Bootstrap estimated true and false positive rates and ROC curve

Werner Adler, Berthold Lausen*

Department of Biometry and Epidemiology, University Erlangen-Nuremberg, Waldstr. 6, 91054 Erlangen, Germany

A R T I C L E   I N F O

Article history:
Available online 27 September 2008

A B S T R A C T

Diagnostic studies and new biomarkers are assessed by the estimated true and false positive rates of the classification rule. One diagnostic rule is considered for high-dimensional predictor data. Cross-validation and the leave-one-out bootstrap are discussed to estimate true and false positive rates of classifiers by the machine learning methods Adaboost, Bagging, Random Forest, (penalized) logistic regression and support vector machines. The .632+ bootstrap estimation of the misclassification error has been previously proposed to adjust the overfitting of the apparent error. This idea is generalized to the estimation of true and false positive rates. Tree-based simulation models with 8 and 50 binary non-informative variables are analysed to examine the properties of the estimators. Finally, a bootstrap estimation of receiver operating characteristic (ROC) curves is suggested and a .632+ bootstrap estimation of ROC curves is discussed. This approach is applied to high-dimensional gene expression data of leukemia and predictors of image data for glaucoma diagnosis.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The use and analysis of high-dimensional data is a common task in clinical research. Diagnostic studies and studies of new biomarkers involve technologies generating thousands of quantitative or binary features. The Wellcome Trust Case Control Consortium (2007) used a genome wide association study with a 500k chip technology to analyse 14 000 cases of seven common diseases and 3000 shared controls. A genotype pattern of a single nucleotide polymorphism (SNP) can be seen as a binary predictor, e.g. (AA) vs. (AC, CC), consequently the analysis of a 500k chip result in thousands of hypothetical binary predictors. Boulesteix et al. (2007) use multiple testing in high dimensions to identify SNP–SNP interaction and Chen et al. (2008) apply support vector machines for detecting SNP–SNP interaction by association studies. A recent review on biostatistical aspects of genome wide association studies is given by Ziegler et al. (2008). Haslinger et al. (2004) found a set of 25 genes out of 12 559 probes of a DNA microarray to predict the VH mutation status of patients with B-cell chronic lymphocytic leukemia. Croner et al. (2005) use microarray gene expression technology to improve conventional prediction of lymph node metastasis in colorectal carcinoma and estimated true and false positive rates by bootstrap techniques. Other examples result from image data, Horn et al. (2007) use data of frequency doubling perimetry and short-wavelength perimetry for Glaucoma classification.

The estimation of misclassification errors based on a dataset with few or many features is a well known problem. Efron (1983) introduced the .632 bootstrap method as an improvement. The .632 method is based on the observation that classifiers are trained only with about two thirds of the available data in the bootstrap process which causes the overestimation of the misclassification error. It is well known that the apparent error, i.e. the error observed with the whole data as training set, underestimates the unknown true misclassification error. Consequently, Efron propose a weighted average of the bootstrap error and the apparent error. Nearest neighbor classifiers or ensemble methods result in overfitted...
models and in estimation of the apparent error of null. Using a no-information error rate and a heuristic estimate of the overfitting rate, Efron and Tibshirani (1997) suggest an improvement of the .632 estimation, the .632+ estimation. Recently, Jiang and Simon (2007) discuss these proposals and an adjusted bootstrap approach in microarray classification. In our paper we discuss the estimation of the true (TPR) and false (FPR) positive rates of classifiers by resampling methods and we assume that the reference standard of the diagnostic test is available (c.f. Food and Drug Administration (2007)). The TPR is the proportion of positive predictions among positives and the FPR is the proportion of positive predictions among negatives. In receiver-operating characteristic (ROC) curves, TPR is plotted against FPR. Each estimated classification rule can be seen as mapping of the (high)-dimensional predictor space on the interval [0, 1]. We restrict our discussion in the paper on ROC analysis of the one-dimensional discriminant function of the classification rule of the two class situation. We propose a leave-one-out bootstrap estimation of TPR and FPR which is given by the average of estimated true and false positive rates of observations excluded from the bootstrap samples. This estimation helps to derive a .632+ bootstrap estimation for TPR and for FPR. We analyse three simulation models to examine the behaviour of the derived estimators for binary predictors in situations where the apparent error tends to be null. Moreover, we discuss ROC curves obtained with .632+ bootstrap estimated TPR and FPR. All computations were done using R version 2.6.1/2.7.0 (R Development Core Team, 2008).

2. Estimated true and false positive rate

A sample \( Z \) is given consisting of \( N \) observations \((x_i, y_i), i = 1, \ldots, N\). The observations are realizations of the random variables \( X \) and \( Y \), where \( X \) is a \( l \)-dimensional vector of predictors and \( Y \) represents the class membership. In classification or supervised learning we want to predict the class for a given vector of predictors \( X \). In a two-class problem where \( Y \in \{0, 1\} \), the classifier performance can be assessed by the true (TPR) and false (FPR) positive rate, defined as the proportion of positives that were classified correctly (TPR) and the proportion of negatives that were classified as positives (FPR). These performance measurement are more informative than the misclassification error, which simply gives the proportion of false classified observations. To be able to discriminate between positives and negatives, we define \( Z_i \) as observations \((x_i, y_i)\) with \( y_i = c, c \in \{0, 1\} \). Then \( N \) is equal to \( n_0 + n_1 \), where \( n_0 \) is the number of observations, for which \( y_i = 0 \) and \( n_1 \) is the number of observations, for which \( y_i = 1 \). The classification rule \( \hat{f}(X) = \hat{P}(Y = 1|X) \) gives the estimated probability that observation \( X \) belongs to class 1. The false positive rate \( FPR(t) \) of a classifier giving \( P(Y = 1|X) \) for a given threshold \( t \in [0, 1] \) is defined as:

\[
FPR(t) = P[P(Y = 1|X) \geq t|Y = 0].
\] (1)

The true positive rate \( TPR(t) \) is defined as:

\[
TPR(t) = P[P(Y = 1|X) \geq t|Y = 1].
\] (2)

An illustration of the properties of a classification method are ROC curves, where \( TPR(t) \) is plotted against \( FPR(t) \) for given thresholds \( t \), i.e. (see Pepe (2003), Eq. (4.3)):

\[
ROC(\cdot) = \{(FPR(t), TPR(t)), t \in [0, 1]\}.
\] (3)

The ROC curve could also be written as \( ROC(\cdot) = \{(t', ROC(t')), t' \in [0, 1]\} \), where the function \( ROC(t') \) maps \( t' \) to \( TPR(t) \) and \( t \) is the threshold given by \( FPR(t) = t' \).

