LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Heidi Seibold, Christoph Bernau, Anne-Laure Boulesteix and Riccardo De Bin

On the choice and influence of the number of boosting steps

Technical Report Number 188, 2015
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

On the choice and influence of the number of boosting steps

Heidi Seibold[1,2,*], Christoph Bernau[3], Anne-Laure Boulesteix[2], and Riccardo De Bin[2]

[1]Epidemiology, Biostatistics and Prevention Institute, University of Zurich
[2]Department of Medical Informatics, Biometry and Epidemiology, University of Munich
[3]Leibniz Supercomputing Centre, Munich

January 7, 2016

Abstract

In biomedical research, boosting-based regression approaches have gained much attention in the last decade. Their intrinsic variable selection procedure and their ability to shrink the estimates of the regression coefficients toward 0 make these techniques appropriate to fit prediction models in the case of high-dimensional data, e.g. gene expressions. Their prediction performance, however, highly depends on specific tuning parameters, in particular on the number of boosting iterations to perform. This crucial parameter is usually selected via cross-validation. The cross-validation procedure may highly depend on a completely random component, namely the considered fold partition. We empirically study how much this randomness affects the results of the boosting techniques, in terms of selected predictors and prediction ability of the related models. We use four publicly available data sets related to four different diseases. In these studies the goal is to predict survival end-points when a large number of continuous candidate predictors are available. We focus on two well known boosting approaches implemented in the R-packages CoxBoost and mboost, assuming the validity of the proportional hazards assumption. Finally, we empirically show how the variability in selected predictors and prediction ability of the model is reduced by averaging over several repetitions of cross-validation in the selection of the tuning parameters.

Keywords: Boosting · Cross-validation · Parameter tuning · High dimensional data · Survival analysis

1 Introduction

Boosting-based regression approaches have gained a lot of attention in the last decade, showing both interesting theoretical properties (Bühlmann and Yu, 2003; Bühlmann, 2006; Tutz and Binder, 2006) and yielding good empirical results in terms of prediction accuracy, including applications to prediction with high-dimensional data. In this paper we focus specifically on two boosting approaches that are based on a solid theoretical framework, implemented in user-friendly software, and able to efficiently cope with high-dimensional data and handle censored survival end-points: the model-based boosting approach (Bühlmann and Yu, 2003),

[*] corresponding author: heidi.seibold@uzh.ch

implemented in the R package *mboost* (Hothorn et al, 2015); and the likelihood-based boosting approach (Tutz and Binder, 2006) adapted to survival end-points by Binder and Schumacher (2008) and implemented in the R package *CoxBoost* (Binder, 2013).

In our analyses we focus on prediction models for time-to-event outcomes: this kind of application, despite being extremely common in biomedical practice, has not been well investigated in statistical literature in the case when a large number of candidate predictors, such as gene expressions, are available. In this context, boosting techniques can play an important role. They have, indeed, two important characteristics which are essential in providing a good prediction model when the number of the predictors exceeds the sample size: the ability to shrink the parameter estimates toward 0, and the identification of the relevant predictors (variable selection). The latter is performed by allowing only a moderate number of parameters to have non-zero values. These two properties suggest the existence of a relation between boosting techniques and methods based on penalized regression. Works which have investigated this connection, mainly focusing on the similarities between L_2-boosting and lasso, are Hastie et al (2001), Efron et al (2004) and Bühlmann and Hothorn (2007).

Another common characteristic of the boosting and the penalized regression techniques is the presence of one or more tuning parameters. In particular, as boosting is an iterative method in which a weak learner is sequentially applied to a suitable modification of the data, the most critical parameter to set is the number of iterations (boosting steps). Its choice greatly impacts the number of involved predictors and the complexity of the resulting prediction model. Despite the importance of this parameter, literature on its choice is scarce. The R packages *mboost* and *CoxBoost* exploit cross-validation-based procedures. In particular, when the working with proportional hazards models, both packages implement the cross-validated partial log-likelihood by Verweij and Van Houwelingen (1993). The package *mboost* also offers a different procedure, based on the Akaike information criterion: introduced by Bühlmann (2006) and investigated in the survival analysis context by Hothorn et al (2006), its use in practice is actually discouraged due to its tendency to overshoot the optimal value (Hofner et al, 2014). This tendency is primarily due to the systematic underestimation of the true degrees of freedom in component-wise boosting algorithms (Mayr et al, 2012). An advantage of AIC-based stopping criteria is that they can be made totally data-driven, avoiding the necessity of pre-specifying a range of values to search for the optimum. The works of Chang et al (2010) and, especially, Mayr et al (2012) focus on this approach, with the latter adjusting for the underestimation of the degrees of freedom using a re-sampling method, at the expense of computation time.

However, the aforementioned approaches are not really well-known and cross-validation is by far the most popular procedure used in practice to choose the number of boosting steps. Unfortunately, cross-validation is often implemented without taking into account its possible drawbacks and the effect that these can have on the tuning procedure. An important problem of cross-validation and related approaches is the high variability of the results (Boulesteix et al., 2013): the output may be completely different for two different random partitions into the K folds used in the procedure, in the sense that different numbers of boosting steps are identified as optimal depending on the considered random partition. As a consequence, the final prediction model - fit using the selected number of boosting steps - may greatly depend on a completely random component, namely the considered partition into the K folds.

In this paper we address the issue of the choice of the number of boosting steps from an empirical perspective. In particular, we specifically address three questions related to the variability of cross-validation-based results: (i) how much does the prediction accuracy of the final prediction model depend on the random CV partition used for the choice of the number of boosting steps? (ii) how much do the set of selected predictors depend on the random CV partition used for the choice of the number of boosting steps? (iii) to what extent can this variability be reduced through adapting the cross-validation tuning procedure by averaging over several random partitions into K folds? Despite the focus on the prediction of censored survival end-points from high-dimensional data, most conclusions are generalizable to other types of end-points and/or other type of predictors.

This paper is structured as follows. Section 2 gives an introduction to the two considered boosting methods, cross-validation for tuning and the evaluation of survival prediction models using the Brier score. An empirical study based on four high-dimensional gene expression data sets, each consisting of both learning and test sets, is presented in Section 3. The effect of considering several partitions in the cross-validation procedure is shown in Section 4. Finally, Section 5 contains some conclusions.

2 Methods

The general idea of a boosting procedure is to repeatedly fit a weak estimator to the data in order to minimize a loss function. Here we focus on the implementation to survival data of the model-based boosting and the likelihood-based boosting approaches. Both depend on two tuning parameters: a penalty parameter, whose choice is usually hardly influential, and the number of boosting steps, m_{stop}, which, on the contrary, greatly affects the performance of the procedure and, consequently, the behavior of the resulting prediction model. In this section, we briefly review the two boosting algorithms, we sketch how to apply the cross-validation technique in order to select m_{stop}, and we provide some information on the Brier score, the measure of prediction ability that we use in the paper.

2.1 Model-based boosting

Model-based boosting is a direct implementation of the gradient boosting idea described in the seminal paper of Friedman (Friedman, 2001), which provides a statistical view of the boosting technique introduced by Freund and Schapire (1996) in the machine learning literature. In the Friedman paper, boosting is characterized as a gradient descent algorithm, where in each iteration a base learner is fit to the negative gradient of a loss function. Here we focus on its adaption to survival data which fit the Cox model assumptions, as implemented in the package *mboost* within the function *glmboost* with argument *family=CoxPH()*. In particular, this version uses the negative partial likelihood as the loss function and the ordinary least squares estimator as the base-learner. The derivation of the negative gradient vector was firstly provided in Ridgeway (1999). Based on the *mboost* function, other implementations using specific weights (Hothorn et al, 2006) or considering non-linear effect for the predictors (e.g., Schmid and Hothorn, 2008) are available through the *mboost* function, but are not considered here.

The package *mboost* implements the component-wise boosting version, the use of which is often motivated by the challenges typical of high-dimensional data. This procedure consists of updating the vector of regression coefficient estimates only one dimension at a time. At each step, for all the vector components, a possible update is computed by fitting a least squares estimator on the gradient vector. Among all possible updates, the one which decreases the loss function the most is selected, and it is added, suitably multiplied by a penalty parameter, to the related regression coefficient estimate. This updating procedure ends when the prespecified number of boosting steps m_{stop} is reached. It is worth stressing the crucial role of this parameter: if it is too small the estimates of the regression coefficients may be insufficiently refined, leading to a prediction model unable to explain the outcome variability; if it is too large, the final model risks being too complex and overfitting the learning data. The number of boosting steps highly affects the variable selection property of the boosting procedure as well: the chance of including a predictor in the model, indeed, increases with the number of iterations. Therefore, if the number of steps performed is too small, a relevant predictor may be excluded from the model, while if it is too large, irrelevant predictors may be included, with high risk, especially in the high-dimensional data context, of overfitting. In contrast, the choice of the penalty term is unimportant, and, in our analyses, we keep the default value (0.10, see, e.g., Bühlmann and Hothorn, 2007).

2.2 Likelihood-based boosting

The second algorithm that we consider is the adaptation to survival data of likelihood-based boosting (Tutz and Binder, 2006), introduced by Binder and Schumacher (2008) and implemented in the R package *CoxBoost*. This algorithm uses a penalized version of the negative partial log-likelihood as the loss function, which it minimizes by repeatedly fitting a first order approximation of the ridge estimator. In the component-wise version used in this paper, only one regression coefficient per iteration is updated, although the R package offers the chance to update more at each step (Binder and Schumacher, 2008). In practice, at each step all possible updates (one for each regression coefficient) are computed, and then the most relevant – namely that which, once plugged into the loss function, leads to the smallest value – is selected. This "best" update is incorporated in an offset term, which is simply the linear predictor obtained in the previous boosting step. Again, the total number of boosting steps performed is highly relevant in determining the behavior of the resulting prediction model, and a good choice of this tuning parameter is again crucial. As with the model-based boosting technique, there is a second tuning parameter to consider, the penalty term. In this case, it is directly applied to the partial log-likelihood, through the L_2 norm which characterizes the ridge regression. The penalty term is usually selected through the rough method implemented in the function *optimCoxBoostPenalty* of the package *CoxBoost*. In this paper: (i) to have a more robust result, we repeat the procedure 100 times and take the median value; (ii) since we will consider several kinds of cross-validation (leave-one-out, 3-, 5-, 10 and 20- fold), we repeat the procedure for each kind of cross-validation and we select the median value among the 5 penalty parameters. The use of a single penalty term for all kinds of cross-validation procedure assures the comparability of their results in terms of the number of boosting steps. Obviously this procedure does not optimize the value of the penalty parameter, but it quickly provides a term with a reasonable magnitude: as with model-based boosting, the choice of the penalty parameter is not crucial. The original paper only claims that a "large enough" value is necessary (Binder and Schumacher, 2008).

2.3 Choice of the tuning parameter based on cross-validation

The number of boosting steps is highly relevant in both boosting procedures considered.We stated in the introduction that the usual way to compute its value is through cross-validation

(CV). The general idea of cross-validation is to mimic the presence of a learning and a test set by splitting the available data set D into K disjoint and approximately equal-sized subsets $D_1, ..., D_K$. Each fold of this split is then separately used as test set to evaluate the behavior of a model fit on the other $K-1$ folds.

