On the choice and influence of the number of boosting steps for high-dimensional linear Cox-models

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

DOI: https://doi.org/10.1007/s00180-017-0773-8

Comput Stat
https://doi.org/10.1007/s00180-017-0773-8

ORIGINAL PAPER

On the choice and influence of the number of boosting steps for high-dimensional linear Cox-models

Heidi Seibold[1,2] · **Christoph Bernau**[3] · **Anne-Laure Boulesteix**[1] · **Riccardo De Bin**[1,4]

Received: 12 January 2016 / Accepted: 13 October 2017

Abstract In biomedical research, boosting-based regression approaches have gained much attention in the last decade. Their intrinsic variable selection procedure and 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

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00180-017-0773-8) contains supplementary material, which is available to authorized users.

✉ Heidi Seibold
heidi.seibold@uzh.ch

Christoph Bernau
Christoph.Bernau@lrz.de

Anne-Laure Boulesteix
boulesteix@ibe.med.uni-muenchen.de

Riccardo De Bin
debin@math.uio.no

1 Institute for Medical Information Processing, Biometry and Epidemiology, LMU Munich, Munich, Germany

2 Epidemiology, Biostatistics and Prevention Institute (EBPI), University of Zurich, Zurich, Switzerland

3 Leibniz Supercomputing Centre, Munich, Germany

4 Department of Mathematics, University of Oslo, Oslo, Norway

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

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 and the linearity of the effects of the predictors. We show that 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 (Mayr et al. 2014a). 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), 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 (2008a) 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 relationship 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

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

cross-validation-based procedures. In particular, when working with proportional hazards models, both packages implement the cross-validated partial log-likelihood by Verweij and 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 it 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 types 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. The first empirical study based on four high-dimensional gene expression data sets, each consisting of both learning and test sets, is presented in Sect. 3. The effect of considering several partitions in the cross-validation procedure is shown in the second empirical study (Sect. 4). Finally, Sect. 5 contains some conclusions. A simulation study in which we investigate the role of correlation between covariates with respect to the number of boosting steps and the prediction is available in the Supplementary Material. R-codes used for this paper are available in the Electronic Supplementary Material.

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

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, sketch how to apply the cross-validation technique in order to select m_{stop}, and provide some information on the Brier score, the measure of prediction ability that we use in the paper. For a more complete review on boosting, please see Mayr et al. (2014b)

2.1 Model-based boosting

Model-based boosting is a direct implementation of the gradient boosting idea described in the seminal paper of 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 pre-specified 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,

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

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 (2008a) 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 2008a). 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 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 2008a).

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 CV 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 $D_1, \ldots, D_K$. Each fold of this split is then separately used as a test set to evaluate the behavior of a model fit on the other $K - 1$ folds.

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

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 Houwelingen (1993),

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

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 same subset (D without observations in D_k). Note that the value of the first term on the right hand side of Eq. 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 over-fitting: 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 Eq. 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). In this context, the Brier score is able to measure both the discriminative ability and the calibration of a model, in contrast, e.g., with the widely used concordance index, which is only able to evaluate the former property (De Bin et al. 2014a). The Brier score 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. 2012). 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

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

"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,

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

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 gene expression of 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 test data totally separate in order to have a reliable evaluation of the prediction abilities of the resulting models. In all analyses, we assume that the covariate effects are linear and that the proportional hazards assumption holds.

Breast cancer data This data set is from a prospective multicenter study conducted by Hatzis et al. (2011) to develop genomic 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, which is 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. (2014b): the sample sizes are 282 patients (with 57 events) and 182 patients

Table 1 The four data sets used in our empirical study

Disease	Sample size (events)		Number of predictors	Reference
	Learning set	Test set		
Breast cancer	282 (57)	182 (41)	22,283	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)	44,754	Metzeler et al. (2008)
Neuroblastoma	242 (40)	120 (35)	9978	Oberthuer et al. (2008)

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

(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 is 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 is publicly available from Gene Expression Omnibus, reference GSE12417.