2.1. Apparent true and false positive rate

The apparent error is defined as the misclassification rate, i.e. the proportion of falsely classified observations among all predictions, which is observed when the dataset is used for both, training and testing the classifier. Accordingly, we define the apparent true positive rate \( \overline{TPR}(t) \) which is obtained using the same data for training the classification rule and testing \( \hat{f}(X) \):

\[
\overline{TPR}(t) = \frac{1}{n_1} \sum_{(x_i,y_i) \in Z_1} I(\hat{f}(x_i) \geq t), \quad t \in [0, 1],
\] (4)

where \( I() \) denotes the indicator function which is one if the condition is true and else zero.

The apparent \( FPR(t) \) is defined as:

\[
\overline{FPR}(t) = \frac{1}{n_0} \sum_{(x_i,y_i) \in Z_0} I(\hat{f}(x_i) \geq t), \quad t \in [0, 1]
\] (5)

and the ROC curve is defined as:

\[
ROC(\cdot) = \{\{(\overline{FPR}(t), \overline{TPR}(t)), t \in [0, 1]\} \}
\] (6)

As the apparent true positive rate tends to overestimate the true \( TPR \), cross-validation or bootstrap can be used to reduce the bias analogously to the estimation of the misclassification error (Efron, 1983; Efron and Tibshirani, 1997).
2.2. Cross-validation estimated TPR and FPR

In $k$-fold cross-validation (where $k$ often is defined as 5 or 10), observation $x_i$ with $i = 1, \ldots, N$ is grouped into one of $k$ partitions noted by $\kappa(i)$. $\kappa(i)$ denotes an indexing function that indicates the partition to which the observation $i$ is allocated by a randomization. In $k$ runs, one of these partitions constitutes the test sample (in each run a different one) while the observations of the remaining $k - 1$ partitions are used for training. The $k$-fold cross-validation estimation of the true positive rate and the false positive rate is defined by:

$$
\text{TPR}^{CV}(t) = \frac{1}{n_1} \sum_{(x_i,y_i) \in Z_1} I(\hat{f}^{\kappa(i)}(x_i) \geq t), \quad t \in [0, 1],
$$

$$
\text{FPR}^{CV}(t) = \frac{1}{n_0} \sum_{(x_i,y_i) \in Z_0} I(\hat{f}^{\kappa(i)}(x_i) \geq t), \quad t \in [0, 1],
$$

where $\hat{f}^{\kappa(i)}(x_i)$ is the classifier trained without the $\kappa(i)$th partition which is used for testing.

The cross-validation estimation of the ROC curve is defined by

$$
\text{ROC}^{CV}(\cdot) = \{(\text{FPR}^{CV}(t), \text{TPR}^{CV}(t)), t \in [0, 1]\}.
$$

2.3. Bootstrap estimated TPR and FPR

Bootstrap methods are applied to gain smoother estimates than by cross-validation, i.e. to reduce the variability. The leave-one-out bootstrap estimator is obtained by drawing $B$ bootstrap samples $Z^b$ of size $N$ with replacement. The observations in the bootstrap samples are used for training, while the remaining observations which constitute the out-of-bag sample are used for testing. Performance estimations are averaged for each prediction over all out-of-bag samples and the leave-one-out bootstrap estimate of the TPR and FPR is defined analogously to the leave-one-out bootstrap error (see Hastie et al. (2001) Eq. (7.50)):

$$
\text{TPR}^{(1)}(t) = \frac{1}{n_1} \sum_{i=1}^{N} I(y_i = 1) \sum_{b \in C^{-1}_i} I(\hat{f}^{ab}(x_i) \geq t), \quad t \in [0, 1],
$$

$$
\text{FPR}^{(1)}(t) = \frac{1}{n_0} \sum_{i=1}^{N} I(y_i = 0) \sum_{b \in C^0_i} I(\hat{f}^{ab}(x_i) \geq t), \quad t \in [0, 1],
$$

where $C^{-1}_c = \{b : (x_i, y_i) \notin Z^b \land y_i = c\}, c \in \{0, 1\}$ with $|C^{-1}_c|$ denoting the number of elements in $C^{-1}_c$ and rule $\hat{f}^{ab}(X)$ is obtained using $Z^b$ for training.

The bootstrap estimated ROC curve is defined by

$$
\text{ROC}^{(1)}(\cdot) = \{(\text{FPR}^{(1)}(t), \text{TPR}^{(1)}(t)), t \in [0, 1]\}.
$$

2.4. .632 bootstrap estimated TPR and FPR

Using bootstrap resampling, about two thirds of the data are used for training. This causes underestimation of the true TPR by the leave-one-out bootstrap. Following the proposal of Efron and Tibshirani, we define the .632 bootstrap estimation as a weighted sum of the apparent and the leave-one-out bootstrap estimations:

$$
\text{TPR}^{.632}(t) = .368 \cdot \text{TPR}(t) + .632 \cdot \text{TPR}^{(1)}(t), \quad t \in [0, 1],
$$

$$
\text{FPR}^{.632}(t) = .368 \cdot \text{FPR}(t) + .632 \cdot \text{FPR}^{(1)}(t), \quad t \in [0, 1].
$$

