Statistical Applications in Genetics and Molecular Biology

Volume 11, *Issue* 5 2012 *Article* 3

A PAUC-based Estimation Technique for Disease Classification and Biomarker Selection

Matthias Schmid, *Friedrich-Alexander-University Erlangen-Nuremberg*
Torsten Hothorn, *University of Munich*
Friedemann Krause, *Roche Diagnostics GmbH*
Christina Rabe, *Roche Diagnostics GmbH*

Recommended Citation:
Schmid, Matthias; Hothorn, Torsten; Krause, Friedemann; and Rabe, Christina (2012) "A PAUC-based Estimation Technique for Disease Classification and Biomarker Selection," *Statistical Applications in Genetics and Molecular Biology*: Vol. 11: Iss. 5, Article 3.
DOI: 10.1515/1544-6115.1792

A PAUC-based Estimation Technique for Disease Classification and Biomarker Selection

Matthias Schmid, Torsten Hothorn, Friedemann Krause, and Christina Rabe

Abstract

The partial area under the receiver operating characteristic curve (PAUC) is a well-established performance measure to evaluate biomarker combinations for disease classification. Because the PAUC is defined as the area under the ROC curve within a restricted interval of false positive rates, it enables practitioners to quantify sensitivity rates within pre-specified specificity ranges. This issue is of considerable importance for the development of medical screening tests. Although many authors have highlighted the importance of PAUC, there exist only few methods that use the PAUC as an objective function for finding optimal combinations of biomarkers. In this paper, we introduce a boosting method for deriving marker combinations that is explicitly based on the PAUC criterion. The proposed method can be applied in high-dimensional settings where the number of biomarkers exceeds the number of observations. Additionally, the proposed method incorporates a recently proposed variable selection technique (stability selection) that results in sparse prediction rules incorporating only those biomarkers that make relevant contributions to predicting the outcome of interest. Using both simulated data and real data, we demonstrate that our method performs well with respect to both variable selection and prediction accuracy. Specifically, if the focus is on a limited range of specificity values, the new method results in better predictions than other established techniques for disease classification.

KEYWORDS: classification, combinations of biomarkers, gradient boosting, partial area under the ROC curve, screening tests, stability selection

Author Notes: Supported by the Interdisciplinary Center for Clinical Research (IZKF) at the University Hospital of the University of Erlangen-Nuremberg (Project J11) and by Deutsche Forschungsgemeinschaft (DFG), grant SCHM 2966/1-1.

1 Introduction

The area under the receiver operating characteristic curve (AUC) is a frequently used measure to assess the prediction accuracy of molecular markers for binary outcomes (Pepe 2003, Pepe et al. 2006). Being a combination of sensitivity and specificity values at all possible thresholds of a marker, the AUC is particularly useful as a summary measure for comparing the performance of biomarkers with respect to prognosis prediction. In many biomedical applications, however, using the AUC as a performance measure is not appropriate because only a part of the area under the ROC curve is relevant. This is especially true for screening tests where specificity rates have to be high because monetary costs require the rate of persons with a false positive result to be small. An example is given in Wild et al. (2010), who developed a combination of serum markers for the prediction of colorectal cancer. In their study, the authors first restricted the set of specificity levels to $\{0.95, 0.98\}$ and then optimized the sensitivity rates of marker combinations for the pre-set specificity values. Consequently, when deriving the optimal marker combination, specificity and sensitivity were not treated symmetrically. The AUC measure, on the other hand, does not account for this asymmetry because it is based on an unweighted combination of sensitivity values over the whole specificity range $[0,1]$. Using the AUC alone would therefore have led to a misjudgement of the performance of the marker combination by Wild et al. (2010).

A more suitable performance measure to evaluate marker combinations in screening tests is the partial area under the ROC curve ("PAUC", McClish 1989, Pepe and Thompson 2000, Dodd and Pepe 2003, Walter 2005). Instead of considering the whole specificity range $[0,1]$, the PAUC is defined as the integral of the ROC curve over a restricted interval of false positive rates $t \in [t_0, t_1]$. This strategy is equivalent to considering specificities within the range $[1-t_1, 1-t_0]$. More formally, if $f_i, f_j \in \mathbb{R}$ denote a pair of independent realizations of a marker f and if $Y_i, Y_j \in \{0,1\}$ are the respective values of the outcome variable, PAUC is defined as

$$\begin{aligned} \mathrm{PAUC}(t_0,t_1) &= \int_{t_0}^{t_1} \mathrm{ROC}(t)\,dt \\ &= \mathrm{P}\big(f_j > f_i \,\big|\, Y_j = 1, Y_i = 0, c_1(t_1) \leqslant f_i \leqslant c_0(t_0)\big)\,, \end{aligned} \tag{1}$$

where $\mathrm{ROC}(t)$ denotes the ROC curve of f and where $c_0, c_1 \in \mathbb{R}$ are the thresholds of f corresponding to the specificity values t_0 and t_1, respectively (see Dodd and Pepe 2003). If $[t_0,t_1] = [0,1]$, PAUC reduces to the the well-known AUC measure. It is seen from (1) that PAUC restricts analysis of the ROC curve to the fraction with false positive rates lying in the interval $[t_0,t_1]$. Optimizing the PAUC criterion therefore results in maximized sensitivity rates within a pre-set specificity range.

Although many authors have highlighted the importance of PAUC, there exist only few methods that use the PAUC as an objective function for finding optimal combinations of biomarkers (Dodd and Pepe 2003, Komori and Eguchi 2010, Wang and Chang 2010). To address this problem, we present a new technique for deriving marker combinations that is explicitly based on the PAUC criterion. The proposed method is embedded into the gradient boosting framework (Bühlmann and Hothorn 2007) and can therefore be applied in high-dimensional data settings where the number of biomarkers exceeds the number of observations. Additionally, the proposed method incorporates a new variable selection technique proposed by Meinshausen and Bühlmann (2010) ("stability selection") that results in sparse prediction rules incorporating only those biomarkers that make relevant contributions to predicting the outcome of interest. Specifically, our method addresses the following issues:

1. *Interpretability of results.* By embedding the proposed method into the gradient boosting framework, it is possible to specify the structure of the marker combination in advance. For example, our method can be adjusted such that it results in an additive combination of the form

$$f(X) = f_{(1)}(X_1) + \cdots + f_{(p)}(X_p) \ , \tag{2}$$

   where $X = (X_1, \ldots, X_p)$ is a set of markers and $f_{(1)}, \ldots, f_{(p)}$ is a set of differentiable functions. With a combination of the form (2) it is possible to quantify the associations between individual markers and the outcome and to obtain estimates of partial effects. From a practical perspective, this is a major advantage over black-box methods such as Support Vector Machines (Vapnik 2000) or Random Forests (Breiman 2001).
2. *Nonlinear effect estimates.* As seen from (2), the marker combination resulting from the proposed method is not restricted to being linear. This feature results in an increased flexibility if compared to linear methods such as the Lasso (Tibshirani 1996).
3. *Sparsity of results.* For practical reasons, it is often necessary to keep the number of markers contained in (2) small. For example, the combination proposed by Wild et al. (2010) contained only six serum markers. Classical gradient boosting algorithms, however, may result in prediction functions with large numbers of selected markers (Bühlmann and Hothorn 2010, Meinshausen and Bühlmann 2010). When using our new method, sparsity of marker combinations is guaranteed by the stability selection technique.

For the rest of the paper, we will refer to the proposed method as PAUC-GBS ("**PAUC** optimization via **G**radient **B**oosting and **S**tability Selection").