Neuroblastoma data The last data set contains information on 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 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 first study is the cross-validation-based choice of the optimal number of boosting steps in model-based and likelihood-based boosting. We consider values between 0 and 200. The lower limit leads to the null model, while the upper limit

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

has been arbitrarily chosen as "sufficiently large" (namely, twice the default in both *mboost* and *CoxBoost*). We investigate how the variability caused by randomness due to the CV 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 the following 2000 times:

- we apply the 3-, 5-, 10- and 20-fold CV 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 CV: 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—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, using the CV procedures implemented both in *mboost* and in *CoxBoost*. This and the following figures contain information on results of regular CV as well as information on results of repeated CV. The repeated CV is discussed in Sect. 4. For now we focus on the white boxplots in Fig. 1, which show results for the regular cross-validation. 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 Fig. 1, third row) 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 decreases with an increase in the number of folds because increasing the number of folds means approaching to (the completely deterministic) leave-one-out CV. Leave-one-out CV produces extreme numbers of steps in *mboost* for 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 CV are higher than most or all numbers of steps computed by other cross-validation procedures. This suggests

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

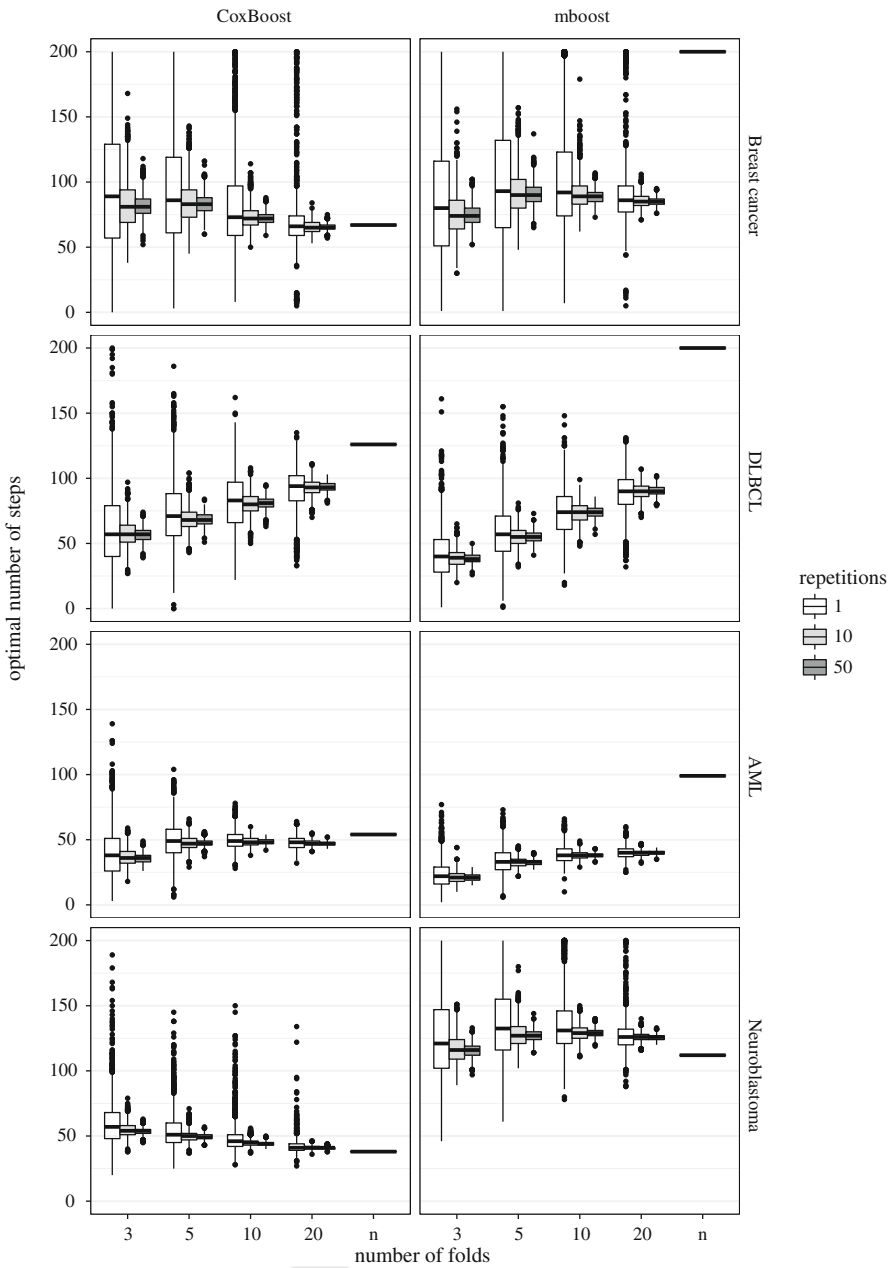

Fig. 1 Number of boosting steps (m_{stop}) selected in the 2000 iterations (except leave-one out CV) computed using different CV folds in the four data sets with both *CoxBoost* (left) and *mboost* (right). The color defines the type of CV. White stands for normal, gray for repeated CV

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

that leave-one-out CV leads to models that are more likely to overfit the data in these cases.

