression, and classification trees (CART). Methods Inf Med 2003; 42 (5): 572-527.
11. Duin RPW. A note on comparing classifiers. Pattern Recognition Letters 1996; 17: 529-536.
12. Van-Gestel T, Suykens JA, Baesens B, Viaene S, Vanthienen J, Dedene G, et al. Benchmarking least squares support vector machine classifiers. Machine Learning 2004; 54 (1): 5-32.
13. Markowetz F, Spang R. Molecular diagnosis – classification, model selection and performance evaluation. Methods Inf Med 2005; 44: 438-443.
14. R Development Core Team. R: A Language and Environment for Statistical Computing: R Foundation for Statistical Computing; 2005.
15. Goldbaum MH, Sample PA, Chan K, Williams J, Lee T-W, Blumenthal E, et al. Comparing machine learning classifiers for diagnosing glaucoma from standard automated perimetry. Investigative Ophthalmology and Visual Science 2002; 43 (1): 162-169.
16. Sample PA, Goldbaum MH, Chan K, Boden C, Lee T, Vasile C, et al. Using machine learning classifiers to identify glaucomatous change earlier in standard visual fields. Investigative Ophthalmology and Visual Science 2002; 43 (8): 2660-2665.
17. Jonas J. Biomorphometrie des Nervus Opticus. Stuttgart: Enke; 1989.
18. Ripley BD. Pattern Recognition and Neural Networks. Cambridge, UK: Cambridge University Press; 1996.
19. Venables WN, Ripley BD. Modern Applied Statistics with S. 4th ed. New York: Springer; 2002.
20. Läuter J. Stabile multivariate Verfahren: Diskriminanzanalyse – Regressionsanalyse – Faktoranalyse. Berlin: Akademie Verlag; 1992.
21. Läuter J, Glimm E, Kropf S. Multivariate tests based on left-spherically distributed linear scores. The Annals of Statistics 1998; 26 (5): 1972-1988.
22. Peters A, Hothorn T, Lausen B. ipred: Improved predictors. R News 2002; 2 (2): 33-36.
23. Hastie T, Tibshirani R, Buja A. Flexible discriminant analysis by optimal scoring. Journal of the American Statistical Association 1994; 1: 1255-1270.
24. Friedman JH. Multivariate adaptive regression splines (with discussion). The Annals of Statistics 1991; 19: 1-141.
25. Hastie T, Buja A, Tibshirani R. Penalized discriminant analysis. The Annals of Statistics 1995; 23: 73-102.
26. Hastie T, Tibshirani R. Discriminant analysis by gaussian mixtures. Journal of the American Statistical Association 1996; 1: 1255-1270.
27. Webb A. Statistical Pattern Recognition. 2nd ed. Chichester, England: Wiley; 2002.
28. Heckerling PS, Gerber BS, Tape TG, Wigton RS. Entering the black box of neural networks. Methods Inf Med 2003; 42 (3): 287-296.
29. Seuchter SA, Eisenacher M, Riesbeck M, Gaebel W, Köpcke W, other members of the A.N.I. Study Group. Methods for predictor analysis of repeated measurements: application to psychiatric data. Methods Inf Med 2004; 43 (2): 184-191.
30. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, editors. Parallel distributed processing (Vol 1). Cambridge, MA: MIT Press; 1986. pp 318-362.
31. Vapnik V. Statistical Learning Theory. New York: Wiley; 1998.
32. Vapnik V. The Nature of Statistical Learning Theory. 2nd ed. New York: Springer; 2000.
33. Meyer D. Support vector machines. R News 2001; 1 (3): 23-26.
34. Kohonen T, Hynninen J, Kangas J, Laaksonen J, Torkkola K. LVQ_PAK: The learning vector quantization program package: Laboratory of Computer and Information Science, Helsinki University of Technology; 1996.
35. Breiman L. Bagging predictors. Machine Learning 1996; 24 (2): 123-140.

36. Breiman L. Random forests. Machine Learning 2001; 45 (1): 5-32.
37. Liaw A, Wiener M. Classification and regression by randomForest. R News 2002; 2 (3): 18-22.
38. Hothorn T, Lausen B. Bundling classifiers by bagging trees. Computational Statistics & Data Analysis 2005; 49: 1068-1078.
39. Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A. e1071: Misc functions of the department of statistics (e1071). TU Wien; 2004.
40. Efron B, Tibshirani R. Improvements on cross-validation: the .632+ bootstrap method. Journal of the American Statistical Association 1997; 92: 548-560.
41. Breiman L. Out-of-bag estimation. Technical Report. Berkeley (CA 94708): Statistics Department, University of California; 1996.
42. Envgeniou T, Pontil M, Eliseeff A. Leave one out error, stability, and generalization of voting combinations of classifiers. Machine Learning 2004; 55 (1): 71-97.
43. Mikelberg F, Parfitt C, Swindale N, Graham S, Drance S, Gosine R. Ability of the Heidelberg Retina Tomograph to detect early glaucomatous visual field loss. Journal of Glaucoma 1995; 4: 242-247.
44. Mardin CY, Horn FK, Jonas J, Budde WM. Preperimetric glaucoma diagnosis by confocal scanning laser tomography of the optic disc. British Journal of Ophthalmology 1999; 83 (3): 299-304.
45. Bathija R, Zangwill LM, Berry CC, Sample PA, Weinreb RN. Detection of early glaucomatous structural damage with confocal scanning laser tomography. Journal of Glaucoma 1998; 127: 7 (2): 121.
46. Uchida H, Brigatti L, Capriolo J. Detection of structural damage from glaucoma with confocal laser image analysis. Investigative Ophthalmology and Visual Science 1996; 37 (12): 2393-2401.

Correspondence to:
Berthold Lausen
Department of Medical Informatics, Biometry and Epidemiology
Friedrich-Alexander-University Erlangen-Nuremberg
Waldstraße 6
91054 Erlangen
Germany
E-mail: Berthold.Lausen@rzmail.uni-erlangen.de