Hindawi
Computational and Mathematical Methods in Medicine
Volume 2017, Article ID 1421409, 8 pages
https://doi.org/10.1155/2017/1421409

Research Article

Probing for Sparse and Fast Variable Selection with Model-Based Boosting

Janek Thomas,[1] Tobias Hepp,[2] Andreas Mayr,[2,3] and Bernd Bischl[1]

[1]*Department of Statistics, LMU München, München, Germany*
[2]*Department of Medical Informatics, Biometry and Epidemiology, FAU Erlangen-Nürnberg, Erlangen, Germany*
[3]*Department of Medical Biometry, Informatics and Epidemiology, University Hospital Bonn, Bonn, Germany*

Correspondence should be addressed to Tobias Hepp; tobias.hepp@uk-erlangen.de

Janek Thomas and Tobias Hepp contributed equally to this work.

Received 9 February 2017; Accepted 13 April 2017; Published 31 July 2017

Academic Editor: Yuhai Zhao

We present a new variable selection method based on model-based gradient boosting and randomly permuted variables. Model-based boosting is a tool to fit a statistical model while performing variable selection at the same time. A drawback of the fitting lies in the need of multiple model fits on slightly altered data (e.g., cross-validation or bootstrap) to find the optimal number of boosting iterations and prevent overfitting. In our proposed approach, we augment the data set with randomly permuted versions of the true variables, so-called shadow variables, and stop the stepwise fitting as soon as such a variable would be added to the model. This allows variable selection in a single fit of the model without requiring further parameter tuning. We show that our probing approach can compete with state-of-the-art selection methods like stability selection in a high-dimensional classification benchmark and apply it on three gene expression data sets.

1. Introduction

At the latest since the emergence of genomic and proteomic data, where the number of available variables p is possibly far higher than the sample size n, high-dimensional data analysis becomes increasingly important in biomedical research [1–4]. Since common statistical regression methods like ordinary least squares are unable to estimate model coefficients in these settings due to singularity of the covariance matrix, varying strategies have been proposed to select only truly influential, that is, informative, variables and discard those without impact on the outcome.

By enforcing sparsity in the true coefficient vector, regularized regression approaches like the *lasso* [5], *least angle regression* [6], *elastic net* [7], and *gradient boosting* algorithms [8, 9] perform variable selection directly in the model fitting process. This selection is controlled by tuning hyperparameters that define the degree of penalization. While these hyperparameters are commonly determined using resampling strategies like cross-validation, bootstrapping, and similar methods, the focus on minimizing the prediction error often results in the selection of many noninformative variables [10, 11].

One approach to address this problem is *stability selection* [12, 13], a method that combines variable selection with repeated subsampling of the data to evaluate selection frequencies of variables. While stability selection can considerably improve the performance of several variable selection methods including regularized regression models in high-dimensional settings [12, 14], its application depends on additional hyperparameters. Although recommendations for reasonable values exist [12, 14], proper specification of these parameters is not straightforward in practice as the optimal configuration would require a priori knowledge about the number of informative variables. Another potential drawback is that stability selection increases the computational demand, which can be problematic in high-dimensional settings if the computational complexity of the used selection technique scales superlinearly with the number of predictor variables.

In this paper, we propose a new method to determine the optimal number of iterations in model-based boosting for variable selection inspired by *probing*, a method frequently used in related areas of machine learning research [15–17] and the analysis of microarrays [18]. The general notion of probing involves the artificial inflation of the data with random noise variables, so-called *probes* or *shadow variables*. While this approach is in principle applicable to the lasso or least angle regression as well, it is especially attractive to use with more computationally intensive boosting algorithms, as no resampling is required at all. Using the first selection of a shadow variable as stopping criterion, the algorithm is applied only once without the need to optimize any hyperparameters in order to extract a set of informative variables from the data, thereby making its application very fast and simple in practice. Furthermore, simulation studies show that the resulting models in fact tend to be more strictly regularized compared to the ones resulting from cross-validation and contain less uninformative variables.

In Section 2, we provide detailed descriptions of the model-based gradient boosting algorithm as well as stability selection and the new probing approach. Results of a simulation study comparing the performance of probing to cross-validation and different configurations of stability selection in a binary classification setting are then presented in Section 3 before discussing the application of these methods on three data sets with measurements of gene expression levels in Section 4. Section 5 summarizes our findings and presents an outlook to extensions of the algorithm.

