The Importance of Knowing When to Stop

A Sequential Stopping Rule for Component-wise Gradient Boosting

A. Mayr; B. Hofner; M. Schmid
Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Keywords
Gradient boosting, resampling methods, early stopping, variable selection, penalized regression

Summary
Objectives: Component-wise boosting algorithms have evolved into a popular estimation scheme in biomedical regression settings. The iteration number of these algorithms is the most important tuning parameter to optimize their performance. To date, no fully automated strategy for determining the optimal stopping iteration of boosting algorithms has been proposed.

Methods: We propose a fully data-driven sequential stopping rule for boosting algorithms. It combines resampling methods with a modified version of an earlier stopping approach that depends on AIC-based information criteria. The new “subsampling after AIC” stopping rule is applied to component-wise gradient boosting algorithms.

Results: The newly developed sequential stopping rule outperformed earlier approaches if applied to both simulated and real data. Specifically, it improved purely AIC-based methods when used for the microarray-based prediction of the recurrence of metastases for stage II colon cancer patients.

Conclusions: The proposed sequential stopping rule for boosting algorithms can help to identify the optimal stopping iteration already during the fitting process of the algorithm, at least for the most common loss functions.

1. Introduction

In modern biometric research, the selection of the most informative variables to generate accurate predictions for a clinical outcome is often one of the key research questions. In the present work, we will re-evaluate a recent study conducted by Barrier et al. [2], where the outcome of interest was the development of metachronous metastases for stage II colon cancer patients after surgery, with 22,283 gene expression levels from tumor samples available as possible predictors. The aim of the study was to develop an accurate and interpretable prediction rule combining the expression levels of a small subset containing only the most informative genes.

One of the most promising methodological approaches to select and estimate an accurate additive predictor is model-based boosting. Boosting was first introduced in the machine learning community with the now famous AdaBoost algorithm for classification [3]. The basic idea of AdaBoost is to iteratively build an ensemble of simple classification tools (“weak learners”) in order to provide a more precise overall classification rule. This is obtained by boosting the importance of misclassified observations in the learning process as well as assigning higher weights to better performing classifiers in the final aggregation. Later, boosting was generalized to a gradient descent algorithm in function space [4, 5] that iteratively fits simple regression models (“base-learners”) to the negative gradient of a loss function and results in a regression-type additive predictor. This algorithm (“component-wise gradient boosting”) is the foundation of the model-based boosting approach: In every boosting iteration, the current estimate of the additive predictor is updated by adding the fit of the best performing base-learner. The resulting final model can be naturally interpreted as a (generalized) additive model with partial effects for each covariate that is contained in the additive predictor. An overview of component-wise gradient boosting algorithms can be found in Bühlmann and Hothorn [6]. However, we also acknowledge that there exist controversial views on boosting [7] with discussion.

The main tuning parameter of boosting algorithms is the stopping iteration, which is commonly denoted as “mstop.” Determining the optimal stopping iteration of a boosting algorithm, i.e., stopping the algorithm before its convergence (“early stopping”) is crucial because this strategy prevents overfitting the data and typically improves prediction accuracy. In case of model-based boosting, early stopping also incorporates variable selection and shrinkage of effect estimates in the fitting process: For every available variable in the candidate model, one base-learner is specified. As in every boosting iteration only the best-fit-
In practice, the ability of boosting algorithms to find a sparse, interpretable model with good prediction accuracy in a fully data-driven manner is only limited by the problem of selecting the optimal stopping iteration \( m_{\text{stop}} \). Many authors have stated that the algorithm should be stopped before convergence to avoid overfitting, and that \( m_{\text{stop}} \) “may be selected using some cross-validation scheme” [6], without giving closer advise on the grid and the maximal value of \( m_{\text{stop}} \) that should be used to search the optimal model. It is hence common practice to run the algorithm until an arbitrarily chosen very large \( m_{\text{stop}} \). Afterwards, the resulting models from all iterations are evaluated, selecting the optimal iteration based on resampling methods or information criteria as the AIC [19]. This strategy is problematic for two reasons: First, it requires a priori knowledge to specify the initial (very large) \( m_{\text{stop}} \). Second, it is computationally ineffective, because in many cases it is unnecessary to run boosting until the large initial iteration as the true optimal \( m_{\text{stop}} \) is in fact much smaller.

