Title

A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index

Journal

Computational and Mathematical Methods in Medicine, 2013(online)

ISSN

1748-670X

Authors

Chen, Yifei

Jia, Zhenyu

Mercola, Dan

et al.

Publication Date

2013

DOI

10.1155/2013/873595

Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2013, Article ID 873595, 8 pages
http://dx.doi.org/10.1155/2013/873595

Research Article

A Gradient Boosting Algorithm for Survival Analysis via Direct Optimization of Concordance Index

Yifei Chen,[1] Zhenyu Jia,[2,3,4] Dan Mercola,[4,5] and Xiaohui Xie[1,6]

[1] *Department of Computer Science, University of California Irvine, Irvine, CA 92697, USA*
[2] *Department of Statistics, The University of Akron, Akron, OH 44325, USA*
[3] *Department of Family and Community Medicine, Northeast Ohio Medical University, Rootstown, OH 44272, USA*
[4] *Department of Pathology and Laboratory Medicine, University of California Irvine, Irvine, CA 92697, USA*
[5] *Institute for Clinical and Translational Cancer Biology, University of California Irvine, Irvine, CA 92697, USA*
[6] *Institute for Genomics and Bioinformatics, University of California Irvine, Irvine, CA 92697, USA*

Correspondence should be addressed to Xiaohui Xie; xhx@ics.uci.edu

Received 9 September 2013; Accepted 8 October 2013

Academic Editor: Lev Klebanov

Survival analysis focuses on modeling and predicting the time to an event of interest. Many statistical models have been proposed for survival analysis. They often impose strong assumptions on hazard functions, which describe how the risk of an event changes over time depending on covariates associated with each individual. In particular, the prevalent proportional hazards model assumes that covariates are multiplicatively related to the hazard. Here we propose a nonparametric model for survival analysis that does not explicitly assume particular forms of hazard functions. Our nonparametric model utilizes an ensemble of regression trees to determine how the hazard function varies according to the associated covariates. The ensemble model is trained using a gradient boosting method to optimize a smoothed approximation of the concordance index, which is one of the most widely used metrics in survival model performance evaluation. We implemented our model in a software package called GBMCI (gradient boosting machine for concordance index) and benchmarked the performance of our model against other popular survival models with a large-scale breast cancer prognosis dataset. Our experiment shows that GBMCI consistently outperforms other methods based on a number of covariate settings. GBMCI is implemented in R and is freely available online.

1. Introduction

Survival analysis focuses on developing diagnostic and prognostic models to analyze the effect of covariates on the outcome of an event of interest, such as death or disease recurrence in disease studies. The analysis is often carried out using regression methods to estimate the relationship between the covariates and the *time to event* variable. In clinical trials, time to events is usually represented by *survival times*, which measure how long a patient with a localized disease is alive or disease-free after treatment, such as surgery or surgery plus adjuvant therapy. The covariates used in predicting survival times often include clinical features, such as age, disease status, and treatment type. More recently, molecular features, such as expression of genes, and genetic features, such as mutations in genes, are increasingly being included in the set of covariates. Survival analysis also has applications in many other fields. For instance, it is often used to model machine failure in mechanical systems. Depending on specific circumstances, survival times may also be referred to as *failure times*.

A major complication for survival analysis is that the survival data are often incomplete due to censoring, because of which standard statistical and machine learning tools on regression cannot be readily applied. The most common type of censoring occurring in clinical trials is right censoring, where the survival time is known to be longer than a certain value but its precise value is unknown. This can be due to multiple reasons. For instance, a patient might withdraw from a clinical trial or a clinical trial might end early such that some patients are not followed up with afterwards.

Many statistical methods have been developed for survival analysis. One major category of these methods adopts a likelihood-based approach. An essential component of the models in this category is the estimation of the hazard function $\lambda(t)$, defined as the event rate at time t conditional on survival up to time t. Different models often impose different assumptions on the forms of the hazard function. In particular, the proportional hazards (PH) model (also called the Cox model), one of the most prevalent models in survival analysis, assumes that different covariates contribute multiplicatively to the hazard function [1–4]. To relax the proportional hazards assumption and allow for more complicated relationships between covariates, parametric models based on artificial neural networks (ANN) [5–8] and ensembles of tree models based on boosting [9–12] have also been proposed. In order to handle the censored data, all these models use an approximation of the likelihood function, called the Cox partial likelihood, to train the predictive model. The partial likelihood function is computationally convenient to use; however, it is unclear how well the full likelihood can be approximated by the partial likelihood.

Many other methods aiming at optimizing a different class of objective functions rather than the partial likelihood have also been proposed. Some of these methods adapt existing regression models to estimate the relationship between survival times and covariates, by taking the censored data into account in training the models [13, 14], while others adopt a classification-based framework and train their models using only the rank information associated with the observed survival times [8, 15, 16]. Recently, random survival forests [17, 18], a new ensemble-of-trees model based upon bagging, became popular in survival analysis. They resort to predicting either the cumulative hazard function or the log-transformed survival time.