2. Methods

2.1. Gradient Boosting. Given a learning problem with a data set $D = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1,\ldots,n}$ sampled i.i.d. from a distribution over the joint space $\mathcal{X} \times \mathcal{Y}$, with a p-dimensional input space $\mathcal{X} = (\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_p)$ and an output space $\mathcal{Y}$ (e.g., $\mathcal{Y} = \mathbb{R}$ for regression and $\mathcal{Y} = \{0, 1\}$ for binary classification), the aim is to estimate a function, $f(\mathbf{x})$, $\mathcal{X} \to \mathcal{Y}$, that maps elements of the input space to the output space as good as possible. Relying on the perspective on boosting as gradient descent in function space, gradient boosting algorithms try to minimize a given loss function, $\rho(y^{(i)}, f(\mathbf{x}^{(i)}))$, $\rho : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$, that measures the discrepancy between a predicted outcome value of $f(\mathbf{x}^{(i)})$ and the true $y^{(i)}$. Minimizing this discrepancy is achieved by repeatedly fitting weak prediction functions, called *base learners*, to previous mistakes, in order to combine them to a strong ensemble [19]. Although early implementations in the context of machine learning focused specifically on the use of regression trees, the concept has been successfully extended to suit the framework of a variety of statistical modelling problems [8, 20]. In this model-based approach, the *base learners* $h(\mathbf{x})$ are typically defined by semiparametric regression functions on $\mathbf{x}$ to build an additive model. A common simplification is to assume that each base learner h_j is defined on only one component x_j of the input space

$$f(\mathbf{x}) = \beta_0 + h_1(x_1) + \cdots + h_p(x_p). \tag{1}$$

For an overview of the fitting process of model-based boosting see Algorithm 1.

Algorithm 1 (model-based gradient boosting). Starting at $m = 0$ with a constant loss minimal initial value $\widehat{f}^{[0]}(\mathbf{x}) \equiv c$, the algorithm iteratively updates the predictor with a small fraction of the base learner with the best fit on the negative gradient of the loss function:

(1) Set iteration counter $m := m + 1$.

(2) While $m \leq m_{\text{stop}}$, compute the negative gradient vector of the loss function:

$$u^{(i)} = -\left.\frac{\partial \rho(y, f)}{\partial f}\right|_{f=\widehat{f}^{[m-1]}(\mathbf{x}^{(i)}), y=y^{(i)}}. \tag{2}$$

(3) Fit every base learner $h_j^{[m]}(x_j)$ separately to the negative gradient vector $\mathbf{u}$.

(4) Find $\widehat{h}_{j^*}^{[m]}(\mathbf{x}_{j^*})$, that is, the base learner with the best fit:

$$j^* = \arg\min_{1 \leq j \leq p} \sum_{i=1}^{n} \left(u^{(i)} - \widehat{h}_j^{[m]}\left(x_j^{(i)}\right)\right)^2. \tag{3}$$

(5) Update the predictor with a small fraction $0 \leq \nu \leq 1$ of this component:

$$\widehat{f}(\mathbf{x})^{[m]} = \widehat{f}(\mathbf{x})^{[m-1]} + \nu \cdot \widehat{h}_{j^*}^{[m]}\left(x_{j^*}\right). \tag{4}$$

The resulting model can be interpreted as a generalized additive model with partial effects for each covariate contained in the additive predictor. Although the algorithm relies on two hyperparameters ν and m_{stop}, Bühlmann and Hothorn [9] claim that the *learning rate* ν is of minor importance as long as it is "sufficiently small," with $\nu = 0.1$ commonly used in practice.

The stopping criterion, m_{stop}, determines the degree of regularization and thereby heavily affects the model quality in terms of overfitting and variable selection [21]. However, as already outlined in the introduction, optimizing m_{stop} using common approaches like cross-validation results in the selection of many uninformative variables. Although still focusing on minimizing prediction error, using a 25-fold bootstrap instead of the commonly used 10-fold cross-validation tends to return sparser models without sacrificing prediction performance [22].

2.2. Stability Selection. The weak performance of cross-validation regarding variable selection partly results from the fact that it pursues the goal of minimizing the prediction error instead of selecting only informative variables. One possible solution is the *stability selection* framework [12, 13], a very versatile algorithm that can be combined with all kinds of variable selection methods like gradient boosting, lasso, or forward stepwise selection. It produces sparser solutions by controlling the number of false discoveries. Stability selection defines an upper bound for the per-family error rate (PFER),

for example, the expected number of uninformative variables $\mathbb{E}(V)$ included in the final model.

Therefore, using stability selection with model-based boosting means that Algorithm 1 is run independently on B random subsamples of the data until either a predefined number of iterations m_{stop} is reached or q different variables have been selected. Subsequently, all variables are sorted with respect to their selection frequency in the B sets. The amount of informative variables is then determined by a user-defined threshold π_{thr} that has to be exceeded. A detailed description of these steps is given in Algorithm 2.

Algorithm 2 (stability selection for model-based boosting [14]).

(1) For $b = 1, \ldots, B$,

  (a) draw a subset of size $\lfloor n/2 \rfloor$ from the data;
  (b) fit a boosting model to the subset until the number of selected variables is equal to q or the number of iterations reaches a prespecified number (m_{stop}).

(2) Compute the selection frequencies per variable j:

$$\widehat{\pi}_j \coloneqq \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}_{\{j \in \widehat{S}_b\}}, \tag{5}$$

where $\widehat{S}_b$ denotes the set of selected variables in iteration b.