Note that in our study the number of boosting steps is allowed to vary from 0 to 200. In some cases (see, e.g., the results for the breast cancer data set, Fig. 1, first row) the upper limit is reached, meaning that the results could be even more extreme with a larger maximum number of boosting steps. Given the relevant increase in computational time and computer memory necessary to consider a higher upper limit, we think that a value of 200 is fairly reasonable and sufficient to demonstrate the problem of variability induced due to the random CV splits.

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. In Fig. 2 we report the number of predictors selected in each of the replications of our experiment for the model-based (*mboost*) and the likelihood-based (*CoxBoost*) boosting procedures, respectively. The downward facing triangles indicate the minimum number of predictors selected, i.e. the number of predictors always selected. The upward facing triangles indicate the maximum number of predictors selected, i.e. the number of predictors selected at least once. For a more precise visualization of the number of predictors always selected and the number of predictors selected at least once, see Figures 6 and 7 in the Supplementary Material. The Supplementary materials also contain the complete tables of the selected predictors, including the information on the number of times they are selected (Tables 2–5 in the Supplementary Material). Note that the number of predictors selected at least once, the number of predictors always selected and the mean number of predictors selected, is equivalent for leave-one-out CV because it is deterministic and was only computed once.

Again, we first focus on the regular CV and ignore the results of the repeated CV for now. 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 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. Note that a boosting model always contains all predictors of a scarcer model (with fewer boosting steps), i.e. the predictors selected in the beginning always stay in the model.

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 Fig. 1, first row). This results in a less evident stabilization in the predictors selected. For example using *CoxBoost*

