BMC Bioinformatics

BioMed Central

Methodology article

Open Access

Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models

Harald Binder*[1,2] and Martin Schumacher[2]

Address: [1]Freiburg Center for Data Analysis and Modeling, University of Freiburg, Eckerstr. 1, 79104 Freiburg, Germany and [2]Institute of Medical Biometry and Medical Informatics, University Medical Center Freiburg, Stefan-Meier-Str. 26, 79104 Freiburg, Germany

Email: Harald Binder* - binderh@fdm.uni-freiburg.de; Martin Schumacher - ms@imbi.uni-freiburg.de

* Corresponding author

Published: 10 January 2008

BMC Bioinformatics 2008, **9**:14 doi:10.1186/1471-2105-9-14

Received: 8 October 2007

Accepted: 10 January 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/14

Abstract

Background: When predictive survival models are built from high-dimensional data, there are often additional covariates, such as clinical scores, that by all means have to be included into the final model. While there are several techniques for the fitting of sparse high-dimensional survival models by penalized parameter estimation, none allows for explicit consideration of such mandatory covariates.

Results: We introduce a new boosting algorithm for censored time-to-event data that shares the favorable properties of existing approaches, i.e., it results in sparse models with good prediction performance, but uses an offset-based update mechanism. The latter allows for tailored penalization of the covariates under consideration. Specifically, unpenalized mandatory covariates can be introduced. Microarray survival data from patients with diffuse large B-cell lymphoma, in combination with the recent, bootstrap-based prediction error curve technique, is used to illustrate the advantages of the new procedure.

Conclusion: It is demonstrated that it can be highly beneficial in terms of prediction performance to use an estimation procedure that incorporates mandatory covariates into high-dimensional survival models. The new approach also allows to answer the question whether improved predictions are obtained by including microarray features in addition to classical clinical criteria.

Background

For models built from high-dimensional data, e.g. arising from microarray technology, often survival time is the response of interest. What is wanted then, is a risk prediction model that predicts individual survival probabilities based on the covariates available. Because of the typically large number of covariates, techniques have been developed that result in sparse models, i.e., models where only a small number of covariates is used. In modern approaches, such as boosting [1] and the Lasso-like path algorithms [2], it is avoided to discard covariates before model fitting, and parameter estimation and selection of covariates is performed simultaneously. This is implemented by (explicitly or implicitly) putting a penalty on the model parameters for estimation. The structure of this penalty is chosen such that most of the estimated parameters will be equal to zero, i.e., the value of the corresponding covariates does not influence predictions obtained from the fitted model.

Often there are clinical covariates, such as a prognostic index, available in addition to microarray features. The

former could be incorporated into the model just like an additional microarray feature, but due to the large number of microarray features compared to the typically small number of clinical covariates there is the danger, that the clinical covariates might be dominated, even when they carry important information. Therefore mandatory inclusion for such covariates is needed. When it is also of interest whether use of microarray features can improve over models based solely on the clinical covariates, i.e., the latter are not only included for increasing prediction performance, the parameters of the clinical covariates have to be estimated unpenalized. Only then the resulting model can be fully compared to models based only on clinical covariates, where typically unpenalized estimates are used.

To our knowledge, existing techniques for estimating sparse high-dimensional survival models do not naturally allow for unpenalized mandatory covariates. In contrast, for the generalized linear model class there is a recent approach that fits this need [3]. We therefore extend this one to survival models. As will be shown, this new approach is closely related to the existing high-dimensional survival modeling techniques when no mandatory covariates are present. Therefore, we first review some of the latter, before developing the extension.

Given observations (t_i, d_i, x_i), $i = 1, ..., n$, where t_i is the observed time to the event of interest for individual i, δ_i takes the value 1 if an event occurred at that time and 0 if the observation has been censored, and $x_i = (x_{i1}, ..., x_{ip})'$ is a vector of covariates obtained at time zero, many approaches for high-dimensional survival data are based on the Cox proportional hazards model for the hazard

$$\lambda(t|x_i) = \lambda_0(t)\exp(F(x_i;\beta)), \quad (1)$$

where $\lambda_0(t)$ is the baseline hazard and $F(x; \beta)$ is a function of the covariates, depending on a parameter vector β. When a linear predictor of the form $F(x; \beta) = x'\beta$ is used, each element of the parameter vector $\beta = (\beta_1, ..., \beta_p)'$ specifies the influence of a single covariate. For estimation, the baseline hazard $\lambda_0(t)$ is left unspecified and an estimate $\hat{\beta}$ is obtained by maximizing the partial log-likelihood

$$l(\beta) = \sum_{i=1}^{n} \delta_i \left(F(x_i;\beta) - \log\left(\sum_{j=1}^{n} I(t_j \le t_i)\exp(F(x_j;\beta)) \right) \right), \quad (2)$$

where $I()$ is an indicator function taking value 1 if its argument is true, i.e., if individual j is still under risk just before time t_i, and value 0 otherwise.

When the number of covariates is large, maximization of (2) can no longer be carried out by standard techniques. In Lasso-like approaches (using a linear predictor) [2,4] a penalty term $\lambda\sum_j|\beta_j|$ is added to the partial log-likelihood (2). The resulting penalized partial log-likelihood then is maximized by quadratic programming techniques or by the more efficient path algorithms [2]. The penalty parameter λ can be determined e.g. by cross-validation. Due to penalizing the absolute value, many elements of the resulting estimate $\hat{\beta}$ will be equal to zero, i.e., the solution will be sparse, larger values of λ leading to more sparseness. Lasso-like approaches have in addition been developed for additive risk models [5] and accelerated failure time models [6].