The .632 estimation of the ROC curve is defined by

$$
\text{ROC}^{.632}(\cdot) = \{(\text{FPR}^{.632}(t), \text{TPR}^{.632}(t)), t \in [0, 1]\}.
$$
2.5. .632+ bootstrap estimated TPR and FPR

In the overfitting situation, where the apparent TPR becomes one, the .632 bootstrap TPR overestimates the true TPR. Efron and Tibshirani (1997) improved the .632 misclassification error estimator by taking the amount of overfitting into account. We estimate the overfitting rate $\hat{R}$ defining no-information TPR and FPR, denoted by $\hat{\gamma}_{\text{TPR}}$ and $\hat{\gamma}_{\text{FPR}}$. The no-information rates would be observed if the predictions were independent of the class memberships $y_i$. In the two-class problem the no-information error rate is estimated using the proportion $\hat{p}$ of responses $y_i$ equaling 1 and the proportion $\hat{q}(t)$ of predictions equaling 1, which depends on threshold $t$. The latter is estimated using all observations of the dataset as training data, where the number of observations in $Z$ is denoted as $|Z|$:

$$\hat{p} = \frac{|Z_i|}{|Z|}, \quad \hat{q}(t) = \frac{\sum_{(x_i,y_i) \in Z} I(\hat{q}^Z(x_i) \geq t)}{|Z|}, \quad t \in [0, 1].$$

(16)

The no-information error rate $\hat{\gamma}_{\text{err}}$ is defined as:

$$\hat{\gamma}_{\text{err}}(t) = \hat{p}(1 - \hat{q}(t)) + (1 - \hat{p})\hat{q}(t), \quad t \in [0, 1].$$

(17)

With the proportion of positives $p$, the error is defined as $p \times (1 - \text{TPR}) + (1 - p) \times (\text{FPR})$. From this and Eq. (17) follows, that the no-information TPR $\hat{\gamma}_{\text{TPR}}(t)$ equals $\hat{q}(t), t \in [0, 1]$ and the no-information FPR $\hat{\gamma}_{\text{FPR}}(t)$ equals $\hat{q}(t), t \in [0, 1]$. Now, we can define the relative overfitting rates:

$$\hat{R}_{\text{TPR}}(t) = \frac{\hat{\gamma}_{\text{TPR}}(t) - \hat{\gamma}_{\text{TPR}}(1)}{\hat{\gamma}_{\text{TPR}}(t) - \hat{\gamma}_{\text{TPR}}(0)},$$

(18)

$$\hat{R}_{\text{FPR}}(t) = \frac{\hat{\gamma}_{\text{FPR}}(t) - \hat{\gamma}_{\text{FPR}}(0)}{\hat{\gamma}_{\text{FPR}}(t) - \hat{\gamma}_{\text{FPR}}(1)}.$$  

(19)

To guarantee that the relative overfitting rate is in $[0, 1]$, the definitions of $\hat{R}_{\text{TPR}}, \hat{\gamma}_{\text{TPR}}^{(1)}$, $\hat{R}_{\text{FPR}}$ and $\hat{\gamma}_{\text{FPR}}^{(1)}$ are modified:

$$\hat{\gamma}_{\text{TPR}}^{(1)}(t) = \max(\hat{\gamma}_{\text{TPR}}^{(1)}(t), \hat{\gamma}_{\text{TPR}}(t)), \quad t \in [0, 1].$$

(20)

$$\hat{R}_{\text{TPR}}^{(1)}(t) = \begin{cases} \frac{\hat{\gamma}_{\text{TPR}}(t) - \hat{\gamma}_{\text{TPR}}^{(1)}(t)}{\hat{\gamma}_{\text{TPR}}(t) - \hat{\gamma}_{\text{TPR}}(0)} & \text{if } \hat{\gamma}_{\text{TPR}}^{(1)}(t) < \hat{\gamma}_{\text{TPR}}(t) \quad t \in [0, 1], \\ 0 & \text{else} \end{cases}$$

(21)

$$\hat{\gamma}_{\text{FPR}}^{(1)}(t) = \min(\hat{\gamma}_{\text{FPR}}^{(1)}(t), \hat{\gamma}_{\text{FPR}}(t)), \quad t \in [0, 1] \quad \text{and}$$

$$\hat{R}_{\text{FPR}}^{(1)}(t) = \begin{cases} \frac{\hat{\gamma}_{\text{FPR}}(t) - \hat{\gamma}_{\text{FPR}}^{(1)}(t)}{\hat{\gamma}_{\text{FPR}}(t) - \hat{\gamma}_{\text{FPR}}(0)} & \text{if } \hat{\gamma}_{\text{FPR}}^{(1)}(t) > \hat{\gamma}_{\text{FPR}}(t) \quad t \in [0, 1], \\ 0 & \text{else} \end{cases}$$

(22)

We define the .632+ bootstrap estimation of the true and false positive rates by:

$$\hat{\text{TPR}}^{.632+}(t) = \hat{\text{TPR}}^{.632}(t) + \left(\hat{\gamma}_{\text{TPR}}^{(1)}(t) - \hat{\gamma}_{\text{TPR}}(t)\right) \cdot \frac{.368 \cdot .632 \cdot \hat{R}_{\text{TPR}}(t)}{1 - .368 \cdot \hat{R}_{\text{TPR}}(t)}, \quad t \in [0, 1],$$

(24)

$$\hat{\text{FPR}}^{.632+}(t) = \hat{\text{FPR}}^{.632}(t) + \left(\hat{\gamma}_{\text{FPR}}^{(1)}(t) - \hat{\gamma}_{\text{FPR}}(t)\right) \cdot \frac{.368 \cdot .632 \cdot \hat{R}_{\text{FPR}}(t)}{1 - .368 \cdot \hat{R}_{\text{FPR}}(t)}, \quad t \in [0, 1].$$

(25)

Hence, the .632+ estimation can be seen as the weighted average of the downward biased leave-one-out bootstrap estimator and the upward biased apparent estimation using weights derived by an estimated overfitting rate.