PAUC-GBS extends existing methods for PAUC optimization in various ways. For example, Dodd and Pepe (2003) and Cai and Dodd (2008) developed regression techniques using the PAUC as outcome variable. It is, however, unclear how to apply these techniques in high-dimensional settings and how to carry out variable selection. As outlined above, this problem is addressed by PAUC-GBS. Wang and Chang (2010) proposed a wrapper-type algorithm to optimize the PAUC over a combination of biomarkers. While the algorithm by Wang and Chang (2010) is applicable in both low- and high-dimensional settings, it is restricted to linear combinations of markers. Also, for reasons of identifiability, the algorithm requires specification of an "anchor marker" that is not subject to variable selection but needs to be included in the marker combination a priori. PAUC-GBS avoids this problem by standardizing the marker combination, thereby giving equal weights to all markers at the beginning of the variable selection process. Komori and Eguchi (2010) proposed a Newton-Raphson-type boosting algorithm for maximizing the PAUC over nonlinear combinations of biomarkers. Similar to the method proposed in this paper, Komori and Eguchi's method does not require pre-specification of an anchor marker. It is, however, computationally expensive because it involves multiple tuning parameters that are needed to determine the optimal marker combination. The computational effort is further increased by the fact that the thresholds c_0, c_1 have to be re-estimated in each iteration. In contrast, PAUC-GBS does not require re-estimation of c_0, c_1 and is relatively easy to tune.

Using both simulated data and the data collected by Wild et al. (2010), we demonstrate that maximizing the PAUC with PAUC-GBS is a suitable strategy for developing medical screening tests. Our results show that PAUC-GBS performs well with respect to both variable selection and prediction accuracy. Specifically, if the focus is on a limited range of specificity values, using PAUC-GBS results in better predictions than other established techniques for the prognosis of binary outcomes. In Section 2, we start with a formal definition of PAUC-GBS and provide recommendations on how to choose the tuning parameters of the algorithm. The results of the simulation study and the analysis of the data collected by Wild et al. (2010) are presented in Section 3. The final section summarizes the main findings of the paper and discusses their consequences for biomedical applications. PAUC-GBS is implemented in the R add-on package **mboost** (Hothorn et al. 2011). An example on how to run PAUC-GBS in R is provided in the appendix of the paper.

2 Methods

2.1 Estimation of PAUC

Let f denote a real-valued marker and $Y \in \{0,1\}$ a binary outcome variable. Suppose that $f(X) = f(X_1,\ldots,X_p)$ is a combination of markers and that we have a random sample of observations $(y_i, x_{i1},\ldots,x_{ip})$, $i = 1,\ldots,n$. Using standard terminology, we refer to the set of observations with $y_i = 1$ as the "diseased group" and to the set of observations with $y_i = 0$ as the "healthy" group. Denote these sets by $\mathscr{D}$ and $\mathscr{H}$, respectively.

For any given threshold c, the sensitivity (or "true positive rate", abbreviated by TPR) is defined as $\mathrm{P}(f > c \,|\, Y = 1)$. Similarly, the specificity (or 1 - "false positive rate", abbreviated by 1 - FPR) is defined as $1 - \mathrm{P}(f > c \,|\, Y = 0) = \mathrm{P}(f \leqslant c \,|\, Y = 0)$. The ROC curve of f is obtained by plotting the FPR values against their corresponding TPR values for all possible thresholds c. By equation (1), $\mathrm{PAUC}(t_0, t_1)$ is defined as the area under the ROC curve within a pre-specified interval of FPR rates (denoted by $[t_0, t_1]$). For the rest of the paper, we restrict our analysis to FPR ranges with a lower FPR bound $t_0 = 0$. This is because of practical considerations (as FPR rates are typically required to be small in biomedical applications) but also because PAUC values with non-zero FPR lower bounds can easily be obtained by setting $\mathrm{PAUC}(t_0, t_1) = \mathrm{PAUC}(0, t_1) - \mathrm{PAUC}(0, t_0)$ (see Wang and Chang 2010).

Parametric and non-parametric estimators of PAUC have been extensively studied in the literature (McClish 1989, Zhang et al. 2002, Pepe 2003, Dodd and Pepe 2003, Walter 2005). In this paper, we will consider a non-parametric estimator of $\mathrm{PAUC}(0, t_1)$ defined as

$$\widehat{\mathrm{PAUC}}(0, t_1) := \frac{1}{n_1} \sum_{i \in \mathscr{D}} \left[t_1 - \min \left\{ \frac{1}{n_0} \sum_{j \in \mathscr{H}} \mathrm{I}(f_j > f_i), t_1 \right\} \right], \tag{3}$$

where $f_i = f(x_{i1},\ldots,x_{ip})$ denotes the marker combination of the i-th observation and where n_0 and n_1 are the cardinalities of the sets $\mathscr{H}$ and $\mathscr{D}$, respectively (see Wang and Chang 2010).

Wang and Chang (2010) showed that, provided that f is known, $\widehat{\mathrm{PAUC}}(0, t_1)$ is strongly consistent for $\mathrm{PAUC}(0, t_1)$ as $\min\{n_0, n_1\} \to \infty$. In practice, however, f is usually not known and has to be estimated from $\mathscr{D}$ and $\mathscr{H}$. In addition, f should depend on only those markers that make relevant contributions to maximizing $\mathrm{PAUC}(0, t_1)$. This requires an estimation technique incorporating both PAUC optimization and variable selection.

2.2 Component-wise Gradient Boosting

To derive an estimate of f, we will use component-wise gradient boosting techniques in combination with stability selection. Component-wise gradient boosting (Bühlmann and Yu 2003, Bühlmann and Hothorn 2007) is a general statistical method to estimate a prediction function f by minimizing the expectation of a loss function $\rho(Y,f)$ over f. The loss function $\rho(Y,f)$ is assumed to be differentiable with respect to f. More formally, the aim is to estimate the "optimal" prediction function f^* defined as

$$f^* := \underset{f}{\operatorname{argmin}}\, \mathbb{E}_{Y,X}\left[\rho(Y,f(X))\right] \tag{4}$$

by using gradient descent techniques. Common examples of loss functions include the squared error loss in Gaussian regression (Bühlmann and Yu 2003) and the negative binomial log likelihood in logistic regression (Friedman et al. 2000, Dettling and Bühlmann 2003). Because the theoretical mean given in (4) is usually unknown in practice, gradient boosting algorithms minimize the empirical risk $\mathscr{R} := \sum_{i=1}^{n} \rho(y_i, f(x_i))$ over f. Due to their wide applicability, boosting techniques and related algorithms have been used to address various types of clinical and biomedical statistical analysis problems (see, e.g., Boulesteix 2004, Teramoto 2009, Wang and Wang 2010).

To use component-wise gradient boosting techniques for PAUC optimization, we first set the (unknown) prediction function f equal to the marker combination discussed in the previous subsection. Also, it would be convenient to use the negative version of $\widehat{\text{PAUC}}(0,t_1)$ as empirical risk function. However, setting $\mathscr{R} = -\widehat{\text{PAUC}}(0,t_1)$ is not feasible because $\widehat{\text{PAUC}}(0,t_1)$ is not differentiable with respect to f_i and f_j. To solve this problem, we follow the approaches of Ma and Huang (2005) and Wang and Chang (2010) and approximate the indicator and the min functions in (3) by sigmoid functions $K(u) = 1/(1+\exp(-u/\sigma))$. Here, σ is a tuning parameter that controls the smoothness of the approximation. Replacing the indicator and min functions in (3) by their smoothed versions results in the following estimator of PAUC:

$$\widehat{\text{PAUC}}_s(0,t_1) := \frac{1}{n_1} \sum_{i \in \mathscr{D}} \left[t_1 - \sigma \cdot \log \frac{1+\exp(t_1/\sigma)}{1+\exp\left[\left(t_1 - \frac{1}{n_0}\sum_{j \in \mathscr{H}} \frac{1}{1+\exp((f_i-f_j)/\sigma)}\right)/\sigma\right]} \right] . \tag{5}$$