In the R implementation of the two boosting procedures analyzed, the evaluation is made in terms of the cross-validated partial log-likelihood introduced by Verweij and Van Houwelingen (1993),

$$cvpl(m) = \sum_{k=1}^{K} \left(pl(\hat{\boldsymbol{\beta}}_m^{(-D_k)}) - pl^{(-D_k)}(\hat{\boldsymbol{\beta}}_m^{(-D_k)}) \right), \tag{1}$$

where $pl(\cdot)$ denotes the complete partial log-likelihood, $pl^{(-D_k)}(\cdot)$ the partial log-likelihood computed without the observations contained in the k-th fold and $\hat{\boldsymbol{\beta}}_m^{(-D_k)}$ denotes the vector of the regression coefficient estimates computed using the $D \backslash D_k$ subset. Note that the value of the first term on the right hand side of Equation 1 increases with increasing proximity of $\hat{\boldsymbol{\beta}}_m^{(-D_k)}$ to the maximum likelihood estimate (mle). The second term, instead, penalizes for possible overfitting: it is computed on the data used to obtain $\hat{\boldsymbol{\beta}}_m^{(-D_k)}$, and therefore it decreases the value of $cvpl(m)$ as much as $\hat{\boldsymbol{\beta}}_m^{(-D_k)}$ explains too much the data variability.

The cross-validated partial log-likelihood is used to estimate the optimal number of boosting steps. The estimates of the regression coefficients, indeed, depends on m, as highlighted by the subscripts in Equation 1. The optimal value m_{stop}, therefore, is obtained by maximizing over m the cross-validated partial log-likelihood.

2.4 Brier score and integrated Brier score

The Brier score is a quadratic score rule originally developed to measure the accuracy of weather forecasts (Brier, 1950) and adapted to the context of survival analysis by Graf et al (1999). It is based on the predicted survival probability $\hat{S}_i(t)$, that, ideally, at time t should be 1 if the subject i is alive, 0 otherwise (Schumacher et al, 2007). If $I(T_i > t)$ indicates whether the observation i is or is not alive at time t, the Brier score can be estimated as

$$\hat{BS}(t) = \frac{1}{n} \sum_{i=1}^{n} \hat{W}_i(t) \left(I(T_i > t) - \hat{S}_i(t) \right)^2$$

where n is the number of the observations in the test data set and $\hat{W}_i(t)$ are weights introduced in order to deal with censored observations (for further details, see Gerds and Schumacher, 2006; Mogensen et al, 2010). Please note that the survival probability estimation $\hat{S}$ is computed using the test set, but is calculated based on the model determined using the learning set.

When plotted with respect to time, the Brier score leads to the so-called prediction error curves, which can be used to graphically investigate the behavior of the predictive model. Alternatively, we can summarize the information in a single value, called the "integrated Brier score", by integrating the Brier score with respect to the time. The integrated Brier score corresponds to the measure of the area under the prediction error curves,

$$I\hat{BS} = \int_0^T \hat{BS}(t)\, dt,$$

Table 1: The four data sets used in our empirical study.

disease	sample size (events)		number of	reference
	learning set	test set	predictors	
breast cancer	282 (57)	182 (41)	22283	Hatzis et al (2011)
diffuse large B-cell lymphoma	149 (79)	73 (48)	7399	Rosenwald et al (2002)
acute myeloid leukemia	163 (103)	79 (32)	44754	Metzeler et al (2008)
neuroblastoma	242 (40)	120 (35)	9978	Oberthuer et al (2008)

where T is the value up to which the integral is considered. In our study, we select T as the largest time value in the test set.

3 Empirical study

3.1 Data

In our analyses, we consider four publicly available medical data sets with survival outcome and information on the gene expression of the patients (see Table 1). Each of these data sets consists of a learning set, using which we compute the optimal number of boosting steps and fit the model, and a test set, for which we compute the integrated Brier score. It is particularly important to keep the learning and the test data totally separated in order to have a reliable evaluation of the prediction abilities of the resulting models.

Breast cancer data: This data set comes from a prospective multicenter study conducted by Hatzis et al (2011) to develop genomics predictors for neoadjuvant chemotherapy. It involves patients with newly diagnosed ERBB2 (HER2 or HER2/neu)-negative breast cancer, for which information is provided on the (possibly censored) distant relapse-free survival time and the gene expressions of 22283 probe sets, obtained through the Affymetrix U133A GeneChip. The data set consists of a learning set, containing information on patients who had their biopsy between June 2000 and December 2006, and an independent test set, whose patients had their biopsy between April 2002 and January 2009. Specifically, we use the observations considered in De Bin et al (2014): the sample sizes are 282 patients (with 57 events) and 182 patients (41 events) for the learning and test sets, respectively. The data are publicly available from the Gene Expression Omnibus, reference GSE25066.

Diffuse large B-cell lymphoma: The second data set comes from the study of Rosenwald et al (2002) on patients with diffuse large B-cell lymphoma. It contains 7399 gene-expression measurements from 240 patients who had no previous history of lymphoma, divided in a learning set (160 patients) and a test set (80 patients). The outcome of interest is the overall survival time. In our paper we use the data set as pre-processed by Bøvelstad et al (2009), which contains the information of only the 222 patients for which the International Prognostic Index is also available. However, we did not consider this predictor in our analysis. As a result of this restriction, the learning and test sets contains 149 and 73 patients, respectively. Due to the presence of censored data, the effective sample sizes are 79 (learning set) and 48 (test set).

Acute myeloid leukemia data: The third data set contains information on patients with acute myeloid leukemia enrolled between 1999 and 2003 (learning set) or in 2004 (test set) in a multicenter trial of the German AML Cooperative Group (Metzeler et al, 2008). The outcome of interest is the overall survival, defined as the time between study entry and death from any cause. The learning set contains 163 patients, of which 103 died. The data consist of the gene-expression measurements of 44754 probe sets, obtained using the Affymetrix HG-U133 A&B microarray. For the 79 patients belonging to the test set (32 events), instead, the gene expressions were derived using Affymetrix HG-U133 plus 2.0 microarray. The data are publicly available from the Gene Expression Omnibus, reference GSE12417.

Neuroblastoma data: The last data set contains information on the patients with neuroblastoma studied by Oberthuer et al (2008). The original learning set consists of 256 patients recruited between 1989 and 2004 for the German Neuroblastoma Trial NB90-NB2004 for which the overall survival time and the gene expressions of 9978 probe sets are available. The test set, instead, consists of 120 patients with the same disease, but collected in several countries (29 in Germany, 26 in the US, 26 in France, 12 in Spain, 11 in Italy, 6 in Belgium, 5 in the UK and 5 in Israel), for which the same outcome and probe sets were measured. In our study, we did not directly use the data from the original study (available from the ArrayExpress database, accession number E-MTAB-16), but those pre-processed by Bøvelstad et al (2009), in which 14 patients are excluded due to missing data. Since it was not possible to recover the original split into learning and test sets, here we randomly split the whole data set into a learning set of 242 patients (40 events) and a test set of 120 patients (35 observations), which are the sample sizes used by Bøvelstad et al (2009).

3.2 Study design

The main focus of our study is on the cross-validation-based choice of the optimal number of boosting steps in model-based and likelihood-based boosting. We consider values between 0 (null model) and 200. We investigate how the variability caused by the randomness due to the cross-validation fold-split affects the results of the boosting procedures in terms of number of iterations performed, selected predictors and prediction ability of the models.

In our analysis, for both boosting techniques we replicate 2000 times the following algorithm:

- we apply the 3-, 5-, 10- and 20-fold cross-validation procedures to compute the optimal number of boosting steps, using only the observations from the learning set;
- we fit a prediction model by applying the boosting technique to the learning set, using the tuning parameter obtained in the previous point;
- we note the number of predictors selected in the model;
- we evaluate the prediction ability of the model by estimating the integrated Brier score on the test set.

In addition, we collect the same information (number of boosting steps, number of selected predictors, integrated Brier score) when using leave-one-out cross-validation: since this procedure is deterministic, this operation is performed only once.

3.3 Results

3.3.1 Number of boosting steps

The first goal of this empirical study is to evaluate how the optimal number of boosting steps (m_{stop}) is influenced by the different random splits – into learning and test sets – of the cross-validation procedure. Figure 1 shows the distribution of the values obtained over 2000 iterations, for each data set and using the cross-validation procedures implemented both in *mboost* and in *CoxBoost*. For now we focus on the white boxplots, which show the results for the regular cross-validation. The gray boxes will be discussed in the following section. Regardless of the boosting technique chosen, the variability of m_{stop} is very large, with values that range from 0 (minimum) to 200, the upper limit that we considered in our experiment. In particular, this means that, using the same data, we can obtain completely different results simply due to the particular fold-split used. The four considered example data sets suggest that this result may be partially mitigated by a large sample size (although this different behavior may of course also be simply due to random variations): we notice that in the acute myeloid leukemia example, in which we have 103 events, we experience less variability (see Figure 1, first row, third column) than in the other data sets, especially when applying *mboost*. Nevertheless, it is worth noting that the sample sizes and, more in general, the characteristics of all our data sets, are typical of biomedical studies, and therefore in practical situations we may experience this large variability in the choice of m_{stop}. As expected, the variability seems to decrease with an increase in the number of folds, because increasing the number of folds means approaching to (the completely deterministic) leave-one-out cross-validation. Leave-one-out cross-validation produces extreme numbers of steps in *mboost* for the all data sets except the Neuroblastoma data set and for *CoxBoost* in the DLBCL data set. All extreme numbers of steps for leave-one-out cross-validation are higher than most or all numbers of steps computed by other cross-validation procedures. This suggests that leave-one-out cross-validation leads to models that are more likely to overfit the data in these cases.

3.3.2 Selected predictors

The high variability in the choice of m_{stop} is not a problem itself, but it may substantially affect the model building process and consequently the properties of the prediction model. We first consider the selection of the relevant predictors. We report in Figure 2 how many predictors are selected in each of the replications of our experiment for the model-based (*mboost*) and the likelihood-based (*CoxBoost*) boosting procedures, respectively. Moreover, we report in Figure 3 the number of predictors selected at least once. Note that the number of predictors selected at least once and the number of predictors always selected is equivalent for leave-one-out cross-validation, because it is deterministic and was only computed once. Again, we first focus on the regular cross-validation, represented as dots. The complete tables of the selected predictors, including the information on the number of times they are selected, are available in the Supplementary material (Tables 2 – 5).

The different values of m_{stop} as determined by the random fold-splits in the cross-validation procedure, greatly influence the prediction models in terms of selected predictors. In particular, extremely low values of m_{stop} prevent the boosting technique from including many predictors in the model: as a consequence, very few predictors are selected in all 2000 replications performed in our study. On the other hand, high values of m_{stop} can result either in higher

Figure 1: Number of boosting steps (m_{stop}) selected in the 2000 iterations (except leave-one out CV) computed using different CV folds in the four data sets with both *CoxBoost* (left) and *mboost* (right). The color defines the type of cross-validation. White stands for normal, gray for repeated cross-validation.