In clinical decision making, physicians and researchers are often more interested in evaluating the *relative risk* of a disease between patients with different covariates than the absolute survival times of these patients. For this purpose, Harrell et al. introduced the important concept of *concordance index* (C-index, concordance C, or simply CI) as a measure of the separation between two survival distributions [19, 20]. Given two survival distributions, the C-index computes the fraction of pairs of patients with consistent risk orders over the total number of validly comparable pairs. Because of its focus on assessing the accuracy of relative risk, the C-index is widely adopted in survival model performance evaluation, where the order of predicted survival times is compared to the order of the observed ones [21–23].

Our goal in this paper is to develop a new survival model to capture the relationship between survival times and covariates by directly optimizing the C-index between the predicted and observed survival times. Although both the Cox model based on partial likelihood and the ranked-based methods mentioned above also utilize only the order information between survival times, the C-index based method provides a more principled way of combining all pairwise order information into a single metric. There have been prior attempts in directly learning the C-index for survival analysis, including a neural network based model [21] and an extension of the Cox model trained using a lower bound of C-index [22]. However, both methods impose parametric assumptions on the effect of covariates on survival times. Our contribution here is to adopt a nonparametric approach to model the relationship between survival times and covariates by using an ensemble of trees and to train the ensemble model by learning the C-index.

In the following, we will provide a detailed description of our ensemble survival model based on learning the C-index. We will derive an algorithm to train the model using the gradient boosting method originally proposed by Friedman [9]. The algorithm is implemented in an R software package called GBMCI (gradient boosting machine for concordance index), which is freely available at https://github.com/uci-cbcl/GBMCI. We benchmark the performance of GBMCI using a large-scale breast cancer prognosis dataset and show that GBMCI outperforms several popular survival models, including the Cox PH model, the gradient boosting PH model, and the random survival forest, in a number of covariate settings.

2. Materials and Methods

2.1. Survival Analysis. We review the basic concepts of survival analysis here. For a systematic treatment, see [24, 25]. In survival analysis, the time to event (death, failure, etc.) t is typically modeled as a random variable, which follows some probability density distribution $p(t)$. The density can be characterized by the *survival function* $S(t) = \Pr(T > t) = \int_t^\infty p(T)\mathrm{d}T$ for $t > 0$. The survival function captures the probability that the event does not happen until time t. A closely-related concept is the *hazard function* $\lambda(t) = \lim_{\Delta t \to 0}(\Pr(t < T < t + \Delta t \mid T > t))/\Delta t = p(t)/S(t)$, which measures the event rate at time t conditioned on survival until t. One can further show that $S(t) = e^{-\int_0^t \lambda(\tau)\mathrm{d}\tau}$.

The likelihood function for right-censored survival data is expressed as

$$L\left(\theta; \{x_i, t_i, \delta_i\}_{i=1}^n\right) = \prod_{i \in E} p\left(t_i \mid x_i, \theta\right) \prod_{j \in C} S\left(t_j \mid x_j, \theta\right) = \prod_{i=1}^n \lambda\left(t_i \mid x_i, \theta\right)^{\delta_i} S\left(t_i \mid x_i, \theta\right). \quad (1)$$

Note the augmentation of our notation (we will follow this convention in the following context unless otherwise stated): θ is the set of regression parameters of the survival/hazard model; δ_i, $i = 1, \ldots, n$, indicates whether the event happens ($\delta = 1$), or not ($\delta = 0$, i.e., the data is censored); x_i, $i = 1, \ldots, n$, are the explanatory covariates that affect the survival time; E is the set of data whose events are observed; and C is the set of censored data. The full maximum-likelihood approach would optimize L over the functional space of S (or λ) and parameter space of θ. Unfortunately, this is often intractable.

2.1.1. Proportional Hazard Model. In his seminal work [1, 2], Cox introduced the *proportional hazard* (PH) model $\lambda(t \mid x, \theta) = \lambda_0(t) \exp\{x^T\theta\}$. $\lambda_0(t)$ is the *baseline* hazard function;

$\exp\{x^T\theta\}$ is the relative hazard, which summarizes the effect of covariates. Cox observed that under the PH assumption, it suffices to estimate θ without the necessity of specifying $\lambda_0(t)$ and optimizing the likelihood (1). Instead, he proposed to optimize the so-called Cox partial likelihood

$$L_p\left(\theta; \{x_i, t_i, \delta_i\}_{i=1}^n\right) = \prod_{i\in E} \frac{\exp\left\{\theta^T x_i\right\}}{\sum_{j:t_j\geq t_i} \exp\left\{\theta^T x_j\right\}}. \quad (2)$$

The Cox model has become very popular in evaluating the covariates' effect on survival data and is generalized to handle time-varying covariates and time-varying coefficients [3, 4]. However, the proportional hazards assumption and the maximization of the partial likelihood remain two main limitations. Nonlinear models, for example, multilayer neural networks [5–7], have been proposed to replace $\theta^T x$. However, they still assume parametric forms of the hazard function and attempt to optimize the partial likelihood.