A detailed derivation of (5), which is not straightforward, can be found in Wang and Chang (2010). By definition, the smoothed PAUC estimator is differentiable with respect to f_i and f_j. Its derivatives are given by

$$\frac{\partial \widehat{\mathrm{PAUC}}_s(0,t_1)}{\partial f_i} = \frac{\frac{1}{n_0 \cdot n_1} \cdot \sum_{j \in \mathscr{H}} \frac{\exp((f_i - f_j)/\sigma)}{\sigma\left[1+\exp((f_i-f_j)/\sigma)\right]^2}}{1+\exp\left\{\left[\frac{1}{n_0}\sum_{j\in\mathscr{H}} \frac{1}{1+\exp((f_i-f_j)/\sigma)} - t_1\right]/\sigma\right\}}, \tag{6}$$

$$\frac{\partial \widehat{\mathrm{PAUC}}_s(0,t_1)}{\partial f_j} = \sum_{i \in \mathscr{D}} \frac{-\frac{1}{n_0 \cdot n_1} \cdot \frac{\exp((f_i - f_j)/\sigma)}{\sigma\left[1+\exp((f_i-f_j)/\sigma)\right]^2}}{1+\exp\left\{\left[\frac{1}{n_0}\sum_{j\in\mathscr{H}} \frac{1}{1+\exp((f_i-f_j)/\sigma)} - t_1\right]/\sigma\right\}}. \tag{7}$$

Choosing appropriate values of the smoothness parameter σ is essential to guarantee the consistency of $\widehat{\mathrm{PAUC}}_s(0,t_1)$. Clearly, small values of σ result in a close approximation of $\widehat{\mathrm{PAUC}}(0,t_1)$ but might also overfit the data. Wang and Chang (2010) showed that

$$\widehat{\mathrm{PAUC}}_s(0,t_1) - \widehat{\mathrm{PAUC}}(0,t_1) \xrightarrow{\min\{n_0,n_1\}\to\infty} 0 \tag{8}$$

if $\sigma = \mathrm{O}(\min\{n_0,n_1\}^{-1/4})$. Therefore, a convenient strategy (that is in line with the approach taken by Wang and Chang 2010) is to set $\sigma = \min\{n_0,n_1\}^{-1/4}$. This strategy worked remarkably well in our simulation studies and will therefore be used in the rest of the paper.

Setting $\mathscr{R} = -\widehat{\mathrm{PAUC}}_s(0,t_1)$, we are able to define the component-wise gradient boosting algorithm for estimating the optimal marker combination f:

1. Initialize an n-dimensional vector $\hat{f}^{[0]}$ with offset values (for example, set $\hat{f}^{[0]} = \mathbf{0}$).
2. For each of the markers specify a *base-learner*. A base-learner is a regression estimator with one input variable and one output variable. Set $m = 0$.
3. Increase m by 1.
4. Compute the negative gradient $-\frac{\partial \mathscr{R}}{\partial f_i}$, $i = 1,\ldots,n$, by using formulas (6) and (7). Evaluate the negative gradient at $\hat{f}^{[m-1]}(x_i)$, $i = 1,\ldots,n$. This yields the negative gradient vector

$$U^{[m]} = \left(U_i^{[m]}\right)_{i=1,\ldots,n} := \left(-\frac{\partial}{\partial f_i}\mathscr{R}\left(y_i, \hat{f}^{[m-1]}(x_i)\right)\right)_{i=1,\ldots,n}. \tag{9}$$

5. Fit the negative gradient vector $U^{[m]}$ to each of the p markers separately by using the base-learners specified in step 2. This yields p vectors of predicted values, where each vector is an estimate of $U^{[m]}$. Select the base-learner that fits $U^{[m]}$ best according to the R^2 goodness-of-fit criterion. Set $\hat{U}^{[m]}$ equal to the fitted values obtained from the best base-learner.

6. Update $\hat{f}^{[m]} \leftarrow \hat{f}^{[m-1]} + \nu \hat{U}^{[m]}$, where $0 < \nu \leqslant 1$ is a real-valued step length factor.
7. Set $\hat{f}^{[m]} \leftarrow \hat{f}^{[m]} - \hat{f}_1^{[m]}$, where $\hat{f}_1^{[m]}$ is the first element of $\hat{f}^{[m]}$.
8. Iterate Steps 3 to 7 until the stopping iteration m_{stop} is reached (the choice of m_{stop} will be discussed below).

By definition of the above algorithm, estimates of the marker combination f are obtained via descending the gradient of the empirical risk $\mathscr{R}$. In each iteration, an estimate of the true negative gradient of $\mathscr{R}$ is added to the current estimate of f, and a structural (regression) relationship between Y and the selected markers is established. Additionally, the boosting algorithm defined above carries out variable selection, as only one base-learner (i.e., one marker) is selected for updating $\hat{f}^{[m]}$ in step 6 (hence the term "component-wise" gradient boosting). Due to the additive update, the final boosting estimate at iteration m_{stop} can be interpreted as an additive prediction function (as defined in (2), cf. Bühlmann and Hothorn 2007). The choice of the step length factor ν has been shown to be of minor importance for the performance of boosting algorithms (Schmid and Hothorn 2008). The only requirement is that ν is "small", such that the algorithm does not overshoot the minimum of $\mathscr{R}$. For the rest of the paper, we will set $\nu = 0.1$.

As seen from (5), $\widehat{\text{PAUC}}_s(0,t_1)$ does not change its value if a real-valued constant is added to the predicted values $\hat{f}_1^{[m]}, \ldots, \hat{f}_n^{[m]}$. Therefore, to guarantee identifiability of $\hat{f}^{[m]}$, we restrict the predicted value of the first observation to zero (step 7 of the algorithm). In contrast to the algorithms proposed by Ma and Huang (2005) and Wang and Chang (2010), this strategy avoids pre-specifying an anchor marker that is not subject to variable selection.

Concerning the choice of the base-learners, it is clear from steps 5 and 6 that the estimates of the partial functions $f_{(j)}(X_j)$ at iteration m_{stop} have the same structure as the base-learners used in each iteration. In this paper, we will either use simple linear models or cubic P-spline functions (with four degrees of freedom and a second-order difference penalty) as base-learners. This corresponds to specifying linear or smooth nonlinear functions $f_{(j)}$ for the marker combination (see Schmid and Hothorn 2008). In contrast to black-box methods, this strategy further guarantees the interpretability of $\hat{f}^{[m_{\text{stop}}]}$.

A much debated question is the choice of the stopping iteration m_{stop}. In high-dimensional settings with $p \gg n$, the stopping iteration determines the number of selected markers and also the amount of regularization for $\hat{f}^{[m_{\text{stop}}]}$. A common strategy is to use cross-validation techniques for determining m_{stop} (see Bühlmann and Hothorn 2007). As pointed out by Bühlmann and Hothorn (2010), however, cross-validation may result in prediction functions with too many selected mark-

ers. Therefore, as an alternative to cross-validation, we will focus on the stability selection method proposed by Meinshausen and Bühlmann (2010).

2.3 Stability Selection

The main idea behind stability selection is to sample randomly from the data and to use the obtained selection probabilities of the markers as a criterion for variable selection. In other words, for any given iteration number m, stability selection keeps those markers with a high probability of being selected up to m in the model but disregards all other markers. As demonstrated by Meinshausen and Bühlmann (2010), stability selection typically results in much sparser models than cross-validation techniques. This result is of considerable importance in biomedical applications where the maximum number of components in a marker set is limited for practical reasons.

To carry out stability selection, one needs to specify an appropriate probability threshold $\pi_{\text{sel}} \in (0.5, 1)$ that determines which selection probabilities are "high" and which are "low". Meinshausen and Bühlmann (2010) observed that "for sensible values in the range of, say, $\pi_{\text{sel}} \in (0.6, 0.9)$, results tend to be very similar". In addition to this empirical result, Meinshausen and Bühlmann (2010) derived a strategy for the selection of π_{sel} that controls the family-wise error rate FWER (i.e., the probability that the set of falsely selected variables is non-empty) at a pre-specified error level. The latter strategy, however, relies on various regularity assumptions regarding the dependency structure of the data. Also, it has been argued that the FWER criterion should be replaced by more liberal criteria for variable selection (Ahmed et al. 2011). In view of these considerations, and because the afore-mentioned regularity assumptions are difficult to verify in practice, we will follow the advice of Meinshausen and Bühlmann (2010) and focus on a probability range $\pi_{\text{sel}} \in (0.6, 0.9)$ for marker selection. Specifically, we will use a probability threshold of $\pi_{\text{sel}} = 0.9$ for the numerical studies presented in Section 3.