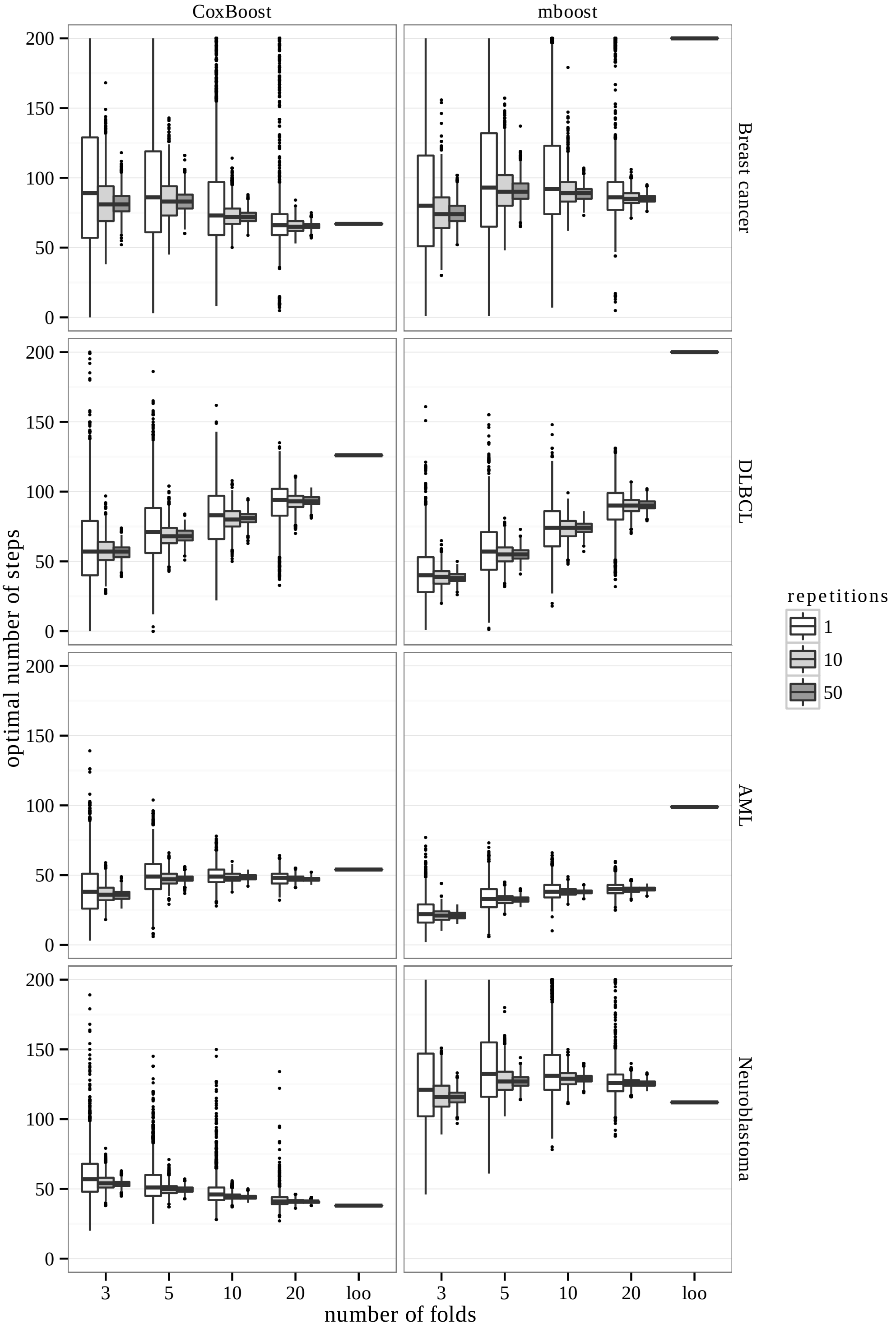


Figure 2: Number of predictors selected in each iteration using different CV folds in the four data sets with both *CoxBoost* and *mboost* (right). The shape defines the type of cross-validation with respect to number of repetitions.

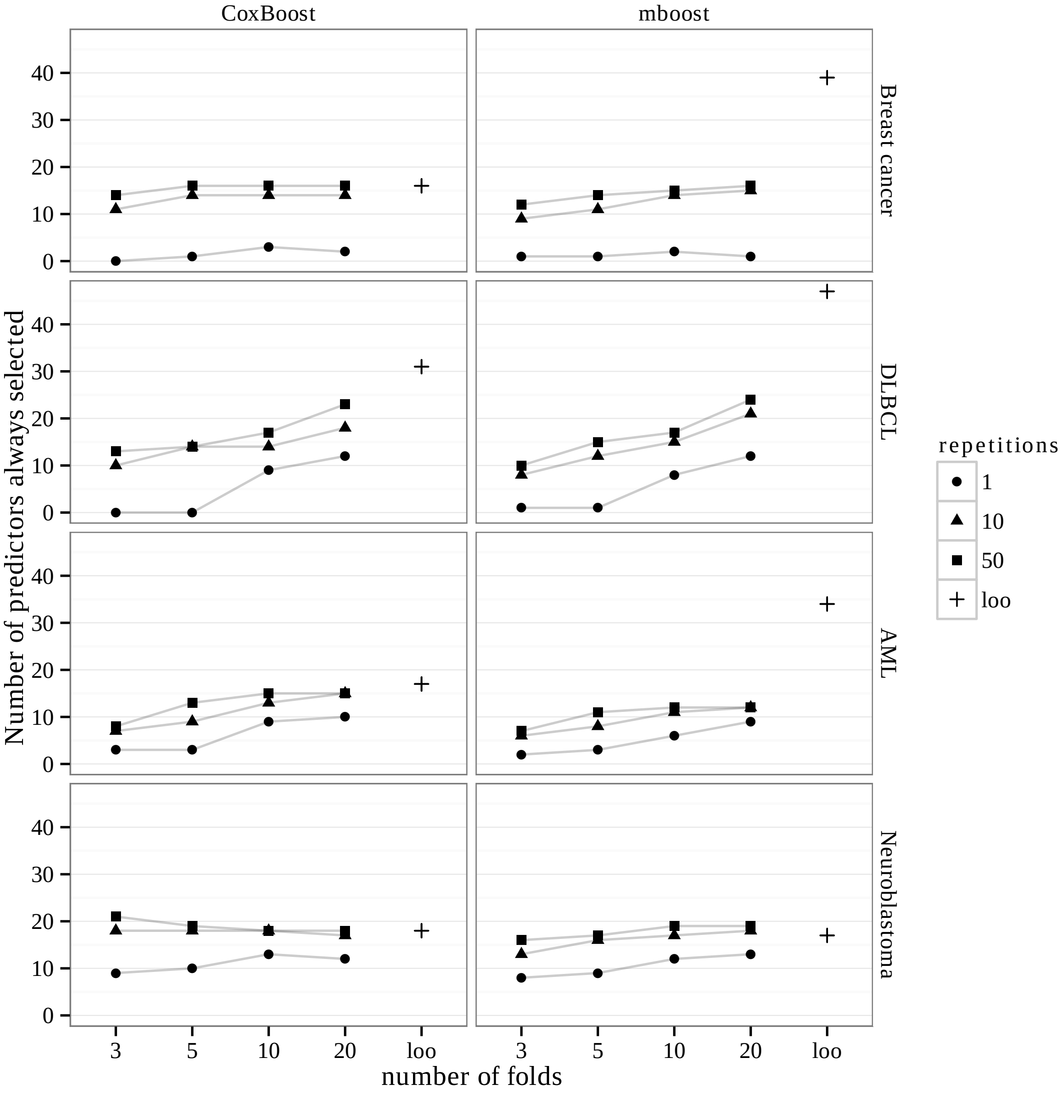
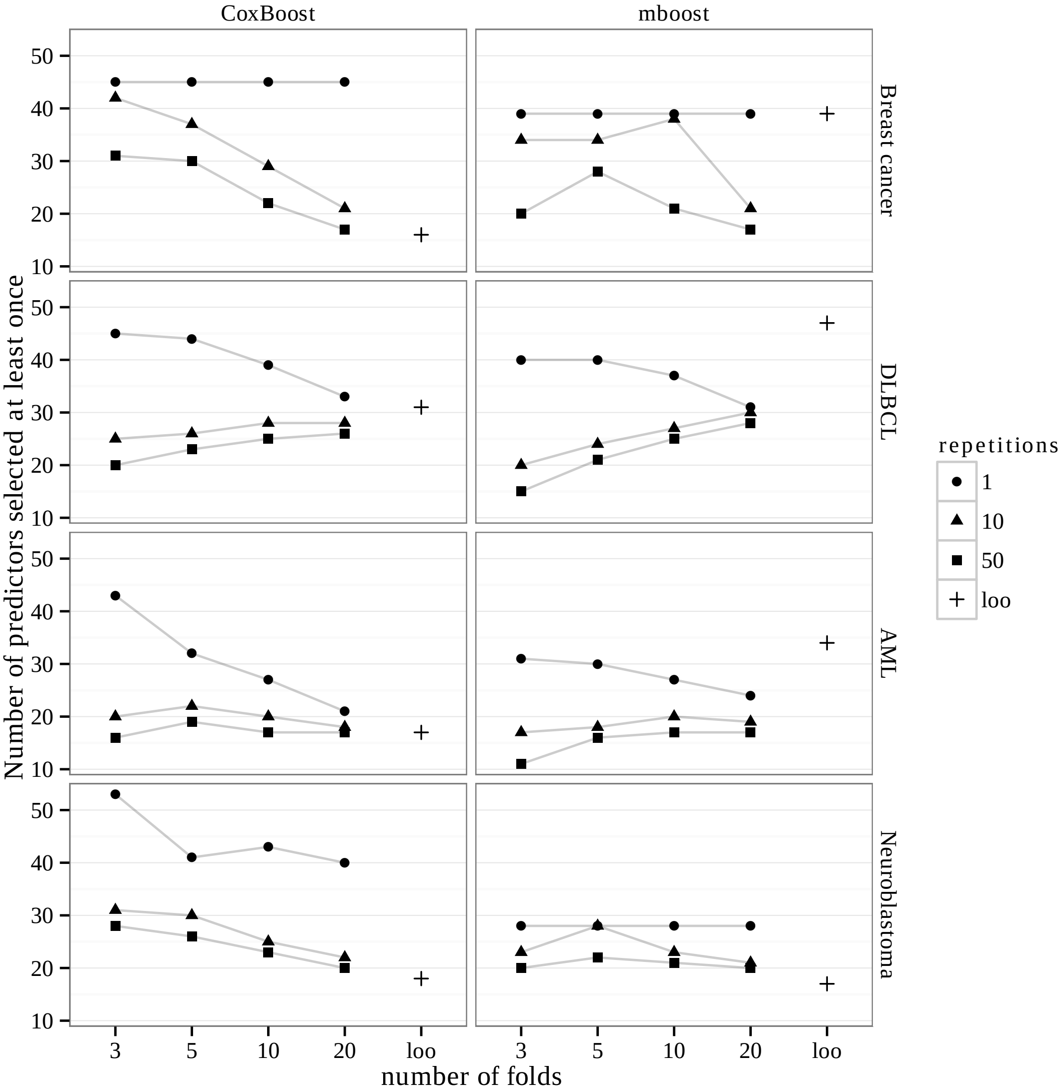


Figure 3: Number of predictors selected at least once in 2000 iterations computed using different CV folds in the four data sets with both *CoxBoost* (left) and *mboost* (right). The shape defines the type of cross-validation with respect to number of repetitions.



values for the estimates of a few predictors or in a high number of selected predictors: in our examples the latter seems to happen, as shown by the relatively large number of predictors selected at least once.

