

Max-Planck-Institut  
für Mathematik  
in den Naturwissenschaften  
Leipzig

An experimental evaluation of boosting methods  
for classification

by

*Rainer Stollhoff, Willi Sauerbrei, and Martin Schumacher*

Preprint no.: 7

2010





# An experimental evaluation of boosting methods for classification

Rainer Stollhoff<sup>1</sup>, Willi Sauerbrei<sup>2</sup>, Martin Schumacher<sup>2</sup>

<sup>1</sup> Max-Planck Institute for Mathematics in the Sciences, Leipzig, Germany

<sup>2</sup> Institute of Medical Biometry and Medical Informatics, University Medical Center, Freiburg, Germany

## Summary

### Objectives:

In clinical medicine, the accuracy achieved by classification rules is often not sufficient to justify their use in daily practice. In order to improve classifiers it has become popular to combine single classification rules into a classification ensemble. Two popular boosting methods will be compared with classical statistical approaches.

### Methods:

Using data from a clinical study on the diagnosis of breast tumors and by simulation we will compare AdaBoost with gradient boosting ensembles of regression trees. We will also consider a tree approach and logistic regression as traditional competitors. In logistic regression we allow to select non-linear effects by the fractional polynomial approach. Performance of the classifiers will be assessed by estimated misclassification rates and the Brier score.

### Results:

We will show that boosting of simple base classifiers gives classification rules with improved predictive ability. However, the performance of boosting classifiers was not generally superior to the performance of logistic regression. In contrast to the computer intensive methods the latter are based on classifiers which are much easier to interpret and to use.

### Conclusions:

In medical applications, the logistic regression model remains a method of choice or, at least, a serious competitor of more sophisticated techniques. Refinement of boosting methods by using optimized number of boosting steps may lead to further improvement.

### Keywords:

Classification, simulation study, boosting, generalized additive models, diagnosis of breast tumors.

## 1 Introduction

Classification or class prediction is an ubiquitous task in clinical medicine. Some recent examples include the prediction of the presence of metastases in cancer patients [1] the differentiation between malignant, benign and normal tissue in women undergoing digital mammography [2], the identification of patients with a history of stroke [3], the prediction of prolonged hospital stay in an intensive care unit [4], or the detection of glaucoma [5], among others. Often, however, classification accuracy or predictive ability of the derived classification or prediction rules is not sufficiently large to justify their use in daily practice [1]. In that situation one could either look for additional, more informative factors that could be included into the classification rule or to employ more sophisticated techniques that go beyond a classical statistical regression or tree-based approach. These may range from artificial neural networks [6], Bayesian network models [7] machine learning approaches or flexible regression models. In this paper, we concentrate on two popular boosting approaches that combine single, simple classification rules into a classification ensemble in order to improve the predictive ability. For these boosting approaches it is believed that they provide an “impressive improvement in performance that seems to be associated with boosting’s resistance to overfitting” [8]. Therefore, we compare them with standard logistic regression and some extensions and a tree-based classification rule as traditional competitors. We investigate the various approaches by means of a large-scaled simulation study and data from a study on the differentiation of benign and malignant breast tumors [9].

In the sequel, we consider the following supervised classification problem, where one is given a set of observations of a pair of random variables  $(X, Y)$ , often called input and output variable. Whereas  $X$  is usually a vector of both binary and continuous random variables, the range of  $Y$  is considered finite and the values correspond to distinct classes. Using this set of observations  $(x_i, y_i)$ ,  $(i = 1, \dots, n)$ , termed a training set, the aim is to find a classification rule  $C$  that for a new realization  $(x, y)$  maps the input variable  $x$  to an output class  $\hat{y} = c(x)$ , such that  $\hat{y} = y$ . In this paper we limit the class structure to a binary random variable  $Y$  taking values 0 or 1 and denote by  $p(x)$  the conditional probability of belonging to class 1 given the input vector  $x$ ,  $P(Y = 1 | X = x)$ . Since in most cases the classes can not be perfectly separated, i.e.  $p(x) \in (0, 1)$  for all  $x$ , we are interested in estimates  $\hat{p}(x)$  of the conditional class probability, as well as the classification  $\hat{y}$  itself.

During the last decade the availability of computing resources has led to the development of new methods better able to deal with increasing sample sizes and input dimensionality. Some of these classification methods are based on the idea of combining single classification rules into a classification ensemble to improve the predictive performance. Following [10] these so called ensemble methods can be roughly divided in three families: Boosting [11], Bagging [12] and Randomization (e.g. random forests [13]). For a review on ensemble methods see [14].

The idea of boosting methods e.g. AdaBoost [15] is to iteratively apply a classifier, in the following called base classifier, to re-weighted versions of the original training set to construct a series of base classification rules. After every iteration all previously obtained base classification rules are then combined to an ensemble e.g. using a majority vote on the class membership. The base classification rules in this ensemble need not be good predictors by themselves; Important is the combination strategy used. Further theoretical work [16,17] has shown that boosting methods belong to the family of generalized additive models [18]. The additive terms and their coefficients correspond to the single classification rules and their weights respectively. This has led to the development of a whole family of boosting methods, often referred to as gradient boosting methods [19]. In this paper we illustrate the merits of and differences between two boosting algorithms: AdaBoost [15] (ada) and gradient boosting [19] (gbm) and compare their results with logistic regression, with and without considering non-linear functions, and with a tree approach.

Often, interpretability is an important aim of an classifier, at least in medicine [20]. Rules derived from a diagnostic study should make sense and for general usefulness they have to be transportable to other settings, otherwise they will be ‘quickly forgotten’, as discussed for prognostic models in Wyatt & Altman [21]. Therefore simpler classification rules may be preferable even if the performance criteria are slightly worse [4]. This issue is also expressed by Terrin et al. [22], who compare the external validity of predictive models derived with logistic regression, classification trees and neural networks.

The paper is organized as follows: The first section introduces the classifiers studied and different criteria for measuring prediction performance, i.e. error rate and Brier score. The second section shows results of the classifiers applied to real data. The problem posed was to construct a diagnostic rule that discriminates between benign and malignant breast tumor. We found striking discrepancies in the results depending on the performance criteria used as well as differences between the tested combinations of boosting algorithm and other classifiers. In

a simulation study we will compare their performance to the logistic regression (logreg) model and to classification trees (tree). In the logistic model we will also consider the fractional polynomial (mfp) approach [23], which extends logreg by determining possible non-linear effects of continuous covariates in a systematic and controlled way. The basic simulation design considers six variables with influence on the outcome. In two extensions we add interactions and non-linear effects, respectively; variable selection will not be considered. Finally, we discuss the results of the example data and of the simulations and give some general conclusions.