The .632+ estimation of the ROC curve is defined by

$$\hat{\text{ROC}}^{.632+}(t) = \{(\hat{\text{FPR}}^{.632+}(t), \hat{\text{TPR}}^{.632+}(t)), \quad t \in [0, 1]\}.$$  

(26)

3. Simulation models

We conducted a simulation study to examine the properties of the estimators for binary predictors in situations where the apparent error tends to be null. To analyse the effect of sample size, we examined the sample sizes $n = 100$ and $n = 400$. We evaluated the estimators of the TPR and FPR by the computation of the relative bias and the relative mean squared error (MSE) of the estimators. We used Adaboost (Friedman, 2002; Culp et al., 2006), Bagging (Breiman, 1996), Random
Fig. 1. Tree structure of simulation model 2: the bold path shows that the expected majority of the observations with values $x_1 = 0, x_2 = 1, x_5 = 0$ belongs to class 0. In the leaves of the tree, the probabilities $P(Y = 1)$ are \{0.05, 0.65, 0.15, 0.75, 0.25, 0.85, 0.35, 0.95\} from left to right.

Forest (Breiman, 2001) and logistic regression with forward variable selection as classifiers. The methods were provided in the R packages ada, ipred, and randomForest. As three of these classifiers are tree-based, we examine the behaviour of the .632+ estimates in situations where apparent TPR and apparent FPR tends to be one by allowing fully grown trees. In R this is achieved by setting \texttt{minsplit} = 2.

We performed 100 Monte-Carlo replications for the simulation of the estimated TPR and FPR. Additionally, we simulated the true TPR and FPR by generating 1000 training data sets for sample size $n_{\text{learn}} = 100$ and $n_{\text{learn}} = 400$, respectively, and 1000 test data sets of size $n_{\text{test}} = 1000$. The simulated true TPR and FPR are obtained by testing classifiers trained on the learning data with the test data and averaging over all 1000 simulation runs.

Being interested in situations where the apparent error tends to be null, we focus on simulations with binary predictors. Our simulation models are characterized by only a few informative variables, i.e. variables that are correlated with the class. We use one informative binary variable in model one, five in model two and six in model three, with eight or 50 no-informative binary variables in different runs. The five informative variables in our second model are binary and conditionally dependent given the class. In the simulation model, this means that first the class membership variable is generated. Then the informative variables are calculated, their value depending on the class variable. The dependency structure can be depicted by a binary tree. The same structure is further examined in simulation model three where an additional continuous informative variable is generated. The design of our simulation models allows to decrease the expected classifier performance by increasing the number of non-informative variables, while the tree structure of the informative part supports their tree-based nature. The first model with only one informative variable serves as simple reference, while the third model introduces a continuous variable and therefore loosens the binary structure of the model two.

3.1. Model 1

In the first model, one informative binary variable $x$ is generated which correctly predicts the true class with probability 0.8, i.e. $P(Y = 1|X = 1) = 0.8$. The Bayes classifier is defined as the rule which defines to the most probable class (Hastie et al., 2001, p. 21) and its error rate is the Bayes error. Therefore, in this case the Bayes error $\text{err}^{\text{Bayes}}$ equals 0.2. Additionally, 50 non-informative binary variables are created. Subsets of these variables of eight and 50 are used for classification.

3.2. Model 2

Here, additionally to the non-informative variables, five informative binary variables $x_1, x_2, x_3, x_4$ and $x_5$ are generated. The probability of $x_v$ ($v \in \{1, 2, 3, 4, 5\}$) to be either 0 or 1 is 0.5. The dependency structure between these five variables and the class membership is defined in Fig. 1. In the leaves of the tree, the probabilities $P(Y = 1)$ are \{0.05, 0.65, 0.15, 0.75, 0.25, 0.85, 0.35, 0.95\} from left to right. The probabilities are chosen to define the Bayes error $\text{err}^{\text{Bayes}}$ equal to 0.2. The bold path in Fig. 1 gives an example, how class membership was determined in the simulation model. Given the values $x_1 = 0, x_2 = 1, x_5 = 0$, the probability $P(Y = 1)$ equals 0.15.

3.3. Model 3

In our third simulation model, we create the tree-based structure from model 2 and additionally generate a continuous variable $x_6$. $x_6$ is uniformly distributed in $[-3, 3]$. The value determines a probability in the following way:

$$ p'_6 = x_6 \cdot 0.25 - 0.5 $$

(27)
and

$$p_6 = \begin{cases} 1 & \text{if } p_6' > 1 \\ 0 & \text{if } p_6' < 0 \\ p_6' & \text{else}. \end{cases} \quad (28)$$

The class probability is given by the mean of the probability resulting from the tree and $p_6$. By the introduction of a continuous variable, the simulation model itself contributes to the continuity of the distribution of the probabilities to belong to one of the two classes and avoids possible artifacts caused by the discrete structure of model two.

4. Simulation results

To be able to discuss the quality of the estimators, we report the relative bias and the relative MSE of the estimation of $TPR$. The bias is defined as $Bias_{TPR}^{sim} = TPR - TPR^{sim}$, where $TPR$ is the simulated expectation of the estimator and $TPR^{sim}$ is the simulated $TPR$. The relative bias is defined as $Bias_{TPR}^{sim} = Bias_{TPR}^{sim} / \sqrt{TPR_{TPR}^{sim} \cdot (1 - TPR^{sim})}$ and the relative MSE is defined by

$$MSE_{TPR}^{sim} = (VAR_{TPR}^{sim} (TPR) + (Bias_{TPR}^{sim})^2) / (TPR_{TPR}^{sim} \cdot (1 - TPR^{sim})).$$

We provide the relative bias of the .632+, the leave-one-out bootstrap and the cross-validation estimation of Random Forest plotted against the simulated $TPR$ of four different situations for simulation models one, two and three in Fig. 2. The first two rows show situations using the large learning set ($n_{learn} = 100$), the lower rows illustrate two situations using the large learning set ($n_{learn} = 400$). For each learning set size, the situation of eight and 50 non-informative variables are shown. In simulation model one, for $TPR^{sim} < 0.7$, the .632+ estimation is comparable to the cross-validation estimation. For $TPR^{sim} > 0.8$, the leave-one-out bootstrap and the .632+ show an increase of relative bias, which results in an upwards biased .632+ estimator and an improvement of the leave-one-out bootstrap. The images also illustrate that the .632+ estimator is able to reduce the downward bias of the leave-one-out bootstrap in simulation model two and for the small learning set in simulation three. For the large learning set in simulation model three, the leave-one-out estimator shows little bias and the .632+ estimator tends to overestimate the $TPR$.