The (relatively) greater stability in the choice of m_{stop} induced by a larger number of folds in the cross-validation procedure results both in an increase in the number of predictors selected in all replications and a decrease in the predictors selected at least once. This is least strong in the application of the breast cancer data: both for *mboost* and *CoxBoost*, the variability of m_{stop} slightly decreases with increasing number of folds but not as strong as in the other applications (see Figure 1, first row). This reflects in a less evident stabilization in the predictors selected. For example using *CoxBoost* the number of predictors always included is 0 for the 3-fold cross-validation, 1 for the 5-fold, 3 for the 10-fold and 2 for the 20-fold for the breast cancer data, whereas for the acute myeloid leukemia data it is 3 for the 3-fold, 3 for the 5-fold, 9 for the 10-fold and 10 for the 20-fold cross-validation. The number of predictors selected at least once is always 45 for the breast cancer data but goes down from 43 (3-fold) to 21 (20-fold) for the acute myeloid leukemia data.

Leave-one-out cross-validation tends to favor more complex models, which are more likely to overfit the learning data. Figures 2 and 3 support that in *mboost* for all data sets except the neuroblastoma data set. For *CoxBoost* a similat behavior can be seen for the DLBCL data. So essentially all examples that showed extremely high values for m_{stop} also show many predictors included in the model.

Finally, we note that in all the four data sets the rank of the predictors based on their inclusion frequencies is slightly different between *mboost* and *CoxBoost.* This is a consequence of the differences in the learning path for the two boosting techniques (for further details, see De Bin, 2015).

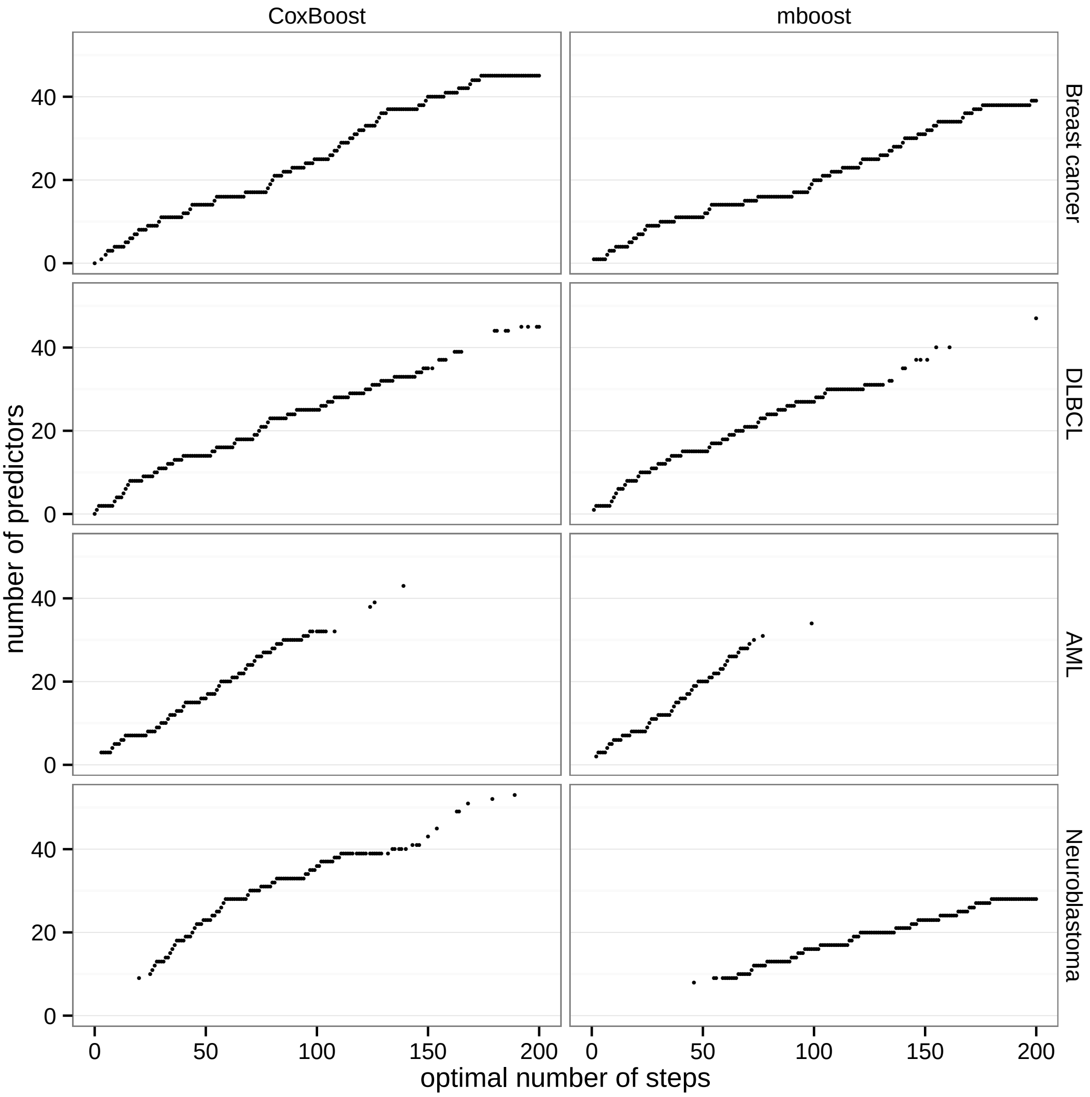
3.3.3 Connection between the number of boosting steps and the number of selected predictors

Through the paper, we stressed the influence of the number of boosting steps on the model sparsity. To better understand this statement, we plot in Figure 4 all values of m_{stop} obtained in our replications against the number of predictors included in the corresponding models. We note that models are less sparse as the value of the optimal number of boosting steps increases, resulting in a non-decreasing function. Steps in this function occur when predictors are chosen that have already been chosen before. Please note that the boosting learning path is deterministic. Therefore, once we know the number of boosting steps (and the penalty factor), we can determine uniquely the fitted model.

Figure 4 shows once again how important a stable selection of the number of boosting steps is. Extremely large values may result in extremely complex models and the other way around for extremely small m_{stop}, with obvious implications in terms of interpretation and prediction accuracy.

We note that the slopes of the curves for *mboost* and *CoxBoost* are fairly similar. The largest difference occurs in the Neuroblastoma data set. Here for the most extreme value that we allow for m_{stop}, namely 200, the number of predictors is much lower for *mboost* (28) than for *CoxBoost* (53). Please note that the slopes of the curves are also strongly related to the value chosen for the penalty parameter. The stronger the penalty (i.e., smaller ν for *mboost*, larger λ for *CoxBoost*, see also De Bin, 2015), the less steep the curve. For *mboost* we used $\nu = 0.1$ and for CoxBoost $\lambda = 2052$ for the breast cancer data, $\lambda = 1422$ for the

Figure 4: Optimal number of steps chosen via cross-validation plotted against the number of predictors included in the respective model, for both *CoxBoost* (left) and *mboost* (right).



DLBCL data, $\lambda = 1854$ for the AML data and $\lambda = 720$ for the neuroblastoma data. Larger values of the penalty parameter correspond to smaller step-wise updates of the coefficients, and consequently to more stationary point; with a larger penalty it may be necessary to perform two boosting steps to obtain the same coefficient update obtained in one step in case of a small penalty.

3.3.4 Prediction ability

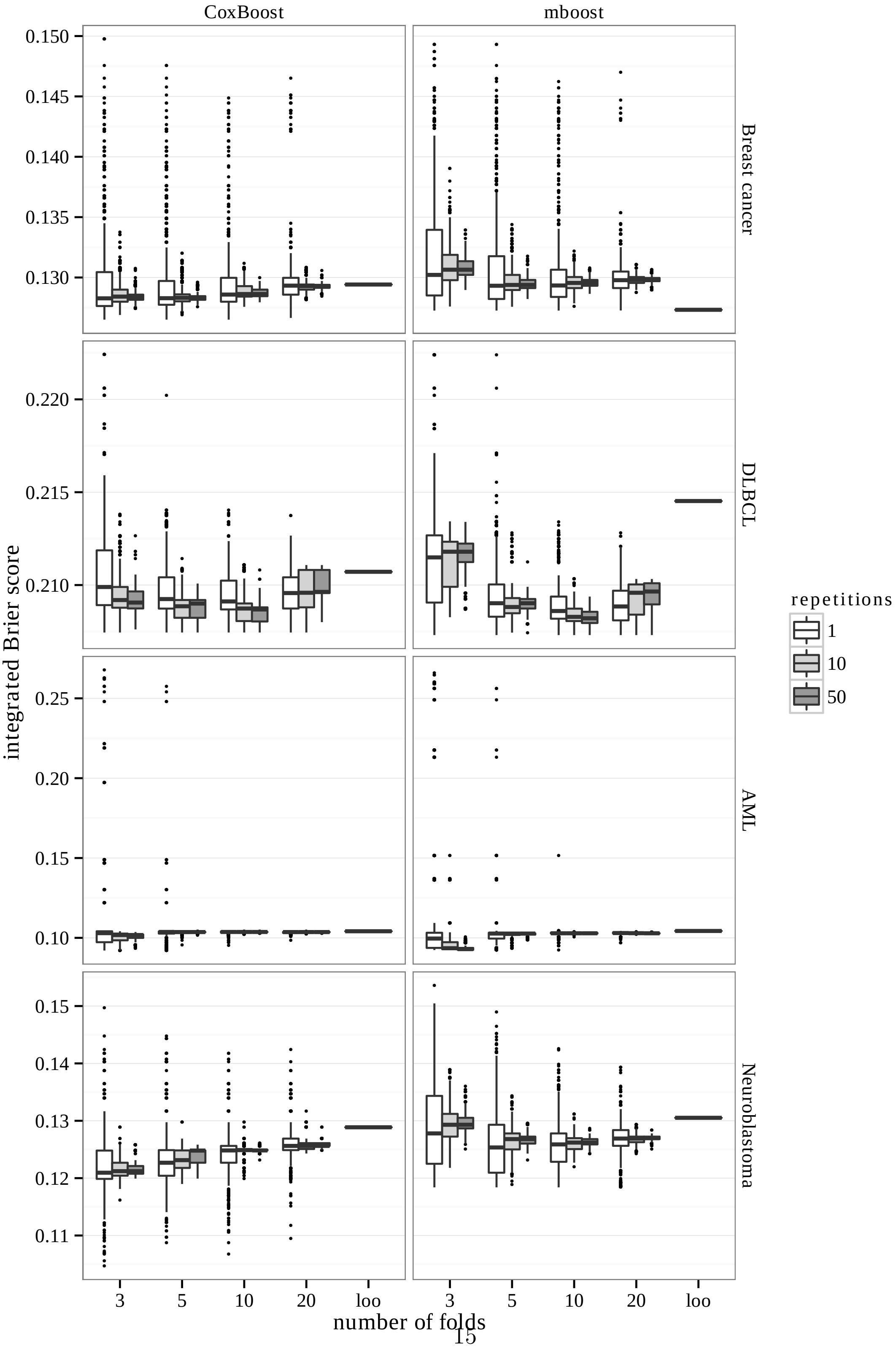
When we are interested in explanatory models, knowledge of the selected predictors and the stability of the resulting model among several repetitions of the same procedure is particularly important. This is not, however, the main focus of boosting: the boosting approach is mainly used in the context of prediction models, where the focus is more on the goodness of the prediction than on the model itself. For example, if we have two strongly correlated predictors, from a predictive point of view it is equivalent to include the former, the latter, or both with two coefficients that combine their effects. For this reason, here we investigate the effect of the randomness of the cross-validation-based choice of m_{stop} on the prediction ability, analyzing the differences in the estimates of the integrated Brier score among the resultant models. We report in the white boxplots of Figure 5 the results for *CoxBoost* (left) and for *mboost* (right) using 3- 5- 10- 20-fold and leave-one-out cross-validation. The results are based on 2000 iterations, except for the leave-one-out cross-validation, for which, obviously, only one value is provided.