A remaining question is how to estimate the selection probabilities of the individual markers $X_1, \ldots, X_p$. Meinshausen and Bühlmann (2010) suggested to use 100 random subsamples of size $\lfloor n/2 \rfloor$ without replacement to obtain probability estimates. Our simulation studies, however, showed that reducing the sample size to only one half of the original observation number results in too sparse models with unsatisfying prediction accuracy. This result can be explained by the fact that the variable selection behavior of boosting algorithms is highly dependent on the sample size (Bühlmann 2006). In other words, reducing the sample size to $\lfloor n/2 \rfloor$ will affect the variable selection behavior of gradient boosting to a large degree, and estimates of selection probabilities will only partly reflect the probabilities of

the whole sample. Regarding prediction accuracy, we obtained better results when increasing the size of the subsamples to $\lfloor 0.8 \cdot n \rfloor$. For the rest of the paper, we will therefore use 20 subsamples of size $\lfloor 0.8 \cdot n \rfloor$ for the estimation of the selection probabilities in each boosting iteration.

2.4 PAUC-GBS

Summarizing Subsections 2.1 to 2.3, we obtain the following algorithm for PAUC optimization via component-wise gradient boosting and stability selection ("PAUC-GBS"):

1. Run component-wise gradient boosting using a sufficiently large number of iterations.
2. Run stability selection using 20 random subsamples of size $0.8 \cdot n$ without replacement.
3. Re-run component-wise gradient boosting, this time using only the markers that were selected by the stability selection procedure.

3 Numerical Results

3.1 Simulation Study

To investigate the variable selection behavior of PAUC-GBS, we carried out a simulation study with 100 independent i.i.d. data sets. Each data set contained $n_0 = 50$ non-diseased and $n_1 = 50$ diseased observations. For each of the 100 data sets we considered a set of 506 markers, where 500 of the 506 markers were non-informative because their values were drawn independently from a standard uniform distribution. Additionally, we considered six informative markers (denoted by $X_1, \ldots, X_6$) whose data values were generated in the following way: In the non-diseased group, we first generated 50 random samples drawn from a random variable B that followed a beta distribution with shape parameters $a = 0.5$ and $b = 100$. Next, we generated the values of the markers $X_1, \ldots, X_3$ using $X_j = B + \varepsilon_j$, $j = 1, 2, 3$, where the noise variables ε_j were independent of X_j and followed a normal distribution with zero mean and standard deviation 0.3. The values of the markers $X_4, \ldots, X_6$ were generated in the same way, this time using a beta distribution with $a = 0.4$ and $b = 0.5$. For the diseased group we followed the same strategy: The values of $X_1, \ldots, X_3$ were generated by adding normally distributed errors to a beta-distributed random variable with $a = 0.1$ and $b = 0.1$ while the

values of $X_4, \ldots, X_6$ were generated analogously, this time using a beta distribution with $a = 1.5$ and $b = 0.3$.

As seen from Figure 1(a), the markers $X_4, \ldots, X_6$ resulted in symmetric ROC curves when used for the prediction of Y. Conversely, the markers $X_1, \ldots, X_3$ were highly specific, i.e., they resulted in relatively large TPR values given small FPR values. The AUC was approximately the same for all six informative markers ($AUC \approx 0.75$). Consequently, we expected an increase in the selection rates of $X_1, \ldots, X_3$ when using PAUC optimization with small upper FPR rates. After data generation, we applied PAUC-GBS to the 100 samples using the two FPR ranges $[0, 0.1]$ and $[0, 1]$. The latter range corresponds to classical AUC optimization. As base-learners we used simple linear models, which resulted in linear marker combinations. In addition to PAUC-GBS, we applied the Lasso method (Tibshirani 1996) to the 100 data sets. This method is based on the binomial log-likelihood criterion and also results in linear marker combinations. Five-fold cross-validation was used to determine the optimal tuning parameter of the Lasso method.

The relative selection rates of $X_1, \ldots, X_6$ computed from the 100 simulation runs are shown in Figures 1(b) and 1(c). Obviously, in case of PAUC-GBS with the small FPR range $[0, 0.1]$, selection rates of the specific markers $X_1, \ldots, X_3$ were larger than the corresponding rates resulting from PAUC-GBS with the whole FPR range $[0, 1]$ (Figure 1(b)). Conversely, the selection rates of the non-specific markers $X_4, \ldots, X_6$ were similar for both FPR ranges. This result can be explained by the fact that optimizing the PAUC over $[0, 1]$ corresponds to maximizing the classical AUC criterion. Hence, because $X_1, \ldots, X_6$ have similar univariate AUC values, differences in the selection rates between specific and non-specific markers were smaller in case of PAUC-GBS with FPR range $[0, 1]$ than in case of PAUC-GBS with FPR range $[0, 0.1]$. As expected, the Lasso method resulted in very similar selection rates for all six informative markers (Figure 1(c)). These results clearly suggest that PAUC-GBS adapts its mechanism for marker selection to the range of desired FPR values. The average selection rates of the non-informative markers were 1.00% for PAUC$(0, 0.1)$, 1.00% for PAUC$(0, 1)$ and 3.96% for the Lasso method.

The stability selection procedure with FPR range $[0, 0.1]$ is illustrated in Figure 1(d). It is seen that two markers (X_2 and X_5) were selected by PAUC-GBS in the first simulation run.

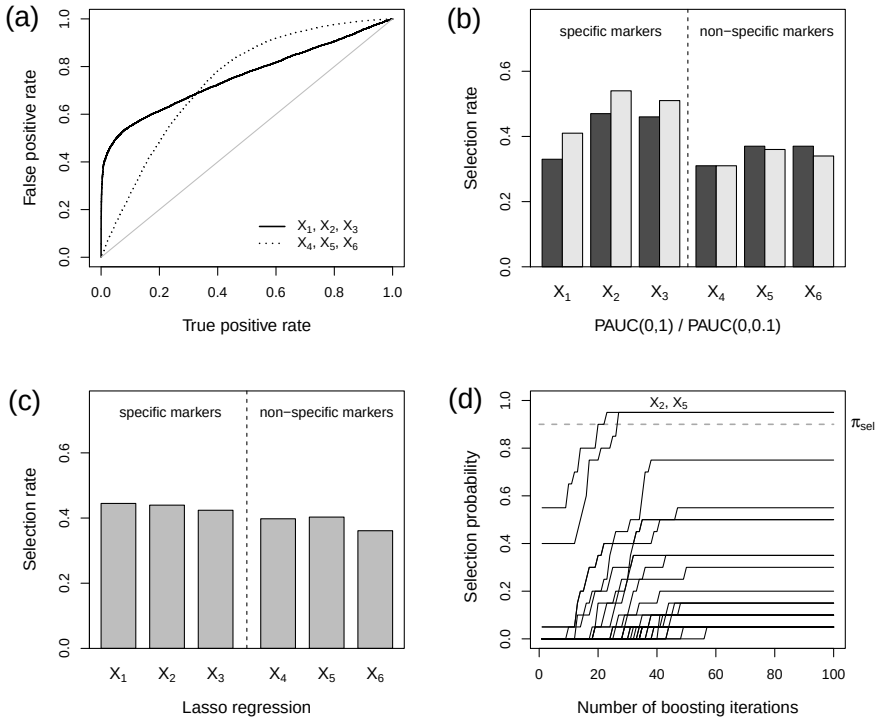

Figure 1: Results of the simulation study. (a) Univariate ROC curves of the markers $X_1,\ldots,X_6$ (estimated from a sample with $n = 20,000$ observations). By definition, the markers X_1, X_2 and X_3 result in the same ROC curves if used for prediction of Y. The same holds true for the ROC curves of X_4, X_5 and X_6. (b) Selection rates of the informative markers $X_1,\ldots,X_6$ obtained from PAUC-GBS with FPR ranges $[0,1]$ (dark grey bars) and $[0,0.1]$ (light grey bars). (c) Relative selection rates of the informative markers $X_1,\ldots,X_6$ obtained from the Lasso method. (d) Plot of the number of boosting iterations against estimated selection probabilities of all markers (obtained from stability selection with FPR range $[0,0.1]$, simulation run #1).