The relative MSE of these estimators in the same simulation situations is shown in Fig. 3. Except for large learning sets in simulations one and three, the MSE of the .632+ estimator lies either below the MSE of the cross-validation estimation or is comparable to it.

Similar figures illustrating the relative bias and relative MSE of Adaboost, Bagging and logistic regression with variable selection are provided in the supplement. The results of Bagging are very similar to those of Random Forest. One exception is the MSE for 50 non-informative variables in the large learning set in simulation model two. Here, the MSE of all estimators is larger for $TPR^{sim} = 0.5$ and decreases as $TPR^{sim}$ increases. The MSE for the leave-one-out estimator is largest, while for $TPR^{sim} > 0.6$, the cross-validation estimator and the .632+ are comparable. The relative bias of Adaboost plotted against the simulated $TPR$ shows the same pattern for all situations: the bias of the leave-one-out bootstrap decreases as the $TPR^{sim}$ increases from 0.5 to 1. In simulation model one, this leads to an improvement of the estimator, while in simulation models two and three, considerable negative relative bias is attained. The relative bias of the .632+ estimator lies above that of the leave-one-out bootstrap in all situations. Consequently, it is biased upwards in simulation model one, while in simulation models two and three the .632+ is an improvement to the leave-one-out bootstrap. The cross-validation estimation shows lowest relative bias in all situations. The MSE of the leave-one-out bootstrap shows that this estimator fails in simulation model two, as the MSE increases dramatically for $TPR^{sim} > 0.7$. The same is true for the small learning set in simulation model three. The .632+ estimator can partly correct this behaviour but also suffers from an increasing MSE in simulation model two when the small learning set is used or when 50 non-informative variables are provided in the large learning set.

In simulation model one and three, the MSE of the .632+ estimator is comparable to the cross-validation estimator.

In Fig. 4, we present the performance of the estimators in simulation model three with 50 non-informative variables in the large learning set by their relative bias, relative MSE and the estimated ROC curve. The simulated ROC curve is plotted in grey color. The cross-validation estimation of the ROC curve is the best approximation to the simulated curve. The .632+ estimated curve tends to overestimate the true curve, except for logistic regression, where the .632+ estimation improves the leave-one-out estimation but is still biased downwards.

Table 1 provides the influence of sample size 100 or 400 and 8 or 50 non-informative variables for a given FPR of 0.4 on the estimations of $TPR$ for Adaboost, Bagging, Random Forest and logistic regression with forward variable selection. Results are reported by simulated expectations and standard deviations. The standard errors of the simulated true positive rates are easily calculated by dividing the standard deviations of $TPR^{sim}$ by $\sqrt{1000}$, the standard errors of estimators by dividing standard deviations by $\sqrt{100} = 10$. Except for logistic regression, the apparent true positive rate is one or very close to it (0.999 for Adaboost, $n_{learn} = 400$, eight non-informative variables). In these situations, the simple .632 estimation overestimates the true $TPR$. As the leave-one-out bootstrap underestimates the $TPR$, the .632+ estimation is an improvement of both, leave-one-out bootstrap and .632 estimation. For $n = 100$ with 50 non-informative variables, the .632+ estimation performs with lowest bias, whereas in other situations, the cross-validation estimation is closest to the simulated $TPR$. For logistic regression with variable selection, where the apparent $TPR$ is not one, the performance of the estimators is heterogenous in the different situations.
Fig. 2. Random Forest: Relative bias \( t \) plotted against simulated TPR\( t \), \( t \in [0, 1] \). Columns one, two and three show results for simulation model one, two and three, respectively. In the first and second row, \( n_{\text{learn}} = 100 \), in rows three and four, \( n_{\text{learn}} = 400 \). Rows one and three show the simulation results with 8 non-informative variables and rows two and four show the results for 50 non-informative variables of estimated TPR (.632+ solid line, oob dashed line and cv dotted line).

In the supplement, we provide similar tables for simulation models one and three (tables 2, 3), as well as a table providing the true positive rates for FPR\textsuperscript{sim} between 0.1 and 0.4 of simulation model one with sample size 400 and eight non-informative variables (table 1) and true positive rates for estimated false positive rates between 0.1 and 0.4 for simulation model three with \( n_{\text{learn}} = 400 \) and 50 non-informative variables (table 4). In simulation model one, the cross-validation estimator shows the lowest bias for Adaboost, Bagging and Random Forest, while the .632+ estimator has a lower standard deviation. The performance of the estimators with logistic regression shows no clear pattern, as in simulation model three but different than in simulation model two, where the cross-validation estimator has lowest bias. Using Adaboost, Bagging and Random Forest, the .632+ estimator is closest to the simulated TPR in all situations of simulation model two, except for the large learning set with eight non-informative variables. Here, again the cross-validation estimator shows lowest bias.