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

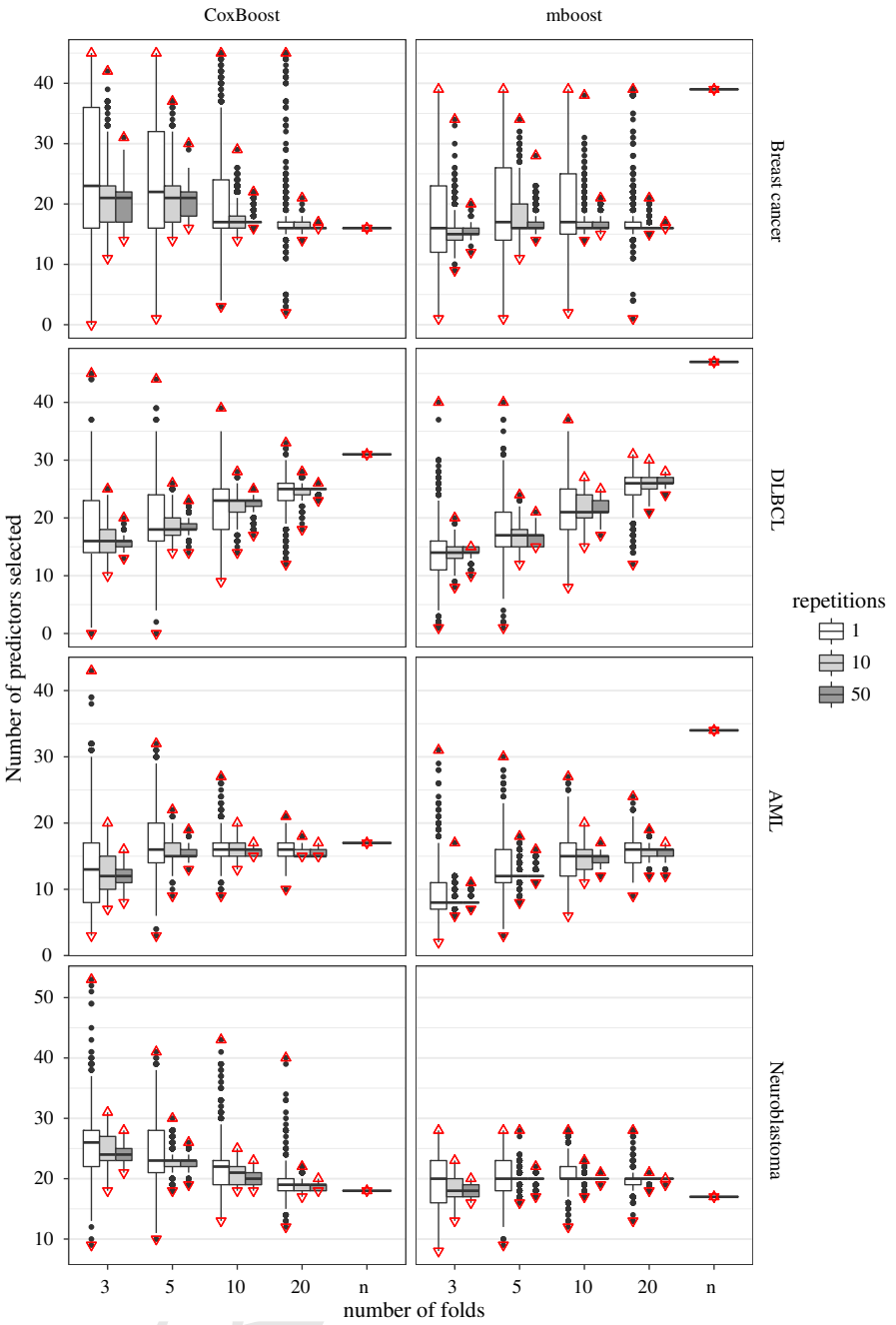

Fig. 2 Number of predictors selected in 2000 iterations computed using different CV folds in the four data sets with both *CoxBoost* (left) and *mboost* (right). The color defines the type of CV. White stands for normal, gray for repeated CV. The triangles indicate the minimum and maximum number of predictors selected (color figure online)

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

the number of predictors always included is 0 for the 3-fold CV, 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 CV. 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. There is no tendency of the median number of selected predictors over data sets. For the DLBCL and AML data it increases with increasing number of folds. For the Breast cancer and Neuroblastoma data it decreases with increasing number of folds for CoxBoost and does not show a clear tendency for mboost.

Leave-one-out CV in general tends to favor more complex models, which are more likely to overfit the learning data. Figure 2 supports that in *mboost* for all data sets except the neuroblastoma data set. For *CoxBoost* the number of predictors selected is particularly high for the DLBCL data. So essentially all examples that show 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* (see Tables 2–5 in the Supplementary Material). This is a consequence of the differences in the learning path for the two boosting techniques (for further details, see De Bin 2016).

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

Throughout the paper, we stressed the influence of the number of boosting steps on the model sparsity. To better understand this statement, we plot in Fig. 3 all values of m_{stop} obtained in *all* iterations against the number of predictors included in the corresponding models. Given a certain m_{stop} the model is deterministic and hence a differentiation between different types of CV is not needed here. We note that models are less sparse as the value of the optimal number of boosting steps increases, resulting in a non-decreasing function. The steps in the curve correspond to those iterations in which the boosting algorithm includes a new predictor into the model. When the algorithm updates the regression coefficient of a previously selected predictor, instead, the curve remains flat. 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 3 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 2016),