3.2 Marker Combinations for the Detection of Colorectal Cancer

Aims and scope. The early detection of colorectal cancer (CRC) is widely acknowledged as a key factor to reduce the mortality from CRC (Etzioni et al. 2003). Fecal occult blood testing (FOBT) and fecal immunochemical testing (FIT), however, which are currently recommended as first-line screening methods for CRC with subsequent colonoscopy for patients tested positive, suffer from low patient com-

pliance. Serum-based screening tests for CRC would therefore be attractive, as they could easily be integrated in regular health checkups. In a prospective trial on the early detection of CRC, Wild et al. (2010) proposed a combination of six serum markers that were analyzed in the course of two European multicenter studies. The combination was derived by applying L_1-penalized regression (Lasso, Tibshirani 1996) to a set of 22 pre-selected biomarkers. Wild et al. (2010) showed that the proposed marker combination constitutes a valuable alternative to fecal occult blood testing. The sensitivity levels of the marker combinations were evaluated at preset specificities. However, since the selection of variables and even the coefficients of the linear marker combination were derived by optimizing the penalized binomial likelihood function, the combination may not be optimal for the intended specificity range.

Data collection and pre-processing. To analyze the performance of PAUC-GBS, we used a CRC cohort of $n_1 = 282$ observations and a control cohort of $n_0 = 248$ observations contained in the original data by Wild et al. (2010), including only cancer cases that were relevant for screening (stages 0 - III). The values of 25 serum markers (containing the afore-mentioned 22 markers) were provided by Roche Diagnostics, Germany. The 25 markers had been pre-selected by Roche from a larger set of potential markers in the course of the trial and were all informative for the diagnosis of CRC. Therefore, in order to test PAUC-GBS in high-dimensional settings, we used the same strategy as Bühlmann and Hothorn (2010) and added a set of 1000 non-informative random variables to the 25 serum markers. The data values of the non-informative random variables were drawn from a standard multivariate normal distribution with zero mean and (equi)correlation $\rho = 0.5$. The values of the 25 serum markers were standardized before analysis.

Cross-validated PAUC analysis. To assess the prediction accuracy of PAUC-GBS, we split the data randomly into 50 learning samples of size $353 \approx 2/3 \cdot n$ each and 50 test sampples of size $177 \approx 1/3 \cdot n$ each. This procedure resulted in 50 cross-validation runs. Cubic P-spline base-learners were used for all markers. To control the smoothness of of effect estimates, we used finite stopping iterations for the component-wise gradient boosting procedure in step 3 of PAUC-GBS (Subsection 2.4). These stopping iterations were determined by applying a five-fold cross-validation procedure to the learning samples (R package mboost, Hothorn et al. 2011). Three FPR ranges ($[0,0.2]$, $[0,0.5]$ and $[0,1]$) were considered for PAUC optimization. The latter range corresponds to optimizing the classical AUC criterion. Similar to Wild et al. (2010), results were evaluated by determining the sensitivity values of cross-validated ROC curves at various pre-set specificity levels.

Benchmark analysis. Using the same learning and test samples, we compared PAUC-GBS to the following alternative classification techniques: (a) Boosting with the negative binomial log likelihood loss and decision stumps as base-learners (*LogitBoost*, Dettling and Bühlmann 2003). Five-fold cross-validation was used to determine the optimal stopping iteration of the LogitBoost algorithm (R package LogitBoost, Dettling 2003). (b) Gradient boosting with the exponential loss function and tree base-learners (*gbm*, Friedman 2001). This algorithm served as a natural reference procedure because it is based on the original AdaBoost algorithm by Freund and Schapire (1997). We used 500 trees in combination with five-fold cross-validated stopping iterations for prediction of Y (R package gbm, Ridgeway 2010). (c) Component-wise gradient boosting with the hinge loss (*HingeBoost*, Wang 2011). HingeBoost was used as reference method because it allows for specifying the same P-spline base-learners as those used for PAUC-GBS. In addition, we applied the same stability selection procedure to HingeBoost as the one used for PAUC-GBS (R package mboost, Hothorn et al. 2011). (d) L_1-penalized regression (*Lasso*, Tibshirani 1996). This method, which is restricted to linear marker combinations, was used as a reference procedure because Wild et al. (2010) derived their original marker combination using Lasso regression. We used five-fold cross-validation to determine the optimal tuning parameter of the Lasso procedure (R package glmpath, Park and Hastie 2011).

Results. Cross-validated sensitivity rates at specificity levels 0.90 and 0.95 are presented in Table 1. The high sensitivity rates obtained from PAUC-GBS clearly demonstrate the benefits of using the PAUC criterion for optimizing marker combinations. Specifically, all PAUC-GBS variants performed better than the HingeBoost method. This result indicates that improvements in sensitivity rates cannot only be attributed to the use of nonlinear marker combinations but to the use of the PAUC criterion instead of other loss functions. It is also seen that the Lasso method performed remarkably well if compared to the nonlinear PAUC-GBS and HingeBoost methods, thereby confirming the results obtained by Wild et al. (2010). The sensitivity rates presented in Table 1 further suggest that the two tree-based methods gbm and LogitBoost are not superior to the methods with an additive prediction function of the form (2). When analyzing the results obtained from the different variants of PAUC-GBS, it is seen that PAUC-GBS with FPR range $[0, 0.2]$ performed best at both specificity levels 0.90 and 0.95. Table 1 also shows that PAUC-GBS with restricted FPR ranges $[0, 0.2]$ and $[0, 0.5]$ performed better than the unrestricted AUC variant with FPR range $[0, 1]$.

The selection rates obtained from PAUC-GBS with FPR range $[0, 0.2]$ are shown in Figure 2. Obviously, the markers CYFRA 21-1, CEA, Ferritin, Seprase, Osteopontin (OPN) and Anti-p53 have the highest selection rates. Selection rates

specificity level	0.95		0.90	
PAUC-GBS(0,0.2)	71.05	(65.63, 75.88)	79.26	(74.79, 81.79)
PAUC-GBS(0,0.5)	70.92	(65.54, 74.45)	79.24	(74.26, 81.98)
PAUC-GBS(0,1.0)	70.00	(65.61, 73.93)	77.97	(74.45, 81.98)
LogitBoost	62.89	(55.55, 68.44)	73.34	(67.19, 78.65)
gbm	54.90	(47.47, 59.89)	64.94	(57.53, 70.60)
HingeBoost	65.07	(59.84, 70.41)	72.82	(69.12, 77.21)
Lasso	70.58	(66.35, 74.48)	77.84	(75.22, 80.84)

Table 1: Prediction of colorectal cancer. The table presents median cross-validated sensitivity rates obtained from the 50 test samples (multiplied by 100). Numbers in brackets are the empirical 25- and 75-quantiles obtained from the test data.

of the other markers are smaller than 50%, indicating that the variable selection behavior of PAUC-GBS is stable. The nonlinear effect estimates of the aforementioned six markers (obtained from applying PAUC-GBS with FPR range $[0, 0.2]$ to the whole data set) are presented in Figure 3. While the partial functions of Ferritin and Seprase show a downwards trend, the function corresponding to OPN is strictly increasing. All six functions are distinctly nonlinear, suggesting that the marginal effects of the markers on prediction accuracy depend heavily on the actual concentration levels of the markers. These effects are not captured by linear methods such as the Lasso (where marginal effects are assumed to be constant, regardless of the concentration level). The function of Anti-p53 has a quadratic structure, suggesting that there is a threshold below which increased concentration levels of Anti-p53 lead to an increased probability of CRC. Conversely, concentration levels above the threshold are negatively correlated with the occurrence of CRC. Again, this effect cannot be captured by linear methods such as the Lasso. In contrast to Anti-p53, the functions corresponding to CYFRA 21-1 and CEA have a monotonic piecewise structure with distinct breakpoints. These breakpoints could conveniently be used to categorize the concentration levels of the two markers in later experiments.

