January 2020

Gradient Boosting for Survival Analysis with Applications in Oncology

Nam Phuong Nguyen
University of South Florida

Gradient Boosting for Survival Analysis with Applications in Oncology

by

Nam Phuong Nguyen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Arts
Department of Mathematics and Statistics
College of Art and Sciences
University of South Florida

Major Professor: Lu Lu, Ph.D.
Mingyang Li, Ph.D.
Dymtro Savchuk, Ph.D.

Date of Approval:
October 30th, 2019

Keywords: Statistics, Classification, Gradient Boosting, Machine Learning, Survival Analysis, Oncology

Table of Contents

List of Figures

List of Tables

Abstract

Cancer is one of the most deadly diseases that the world has been fighting against over decades. An enormous number of research has been conducted, via a wide scale of approaches, raging from genetic analysis to mathematical modeling. Survival analysis is a well-performed methodology frequently used to estimate the survival probability of a patient. Although there has been a large number of methods for survival analysis, efficient exploration of a high-dimensional feature space has been challenging due to its computational cost and complexity. This thesis adapts the component-wise gradient boosting algorithms for cancer survival analysis, and also proposes a new gradient boosting algorithm based on optimizing the Brier's score. The new method is illustrated with the analysis of the microarray data of diffuse large B-cell lymphoma (DLBCL). The new gradient boosting approach not only has identified similar important biomarkers as previous statistical studies on the same data set, but also offers more insights and gained understanding on medical aspects. In addition, the performance of the new method is demonstrated through a simulation study and compared with a variety of statistical and machine learning methods.

1 Introduction

1.1 Cancer

Cancer is a collection of related diseases, which the body's cells start to divide without stopping and inflate the surrounding tissues. Normally, human cells grow and duplicate in demand of body need. New cells are continuously reproduced to replace dead or damaged cells. In appearance of cancer, however, cells growth abnormally, old and damaged cells are not eliminated as usual and new cells are duplicated abundantly. In many cases, these cells can keep dividing without limitation and may form tumors. Malignant tumors, which are dangerous and deadly, invade the nearby tissues and compete with normal cells for nutrient. Additionally, this type of cancerous tumors can break off and metastasize to a new distant site via the blood or lymph system. The metastasis may form new tumors far from the original tumor (primary site of cancer). On the other hand, benign tumors do not invade the neighbor environment and can be removed. Hardly ever benign tumors grow back, while the malignant tumors sometimes do [2]. Based on the annual report of American Cancer Society, 1,762,450 new cancer cases were recorded in 2019, resulting in 606,880 deaths [2]. The most common site with cancer development is the digestive system with 328,030 new cases, discovered in the same year. The trend in age-adjusted cancer death rates (both sexes) from 1930 to 2016 has witnessed a significant increase in lung and bronchus sites and a gradual decrease in stomach site. The survival rate within 5 years for all cancers has grown remarkably since early 1960s, and is nearly doubled from 39% to 70% among whites and tripled from 27% to 63% among blacks. This thesis develops a new

machine learning technique for improved prediction of cancer survival probability by using the gradient boosting method for optimizing the Brier score over a high-dimensional feature space. The new method offers improved precision of prediction, computational efficiency, and improved understanding of the contributing features (important biomarkers). We will illustrate the model via a data set for diffuse large B-cell lymphoma.

1.2 Survival data analysis

In the current data-driven world for cancer studies, survival data can be analyzed by a great number of methods. However, those methods might be categorized into two major classes, which are statistical and machine learning methods. Traditionally statistical methods play a crucial role in survival analysis. Non-parametric methods such as Kaplan-Meier product limit estimators [3], Nelson-Aalen estimators [4] and life-table [5] laid a strong foundation for early related research. Nevertheless, non-parametric methods do not offer insights on the contribution of quantitative covariates to the survival probability. Thus, semi-parametric approaches such as the Cox's regression models [6] have been more popular for survival analysis, which allow us to take into account the impacts of covariates based on the proportional hazards assumption. Some commonly used Cox's regression models include penalized Cox regression such as Lasso and Ridge regression [7] [8] and time-dependent Cox models [9]. However, the restriction of Cox models is often due to the violation of proportional hazards assumption. In fact, the assumption is often not met for many survival data. Therefore, parametric models such as linear regression (Buckley James) [1], penalized regression [10] or the accelerated failure time (AFT) models [11] are often used. However, when we have a high dimensional feature space, the application of these statistical methods could be limited due to its associated computational complexity. Hence, the machine learning methods which are more powerful for big data analysis could be a more popular alternative for analyzing high dimensional survival data. Some well-known machine learning tools that have been adapted for survival analysis include survival trees [12] [13], Bayesian network [14], neutral

network [15], support vector machine [16], ensemble methods such as random survival forest [17] and bagging survival trees [18], gradient boosting [19] [20] and some more advanced learning methods such as active learning, transfer learning [21] and multi-task learning [22].

This thesis mainly focuses on gradient boosting methods, which offers both the simplicity of statistical methods and the computational power of machine learning techniques. Existing work on applying boosting methods to survival analyses have explored a variety of objective (loss) functions such as the negative partial log-likelihood [19], the ranked-based Gehan loss [23], the (smooth) C-index [24]. In this thesis, we are going to propose a new gradient boosting method using the modified Brier score as the loss function to optimize a aspect of model performance and demonstrate its advantage over some existing gradient boosting methods through a simulation study.

1.3 Diffuse Large B-cell Lymphoma (DLBCL) Studies

Diffuse large-B-cell lymphoma is a disease involving molecular heterogeneity. In [25], three gene expression signatures, which are germinal-center B-cell, stromal-1 and stromal-2, were used to build a multivariate model. The Affymetrix U113 plus 2.0 microarrays was used to profile the gene-expression. The data set explored in this thesis includes microarray data of diffuse large B-cell lymphoma of patients received the gold standard CHOP treatment [25]. Originally, survival time and status of 181 patients were recorded with 54,675 covariates. However, those covariates are filtered by a pre-selection procedure for high-dimensional data. Previous study [25] showed that the survival rate post treatment was influenced by differences in immune cells, fibrosis and angiogenesis. The previous study [25] also had access to a larger data set, which contains demographic of patients, Ann arbor stage, extra nodal sites, ECOG performance status and IPI score. Rosenwald et al. [26] indicated that the cure rate of patients who have extensive disease was around 0.35 to 0.40. It is also showed in [25] that the relative risk of death was 2.76% with 95% CI (1.9,3.9). Zhu Wang and C.Y.Wang [1]

used the same data set to evaluate the performance of Buckley-James boosting algorithm. Twelve probe sets were selected in this study which summarized shown in Table 1.1.