An alternative approach for fitting of sparse high-dimensional models is provided by gradient boosting techniques [1,7]. The underlying principle is that of stepwise optimization of a function $F(x; \beta)$ in function space by minimizing a loss function. For fitting a Cox model, the negative partial log-likelihood is used as a loss function [8]. In each step $k = 1, ..., M$ the negative gradient of the loss function, evaluated for the current estimate $F_{k-1}(x; \hat{\beta}_{k-1})$ at the observations, is fitted e.g. by standard least squares techniques. The resulting fit $f_k(x; \hat{\gamma}_k)$, which depends on some parameter vector γ_k, then is used to updated the overall fit via $F_k(x; \hat{\beta}_k) = F_{k-1}(x; \hat{\beta}_{k-1}) + \varepsilon \hat{f}_k(x; \hat{\gamma}_k)$, where ε is some small positive value.

In componentwise boosting a linear predictor of the form $F_k(x; \hat{\beta}_k) = x'\hat{\beta}_k$ is used and only one element of $\hat{\beta}_k$ is updated in each boosting step [9]. The parameter to be updated in step k is determined by evaluating fits to the gradient $\hat{f}_{kj}(x_i; \hat{\gamma}_j) = \hat{\gamma}_j x_{ij}$, $j = 1, ..., p$, where $\hat{\gamma}_j$ is determined by least-squares, and selecting that one that improves the overall fit the most. This results in sparse fits similar to Lasso-like approaches, with many of the estimated coefficients being equal to zero.

For linear models with squared-error loss function, gradient boosting is equivalent to iterative fitting of residuals. This idea has been adapted to the generalized linear model setting as an alternative to the gradient approach [3]. In each boosting step, estimation is performed by a standard Newton-Raphson step, based on a penalized likelihood, where previous boosting steps are incorporated as an offset. An advantage of this offset-based boost-

ing approach is that it allows for very flexible penalty structure, including unpenalized mandatory covariates. Adapting it for survival models would help to resolve the highlighted issues arising when clinical covariates should be included in high-dimensional survival models.

One could also try to adapt existing gradient boosting techniques to allow for unrestricted mandatory components, but we think the offset-based approach is a more natural starting point. Alternatively, approaches such as the grouped Lasso [10,11], which allow for groups of covariates with varying penalization, could potentially be adapted by introducing groups with no penalization. As this has not yet been considered by their authors, and also the group Lasso approach for the Cox model [12] no longer uses simultaneous estimation of all parameters, we do not follow this route here.

In the following will therefore adapt the offset-based boosting approach from [3] for estimating Cox proportional hazards models. The resulting advantage of allowing for unpenalized mandatory components for clinical covariates will be illustrated with data from patients with diffuse large B-cell lymphoma.

Results and discussion

Algorithm

The aim of the new *CoxBoost* approach is to estimate the parameter vector β for a linear predictor $F(x; \beta) = x'\beta$ in the Cox proportional hazards model (1). Typical gradient boosting approaches either use all covariates for the fitting of the gradient in each step, e.g. based on regression trees, or, in componentwise boosting, update only one element of the estimate of β, corresponding to only one covariate. The flexibility of the offset-based approach in [3] partly is due to considering a flexible set of candidate sets, i.e., a set of sets of covariates, for updating in a specific boosting step. This is adapted for the CoxBoost approach. In boosting step $k = 1, ..., M$ there are q_k predetermined candidates sets of covariates with indices $\mathcal{I}_{kl} \subseteq \{1, ..., p\}$, $l = 1, ..., q_k$. For each of these q_k sets a simultaneous update of the parameters for the corresponding covariates is evaluated. The candidate set that improves the overall fit the most will then be selected for the update.

With $\hat{\beta}_{k-1} = (\hat{\beta}_{k1}, ..., \hat{\beta}_{kp})'$ being the actual estimate of the overall parameter vector β after step k - 1 of the algorithm, and $\hat{\eta}_{i,k-1} = x_i'\hat{\beta}_{k-1}$ being the corresponding linear predictors, potential updates for the elements of $\hat{\beta}_{k-1}$ corresponding to $\mathcal{I}_{kl}$ are obtained by maximizing the penalized partial log-likelihood

$$l_{pen}(\gamma_{kl}) = \sum_{i=1}^{n} \delta_i \left(\eta_{i,k-1} + x'_{i,\mathcal{I}_{kl}}\gamma_{kl} - \log\left(\sum_{j=1}^{n} I(t_j \le t_i) \exp(\eta_{i,k-1} + x'_{i,\mathcal{I}_{kl}}\gamma_{kl}) \right) \right) - \lambda\gamma'_{kl}P_{kl}\gamma_{kl} \tag{3}$$

with respect to the parameter vector γ_{kl} of size $|\mathcal{I}_{kl}|$, where $x_{i,\mathcal{I}_{kl}}$ is the covariate vector for subject i containing only those covariates with indices in $\mathcal{I}_{kl}$. The penalty parameter λ which has to be selected, results in a cautious update, if it is large enough. The penalty matrices P_{kl} can be specified separately for each boosting step and each candidate set, which provides considerable flexibility of the CoxBoost approach. Typically these will be diagonal matrices, for penalizing each covariate separately, but by varying the size of the diagonal elements, differential penalization is introduced. In contrast, for gradient boosting approaches the fitting in each step is performed unpenalized and only afterwards the update is multiplied by a small shrinkage factor ε, thus applying equal penalization to all covariates. For the present application of the CoxBoost approach we will use only diagonal elements 1 and 0, for "penalization" and "no penalization".