Chang et al. [1] tried to overcome these problems by proposing an AIC change-point detection algorithm that aims at stopping the boosting algorithm already during the initial fitting process. The algorithm is stopped if the minimum of the AIC (which is re-computed in each iteration) is reached. The main problem of this approach is that it solely depends on the AIC criterion, which requires estimates for the degrees of freedom of the model in each iteration. This issue is problematic because the true degrees of freedom for boosting models are unknown, and can only be assessed in simulation settings where the true data-generating process is known. Even worse, the available AIC estimates for boosting models are known to underestimate the true degrees of freedom [20]. This problem results in a tendency of stopping the boosting algorithm too late and can therefore lead to substantial overfitting.

To overcome these limitations, we pose a new sequential stopping rule for boosting algorithms that combines a modified version of AIC-based stopping with resampling. With the new sequential stopping rule, the boosting algorithm is capable to tune itself automatically by first choosing a possible grid that is based on the AIC-type strategy [1] – thereby making use of the known fact that AIC-based stopping approaches tend to stop too late. Afterwards, the optimal \( m_{\text{stop}} \) is selected automatically on this grid by computing a cross-validated estimate of \( m_{\text{stop}} \) based on subsampling. The proposed sequential stopping rule is not only computationally more effective than standard approaches in many cases (because the number of boosting iterations to be carried out is drastically reduced) but also results in a fully data-driven stopping rule without further specifications to be made by the researcher. To the best of our knowledge, this is the first approach to sequentially stop boosting algorithms that works completely data driven and can lead to a fully automated optimal variable selection in regression models for potentially high-dimensional data.

The paper is organized as follows. Section 2 starts with a short description of AIC-based early stopping and discusses the problems arising from this approach. We then present the new “subsampling after AIC” stopping rule and shortly describe its computational aspects. Section 3 contains the results of a simulation study using high-dimensional data with a small number of informative predictor variables and a large number of non-informative covariates. We show that the new stopping rule is not only computationally efficient but also leads to improved prediction accuracy and results in sparser, better interpretable models (compared to AIC-based early stopping). In Section 4, we apply the new sequential stopping rule to construct prediction models for the development of distant metastases after surgery for stage II colon cancer patients (using the above-described data by Barrier et al. [2]). In this case, the sequential stopping rule clearly outperforms AIC-based early stopping strategies with respect to all performance measures. Section 5 summarizes the main findings of the paper and discusses their consequences for biomedical applications.
Box 1  Simplified overview of component-wise boosting algorithm

Initialization:
- Step 1: Initialize iteration counter \( m = 0 \) and the additive predictor \( \hat{y}^{[0]} = 0 \).
- Step 2: Specify a base-learner for every covariate.

Boosting:
- Step 3: Set \( m := m + 1 \)
- Step 4: Compute the negative gradient vector of the loss with respect to the predictor and evaluate at \( \hat{y}^{[m-1]} \).
- Step 5: Fit the covariates separately to the negative gradient vector using their corresponding base-leaners.
- Step 6: Select the best fitting base-learner and add a proportion of its fitted values to \( \hat{y}^{[m]} \).

Iteration:
- Step 7: Iterate Step 3 to Step 6 until \( m = m_{\text{stop}} \).

2. Methods

2.1 AIC-based Stopping Procedures

The stopping iteration \( m_{\text{stop}} \) is the main tuning parameter of boosting algorithms – its selection reflects the common bias-variance trade-off in statistical modelling: Larger values for \( m_{\text{stop}} \) lead to a bigger model with a small bias but large variance whereas small values for \( m_{\text{stop}} \) lead to sparse models with a smaller variance but also to a larger bias with respect to the effect estimates of the covariates. A schematic overview of a component-wise gradient boosting algorithm is presented in Box 1. For a more technical description we refer to [6].