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

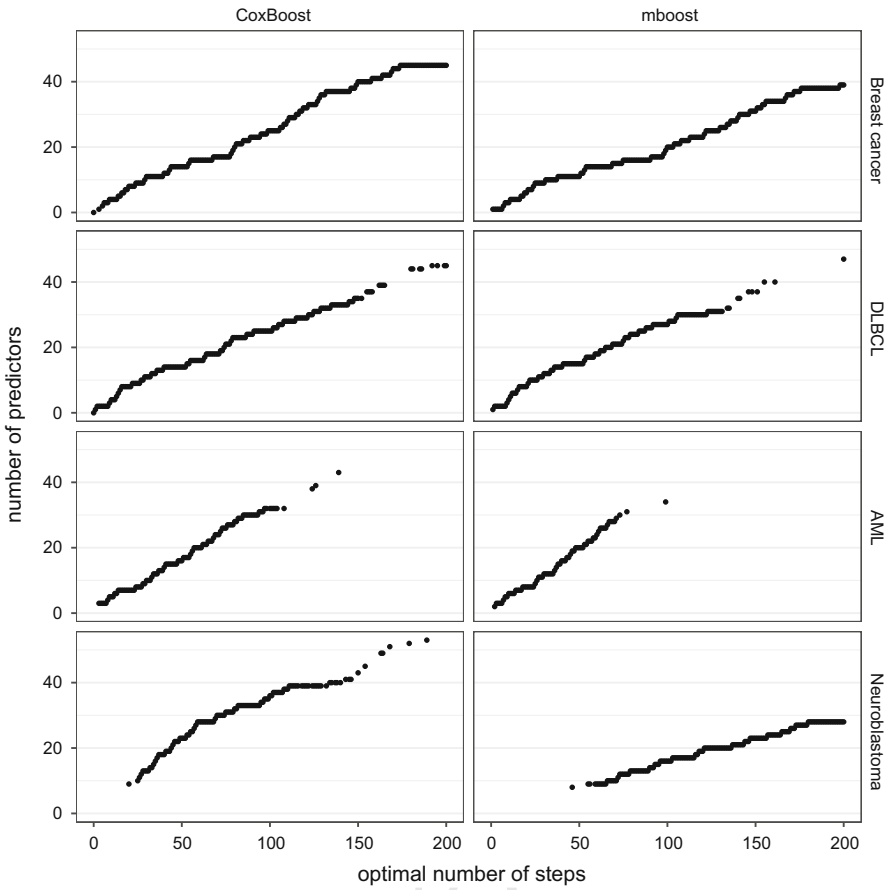

Fig. 3 Optimal number of steps plotted against the number of predictors included in the respective model, for both *CoxBoost* (left) and *mboost* (right)

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 DLBCL data, $\lambda = 1854$ for the AML data and $\lambda = 720$ for the neuroblastoma data. These values were computed using the procedure described in Sect. 2.2. Larger values of the penalty parameter correspond to smaller step-wise updates of the coefficients, and consequently there are more iterations without adding new predictors (flat parts of the curves in Fig. 3); 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

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

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 Fig. 4 the results for *CoxBoost* (left) and for *mboost* (right) using 3-, 5-, 10-, 20-fold and leave-one-out CV. The results are based on 2000 iterations, except for the leave-one-out CV, 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 Fig. 4). Strongly unexpected, leave-one-out CV 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 predictors with weak effects are 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. 2014b). Please note that, in general, the inclusion of more 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 CV, which is known to favor data-specific models. In all other cases, indeed, the integrated Brier score from leave-one-out CV is higher than the median of the integrated Brier score from other folds, including *CoxBoost* on the breast cancer data set.

Figure 5 shows the connection between the number of boosting steps and the integrated Brier score for all analyses. Again, note that given a certain m_{stop} the model is deterministic and thus we do not differentiate between types of CV here. For the Breast cancer and the Neuroblastoma data sets the figures suggest that m_{stop} greater than 200 should have been chosen, whereas for the other two data sets 200 was more than enough. The figure also gives information on why for the AML data set there are such outliers in the integrated Brier score seen in Fig. 4: The prediction performance is very bad if there are only very few boosting steps but improves quickly with an increase in the number of boosting steps.