(3) Select variables with a selection frequency of at least π_{thr}, which yields a set of stable covariates:

$$\widehat{S}_{\text{stable}} \coloneqq \left\{ j : \widehat{\pi}_j \geq \pi_{\text{thr}} \right\}. \tag{6}$$

Following this approach, the upper bound for the PFER can be derived as follows [12]:

$$\mathbb{E}(V) \leq \frac{q^2}{(2\pi_{\text{thr}} - 1)\, p}. \tag{7}$$

With additional assumptions on exchangeability and shape restrictions on the distribution of simultaneous selection, even tighter bounds can be derived [13]. While this method is successfully applied in a large number of different applications [23–26], several shortcomings impede the usage in practice. First off, three additional hyperparameters π_{thr}, PFER, and q are introduced. Although only two of them have to be specified by the user (the third one can be calculated by assuming equality in (7)), it is not intuitively clear which parameter should be left out and how to specify the remaining two. Even though recommendations for reasonable settings for the selection threshold [12] or the PFER [14] are proposed, the effectiveness of these settings is difficult to evaluate in practical settings. The second obstacle in the usage of stability selection is the considerable computational power required for calculation. Overall B boosting models ([13] recommends $B = 100$) have to be fitted and a reasonable m_{stop} has to be found as well, which will most likely require cross-validation. Even though this process can be parallelized quite easily, complex model classes with smooth and higher-order effects can become extremely costly to fit.

2.3. Probing. The approach of adding *probes* or *shadow variables*, for example, artificial uninformative variables to the data, is not completely new and has already been investigated in some areas of machine learning. Although they share the underlying idea to benefit from the presence of variables that are known to be independent from the outcome, the actual implementation of the concept differs (see Guyon and Elisseeff (2003) [15] for an overview). An especially useful approach, however, is to generate these additional variables as randomly shuffled versions of all observed variables. These permuted variables will be called *shadow variables* for the remainder of this paper and are denoted as $\tilde{x}_j$. Compared to adding randomly sampled variables, shadow variables have the advantage that the marginal distribution of x_j is preserved in $\tilde{x}_j$. This approach is tightly connected to the theory of permutation tests [27] and is used similarly for *all-relevant* variable selection with random forests [28].

Implementing the *probing* concept to the sequential structure of model-based gradient boosting is rather straightforward. Since boosting algorithms proceed in a greedy fashion and only update the effect which yields the largest loss reduction in each iteration, selecting a shadow variable essentially implies that the best possible improvement at this stage relies on information that is known to be unrelated to the outcome. As a consequence, variables that are selected in later iterations are most likely correlated to y only by chance as well. Therefore, all variables that have been added prior to the first shadow variable are assumed to have a true influence on the target variable and should be considered informative. A description of the full procedure is presented in Algorithm 3.

Algorithm 3 (probing for variable selection in model-based boosting).

(1) *Expand* the data set X by creating randomly shuffled images $\tilde{x}_j$ for each of the $j = 1, \ldots, p$ variables x_j such that

$$\tilde{x}_j \in S_{x_j}, \tag{8}$$

where S_{x_j} denotes the symmetric group that contains all $n!$ possible permutations of x_j.

(2) *Initialize* a boosting model on the inflated data set

$$\overline{X} = \left[x_1 \cdots x_p \ \tilde{x}_1 \cdots \tilde{x}_p \right] \tag{9}$$

and start iterations with $m = 0$.

(3) *Stop if* the first $\tilde{x}_j$ is selected; see Algorithm 1 step (3).

(4) *Return* only the variables selected from the original data set X.

The major advantage of this approach compared to variable selection via cross-validation or stability selection is that one model fit is enough to find informative variables and no expensive refitting of the model is required. Additionally, there is no need for any prespecification like the search space (m_{stop}) for cross-validation or additional hyperparameters (q, π_{thr}, PFER) for stability selection. However, it should be noted that, unlike classical cross-validation, probing aims at optimal variable selection instead of prediction performance of the algorithm. Since this usually involves stopping much earlier, the effect estimates associated with the selected variables are most likely strongly regularized and might not be optimal for predictions.

3. Simulation Study

In order to evaluate the performance of our proposed variable selection method, we conduct a benchmark simulation study where we compare the set of nonzero coefficients determined by the use of shadow variables as stopping criterion to cross-validation and different configurations of stability selection. We simulate n data points for p variables from a multivariate normal distribution $X \sim \mathcal{N}(0, \Sigma)$ with Toeplitz correlation structure $\Sigma_{ij} = \rho^{|i-j|}$ for all $1 < i, j < p$ and $\rho = 0.9$. The response variable $y^{(i)}$ is then generated by sampling Bernoulli experiments with probability

$$\pi^{(i)} = \frac{\exp\left(\eta^{(i)}\right)}{1 + \exp\left(\eta^{(i)}\right)}, \tag{10}$$

with $\eta^{(i)}$ the linear predictor for the ith observation $\eta^{(i)} = X^{(i)}\beta$ and all nonzero elements of β sampled from $\mathcal{U}(-1, 1)$. Since the total amount of nonzero coefficients determines the number of informative variables in the setting, it is denoted as p_{inf}.