The selection of \( m_{\text{stop}} \) is often based on information criteria such as AIC [19], which is defined as

\[
\text{AIC}(m) = -2 \cdot \log(L_m) + 2 \cdot df_m.
\]

Here, \( L_m \) is the likelihood of the statistical model and \( df_m \) are the degrees of freedom, where the index \( m \) indicates that both values are derived from the model at boosting iteration \( m \). If AIC-based stopping is used, the optimal stopping iteration \( m_{\text{stop}} \) is the one minimizing AIC\((m)\). The negative log-likelihood of the model decreases with each boosting iteration (because the fit improves and the likelihood increases) whereas \( df_m \) (which also increase with \( m \)) penalize too complex models. Thus, overfitting is prevented by the fact that the combination of \( L_m \) and \( df_m \) results in a trade-off between fitting the data too closely and underestimating the complexity of the model.

The recently proposed method by Chang et al. [1] to find the optimal value of \( m_{\text{stop}} \) is based on an AIC change-point detection algorithm that avoids the common practice of first carrying out an a priori selected large amount of boosting iterations and performing a search of the optimal value of \( m_{\text{stop}} \) within the pre-selected range of iterations afterwards. A schematic overview of this method is presented in Box 2. For additional details we refer to [1].

Chang et al. state that their change-point detection method can lead to substantial computational savings as it reduces the amount of boosting iterations that need to be computed by stopping the algorithm already during the fitting process. However, they still expect the user to have prior information on the approximate values of the optimal \( m_{\text{stop}} \), as they are searching segments of iterations which are ultimately limited by a user-specified final mstop.

The main problem of the Chang et al.’s sequential method to detect the optimal \( m_{\text{stop}} \), however, is its sole dependency on information criteria such as the AIC. These criteria are known to be unstable with respect to variable selection [21] and rely on an estimation approach of the degrees of freedom that underestimates the true degrees of freedom for component-wise boosting algorithms [20]. For boosting with the L2 loss, for example, the degrees of freedom at iteration \( m \) are estimated by the trace of the approximate boosting hat matrix \( B_m \) [6]:

\[
df_m = \text{trace}(B_m).
\]

The trace of \( B_m \) can be computed in an iterative way during the fitting process. The estimation formula, which can be found in [6] (p 494), depends on the set of covariates that were selected until iteration \( m \). However, component-wise boosting algorithms update only the best fitting component in each iteration step. This searching process is neglected in the estimation of degrees of freedom by the trace of the hat matrix \( B_m \). This leads to the underestimation of the true \( df_m \) (Fig. 1) and results in a tendency of stopping the boosting algorithm too late. The optimal \( m_{\text{stop}} \) is therefore overestimated – this problem will be further illustrated in Section 3.

2.2. Subsampling after AIC

We now introduce a sequential stopping rule for boosting algorithms that extends and improves the approach of Chang et al. [1]. The AIC change point detection method is modified, so that no a priori information on the approximate value of \( m_{\text{stop}} \) is needed. As in [1] (Box 2), we specify a segment width which is set to be 40 boosting iterations. Instead of approximating the slopes at both ends of the segments as in [1] (which depends on additional iterations and on a final stopping
iteration), we simply check after every 40 iterations whether the current optimal stopping iteration minimizing the AIC (denoted as \( m_{AIC} \) in the remainder of the paper) is located in the border region of the current segment. If \( m_{AIC} \) is too close to the current iteration \( m \), we carry out another 40 boosting steps. This procedure is repeated until \( m_{AIC} \) is at least 10% smaller than \( m \).