2.1.2. Concordance Index. The *C-index* is a commonly used performance measure of survival models. Intuitively, it is the fraction of all pairs of patients whose predictions have correct orders over the pairs that can be ordered. Formally, the C-index is

$$\begin{aligned} \mathrm{CI} &= \frac{1}{|\mathscr{P}|} \sum_{(i,j)\in\mathscr{P}} I\left(F\left(x_i\right) < F\left(x_j\right)\right) \\ &= \frac{1}{|\mathscr{P}|} \sum_{i\in E} \sum_{j:t_j>t_i} I\left(F\left(x_i\right) < F\left(x_j\right)\right). \end{aligned} \quad (3)$$

$\mathscr{P}$ is the set of validly orderable pairs, where $t_i < t_j$; $|\mathscr{P}|$ is the number of pairs in $\mathscr{P}$; $F(x)$ is the prediction of survival time; I is the indicator function of whether the condition in parentheses is satisfied or not. In the PH setting, the predicted survival time can be equivalently represented by the negative log relative hazard. The C-index estimates the probability that the order of the predictions of a pair of comparable patients is consistent with their observed survival information.

2.2. Gradient Boosting Machine. The *gradient boosting machine* (GBM) is an ensemble learning method, which constructs a predictive model by additive expansion of sequentially fitted weak learners [9, 10]. The general problem is to learn a functional mapping $y = F(x;\beta)$ from data $\{x_i, y_i\}_{i=1}^n$, where β is the set of parameters of F, such that some cost function $\sum_{i=1}^n \Phi(y_i, F(x_i;\beta))$ is minimized. Boosting assumes $F(x)$ follows an "additive" expansion form $F(x) = \sum_{m=0}^M \rho_m f(x;\tau_m)$, where f is called the *weak* or *base learner* with a weight ρ and a parameter set τ. Accordingly, $\{\rho_m, \tau_m\}_{m=1}^M$ compose the whole parameter set β. They are learnt in a greedy "stage-wise" process: (1) set an initial estimator $f_0(x)$; (2) for each iteration $m \in \{1, 2, \ldots, M\}$, solve $(\rho_m, \tau_m) = \arg\min_{\rho,\tau} \sum_{i=1}^n \Phi(y_i, F_{m-1}(x_i) + \rho f(x_i;\tau))$. GBM approximates (2) with two steps. First, it fits $f(x;\tau_m)$ by

$$\tau_m = \arg\min_{\tau} \sum_{i=1}^n \left(g_{im} - f(x_i;\tau)\right)^2, \quad (4)$$

where

$$g_{im} = -\left[\frac{\partial\Phi\left(y_i, F\left(x_i\right)\right)}{\partial F\left(x_i\right)}\right]_{F(x)=F_{m-1}(x)}. \quad (5)$$

Second, it learns ρ by

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^n \Phi\left(y_i, F_{m-1}\left(x_i\right) + \rho f\left(x_i;\tau_m\right)\right). \quad (6)$$

Then, it updates $F_m(x) = F_{m-1}(x) + \rho_m f(x;\tau_m)$. In practice, however, *shrinkage* is often introduced to control overfitting, and the update becomes $F_m(x) = F_{m-1}(x) + \nu\rho_m f(x;\tau_m)$, where $0 < \nu \leq 1$. If the weak learner is the regression tree, the complexity of $f(x)$ is determined by tree parameters, for example, the tree size (or depth), and the minimum number of samples in terminal nodes. Besides using proper shrinkage and tree parameters, one could improve the GBM performance by *subsampling*, that is, fitting each base learner on a random subset of the training data. This method is called *stochastic gradient boosting* [10].

Compared to parametric models such as *generalized linear models* (GLM) [26] and neural networks, GBM does not assume any functional form of F but uses additive expansion to build up the model. This nonparametric approach gives more freedom to researchers. GBM combines predictions from the ensemble of weak learners and so tends to yield more robust results than the single learner. Empirically, it also works better than the bagging-based random forests [27], probably due to its functional optimization motivation. However, it requires the cost function Φ to be differentiable with respect to F. GBM has been implemented in the popular open-source R package "gbm" [12] which supports several regression models.