When comparing the results presented in Figure 2 to the results by Wild et al. (2010), it is seen that the six marker candidates selected most frequently by PAUC-GBS are the same as those selected by the original algorithm. Also, the functional forms of the monotonic predictor-response relationships (Figure 3) are consistent with the results obtained from the original algorithm: Applying the strategy by Wild et al. (2010) to the CRC data resulted in positive coefficients for CYFRA 21-1, CEA and OPN while coefficients for Ferritin and Seprase were negative.

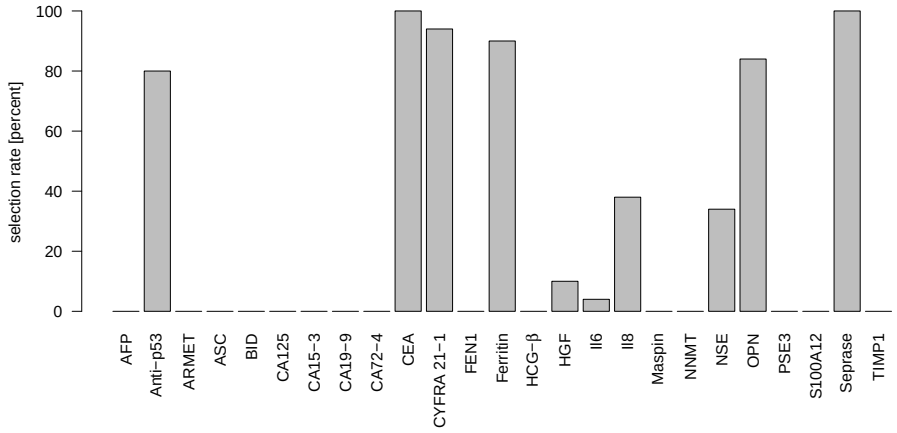

Figure 2: Prediction of colorectal cancer. Bars represent the selection rates of the 25 informative markers obtained from cross-validation (PAUC-GBS with FPR range $[0, 0.2]$).

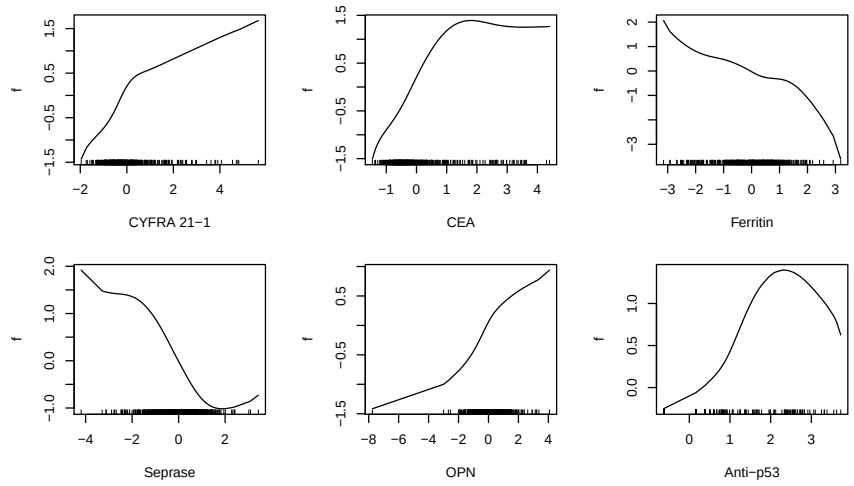

Figure 3: Prediction of colorectal cancer. The graphs represent the estimated partial functions of CYFRA 21-1, CEA, Ferritin, Seprase, Osteopontin (OPN) and Anti-p53 obtained from applying PAUC-GBS with FPR range $[0, 0.2]$ to the whole data set. Note that all markers were standardized before analysis.

4 Summary and Conclusion

Motivated by recent developments in molecular biology, extensive research has been undertaken to improve statistical methods for binary classification and marker selection. Although the performance of these methods has steadily improved, many of them share a well-recognized problem: The performance criteria used for the *derivation* of a prediction rule often differ from the criteria that are used for the practical *evaluation* of the rule. Consequently, many statistical techniques can potentially be improved if evaluation criteria are used directly for optimization and variable selection (cf. Cortes and Mohri 2004). Wild et al. (2010), for example, used the Lasso method (which is based on the binomial log-likelihood criterion) to derive their marker combination for the detection of CRC but evaluated the newly-found prediction rule by using sensitivity criteria at pre-specified specificity levels. The PAUC-GBS approach, on the other hand, circumvents the use of the binomial log-likelihood criterion by using the PAUC criterion for *both* optimization and evaluation of prediction rules. Sparsity of prediction rules is guaranteed by using the stability selection method by Meinshausen and Bühlmann (2010) instead of conventional cross-validation methods. The numerical results presented in Section 3 demonstrate that our method works remarkably well with respect to both variable selection and prediction accuracy. Although the differences between PAUC-GBS and the Lasso method appear to be small in our analysis on the detection of CRC, it is important to note that even small increases in performance may lead to enormous benefits for both patients and pharmaceutical companies.

Apart from these considerations, the PAUC-GBS algorithm presented in Section 2.4 can further be extended to address the following issues:

- *Alternative strategies for smoothing the PAUC criterion.* The approximation of the PAUC criterion used by PAUC-GBS can be refined by using different smoothing parameters for the indicator and min functions in equation (3). Wang and Chang (2010) argued that this approach is particularly useful when analyzing imbalanced data with a large amount of non-diseased people. Integrating two smoothing parameters into PAUC-GBS is straightforward, and the recommendations on how to choose the tuning parameters of PAUC-GBS apply accordingly.
- *Restriction to linear prediction rules.* As demonstrated in Section 3.1, PAUC-GBS is not only useful to detect nonlinear predictor-response relationships but can easily be restricted to estimate linear marker combinations only. This is achieved by using simple linear models as base-learners in step 1 of PAUC-GBS.

- *Mandatory predictor variables that are not subject to variable selection.* In clinical applications, it is often useful to investigate the added predictive value of a marker combination over traditional clinical variables such as sex and age. Incorporating these variables (that are not subject to variable selection) in PAUC-GBS is easily accomplished by first optimizing the PAUC criterion with the clinical variables only and by using the predictions obtained from this model as offset values in PAUC-GBS (cf. Boulesteix and Hothorn 2010).

The PAUC-GBS method proposed in this paper enforces sparsity in marker combinations by combining component-wise gradient boosting with stability selection. An alternative approach to feature selection in AUC regression has been proposed by Wang et al. (2011), who suggested to maximize an L_1-penalized version of the empirical AUC criterion (with specificity range $[0,1]$) over linear combinations of predictor variables. With this approach, sparsity of marker combinations is accomplished by shrinking the coefficients of non-informative predictor variables to zero. In contrast to the boosting method presented in this paper, Wang et al. (2011) did not approximate the empirical AUC criterion by a smoothed estimate but optimized the penalized empirical AUC directly with the help of support vector machine regression ("ROC-SVM"). Similar to the algorithms by Ma and Huang (2005) and Wang and Chang (2010), ROC-SVM requires the specification of an anchor marker (termed "baseline variable" by Wang et al. 2011) that enters the model a priori. A particularly appealing feature of ROC-SVM is the integration of hierarchical structures into the variable selection procedure. For example, if a medical screening test consists of several "stem" variables that are only measured if the value of a corresponding "root" variable has been collected beforehand, it is reasonable to select a stem variable only in combination with the stem variable. Estimation of such hierarchical structures is accomplished by using an appropriately specified L_1 penalty for ROC-SVM.

When comparing ROC-SVM to the PAUC-GBS method proposed in this paper, two conceptual issues arise:

1. From a practical perspective, it would be desirable to extend ROC-SVM to PAUC regression with a restricted range of specificity values. This strategy would amount to performing L_1-penalized SVM regression using the empirical PAUC criterion (3) in order to derive linear combinations of biomarkers.
2. Although hierarchical structures among predictor variables are not considered in this paper, it is of interest to extend PAUC-GBS in this direction. Integration of hierarchical structures could, for example, be achieved by an appropriate specification of multivariable base-learners ensuring that a stem variable is always selected together with its root variable.