As AIC-based methods have a tendency of stopping too late, we will additionally incorporate subsampling to select the optimal \( m_{stop} \) based on the predictive risk of the model. Subsampling is a resampling technique that randomly samples a pre-defined fraction of observations from the observed data. In comparison to the more common bootstrap procedure, subsampling is carried out without replacement – thereby avoiding a possible bias towards overly complex models in high-dimensional data situations [22, 23]. In our case, the resulting sub-samples have half the size of the original (full) data set and are used as training data. The observations that are not included in the respective sub-sample are regarded as test data set. This strategy can be used to select \( m_{stop} \) by carrying out boosting on each training data set and selecting the value of \( m_{stop} \), that minimizes the aggregated predictive risk on the test data set. Resampling techniques, which are a very general approach to evaluate the prediction accuracy of statistical models, have the advantage that they do not rely on potentially biased estimates of \( df_m \). However, they can become computationally expensive especially for high-dimensional data sets with a large amount of candidate variables. It is therefore of special interest to avoid the computation of an unnecessary high number of boosting iterations by limiting the size of the grid to be searched. Note that the computation of the AIC criterion can become inefficient, especially if the number of observations in a data set is large and the boosting hat matrix \( B_m \) is thus of high dimensionality.

The proposed combination of the sequential strategy by Chang et al. and subsampling is presented in Box 3: After carrying out the search for \( m_{AIC} \), we use 25-fold subsampling (with sampling probability of 0.5) to select the optimal \( m_{stop} \) on the grid 1, \( m_{AIC} \), minimizing the predictive risk on the test samples.

### 2.3 Computational Demand and Implementation

The computational demand of the proposed strategy usually depends on the characteristics of the data to be analyzed. For resampling techniques such as subsampling in combination with boosting the demand depends on the number of samples \( B \) to be drawn and the running-time increases with the number of candidate variables \( p \). The computation of the AIC is demanding in situations with many observations, as the \((n \times n)\) hat matrices get big. The combination of both is of course slower than the single usage of the AIC and

**Box 2** Simplified overview of AIC-based change point detection [1]

![Fig. 1](image_url)

**Fig. 1** Comparison of the true degrees of freedom [9] to their corresponding estimates based on the trace of the hat matrix. The solid line represents the mean estimated \( df_m \) obtained from 100 simulation runs when considering the classical scenario with \( p = 100 \) (see Section 3).
Search for $m_{AIC}$:
Step 1: Initialize $m = 40$
Step 2: Run boosting algorithm until iteration $m$
Step 3: Stop if $m_{AIC} \leq 0.9 \cdot m$
Else set $m := m + 40$ and iterate Steps 2 and 3

Subsampling:
Step 4: Search for $m_{stop}$ on the grid 1, ..., $m_{AIC}$ by 25-fold subsampling with sampling probability 0.5.

Subset the model:
Step 5: Set the model to iteration $m_{stop}$.

Box 3  Schematic overview of the subsampling after AIC algorithm

could in some specific cases even be slower than the traditional search for $m_{stop}$ on a larger grid. This could happen if $n$ is very large or the optimal $m_{stop}$ is close to the upper border of the initial grid. However, computing-time can be reduced by applying parallel computing or by decreasing the number of samples, e.g. to $B = 10$ like in classic cross-validation.

All analyses presented in this paper were carried out using the statistical computing environment R [24]. For model-based boosting we used the add-on package mboost. The implementation of the subsampling after AIC stopping rule is straightforward due to the update function of the mboost package, which allows changing the number of boosting-iterations by simple indexing [25]. The AIC of the models was computed using the function AIC() while subsampling was carried out using the cv() and cvrisk() functions with the argument type="subsampling". The cvrisk() function has built-in solutions for parallel computing (using the package multicore), which makes the resampling step computationally more efficient. For all computations we used the default-settings of the mboost package (except where indicated). Note that the “subsampling after AIC” stopping rule can be easily modified; instead of subsampling one can use any other cross-validation scheme (e.g. bootstrapping) by simply changing the options in cv(). Furthermore, the AIC can be easily replaced by other information criteria (e.g., BIC). Currently, the function AIC() is only available for the two most common loss-functions, which lead to Gaussian and binomial regression.

In order to guarantee the reproducibility of our results, all R-CODE that led to the results presented in this paper is available as online supplementary material.

3. Simulation Study

The aims of our simulation study were i) to evaluate the potential problems with AIC based early stopping, ii) to assess the performance and the running-time of the proposed sequential stopping rule and iii) to investigate its impact on variable selection.