Table 1.1: Probe set ID and corresponding gene symbol derived from Zhu Wang and C.Y.Wang BJ boosting [1]

Probe set	Gene symbol	Coefficient
1558999_x_at	LOC283922/PDPR	−0.104
1561016_x		−0.098
1562727_at		−0.045
1568732_at		0.116
212713_at	MFAP4	0.058
224043_at	UPB1	0.121
229839_at	SCARA5	0.112
237515_at	TMEM56	−0.019
237797_at	DNM1L	0.212
240811_at		−0.010
242758_x_at	JMJD1A	0.112
244346_at		0.111

2 Statistical Methods for Survival Analysis

2.1 Survival Data and Functions

In the study of time-to-event data, we may not be able to observe the exact event time for the units in the study. For instance, in a study about the survival time of a particular cancer, we are interested in when the death of a patient occurs. However, not all patients die during the study period, For patients who have survived by the end of the study, the exact event times are not observed, i.e the "failure" are expected to occur after the recorded time period. On the other hand, in retrospective studies, the failures have occurred before the termination of the study. Two main objectives of survival modeling are to estimate the survival probability of a patient and to determine which covariates have significant impacts on the survival probability.

Let (T_i, δ_i, Z_i) be the survival data for the i^{th} patient, where T_i is the observed event time, δ_i is the status death or being alive and Z_i is a set of covariates associated with patient i. The survival function, which is the probability of an object surviving beyond a given time t, is

$$S(t) = P(T \geq t) = 1 - P(T \leq t) = 1 - F_T(t), \text{for } t \geq 0,$$

where $F_T(t)$ is the cumulative distribution function at time t. The hazard function is a conditional density, given that the event of interest has not happened prior to time t, which can be expressed as

$$h(t) = \lim_{h \to 0} \frac{P(t \leq T < t + h | T \geq t)}{h} = \frac{f(t)}{S(t)}.$$

The cumulative hazard function is given by

$$H(t) = \int_0^t h(s)ds = -\log S(t)$$

We also have a useful relationship given by

$$h(t) = -\frac{d}{dt}\log[1 - F(t)] = -\frac{d}{dt}\Big(-\log S(t)\Big).$$

2.2 Kaplan-Meier Product Limit Estimator

Kaplan-Meier (KM) product limit estimator is a non-parametric statistic used to estimate the survival function of the time-to-event (lifetime) data [3]. Sometimes, patients in a particular treatment program may leave the program. This will result in partial observations, i.e it is known the patient has survived a certain time but the exact failure time is unknown. The KM estimator of survival function - first introduced by Edward L.Kaplan and Paul Meier, is given by

$$\hat{S}(t) = \prod_{i:t_i \leq t} \left(1 - \frac{d_i}{n_i}\right),$$

where t_i is the observed event time, d_i is the number of events that are recorded at time t and n_i is the number of patients survived by time t_i. The KM estimator is one of the most commonly used statistics in non-parametric survival analysis, due to its ease for computation and visualization. Nevertheless, it does not allow an connection with qualitative predictors. Survival data sets with continuous covariates are often treated by parametric survival models or Cox proportional hazards models. To assess the uncertainty of KM estimates, we use $\hat{\gamma_i} = \frac{d_i}{n_i}$ to calculate binomial proportions of death in the time interval $[t_i, t_{i+1})$. The variance of γ_i is then given by

$$Var(\hat{\gamma}) = \frac{\hat{\gamma_i}(1 - \hat{\gamma_i})}{n_i}.$$

Then the Greenwoods's formula yields

$$\hat{\text{Var}}(\hat{S}(t)) = [\hat{S}(t)]^2 \sum_{i:t_i \leq t} \frac{d_i}{(n_i - d_i)n_i}.$$

Even though the non-parametric method is flexible, it is difficult to incorporate the quantitative covariates. That means it is hard to connect how individuals differ in their survival probability with other features through the KM estimates. However, the KM curve can serve as an a useful tool to validate the Cox proportional hazards assumption. If the KM curves of two groups intersect, it may be an evidence of non-proportional hazards between the two survival curves.

2.3 Cox Proportional Hazards Models

The proportional hazards model, which is in a family of semi-parametric model, assumes that the ratio of hazards of groups remain a constant as time changes [6] [27]. Let $\mathbf{z}_i = (z_{i1}, ..., z_{ip}), i = 1, ..., n$ be the set of covariates associated with the i^{th} patient. The hazard function of Cox proportional hazards model is expressed as

$$\lambda(t|\mathbf{z}_i) = \lambda_0(t) \exp(\eta) = \lambda_0(t) \exp(\mathbf{z}_i^T \mathbf{w}),$$

where $\lambda_0(t)$ is the baseline hazard (when all covariates are zeros), $\mathbf{w} = \{w_1, ..., w_p\}$ is a vector of coefficients and $\eta = \mathbf{z}_i^T \mathbf{w} = w_1 z_{i1} + ... + w_p z_{ip}$ is the prognostic index. The parameters in Cox proportional hazards model is estimated via maximizing the partial likelihood, which is given by:

$$pl(\mathbf{w}) = \prod_{i=1}^{n} \left[\frac{\exp(z_i^T \mathbf{w})}{\sum_{j \in R(T_i)} \exp(z_j^T \mathbf{w})} \right]^{\delta_i}.$$

In the above equation, $R(T_i)$ is the set of patients at risk at time t_i, and is defined as $R(t_i) := \{j : T_j \geq t_i\}$. Tsiatis (1981) [28] and Andersen and Gill (1982) [29] proved that it is possible to use the partial likelihood in place of the full likelihood to make inferences about

$\mathbf{w}$. First, we derive the log-partial likelihood by taking logarithm of the partial likelihood function, which can be writen as

$$l(\mathbf{w}) = \log pl(\mathbf{w}) = \sum_{i=1}^{n} \delta_i \mathbf{z}_i^T \mathbf{w} - \log \left(\sum_{j \in R(T_i)} \exp(\mathbf{z}_j^T \mathbf{w}) \right) \tag{1}$$

Let $G(\mathbf{w})$ be the partial likelihood score function, which is given by the first derivative of the log-partial likelihood as in

$$G(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} l(\mathbf{w}).$$

The solution of $G(\mathbf{w}) = 0$, denoted by $\hat{\mathbf{w}}$ is the maximum partial likelihood estimate of $\mathbf{w}$. Tsiatis, et.al [28] also proved that $\frac{\hat{\mathbf{w}}-\mathbf{w}}{SE(\hat{\mathbf{w}})}$ asymptotically follows the standard normal distribution. Additionally, the variance of $\hat{\mathbf{w}}$ can be computed as

$$\mathrm{Var}(\hat{\mathbf{w}}) = \left[- \frac{\partial^2}{\partial \mathbf{w}^2} l(\mathbf{w}) \right]^{-1} |_{\mathbf{w}=\hat{\mathbf{w}}}.$$

The proportional assumption can be validated by several methods, such as visualization through the Kaplan-Meier curves or using the Schoenfeld residual test [30].