The parameter estimates $\hat{\gamma}_{kl}$ for evaluating the candidate sets are obtained by penalized partial likelihood techniques [13]. Using the starting value $\hat{\gamma}_{kl_0} = 0$, the first Newton-Raphson step is

$$\hat{\gamma}_{kl} = I_{pen}^{-1}(\hat{\gamma}_{kl_0})U(\hat{\gamma}_{kl_0}), \tag{4}$$

where $U(\gamma) = (\partial l/\partial\gamma)(\gamma)$ is the score function and $I_{pen}(\gamma) = (\partial^2 l/\partial\gamma\partial\gamma')(\gamma) + \lambda P_{kl}$ is the information matrix, obtained from the first and second derivatives of the unpenalized partial log-likelihood $l(\gamma_{kl})$, i.e., (3) without the penalty term. As further updates can take place in later boosting steps, only one Newton-Raphson step is performed.

Given the sets of sets of indices $\mathcal{I}_k = \{\mathcal{I}_{k1}, ..., \mathcal{I}_{q_k}\}$, corresponding penalty matrices P_{kl}, $k = 1, ..., M$, and the penalty parameter λ, the general CoxBoost algorithm is as following:

1. Initialize $\hat{\eta}_{i,0} = 0$, $i = 1, ..., n$, and $\hat{\beta}_0 = (0, ..., 0)'$.

2. Repeat for $k = 1, ..., M$

(a) Obtain potential updates $\hat{\gamma}_{kl}$ for the candidate sets $\mathcal{I}_{kl}$, $l = 1, ..., q_k$, via (4).

(b) Determine the best update l^* which maximizes the penalized partial log-likelihood (3).

(c) Obtain the updated parameter vector $\hat{\beta}_k$ vector via

$$\hat{\beta}_{kj} = \begin{cases} \hat{\beta}_{k-1,j} + \hat{\gamma}_{kl(j)} & j \in \mathcal{I}_{kl^*} \\ \hat{\beta}_{k-1,j} & j \notin \mathcal{I}_{kl^*} \end{cases} \quad j = 1, ..., p,$$

where $\hat{\gamma}_{kl(j)}$ is that element of $\hat{\gamma}_{kl}$ that corresponds to $\hat{\beta}_{k,j}$, and update $\hat{\eta}_{k,i} = x_i' \hat{\beta}_k$, $i = 1, ..., n$.

Note that the step size for the updates in part 2c) of the algorithm is 1. This is in contrast to gradient boosting algorithms, where the fits $\hat{f}_k(x, \hat{\gamma}_k)$ to the gradient are multiplied by some small positive value ε before updating. In the CoxBoost algorithm the role of ε is taken by the penalty parameter λ during estimation. In the following, for unpenalized mandatory components the corresponding elements of the penalty matrix P_{kl} are taken to be zero, resulting in fast building up of coefficient estimates.

Componentwise CoxBoost with mandatory covariates

Componentwise CoxBoost, similar to componentwise ridge boosting [3], is obtained when in each boosting step only one element of the overall parameter vector is updated, i.e., $\mathcal{I}_k = \{\{1\}, ..., \{p\}\}$, $k = 1, ..., M$. In this setup CoxBoost is very similar to the idea of stagewise regression described in [14]. Based on the results given there and in [3] we expected the resulting coefficient paths, i.e., the estimated parameters in the course of the boosting steps, to be very similar to Lasso-like approaches. For strong correlations between covariates, again due to its similarity to stagewise regression, it is expected that the coefficient paths of componentwise CoxBoost are even more stable, i.e., more monotone, than that of Lasso-like approaches [15].

There are two approaches for incorporating mandatory covariates into the CoxBoost algorithm. Given the indices of the mandatory covariates $\mathcal{I}_{mand}$, the indices from componentwise CoxBoost can be augmented via $\mathcal{I}_k = \{\{1\} \cup \mathcal{I}_{mand}, ..., \{p\} \cup \mathcal{I}_{mand}\}$, omitting components $\{j\} \cup \mathcal{I}_{mand}$ where $j \in \mathcal{I}_{mand}$. This allows for simultaneous estimation of the parameters of mandatory and optional covariates. When the diagonal elements of the penalty matrices P_{kl} corresponding to $\mathcal{I}_{mand}$ are set to zero, while the others still have a value larger than zero, this furthermore leads to unpenalized estimation of the parameters of the mandatory covariates. When one wants to evaluate whether the optional covariates provide additional predictive power compared to the mandatory covariates, this is the appropriate penalty structure. Alternatively, mandatory covariates can be introduced by updating their parameters before each step of componentwise CoxBoost. This corresponds to $\mathcal{I}_{2k-1} = \{\mathcal{I}_{mand}\}$, $\mathcal{I}_{2k} = \{\{1\}, ..., \{p\}\}$ (omitting components $\{j\}$ where $j \in \mathcal{I}_{mand}$), $k = 1, ..., M$. Again, for evaluating the additional predictive performance obtained from the optional covariates we suggest to use penalty equal to zero for the mandatory covariates.

Implementation

There are several implementation decisions to be made for the CoxBoost algorithm. At the lowest level, a criterion for selecting the best update l^* in each step has to be chosen. Ideally, the penalized partial log-likelihood (3) or some variant of it that incorporates model complexity (such as AIC) would be used. While for a small number of covariates, say $p < 100$, this is computationally unproblematic, for large p it is no longer feasible to evaluate this criterion for each candidate set in each step. As an approximation, we therefore propose to employ a penalized version of the score statistic

$$U(\gamma)' I_{pen}^{-1}(\gamma) U(\gamma)$$

evaluated at $\hat{\gamma}_{kl_0}$. This is based on a low-order Taylor expansion of the penalized partial log-likelihood (3) and requires no extra computation. In our experiments, selecting boosting step updates by the largest value of this score statistic was very close to selecting by the penalized partial log-likelihood itself, but considerably reduced computation time.