All simulations were based on a linear model with the outcome variable $Y$, three informative variables and a normally distributed error term (cf. [1]). Estimation was based on a set of $p$ covariates (including the three informative variables of the linear model and the intercept). More formally, the model is given by

$$Y = 1 + 5 \cdot X_1 + 2 \cdot X_2 + X_3 + \epsilon,$$

where $X$ denotes the covariate matrix and $I$ is the identity matrix. Hence the components of $X$ are independent and all have variance 1. The sample size was set as $n = 100$ and the dimensionality of $X$ was chosen to be $p = 100$ (classical scenario) and $p = 500$ (high-dimensional scenario). As baselearners we used simple linear models.

In a first step, we compared the estimated $df_{m_{stop}}$ obtained from the AIC-based methods to the true $df_{m_{stop}}$ [9, 20]. The results, which are shown in Figure 1, suggest that the true $df$ are clearly underestimated. Specifically, Figure 1 underlines the problems of selecting $m_{stop}$ purely based on information criteria such as the AIC.

In a second step, we analyzed the performance of the subsampling after AIC stopping rule by comparing the resulting $m_{stop}$ values to the ones that were based only on AIC. Furthermore, we evaluated the resulting prediction accuracy on an additional test data set of size $n = 1000$. The results obtained from 200 simulation runs for both scenarios are displayed in Figure 2. The new sequential stopping rule resulted in considerably smaller values of $m_{stop}$ for both the classical and the high-dimensional scenarios. Also, it clearly outperformed the AIC approach when it comes to prediction accuracy. This result further suggests that the stopping iterations obtained from the AIC approach are too large, leading to a substantial loss in the prediction accuracy. As seen from Figure 1, this problem is due to the overestimation of the degrees of freedom $df_{m_{stop}}$. This effect was even more pronounced in the high-dimensional scenario than in the classical scenario.

In addition to the subsampling after AIC method, we also considered a pure subsampling approach searching a much larger grid without a sequential strategy. This approach can be seen as the gold standard in the case that the grid is chosen large enough to extend (far) beyond the “true” optimal $m_{stop}$. As seen from Figure 2, the pure subsampling approach did not lead to an improved accuracy compared to the sequential approach. The usage of $m_{AIC}$ to determine a smaller grid to be searched by subsampling therefore yields the same results as searching a much larger grid—without depending on an arbitrary specification of a final $m_{stop}$. We also compared these results to other resampling techniques like cross-validation or bootstrapping which did not lead to further improvements (results not presented here).
Fig. 2  The boxplots show the results of the simulation study for the classical scenario (left) and the high-dimensional scenario (right). The plots in the first row display the estimated optimal stopping iterations; the second row shows boxplots for the predictive risk; the boxplots in the third row display the actual running times for the different stopping strategies. Gray lines correspond to the same simulation runs (i.e., to the same training and test data).
In the third row of Figure 2 we present the actual running times of the different stopping strategies using a standard laptop (Dual Core 2 × 2.13 GHz and 3 GB RAM) and applying parallel computing. In the classical scenario, subsampling after AIC led to computational savings compared to the traditional search of a large grid. However, in the high-dimensional setting it gets clear that these savings cannot be guaranteed: in some simulation runs where \( m_{\text{stop}} \) is close to the border of the larger grid, the traditional method may be faster than the sequential approach.

In a third step, we compared the different stopping rules with respect to selection of the informative predictor variables. The subsampling after AIC method led to much sparser models, resulting in a mean selection rate of 3.5% for the non-informative variables in the high-dimensional scenario (14.1% in the classical scenario) compared to 37.9% (classical scenario) and 42.6% (high-dimensional scenario) for the purely AIC-based stopping. All informative variables were selected by both methods in all simulation runs.

As an alternative to boosting with early stopping, we additionally considered the Lasso as competing penalized regression tool. We found very similar results to those obtained by boosting (stopped with our sequential strategy, results not presented here). This confirms the strong association of both approaches highlighted in the literature [9, 20].