2.4 Accelerated Failure Time Models

An accelerated failure time (AFT) model is characterized as an parametric survival model, which has a completely specified the distribution of survival time [31]. Unlike proportional hazards models, an accelerated failure time model assumes that the survival time is accelerated or decelerated under the effect of covariates, i.e the model describes the acceleration or deceleration of the survival time as a function of the covariates, which can be mathematically presented as

$$S(t) = S_0(kt), t \geq 0,$$

where S and S_0 are survival functions of two populations and k is a constant accelerating factor. Let $\lambda(t)$ be the hazard function of an AFT model and $\theta = \exp(-Z_i^T \mathbf{w})$, then the accelerated failure time model can be given by:

$$\lambda(t|\theta) = \theta\lambda_0(\theta t).$$

Let T_i be the failure time of the i^{th} instance, and then we have

$$\log(T_i) = -\log(\theta) + \log(\theta T_i) = Z_i^T \mathbf{w} + \sigma\epsilon_i,$$

where $\mathbf{w} = (w_1, ..., w_p)^T$ and σ are the unknown parameters.

The estimation and inference of the AFT models have been introduced by Kalbeisch and Prentice [32]. Let $Y_i = \min(\log T_i, \log C_i)$, where C_i represents the censoring time of patient i, which is assumed to be independent of $\mathbf{w}$ and σ. Let y_i be the observed data of Y_i, then the log-likelihood function is given by

$$l(\mathbf{w}, \sigma) = \sum_{i=1}^{n} \delta_i \left[\log f\left(\frac{y_i - z_i^T \mathbf{w}}{\sigma}\right) - \log(\sigma) \right] + (1 - \delta_i) \log S\left(\frac{y_i - z_i^T \mathbf{w}}{\sigma}\right),$$

where $f(.)$ and $S(.)$ are the corresponding density and survival functions of ϵ_i. Let $e_i = \frac{y_i - z_i^T \mathbf{w}}{\sigma}$, then the score function of the log-likelihood of the AFT models can be written in the matrix form of

$$U(\mathbf{w}, \sigma) = \begin{bmatrix} U_{\mathbf{w}}(\mathbf{w}, \sigma) \\ U_{\sigma}(\mathbf{w}, \sigma) \end{bmatrix} = \begin{bmatrix} \sigma^{-1} Z^T \Gamma \\ \sigma^{-1}(\mathbf{e}^T \Gamma - \mathbf{1}^T \delta) \end{bmatrix},$$

where $\Gamma = (\gamma_1, ..., \gamma_p)^T$,

$$\gamma_i = -\left[\delta_i \frac{\partial \log f(e_i)}{\partial e_i} + (1 - \delta_i) \frac{\partial \log S(e_i)}{\partial e_i} \right],$$

$\mathbf{1}^T = (1, ..., 1)^T$ and $\mathbf{e} = [\exp(e_1), ..., \exp(e_p)]^T$. The maximum likelihood estimates (MLEs) for $\mathbf{w}$ and σ are derived by solving $U(\mathbf{w}, \sigma) = \mathbf{0}$, which can be obtained by a numerical nonlinear optimizer, such as the Newton-Raphson algorithm (which is the optimization algorithm used in the "survreg" function in the R survival package).

One of the most commonly used AFT models is the log-logistic model, which has a non-monotonic hazard function. The hazard function of the log-logistic model increases at the early lifetime, and then decreases later on. The distribution of the errors of the log-logistic model is given by

$$\gamma_i = \delta_i e_i \frac{1 - \delta_i}{1 - \Phi(e_i)} \frac{\partial}{\partial e_i} \Phi(e_i),$$

where $\Phi(.)$ is the cumulative density function of the standard normal distribution. In contrast, the Weibull distribution has a PDF given by

$$f(t) = \frac{\alpha}{\beta} \left(\frac{t}{\beta} \right)^{\alpha - 1} \exp(-(t/\beta))^{\alpha},$$

where α and β are the shape and scale parameters, respectively. The Weibull distribution has a monotonic hazard function, whose shape is dependent on the shape parameter. Particularly, the shape parameter less than one indicates a monotonically decreasing hazard function. In the case of a unit shape parameter, the Weibull distribution is simply an exponential distribution with a constant failure rate. The error distribution follows a standard extreme value distribution, which is given by

$$\gamma_i = \exp(e_i) - \delta_i.$$

The uncertainty of the estimated parameters $(\mathbf{w}^T, \sigma)$ can be performed by hypothesis testing [32]. Let $\Theta = (\mathbf{w}, \sigma^T)$ be the parameter space, which can be partitioned into $\Theta = \{\theta^{(1)}, \theta^{(2)}\} = \{(\mathbf{w}_1, \sigma_1), (\mathbf{w}_2, \sigma_2)\}$. Let $\Theta_0 = \{(\theta^{(1)}, \theta_0^{(2)})\}$, then the hypothesis test on H_0 :

$\theta^{(2)} = \theta_0^{(2)}$ is performed by the likelihood ratio test

$$\Lambda = 2\left[l(\hat{\theta}^{(1)}, \hat{\theta}^{(2)}) - l(\theta^{(1)}, \theta_0^{(2)})\right].$$

It is previously proved that Λ has a χ^2 distribution with its degree of freedom equals the length of $\theta^{(2)}$.

3 Machine Learning Methods

3.1 General Boosting Methods

In this section, we are going to introduce the fundamental of the boosting methods, starting from the forward stagewise boosting, continuing on with the additive tree-based boosting, and then proceeding with the gradient boosting, which has been the most popular boosting methods and have demonstrate their successes through many applications and being among the winner solutions of many Kaggle competitions. The main idea is to build an ensemble of weak learners by optimizing a loss function of interest via a steepest decent algorithm. To scale for a high-dimenational feature space, the component-wise gradient boosting methods with various choices of the loss function will be explored.