The proposed PAUC-GBS method is implemented in the R add-on package **mboost**, which provides a well-established infrastructure for component-wise gradient boosting algorithms (Hothorn et al. 2011). Because mboost is based on a modular structure using separate implementations for the risk functions and the base-learners of a boosting algorithm, the smoothed negative PAUC criterion can easily be incorporated as a risk function into the package. An example on how to run PAUC-GBS in R is provided in the appendix of the paper.

Appendix - R code used for the simulation study

In this section we provide an example on how to run PAUC-GBS using the R add-on package **mboost** (Hothorn et al. 2011). Specifically, we show how to obtain the results of the simulation study presented in Section 3.1.

All risk functions implemented in mboost are specified via the `family` argument of the `gamboost` function. Typical examples of risk functions are given by the squared error loss and the negative binomial log-likelihood loss. Because the implementation of the risk functions is essentially independent of the other arguments of the `gamboost` function, it suffices to write a new `Family` function that implements the negative PAUC loss. This function is subsequently passed to the `gamboost` function. Concerning the specification of the base-learners and the step length factor of gradient boosting, we use the well-established infrastructure of the mboost package.

To incorporate the smoothed negative PAUC risk function into mboost, we first define a corresponding `Family` function named `PAUC`:

```
# load the mboost package
library(mboost)

# define the smoothed negative PAUC risk function;
# fprup corresponds to the upper limit of the FPR range

PAUC <- function (fprup = 1, sigma = 0.1) {

    approxGrad <- function(x) {
        exp(-x/sigma) / (sigma * (1 + exp(-x/sigma))^2)
    }
    approxLoss <- function(x) {
        1 / (1 + exp(-x / sigma))
    }
    Family(
        # implement the gradient of PAUC (formulas (2.6) and (2.7))
        ngradient = function(y, f, w = 1) {
        if (!all(w %in% c(0,1)))
            stop(sQuote("weights"), " must be either 0 or 1 for family ",
                 sQuote("PAUC"))
```

```
if (length(w) == 1) w <- rep(1, length(y))
ind1 <- which(y == 1)
ind0 <- which(y == -1)
n1 <- length(ind1)
n0 <- length(ind0)
if (length(f) == 1) {
    f <- rep(f, length(y))
    }
f <- f - f[w == 1][1]
# build weight matrix
tmp <- matrix(w[ind1], nrow = n0, ncol = n1, byrow = TRUE)
weightmat <- matrix(w[ind0], nrow = n0, ncol = n1) * tmp
# differences between "diseased" and "non-diseased"
M0 <- matrix(-f[ind1], nrow = n0, ncol = n1, byrow = TRUE) + f[ind0]
M1 <- approxGrad(M0) * weightmat
M2 <- approxLoss(M0) * weightmat
denom <- 1 + exp( (colSums(M2) / sum(w[ind0]) - fprup) / sigma )
ng <- vector(length(y), mode = "numeric")
ng[ind1] <- colSums(M1) / denom / sigma / (sum(w[ind1]))
ng[ind0] <- rowSums(- sweep(M1, 2, denom, FUN = "/")) / sigma /
            sum(w[ind1])
return(ng)
},
# implement the smoothed negative PAUC risk (formula (2.5))
risk = function(y, f, w = 1) {
if (length(w) == 1) w <- rep(1, length(y))
ind1 <- which(y == 1)
ind0 <- which(y == -1)
n1 <- length(ind1)
n0 <- length(ind0)
if (length(f) == 1) {
    f <- rep(f, length(y))
    }
f <- f - f[w == 1][1]
tmp <- matrix(w[ind1], nrow = n0, ncol = n1, byrow = TRUE)
weightmat <- matrix(w[ind0], nrow = n0, ncol = n1) * tmp
M0 <- matrix(-f[ind1], nrow = n0, ncol = n1, byrow = TRUE) + f[ind0]
M1 <- approxGrad(M0) * weightmat
M2 <- approxLoss(M0) * weightmat
num <- 1 + exp(fprup / sigma)
denom <- 1 + exp( (fprup - colSums(M2) / sum(w[ind0])) / sigma )
return( - (sum( fprup - sigma * log(num / denom) )) /
       (sum(w[ind1])) )
},
weights = "case", offset = function(y, w) {
0
},

check_y = function(y) {
if (!is.factor(y))
    stop("response is not a factor but ",
         sQuote("family = PAUCSigma()"))
if (nlevels(y) != 2)
    stop("response is not a factor at two levels but ",
         sQuote("family = AUC()"))
if (length(unique(y)) != 2)
    stop("only one class is present in response.")
ind1 <- which(y == levels(y)[2])
ind0 <- which(y == levels(y)[1])
n1 <- length(ind1)
```

```
        n0 <- length(ind0)
        c(-1, 1)[as.integer(y)]
        },
        rclass = function(f) (f > 0) + 1,
        name = paste("(1 - Partial AUC)-Loss"))
}
```

Having defined the `PAUC` risk function, we can use the `gamboost` function to obtain the results of the simulation study presented in Section 3.1:

```
# set n_0 = n_1 = 50 and start simulation study
n <- 50
coefList <- coefList2 <- list()

for (k in 1:100){

    # set seed to make results reproducible
    set.seed(k*2)

    # generate values of the informative predictor variables
    pred0 <- c(rbeta(n, 0.5, 100), rbeta(n, 0.1, 0.1))
    pred1 <- pred0 + rnorm(2 * n, sd = 0.3)
    pred1 <- as.numeric(scale(pred1, center = TRUE, scale = TRUE))
    pred2 <- pred0 + rnorm(2 * n, sd = 0.3)
    pred2 <- as.numeric(scale(pred2, center = TRUE, scale = TRUE))
    pred3 <- pred0 + rnorm(2 * n, sd = 0.3)
    pred3 <- as.numeric(scale(pred3, center = TRUE, scale = TRUE))
    pred0 <- c(rbeta(n, 0.4, 0.5), rbeta(n, 1.5, 0.3))
    pred4 <- pred0 + rnorm(2 * n, sd = 0.3)
    pred4 <- as.numeric(scale(pred4, center = TRUE, scale = TRUE))
    pred5 <- pred0 + rnorm(2 * n, sd = 0.3)
    pred5 <- as.numeric(scale(pred5, center = TRUE, scale = TRUE))
    pred6 <- pred0 + rnorm(2 * n, sd = 0.3)
    pred6 <- as.numeric(scale(pred6, center = TRUE, scale = TRUE))
    D <- data.frame(pred1, pred2, pred3, pred4, pred5, pred6)

    # generate values of the outcome variable
    y <- c(rep(0, n), rep(1, n))
    y <- as.factor(y)

    # generate values of the non-informative predictor variables
    noninform <- matrix(runif(500 * 2 * n), nrow = 2 * n)
    Noninform <- data.frame(noninform)
    Noninform <- data.frame(scale(Noninform, center = TRUE, scale = TRUE))
    names(Noninform) <- paste("x", 1:500, sep = "")
    namesvec <- c(paste("pred", 1:6, sep = ""), names(Noninform))
    Data <- data.frame(D, Noninform)
    INT <- rep(1, nrow(Data))

    # specify the base-learners for component-wise gradient boosting
    formula1 <- as.formula(paste("y ~", paste("bols(", namesvec, ",
                          intercept = FALSE)", sep = "",  collapse = " + ")))

    # run component-wise gradient boosting with FPR range [0,0.1]
    model1 <- gamboost(formula1, data = Data, family = PAUC(fprup = 0.1,
                      sigma = 0.376), control=boost_control(trace = TRUE,
                      mstop = 50, nu = 0.1))
```

```
    # generate subsamples for stability selection
    cv5f1 <- cv(model.weights(model1), type = "kfold", B = 5)
    cv5f2 <- cv(model.weights(model1), type = "kfold", B = 5)
    cv5f3 <- cv(model.weights(model1), type = "kfold", B = 5)
    cv5f4 <- cv(model.weights(model1), type = "kfold", B = 5)
    cv5f <- cbind(cv5f1, cv5f2, cv5f3, cv5f4)

    # run stability selection
    STAB <- stabsel(model1, FWER = 0.1, cutoff = 0.9, folds = cv5f)

    # re-run component-wise gradient boosting,
    # this time using the selected variables only
    if(length(names(STAB$selected)) > 0){
        blsnames2 <- paste(names(STAB$selected), sep = "", collapse = "+")
        formula2 <- as.formula(paste("y ~ ", paste(blsnames2, sep = "+")))} else {
        blsnames2 <- ""
        formula2 <- as.formula(paste("y ~ bols(INT, intercept = FALSE)"))
        }

    model1 <- gamboost(formula2, data=Data, family=PAUC(fprup = 0.1,
                       sigma = 0.376), control = boost_control(trace = TRUE,
                       mstop = 200, nu = 0.1))

    # save results
    coefList[[k]] <- coef(model1)
    save(coefList, file = "coef0001.rda")


    # run component-wise gradient boosting with FPR range [0,1]
    model2 <- gamboost(formula1, data = Data, family = PAUC(fprup = 1,
                       sigma = 0.376), control = boost_control(trace = TRUE,
                       mstop = 50, nu = 0.1))

    # run stability selection
    STAB <- stabsel(model2, FWER = 0.1, cutoff = 0.9, folds = cv5f)

    # re-run component-wise gradient boosting,
    # this time using the selected variables only
    if(length(names(STAB$selected)) > 0){
        blsnames2 <- paste(names(STAB$selected), sep = "", collapse = "+")
        formula2 <- as.formula(paste("y ~ ", paste(blsnames2, sep = "+")))} else {
        blsnames2 <- ""
        formula2 <- as.formula(paste("y ~ bols(INT, intercept = FALSE)"))
        }

    model2 <- gamboost(formula2, data = Data, family = PAUC(fprup = 1,
                       sigma = 0.376), control = boost_control(trace = TRUE,
                       mstop = 200, nu = 0.1))

    # save results
    coefList2[[k]] <- coef(model2)
    save(coefList2, file = "coef0010.rda")

}
```