## 2 Methods

### 2.1 Classifiers

All of the classifiers studied belong to the family of generalized additive models [18]. A generalized additive model for the conditional class probabilities  $p(x)$  can be written as

$$p(x) = g\left(\sum_{j=1}^J \beta_j h_j(x)\right), \quad (1)$$

where the  $h_j$  and  $\beta_j(1, \dots, J)$  are real-valued functions of the input variables and their corresponding regression coefficients, respectively. The link function  $g$  is chosen to map onto  $[0,1]$ , e.g. by taking the inverse of the logit function  $g^{-1}(p(x)) = \log \frac{p(x)}{1-p(x)}$ . The

functions  $h_j$  are usually chosen as functions of only a small subset of the input variables and are often all members of the same parametric family, each characterized by a parameter vector  $\gamma_j$ . The coefficients  $\beta_j$  as well as the functions  $h_j$  or rather the parameter vectors  $\gamma_j$  of the model are then estimated from the training set.

A possible classification rule is to calculate the estimated probability  $\hat{p}(x)$  for a given observation with input vector  $x$  and compare it with an appropriately chosen classification threshold  $t$ . The observation is assigned to class 1 if  $\hat{p}(x) \geq t$ .

#### 1.1.1. Logistic Regression

Logistic regression models probably constitute the approach used most often in medical statistics. A logistic regression model uses the inverse logit transformation as the link function and often assumes linear projections for each of the input variables, i.e.  $h_j(x) = x_j$  for

$(j = 1, \dots, k)$ , where the subscript denotes projection to the  $j$ -th component and  $k$  is the dimension of the input vector  $x$ .

Here we always fit linear logistic regression models (`logreg`) by maximum likelihood estimation. We do not apply variable selection strategies.

### 1.1.2. Multivariate Fractional Polynomials

The assumption of log-linearity in linear logistic regression models can be relaxed. One way is to take non-linear transformations of a single input variable  $x_j$  as functions  $h_j$ . Fractional polynomials [23] limit the transformations to a fixed set of polynomials with fractional powers of low degree and the logarithmic transformation. Here we use multivariate fractional polynomials [24] (`mfp`). All variables are included. The variable transformations are determined according to the RA2 algorithm described in [25], with 2 as the maximal degree of the fractional polynomials and 0.05 as the significance level for the selection of the function.

### 1.1.3. Classification and Regression Trees

Classification and Regression trees [26] (`tree`) partition the space spanned by the input variables into  $J$  disjoint regions  $R_j, (j = 1, \dots, J)$  and assign to each region a class label or regression value, respectively. For classification trees conditional class probabilities  $p_j := P_j(y = 1 | x \in R_j)$  can be estimated by taking the ratio of class 1 training set observations in  $R$  divided by the total number of training set observations in  $R$ . The classification rule of a tree can be written as a generalized additive model:

$$p(x) = \sum_{j=1}^J p_j 1(x \in R_j), \quad (2)$$

where  $1(x \in R_j)$  is the indicator function taking the value one if  $x \in R_j$  and zero otherwise.

The regions  $R_j$  are obtained by a series of univariate, binary splits into regions with lower impurity, where impurity refers to class separation within a region. We choose the Gini-index as the impurity measure and stop splitting if it would lead to regions either containing less than five training set observations or to a decrease in overall impurity of less than 1%. We do not prune the trees.

#### 1.1.4. Boosting

In this study we contrast AdaBoost ensembles of classification trees with gradient boosting ensembles of regression trees. In AdaBoost ensembles the functions  $h_j$  are the iteratively constructed base classification rules and the regression coefficients  $\beta_j$  are a function of the weighted error rates of the  $h_j$ . The link function is the inverse of the logit function. We used the AdaBoost.M1 algorithm as given in [15]. In gradient boosting ensembles the  $h_j$  are regression functions and are iteratively fitted against the residuals of the previous ensemble using the inverse of the logit as link function. We therefore choose the corresponding log-likelihood as loss function. The base classifiers, resp. regression functions of the boosting ensembles are either classification/regression stumps consisting of a single binary split or small classification/regression trees with a maximum of three consecutive splits, further referred to as `ada.stump` and `ada.tree` and accordingly `gbm.stump` and `gbm.tree`. The number of trees, resp. boosting iterations is limited to 400. We used an early stopping rule, halting the iterations if the error (misclassification error or logscore resp.) on the training set doesn't change significantly over 20 iterations. For gradient boosting we also applied a global shrinkage factor of 0.1. Pre-simulation investigations showed that the use of early stopping and shrinkage leads to performance close to the optimum achievable in 400 iterations. This was also confirmed in a post-hoc analysis of the results. For all designs the difference in performance using the stopped and the optimal number of iterations for the boosting ensemble was around or below the sample standard deviation of the reported mean performance. Only for `gbm.stump` and only in Designs B and C the optimal number of iterations was 400, i.e. performance was increasing up to the maximum number of iterations.

## 2.2 Performance criteria for comparison

Misclassification rate or error rate is probably the performance criterion used most often to compare different classification methods and various methods have been proposed for error rate estimation [27,28,29]. Restriction to error rate as the only measure of prediction accuracy is sufficient, if the class membership is determined by the input variables i.e. the conditional probabilities for class membership only take values 1 or 0.

To evaluate the estimates for the conditional class probabilities we use the Brier or quadratic

score, which is defined as:  $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{p}(y_i))^2$ , where  $\hat{p}(x)$  denotes the estimated probability for belonging to class 1 given the input variables  $x$  and the summation is over all



test set observations. It can be easily shown [28] that for two different estimates  $\hat{p}_1$  and  $\hat{p}_2$  their difference in precision [29] as measured by  $E_x \left[ (p(x) - \hat{p}_1(x))^2 \right] - E_x \left[ (p(x) - \hat{p}_2(x))^2 \right]$  equals their difference in expected Brier score. Note that by using the Brier score differences in precision can be estimated without knowledge of the true probabilities  $p(x)$ . In addition to giving an estimate of the performance of every single classification method, as could be obtained by using other criteria, e.g. log-likelihood, a plot of Brier scores also enables a direct comparison of the precision of the estimated conditional probabilities, for a comprehensive discussion see [30]. We also computed the log-likelihood, as well as the area under the ROC curve. They lead to similar results [31].

In the example we give resubstitution estimates, i.e. using the training data twice to develop the rule and to assess its performance, and/or estimates using 10-fold crossvalidation for both criteria. In the simulations we give the mean values obtained by 50 repetitions for both criteria as well as upper bounds on the sample standard deviations.