For including mandatory covariates, computational considerations led us to use the CoxBoost variant with separate updating of the mandatory parameters. This avoids frequent inversion of $I_{pen}(\gamma)$, because in the componentwise updating step of this variant for the optional covariates this reduces to a simple division. The CoxBoost algorithm has two tuning parameters, the penalty parameter λ and number of boosting steps M. While selection of the latter is critical to avoid overfitting, the penalty parameter is of minor importance, as long as it is large enough.

We therefore suggest to select only the number of boosting steps by a procedure such as cross-validation. The penalty parameter λ is selected only very coarsely such that the corresponding selected number of boosting steps M is larger than 50. This approach was seen to work well for offset-based boosting for generalized linear models [3].

The algorithm has been implemented in the statistical environment R [16] in the package "CoxBoost", which is available from the authors.

Example

We illustrate the CoxBoost algorithm with the diffuse large B-cell lymphoma data from the study in [17]. A review of attempts to build predictive survival models from such data is found in [18]. There is a potentially censored survival time response for 240 patients with a median follow up of 2.8 years, where 57% of the patients died during that time. For prediction there are 7399 microarray features available. In addition, the International Prognostic Index (IPI), a well-established prognostic score derived from five clinical covariates [19], is available for 222 patients. As we want to investigate whether the microarray features increase predictive performance compared to a purely clinical model based on the IPI, analyses are restricted to this smaller set of patients. Missing values for the microarray features were imputed as described in [20].

In [17] the data is split into a training set where the parameters are estimated, and a test set where prediction performance is evaluated. The disadvantage of this is that not all data is available for model building and parameter estimation. We employ an alternative approach [20], based on bootstrap samples, which allows to use all observations for model fitting, but nevertheless results in accurate prediction error estimates. For evaluation of prediction performance the Brier score is used, i.e., the (expected) squared difference between predicted survival probability at a time t and the true state (1 for being still under risk, and 0 if an event occurred). This can be be plotted as a function of time, resulting in prediction error curves. For estimation of the latter, prediction error estimates obtained from single bootstrap samples are aggregated into a .632+ estimate. An additional summary measure is obtained when for every single bootstrap sample a .632+ prediction error curve is calculated and integrated (in our case up to time 10). See the Methods section for more details.

As a conservative reference for performance comparison the Kaplan-Meier prediction is used, a non-parametric estimate of the survival probability over time. That way it can be checked whether procedures potentially perform worse than a prediction that does not use any covariate information at all. The performance of componentwise CoxBoost is furthermore compared to that of gradient boosting for the Cox model [1] (R package "mboost" [21]) and that of CoxPath, a Lasso-like path algorithm for fitting the Cox model [2] (R package "glmpath" [22]). For fitting models with these procedures only the microarray features (i.e., not the IPI) are used. In addition, componentwise CoxBoost with the IPI as an additional optional and as an unpenalized mandatory covariate is compared to a simple Cox model that has the IPI as its only covariate. The tuning parameters, i.e., the number of boosting steps and the number of path algorithm steps, are chosen by 5-fold cross-validation with respect to the partial log-likelihood. All other settings are at the default values of the respective implementations.

Before looking at prediction performance, we investigate the influence of unpenalized mandatory covariates on the coefficient paths, i.e., the parameter estimates for the individual covariates plotted against the norm of the parameter vector (which increases in the course of the CoxPath/boosting steps). Figure 1 shows the coefficient paths for CoxPath, gradient boosting, componentwise CoxBoost, and componentwise CoxBoost with the IPI as a mandatory covariate (where the parameter estimates for the IPI are not shown). The estimates corresponding to the number of CoxPath steps and the number of boosting steps selected by cross-validation are indicated by vertical lines. Covariates that receive non-zero parameter estimates by all four approaches in that cross-validation solutions are indicated by solid curves, the others by dashed curves. For the former, and other microarray features with corresponding parameter estimates that are large in absolute value, the UNIQIDs are given in the right margins of the plots.

It is seen that the coefficients paths for componentwise CoxBoost, gradient boosting and CoxPath are very similar. For the latter they are a bit more unstable, in the sense that they are not monotone, which is to be expected based on the results in [15]. Nevertheless, the six microarray features with the largest absolute value of the parameter estimates are the same for all three approaches.

The coefficient paths of CoxBoost with the IPI as a mandatory covariate are different, with only a small number of distinct covariates receiving large parameter estimates. The reason for this might be that the mandatory covariate already explains much of the variation in the response and there is less incentive to boost a large number of parameters to fit the remaining variability. The number of boosting steps selected by cross-validation (indicated by vertical lines) also supports this, as it is smaller compared to simple componentwise CoxBoost when IPI is present as a mandatory covariate. In this example, including an unpe-