Overall, we consider 12 different simulation scenarios defined by all possible combinations of $n \in \{100, 500\}$, $p \in \{100, 500, 1000\}$, and $p_{\text{inf}} \in \{5, 20\}$. Specifically, this leads to the evaluation of 2 low-dimensional settings with $p < n$, 4 settings with $p = n$, and 6 high-dimensional settings with $p > n$. Each configuration is run 100 times. Along with new realizations of X and y, we also draw new values for the nonzero coefficients in β and sample their position in the vector in each run to allow for varying correlation patterns among the informative variables. For variable selection with cross-validation, 25-fold bootstrap (the default in `mboost`) is used to determine the final number of iterations. Different configurations of stability selection were tested to investigate whether and, if so, to what extent these settings affect the selection. In order to explicitly use the upper error bounds of stability selection, we decided to specify 9 combinations with PFER $\in \{1, 2.5, 8\}$ and $\pi_{\text{thr}} \in \{0.6, 0.75, 0.9\}$ and calculate q from (7). Aside from the learning rate ν, which is set to 0.1 for all methods, no further parameters have to be specified for the probing scheme. Two performance measures are considered for the evaluation of the methods with respect to variable selection: first, the true positive rate (TPR) as the fraction of (correctly) selected variables from all true informative variables and, second, the false discovery rate (FDR) as the fraction of uninformative variables in the set of selected variables. To ensure reproducibility the R package `batchtools` [29] was used for all simulations.

The results of the simulations for all settings are illustrated in Figure 1. With TPR and FDR on the y-axis and x-axis, respectively, solutions displayed in the top left corner of the plots therefore successfully separate p_{inf} informative variables from the ones without true effect on the response. Although already using a sparse cross-validation approach, the FDR of variable selection via cross-validation is still relatively high, with more than 50% false positives in the selected sets in the majority of the simulated scenarios. Whereas this seems to be mostly disadvantageous in the cases where $p_{\text{inf}} = 5$, the trend to more greedy solutions leads to a considerably higher chance of identifying more of the truly informative variables if $p_{\text{inf}} = 20$ or with very high p, however, still at the price of picking up many noise variables on the way. Pooling the results of all configurations considered for stability selection, the results cover a large area of the performance space in Figure 1, thereby probably indicating high sensitivity on the decisions regarding the three tuning parameters.

Examining the results separately in Figure 2, the dilemma is particularly clearly illustrated for $p_{\text{inf}} = 20$ and $n = 500$. Despite being able to control the upper bounds for expected false positive selections, only a minority of the true effects are selected if the PFER is set too conservative. In addition, the high variance of the FDR observed for these configurations in some settings somewhat counteracts the goal to achieve more certainty about the selected variables one might probably pursue by setting the PFER very low. The performance of probing, on the other hand, reveals a much more stable pattern and outperforms stability selection in the difficult $p_{\text{inf}} = 20$ and $n = 100$ settings. In fact, the TPR is either higher or similar to all configurations used for stability selection, but exhibiting slightly higher FDR especially in settings with $n = 500$. Interestingly, probing seems to provide results similar to those of stability selection with PFER = 8, raising the question if the use of shadow variables allows statements about the number of expected false positives in the selected variable set.

Considering the runtime, however, we can see that probing is orders of magnitudes faster with an average runtime of less than a second compared to 12 seconds for cross-validation and almost one minute for stability selection.

4. Application on Gene Expression Data

In this section we exploit the usage of probing as a tool for variable selection on three gene expression data sets. More specifically, this includes data from using oligonucleotide arrays for colon cancer detection [30] with 40 tumor and 22 regular colon tissue samples and $p = 2000$ measured genes expression levels. In addition, we analyse data from a study aiming to predict metastasis of breast carcinoma [31], where patients were labelled good or poor ($n = 111$ and $n = 57$, resp.) depending on whether they remained event-free for a five-year period after diagnosis or not. The data set contains log-transformed expression levels of $p =$