3.2 Forward Stagewise Boosting

Boosting has become one of the most efficient and well-performed learning structures, which has been continuously developing in the last two decades. The first simple form of the boosting procedure was proposed by Schapire (1990) [33], and had drawn an extensive attentions of many researchers. Originally, the algorithm was created for classification problems; however, it was later on adapted to handle regression problems as well. One of the well-known developments is Adaptive Boosting (AdaBoost) for classification problem formulated by Yoav Freund and Robert Schapire. [34] Then Friedman et al. [35] introduced the additive tree models, which are applicable for both regression and classification problems. The idea behind boosting is to ensemble weak learners, which are slightly better than random guessing, into

a more powerful model for improved predictions. Forward stagewise additive modeling [36] could result in an easy interpretation (Algorithm 1). Mathematically, the model is given by

$$f(\mathbf{z}) = \sum_{m=1}^{M} \mathbf{w}_m b(\mathbf{z}; \theta_m), \tag{2}$$

where $\mathbf{w}_m, m = 1, ..., M$ are the coefficients (weights) and $b(\mathbf{z}; \theta_m)$ are real functions of $\mathbf{z}$, characterized by a set of parameters θ_m. We also need to define the loss (objective) function, denoted by $\Phi(\mathbf{y}, f(\mathbf{z}, \theta))$. The loss function must be a convex function. There are many choices of the loss function such as the squared-error loss, the absolute-error loss, Huber loss, etc. [36] In survival analysis, the negative partial log-likelihood function can also be used as the loss function [37].

Beginning with a chosen initial value of $f_0(\mathbf{z})$, typically chosen as zero, forward stagewise modeling adds a new basis function at each iteration, keeping the same set of parameters and coefficients that have been already included in the model. The aim of the m^{th} iteration is to find the "best" base leaner and its weights $b(\mathbf{z}, \theta_m)$ and $\mathbf{w}_m$ which minimizes the loss function $\Phi(\mathbf{y}, f_m(\mathbf{z}))$. This result, says $\mathbf{w}_m b(\mathbf{z}; \theta_m)$ is added to the result derived from the $(m-1)^{th}$ iteration to form a new model. The parameter estimation is relying on the choice of base leaners and the loss function. For example, in the case of the simple linear regression (SLR) base learners and the squared-error loss function, we have

$$\Phi(\mathbf{y}, f_m(\mathbf{z})) = \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} \mathbf{w}_m b(\mathbf{z}; \theta_m) \right)^2 = \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} \mathbf{w}_m \theta_m Z \right)^2$$

The estimates for $\hat{\theta}_m$ and $\hat{\mathbf{w}}_m$ are solutions to the system of equations

$$\sum_{i=1}^{n} \frac{\partial}{\partial \mathbf{w}_m} \Phi(y_i, f(\mathbf{z})) = 0$$

and

$$\sum_{i=1}^{n} \frac{\partial}{\partial \theta_m} \Phi(y_i, f(\mathbf{z})) = 0,$$

which can be obtained by numerical methods. Nevertheless, in the next section, we will introduce the boosting trees, which use the tree-based base learners. The parameter estimation process is more complicated than the above example.

Algorithm 1 Forward stage-wise boosting Additive Model

1. Initialize $f_0(\mathbf{z}) = 0$
2. For $m = 1$ to M:
   (a) Compute

$$(\hat{\mathbf{w}}_m, \hat{\theta}_m) = \arg \min_{\mathbf{w}_m, \theta_m} \sum_{i=1}^{N} \Phi\Big(\mathbf{y}, f_{m-1}(\mathbf{z}_i) + \mathbf{w}_m b(\mathbf{z}_i; \theta_m)\Big).$$

   (b) Update $f_m(\mathbf{z}) = f_{m-1}(\mathbf{z}) + \hat{\mathbf{w}}_m b(\mathbf{z}; \hat{\theta}_m)$

3.2.1 Boosting Trees

In addition to the forward stagewise additive modeling, boosting tree is another powerful boosting method. Classification and regression trees (CARTs) partition the feature space into disjoint regions $R_j, i = 1, ..., J$ [36]. Each of the regions, defined by their terminal nodes will be assigned a constant C_j, then the prediction is obtained by: $\mathbf{z} \in R_j \implies f(\mathbf{z}) = C_j$. Generally, the mathematical form of a tree is

$$T(\mathbf{z}; \Theta) = \sum_{j=1}^{J} C_j I(\mathbf{z} \in R_j),$$

where $\Theta = \{R_j, C_j\}_{j=1}^{J}$ are the parameters associated with the J prediction regions. The estimation of the model parameters is basically a optimization problem, which is to minimize the empirical risk $\sum_{j=1}^{J} \sum_{z_j \in R_j} \Phi(y_i, C_j)$. The solution can be obtained using the greedy or

top-down recursive partitioning strategy for finding R_j such that

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{i=1}^{N} \hat{\Phi}(y_i, T(\mathbf{z}; \Theta)).$$

The boosting tree model also has an forward stagewise expression, as it can be represented by

$$f_M(z) = \sum_{m=1}^{M} T(\mathbf{z}; \Theta_m).$$

The implementation of boosting trees with the gradient boosting in XGBoost package has achieved tremendously outstanding result among current data analysis competitions. [38]

Algorithm 2 Steepest Descent Algorithm

1. Initialize $f_0 = 0$
2. For $m = 1$ to M:
   (a) Compute the negative gradient vector $\mathbf{u}^{[m]}$

$$\mathbf{u}_i^{[m]} = -\left[\frac{\delta\Phi(y_i, f(\mathbf{z}_i))}{\delta f(\mathbf{z}_i)}\right]_{f(\mathbf{z}_i)=f_{m-1}(\mathbf{z}_i)}$$

   (b) Compute the step length ν_m

$$\nu_m = \arg\min_{\nu} \Phi(f_{m-1} + \nu\mathbf{u}^{[m]})$$

   (c) Update

$$f_m = f_{m-1} + \nu_m\mathbf{u}^{[m]}$$

The main task of algorithms introduced above is to find a set of parameters which minimizes the loss function. The optimization problem can be solved numerically by the gradient descent algorithm (analytically called method of steepest descent) [36]. We can restate the optimization as the following. Let $\Phi(f) = \sum_{i=1}^{N} \Phi(y_i, f(z_i))$ be a convex and differential loss function. We need to find $f = \{f(z_1), ..., f(z_N)\}$ which minimizes $\Phi(f)$, mathematically

denoted by

$$\hat{f} = \arg\min_{f} \Phi(f)$$

The solution given by the steepest descent algorithm is expressed as

$$f_M = \sum_{m=0}^{M} \nu_m \mathbf{u}^{[m]},$$

where $\nu_m \mathbf{u}^{[m]}$ is the increment vector at the m^{th} step. The steepest descent is a greedy algorithm, since the negative gradient vector $\mathbf{u}^{[m]}$ is the local direction where the loss function decreases most quickly.

Algorithm 2 can be adapted into boosted tree modeling, to handle the difficulty of finding R_j.