4. mRNA-based Prediction of the Development of Metachronous Metastases for Stage II Colon Cancer Patients

To evaluate the performance of the subsampling after AIC stopping rule in real data settings, we constructed a prediction rule for stage II colon cancer prognosis based on tumor gene expression data (see [2] for details). The aim of the original study was to develop an accurate prognosis at the time of surgery with respect to the future development of metachronous metastases based on mRNA of the removed tumor. This could help to identify patients where postoperative adjuvant chemotherapy might be beneficial.

The sample consists of 50 patients who underwent surgery due to stage II colon cancer and did not receive postoperative adjuvant chemotherapy. Twenty-five patients developed metachronous metastases in the following five years after surgery (affecting the liver in 22 cases and the lung in five cases), whereas 25 remained disease free for at least 60 months. During the surgery, tumor samples were collected that contained at least 80% tumor cells. The samples were then profiled using the Affymetrix HGU133A GeneChip. The associated gene expression measures for the 22,283 genes were processed by the authors of the original publication [2] using the R add-on package rma [26].

The original analysis was following a two-step approach, which first identified the 30 most differentially expressed genes and later applied linear discriminant analysis to separate the group of patients who developed metachronous metastases from the disease-free ones, based on the pre-selected 30 candidate genes only. Unlike [2], we considered all 22,283 genes simultaneously and used boosting to fit a binomial logistic regression model for the prediction of the future development of metachronous metastases. We took advantage of the ability of boosting to identify the most influential predictors in high-dimensional regression settings, leaving it up to the boosting algorithm to select a sparse and accurate prediction model.

Binary regression for microarray analysis was proposed by Schimek [27] who applied penalized logit and Bayesian probit regression for a study on breast cancer [28]. Dettling and Bühlmann [29], who analyzed the same data set as [28], reported much lower error rates applying boosting based on decision trees. However they reduced the dimensionality of the data by only using a pre-selected subset of genes. An evaluation of boosting methods for classification can be found in [16]. The authors applied AdaBoost and gradient boosting with regression tree base-learners. They stopped the algorithms at a maximum \( m_{\text{stop}} \) of 400 and compared them to classic, non-boosting classifiers in low-dimensional settings. The boosting methods showed a slightly improved accuracy but failed to clearly outperform classic logistic regression. The authors concluded that in medical applications standard logistic regression stays the method of choice (due to the easier interpretation) but that an optimized \( m_{\text{stop}} \) might lead to further improvements of the more sophisticated boosting methods [16]. However, results of boosting with linear or non-linear regression models as base-learners are equally easy to interpret as classical approaches while the latter are not working for high-dimensional data with \( p \gg n \).

As in our setting with 22,283 candidate genes classical logistic regression was not feasible, we applied component-wise gradient boosting [6] with the negative binomial log likelihood loss and simple linear regression models as base-learners. This strategy leads essentially to statistical prediction rules that have the same interpretation as those resulting from standard logistic regression. To optimize the stopping iteration of the algorithm we applied the subsampling after AIC sequential stopping rule and compared it to AIC-based stopping. Prediction accuracy was evaluated by carrying out leave-one-out (LOO) cross-validation [30]: Each patient was left out once and the boosting classification rule was learned using the reduced data set. In a next step, the resulting 50 rules were evaluated by comparing the true outcome status of the left out observations to their respective tumor mRNA-based predictions. As a result we were able to compare the accuracy of the resulting classifiers from sequential stopping and AIC stopping on all 50 patients of the original sample.

The stopping iterations obtained from the new sequential stopping rule ranged from 7 to 22 iterations on the LOO-samples (median = 11.5) whereas the usage of a pure AIC approach led to much larger \( m_{\text{stop}} \) ranging from 203 to 289 iterations (median = 254.5). As a result of the much earlier stopping, the corresponding models from sequential stopping were sparser, containing only seven genes on average (median, range: 4 – 14) compared to 35 (range: 29 to 41) for purely AIC-based stopping. This result is in concordance with the simulation results obtained in Section 3.
further highlights the ability of boosting to lead to sparse models and to identify the most influential predictors if the stopping iteration is chosen correctly.