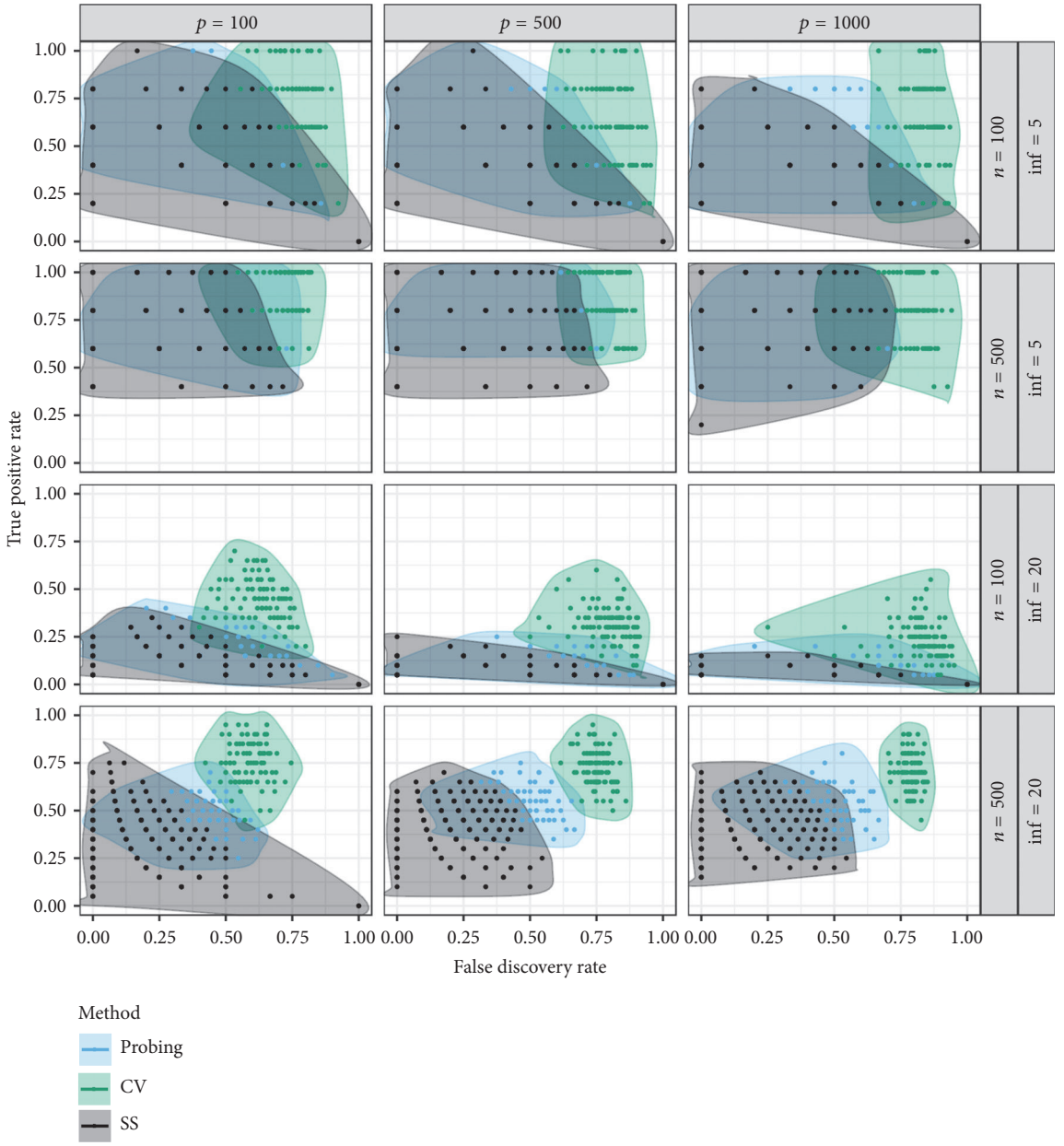

Figure 1: True positive rate (on y-axis) and false discovery rate (on x-axis) for three different, boosting-based variable selection algorithms, probing (black), stability selection (green), cross-validation (blue), and different simulation settings: $n \in \{100, 500\}$, $p \in \{100, 500, 1000\}$, and $p_{\text{inf}} \in \{5, 20\}$. All settings of stability selection are combined. Shaded areas are smooth hulls around all observed values.

2905 genes. The last example examines riboflavin production by *Bacillus subtilis* [32] with $n = 71$ observations of log-transformed riboflavin production rates and expression level for $p = 4088$ genes. All data are publicly available via `R` packages `datamicroarray` and `hdi`. Our proposed probing approach is implemented in a fork of the `mboost` [33] software for component-wise gradient boosting. It can be easily used by setting `probe=TRUE` in the `glmboost()` call.

In order to evaluate the results provided by the new approach, we analysed the data using cross-validation, stability selection [34], and the lasso [35] for comparison. Table 1 shows the total number of variables selected by each method along with the size of the intersection between the sets. Starting with the probably least surprising result, boosting with cross-validation leads to the largest set of selected variables in all examples, whereas using probing as stopping criterion instead clearly reduces these sets. Since both approaches are based on the same regularization profile until the first shadow variable enters the model, the less regularized solution of cross-validation always contains all variables selected with probing. For stability selection, we used the conservative approach with PFER = 1 and $q = 20$ as suggested by Bühlmann et al. (2014) [32]. As a consequence, the set of variables considered to be informative further

Table 1: Total number of selected variables and intersection size for four variable selection techniques (boosting with 25-fold bootstrap, probing, stability selection, and the lasso with 10-fold cross-validation) on three gene expression data sets. The last column compares algorithm runtime in seconds.