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

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

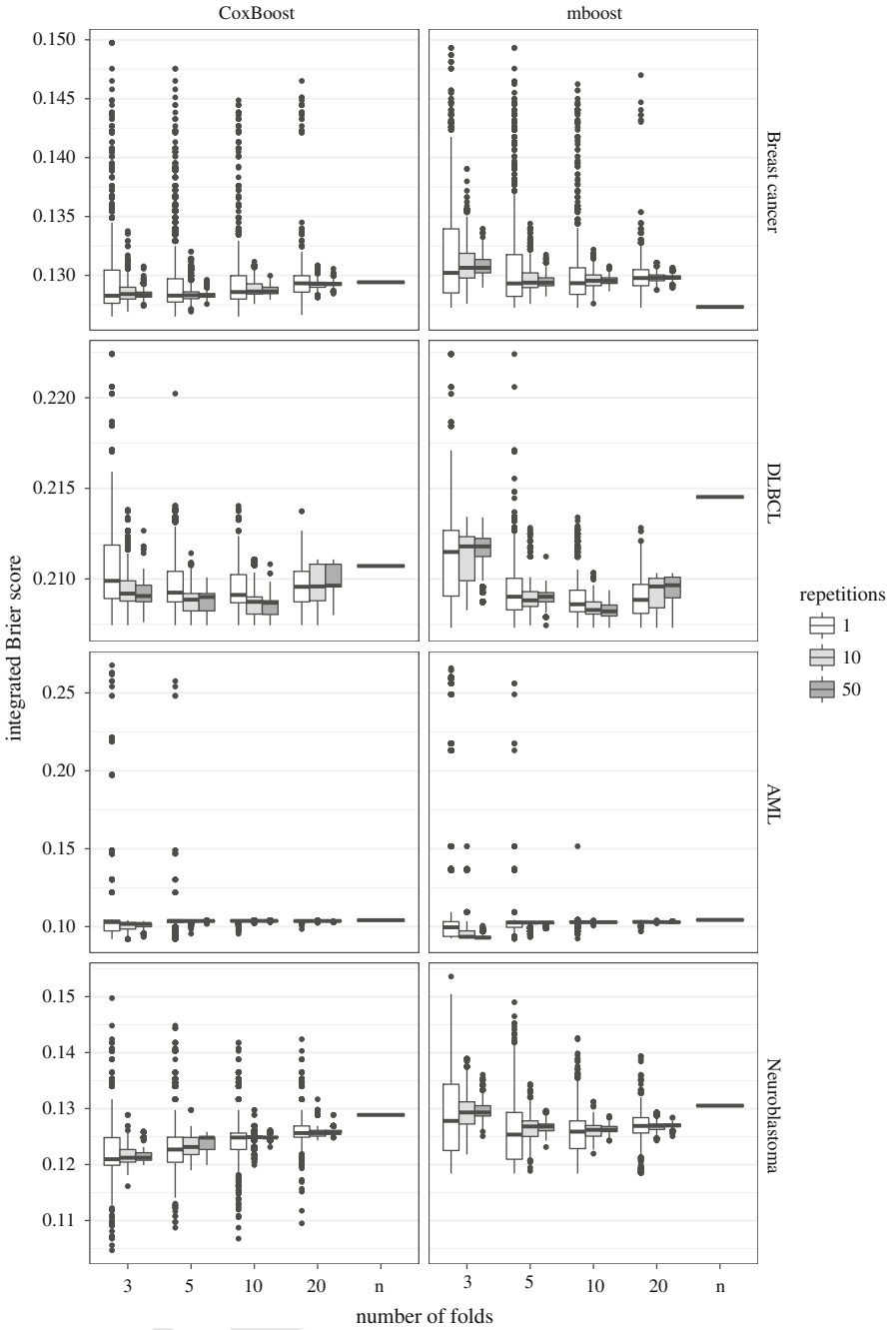

Fig. 4 Integrated Brier score for models computed using different CV folds and a different number of repetitions in the four data sets, for both *CoxBoost* (left) and *mboost* (right)