### 2.3 Algorithms

All computations were performed using the statistical programming language R (version 1.9.1) [32]. Specifically we chose `glm()` for implementation of logistic regression, `fracpoly(version 1.1.0)`, a previous version of the now available `mfp` package [33], for multivariate fractional polynomials, `rpart` (version 3.1-19) [34] for classification and regression trees, `gbm` (version 1.2) [35] for gradient boosting. At the time the simulations were performed, there was no R-package for the original AdaBoost.M1[15] algorithm. We implemented the AdaBoost.M1 algorithm in R using `rpart` to construct single classification trees. The following control parameter settings were used:

`rpart.control: minsplit=10,minbucket=5,cp=.01,xval=0,maxsurrogate = 0, usesurrogate=0`

AdaBoost, `rpart.control: same as above with additional maxdepth = 3 (or 1 resp.)`

`gbm: interaction.depth=3 (or 1 resp.) ,n.minobsinnode=5,shrinkage=.1`

### 3 Example: Breast Tumor Diagnosis

We used the classifiers described above to construct a diagnostic rule that differentiates between benign and malignant breast tumors. The data set contains measurements of 133 cancer patients, and 325 woman with benign tumors, the prevalence is thus 29%. Class labelling is 1 for malignant and 0 for benign tumors. The input variables are the age of the

patient, the number of tumor arteries (ipsi- and contralateral) and the maximum, average and sum of all peak systolic flow velocities, obtained by Doppler sonography. Some of these variables are strongly correlated. The class membership (malign and benign) was determined afterwards and independent of the earlier measurements by histologic and cytologic diagnosis. A complete description and detailed analysis of the data set is given in [9]. Here we only include variables that could be calculated in all patients to avoid missing values. An analysis including all variables and excluding 72 patients with missing values for some variables gives similar results [31].

### 3.1 Results

All methods were first applied to the whole data set without setting aside a separate test data set. Figure 1 shows boxplots of the estimated conditional class probabilities for a malignant tumor, separately for patients with benign (0) and malignant (1) tumors. As can be seen most classifiers achieve a good or even perfect separation of the two classes for a wide range of possible classification thresholds.

Since it is well-known that resubstitution estimates systematically overestimate the true performance [36], especially for flexible classification methods [27], we used 10-fold crossvalidation, which has been shown to provide more reliable estimates [37]. The Brier score was estimated by cross-validation as 0.06 to 0.07 for all seven methods pointing to similar predictive performance in this data. This is in sharp contrast to the impression that one would get from figure 1 that is reflected in much smaller values of the resubstitution estimates of the Brier score ranging from 0.001 to 0.05.

Results indicate an overoptimistic performance that is only small for logreg and mfp, severe for tree and extreme for the boosting approaches. Thus this example shows that boosting approaches may not generally lead to improved prediction performance and may also be prone to overfitting, at least to some extent.

## 4 Simulation study

In order to investigate and illustrate the differences between the classifiers used to construct a diagnostic rule, we performed a simulation study to obtain more general results. We chose the design of the study such that it highlights the different characteristics of the classifiers studied, e.g. trees are better suited to detect interactions and mfp aims to model non-linear effects of continuous covariates. Hence it can and should be regarded as being deliberately biased.

## 4.1 Design

Because all of the methods belong to the family of generalized additive models we use a logistic additive model for the simulation studies. This allows a direct comparison of simulation results with conjectures following theoretical analysis. Table 1 gives the three simulation designs used in this study.

The first design is a simple log-linear model incorporating six random variables: Three normal distributed continuous variables and three Bernoulli distributed binary variables. See Table 1 for a detailed description. The coefficients were chosen such that the univariately explained fraction of variation of  $\log \frac{p}{1-p}$  was approximately 0.3, 0.15, 0.05 for  $X_1, \dots, X_6$  respectively. To study for varying separability [29] of the classes we multiplied by a factor  $\lambda_{lin}$  that was lowered stepwise from 1 to  $\frac{1}{8}$ . Note that the variation of  $\log \frac{p}{1-p}$  is proportional to  $\lambda_{lin}^2$  and therefore the separability of the classes scales with  $\lambda_{lin}$ , in the following referred to as a scaling factor. The intercept term  $\lambda_0$  was varied to obtain equally sized classes.

In the second design, we include six additional random variables: Again three Bernoulli and three normal distributed variables with different means and variances. These variables are then multiplied to form three interaction terms and the corresponding coefficients were chosen such that each interaction term has approximately equal variance. The scaling factors  $\lambda_{lin}$  and  $\lambda_{prod}$  were chosen as to balance the influence of linear and interaction terms. For  $\lambda_{lin}=\lambda$  the sum of linear and the sum of interaction terms contribute almost equally to the variation of  $\log \frac{p}{1-p}$ .

Since we suspect that in more realistic scenarios not all variables have a linear effect, in Design-C we include non-linear transformations of three additional random variables. Two of the variables are normal distributed, the other is drawn from a  $\chi^2$  distribution. The transformations chosen are a quadratic and a cubic polynomial for the two normal distributed variables as well as the logarithmic transformation for the  $\chi^2$  distributed variable. Again all non-linear terms have approximately equal variance and the contribution of their sum is

balanced, s.t. the sum of linear terms and the sum of non-linear terms contribute almost equally to the variation of  $\log \frac{p}{1-p}$ .

Table 2 gives the values used for the pairs and  $(\lambda_{lin}^2, \lambda_{prod}^2)$  and  $(\lambda_{lin}^2, \lambda_{trans}^2)$  respectively. These determine the relative influence of the linear terms and the non-linear or interaction terms. By varying these scaling factors we can thus study the performance of the classifiers over a wide range of scenarios. Starting with situations in which only linear terms influence the class membership, up to situations in which the class membership is largely determined by non-linear transformations of the variables or bivariate interactions. In Table 2 and further on we choose to display the squares of the scaling factors, since the variation of  $\log \frac{p}{1-p}$  is proportional to the squares of the scaling factors. In the following we only give values for  $\lambda_{lin}^2$  and omit denoting the values of  $\lambda_{prod}^2$  and  $\lambda_{trans}^2$  for ease of notation; they can be looked up in Table 2. Note that for all pairs  $(\lambda_{lin}^2, \lambda_{prod}^2)$  and  $(\lambda_{lin}^2, \lambda_{trans}^2)$  resp. the variation of  $\log \frac{p}{1-p}$  and therefore the class separability is almost the same.

For all simulations and choice of parameters we constructed 50 training sets of size 1000 each and a test data set consisting of 10.000 observations of the variables. In [31] we also investigated the influence of the training set size. Resulting differences in separability of class membership between the 50 training sets are small compared to the differences in performance between the classifiers.

The true class membership of every observation given its input variables is chosen randomly with probability according to the specific model given for  $\log \frac{p}{1-p}$ .

## 4.2 Results

### 1.1.5. Design-A

Figure 2 displays the mean performance in terms of error rate and Brier score of the boosting and tree classifiers, plotted as a function of the squared scaling factor  $\lambda_{lin}^2$ . Also given is the performance obtained using the true conditional class probabilities, i.e. Bayes classification. Here and in all other results reported, the sample standard deviations (SDs) of the means of both performance criteria are comparably small. They will not be plotted for the sake of

visibility, but upper bounds will be given in the captions, with a separate value for classification trees, which show a more variable performance than all other methods studied.

For all values of  $\lambda_{lin}^2$  boosting ensembles of decision stumps show better performance than tree ensembles. Whereas the boosting ensembles do not seem to differ in error rates, the Brier score shows big differences between AdaBoost and gradient boosting, regardless of tree size. The latter have smaller values of the Brier score. Note that in models with low scaling factor, i.e. low class separability the performance of the ensembles can be worse than that of single trees.