5. Application: Prediction of VH mutation status

Haslinger et al. (2004) found a set of 25 genes out of 12,559 probes of a DNA microarray to predict the VH mutation status of patients with B-cell chronic lymphocytic leukemia and using 12 genes for a predictor the authors report a cross-validation estimate of the misclassification error of 4%. After preprocessing the microarray data and selecting in each resample 50 genes of 12,559, we assessed the predictive value for 49 patients with VH mutations out of 99 patients. We applied the data analysis as discussed in Croner et al. (2005) and Croner et al. (2008). Table 2 shows estimated error, TPR and FPR for Bagging, Random forest, support vector machine (SVM), penalized logistic regression with step wise forward variable selection of the predictors and logistic regression without. SVM is included for comparison to the data analysis of Croner et al. (2005) and other applications. The point estimates do not show one order of the different estimation methods. Random forest tends to
lowest .632+ estimated missclassification rates of 0.157, highest TPR of 0.872 and FPR of 0.189. The original data analysis of Haslinger et al. (2004) was too optimistic computing a misclassification error without assessing the gene selection.

6. Application: Glaucoma diagnosis

Glaucoma is a major reason for blindness worldwide (Quigley, 1996). This eye disease causes optic nerve atrophy which leads to a reduction of the visual field. The diagnostic study consists of 98 normal and 98 glaucomatous observations. Both groups are matched by age (normal: 54.7 ± 9.3 years, glaucomatous: 54.7 ± 9.3 years) and sex to prevent for possible confounding. The glaucoma group consists of primary open-angle glaucoma and normal pressure glaucoma, while in the control group ocular hypertension is included (Mardin et al., 2003). The observations consist of 63 measurements obtained by the Heidelberg Retina Tomograph (HRT). This device is used to create laser scanning images of the eye background. Based on these images, variables describing the geometry of the optic nerve head are calculated. These variables include areas, volumes and third moments. For a detailed description see Adler et al. (2008).

Table 3 shows the estimates of the classification rates, TPR and FPR for the three classifiers with threshold $t = 0.5$. Random Forest tends to lowest estimated misclassification rates and FPR and highest TPR. The point estimates for Random Forest do not show one specific order of the different estimation methods. For the methods Adaboost, Bagging and Random Forest we also calculated the repeated leave-one-out (rloob) and adjusted bootstrap estimator (ABS) proposed by Jiang and Simon (2007). The repeated leave-one-out was in all cases very similar to the leave-one-out bootstrap. The estimations of $\frac{\hat{Err}_{rloob}}{\hat{TPR}_{rloob}}$, $\frac{\hat{TPR}_{rloob}}{\hat{TPR}_{rloob}}$ were for Adaboost: 0.163/0.817/0.143, for Bagging: 0.180/0.801/0.160 and for Random Forest: 0.156/0.820/0.132. The adjusted bootstrap turned out to be an estimator that produces results somewhere between...
the .632+, the leave-one-out and cross-validation estimators. \( \frac{\hat{\text{Err}}_{\text{ABS}}}{\hat{T}\text{PR}_{\text{ABS}}} / \frac{\hat{\text{FPR}}_{\text{ABS}}}{\hat{T}\text{PR}} \) were for Adaboost: 0.143/0.824/0.117, for Bagging: 0.176/0.797/0.150 and for Random Forest: 0.140/0.824/0.104.

7. Discussion

Cross-validation, bootstrap and .632+ estimation is often used for the misclassification error. Using a high number of features classifiers or machine learning methods overfit the data. The apparent error is estimated by null indicating a complete fit of the data. In these situations the out-of-bag bootstrap and .632+ estimation tend to be substantially different (Hothorn and Lausen, 2003). Adler et al. (2008) get up to 40% higher estimates for the out-of-bag bootstrap compared to the .632+. Recently, Jiang and Simon (2007) suggested a new adjusted bootstrap approach of the .632+ principle and the authors were able to show in special situations a slightly better performance of the new proposal compared to .632+.

We proposed and applied cross-validation, leave-one-out bootstrap and the .632+ method to the estimation of TPR, FPR and ROC curve. The results of three simulation models indicate that our proposal of .632+ estimation of TPR and FPR can be used and the properties are similar as the known results for the estimation of the misclassification error. We modify the .632+ principle of using a relative overfitting rate based on the no-information error rate for our proposal for .632+ estimation of TPR and FPR. Our proposal of .632+ estimation of the ROC curve is derived by the .632+ estimators of TPR and FPR. We do not use a measure of relative overfitting of the ROC curve. Such a measure may be defined by the no-information ROC curve, the diagonal. Another idea of .632+ estimation of the ROC curve is suggested by Gerds et al. (2008). In our paper we do not discuss the selection of the best classifier, but the estimation can be applied for bundling classifiers (Hothorn and Lausen, 2005) and indirect classification (Peters et al., 2005). Our paper provides the theory for the estimation of TPR, FPR
Table 1
Simulation model 3. Simulated true TPR, simulated $\hat{TPR}$ for Adaboost, Bagging, Random Forest, and logistic regression for given specificity of 0.4, sample size 100 and 400, 8 and 50 non-informative variables. In brackets standard deviation of simulation.