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

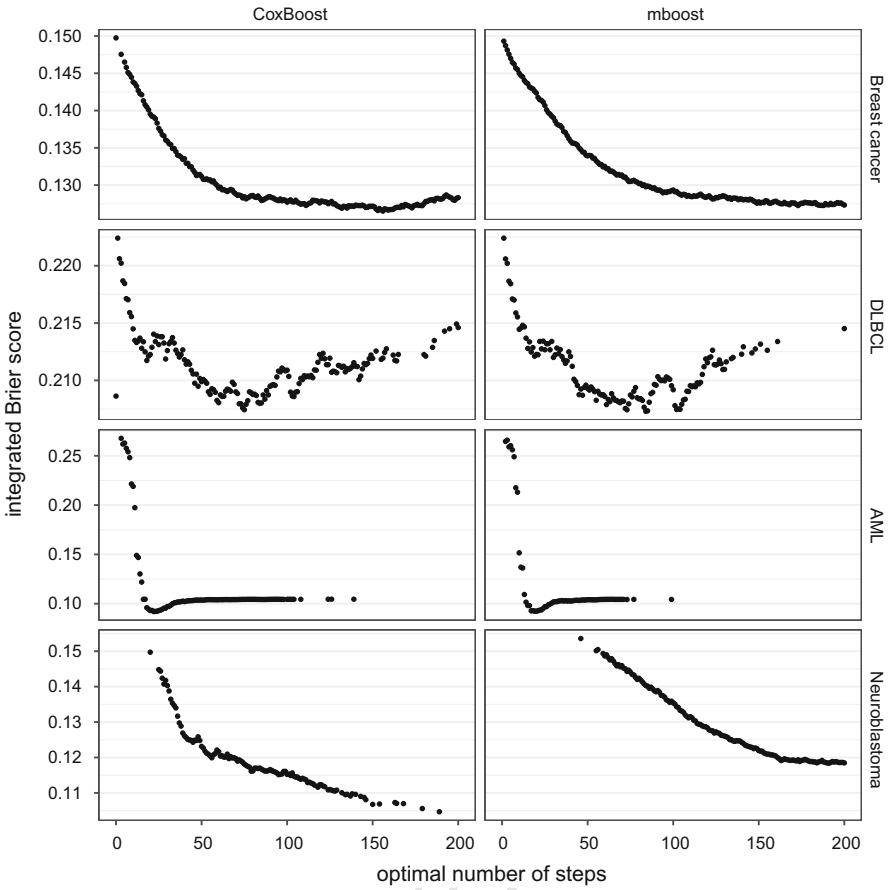

Fig. 5 Optimal number of steps plotted against the integrated Brier score, for both *CoxBoost* (left) and *mboost* (right)

(Kohavi 1995), and transform the 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,

$$rcvpl(m) = \sum_{r=1}^{R} cvpl_r(m)$$

with R being the number of repetitions and $cvpl_r(m)$ the cross-validated partial log-likelihood of the r-th repetition. Note that due to the random nature of cross-

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

validation the subsets $D_1, \ldots, D_K$ (see Eq. 1) are different for each repetition $r = 1, \ldots, R$.

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 Sect. 3. Leave-one-out CV 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 CV 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

In this section we focus on the impact of repeated CV and with this, also address the parts of the previous figures that were not addressed in Sect. 3.

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 partial 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 CV and the 10 times repeated CV 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 partial 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 (Figure 7 in the Supplementary Material). 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

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

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 repetitions we are relatively close to a deterministic result, i.e. the inclusion frequencies of the predictors is mostly 2000 (always) or 0 (never). The median number of predictors selected barely changes except for the Breast cancer data, for which the median number of predictors selected is lower when the cross-validation is repeated. 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 partial log-likelihood for the choice of m_{stop}. As can be seen in Fig. 4, the avoidance of extreme values for the tuning parameter results in the disappearance of the worst prediction performances obtained with the simple cross-validated partial log-likelihood. For the acute myeloid leukemia example for both *mboost* and *CoxBoost* the bad predictions experienced in the previous section do not occur. 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 CV. To support our findings through Fig. 4 and to analyse the importance of both the number of folds in CV and the number of repetitions, we computed a linear model with the interquartile range of the integrated Brier score as endpoint including main effects of repeated CV and the number of folds. We computed the interquartile range from the 2000 iterations for each method (CoxBoost and mboost), data set, number of folds and number of CV repetitions, which results in 96 values. We computed a separate model for mboost and CoxBoost. The models show that using 10 repetitions instead of 1 has a significant impact whereas using 50 repetitions instead of 10 is not as pronounced for both mboost and CoxBoost (see Table 14 in the Supplementary Material). The more folds are used the lower the interquartile range of the integrated Brier score, but the only confidence interval where both limits are (at least slightly) negative is in the comparison of 5 versus 3 folds. The effects estimated from the linear models depend strongly on the four data sets selected. However a simulation study showed comparable results (see Table 1 in the Supplementary Material) which further supports our findings.