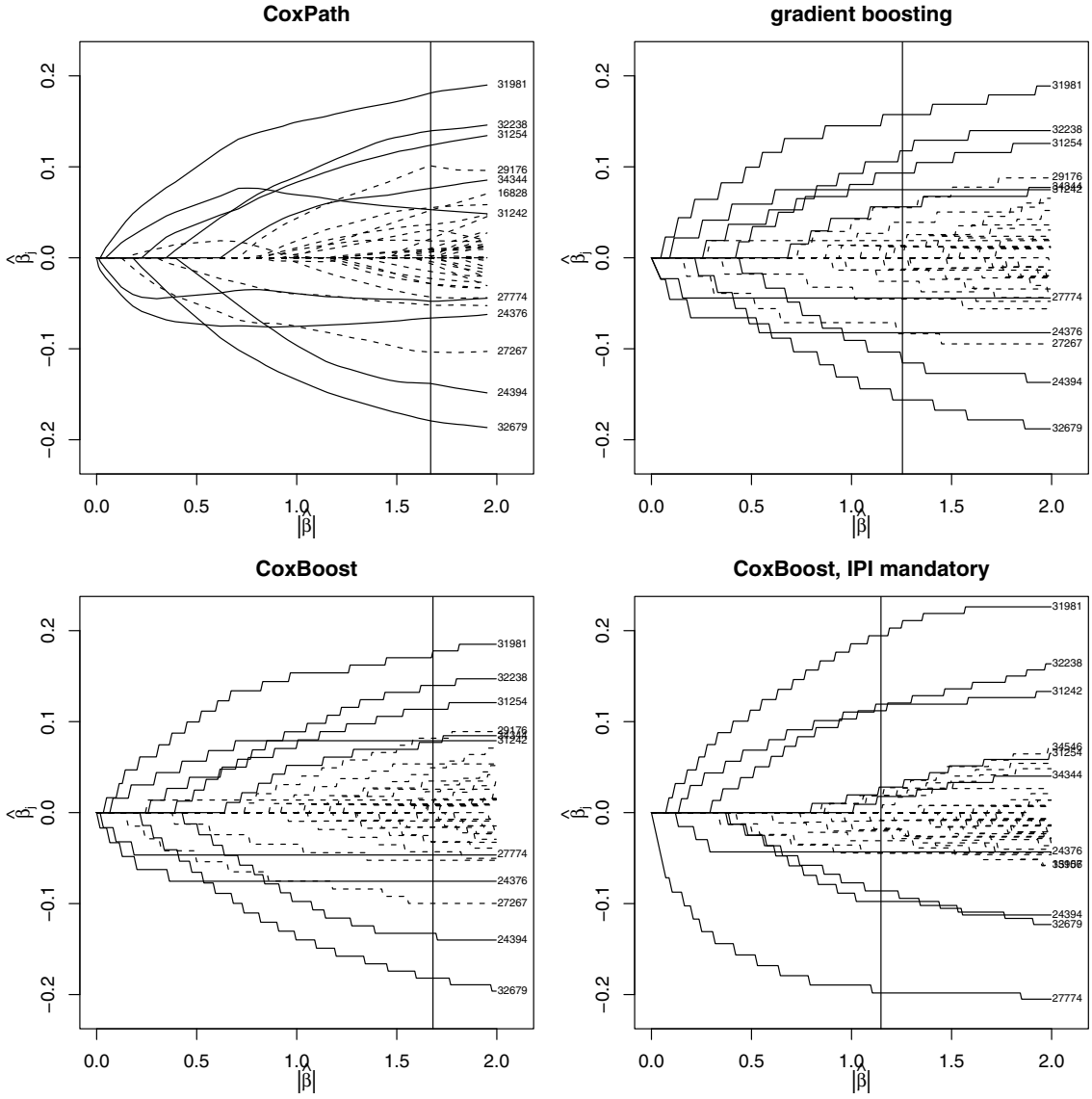

Figure 1

Coefficient paths for CoxBoost. Estimated parameters plotted against the norm of the parameter vector for CoxPath (top left), gradient boosting (top right), componentwise CoxBoost (bottom left), and CoxBoost with a mandatory covariate (bottom right). CoxPath steps and boosting steps selected by cross-validation are indicated by vertical lines. Covariates selected by all approaches up to this number of steps are indicated by solid curves, the others by dashed curves. For them and other strong covariates the UNIQID is given.

nalized mandatory covariate also changes the ranking of the microarray features with respect to the absolute values of the parameter estimates. After inclusion of the IPI the microarray feature with UNIQID 27774 is associated with a strong protective effect, while it seemed to be of minor importance judged by the other fits. In contrast, the feature with UNIQID 32679 is deemed to be less important when the IPI is included as an unpenalized mandatory covariate. So the latter clearly changes the interpretation of the fitted models.

The left panel of Figure 2 shows the .632+ prediction error estimates for all models that incorporate only microarray features, i.e., CoxPath (dotted curve), gradient boosting (dash-dotted curve), and componentwise CoxBoost (solid curve). It is seen that all three perform very similar. The prediction error is well below the Kaplan-Meier bench-

mark (gray curve), which does not employ any covariate information. This is not self-evident, as for example in the evaluation in [20] some other procedures failed with respect to this criterion. So the offset-based boosting approach does not seem to result in a loss of prediction performance and it therefore is a reasonable basis for an approach incorporating unpenalized mandatory covariates. While according to the prediction error curve estimates there seems to be no disadvantage for CoxPath, the out-of-bag partial log-likelihood, i.e., the mean partial log-likelihood evaluated for the observations not in the respective bootstrap samples, is the smallest for this procedure (-183.8). For gradient boosting and componentwise CoxBoost it is -181.5 (with standard errors of about 1.4), i.e., also with respect to this error measure there seems to be no disadvantage of using the CoxBoost approach. A similar pattern is seen for models that incorporate the IPI as an optional covariate in addition to microarray features (middle panel of Figure 2). There is a general improvement over models that did not include the IPI, with all procedures again performing very similar. According to the prediction error curve estimates there may be a slight advantage for CoxPath, which seems to gain the most prediction performance. However, the out-of-bag partial log-likelihood is again the smallest for this procedure (-180.3), while for gradient boosting it is -180.0, and for CoxBoost it is even -177.8.