<table>
<thead>
<tr>
<th></th>
<th>Adaboost</th>
<th>Bagging</th>
<th>Random Forest</th>
<th>Log. reg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(100, 8)</td>
<td>$TPR_{im}$</td>
<td>0.653 (0.052)</td>
<td>0.623 (0.055)</td>
<td>0.648 (0.049)</td>
</tr>
<tr>
<td>(100, 8)</td>
<td>$TPR_{0.5}$</td>
<td>0.665 (0.064)</td>
<td>0.637 (0.065)</td>
<td>0.66 (0.066)</td>
</tr>
<tr>
<td>(100, 8)</td>
<td>$TPR_{(1)}$</td>
<td>0.622 (0.053)</td>
<td>0.599 (0.052)</td>
<td>0.618 (0.054)</td>
</tr>
<tr>
<td>(100, 8)</td>
<td>$TPR_{1}$</td>
<td>0.645 (0.064)</td>
<td>0.615 (0.068)</td>
<td>0.639 (0.068)</td>
</tr>
<tr>
<td>(100, 8)</td>
<td>$TPR_{50}$</td>
<td>0.761 (0.033)</td>
<td>0.746 (0.033)</td>
<td>0.758 (0.034)</td>
</tr>
<tr>
<td>(100, 8)</td>
<td>$TPR_{100}$</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0.002)</td>
</tr>
<tr>
<td>(100, 50)</td>
<td>$TPR_{im}$</td>
<td>0.636 (0.056)</td>
<td>0.632 (0.058)</td>
<td>0.611 (0.051)</td>
</tr>
<tr>
<td>(100, 50)</td>
<td>$TPR_{0.5}$</td>
<td>0.641 (0.071)</td>
<td>0.634 (0.069)</td>
<td>0.598 (0.072)</td>
</tr>
<tr>
<td>(100, 50)</td>
<td>$TPR_{(1)}$</td>
<td>0.603 (0.058)</td>
<td>0.597 (0.056)</td>
<td>0.569 (0.056)</td>
</tr>
<tr>
<td>(100, 50)</td>
<td>$TPR_{1}$</td>
<td>0.622 (0.07)</td>
<td>0.624 (0.075)</td>
<td>0.585 (0.078)</td>
</tr>
<tr>
<td>(100, 50)</td>
<td>$TPR_{50}$</td>
<td>0.749 (0.037)</td>
<td>0.745 (0.035)</td>
<td>0.727 (0.035)</td>
</tr>
<tr>
<td>(100, 50)</td>
<td>$TPR_{100}$</td>
<td>1 (0)</td>
<td>1 (0)</td>
<td>1 (0)</td>
</tr>
<tr>
<td>(400, 8)</td>
<td>$TPR_{im}$</td>
<td>0.695 (0.032)</td>
<td>0.658 (0.033)</td>
<td>0.706 (0.029)</td>
</tr>
<tr>
<td>(400, 8)</td>
<td>$TPR_{0.5}$</td>
<td>0.721 (0.031)</td>
<td>0.684 (0.033)</td>
<td>0.744 (0.027)</td>
</tr>
<tr>
<td>(400, 8)</td>
<td>$TPR_{(1)}$</td>
<td>0.668 (0.028)</td>
<td>0.635 (0.028)</td>
<td>0.689 (0.025)</td>
</tr>
<tr>
<td>(400, 8)</td>
<td>$TPR_{1}$</td>
<td>0.687 (0.036)</td>
<td>0.651 (0.037)</td>
<td>0.702 (0.031)</td>
</tr>
<tr>
<td>(400, 8)</td>
<td>$TPR_{50}$</td>
<td>0.79 (0.017)</td>
<td>0.769 (0.018)</td>
<td>0.803 (0.016)</td>
</tr>
<tr>
<td>(400, 8)</td>
<td>$TPR_{100}$</td>
<td>0.999 (0.002)</td>
<td>1 (0)</td>
<td>1 (0.001)</td>
</tr>
<tr>
<td>(400, 50)</td>
<td>$TPR_{im}$</td>
<td>0.698 (0.037)</td>
<td>0.687 (0.039)</td>
<td>0.683 (0.032)</td>
</tr>
<tr>
<td>(400, 50)</td>
<td>$TPR_{0.5}$</td>
<td>0.73 (0.034)</td>
<td>0.725 (0.036)</td>
<td>0.717 (0.033)</td>
</tr>
<tr>
<td>(400, 50)</td>
<td>$TPR_{(1)}$</td>
<td>0.676 (0.031)</td>
<td>0.671 (0.033)</td>
<td>0.664 (0.029)</td>
</tr>
<tr>
<td>(400, 50)</td>
<td>$TPR_{1}$</td>
<td>0.697 (0.042)</td>
<td>0.686 (0.046)</td>
<td>0.676 (0.039)</td>
</tr>
<tr>
<td>(400, 50)</td>
<td>$TPR_{50}$</td>
<td>0.795 (0.02)</td>
<td>0.792 (0.021)</td>
<td>0.788 (0.019)</td>
</tr>
<tr>
<td>(400, 50)</td>
<td>$TPR_{100}$</td>
<td>1 (0.001)</td>
<td>1 (0)</td>
<td>1 (0)</td>
</tr>
</tbody>
</table>

Table 2
Prediction of VH mutation status: Estimated misclassification rates, TPR and FPR for $t = 0.5$. s.f. denotes stepwise forward selection of the predictors.

<table>
<thead>
<tr>
<th></th>
<th>$Err_T$</th>
<th>$Err_T^{CV}$</th>
<th>$Err_T^{(1)}$</th>
<th>$Err_T^{632}$</th>
<th>$Err_T^{632+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagging</td>
<td>0.000</td>
<td>0.175</td>
<td>0.217</td>
<td>0.137</td>
<td>0.163</td>
</tr>
<tr>
<td>Log. Reg.</td>
<td>0.000</td>
<td>0.336</td>
<td>0.395</td>
<td>0.250</td>
<td>0.352</td>
</tr>
<tr>
<td>PenalizedLR (sf)</td>
<td>0.101</td>
<td>0.218</td>
<td>0.287</td>
<td>0.219</td>
<td>0.243</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.000</td>
<td>0.168</td>
<td>0.211</td>
<td>0.133</td>
<td>0.157</td>
</tr>
<tr>
<td>SVM</td>
<td>0.071</td>
<td>0.163</td>
<td>0.210</td>
<td>0.158</td>
<td>0.170</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$TPR_T$</th>
<th>$TPR_T^{CV}$</th>
<th>$TPR_T^{(1)}$</th>
<th>$TPR_T^{632}$</th>
<th>$TPR_T^{632+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagging</td>
<td>1.000</td>
<td>0.871</td>
<td>0.819</td>
<td>0.885</td>
<td>0.868</td>
</tr>
<tr>
<td>Log. Reg.</td>
<td>1.000</td>
<td>0.673</td>
<td>0.596</td>
<td>0.744</td>
<td>0.634</td>
</tr>
<tr>
<td>PenalizedLR (sf)</td>
<td>0.900</td>
<td>0.784</td>
<td>0.714</td>
<td>0.782</td>
<td>0.758</td>
</tr>
<tr>
<td>Random Forest</td>
<td>1.000</td>
<td>0.878</td>
<td>0.823</td>
<td>0.888</td>
<td>0.872</td>
</tr>
<tr>
<td>SVM</td>
<td>0.940</td>
<td>0.848</td>
<td>0.805</td>
<td>0.855</td>
<td>0.843</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$FPR_T$</th>
<th>$FPR_T^{CV}$</th>
<th>$FPR_T^{(1)}$</th>
<th>$FPR_T^{632}$</th>
<th>$FPR_T^{632+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagging</td>
<td>0.000</td>
<td>0.222</td>
<td>0.254</td>
<td>0.160</td>
<td>0.197</td>
</tr>
<tr>
<td>Log. Reg.</td>
<td>0.000</td>
<td>0.346</td>
<td>0.385</td>
<td>0.243</td>
<td>0.338</td>
</tr>
<tr>
<td>PenalizedLR (sf)</td>
<td>0.102</td>
<td>0.219</td>
<td>0.288</td>
<td>0.220</td>
<td>0.244</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.000</td>
<td>0.214</td>
<td>0.245</td>
<td>0.155</td>
<td>0.189</td>
</tr>
<tr>
<td>SVM</td>
<td>0.182</td>
<td>0.173</td>
<td>0.224</td>
<td>0.172</td>
<td>0.184</td>
</tr>
</tbody>
</table>