2.2.1. Boosting the Proportional Hazard Model. Ridgeway [11] adapted GBM for the Cox model. The cost function is the negative log partial likelihood:

$$\Phi\left(y, F\right) = -\sum_{i=1}^n \delta_i \left\{F\left(x_i\right) - \log\left(\sum_{j:t_j\geq t_i} e^{F(x_j)}\right)\right\}. \quad (7)$$

One can then apply (4), (5), and (6) to learn each additive model. In the "gbm" package, this cost function corresponds to the "coxph" distribution and is further optimized to refit terminal nodes with Newton's method. We denote this particular GBM algorithm as GBMCOX and its implementation in the "gbm" package as "gbmcox."

2.3. Concordance Index Learning via Gradient Boosting. We now propose a gradient boosting algorithm to learn the C-index. As the C-index is a widely used metric to evaluate survival models, previous works [21, 22] have investigated the possibility to optimize it, instead of Cox's partial likelihood. However, these works are limited to parametric models, such as linear models or neural networks. Our key contribution is to tackle the problem from a nonparametric ensemble perspective based on gradient boosting.

Optimizing the C-index directly is difficult because of its discrete nature, that is, the summation over indicator functions in (3). We resort to the differentiable approximation proposed in [21], which adopts the logistic sigmoid function in each term. We call it the *smoothed concordance index* (SCI). Specifically,

$$\mathrm{SCI} = \frac{1}{|\mathscr{P}|} \sum_{(i,j)\in\mathscr{P}} \frac{1}{1 + e^{\alpha(F(x_i) - F(x_j))}}, \tag{8}$$

where α is a hyperparameter that controls the steepness of the sigmoid function (accordingly, the approximability of SCI to CI) and $F(x)$ is the prediction of survival time. Let $\Phi(y, F) = -\mathrm{SCI}$. Then, at each iteration $m > 0$ of gradient boosting,

$$\begin{aligned} g_{im} &= \left[\frac{\partial \mathrm{SCI}}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \\ &= \frac{\alpha}{|\mathscr{P}|} \left\{ \sum_{(k,i)\in\mathscr{P}} \frac{e^{\alpha(F_{m-1}(x_k) - F_{m-1}(x_i))}}{\left[1 + e^{\alpha(F_{m-1}(x_k) - F_{m-1}(x_i))}\right]^2} \right. \\ &\qquad \left. - \sum_{(i,j)\in\mathscr{P}} \frac{e^{\alpha(F_{m-1}(x_i) - F_{m-1}(x_j))}}{\left[1 + e^{\alpha(F_{m-1}(x_i) - F_{m-1}(x_j))}\right]^2} \right\}. \end{aligned} \tag{9}$$

So the base learner $f(x; \tau_m)$ can be fitted using $\{g_{im}\}_{i=1}^n$ and (4). Next,

$$\rho_m = \arg\max_{\rho} \frac{1}{|\mathscr{P}|} \sum_{(i,j)\in\mathscr{P}} \frac{1}{1 + e^{\alpha(F_{m-1}(x_i) + \rho f(x_i;\tau_m) - F_{m-1}(x_j) - \rho f(x_j;\tau_m))}}. \tag{10}$$

Although differentiable, SCI has a complicated error surface and is neither convex nor concave. This brings two problems. First, the algorithm's performance depends on its initialization which may lead to different local optima; second, it is difficult to find the global solution of ρ_m in (10). In our implementation, we set the initial estimation $\{f_0(x_i)\}_{i=1}^n$ as the prediction from a fitted PH model and use line search to detect ρ_m locally. Empirically, we have found that these heuristics work well for the algorithm.

Algorithm 1, named as GBMCI, summarizes our whole algorithm, which also incorporates the stochastic boosting mechanism [10]. Note that ensemble size M is an important parameter that requires tuning, as small M may not capture the true model, while large M makes the algorithm apt to overfitting. In practice, it is often selected by cross validation. We implement GBMCI in the "gbm" package, under a new distribution called "sci," which shares the same regression tree engine and complete software architecture as "gbmcox" does. We name our implementation of GBMCI as "gbmsci."

3. Results

3.1. Dataset and Feature Extraction. We illustrate the utility of GBMCI on a large breast cancer dataset, which was originally released by Curtis et al. [28]. The dataset was adopted by the Sage Dream Breast Cancer Challenge (BCC) [29], where it was named *Metabric*. It contains gene expressions, copy number variations, clinical information, and survival data of 1,981 breast cancer patients. The gene expression data consist of 49,576 microarray probes; the copy number data consist of 18,538 SNP probes; the clinical data contain 25 clinical covariates; the survival data contain the survival time and status (dead or censored). Following the convention of BCC, we reserve 1001 patients for training and the other 980 for testing. We applied several successful feature selection schemes from the top competitors in BCC. See Table 1 for details on how these features were generated.