The effect of various ways for dealing with clinical covariates is illustrated in the right panel of Figure 2. There the .632+ prediction error estimate for componentwise CoxBoost is given, together with prediction error curves for CoxBoost approaches that incorporate the IPI, either as an optional (curve with plus symbols) or as a mandatory covariate (curve with triangles). In addition, the estimated prediction error curve for a standard Cox model that incorporates only the IPI is given (dashed curve). The performance of a microarray-only CoxBoost fit (solid curve) is roughly similar to the Cox model that includes only the clinical information from the IPI (out-of-bag partial log-likelihood: -177.8), with an advantage for the latter for early prediction times. So both types of model might contain the same amount of information. The question whether the microarray features contain information that is different from that of clinical covariates is therefore still unanswered. When, as a first step, the IPI is included as an additional optional covariate for componentwise CoxBoost, as already noted, there is a distinct increase in prediction performance compared to componentwise CoxBoost based only on microarray features. So the two types of covariates seem to contain (at least partially) dif-

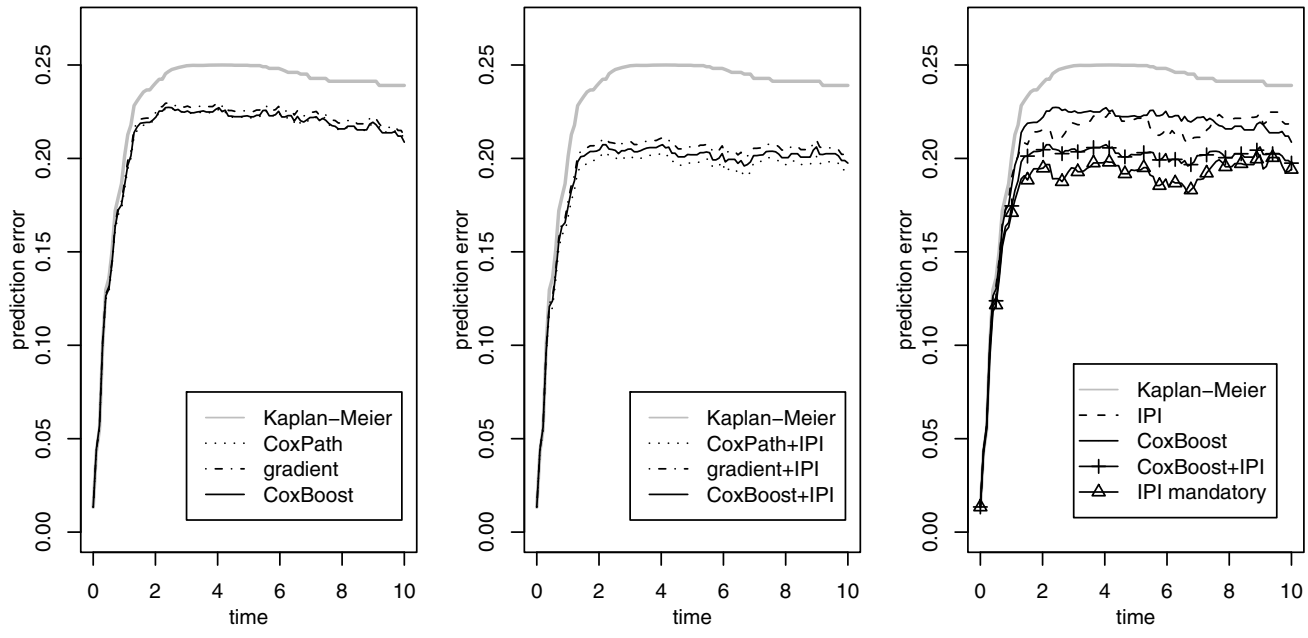

Figure 2
.632+ prediction error estimates. .632+ prediction error curve estimates for microarray-only models (left panel) fitted by CoxPath, gradient boosting, and componentwise CoxBoost, for models including the IPI as an additional optional covariate (middle panel), and for the CoxBoost fit that incorporates the IPI as a mandatory covariate (right panel). The Kaplan-Meier benchmark is indicated by gray curves.

ferent information. When the IPI is included as an unpenalized mandatory covariate the performance increases even more (out-of-bag partial log-likelihood: -175.3). This shows that it is really necessary to assign the IPI this special role, as otherwise it cannot exert its full predictive potential. Here the flexibility of the CoxBoost approach allows to incorporate subject matter knowledge, i.e., knowing that the IPI is a good predictor, to increase predictive performance. CoxBoost with the IPI as a mandatory covariate also allows for a valid comparison to the Cox model that contains IPI as its only covariate. As in both models the parameters for the IPI are estimated unpenalized, the exact additional value of the microarray features in terms of predictive performance can be seen from the difference between the two curves.

Figure 3 shows boxplots of the integrated prediction error estimates (up to time 10) calculated for the single bootstrap samples, to convey an impression of the variability underlying the estimates in Figure 2. It is seen that the conclusions drawn from the prediction error curve estimates hold, even when variability is taken into account. For microarray-only models and models that incorporate

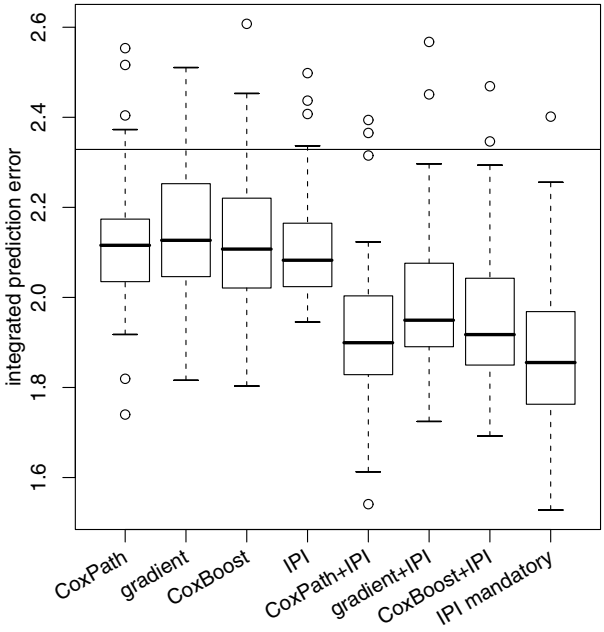

Figure 3
Variability of the .632+ prediction error estimates. Integrated prediction error curve estimates from single bootstrap samples for CoxPath, gradient boosting, componentwise CoxBoost, and an IPI-only Cox model ("IPI"), for corresponding models where the IPI is included as a an additional optional covariate ("...+IPI"), and for CoxBoost fits that incorporate the IPI as a mandatory covariate ("IPI mandatory"). The Kaplan-Meier benchmark value is indicate by a horizontal line.

the IPI only as an optional covariate, CoxPath, gradient boosting, and CoxBoost perform very similar, but when the IPI is incorporated as an unpenalized mandatory covariate, there is an advantage in terms of prediction performance for CoxBoost.