As a consequence of the decrease in the variability of m_{stop}, and the relative decrease in the variability in terms of selected predictors, the variability of the integrated Brier score decreases with an increase in the number of cross-validation folds. We note a peculiar behavior in the acute myeloid leukemia example: despite it having the lowest variability in terms of m_{stop}, it shows a high variability in terms of integrated Brier score, with several cases of extremely high values (visualized by the outlier-points in the box-plots of Figure 5). Strongly unexpected, leave-one-out cross-validation leads to good results for *mboost* on the breast cancer data set. For some unknown reasons in this case the more complex model is the better model. This does not happen often, and may be a particularity of this data set, in which weak effect predictors may result relevant. Note that this result may explain why in the original study a complex gene-signature (up to 73 probe-sets) leads to good results, which have not been obtained when focusing on sparse models (see, e.g. De Bin et al, 2014). Please note that, in general, the inclusion of this kind of predictors decreases the model portability (the model is too specific for the learning data). In this sense, it is not surprising that this result has been obtained by using leave-one-out cross-validation, which is known to favor data-specific models. In all other cases, indeed, the integrated Brier score from leave-one-out cross-validation is higher than the median of the integrated Brier score from other folds, including *CoxBoost* on the breast cancer data set. In these cases, leave-one-out cross-validation seems not to be able to overcome the tendency for sparsity of boosting.

4 Effect of repeated cross-validation

In the previous section we saw that the randomness of the folds split in the cross-validation procedure causes variation in the results and the prediction ability. From a theoretical point of view, to avoid this problem we should consider all the combinations of the n observations in K folds, following the theory of complete cross-validation (Kohavi, 1995), and transform the

Figure 5: Integrated Brier score for models computed using different CV folds and a different number of repetitions in the four data sets, for both *CoxBoost* (left) and *mboost* (right).



estimator of m_{stop} based on the cross-validated likelihood into a complete U-statistic. With the usual sample size of a medical study, this is clearly computationally unfeasible (see also Fuchs et al, 2013). Between the current case of only one split and the theoretical case of all splits, nonetheless, there are several intermediate cases, in which we can obtain a more stable result in an acceptable amount of time. For this reason, we suggest the use of a repeated cross-validation procedure for the choice of the tuning parameter: instead of considering the cross-validated partial log-likelihood, one should consider a repeated cross-validated partial log-likelihood,

$$rcvpl(m) = \sum_{(D_1,\ldots,D_K)\in\mathcal{D}_I} \sum_{k=1}^{K} \left(pl(\hat{\boldsymbol{\beta}}_m^{(-D_k)}) - pl^{(-D_k)}(\hat{\boldsymbol{\beta}}_m^{(-D_k)}) \right),$$

where $\mathcal{D}_I$ denotes the random set of the I splits into K subsets of the sample D considered in our analysis.

Again, the optimal value of m_{stop} is computed by maximizing the function over m.

4.1 Study design

The repeated cross-validated likelihood should be based on the maximum feasible number of different splits, i.e. the largest I that is within the constraints of reasonable calculation time. In our study, involving 2000 replications of 4 kinds of cross-validation, we consider $I = 10$ as well as $I = 50$. Obviously, when the goal is to fit a prediction model based on a specific sample, a larger number can be considered.

The data sets and the methods used in this section are the same as Section 3. Leave-one-out cross-validation is not considered again because the results do not change. We fit a prediction model using the tuning parameter computed in a 3-, 5-, 10- and 20-fold cross-validation procedure and we consider the selected predictors and the prediction ability in terms of integrated Brier score. The procedure is repeated 2000 times.

4.2 Results

4.2.1 Number of boosting steps

Figure 1 shows the improvements in stability in the choice of the optimal number of boosting steps using the repeated cross-validated log-likelihood: if we compare the results of repeated cross-validation in gray and normal cross-validation in white, we note a pronounced decrease in the variability, both in terms of interquartile and total range. The decrease between normal cross-validation and the 10 times repeated cross-validation is greater than the decrease between 10 and 50 repetitions. The medians of the distributions are almost equal with a light tendency of being lower when computed with the repeated cross-validated log-likelihood. The reason probably lies in the avoidance of the highest values that characterized the distributions in the original cross-validation procedure. The absence of the extreme values (especially those on the borders, namely 0 and 200), in particular, is the most positive improvement obtained by implementing the repeated cross-validation, because it prevents situations in which m_{stop} is chosen incorrectly due to a particularly unfortunate partition of the observations.

4.2.2 Selected predictors

The superiority of a more stable choice for the optimal number of boosting steps is clear when examining selected predictors (Figures 2 and 3). Avoiding underestimation and overestimation of m_{stop}, indeed, leads to the identification of a clear group of relevant predictors always selected in our 2000 replications, and to the decrease of the rarely selected predictors: the latter property is particularly evident in the acute myeloid leukemia example, in which the maximum number of selected predictors is 22 when using 10 repetitions and 19 with 50 repetitions. We note that with 50 replications we are relatively close to a deterministic result, i.e. the inclusion frequencies of the predictors is mostly 2000 (always) or 0 (never). The complete information on which predictors were selected is shown in Tables 2 – 13 in the supplementary material.

4.2.3 Prediction ability

The analysis of the integrated Brier score also reflects the advantages of using a repeated cross-validated log-likelihood for the choice of m_{stop}. As can be seen in Figure 5, the avoidance of extreme values for the tuning parameter results in the disappearance of the worst prediction performances obtained with the simple cross-validated log-likelihood. For the acute myeloid leukemia example for both *mboost* and *CoxBoost* the bad predictions experienced in the previous section do not occurs. The improvement between 10 and 50 repetitions of cross-validation is not as striking as between none and 10 repetitions but with 50 repetitions we come even closer to a stable result, especially for 3-fold cross-validation.

5 Conclusions

Boosting techniques have proved to be useful tools in selecting a prediction model, especially in the important case in which the number of predictors is much higher than the number of observations. One weakness of boosting is the strong dependence on the tuning parameter m_{stop}, namely the number of boosting steps. Please note that several statistical methods share this weakness. To now, there has not been a convincing theory developed on the choice of this parameter and practitioners are compelled to use a cross-validation procedure. We have seen that this solution is sub-optimal, since it may lead to surprisingly different results in terms of selected predictors and prediction ability of the model depending on the particular partition of the observations into the cross-validation folds. A particularly unfortunate split may cause a severe underestimation or overestimation of the optimal value of boosting steps, with the consequence that the boosting algorithm may produce a very misleading model. We have seen that this problem affects the cross-validation procedure irrespectively of the number of folds used. In our study, we showed that the implementation of a repeated cross-validation procedure decreases the variability in the choice of the tuning parameter and produces a more robust result: as a consequence, far fewer extreme values of m_{stop} would be expected. The results of the 10-replication cross-validated partial log-likelihood suggest that few replications are sufficient to greatly improve the selection of the best tuning parameter. The extension to 50 replications shows that increasing the number of replications may lead to even better results. As often happens, however, there is no free-lunch solution, and an increase in replications also results in a large increase in the number of computations to perform. Therefore, the trade-off between variability reduction and computational time plays an important role in the choice of the number of replications. In our opinion 10 (or only a few more, let us say 15 or 20) replications may be sufficient to avoid extreme cases and, consequently, obtain reliable results. Nevertheless, we note that the advances in computational techniques (e.g., parallel computing) and computational power (better hardware) constantly relax the computational time issues, and in the future more replications may be implemented without noticeable drawbacks.

Acknowledgements We thank Rory Wilson for language improvements. HS and RDB were supported by grants BO3139/4-1, BO3139/4-2 and BO3139/2-2 to ALB from the German Research Foundation (DFG).

References

Binder H (2013) CoxBoost: Cox models by likelihood based boosting for a single survival endpoint or competing risks. URL http://CRAN.R-project.org/package=CoxBoost, R package version 1.4

Binder H, Schumacher M (2008) Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. BMC Bioinformatics 9:14

Boulesteix AL, Sauerbrei W (2011) Added predictive value of high-throughput molecular data to clinical data and its validation. Briefings in Bioinformatics 12:215–229

Boulesteix AL, Richter A, Bernau C (2013) Complexity selection with cross-validation for lasso and sparse partial least squares using high-dimensional data. In Algorithms from and for Nature and Life, pp. 261–268. Springer.

Bøvelstad H, Nygård S, Borgan Ø (2009) Survival prediction from clinico-genomic models - a comparative study. BMC Bioinformatics 10:413

Brier GW (1950) Verification of forecasts expressed in terms of probability. Monthly Weather Review 78:1–3

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

Bühlmann P, Hothorn T (2007) Boosting algorithms: Regularization, prediction and model fitting. Statistical Science 22:477–505

Bühlmann P, Yu B (2003) Boosting with the L_2 loss: regression and classification. Journal of the American Statistical Association 98:324–339

Chang YCI, Huang Y, Huang YP (2010) Early stopping in $l_2$boosting. Computational Statistics & Data Analysis 54:2203–2213

De Bin R, Sauerbrei W, Boulesteix AL (2014) Investigating the prediction ability of survival models based on both clinical and omics data: two case studies. Statistics in Medicine 33:5310–5329

De Bin R (2015) Boosting in Cox regression: a comparison between the likelihood-based and the model-based approaches with focus on the R-packages CoxBoost and mboost. Computational Statistics DOI:10.1007/s00180-015-0642-2

Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. The Annals of Statistics 32:407–499

Freund Y, Schapire R (1996) Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., pp 148–156

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. The Annals of Statistics 29:1189–1232

Fuchs M, Hornung R, De Bin R, Boulesteix, A L (2013) A U-statistic estimator for the variance of resampling-based error estimators. Technical Report 148, University of Munich

Gerds TA, Schumacher M (2006) Consistent estimation of the expected brier score in general survival models with right-censored event times. Biometrical Journal 48(6):1029–1040

Graf E, Schmoor C, Sauerbrei W, Schumacher M (1999) Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine 18:2529–2545

Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, New York

Hatzis C, Pusztai L, Valero V, Booser DJ, Esserman L, Lluch A, Vidaurre T, Holmes F, Souchon E, Wang H, et al (2011) A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer. JAMA: the journal of the American Medical Association 305(18):1873

Hofner B, Mayr A, Robinzonov N, Schmid M (2014) Model-based boosting in R: a hands-on tutorial using the R package mboost. Computational Statistics 29:3–35

Hothorn T, Bühlmann P, Dudoit S, Molinaro A, Van Der Laan MJ (2006) Survival ensembles. Biostatistics 7:355–373

Hothorn T, Buehlmann P, Kneib T, Schmid M, Hofner B (2015) mboost: Model-Based Boosting. URL http://CRAN.R-project.org/package=mboost, R package version 2.4-2

Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of International Joint Conference on Artificial Intelligence, pp 1137–1145

Mayr A, Hofner B, Schmid M (2012) The importance of knowing when to stop. a sequential stopping rule for component-wise gradient boosting. Methods of Information in Medicine 51:178–186

Metzeler KH, Hummel M, Bloomfield CD, Spiekermann K, Braess J, Sauerland MC, Heinecke A, Radmacher M, Marcucci G, Whitman SP, et al (2008) An 86-probe-set gene-expression signature predicts survival in cytogenetically normal acute myeloid leukemia. Blood 112(10):4193–4201