and ROC curves for bundling classifiers as applied to microarray data by Croner et al. (2005) and for glaucoma diagnosis by Horn et al. (2007).

Our definition of the leave-one-out bootstrap estimation of the ROC curve follows the definition of the leave-one-out bootstrap estimation of the misclassification error. The bootstrap estimation of the ROC curve is defined by the set of all classification results for all out-of-bag samples. Hothorn et al. (2004) used this principle to propose an aggregated estimated survival tree. Schumacher et al. (2007) generalize the .632+ principle to the assessment of survival prediction models and were able to show that the .632+ principle works in a high-dimensional setting even for models when the apparent error is close to zero. Gerds et al. (2008) give an overview on the performance of risk prediction models.
Table 3
Glaucoma diagnosis: Estimated misclassification rates, TPR and FPR for threshold $t = 0.5$. s.f. denotes stepwise forward selection of the predictors.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{Err}$</th>
<th>$\hat{Err}^{CV}$</th>
<th>$\hat{Err}^{(1)}$</th>
<th>$\hat{Err}^{632}$</th>
<th>$\hat{Err}^{632+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaboost</td>
<td>0.000</td>
<td>0.149</td>
<td>0.162</td>
<td>0.102</td>
<td>0.116</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.000</td>
<td>0.161</td>
<td>0.181</td>
<td>0.115</td>
<td>0.132</td>
</tr>
<tr>
<td>PenalizedLR (sf)</td>
<td>0.138</td>
<td>0.189</td>
<td>0.200</td>
<td>0.177</td>
<td>0.180</td>
</tr>
<tr>
<td>RandomForest</td>
<td>0.000</td>
<td>0.143</td>
<td>0.154</td>
<td>0.097</td>
<td>0.110</td>
</tr>
<tr>
<td>SVM</td>
<td>0.087</td>
<td>0.133</td>
<td>0.149</td>
<td>0.126</td>
<td>0.128</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{TPR}$</th>
<th>$\hat{TPR}^{CV}$</th>
<th>$\hat{TPR}^{(1)}$</th>
<th>$\hat{TPR}^{632}$</th>
<th>$\hat{TPR}^{632+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaboost</td>
<td>1.000</td>
<td>0.869</td>
<td>0.826</td>
<td>0.890</td>
<td>0.874</td>
</tr>
<tr>
<td>Bagging</td>
<td>1.000</td>
<td>0.860</td>
<td>0.807</td>
<td>0.878</td>
<td>0.857</td>
</tr>
<tr>
<td>PenalizedLR (sf)</td>
<td>0.847</td>
<td>0.819</td>
<td>0.813</td>
<td>0.825</td>
<td>0.825</td>
</tr>
<tr>
<td>RandomForest</td>
<td>1.000</td>
<td>0.879</td>
<td>0.826</td>
<td>0.890</td>
<td>0.874</td>
</tr>
<tr>
<td>SVM</td>
<td>0.878</td>
<td>0.851</td>
<td>0.846</td>
<td>0.858</td>
<td>0.857</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>$\hat{FPR}$</th>
<th>$\hat{FPR}^{CV}$</th>
<th>$\hat{FPR}^{(1)}$</th>
<th>$\hat{FPR}^{632}$</th>
<th>$\hat{FPR}^{632+}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaboost</td>
<td>0.000</td>
<td>0.165</td>
<td>0.150</td>
<td>0.095</td>
<td>0.106</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.000</td>
<td>0.179</td>
<td>0.169</td>
<td>0.107</td>
<td>0.122</td>
</tr>
<tr>
<td>PenalizedLR (sf)</td>
<td>0.122</td>
<td>0.198</td>
<td>0.213</td>
<td>0.180</td>
<td>0.185</td>
</tr>
<tr>
<td>RandomForest</td>
<td>0.000</td>
<td>0.162</td>
<td>0.134</td>
<td>0.085</td>
<td>0.094</td>
</tr>
<tr>
<td>SVM</td>
<td>0.051</td>
<td>0.118</td>
<td>0.144</td>
<td>0.110</td>
<td>0.115</td>
</tr>
</tbody>
</table>

Acknowledgements

The authors thank the three referees and the associate editor for helpful comments and suggestions which improved the manuscript. The research was supported by the Deutsche Forschungsgemeinschaft (DFG), collaborative research area SFB 539-A4/C1.

Appendix. Supplementary data


References