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 tuning parameter m_{stop}, namely the number of boosting steps. Please note that several statistical methods share this weakness. Until 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

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X

prediction ability of the model depending on the particular partition of the observations into the CV 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 CV procedure irrespectively of the number of folds used. In our study, we showed that the implementation of a repeated CV 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. In this work we focused on boosting for high-dimensional linear Cox-models. We believe that repeated cross-validation will lead to similar improvements in other contexts. Details, however, have to be studied.

Acknowledgements We thank Rory Wilson and Jenny Lee for language improvements. HS and RDB were supported by Grants BO3139/4-1, BO3139/4-2 and BO3139/2-3 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, R package version 1.4. http://CRAN.R-project.org/package=CoxBoost

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

Binder H, Schumacher M (2008b) Adapting prediction error estimates for biased complexity selection in high-dimensional bootstrap samples. Stat Appl Genet Mol Biol 7:12

Boulesteix AL, Richter A, Bernau C (2013) Complexity selection with cross-validation for lasso and sparse partial least squares using high-dimensional data. In: Lausen B, Van den Poel D, Ultsch A (eds) Algorithms from and for nature and life. Springer, Berlin, pp 261–268

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

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

Bühlmann P (2006) Boosting for high-dimensional linear models. Ann Stat 34:559–583

Bühlmann P, Hothorn T (2007) Boosting algorithms: regularization, prediction and model fitting. Stat Sci 22:477–505

Bühlmann P, Yu B (2003) Boosting with the L_2 loss: regression and classification. J Am Stat Assoc 98:324–339

Chang YCI, Huang Y, Huang YP (2010) Early stopping in l_2 boosting. Comput Stat Data Anal 54:2203–2213

De Bin R (2016) Boosting in Cox regression: a comparison between the likelihood-based and the model-based approaches with focus on the R-packages CoxBoost and mboost. Comput Stat 31:513–531

De Bin R, Herold T, Boulesteix AL (2014a) Added predictive value of omics data: specific issues related to validation illustrated by two case studies. BMC Med Res Methodol 14:117
De Bin R, Sauerbrei W, Boulesteix AL (2014b) Investigating the prediction ability of survival models based on both clinical and omics data: two case studies. Stat Med 33:5310–5329
Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 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, pp 148–156
Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29:1189–1232
Fuchs M, Hornung R, De Bin R, Boulesteix, AL (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. Biom J 48(6):1029–1040
Graf E, Schmoor C, Sauerbrei W, Schumacher M (1999) Assessment and comparison of prognostic classification schemes for survival data. Stat Med 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. J Am Med Assoc 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. Comput Stat 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, R package version 2.4-2. http://CRAN.R-project.org/package=mboost
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 Inf Med 51:178–186
Mayr A, Binder H, Gefeller O, Schmid M (2014a) Extending statistical boosting. Methods Inf Med 53:428–435
Mayr A, Binder H, Gefeller O, Schmid M (2014b) The evolution of boosting algorithms. Methods Inf Med 53:419–427
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 (2012) Evaluating random forests for survival analysis using prediction error curves. J Stat Soft 50(11):1–23
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. Clin Cancer Res 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. N Engl J Med 346(25):1937–1947
Schmid M, Hothorn T (2008) Flexible boosting of accelerated failure time models. BMC Bioinform 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. Stat Med 12:2305–2314

Journal: 180 Article No.: 0773 TYPESET DISK LE CP Disp.:2017/11/25 Pages: 21 Layout: Small-X