	Cross-validation	Probing	Stability selection	Lasso (`glmnet`)	Runtime (sec.)
			Colon cancer		
Cross-validation	**9**				10.52
Probing	5	**5**			1.78
Stability selection	3	3	**3**		49.4
Lasso (`glmnet`)	7	5	3	**7**	0.4
			Breast carcinoma		
Cross-validation	**32**				24
Probing	14	**14**			4.39
Stability selection	1	1	**1**		102.28
Lasso (`glmnet`)	14	14	1	**14**	1.13
			Riboflavin production		
Cross-validation	**50**				14.2
Probing	10	**10**			6.89
Stability selection	5	5	**5**		66.46
Lasso (`glmnet`)	23	7	4	**30**	0.68

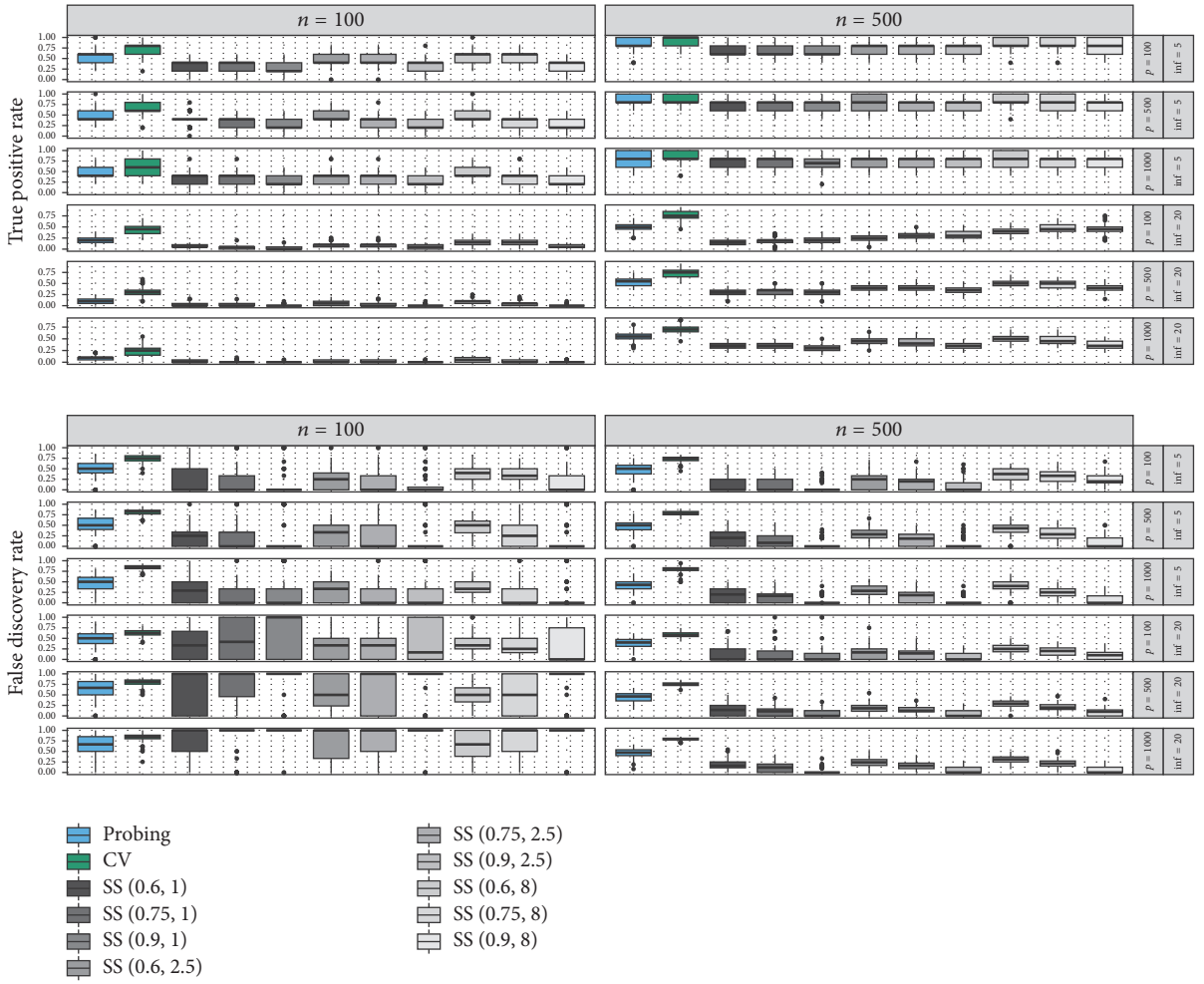

Figure 2: Boxplots of true positive rate (top) and false discovery rate (bottom) for different simulation settings and the three boosting-based, variable selection algorithms. Different Stability selection settings are denoted by SS(π_{thr}, PFER).

shrinks in all three scenarios. Again, these results clearly reflect the findings from the simulation study in Section 3, placing the probing approach between stability selection with probably overly conservative error bound and the greedy selection with cross-validation.