Since the simulation model is log-linear, logistic regression produces estimated probabilities indistinguishable from the true conditional class probabilities. Because the multivariate fractional polynomials algorithm always selected the variables as linear terms the estimated models are exactly the same as those of logistic regression. We did not include these methods with the best performance in the figure for the sake of clarity. When looking at the error rates (not shown) we obtained very similar results.

#### 1.1.6. Design-B

The second series of simulations investigates the differences between the approaches in the presence of interactions. Table 3 shows the mean performance, in terms of the Brier score of all the classifiers investigated, when the linear terms explain 0.75 of the variation of  $\log \frac{p}{1-p}$ . For both criteria, error rate and Brier score, the classifier developed with `logreg` (identical to `mfp`) comes closest to the Bayes Classifier using the true conditional class probabilities. Results of the ensemble classifiers are a bit worse, performance of the tree classifier is much worse.

Figure 3 shows a plot of the mean performance of the classifiers investigated as a function of the squared scaling factor  $\lambda_{lin}^2$ .

Since decision stumps - consisting of a single univariate split - are in principle unable to incorporate multivariate interactions, the performance of stump ensembles decreases as the contribution of the bivariate interaction terms resp.  $\lambda_{prod}^2$  increases and the contribution of the univariate linear terms resp.  $\lambda_{lin}^2$  decreases. Clearly the performance of logistic regression also worsens with decreasing  $\lambda_{lin}^2$ . Single classification trees can easily model

interactions, the performance increase at the borders of the plot is probably due to the concentration of explained variation of  $\log \frac{p}{1-p}$  on a small number of variables, which means that a small number of 'strong' predictors dominate the model.

#### 1.1.7. Design-C

The last simulation study compares the classifiers if some variables have non-linear effects. The mean performance of all the classifiers investigated, is very similar to that observed in Design B in the corresponding situation.

Increasing the influence of the non-linear terms does not affect performance of both kinds of stump ensembles, as can be seen in Figure 4. This also holds true for tree ensembles, which are not shown for the sake of clarity.

On the other hand the performance of the logistic regression decreases with increasing influence of non-linear terms. However the multivariate fractional polynomials did choose the true variable transformations in almost all repetitions leading to good performance.

#### 1.1.8. Differences between boosting ensembles

To emphasize the differences between AdaBoost and Gradient Boosting we investigated the estimated conditional class probabilities in the simplest situation of Design-A with scaling factor  $\lambda \frac{2}{lin} = 1$  in more detail. The upper part of Figure 5 plots the quantiles of the estimated conditional probabilities against the true conditional probabilities for every repetition. The `ada.tree` estimates of the probabilities are far from the decision threshold at .5, more often than given by the true probabilities. This means that the estimates are "overconfident". On the other hand the `gbm.tree` estimates show a distribution comparable to that of the true class probabilities. This behavior can also be seen in a plot of all pairs  $(p(x_1), \hat{p}(x_1))_{i=1, \dots, n}$  for a single simulation run given in the lower part of Figure 5. Whereas the estimates of `gbm.tree` are scattered evenly around the diagonal, the `ada.tree` estimates are mainly close to 0 or 1. However, since both boosting ensembles misclassify a comparable number of observations - i.e. the number of points in the lower right and upper left corner of the lower part of Figure 5 - this does not lead to big differences in error rate.

## 5 Discussion

The real world problem of constructing a diagnostic rule to differentiate between malignant and benign breast tumors showed that simple classifiers can be improved by combining them into an ensemble using boosting methods. One has to be careful though in measuring the

performance of such boosting ensembles. In our case an ensemble of classification trees combined using gradient boosting showed perfect classification performance on the training set. When evaluated using cross validation, however, the performance dropped to levels comparable to those of common classifiers such as logistic regression. This is due to the underestimation of the true error of flexible classifiers if using resubstitution estimates [27]. Based on cross validation estimates the small advantage in performance of the boosting methods are outweighed by the difficulty in interpretation and general use of ensemble methods. In the original analysis [9] Sauerbrei et al. used variable selection methods and investigated for possible non-linear effects of continuous variables. A simple classifier using age, ipsi- and contro-lateral arteries and a threshold of 0.29 for the probability had an apparent error rate of 0.063. Using a bootstrap approach the estimated overoptimism was even smaller (about 0.004) than the estimated overoptimism from logreg and mfp in this study.

The comparison of the different classifiers and especially the comparison between the two different boosting algorithms studied depends strongly on the performance criterion used. If judged only by the error rate the simulations in this study do not show big differences between AdaBoost and gradient boosting. However, error rate is not a strictly proper measure of precision [29], i.e. the error rate is not uniquely minimized by the true conditional class probabilities. The use of a simulation study enabled us to assess the performance of the classifiers in estimating the true conditional class probabilities. Looking at the precision of the classifiers by comparing their Brier scores, gradient boosting clearly outperforms AdaBoost in all of the simulation studies investigated. A more detailed analysis of the linear case revealed that AdaBoost tends to produce over-confident estimates of the conditional probabilities.

In our implementation of AdaBoost we used an early stopping rule, halting the iteration if the ensembles error on the training set doesn't change significantly over 20 iterations. Although the exact form of the early stopping rule doesn't seem very important, as long as early stopping is implemented [1, 10, 39], it could be that a more deliberate choice of stopping rule could counter this over-confidence.

The performance of boosting ensembles depends on the difficulty of the problem under investigation. This was studied in Design-A by multiplying the logit-transform of the true conditional probabilities with a decreasing scaling factor  $\lambda \frac{2}{lin} \in [0,1]$  thereby lowering the true conditional probabilities towards the decision threshold at .5. We observed that the performance of boosting ensembles decreased more than that of the single classification tree. It is important to note that although one arrives at the same conclusion this definition of

difficulty differs from random permutation of the class labels as used in other simulation studies [8,17,10]. In all models of this study regions of the input space far from the decision hyperplane still tend to have true conditional class probabilities close to zero or one respectively, which is not the case for random permutation of class labels.

In an additional study [31] we also investigated the inclusion of noise by providing not the true input variables but only correlated ones in the training sets and obtained comparable results.

Although decision stumps consisting of only a single univariate split are not as good classifiers as larger decision trees, an ensemble of stumps can outperform an ensemble of trees. In the linear and non-linear univariate models studied in Design-A and Design-C stumps seem to be better suited as base classifier than fully grown trees, whereas in Design-B they are gradually outperformed by trees as more and more emphasis is put on bivariate interaction terms. This is in accord with theoretical work [38,17] suggesting that in an ensemble classification trees with a small amount of nodes might be sufficient, if the dimension of the input space is low or if the influence of multivariate terms is limited.

Whereas here the classification trees used as single classifiers are fully grown, unpruned trees with a minimum observation number in every final node, the size of the trees used as base classifiers in boosting ensembles was in advance fixed to one resp. three subsequent splits. While fixing this number is a requirement of the R-implementation `gbm()` used in this study, other methods of limiting the size of trees, e.g. pruning have also been studied in the context of boosting [39].