3.2. Experimental Settings. As a boosting model, GBMCI's main competitor is the boosted proportional hazard model GBMCOX. As they share identical software environment with a common regression tree engine, the comparison should be reliable and reasonable. For baseline evaluation, we investigate the performance of the PH model with a stepwise Akaike information criterion (AIC) model selection scheme (denoted as "cox"). In addition, we also consider the popular random survival forest (RSF) approach by Ishwaran et al. [18], which is implemented in the R package *randomSurvivalForest* [30] (denoted as "rsf"). We use the concordance index as the evaluation criteria. All experiments are performed in R 2.15.1 software environment.

For "gbmsci," the hyperparameter α controls how well SCI approximates CI. Large α values make the approximation good, but the gradient can be very large or even ill defined and *vice versa*. In practice, we find $\alpha = 1$ strikes a good balance between approximability and numerical stability. The line-search range is $[0, 100]$ along the gradient direction. The shrinkage ν in "gbm" is 0.001 by default. In our experiments, we find $\nu = 0.002$ works well for "gbmcox" and $\nu = 1$ does for "gbmsci." We do not essentially apply shrinkage for "gbmsci," because the small line-search range $[0, 100]$ does not necessarily detect the global optimal ρ, thus it implicitly contributes to shrinkage. This is mainly for computational efficiency purpose. "gbmsci" and "gbmcox" share other important parameter configurations: maximum number of trees is 1500 (actual number is automatically tuned by 5-fold cross validation); tree depth is 6; n_s/n (see Algorithm 1) is 1 or 0.5. For "rsf," the number of trees is 1500; other parameters use default configurations.

3.3. Empirical Comparison. Each method is tested using the five feature representations in Table 1. For "gbmsci" and "gbmcox", as cross validation introduces randomness by partitioning the training data, we repeat the experiment 50 times. Their predictive concordance indices are shown in Figures 1 and S1 (see Figure S1 in Supplementary Material available online at http://dx.doi.org/10.1155/2013/873595). For "cox", the predictive concordance indices are shown in Table 2, which also summarizes the performances of "gbmsci" and "gbmcox." For "rsf," we also do 50 random tests because of bootstrapping when growing trees. The predictive concordance indices are shown in Figure S2.

Figures 1 and S1 show that "gbmsci" fairly consistently outperforms "gbmcox." The advantage is notable when using the features of *cl*, *clge*, *ge*, and *mt* (without subsampling)

Initialize $\{f_0(x_i)\}_{i=1}^n$ with the prediction of Cox's PH model
Set shrinkage ν, and subsampling size $n_s \leq n$
For $m = 1 : M$ **do**
  Compute $\{g_{im}\}_{i=1}^n$ by (9)
  Randomly select a subset $\{x_i, t_i, \delta_i\}_{i=1}^{n_s}$ from the whole dataset
  Fit the weak learner $f(x; \tau_m)$, for example, a regression tree, upon $\{x_i, g_{im}\}_{i=1}^{n_s}$
  Compute ρ_m by (10) using line-search
  Update $\{F_m(x_i)\}_{i=1}^n$ by $F_m(x) = F_{m-1}(x) + \nu\rho_m f(x; \tau_m)$
end for

Algorithm 1: (Stochastic) gradient boosting machine for concordance index learning (GBMCI).

Table 1: The five sets of features extracted from the Metabric breast cancer dataset.

Category	Abbreviation	Explanation
Clinical feature	*cl*	A subset of clinical covariates is selected by fitting the Cox model with AIC in a stepwise algorithm. The frequently selected features include age at diagnosis, lymph node status, treatment type, tumor size, tumor group, and tumor grade.
Gene feature	*ge*	A subset of gene expression microarray probes using Illumina HT 12v3 platform is selected whose concordance indices to the survival data are ranked highest (positive concordant) or lowest (negative concordant). A few examples are, "ILMN_1683450," "ILMN_2392472," "ILMN_1700337."
Clinical and gene feature	*clge*	A combination of previously selected clinical features and gene expression features is used to fit the Cox model with AIC in a stepwise algorithm, yielding a refined subset of features.
Metagene feature	*mt*	The high-dimensional gene expression data is fed into an iterative *attractor finding* algorithm, yielding a few Attractor Metagenes which are found commonly present in multiple cancer types [31]. Some multicancer attractors are strongly associated with the tumor stage, grade, or the lymphocyte status.
Clinical and Metagene feature	*mi*	A minimum subset of metagenes which has strong prognosis power for breast cancer [31], combined with several important clinical covariates, such as age at diagnosis and treatment type.

and substantial when using *mi*. "gbmsci" performs slightly worse only when using *mt* (with subsampling) but is still comparable. Further more, all differences except *mt* (with subsampling) are statistically significant (Student's t-test, all P values $< 10^{-13}$). We also note that subsampling generally improves the predictive power of both "gbmsci" and "gbmcox," except when using *cl*. This is consistent with the theoretical argument of [10, 11].