References

Ahmed, I., A.-L. Hartikainen, M.-R. Järvelin, and S. Richardson (2011): "False discovery rate estimation for stability selection: Application to genome-wide association studies," *Statistical Applications in Genetics and Molecular Biology*, 10, Article 55.

Boulesteix, A.-L. (2004): "PLS dimension reduction for classification with microarray data," *Statistical Applications in Genetics and Molecular Biology*, 3, Article 33.

Boulesteix, A.-L. and T. Hothorn (2010): "Testing the additional predictive value of high-dimensional data," *BMC Bioinformatics*, 11:78.

Breiman, L. (2001): "Random forests," *Machine Learning*, 45, 5–32.

Bühlmann, P. (2006): "Boosting for high-dimensional linear models," *The Annals of Statistics*, 34, 559–583.

Bühlmann, P. and T. Hothorn (2007): "Boosting algorithms: Regularization, prediction and model fitting (with discussion)," *Statistical Science*, 22, 477–522.

Bühlmann, P. and T. Hothorn (2010): "Twin boosting: Improved feature selection and prediction," *Statistics and Computing*, 20, 119–138.

Bühlmann, P. and B. Yu (2003): "Boosting with the L_2 loss: Regression and classification," *Journal of the American Statistical Association*, 98, 324–338.

Cai, T. and L. E. Dodd (2008): "Regression analysis for the partial area under the ROC curve," *Statistica Sinica*, 18, 817–836.

Cortes, C. and M. Mohri (2004): "AUC optimization vs. error rate minimization," in S. Thrun, L. K. Saul, and B. Schölkopf, eds., *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*, Cambridge: MIT Press.

Dettling, M. (2003): *LogitBoost: An Implementation of the LogitBoost Classification Algorithm*, R package version 1.1. http://stat.ethz.ch/∼dettling/boosting.html.

Dettling, M. and P. Bühlmann (2003): "Boosting for tumor classification with gene expression data," *Bioinformatics*, 19, 1061–1069.

Dodd, L. E. and M. S. Pepe (2003): "Partial AUC estimation and regression," *Biometrics*, 59, 614–623.

Etzioni, R., N. Urban, S. Ramsey, M. McIntosh, S. Schwartz, B. Reid, J. Radich, G. Anderson, and L. Hartwell (2003): "The case for early detection," *Nature Reviews Cancer*, 3, 243–252.

Freund, Y. and R. Schapire (1997): "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, 55, 119–139.

Friedman, J. H. (2001): "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, 29, 1189–1232.

Friedman, J. H., T. Hastie, and R. Tibshirani (2000): "Additive logistic regression: A statistical view of boosting (with discussion)," *The Annals of Statistics*, 28, 337–407.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2011): *mboost: Model-Based Boosting*, R package version 2.1-0. https://r-forge.r-project.org/projects/mboost.

Komori, O. and S. Eguchi (2010): "A boosting method for maximizing the partial area under the ROC curve," *BMC Bioinformatics*, 11:314.

Ma, S. and J. Huang (2005): "Regularized ROC method for disease classification and biomarker selection with microarray data," *Bioinformatics*, 21, 4356–4362.

McClish, D. (1989): "Analyzing a portion of the ROC curve," *Medical Decision Making*, 9, 190–195.

Meinshausen, N. and P. Bühlmann (2010): "Stability selection," *Journal of the Royal Statistical Society, Series B*, 72.

Park, M. Y. and T. Hastie (2011): *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model*, R package version 0.95. http://cran.r-project.org/web/packages/glmpath.

Pepe, M. S. (2003): *The Statistical Evaluation of Medical Tests for Classification and Prediction*, Oxford: Oxford University Press.

Pepe, M. S., T. Cai, and G. Longton (2006): "Combining predictors for classification using the area under the receiver operating characteristic curve," *Biometrics*, 62, 221–229.

Pepe, M. S. and M. L. Thompson (2000): "Combining diagnostic test results to increase accuracy," *Biostatistics*, 1, 123–140.

Ridgeway, G. (2010): *gbm: Generalized Boosted Regression Models*, R package version 1.6-3.1. http://cran.r-project.org/web/packages/gbm.

Schmid, M. and T. Hothorn (2008): "Boosting additive models using component-wise P-splines," *Computational Statistics & Data Analysis*, 53, 298–311.

Teramoto, R. (2009): "Balanced gradient boosting from imbalanced data for clinical outcome prediction," *Statistical Applications in Genetics and Molecular Biology*, 8, Article 20.

Tibshirani, R. (1996): "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society, Series B*, 58, 267–288.

Vapnik, W. (2000): *The Nature of Statistical Learning Theory*, New York: Springer, 2nd edition.

Walter, S. D. (2005): "The partial area under the summary ROC curve," *Statistics in Medicine*, 24, 2025–2040.

Wang, Y., H. Chen, R. Li, N. Duan, and R. Lewis-Fernandez (2011): "Prediction-based structured variable selection through the receiver operating characteristic curves," *Biometrics*, 67, 896–905.

Wang, Z. (2011): "HingeBoost: ROC-based boost for classification and variable selection," *International Journal of Biostatistics*, 7.

Wang, Z. and Y.-C. I. Chang (2010): "Marker selection via maximizing the partial area under the ROC curve of linear risk scores," *Biostatistics*, 12, 369–385.

Wang, Z. and C. Y. Wang (2010): "Buckley-James boosting for survival analysis with high-dimensional biomarker data," *Statistical Applications in Genetics and Molecular Biology*, 9, Article 24.

Wild, N., H. Andres, W. Rollinger, F. Krause, P. Dilba, M. Tacke, and J. Karl (2010): "A combination of serum markers for the early detection of colorectal cancer," *Clinical Cancer Research*, 16, 6111–6121.

Zhang, D. D., X.-H. Zhou, D. H. Freeman, and J. L. Freeman (2002): "A non-parametric method for the comparison of partial areas under ROC curves and its application to large health care data sets," *Statistics in Medicine*, 21, 701–15.