We investigated several ensemble approaches, but also compared their performance to standard statistical techniques. In the breast cancer example the severe overoptimism of the tree approach was obvious from the estimated cross-validated error rates. The overoptimism was even more pronounced for the ensemble methods, but relatively small for logistic regression. The good performance of the logistic regression was also confirmed in the simulation studies, even for design B with interactions included. In design C we also included non-linear effects of continuous predictors, a usual phenomenon in real data. Stronger non-linear effects were detected by the MFP approach, which resulted in improved predictions. In contrast to classifiers based on ensemble methods trees are comprehensible for clinicians [4], see for example the tree derived for the breast tumor diagnosis data [9]. Models from logistic regression, with or without transformed predictors, derive a diagnostic index which can easily be transferred to a probability of disease. If the model is not too complex, e.g. by including



complicated terms, they are easy to interpret and to use in general classification problems. This is certainly an undoubtful advantage of them. Concerning performance assessment, measured by error rate and the Brier Score, they even outperformed the ensemble methods used here. Such a good performance of logistic regression was not always found in other studies [40]. However, investigations on classifiers usually prefer to consider error rates of benchmark data sets. We consider simulation studies, which allow to examine the methods over a wide range of scenarios, carefully chosen such that they might reflect typical situation encountered in applications, having the advantage that the underlying data-generating mechanism is known. For the comparison of classifiers, we used the Brier score as a more subtle performance measure that takes the predicted probabilities of each individual into account. In a similar way, likelihood-based criteria like the Kullback-Leibler information could be used as well [28,29]. A limitation of our investigations clearly is that we did not refine the boosting methods by using an optimized number of boosting steps which should lead to further improvement. Instead, our major aim was to compare two popular boosting approaches without further “fine-tuning” to traditional competitors. The results of our investigation underline that this “fine-tuning” may absolutely necessary for achieving a substantial gain in accuracy.

## 6 Conclusions

Application of boosting to combine simple base classifiers into ensembles can lead to classification rules with improved prediction ability. Theoretical analysis [41,42] lead to consistency results for boosting ensembles for a wide class of base classifiers. Comparisons of boosting to other ensemble methods [10,43] has shown that in the presence of noise random ensembles, e.g. random forests [13] give more favorite results. Also using more robust base classifiers [44] or a stochastic gradient descent [45] can improve the performance of boosting algorithms.

However, in real analysis of medical data not only the lack of interpretation and applicability is an important disadvantage of using boosting classifiers. In our study of interacting variables the performance of logistic regression was at least comparable to the computer intensive approaches even if interaction terms contributed equally wrt. linear terms to the total variation, a strong deviation from the linear model assumptions.

This result supports the assessment of Hand concerning the illusion of progress from complex classifiers [46]. But there may be some room for improvement by further fine-tuning of the boosting approaches that has been beyond the scope of our investigations.

## References

1. Schwarzer G, Nagata T, Mattern D, Schmelzeisen R, Schumacher M: Comparison of fuzzy inference, logistic regression, and classification trees (CART) for the prediction of cervical lymph node metastasis in carcinoma of the tongue. *Methods of Information in Medicine* 2003;42(5):572-577.
2. Campos LF, Silva AC, Barros AK. Independent component analysis and neural networks applied for classification of malignant, benign and normal tissue in digital mammography. *Methods of Information in Medicine* 2007;46(2):212-5.
3. Tjortjis C, Saraee M, Theodoulidis B, Keane JA. Using T3, an improved decision tree classifier, for mining stroke-related medical data. *Methods of Information in Medicine* 2007;46(5):523-9.
4. Verduijn M, Peek N, Voorbraak V, de Jonge E, de Mol BA. Modeling length of stay as an optimized two-class prediction problem. *Methods of Information in Medicine* 2007;46(3):352-9.
5. Adler W, Peters A, Lausen B: Comparison of classifiers applied to confocal scanning laser Ophthalmoscopy data. *Methods of Information in Medicine* 2008;47:38-46.
6. Linder R, König IR, Weimar C, Diener HC, Pöpl SJ, Ziegler A. Two models for outcome prediction - a comparison of logistic regression and neural networks. *Methods of Information in Medicine* 2006;45(5):536-40.
7. Sakai S, Kobayashi K, Nakamura J, Toyabe S, Akazawa K. Accuracy in the diagnostic prediction of acute appendicitis based on the Bayesian network model. *Methods of Information in Medicine* 2007;46(6):723-6.
8. Bühlmann P, Yu B. Boosting with the L2-Loss. *Regression and Classification*. *Journal of the American Statistical Association* 2003;98:324-339.
9. Sauerbrei W, Madjar H, Prömpeler HJ: Use of logistic regression and a classification tree approach for the development of diagnostic rules: Differentiation of benign and malignant breast tumors based on color doppler flow signals. *Methods of Information in Medicine* 1998;37:226-234.
10. Dietterich T. An experimental comparison of three methods for constructing ensembles of decision trees. *Machine Learning* 2000;40:139-157.
11. Freund Y. Boosting a weak learning algorithm by majority. *Information and Computation* 1995;121(2):256-285.
12. Breiman L. Bagging Predictors. *Machine Learning* 1996;26(2):123-140.
13. Breiman L. Random forests. *Machine Learning* 2001;45(1):5-32.
14. Dietterich TG. Ensemble learning. In *The Handbook of Brain Theory and Neural Networks*, Second Edition. Cambridge: MIT Press, MA; 2002.
15. Freund Y, Schapire R. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996;148-156.
16. Breiman L. Arcing Classifiers. *Annals of Statistics* 1998;26(3):801-849.
17. Friedman J, Hastie T, Tibshirani R. Additive logistic regression: A statistical view of boosting. *Annals of Statistics* 2000;2:334-374.
18. Hastie T, Tibshirani R. *Generalized Additive Models*. London: Chapman & Hall; 1990.

19. Friedman J. Greedy function approximation: a gradient boosting machine. Technical Report. Department of Statistics, Stanford University; 1999.
20. Marshall RJ. Comparison of misclassification rates of search partition analysis and other classification methods. *Statistics in Medicine* 2006;25:3787-3797.
21. Wyatt JC, Altman DG. Prognostic models: clinically useful or quickly forgotten? *British Medical Journal* 1995;311:1539-1541.
22. Terrin N, Schmid CH, Griffith JL, D'Agostino RB, Sr, Selker HP. External validity of predictive models: A comparison of logistic regression, classification trees, and neural networks. *Journal of Clinical Epidemiology* 2003;56:721-729.
23. Royston P, Altman D. Regression using fractional polynomials of continuous covariates: Parsimonious parametric modelling (with discussion). *Applied Statistics* 1994;43(3):429-467.
24. Sauerbrei W, Royston P. Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society, Series A* 1999;162:71-94.
25. Sauerbrei W, Royston P. Corrigendum to: Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society, Series A* 2002;165:399-400.
26. Breiman L, Friedman J, Olshen R, Stone C. *Classification and Regression Trees*. Wadsworth, CA: 1984
27. Wehberg S, Schumacher M: A comparison of nonparametric error rate estimation methods in classification problems. *Biometrical J*, 2004;46:35-47.
28. Hand D. Measuring diagnostic accuracy of statistical prediction rules. *Statistica Neerlandica* 2000;53:3-16.
29. Hand D. *Construction and Assessment of Classification Rules*. N.Y.: John Wiley; 1997.
30. Gerds TA, Cai T, Schumacher M: The performance of risk prediction models. *Biometrical Journal* 2008;50:457-479.
31. Stollhoff R. Verbesserung von Klassifikationsverfahren durch Boosting bei binärer Zielgröße. (Improvement of classification approaches for a binary outcome by using boosting.) Diploma thesis, in German, Department of Mathematics, Albert-Ludwigs-Universität Freiburg i.Br.; 2004.
32. Ihaka R, Gentleman R. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 1996;5:299-314.
33. Ambler G, Benner A. Software R contributed package: Multivariate fractional polynomials. 2006 (<http://cran.r-project.org>).
34. Therneau TM, Atkinson B. Software R Contributed Package: Recursive Partitioning. 2002 (<http://cran.r-project.org>)
35. Ridgeway G. Software R Contributed Package: Generalized Boosted Regression Models. 2006 (<http://cran.r-project.org>)
36. Efron B. How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association* 1986;394(1):461-470.
37. Molinaro A, Simon R, Pfeiffer R. Prediction error estimation: a comparison of resampling methods. *Bioinformatics* 2005;21(15):3301-3307.