Algorithm 3 Gradient Tree Boosting Algorithm

1. Initialize
$$f_0(\mathbf{y}) = \arg\min_{\theta} \sum_{i=1}^{N} \Phi(\mathbf{y}, \theta).$$
2. For $m = 1$ to M:
   (a) For $i = 1, ..., N$ compute the pseudo residuals
   $$r_{im} = -\left[\frac{\delta \Phi(y_i, f(z_i))}{\delta f(z_i)}\right]_{f=f_{m-1}}.$$
   (b) Fit a regression tree to predict r_{im}, obtaining terminal regions $R_{jm}, j = 1, ..., J_m$
   (c) For $j = 1, ..., J_m$ compute
   $$\theta_{jm} = \arg\min_{\theta} \sum_{z_i \in R_{jm}} \Phi(\mathbf{y}, f_{m-1}(\mathbf{z}) + \theta)$$
   (d) Update
   $$f_m(\mathbf{z}) = f_{m-1}(\mathbf{z}) + \sum_{j=1}^{J_m} \theta_{jm} I(\mathbf{z} \in R_{jm}).$$
3. Output $\hat{f}(\mathbf{z}) = f_M(\mathbf{z})$

3.3 Gradient Boosting Based on Optimizing the Partial Likelihood

Although the gradient boosting algorithm has a great number of advantages, there exists a concerning issue, which is variable selection. In contemporary data-driven world, we usually deal with high-dimensional data, which demands computationally expensive analysis, as well as appropriate methods for variable selection. Especially, microarray data used in biomedical research has an enormous number of predictors, much greater than the number of observations. Therefore, Cox's proportional hazard model is not practical to be used. Component-wise boosting for multivariate linear model proposed by Peter Buhlmann et.al (2006) is a remedy for the issue [37] [39]. Instead of fitting the base-learners for the entire set of covariates $Z = \{\mathbf{z}_1, ..., \mathbf{z}_p\}$, component-wise boosting uses base-learners to fit one variable at a time $\mathbf{z}_j, j = 1, ..., p$, and then select the best update in each iteration see (Algorithm 4). The negative log-partial likelihood is chosen to be the loss function

$$\Phi(\delta_i, f(\mathbf{z}, \mathbf{w})) = -l(\mathbf{w}) = -\log pl(\mathbf{w}) = -\sum_{i=1}^{n} \delta_i \mathbf{z}_i^T \mathbf{w} + \log\left(\sum_{j \in R(T_i)} \exp(z_j^T \mathbf{w})\right).$$

The final model has the same form as the regular Cox's regression model, however the set of weights $\mathbf{w}$ is obtained by Algorithm 4.

3.4 Gradient Boosting via Optimizing the Concordance Index (C- index)

The concordance index (C-index) or Harrell's C_H is one of tools used for evaluating the performance of a survival model. Given a pair of patients (i, j) with prognostic indices (η_i, η_j) and the corresponding survival times (T_i, T_j). The C-index is defined as the probability of a patient having larger prognostic index but actually survives in shorter period of time.

Algorithm 4 Component-wise Gradient Boosting

1. Initialize $\hat{\mathbf{w}}^{[0]} = (0, ..., 0)$ and the learning rate $0 \leq \nu \leq 1$.
2. For $m = 1$ to M, compute:

$$\mathbf{u}^{[m]} = \frac{\partial \Phi(\mathbf{y}, f(\mathbf{z}, \mathbf{w}))}{\partial f(\mathbf{z}, \mathbf{w})}\bigg|_{\mathbf{w}=\hat{\mathbf{w}}^{[m-1]}} = -\frac{\partial l(\mathbf{w})}{\partial \mathbf{w}}\bigg|_{\mathbf{w}=\hat{\mathbf{w}}^{[m-1]}}$$

3. For $\mathbf{z}_j$ in $Z = \{\mathbf{z}_1, ..., \mathbf{z}_p\}$:
   (a) Fit base-learners $b(\mathbf{z}_j, \theta_j)$ with response variable $\mathbf{u}^{[m]}$ and obtain the least squared estimator $\hat{\theta}_j$.
   (b) Select an index k in $\{1, ..., p\}$, such that:

$$k = \arg\min_j \sum_{i=1}^{n} (\mathbf{u}^{[m]} - \mathbf{z}_j^T \hat{\theta}_j)^2$$

4. Update $\mathbf{w}^{[m]} = \mathbf{w}^{[m-1]} + \nu \hat{\theta}_k$

Particularly, C-index is given by

$$C = P(\eta_i > \eta_j | T_i < T_j).$$

Andreas Mayr et al. [40] used the smoothed C-index as the loss function of component-wise gradient boosting algorithm. Due to the natural discrete form of C-index, the authors proposed a smoothed version of this statistic, which is given by approximating the indicator function $I(\eta_j > \eta_i)$ by sigmoid function.

$$\text{sig}(\eta_j - \eta_i) = \frac{1}{1 + \exp(-\frac{\eta_j - \eta_i}{\sigma})}.$$

It is also assumed that the survival probability is estimated non-parametrically by a Kaplan-Meier curve $K(.)$. The smoothed C-index is a continuous function of time T and prognostic

index η, which has mathematical form

$$\hat{C}_{smt}(t,\eta) = \frac{\sum_{i,j} \frac{\delta_j}{K(t_j)^2} I(t_j < t_i)\text{sig}(\eta_j - \eta_i)}{\sum_{i,j} \frac{\delta_j}{K(t_j)^2} I(t_j < t_i)} \tag{3}$$

In the same study, the authors achieved a model with the best discriminatory C-index by altering the loss function $l(\mathbf{w})$ in Algorithm 4 by the negative of equation 3 . However, we will later show that in the trade-off of the choice of loss function, the Brier's score of (smooth) C-index is outperformed by gradient boosting Brier's score.

3.5 Gradient Boosting via Optimization of the Gehan Loss

Algorithm 5 Component-wise Gradient Boosting using rank-based Gehan loss.