Mogensen UB, Ishwaran H, Gerds TA, Universitet K, Afdeling B (2010) Evaluating random forests for survival analysis using prediction error curves. Department of Biostatistics, University of Copenhagen

Oberthuer A, Kaderali L, Kahlert Y, Hero B, Westermann F, Berthold F, Brors B, Eils R, Fischer M (2008) Subclassification and individual survival time prediction from gene expression data of neuroblastoma patients by using caspar. Clinical Cancer Research 14(20):6590–6601

Ridgeway G (1999) Generalization of boosting algorithms and applications of Bayesian inference for massive datasets. PhD thesis, University of Washington

Rosenwald A, Wright G, Chan WC, Connors JM, Campo E, Fisher RI, Gascoyne RD, Muller-Hermelink HK, Smeland EB, Giltnane JM, et al (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. New England Journal of Medicine 346(25):1937–1947

Schmid M, Hothorn T (2008) Flexible boosting of accelerated failure time models. BMC Bioinformatics 9:269

Schumacher M, Binder H, Gerds T (2007) Assessment of survival prediction models based on microarray data. Bioinformatics 23:1768–1774

Tutz G, Binder H (2006) Generalized additive modeling with implicit variable selection by likelihood-based boosting. Biometrics 62:961–971

Verweij PJ, Van Houwelingen HC (1993) Cross-validation in survival analysis. Statistics in Medicine 12:2305–2314

Supplementary material

Table 2: Number of times each predictor of the Breast cancer data set was selected within 2000 iterations (except leave-one-out CV) using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20	loo
x221874_at	1995	2000	2000	2000	1
x205428_s_at	1994	1998	2000	2000	1
x211110_s_at	1993	1997	2000	1999	1
x212811_x_at	1989	1995	1999	1996	1
x203860_at	1974	1988	1980	1977	1
x201928_at	1965	1979	1974	1969	1
x221681_s_at	1957	1973	1963	1969	1
x217769_s_at	1948	1966	1957	1969	1
x220298_s_at	1925	1941	1955	1969	1
x214778_at	1890	1902	1937	1969	1
x219648_at	1879	1894	1935	1969	1
x207417_s_at	1774	1796	1888	1964	1
x209383_at	1737	1767	1852	1955	1
x202145_at	1725	1754	1839	1948	1
x210254_at	1550	1621	1659	1754	1
x212531_at	1540	1611	1644	1723	1
x207639_at	1316	1358	1217	893	0
x207750_at	1164	1138	872	363	0
x218650_at	1150	1126	845	331	0
x200927_s_at	1130	1108	825	307	0
x203208_s_at	1116	1089	801	287	0
x210820_x_at	1059	1022	685	216	0
x214952_at	1002	962	611	172	0
x214465_at	918	857	533	141	0
x217505_at	845	782	464	130	0
x204527_at	761	683	380	113	0
x201932_at	734	659	362	112	0
x205476_at	707	627	345	110	0
x203889_at	698	618	339	110	0
x222009_at	647	566	315	106	0
x218701_at	621	533	296	105	0
x203576_at	603	509	289	105	0
x201097_s_at	563	468	276	104	0
x217566_s_at	516	426	238	100	0
x215369_at	509	420	237	98	0
x209149_s_at	503	406	230	98	0
x206847_s_at	470	383	222	94	0
x210648_x_at	351	282	153	85	0
x217944_at	333	255	144	85	0
x207680_x_at	329	249	141	85	0
x214191_at	281	205	116	79	0
x214386_at	241	183	96	69	0
x208603_s_at	214	162	81	60	0
x203892_at	208	160	76	59	0
x220067_at	183	139	68	47	0

(b) mboost

	3	5	10	20	loo
x221874_at	2000	2000	2000	2000	1
x205428_s_at	1986	1994	2000	1999	1
x211110_s_at	1986	1993	1999	1999	1
x212811_x_at	1979	1990	1996	1999	1
x219648_at	1959	1969	1976	1993	1
x203860_at	1946	1964	1969	1992	1
x201928_at	1934	1949	1967	1992	1
x221681_s_at	1917	1941	1960	1992	1
x217769_s_at	1908	1934	1959	1992	1
x220298_s_at	1833	1900	1946	1992	1
x214778_at	1723	1852	1934	1992	1
x207417_s_at	1502	1713	1877	1986	1
x209383_at	1467	1686	1862	1983	1
x202145_at	1457	1669	1852	1980	1
x210254_at	1180	1445	1622	1808	1
x212531_at	1098	1334	1487	1593	1
x207639_at	833	1040	1037	754	1
x200927_s_at	718	924	861	490	1
x218650_at	708	912	843	447	1
x207750_at	690	892	813	417	1
x203208_s_at	621	833	747	318	1
x203576_at	574	772	683	269	1
x214465_at	530	709	617	218	1
x214952_at	463	623	527	191	1
x210820_x_at	455	609	515	190	1
x217505_at	403	530	449	179	1
x204527_at	373	485	428	176	1
x205476_at	358	466	411	176	1
x218701_at	333	426	387	172	1
x201932_at	329	417	385	172	1
x203889_at	294	371	361	168	1
x210648_x_at	268	339	353	166	1
x222009_at	253	321	345	163	1
x217566_s_at	241	311	333	163	1
x201097_s_at	187	267	300	162	1
x206847_s_at	186	259	297	160	1
x209149_s_at	174	230	278	160	1
x215369_at	152	208	263	160	1
x207680_x_at	38	60	86	79	1

Table 3: Number of times each predictor of the DLBCL data set was selected within 2000 iterations (except leave-one-out CV) using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20	loo
V1685	1993	1998	2000	2000	1
V1829	1986	1998	2000	2000	1
V3836	1950	1997	2000	2000	1
V1192	1944	1997	2000	2000	1
V3176	1927	1995	2000	2000	1
V5031	1916	1994	2000	2000	1
V7361	1909	1994	2000	2000	1
V3805	1895	1991	2000	2000	1
V1680	1834	1979	2000	2000	1
V2583	1773	1957	1997	2000	1
V1460	1740	1951	1996	2000	1
V2906	1668	1924	1990	2000	1
V6611	1616	1894	1979	1998	1
V6960	1512	1833	1966	1993	1
V777	1135	1574	1810	1938	1
V1988	1080	1528	1776	1930	1
V1302	854	1294	1635	1875	1
V4885	824	1268	1589	1864	1
V4130	635	991	1369	1764	1
V1101	603	917	1315	1737	1
V6370	577	884	1280	1719	1
V704	529	788	1203	1642	1
V97	504	761	1173	1611	1
V5734	356	548	854	1353	1
V4261	302	448	713	1166	1
V4481	190	264	363	561	1
V5837	160	219	278	404	1
V5701	129	199	200	276	1
V3582	97	131	86	92	1
V34	67	73	35	23	1
V5836	58	61	27	8	1
V2441	49	51	19	4	0
V21	33	32	5	1	0
V5700	17	17	3	0	0
V6456	16	14	3	0	0
V4795	11	10	1	0	0
V6989	11	10	1	0	0
V3458	8	4	1	0	0
V6391	8	4	1	0	0
V1010	8	1	0	0	0
V4328	8	1	0	0	0
V4723	8	1	0	0	0
V5984	8	1	0	0	0
V6686	8	1	0	0	0
V247	5	0	0	0	0

(b) mboost

	3	5	10	20	loo
V1685	2000	2000	2000	2000	1
V1829	1983	1999	2000	2000	1
V3836	1951	1993	2000	2000	1
V1192	1944	1992	2000	2000	1
V3176	1937	1991	2000	2000	1
V5031	1924	1991	2000	2000	1
V7361	1881	1986	2000	2000	1
V3805	1857	1986	2000	2000	1
V3826	1754	1970	1998	2000	1
V1680	1724	1964	1998	2000	1
V2583	1557	1921	1998	2000	1
V1460	1411	1879	1996	2000	1
V2906	1236	1796	1989	1999	1
V6611	1163	1752	1971	1999	1
V6960	961	1626	1917	1994	1
V777	507	1194	1724	1947	1
V1988	483	1156	1698	1943	1
V1302	363	942	1555	1908	1
V4885	317	814	1457	1883	1
V1101	254	693	1379	1855	1
V97	194	564	1245	1786	1
V704	132	387	990	1668	1
V6370	126	364	961	1638	1
V4130	95	296	833	1541	1
V5734	57	212	591	1350	1
V4261	49	159	458	1122	1
V3582	33	120	335	908	1
V4481	19	68	144	444	1
V5701	13	45	87	269	1
V5837	12	39	81	230	1
V5836	2	15	9	13	1
V5700	2	8	2	0	1
V21	2	5	2	0	1
V2441	2	5	2	0	1
V34	2	5	2	0	1
V6456	2	4	1	0	1
V6989	2	4	1	0	1
V1010	1	2	0	0	1
V3458	1	2	0	0	1
V4795	1	2	0	0	1
V247	0	0	0	0	1
V4328	0	0	0	0	1
V4723	0	0	0	0	1
V5984	0	0	0	0	1
V6391	0	0	0	0	1
V6686	0	0	0	0	1
V7282	0	0	0	0	1

Table 4: Number of times each predictor of the AML data set was selected within 2000 iterations (except leave-one-out CV) using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20	loo
x201540_at	2000	2000	2000	2000	1
x203373_at	2000	2000	2000	2000	1
x218086_at	2000	2000	2000	2000	1
x209386_at	1990	1998	2000	2000	1
x229715_at	1988	1996	2000	2000	1
x202685_s_at	1959	1996	2000	2000	1
x211626_x_at	1912	1989	2000	2000	1
x211597_s_at	1607	1915	2000	2000	1
x209856_x_at	1452	1874	2000	2000	1
x216794_at	1358	1834	1999	2000	1
x233612_at	1222	1755	1990	1999	1
x243809_at	1175	1722	1982	1999	1
x224710_at	1046	1633	1946	1992	1
x239099_at	915	1517	1843	1898	1
x208049_s_at	876	1478	1797	1854	1
x210584_s_at	608	1086	1187	1013	1
x233089_at	510	915	837	605	1
x232996_at	421	686	448	199	0
x223757_at	399	623	373	154	0
x209794_at	372	576	323	112	0
x237875_at	257	353	100	11	0
x213416_at	206	250	42	0	0
x216620_s_at	169	186	21	0	0
x206237_s_at	153	174	13	0	0
x217201_at	132	114	9	0	0
x205266_at	119	101	8	0	0
x227326_at	107	67	2	0	0
x41469_at	83	40	0	0	0
x207582_at	69	33	0	0	0
x228838_at	53	23	0	0	0
x239111_at	30	8	0	0	0
x224822_at	20	1	0	0	0
x218412_s_at	4	0	0	0	0
x218812_s_at	4	0	0	0	0
x223410_s_at	4	0	0	0	0
x226612_at	4	0	0	0	0
x234358_at	4	0	0	0	0
x243660_at	4	0	0	0	0
x235488_at	3	0	0	0	0
x219143_s_at	1	0	0	0	0
x224498_x_at	1	0	0	0	0
x228860_at	1	0	0	0	0
x240437_at	1	0	0	0	0