From Table 2, one can see that "gbmsci" performs better than "cox" overall. The advantage is notable when using *cl* (without subsampling) and substantial when using *clge*, *mt*, and *mi*. For other cases, "gbmsci" and "cox" are comparable. On the other hand, "gbmcox" performs better than or comparable to "cox" for *cl*, *clge*, and *mt* but does slightly worse for *ge* and *mi*. Comparing Figures 1, S1 with S2, one can see that "gbmsci" outperforms "rsf" in most cases, while "gbmcox" also performs better than "rsf" overall.

To summarize the comparative study, GBMSCI outperforms GBMCOX, Cox PH, and RSF in most of the feature-subsampling settings. The results also shed light on the importance of feature representation. First, gene expression data may have potential prognosis power given well-designed feature extraction schemes, for example, the Attractor Metagene (*mt*). Second, combining clinical and gene features together seems to provide enhanced prognosis power over using them separately. This is the case in both the original gene space (*clge*) and the transformed space (*mi*).

4. Discussion

Many machine learning techniques have been adapted and developed for survival analysis [23, 32, 33]. In particular, several important parametric models, such as neural networks and support vector regression, have been generalized to handle censored data. They provide survival studies with more comprehensive and flexible methodologies. However, ensemble methods are mostly limited to either direct adaptation of boosting to the classical PH model [11, 12] or bagging approaches such as random survival forests [17, 18]. Our proposed algorithm generalizes the gradient boosting machine to learn the C-index directly, which provides a new ensemble learning methodology for survival analysis. As the C-index is a ranking function in essence [22], our model also serves as an ensemble treatment to the *ranking problem* for survival data. This is novel and has not been addressed previously [14, 34, 35].

By studying the large-scale *Metabric* breast cancer dataset, we found that "gbmsci" overall performs better than "gbmcox," "cox," and "rsf" in terms of predictive C-indices. The improvement is notable and consistent when various feature

Table 2: Numerical statistics of predictive concordance indices of GBM models and the Cox model on the breast cancer dataset. The five feature representations are explained in Table 1. "gbmsci"-I and "gbmcox"-I run without subsampling ($n_s/n = 1$), while "gbmsci"-II and "gbmcox"-II run with subsampling ($n_s/n = 0.5$). The numerics in each entry show the average C-index and the standard deviation (in parentheses) over 50 random runs. The best performance in each column is highlighted by the bold font.

Model	Feature Representation				
	cl	*clge*	*ge*	*mt*	*mi*
"gbmsci"-I	**0.7107** (0.0015)	0.7287 (0.0005)	0.6599 (0.0004)	0.7145 (0.0004)	**0.7416** (0.0010)
"gbmcox"-I	0.7039 (0.0008)	0.7268 (0.0013)	0.6523 (0.0007)	0.7110 (0.0014)	0.7222 (0.0003)
"gbmsci"-II	0.7063 (0.0011)	**0.7341** (0.0014)	**0.6617** (0.0020)	0.7169 (0.0017)	0.7405 (0.0015)
"gbmcox"-II	0.6983 (0.0009)	0.7298 (0.0008)	0.6549 (0.0014)	**0.7173** (0.0010)	0.7306 (0.0008)
"cox"	0.7042	0.7140	0.6590	0.6659	0.7299

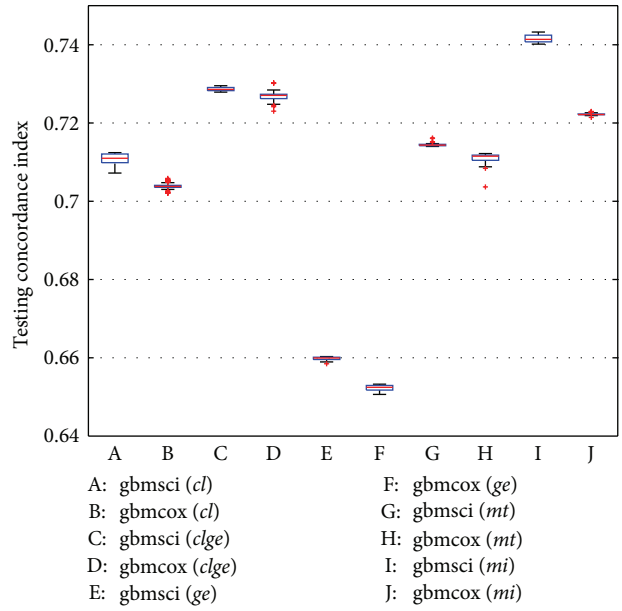

Figure 1: Predictive performance I of GBM methods on the breast cancer dataset. The box plots show the predictive concordance indices of "gbmsci" and "gbmcox" in 50 random experiments without subsampling, using the five feature representations explained in Table 1. In each box plot, the central red line indicates the median C-index; the blue box is the [25%, 75%] area; the black whiskers reach the upper and lower extremes not including outliers; the red "+" symbols represent the outliers.

representations were applied. This study also demonstrates the enhanced prognosis power when gene expression profiles and clinical variables are combined and when the gene space is remapped in the predictive model. Interestingly, "gbmsci" typically outperforms "gbmcox" and "cox" when using these informative features. This may provide useful clues for clinical decision making. Moreover, we also confirm the utility of the subsampling scheme of gradient boosting.