1. Initialize $\hat{\mathbf{w}}^{[0]} = (0, ..., 0)$ and the learning rate $0 \leq \nu \leq 1$.
2. For $m = 1$ to M, compute:

$$\mathbf{u}^{[m]} = -\frac{\partial}{\partial \mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \left[\frac{1}{n} \delta_i \sum_{j=1}^{n} \big(r_i(\mathbf{w}) - r_j(\mathbf{w})\big) I\big(r_i(\mathbf{w}) \leq r_j(\mathbf{w})\big) \right] \Big|_{\mathbf{w}=\hat{\mathbf{w}}^{[m-1]}}$$

3. For $\mathbf{z}_j$ in $Z = \{\mathbf{z}_1, ..., \mathbf{z}_p\}$:
   (a) Fit base-learners $b(\mathbf{z}_j, \theta_j)$ with response variable $\mathbf{u}^{[m]}$ and obtain the least squared estimator $\hat{\theta}_j$.
   (b) Select an index k in $\{1, ..., p\}$, such that:

$$k = \arg\min_j \sum_{i=1}^{n} (\mathbf{u}^{[m]} - \mathbf{z}_j^T \hat{\theta}_j)^2$$

4. Update $\mathbf{w}^{[m]} = \mathbf{w}^{[m-1]} + \nu \hat{\theta}_k$

The proportional hazards assumption is vital for above algorithms, since the final models will be Cox regression model. However this assumption is often neglected, thus AFT model can be a promising alternatives for Cox proportional hazard model. There are a well-known method proposed to approach boosting AFT model, namely Hothorn et al. (2006) using the

inverse-probability weighting (IPW). Nevertheless, Johnson and Q.Long et al. (2011) [23] proposed tree-boosting rank-based Gehan loss for semi-accelerated failure time model, which defines as the weighted sum of pairwise differences. Let O_i be the fully observed failure time, we have

$$\Phi(\delta, \mathbf{w}) = -\frac{1}{n}\sum_{i=1}^{n}\left[-\frac{1}{n}\delta_i \sum_{j=1}^{n}\big(r_i(\mathbf{w}) - r_j(\mathbf{w})\big) I\big(r_i(\mathbf{w}) \leq r_j(\mathbf{w})\big)\right],$$

where $r_i(\mathbf{w}) = \log O_i - \mathbf{w}^T\mathbf{z}_i$.

We employed the Gehan rank-based gradient boosting in our research, due to several reasons, which will be address in chapter 4.

4 Gradient Boosting via Optimization of Modified Brier Score

The Brier score is developed to evaluate the accuracy of probabilistic predictions in survival analysis. The response of survival analysis is often binary (death or alive). The Brier score measured the squared discrepancy between the response and the predicted survival probability. A better model should have a lower Brier score. Therefore the objective of each boosting step is to minimize the modified Brier score, which is defined as

$$\mathbf{B}(\mathbf{w}) = \Phi(\delta_i, f(\mathbf{z}_i, \mathbf{w})) = \sum_{i=1}^{n}(\delta_i - S(t|\mathbf{z}))^2 = \sum_{i=1}^{n}\left(\delta_i - S_0(t)^{exp(\mathbf{z}_i^T\mathbf{w})}\right)^2. \tag{4}$$

The modified version neglects the constant term $1/n$ in the original version, which will not affect the optimization. The gradient vector of the i^{th} instance is given by

$$\mathbf{u}_i = \frac{\partial\Phi(\delta_i, f(\mathbf{z}_i, w))}{\partial f(\mathbf{z}_i, w)} = \frac{\partial\mathbf{B}(\mathbf{w})}{\partial\mathbf{w}}.$$

By taking partial derivatives of the components $w_j, j = 1, ..., p$ of the coefficient vector $\mathbf{w}$, we have

$$u_{ij} = \frac{\partial\mathbf{B}(\mathbf{w})}{\partial w_i} = -2z_{ij}^T\log(S_0)\left[\delta_i - S_0(t)^{\exp(\mathbf{z}_i^T\mathbf{w})}\right]\exp(\mathbf{z}_i^T\mathbf{w})S_0(t)^{\exp(\mathbf{z}_i^T\mathbf{w})}.$$

Unlike the above algorithms, here the target of boosting is not the prognostic index, but instead being the survival probability. Thus, the form of first order derivative is more

Algorithm 6 Component-wise Gradient Boosting using Brier's score as the loss function

1. Initialize $\mathbf{w}^{[0]} = (0, ..., 0)$ and learning rate $0 \leq \nu \leq 1$.
2. For $m = 1$ to M
   (a) Compute the gradient vector:

$$u_{ij} = \frac{\partial \Phi(\delta_i, f(\mathbf{z}_i, \mathbf{w}))}{\partial f(\mathbf{z}_i, \mathbf{w})}$$

$$= -2z_{ij}^T \log(S_0) \left[\delta_i - S_0(t)^{\exp(\mathbf{z}_i^T \mathbf{w})} \right] \exp(\mathbf{z}_i^T \mathbf{w}) S_0(t)^{\exp(\mathbf{z}_i^T \mathbf{w})}. \Big|_{\mathbf{w}=\mathbf{w}^{[m-1]}}$$

   (b) Compute all possible updates of weight vector $\mathbf{w}^{[m]}$ using least square method:

$$\hat{\theta}_j = (\mathbf{z}_j^T \mathbf{z}_j)^{-1} \mathbf{z}_j^T \mathbf{u}_j$$

   (c) Find the best update k, which is

$$k = \arg\min_j \sum_{i=1}^{n} (\mathbf{u}_i - \mathbf{z}_j^T \hat{\theta}_j)^2$$

   (d) Update

$$\mathbf{w}_k^{[m]} = \mathbf{w}_k^{[m-1]} + \nu \hat{\theta}_k$$

3. Output: Weight vector $\mathbf{w}^{[M]}$

---

complicated. The variable selection is performed via the component-wise approach. For the i^{th} observation, we obtain a gradient vector u_{ij} of length p. Thus, at each iteration, a gradient matrix of dimension $n \times p$ is produced, which can be written as :

$$\begin{bmatrix} z_{11} & ... & z_{1p} \\ z_{21} & ... & z_{2p} \\ & & \\ z_{n1} & ... & z_{np} \end{bmatrix}_{n \times p} \begin{bmatrix} w_{11} & ... & w_{1p} \\ w_{21} & ... & w_{2p} \\ & & \\ w_{p1} & ... & w_{pp} \end{bmatrix}_{p \times p} = \begin{bmatrix} u_{11} & ... & u_{1p} \\ u_{21} & ... & u_{2p} \\ & & \\ u_{n1} & ... & u_{np} \end{bmatrix}_{n \times p}$$

Let $\mathbf{z}_i$, $\mathbf{w}_i$ and $\mathbf{u}_i$ be column vectors of length n in above matrix. We can rewrite the matrices

above as partition matrices in the form of

$$\left[\mathbf{z}_1| \quad ... \quad |\mathbf{z}_p\right] \left[\mathbf{w}_1| \quad ... \quad |\mathbf{w}_p\right] = \left[\mathbf{u}_1| \quad ... \quad |\mathbf{u}_p\right]$$

The component-wise routine is used to fit the base learner on response $\mathbf{u}_i$ and predictor $\mathbf{z}_i$, reproducing all possible updates on the set of weights.