38. Breiman L. Some infinity theory for predictor ensembles. Technical Report 577, Statistics Department, University of California at Berkeley; 2000.
39. Drucker H. Effect of pruning and early stopping on performance of a boosting ensemble. *Computational Statistics & Data Analysis* 2002;38:393-406.
40. Hothorn T, Lausen B. Bundling classifiers by bundling trees. *Computational Statistics & Data Analysis* 2005;49:1068-1078.
41. Jiang W. Process Consistency for AdaBoost. *Annals of Statistics* 2004;32(1):13-29.
42. Lugosi G, Vayatis N. On the Bayes-risk consistency of regularized boosting methods. *Annals of Statistics* 2004;32(1):30-55.
43. Hamza M, Laroque D. An empirical comparison of ensemble methods based on classification trees. *Journal of Statistical Computation and Simulation* 2005;78(8):629-643.
44. Dettling M. BagBoosting for tumor classification with gene expression data. *Bioinformatics* 2004;20(18):3583-3593.
45. Friedman J. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 2002;38:367-378.
46. Hand D. Classifier technology and the illusion of progress. *Statistical Science* 2006;21:1-14 (disc.:15-34).

Table 1: Simplified description of the simulation design (means are subtracted and variances scaled)

Data	Predictors	Model
Design-A	$X_1 \sim N(0, 3)$	$\log \frac{p}{1-p} =$ $\lambda_0 + \lambda_{lin} [-X_1 + X_2 + X_3 + 2.5 X_4 + 5.25 X_5 + 6 X_6 + 20.75]$
	$X_2 \sim N(0, 2.1)$	
	$X_3 \sim N(0, 1.25)$	
	$X_4 \sim B(1, 0.5)$	
	$X_5 \sim B(1, 0.8)$	
	$X_6 \sim B(1, 0.5)$	
Design-B	$X_1, \dots, X_6$	$\log \frac{p}{1-p} =$ $\lambda_0 + \lambda_{lin} [-X_1 + X_2 + X_3 + 2.5 X_4 + 5.25 X_5 + 6 X_6 + 20.75]$ $+ \lambda_{prod} [0.95 Y_1 Y_2 + 4.75 Y_3 Y_4 + 9.5 Y_5 Y_6]$
	as in Design A	
	$Y_1 \sim N(0, 1)$	
	$Y_2 \sim N(4, 3)$	
	$Y_3 \sim N(2, 1)$	
	$Y_4 \sim B(1, 0.25)$	
	$Y_5 \sim B(1, 0.7)$	
	$Y_6 \sim B(1, 0.9)$	
Design-C	$X_1, \dots, X_6$	$\log \frac{p}{1-p} =$ $\lambda_0 + \lambda_{lin} [-X_1 + X_2 + X_3 + 2.5 X_4 + 5.25 X_5 + 6 X_6 + 20.75]$ $+ \lambda_{trans} \left[ \frac{1}{3} Z_1^2 + \frac{5}{6} \log(Z_2) + 0.4 Z_3^3 \right]$
	as in Design A	
	$Z_1 \sim N(0, 2)$	
	$Z_2 \sim X^2(1, 0)$	
	$Z_3 \sim N(0.5, 1)$	

Table 2: Squares of the pairs of scaling factors used in Design-B and Design-C. Note that whereas for (1,0) only linear terms enter the model, for  $\left(\frac{1}{4}, \frac{3}{4}\right)$  the sum of all non-linear or interaction terms resp. explains  $\frac{3}{4}$  of the variation of  $\log \frac{p}{1-p}$  and thus dominates the model. We suppose that the reverse setting of  $\left(\frac{3}{4}, \frac{1}{4}\right)$  is a more realistic scenario.

	$\lambda_{lin}^2$		1	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{4}$
$\lambda_{prod}^2$	resp.	$\lambda_{trans}^2$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$

Table 3: Mean performance (Brier score) of all classifiers for  $\lambda_{lin}^2=0.75$  for design B. Also given is the performance using the true conditional class probabilities. All SDS are below 0.011 for tree and below 0.006 for all other methods

	true	logreg	mfp	tree	ada.stumps	ada.tree	gbm.stumps	gbm.trees
Design B	0.07	0.09	0.09	0.18	0.11	0.13	0.10	0.10

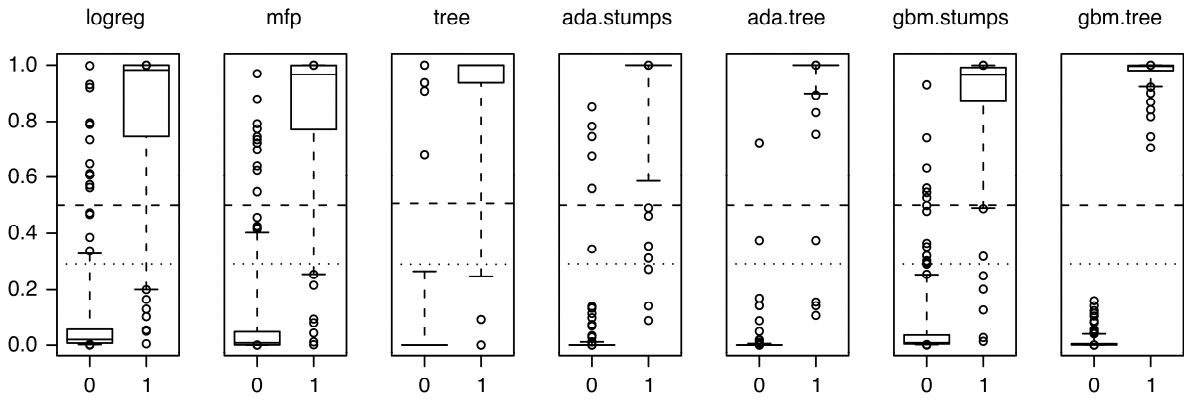


Figure 1: Boxplots of the estimated conditional class probabilities for a malignant tumor (class 1) obtained using the whole data set. The upper and lower whiskers correspond to the 90% - and 10% -quantiles respectively. The horizontal lines show possible classification thresholds at .5 and .29, the prevalence of malignant tumors.