Although relying on a much smaller number of genes (or actually partly because of it) the sequential stopping strategy clearly outperformed the AIC-based approach with respect to prediction accuracy: 84% of the patients were classified correctly compared to only 68% for the AIC approach. This implies that our novel “subsampling after AIC” approach predicted in 84% of the cases the development or non-development of metachronous metastases within 60 months after surgery correctly. These results were obtained by relying only on the mRNA of the tumor samples. The increased accuracy is reflected also by the other performance measures such as sensitivity (96% vs. 88%), specificity (72% vs. 48%) and the area under the receiver-operating characteristics curve (AUC; 0.867 vs. 0.750, see also Fig. 3). As an alternative to boosting, we again considered the Lasso as competing penalized regression tool and found comparable results (AUC = 0.819, see Fig. 3).

These results highlight the need of an optimized stopping criterion for boosting: Both AIC-based stopping and the sequential approach incorporated a sophisticated model-based boosting algorithm, but only with an accurate selection of m_stop this procedure can really play out its strength.

5. Conclusion

Model-based boosting algorithms have evolved into one of the most promising tools in modern biomedical research, as they combine a sophisticated ensemble prediction method developed in the machine learning context with classical regression modelling. Specifically, the application of component-wise gradient boosting to biomedical regression problems overcomes the problems that standard fitting algorithms are usually faced with in the presence of high-dimensional data (p > n). As demonstrated in Section 4 of the paper, this issue is especially important for the development of prediction rules that are based on molecular markers. In this context, boosting incorporates intrinsic variable selection and shrinkage of effect estimates, leading to both sparse and biologically interpretable models that help to identify the most influential covariates while resulting in more accurate predictions than standard approaches.

As demonstrated in the present work, the correct selection of the stopping iteration m_stop is crucial for fully exploiting the benefits of boosting algorithms. Choosing an appropriate value of m_stop is particularly important because the stopping iteration is the main tuning parameter for controlling the bias-variance trade-off in the resulting statistical model fit. In absence of an automated stopping rule, researchers have to carry out a large number of boosting iterations before evaluating the resulting models and subset them to an optimized smaller value of m_stop. This “traditional” procedure can be improved by stopping the algorithm already in the initial fitting process. Furthermore, conventional strategies for determining the stopping iteration depend on the pre-selection of a maximum number of iterations that has to be specified according to the prior knowledge of the researcher. This approach is in contrast to the pure data-driven concept of boosting.

With the proposed subsampling after AIC stopping rule we overcome these limitations by sequentially stopping (and tuning) the boosting algorithm during the fitting process without further specifications made by the researcher. The subsampling after AIC stopping rule is therefore an important step towards developing effective automated stopping procedures for boosting algorithms – at least for the most common loss functions yielding Gaussian or binomial regression where estimates for the degrees of freedom of the resulting model are available. For other loss functions, or when the computation of the AIC becomes inefficient in large data sets, the proposed method could be easily adopted to a pure sequential resampling approach by applying subsampling in a block-wise fashion. This way, the initial search for m_AIC can be omitted. However, for very large data sets also sequential resampling methods can...
become potentially problematic due to the increased memory demand of boosting objects. Further research is warranted on this type of sequential stopping procedures for loss functions without existing AIC estimates.

As shown in this work, the new sequential approach clearly outperformed pure AIC-based stopping (and also AIC-based earlier sequential approaches [1]) in terms of prediction accuracy. The resulting models were also sparser and identified the most relevant predictors in a set of simulation studies (Section 3) as well as in the prediction of the development of metachronous metastases for stage II colon cancer patients by tumor mRNA (Section 4). Although these advantages are inherited from the boosting algorithm itself, they can only show their full strength if the algorithm is stopped early enough.

Acknowledgements
The work of authors AM and MS was supported by the Interdisciplinary Center for Clinical Research (IZKF) at the University Hospital of the Friedrich-Alexander-Universität Erlangen-Nürnberg (Project J11). The work of BH was supported by the DFG project HO 3242/1–3.

References