In the later section, it is found that the algorithm outperforms its competitors in terns of minimizing the Brier score, both in training and testing cohorts.

5 Parametric Analysis for the DLBCL data

5.1 Kaplan-Meier Estimate of the Survival Function

The Kaplan-Meier curves provides us a general view on the data set. Within first 5 years, the baseline survival probability drops significantly to 46.7%. In the next 5 years, the rate is continuing drop but gradually instead, to 34.6% by 10.6 years. Patients, who have been survived beyond year 11^{th} possess a survival chance of 31.8%. A first glance at the histogram of survival time (in year) (Figure 5.1), we proposed that an exponential distribution would be the best fit for the baseline survival distribution. However, the analysis thereafter will include exponential distribution, as well as the Weibull 2 parameters, log-normal and log-logistic. We found that the log-logistics model is the best-fitted distribution the baseline survival time. (Table 5.1)

5.2 Parametric Estimate of the Survival Function

Table 5.1: Results of parametric models on DLBCL survival time with comparing metric of AIC information criteria.

	Exponential	Weibull	Log-normal	Log-logistics
Number of parameters	1	2	2	2
AIC	624.97	571.77	582.54	566.54

The probability density function of Log-logistics distribution (Fisk distribution) is given

by:

$$f(t;\alpha,\beta) = \frac{(\beta/\alpha)(t/\alpha)^{\beta-1}}{(1+(t/\alpha)^{\beta})^2}, t \geq 0$$

The scale parameter α is a positive number and also median value of the distribution. On the other hand, shape parameter $\beta < 1$ indicates that there does not exit unimodality in the model. As a consequence, the expected life time is undefined. The cumulative distribution function is

$$F(t;\alpha,\beta) = \frac{1}{1+(t/\alpha)^{-\beta}}$$

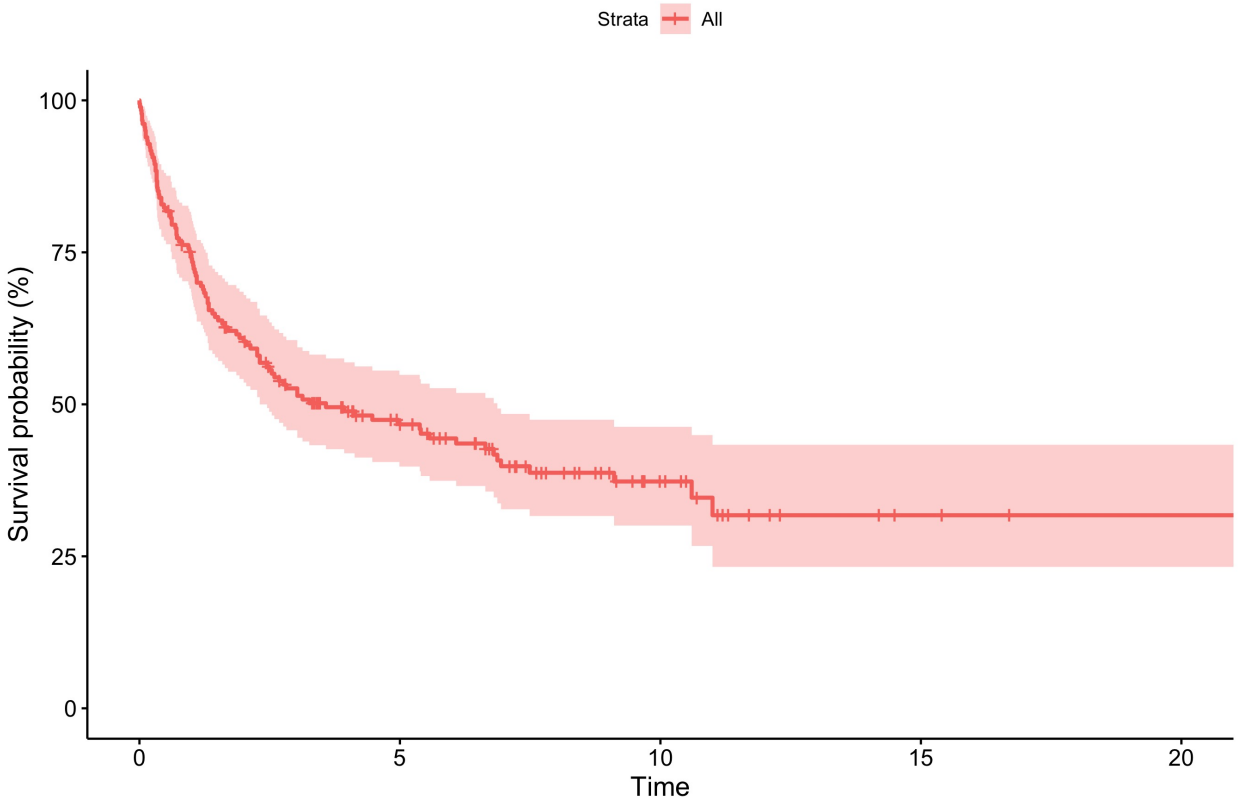

Figure 5.1: Kaplan-Meier product-limit estimate of survival time with 95% confident interval (shaded region)

By using maximum likelihood estimation and Newton-Raphson algorithm [41] implemented in R-studio, MLEs of scale and shape parameters are $\hat{\alpha} = 4.071465$ and $\hat{\beta} = 0.715382$, respectively.