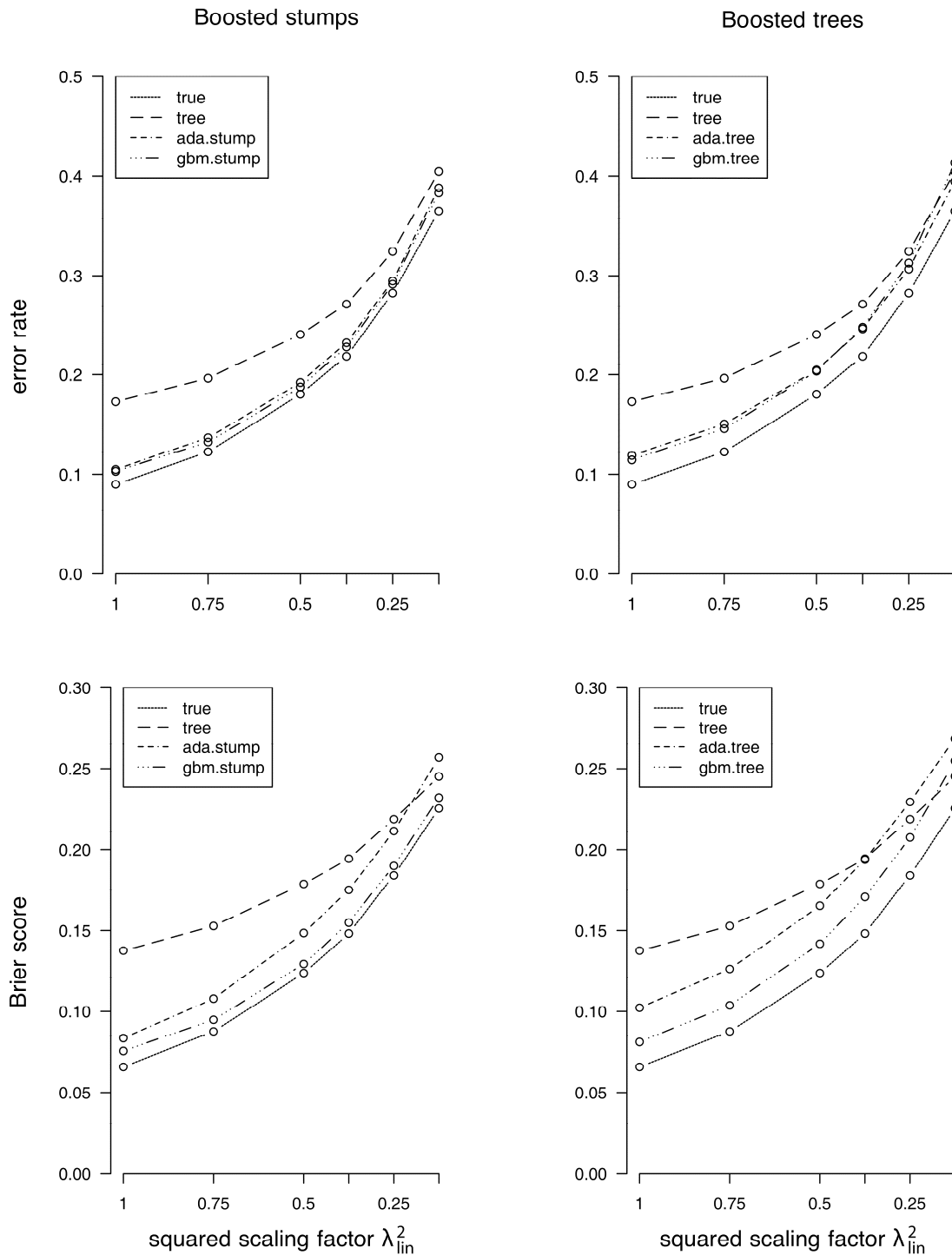


Figure 2: Design-A: Mean performance of the boosting and tree classifiers as a function of the squared scaling factor  $\lambda_{lin}^2$ . Also given is the performance obtained using the true conditional class probabilities. All SDs are below .013 for tree and below .008 for all other methods.