Since so far all approaches rely on boosting algorithms, we additionally considered variable selection with the lasso. We used the default settings of the `glmnet` package for R to calculate the lasso regularization path and determine the final model via 10-fold cross-validation [35]. Although the lasso already tends to result in sparser models under these conditions compared to model-based boosting [22], `glmnet` additionally uses a "one-standard-error rule" to regularize the solution even further. In fact, this leads to the selection of an identical set of genes as probing for the breast carcinoma example, but the final models estimated for both other examples still contain a higher number of variables. This is especially the case for the data on riboflavin production, where the lasso solution is further not simply a subset of the cross-validated boosting approach and only agrees on 23 mutually selected variables. Interestingly, even one of the 5 variables proposed by stability selection is also missing. The R code used for this analysis can be found in the Supplementary Material of this manuscript available online at https://doi.org/10.1155/2017/1421409.

5. Conclusion

We proposed a new approach to determine the optimal number of iterations for sparse and fast variable selection with model-based boosting via the addition of probes or shadow variables (*probing*). We were able to demonstrate via a simulation study and the analysis of gene expression data that our approach is both a feasible and convenient strategy for variable selection in high-dimensional settings. In contrast to common tuning procedures for model-based boosting which rely on resampling or cross-validation procedures to optimize the prediction accuracy [21], our probing approach directly addresses the variable selection properties of the algorithm. As a result, it substantially reduces the high number of false discoveries that arise with standard procedures [14] while only requiring a single model fit to obtain the set of parameters.

Aside from the very short runtime, another attractive feature of probing is that no additional tuning parameters have to be specified to run the algorithm. While this greatly increases its ease of use, there is, of course, a trade-off regarding flexibility, as the lack of tuning parameters means that there is no way to steer the results towards more or less conservative solutions. However, a corresponding tuning approach in the context of probing could be to allow a certain amount of selected probes in the model before deciding to stop the algorithm (cf. Guyon and Elisseeff, 2003 [15]). Although variables selected after the first probe can be labelled informative less convincingly, this resembles the uncertainty that comes with specifying higher values for the error bound of stability selection.

A potential drawback of our approach is that due to the stochasticity of the permutations, there is no deterministic solution and the selected set might slightly vary after rerunning the algorithm. In order to stabilize results, probing could also be used combined with resampling to determine the optimal stopping iteration for the algorithm by running the procedure on several bootstrap samples first. Of course, this requires the computation of multiple models and therefore again increases the runtime of the whole selection procedure.

Another promising extension could be a combination with stability selection. With each model stopping at the first shadow variable, only the selection threshold π_{thr} has to be specified. However, since this means a fundamental change of the original procedure, further research on this topic is necessary to better assess how this could affect the resulting error bound.

While in this work we focused on gradient boosting for binary and continuous data, there is no reason why our results should not also carry over to other regression settings or related statistical boosting algorithms as likelihood-based boosting [36]. Likelihood-based boosting follows the same principle idea but uses different updates, coinciding with gradient boosting in case of Gaussian responses [37]. Further research is also warranted on extending our approach to multidimensional boosting algorithms [25, 38], where variables have to be selected for various models simultaneously.

In addition, probing as a tuning scheme could be generally also combined with similar regularized regression approaches like the lasso [5, 22]. Our proposal for model-based boosting hence could be a starting point for a new way of tuning algorithmic models for high-dimensional data, not with the focus on prediction accuracy, but addressing directly the desired variable selection properties.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work of authors Tobias Hepp and Andreas Mayr was supported by the Interdisciplinary Center for Clinical Research (IZKF) of the Friedrich-Alexander-University Erlangen-Nürnberg (Project J49). The authors additionally acknowledge support by Deutsche Forschungsgemeinschaft and Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) within the funding programme Open Access Publishing.

References

[1] R. Romero, J. Espinoza, F. Gotsch et al., "The use of high-dimensional biology (genomics, transcriptomics, proteomics, and metabolomics) to understand the preterm parturition syndrome," *BJOG: An International Journal of Obstetrics and Gynaecology*, vol. 113, no. s3, pp. 118–135, 2006.

[2] R. Clarke, H. W. Ressom, A. Wang et al., "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature Reviews Cancer*, vol. 8, no. 1, pp. 37–49, 2008.

[3] P. Mallick and B. Kuster, "Proteomics: a pragmatic perspective," *Nature Biotechnology*, vol. 28, no. 7, pp. 695–709, 2010.

[4] M. L. Bermingham, R. Pong-Wong, A. Spiliopoulou et al., "Application of high-dimensional feature selection: evaluation for genomic prediction in man," *Scientific Reports*, vol. 5, Article ID 10312, 2015.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[7] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[9] P. Bühlmann and T. Hothorn, "Boosting algorithms: regularization, prediction and model fitting," *Statistical Science*, vol. 22, no. 4, pp. 477–505, 2007.