Conclusion

Modern techniques for the fitting of predictive survival models, such as Lasso-like approaches and boosting, are capable of handling the large number of covariates often arising in bioinformatics applications, e.g. from microarrays. What has been missing is an approach for incorporating mandatory covariates into such models. We therefore adapted an offset-based boosting approach, which allows for flexible penalization of covariates, for the estimation of Cox proportional hazard models.

The flexible penalty structure of the new approach allows for unrestricted estimation of the parameters for mandatory covariates. As seen in an example application, this also influences the coefficient paths for the optional covariates, in this case resulting in a more transparent structure. The main benefit, on the one hand, was increased prediction performance by combining clinical and microarray information. On the other hand, the increase of prediction performance over a microarray-only model and a purely clinical predictive model helped to answer the question about the additional benefit arising from microarray technology for predicting survival. In the example, including a mandatory covariate also affected the ranking of microarray features with respect to absolute value of the parameter estimates and therefore potentially changed the clinical implications of the result.

Componentwise gradient boosting approaches could potentially also be adapted for incorporating unpenalized mandatory covariates. However, simply augmenting the componentwise base learners by mandatory components would not be sufficient, as in gradient boosting the base learner fits are multiplied by some small constant ε before adding them to the overall fit. Therefore the building up of the coefficient estimates for mandatory covariates would still be rather slow. Introducing intermediate steps with $\varepsilon = 1$, where only mandatory covariates are updated, could address this. However, the offset-based boosting approach, which we used as a basis for the CoxBoost algorithm, more naturally allows for unpenalized mandatory components.

Incorporating unpenalized mandatory covariates is only one of the many possible ways of leveraging clinical information and subject matter knowledge using the proposed boosting approach. For example, information from clustering of the microarray features could be incorporated, by distributing boosting steps over a set of clusters. Further

refinements of the boosting scheme and the penalization structure could be devised, for further increasing prediction performance and to more generally increase the usefulness of the resulting predictive model.

Methods

Measures of prediction error

By estimating the parameter vector $\hat{\beta}$ of a Cox model (1), a risk prediction model

$$\hat{r}(t \mid x_i) = \exp(-\hat{\Lambda}_0(t)\exp(x_i'\hat{\beta}))$$

is obtained, where $\hat{\Lambda}_0(t)$ denotes the Breslow estimator of the cumulative baseline hazard $\Lambda_0(t) = \int_0^t \lambda_0(s)ds$. It predicts the event status

$$Y_i(t) = I(T_i > t),$$

where $I()$ takes the value 1 if its argument is true and 0 otherwise. T_i is the survival time of subject i, that is unobserved in case of censoring. The observed time t_i therefore is $t_i = min(T_i, C_i)$, where C_i is the censoring time.

The *true prediction error curve* then is

$$Err(t; r) = E[(Y(t) - r(t \mid x))^2].$$

It can be estimated from a sample via

$$\overline{err}(t; r) = \frac{1}{n}\sum_{i=1}^{n}(Y_i(t) - r(t \mid x_i))^2 W(t; G), \qquad (5)$$

where weights $W(t; \hat{G})$ have to be introduced to account for censoring. To obtain a consistent estimate of the true prediction error curve they have to be chosen to be

$$W(t; G) = \frac{I(t_i \le t)\delta_i}{G(t_i - \mid x_i)} + \frac{I(t_i > t)}{G(t \mid x_i)},$$

where $\hat{G}(t|x)$ is a consistent estimate of $P(C > t|x)$. We use a Kaplan-Meier estimator for the latter. For more details see [23].

.632+ prediction error estimates

Evaluating (5) with the data that was used for estimating $\hat{\beta}$ potentially underestimates the prediction error. We therefore generate sets of indices $\mathcal{J}_b \subset \{1, ..., n\}$, $b = 1, ..., B$, for $B = 100$ bootstrap samples, each of size $0.632n$. Sampling without replacement is used to avoid a potential complexity selection bias (i.e., for selecting the number of boosting steps or CoxPath steps) indicated e.g. in [24]. The bootstrap cross-validation error estimate is then obtained by

$$\widehat{Err}_{B0}(t, r) = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{b_0}\sum_{i \notin \mathcal{J}_b}(Y_i(t) - r_b(t \mid x_i))^2 W(t, G), \qquad (6)$$

where b_0 is the number of observations not in $\mathcal{J}_b$, i.e., $0.368n$, and $\hat{r}_b$ is the model fitted to the observations with indices in $\mathcal{J}_b$.

As (6) is known to be biased upwards, we use the .632+ estimate

$$\widehat{Err}_{.632+}(t, r) = \{1 - \omega(t)\}\overline{err}(t, r) \mid \omega(t)\widehat{Err}_{B0}(t, r), \qquad (7)$$

with $\hat{\omega}(t) = .632/(1 - .368\ \hat{R}(t))$, where $\hat{R}(t)$ is the relative overfitting rate $\hat{R}(t) = \frac{\widehat{Err}_{B0}(t,\hat{r}) - \overline{err}(t,\hat{r})}{NoInf(t,\hat{r}) - \overline{err}(t,\hat{r})}$, with $NoInf(t, r) = \frac{1}{n^2}\sum_{j=1}^{n}\sum_{i=1}^{n}\{Y_i(t) - r(t \mid x_j)\}^2 W(t, G)$. For more details see [25].