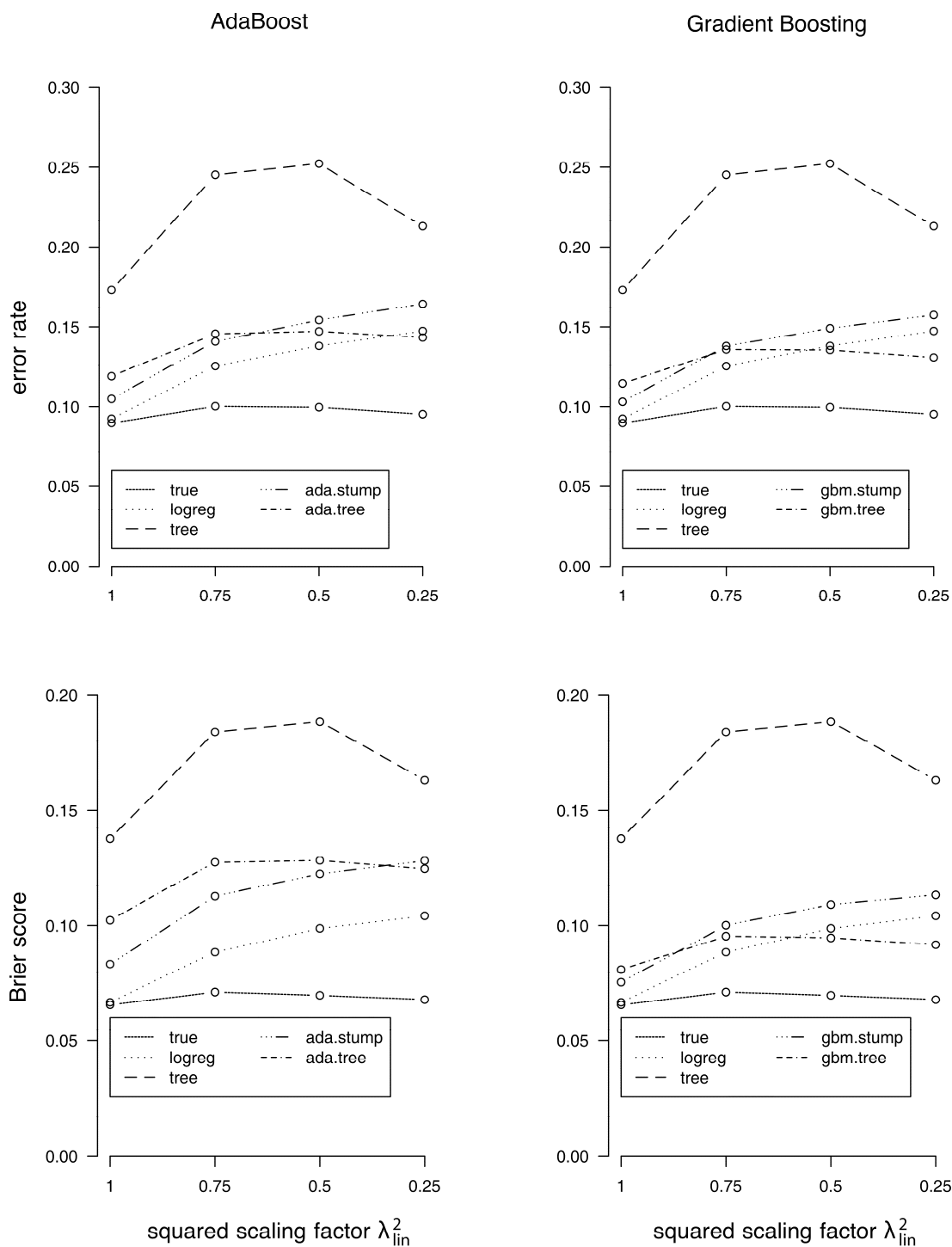


Figure 3: Design-B: Mean performance of the boosting and tree classifiers as a function of the scaling factor  $\lambda_{lin}^2$ . Also given is the performance obtained using the true conditional class probabilities. All SDs are below .012 for tree and below .006 for all other methods.

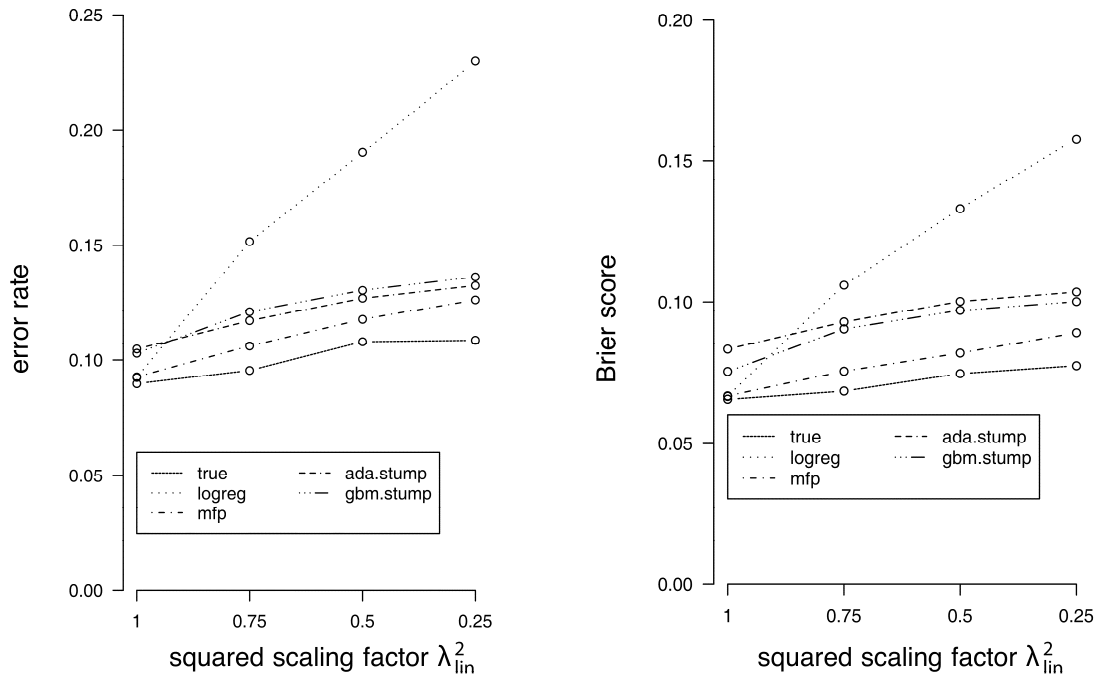


Figure 4: Design-C: Mean performance of the boosting and tree classifiers as a function of the scaling factor  $\lambda_{lin}^2$ . Also given is the performance obtained using the true conditional class probabilities. All SDs are below .013 for tree and below .006 for all other methods.

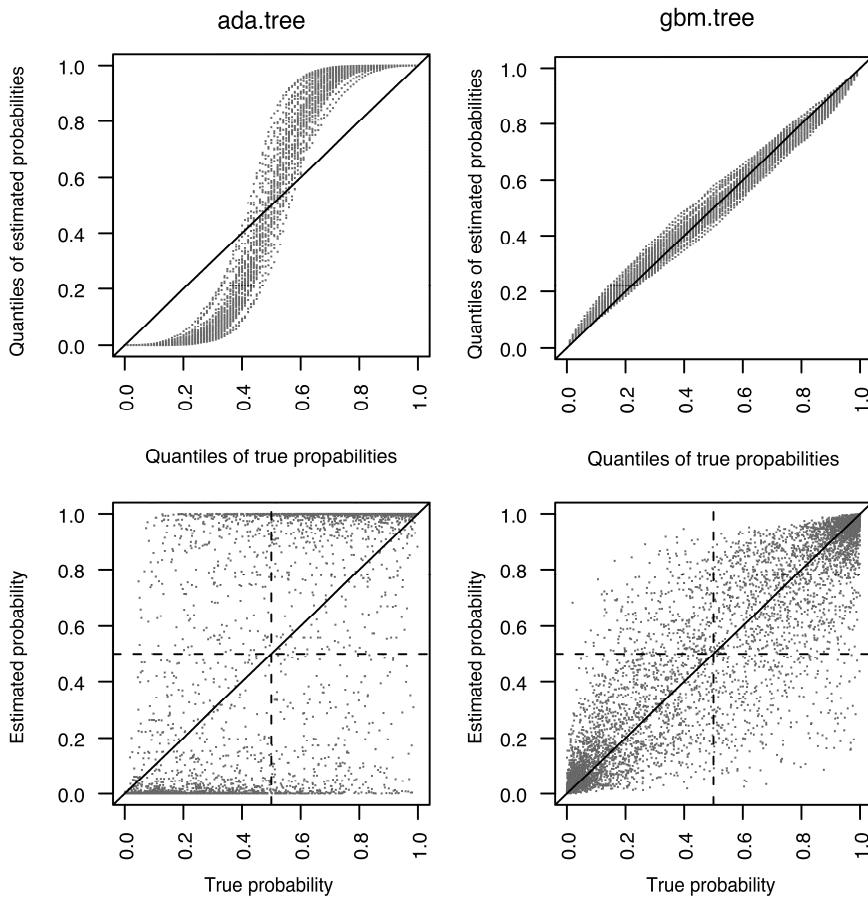


Figure 5: Estimated vs. true conditional probabilities of test set observations in Design-A with  $\lambda_{lin}^2$ . The upper figure displays all 50 quantile-quantile-plots, the lower a single plot of all pairs  $(p(x_i), \hat{p}(x_i))_{i=1, \dots, 10000}$ .