Although GBMCI has free parameters that require tuning, for example, α and the line-search range, they empirically work well among different experiments once they have been well tuned. In addition, the algorithm still renders similar performance, when α is within a reasonable neighborhood of 1 (e.g., $\alpha = 2$). One possible reason for the robustness is that both the objective function (8) and the gradient (9) are upper- and lower-bounded (as can be shown through basic algebraic manipulations). Such bounds are not typically available when optimizing other objective functions for different regression problems, such as the partial likelihood for the Cox model, the mean absolute error for the Lasso regression, and the polynomial alternative of SCI as proposed by [21].

The proposed algorithm has room for improvement. First, current initialization and line-search steps, although working well in practice, are not necessarily the globally optimal strategy. For initialization, one potential alternative is to fit PH models by subsampling or bootstrapping of the training data. To better address the problems, one may have to design other initialization heuristics or adopt a global optimization technique such as Monte Carlo methods. Second, GBMCI is computationally more intensive than other methods, because of the pairwise sigmoid computation in (9) and (10). Fortunately, GBMCI is easily parallelizable, which should help in dealing with large datasets. Third, biomedical research often deals with high-throughput data, for example, microarray gene expression profiling and next generation sequencing data, which require feature selection and dimension reduction. GBMCI does not tackle this task yet. However, as node-splittings of regression trees implicitly perform feature extraction, one could either run GBMCI several iterations and preselect informative variables as a "warm-up" step before the main learning routine or start GBMCI with all variables, iteratively rank their node-split frequencies, and refine the variable pool. These would allow GBMCI perform feature selection and concordance index learning in a unified framework.

Last but not least, we note that ensemble methods are in general more expensive than the Cox model, because of the necessity of tuning parameters, training ensemble weak learners, and cross validation. The tradeoff between predictive power and computational cost remains a question that depends on the specific case requirement. For example, given a particular prognosis analysis task, the Cox model may provide a quick baseline evaluation; ensemble methods could be applied, if higher predictive accuracy and more thorough investigation of covariates' effect are required.

5. Conclusion

To summarize, we have developed a new algorithm (GBMCI) for survival analysis. It performs concordance index learning

nonparametrically within the gradient boosting framework. It does not impose parametric assumptions on hazard functions, and it expands the ensemble learning methodologies of survival analysis. We implemented GBMCI in an open-source R package, and tested it using a comprehensive cancer prognosis study. GBMCI consistently performs better than three state-of-the-art survival models (the Cox PH model, its boosting expansion, and the random survival forest) over several feature representations. This study also illustrates the importance of feature engineering of clinical and gene expression data in cancer prognosis studies.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work is partially supported by Grants NSF DBI-0846218, NCI UO1CA11480 (SPECS UO1), NCI UO1CA152738 (EDRN), UO11CA162147, UCI Chao Family Comprehensive Cancer Center, NCI P30CA62203, and by the U.S. Department of Defense Congressionally Mandated Research Program Grant PC120465. The *Metabric* dataset was accessed through Synapse (https://synapse.sagebase.org/). The authors thank Dr. Wei-Yi Cheng for kindly sharing the Attractor Metagene algorithm and Daniel Newkirk and Jacob Biesinger for helpful discussions.

References

[1] D. R. Cox, "Regression models and life-tables," *Journal of the Royal Statistical Society B*, vol. 34, no. 2, pp. 187–220, 1972.

[2] D. R. Cox, "Partial likelihood," *Biometrika*, vol. 62, no. 2, pp. 269–276, 1975.

[3] P. K. Andersen and R. D. Gill, "Cox's regression model for counting processes: a large sample study," *The Annals of Statistics*, vol. 10, no. 4, pp. 1100–1120, 1982.

[4] T. Martinussen and T. H. Scheike, *Dynamic Regression Models For Survival Data*, Springer, New York, NY, USA, 2006.

[5] D. Faraggi and R. Simon, "A neural network model for survival data," *Statistics in Medicine*, vol. 14, no. 1, pp. 73–82, 1995.

[6] L. Mariani, D. Coradini, E. Biganzoli et al., "Prognostic factors for metachronous contralateral breast cancer: a comparison of the linear Cox regression model and its artificial neural network extension," *Breast Cancer Research and Treatment*, vol. 44, no. 2, pp. 167–178, 1997.