As a summary measure we propose to use the integrated prediction error estimate

$$I\widehat{Err}_{.632+}(t^*, r) = \int_0^{t^*}\widehat{Err}_{.632+}(s, r)ds. \qquad (8)$$

For getting and impression of the variability underlying (7) and (8), (7) is calculated separately for every bootstrap sample, i.e., the outer sum in (6) reduces to one term, and the corresponding integrated prediction error estimates are obtained. The variability of the resulting $B = 100$ individual integrated prediction error estimates can then be compared between different prediction models, e.g. by boxplots.

Authors' contributions

HB developed and implemented the initial version of the proposed algorithm, applied it to the example data and wrote most of the manuscript. MS contributed design decisions for the algorithm, helped with interpretation of the results for the example data and revised the manuscript.

Acknowledgements

We gratefully acknowledge support from Deutsche Forschungsgemeinschaft (DFG Forschergruppe FOR 534).

References

1. Bühlmann P, Hothorn T: **Boosting Algorithms: Regularization, Prediction and Model Fitting.** *Statistical Science* 2008. **to appear**
2. Park MY, Hastie T: **L_1-Regularization Path Algorithms for Generalized Linear Models.** *Journal of the Royal Statistical Society B* 2007, **69(4):**659-677.
3. Tutz G, Binder H: **Boosting Ridge Regression.** *Computational Statistics & Data Analysis* 2007, **51(12):**6044-6059.
4. Tibshirani R: **The Lasso Method for Variable Selection in the Cox Model.** *Statistics in Medicine* 1997, **16(4):**385-395.
5. Ma S, Huang J: **Additive Risk Survival Model with Microarray Data.** *BMC Bioinformatics* 2007, **8(192):**.
6. Datta S, Le-Rademacher J, Datta S: **Predicting Patient Survival from Microarray Data by Accelerated Failure Time Modeling Using Partial Least Squares and LASSO.** *Biometrics* 2007, **63:**259-271.
7. Friedman JH, Hastie T, Tibshirani R: **Additive Logistic Regression: A Statistical View of Boosting.** *The Annals of Statistics* 2000, **28:**337-407.
8. Ridgeway G: **The State of Boosting.** *Computing Science and Statistics* 1999, **31:**172-181.
9. Bühlmann P: **Boosting for High-Dimensional Linear Models.** *The Annals of Statistics* 2006, **34(2):**559-583.
10. Yuan M, Lin Y: **Model Selection and Estimation in Regression With Grouped Variables.** *Journal of the Royal Statistical Society B* 2006, **68:**49-67.
11. Kim Y, Kim J, Kim Y: **Blockwise Sparse Regression.** *Statistica Sinica* 2006, **16(2):**375-390.
12. Ma S, Son X, Huang J: **Supervised Group Lasso With Applications to Microarray Data Analysis.** *BMC Bioinformatics* 2007, **8(60):**.
13. Verweij PJM, van Houwelingen HC: **Penalized Likelihood in Cox Regression.** *Statistics in Medicine* 1994, **13:**2427-2436.
14. Efron B, Hastie T, Johnstone I, Tibshirani R: **Least Angle Regression.** *The Annals of Statistics* 2004, **32(2):**407-499.
15. Hastie T, Taylor J, Tibshirani R, Walther G: **Forward Stagewise Regression and the Monotone Lasso.** *Electronic Journal of Statistics* 2006, **1:**1-29.
16. R Development Core Team: *R: A Language and Environment for Statistical Computing* 2007 [http://www.R-project.org]. R Foundation for Statistical Computing, Vienna, Austria ISBN 3-900051-07-0
17. Rosenwald A, Wright G, Chan WC, Connors JM, Campo E, Fisher RI, Gascoyna RD, Muller-Hermelink HK, Smeland EB, Staudt LM: **The Use of Molecular Profiling to Predict Survival After Chemotherapy for Diffuse Large-B-cell Lymphoma.** *The New England Journal of Medicine* 2002, **346(25):**1937-1946.
18. Segal M: **Microarray Gene Expression Data With Linked Survival Phenotypes: Diffuse Large-B-cell Lymphoma Revisited.** *Biostatistics* 2006, **7(2):**268-285.
19. The International Non-Hodgkin's Lymphoma Prognostic Factors Project: **A Predictive Model for Aggressive Non-Hodgkin's Lymphoma: Report of the Jury.** *New England Journal of Medicine* 1993, **329:**987-994.
20. Schumacher M, Binder H, Gerds TA: **Assessment of Survival Prediction Models Based on Microarray Data.** *Bioinformatics* 2007, **23(14):**1768-1774.
21. Hothorn T, Thomas Kneib PB, Schmid M: *mboost: Model-Based Boosting* 2007. R package version 1.0-0
22. Park MY, Hastie T: *glmpath: L1 Regularization Path for Generalized Linear Models and Cox Proportional Hazards Model* 2007. R package version 0.94
23. Gerds TA, Schumacher M: **Consistent Estimation of the Expected Brier Score in General Survival Models with Right-Censored Event Times.** *Biometrical Journal* 2006, **48(6):**1029-1040.
24. Steck H, Jaakkola T: **Bias-Corrected Bootstrap and Model Uncertainty.** *Advances in Neural Information Processing Systems* 2003, **16:**.
25. Gerds TA, Schumacher M: **Efron-type measures of prediction error for survival analysis.** *Biometrics* 2007, **63(4):**1283-1287.