(b) mboost

	3	5	10	20	loo
x201540_at	2000	2000	2000	2000	1
x218086_at	2000	2000	2000	2000	1
x203373_at	1998	2000	2000	2000	1
x209386_at	1965	1998	2000	2000	1
x229715_at	1947	1997	2000	2000	1
x202685_s_at	1866	1994	2000	2000	1
x211626_x_at	1637	1957	1999	2000	1
x211597_s_at	1333	1885	1999	2000	1
x243809_at	812	1666	1993	2000	1
x224710_at	741	1602	1984	1998	1
x216794_at	670	1551	1978	1998	1
x233612_at	494	1326	1887	1977	1
x239099_at	272	814	1338	1631	1
x208049_s_at	237	740	1193	1507	1
x232996_at	215	664	1065	1361	1
x210584_s_at	163	513	831	1032	1
x209856_x_at	117	357	504	614	1
x233089_at	90	279	367	383	1
x223757_at	75	239	295	294	1
x209794_at	59	173	193	170	1
x237875_at	25	67	63	33	1
x213416_at	20	41	37	13	1
x217201_at	14	21	19	2	1
x206237_s_at	8	16	11	1	1
x216620_s_at	7	13	9	0	1
x227326_at	7	11	5	0	1
x41469_at	4	4	1	0	1
x205266_at	4	3	0	0	1
x207582_at	2	1	0	0	1
x228838_at	1	1	0	0	1
x239111_at	1	0	0	0	1
x223410_s_at	0	0	0	0	1
x224498_x_at	0	0	0	0	1
x234358_at	0	0	0	0	1

Table 5: Number of times each predictor of the Neuroblastoma data set was selected within 2000 iterations (except leave-one-out CV) using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20	loo
V2192	2000	2000	2000	2000	1
V3384	2000	2000	2000	2000	1
V3463	2000	2000	2000	2000	1
V3595	2000	2000	2000	2000	1
V403	2000	2000	2000	2000	1
V676	2000	2000	2000	2000	1
V6816	2000	2000	2000	2000	1
V7636	2000	2000	2000	2000	1
V986	2000	2000	2000	2000	1
V7718	1999	2000	2000	2000	1
V7514	1998	1999	2000	2000	1
V9663	1998	1996	2000	2000	1
V2178	1997	1996	2000	1999	1
V5368	1977	1973	1992	1995	1
V2479	1962	1952	1965	1978	1
V5070	1954	1933	1952	1956	1
V25	1938	1912	1922	1913	1
V5323	1924	1898	1895	1846	1
V7976	1823	1734	1624	1179	0
V6832	1716	1587	1257	554	0
V1633	1667	1519	1138	396	0
V8049	1626	1445	1034	301	0
V269	1468	1215	742	129	0
V7901	1250	917	448	57	0
V1542	1141	799	343	46	0
V6277	1028	682	273	38	0
V3926	959	639	245	36	0
V1976	910	590	219	35	0
V4226	463	265	90	9	0
V6822	436	244	78	8	0
V553	325	171	61	7	0
V2453	231	105	43	6	0
V4138	199	90	41	6	0
V390	90	38	25	3	0
V2326	80	30	25	2	0
V3243	66	27	20	2	0
V7863	56	24	18	2	0
V2758	37	16	15	2	0
V9221	32	15	12	2	0
V996	14	4	2	1	0
V3247	9	1	2	0	0
V3362	7	0	1	0	0
V6756	7	0	1	0	0
V4770	6	0	0	0	0
V8501	6	0	0	0	0
V1108	5	0	0	0	0
V2221	5	0	0	0	0
V3009	5	0	0	0	0
V6115	5	0	0	0	0
V380	3	0	0	0	0
V7697	3	0	0	0	0

(b) mboost

	3	5	10	20	loo
V2192	2000	2000	2000	2000	1
V3384	2000	2000	2000	2000	1
V3463	2000	2000	2000	2000	1
V3595	2000	2000	2000	2000	1
V403	2000	2000	2000	2000	1
V6816	2000	2000	2000	2000	1
V7636	2000	2000	2000	2000	1
V986	2000	2000	2000	2000	1
V676	1999	2000	2000	2000	1
V7718	1980	1999	2000	2000	1
V7514	1958	1996	2000	2000	1
V9663	1954	1996	2000	2000	1
V2178	1914	1988	1999	2000	1
V2479	1776	1940	1995	1997	1
V25	1733	1917	1992	1996	1
V5368	1673	1890	1987	1996	1
V5070	1486	1805	1959	1987	1
V6832	1137	1526	1720	1779	0
V7976	1097	1455	1637	1685	0
V5323	1016	1380	1528	1483	0
V8049	659	889	759	285	0
V1633	572	725	555	114	0
V269	514	665	481	87	0
V7901	378	465	306	39	0
V6277	307	349	227	28	0
V3926	267	304	185	26	0
V1976	250	290	163	25	0
V1542	193	233	129	20	0

Table 6: Number of times each predictor of the Breast cancer data set was selected within 2000 iterations with 10 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
x201928_at	2000	2000	2000	2000
x203860_at	2000	2000	2000	2000
x205428_s_at	2000	2000	2000	2000
x211110_s_at	2000	2000	2000	2000
x212811_x_at	2000	2000	2000	2000
x214778_at	2000	2000	2000	2000
x217769_s_at	2000	2000	2000	2000
x219648_at	2000	2000	2000	2000
x220298_s_at	2000	2000	2000	2000
x221681_s_at	2000	2000	2000	2000
x221874_at	2000	2000	2000	2000
x207417_s_at	1999	2000	2000	2000
x209383_at	1995	2000	2000	2000
x202145_at	1992	2000	2000	2000
x210254_at	1932	1966	1985	1993
x212531_at	1921	1963	1982	1988
x207639_at	1575	1716	1439	644
x207750_at	1178	1291	526	9
x218650_at	1138	1244	464	5
x200927_s_at	1091	1186	402	2
x203208_s_at	1033	1132	351	1
x210820_x_at	852	914	195	0
x214952_at	696	720	96	0
x214465_at	481	473	35	0
x217505_at	386	338	19	0
x204527_at	231	180	5	0
x201932_at	196	146	1	0
x205476_at	169	117	1	0
x203889_at	153	108	1	0
x222009_at	115	68	0	0
x218701_at	104	56	0	0
x203576_at	80	49	0	0
x201097_s_at	59	36	0	0
x217566_s_at	39	17	0	0
x215369_at	35	16	0	0
x209149_s_at	34	13	0	0
x206847_s_at	23	9	0	0
x210648_x_at	2	0	0	0
x217944_at	2	0	0	0
x207680_x_at	1	0	0	0
x214191_at	1	0	0	0
x214386_at	1	0	0	0

(b) mboost

	3	5	10	20
x201928_at	2000	2000	2000	2000
x203860_at	2000	2000	2000	2000
x205428_s_at	2000	2000	2000	2000
x211110_s_at	2000	2000	2000	2000
x212811_x_at	2000	2000	2000	2000
x217769_s_at	2000	2000	2000	2000
x219648_at	2000	2000	2000	2000
x221681_s_at	2000	2000	2000	2000
x221874_at	2000	2000	2000	2000
x220298_s_at	1998	2000	2000	2000
x214778_at	1996	2000	2000	2000
x207417_s_at	1907	1993	2000	2000
x209383_at	1879	1990	2000	2000
x202145_at	1848	1988	2000	2000
x210254_at	1277	1844	1984	2000
x212531_at	992	1689	1888	1971
x207639_at	349	999	904	332
x200927_s_at	192	676	466	34
x218650_at	168	620	411	23
x207750_at	152	585	354	14
x203208_s_at	90	445	230	2
x203576_at	70	345	127	0
x214465_at	39	218	68	0
x214952_at	15	118	29	0
x210820_x_at	12	109	25	0
x217505_at	6	46	13	0
x204527_at	4	33	11	0
x205476_at	4	26	8	0
x218701_at	3	19	7	0
x201932_at	3	15	5	0
x203889_at	2	6	2	0
x210648_x_at	2	4	1	0
x222009_at	2	2	1	0
x217566_s_at	1	2	1	0
x201097_s_at	0	0	1	0
x206847_s_at	0	0	1	0
x209149_s_at	0	0	1	0
x215369_at	0	0	1	0

Table 7: Number of times each predictor of the DLBCL data set was selected within 2000 iterations with 10 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
V1192	2000	2000	2000	2000
V1680	2000	2000	2000	2000
V1685	2000	2000	2000	2000
V1829	2000	2000	2000	2000
V2583	2000	2000	2000	2000
V3176	2000	2000	2000	2000
V3805	2000	2000	2000	2000
V3836	2000	2000	2000	2000
V5031	2000	2000	2000	2000
V7361	2000	2000	2000	2000
V1460	1997	2000	2000	2000
V2906	1994	2000	2000	2000
V6611	1981	2000	2000	2000
V6960	1931	2000	2000	2000
V777	1374	1931	1998	2000
V1988	1188	1880	1997	2000
V1302	580	1501	1957	2000
V4885	515	1427	1948	2000
V4130	158	725	1727	1999
V1101	113	561	1613	1998
V6370	98	490	1539	1994
V704	50	307	1301	1982
V97	42	255	1193	1973
V5734	11	50	496	1701
V4261	3	20	208	1341
V4481	0	2	6	131
V5837	0	0	5	34
V5701	0	0	1	9

(b) mboost

	3	5	10	20
V1192	2000	2000	2000	2000
V1685	2000	2000	2000	2000
V1829	2000	2000	2000	2000
V3176	2000	2000	2000	2000
V3805	2000	2000	2000	2000
V3836	2000	2000	2000	2000
V5031	2000	2000	2000	2000
V7361	2000	2000	2000	2000
V3826	1999	2000	2000	2000
V1680	1996	2000	2000	2000
V2583	1938	2000	2000	2000
V1460	1810	2000	2000	2000
V2906	1525	1998	2000	2000
V6611	1309	1994	2000	2000
V6960	785	1957	2000	2000
V777	54	1253	1991	2000
V1988	40	1158	1988	2000
V1302	6	652	1933	2000
V4885	4	361	1860	2000
V1101	1	226	1726	2000
V97	0	81	1456	2000
V704	0	8	908	1989
V6370	0	5	792	1985
V4130	0	1	508	1953
V5734	0	0	160	1736
V4261	0	0	51	1373
V3582	0	0	7	832
V4481	0	0	0	63
V5701	0	0	0	7
V5837	0	0	0	4

Table 8: Number of times each predictor of the AML data set was selected within 2000 iterations with 10 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
x201540_at	2000	2000	2000	2000
x202685_s_at	2000	2000	2000	2000
x203373_at	2000	2000	2000	2000
x209386_at	2000	2000	2000	2000
x211626_x_at	2000	2000	2000	2000
x218086_at	2000	2000	2000	2000
x229715_at	2000	2000	2000	2000
x211597_s_at	1957	2000	2000	2000
x209856_x_at	1828	2000	2000	2000
x216794_at	1704	1999	2000	2000
x233612_at	1406	1998	2000	2000
x243809_at	1282	1993	2000	2000
x224710_at	959	1982	2000	2000
x239099_at	631	1923	1996	2000
x208049_s_at	530	1885	1994	2000
x210584_s_at	115	979	1233	907
x233089_at	57	557	537	177
x232996_at	11	184	71	3
x223757_at	7	143	33	0
x209794_at	5	87	15	0
x237875_at	0	7	0	0
x213416_at	0	1	0	0