[7] R. M. Ripley, A. L. Harris, and L. Tarassenko, "Neural network models for breast cancer prognosis," *Neural Computing and Applications*, vol. 7, no. 4, pp. 367–375, 1998.

[8] R. M. Ripley, A. L. Harris, and L. Tarassenko, "Non-linear survival analysis using neural networks," *Statistics in Medicine*, vol. 23, no. 5, pp. 825–842, 2004.

[9] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[10] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics and Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

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

[12] G. Ridgeway, "Generalized boosted models: a guide to the gbm package," 2007.

[13] P. K. Shivaswamy, W. Chu, and M. Jansche, "A support vector approach to censored targets," in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM '07)*, pp. 655–660, October 2007.

[14] V. van Belle, K. Pelckmans, S. van Huffel, and J. A. K. Suykens, "Support vector methods for survival analysis: a comparison between ranking and regression approaches," *Artificial Intelligence in Medicine*, vol. 53, no. 2, pp. 107–118, 2011.

[15] G. C. Cawley, N. L. C. Talbot, G. J. Janacek, and M. W. Peck, "Sparse bayesian kernel survival analysis for modeling the growth domain of microbial pathogens," *IEEE Transactions on Neural Networks*, vol. 17, no. 2, pp. 471–481, 2006.

[16] L. Evers and C.-M. Messow, "Sparse kernel methods for high-dimensional survival data," *Bioinformatics*, vol. 24, no. 14, pp. 1632–1638, 2008.

[17] T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, and M. J. van der Laan, "Survival ensembles," *Biostatistics*, vol. 7, no. 3, pp. 355–373, 2006.

[18] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer, "Random survival forests," *The Annals of Applied Statistics*, vol. 2, no. 3, pp. 841–860, 2008.

[19] F. E. Harrell Jr., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati, "Evaluating the yield of medical tests," *Journal of the American Medical Association*, vol. 247, no. 18, pp. 2543–2546, 1982.

[20] F. E. Harrell, K. L. Lee, and D. B. Mark, "Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors," *Statistics in Medicine*, vol. 15, no. 4, pp. 361–387, 1996.

[21] L. Yan, D. Verbel, and O. Saidi, "Predicting prostate cancer recurrence via maximizing the concordance index," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 479–485, ACM, August 2004.

[22] V. C. Raykar, H. Steck, B. Krishnapuram, C. Dehing-Oberije, and P. Lambin, "On ranking in survival analysis: bounds on the concordance index," in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., pp. 1209–1216, The MIT Press, Cambridge, Mass, USA, 2008.

[23] A. Khosla, Y. Cao, C. C.-Y. Lin, H.-K. Chiu, J. Hu, and H. Lee, "An integrated machine learning approach to stroke prediction," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 183–191, ACM, July 2010.

[24] D. Roxbee Cox and D. Oakes, *Analysis of Survival Data*, vol. 21, Chapman & Hall/CRC, Boca Raton, Fla, USA, 1984.

[25] P. D. Allison, *Survival Analysis Using SAS: A Practical Guide*, SAS Institute, 2010.

[26] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, vol. 37, Chapman & Hall/CRC, Boca Raton, Fla, USA, 1989.

[27] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2009.

[28] C. Curtis, S. P. Shah, S. F. Chin et al., "The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups," *Nature*, vol. 486, no. 7403, pp. 346–352, 2012.

[29] A. A. Margolin, E. Bilal, E. Huang et al., "Systematic analysis of challenge-driven improvements in molecular prognostic models for breast cancer," *Science Translational Medicine*, vol. 5, no. 181, pp. 181re1–181re1, 2013.

[30] H. Ishwaran and U. B. Kogalur, "Random survival forests for r," *New Functions for Multivariate Analysis*, p. 25, 2007.

[31] W. Y. Cheng, T. H. Ou Yang, and D. Anastassiou, "Biomolecular events in cancer revealed by attractor metagenes," *PLoS Computational Biology*, vol. 9, no. 2, Article ID e1002920, 2013.

[32] B. Zupan, J. Demšar, M. W. Kattan, J. R. Beck, and I. Bratko, "Machine learning for survival analysis: a case study on recurrence of prostate cancer," *Artificial Intelligence in Medicine*, vol. 20, no. 1, pp. 59–75, 2000.

[33] M. W. Kattan, "Comparison of Cox regression with other methods for determining prediction models and nomograms," *The Journal of Urology*, vol. 170, no. 6, pp. S6–S10, 2003.

[34] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Journal of Machine Learning Research*, vol. 4, no. 6, pp. 933–969, 2004.

[35] C. J. C. Burges, K. M. Svore, P. N. Bennett, A. Pastusiak, and Q. Wu, "Learning to rank using an ensemble of lambda-gradient models," *Journal of Machine Learning Research*, vol. 14, pp. 25–35, 2011.