[10] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the lasso," *The Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[11] C. Leng, Y. Lin, and G. Wahba, "A note on the lasso and related procedures in model selection," *Statistica Sinica*, vol. 16, no. 4, pp. 1273–1284, 2006.

[12] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 72, no. 4, pp. 417–473, 2010.

[13] R. D. Shah and R. J. Samworth, "Variable selection with error control: another look at stability selection," *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 75, no. 1, pp. 55–80, 2013.

[14] B. Hofner, L. Boccuto, and M. Göker, "Controlling false discoveries in high-dimensional situations: boosting with stability selection," *BMC Bioinformatics*, vol. 16, no. 1, article 144, 2015.

[15] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[16] J. Bi, K. P. Bennett, M. Embrechts, C. M. Breneman, and M. Song, "Dimensionality reduction via sparse support vector machines," *Journal of Machine Learning Research*, vol. 3, pp. 1229–1243, 2003.

[17] Y. Wu, D. D. Boos, and L. A. Stefanski, "Controlling variable selection by the addition of pseudovariables," *Journal of the American Statistical Association*, vol. 102, no. 477, pp. 235–243, 2007.

[18] V. G. Tusher, R. Tibshirani, and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 9, pp. 5116–5121, 2001.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.

[20] G. Ridgeway, "The state of boosting," *Computing Science and Statistics*, vol. 31, pp. 172–181, 1999.

[21] A. Mayr, B. Hofner, and M. Schmid, "The importance of knowing when to stop: a sequential stopping rule for component-wise gradient boosting," *Methods of Information in Medicine*, vol. 51, no. 2, pp. 178–186, 2012.

[22] T. Hepp, M. Schmid, O. Gefeller, E. Waldmann, and A. Mayr, "Approaches to regularized regression—a comparison between gradient boosting and the lasso," *Methods of Information in Medicine*, vol. 55, no. 5, pp. 422–430, 2016.

[23] A.-C. Haury, F. Mordelet, P. Vera-Licona, and J.-P. Vert, "TIGRESS: trustful inference of gene regulation using stability selection," *BMC Systems Biology*, vol. 6, article 145, 2012.

[24] S. Ryali, T. Chen, K. Supekar, and V. Menon, "Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty," *NeuroImage*, vol. 59, no. 4, pp. 3852–3861, 2012.

[25] J. Thomas, A. Mayr, B. Bischl, M. Schmid, A. Smith, and B. Hofner, Stability selection for component-wise gradient boosting in multiple dimensions, 2016.

[26] A. Mayr, B. Hofner, and M. Schmid, "Boosting the discriminatory power of sparse survival models via optimization of the concordance index and stability selection," *BMC Bioinformatics*, vol. 17, no. 1, article 288, 2016.

[27] H. Strasser and C. Weber, "The asymptotic theory of permutation statistics," *Mathematical Methods of Statistics*, vol. 8, no. 2, pp. 220–250, 1999.

[28] M. B. Kursa, A. Jankowski, and W. . Rudnicki, "Boruta—a system for feature selection," *Fundamenta Informaticae*, vol. 101, no. 4, pp. 271–285, 2010.

[29] M. Lang, B. Bischl, and D. Surmann, "batchtools: Tools for R to work on batch systems," *The Journal of Open Source Software*, vol. 2, no. 10, 2017.

[30] U. Alon, N. Barka, D. A. Notterman et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

[31] E. Gravier, G. Pierron, A. Vincent-Salomon et al., "A prognostic DNA signature for T1T2 node-negative breast cancer patients," *Genes Chromosomes and Cancer*, vol. 49, no. 12, pp. 1125–1134, 2010.

[32] P. Bühlmann, M. Kalisch, and L. Meier, "High-dimensional statistics with a view toward applications in biology," *Annual Review of Statistics and Its Application*, vol. 1, no. 1, pp. 255–278, 2014.

[33] T. Hothorn, P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner, mboost: Model-Based Boosting. R package version R package version 2.7-0, 2016.

[34] B. Hofner and T. Hothorn, stabs: Stability Selection with Error Control. R package version R package version 0.5-1, 2015.

[35] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, 2010.

[36] G. Tutz and H. Binder, "Generalized additive modeling with implicit variable selection by likelihood-based boosting," *Biometrics*, vol. 62, no. 4, pp. 961–971, 2006.

[37] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, "The evolution of boosting algorithms: From machine learning to statistical modelling," *Methods of Information in Medicine*, vol. 53, no. 6, pp. 419–427, 2014.

[38] A. Mayr, N. Fenske, B. Hofner, T. Kneib, and M. Schmid, "Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting," *Journal of the Royal Statistical Society. Series C. Applied Statistics*, vol. 61, no. 3, pp. 403–427, 2012.