(b) mboost

	3	5	10	20
x201540_at	2000	2000	2000	2000
x202685_s_at	2000	2000	2000	2000
x203373_at	2000	2000	2000	2000
x209386_at	2000	2000	2000	2000
x218086_at	2000	2000	2000	2000
x229715_at	2000	2000	2000	2000
x211626_x_at	1946	2000	2000	2000
x211597_s_at	1632	2000	2000	2000
x243809_at	420	1973	2000	2000
x224710_at	317	1941	2000	2000
x216794_at	228	1900	2000	2000
x233612_at	57	1584	1999	2000
x239099_at	2	473	1608	1942
x208049_s_at	2	311	1394	1826
x232996_at	2	209	1125	1615
x210584_s_at	2	76	592	1014
x209856_x_at	2	14	154	246
x233089_at	0	3	27	54
x223757_at	0	0	7	18
x209794_at	0	0	1	0

Table 9: Number of times each predictor of the Neuroblastoma data set was selected within 2000 iterations with 10 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
V2178	2000	2000	2000	2000
V2192	2000	2000	2000	2000
V2479	2000	2000	2000	2000
V25	2000	2000	2000	2000
V3384	2000	2000	2000	2000
V3463	2000	2000	2000	2000
V3595	2000	2000	2000	2000
V403	2000	2000	2000	2000
V5070	2000	2000	2000	2000
V5368	2000	2000	2000	2000
V676	2000	2000	2000	2000
V6816	2000	2000	2000	2000
V7514	2000	2000	2000	2000
V7636	2000	2000	2000	2000
V7718	2000	2000	2000	2000
V9663	2000	2000	2000	2000
V986	2000	2000	2000	2000
V5323	2000	2000	2000	1999
V7976	1994	1988	1912	1229
V6832	1967	1905	1304	83
V1633	1946	1841	1017	29
V8049	1917	1716	713	6
V269	1742	1214	149	0
V7901	1251	476	13	0
V1542	939	254	3	0
V6277	636	127	0	0
V3926	507	92	0	0
V1976	395	55	0	0
V4226	23	1	0	0
V6822	19	1	0	0
V553	2	0	0	0

(b) mboost

	3	5	10	20
V2178	2000	2000	2000	2000
V2192	2000	2000	2000	2000
V3384	2000	2000	2000	2000
V3463	2000	2000	2000	2000
V3595	2000	2000	2000	2000
V403	2000	2000	2000	2000
V676	2000	2000	2000	2000
V6816	2000	2000	2000	2000
V7514	2000	2000	2000	2000
V7636	2000	2000	2000	2000
V7718	2000	2000	2000	2000
V9663	2000	2000	2000	2000
V986	2000	2000	2000	2000
V2479	1999	2000	2000	2000
V25	1990	2000	2000	2000
V5368	1974	2000	2000	2000
V5070	1824	1998	2000	2000
V6832	1051	1830	1987	2000
V7976	922	1740	1973	1983
V5323	682	1576	1882	1888
V8049	86	354	223	2
V1633	19	116	19	0
V269	11	78	5	0
V7901	0	8	0	0
V1976	0	2	0	0
V3926	0	2	0	0
V6277	0	2	0	0
V1542	0	1	0	0

Table 10: Number of times each predictor of the Breast cancer data set was selected within 2000 iterations with 50 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
x201928_at	2000	2000	2000	2000
x202145_at	2000	2000	2000	2000
x203860_at	2000	2000	2000	2000
x205428_s_at	2000	2000	2000	2000
x207417_s_at	2000	2000	2000	2000
x209383_at	2000	2000	2000	2000
x211110_s_at	2000	2000	2000	2000
x212811_x_at	2000	2000	2000	2000
x214778_at	2000	2000	2000	2000
x217769_s_at	2000	2000	2000	2000
x219648_at	2000	2000	2000	2000
x220298_s_at	2000	2000	2000	2000
x221681_s_at	2000	2000	2000	2000
x221874_at	2000	2000	2000	2000
x210254_at	1999	2000	2000	2000
x212531_at	1999	2000	2000	2000
x207639_at	1923	1979	1707	391
x207750_at	1346	1531	205	0
x218650_at	1270	1431	144	0
x200927_s_at	1181	1345	87	0
x203208_s_at	1082	1245	46	0
x210820_x_at	741	821	8	0
x214952_at	428	470	0	0
x214465_at	173	147	0	0
x217505_at	81	71	0	0
x204527_at	14	5	0	0
x201932_at	8	3	0	0
x205476_at	4	3	0	0
x203889_at	2	3	0	0
x222009_at	1	2	0	0
x218701_at	1	0	0	0

(b) mboost

	3	5	10	20
x201928_at	2000	2000	2000	2000
x203860_at	2000	2000	2000	2000
x205428_s_at	2000	2000	2000	2000
x207417_s_at	2000	2000	2000	2000
x211110_s_at	2000	2000	2000	2000
x212811_x_at	2000	2000	2000	2000
x214778_at	2000	2000	2000	2000
x217769_s_at	2000	2000	2000	2000
x219648_at	2000	2000	2000	2000
x220298_s_at	2000	2000	2000	2000
x221681_s_at	2000	2000	2000	2000
x221874_at	2000	2000	2000	2000
x209383_at	1998	2000	2000	2000
x202145_at	1997	2000	2000	2000
x210254_at	1526	1996	2000	2000
x212531_at	932	1959	1999	2000
x207639_at	70	988	740	85
x200927_s_at	12	402	119	0
x218650_at	7	343	73	0
x207750_at	5	287	46	0
x203208_s_at	0	140	5	0
x203576_at	0	56	0	0
x214465_at	0	13	0	0
x204527_at	0	1	0	0
x205476_at	0	1	0	0
x210820_x_at	0	1	0	0
x214952_at	0	1	0	0
x217505_at	0	1	0	0

Table 11: Number of times each predictor of the DLBCL data set was selected within 2000 iterations with 50 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
V1192	2000	2000	2000	2000
V1460	2000	2000	2000	2000
V1680	2000	2000	2000	2000
V1685	2000	2000	2000	2000
V1829	2000	2000	2000	2000
V2583	2000	2000	2000	2000
V2906	2000	2000	2000	2000
V3176	2000	2000	2000	2000
V3805	2000	2000	2000	2000
V3836	2000	2000	2000	2000
V5031	2000	2000	2000	2000
V6611	2000	2000	2000	2000
V7361	2000	2000	2000	2000
V6960	1999	2000	2000	2000
V777	1597	1999	2000	2000
V1988	1342	1997	2000	2000
V1302	245	1794	2000	2000
V4885	188	1703	1999	2000
V4130	3	525	1966	2000
V1101	1	286	1887	2000
V6370	0	198	1824	2000
V704	0	50	1519	2000
V97	0	28	1371	2000
V5734	0	0	200	1957
V4261	0	0	20	1581
V4481	0	0	0	9

(b) mboost

	3	5	10	20
V1192	2000	2000	2000	2000
V1680	2000	2000	2000	2000
V1685	2000	2000	2000	2000
V1829	2000	2000	2000	2000
V3176	2000	2000	2000	2000
V3805	2000	2000	2000	2000
V3826	2000	2000	2000	2000
V3836	2000	2000	2000	2000
V5031	2000	2000	2000	2000
V7361	2000	2000	2000	2000
V2583	1999	2000	2000	2000
V1460	1990	2000	2000	2000
V2906	1813	2000	2000	2000
V6611	1548	2000	2000	2000
V6960	510	2000	2000	2000
V777	0	1400	2000	2000
V1988	0	1205	2000	2000
V1302	0	352	1999	2000
V4885	0	94	1998	2000
V1101	0	20	1981	2000
V97	0	1	1813	2000
V704	0	0	873	2000
V6370	0	0	683	2000
V4130	0	0	254	2000
V5734	0	0	15	1947
V4261	0	0	0	1577
V3582	0	0	0	776
V4481	0	0	0	2

Table 12: Number of times each predictor of the AML data set was selected within 2000 iterations with 50 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
x201540_at	2000	2000	2000	2000
x202685_s_at	2000	2000	2000	2000
x203373_at	2000	2000	2000	2000
x209386_at	2000	2000	2000	2000
x211597_s_at	2000	2000	2000	2000
x211626_x_at	2000	2000	2000	2000
x218086_at	2000	2000	2000	2000
x229715_at	2000	2000	2000	2000
x209856_x_at	1994	2000	2000	2000
x216794_at	1946	2000	2000	2000
x233612_at	1664	2000	2000	2000
x243809_at	1494	2000	2000	2000
x224710_at	845	2000	2000	2000
x239099_at	349	1998	2000	2000
x208049_s_at	235	1994	2000	2000
x210584_s_at	2	949	1357	872
x233089_at	0	248	274	35
x232996_at	0	8	0	0
x223757_at	0	3	0	0

(b) mboost

	3	5	10	20
x201540_at	2000	2000	2000	2000
x202685_s_at	2000	2000	2000	2000
x203373_at	2000	2000	2000	2000
x209386_at	2000	2000	2000	2000
x211626_x_at	2000	2000	2000	2000
x218086_at	2000	2000	2000	2000
x229715_at	2000	2000	2000	2000
x211597_s_at	1860	2000	2000	2000
x243809_at	125	2000	2000	2000
x224710_at	52	2000	2000	2000
x216794_at	17	2000	2000	2000
x233612_at	0	1874	2000	2000
x239099_at	0	219	1843	1998
x208049_s_at	0	91	1579	1979
x232996_at	0	36	1125	1858
x210584_s_at	0	3	328	1063
x209856_x_at	0	0	4	48

Table 13: Number of times each predictor of the Neuroblastoma data set was selected within 2000 iterations with 50 repetitions and using different CV folds with both methods CoxBoost and mboost. Predictors that were never selected are not shown.

(a) CoxBoost

	3	5	10	20
V2178	2000	2000	2000	2000
V2192	2000	2000	2000	2000
V2479	2000	2000	2000	2000
V25	2000	2000	2000	2000
V3384	2000	2000	2000	2000
V3463	2000	2000	2000	2000
V3595	2000	2000	2000	2000
V403	2000	2000	2000	2000
V5070	2000	2000	2000	2000
V5323	2000	2000	2000	2000
V5368	2000	2000	2000	2000
V676	2000	2000	2000	2000
V6816	2000	2000	2000	2000
V7514	2000	2000	2000	2000
V7636	2000	2000	2000	2000
V7718	2000	2000	2000	2000
V9663	2000	2000	2000	2000
V986	2000	2000	2000	2000
V7976	2000	2000	1997	1291
V6832	2000	1994	1425	4
V1633	2000	1979	918	0
V8049	1998	1917	432	0
V269	1941	1288	9	0
V7901	1316	158	0	0
V1542	715	29	0	0
V6277	265	1	0	0
V3926	163	0	0	0
V1976	78	0	0	0

(b) mboost

	3	5	10	20
V2178	2000	2000	2000	2000
V2192	2000	2000	2000	2000
V2479	2000	2000	2000	2000
V25	2000	2000	2000	2000
V3384	2000	2000	2000	2000
V3463	2000	2000	2000	2000
V3595	2000	2000	2000	2000
V403	2000	2000	2000	2000
V5368	2000	2000	2000	2000
V676	2000	2000	2000	2000
V6816	2000	2000	2000	2000
V7514	2000	2000	2000	2000
V7636	2000	2000	2000	2000
V7718	2000	2000	2000	2000
V9663	2000	2000	2000	2000
V986	2000	2000	2000	2000
V5070	1987	2000	2000	2000
V6832	1018	1987	2000	2000
V7976	718	1958	2000	2000
V5323	355	1821	1997	1998
V8049	0	33	20	0
V1633	0	1	0	0