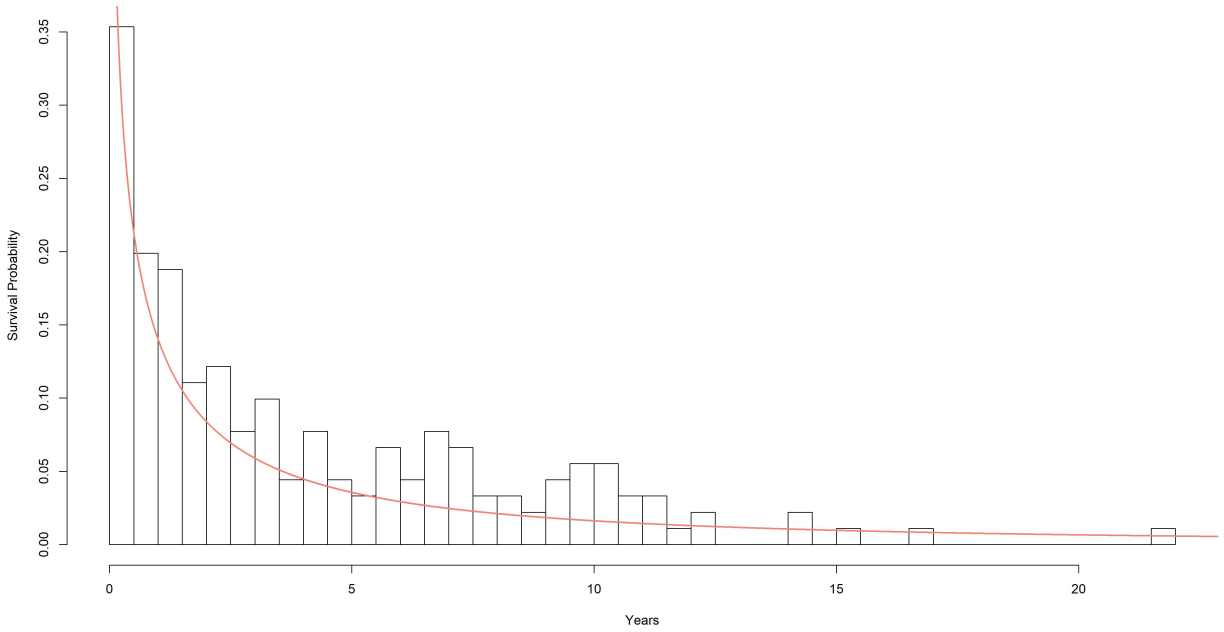

Figure 5.2: Histogram and probability density function (PDF) of log-logistic fit on the survival time.

As a result, the density distribution of survival time has the form

$$f(t) = \frac{0.1757(t/4.07165)^{0.284618}}{(1 + (t/4.071465)^{0.715382})^2}, t > 0.$$

Figure 5.3 illustrates that the assumed density curve has long fat tail and positive skewness. The majority of area under the curve (71.77%) is within 10 years, i.e $P(T \leq 10) = 0.7177$. The medical interest is often to estimate the survival rate of a patient with first 5 years, which is

$$P\{\text{A patient survives up to 5 years}\} = P(T \leq 5) = 0.5367.$$

Noticeably, the survival rate between year 5^{th} to year 10^{th} drops to under 11.8718%, given that this patient has survived beyond year 5. Thus, we can classify patients surviving in this period as high risk group, in contrast to who are still alive up to 5 years (Figure 5.3).

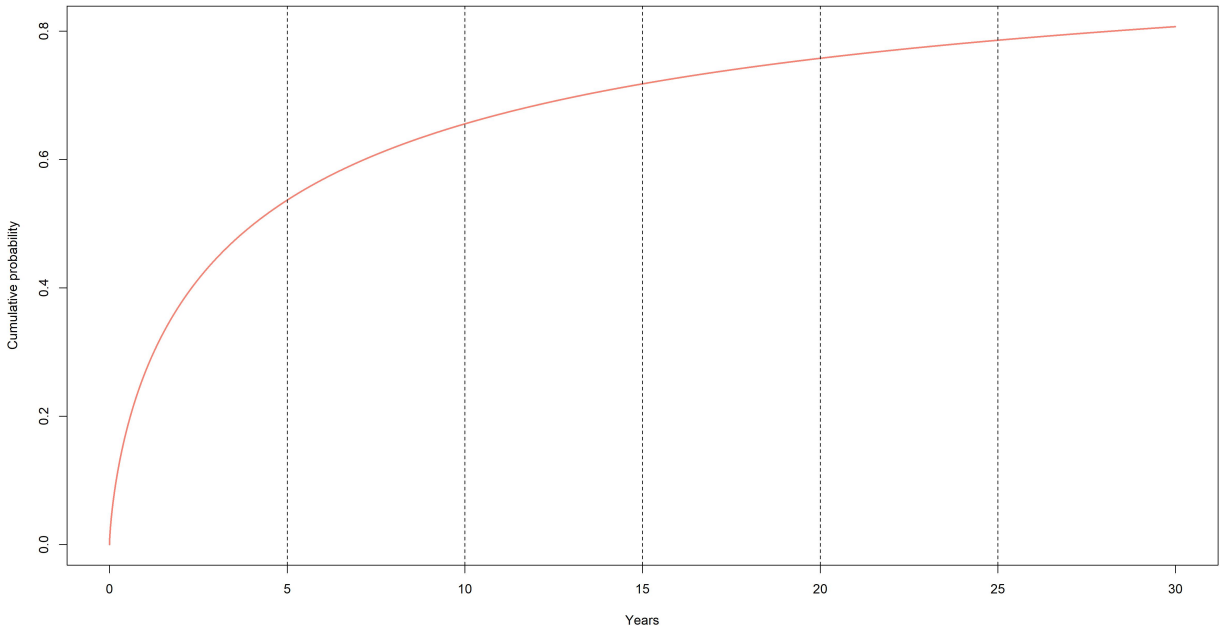

Figure 5.3: Corresponding cumulative distribution function (CDF) of log-logistic fit on the survival time

5.3 Boosted Tree Classification Model

In the next part of this chapter, we will validate the proportional hazards assumption. Using tests and plots from the scaled Schoenfeld residuals. The violation of the proportional hazard assumption is represented by a non-zero slope. Therefore, any nonlinear relationship, which is discovered in the plot of the residuals as a function of time, can indicate the violation of proportional hazard assumption. Additionally, in case of high dimensional feature space, [42] proposed that covariates with most importance score (best separate alive and deceased patients) can be used to validate the assumption. Firstly, we create a tree-based classifier which implement the Extreme Gradient Boost (XGBoost) [38] with the binary logistic function as the loss function. To perform the classifying process, we create a grid search for hyper-parameters, which include the learning rate ν and maximum depth of a tree T.

It is shown that the stopping criteria is highly related to the learning rate, thus we

Table 5.2: First 20 covariates from the Cox's proportional hazard test, with Chi-square statistic and p-value.

	ρ	χ^2	p-value
X217320_at	0.03408516	0.61760805	0.43193754
X242936_at	−0.04952279	1.52970279	0.21615706
X1553607_at	−0.14325457	10.14971603	0.00144323
X238498_at	0.06975792	1.51952073	0.21769203
X1561316_at	−0.19253580	18.86935049	0.00001400
X233046_at	0.01307213	0.05749129	0.81050593
X238845_at	0.05242758	2.03359225	0.15385658
X1555939_at	−0.02398130	0.34493307	0.55699551
X229839_at	0.06160926	2.06830635	0.15038845
X1568574_x_at	−0.11670569	4.24673681	0.03932583
X1560025_at	0.12852035	8.58176531	0.00339546
X232947_at	−0.22203410	28.38298975	0.00000010
X241235_at	−0.17320187	20.68739285	0.00000541
X224221_s_at	−0.09565856	6.28885046	0.01214998
X1568751_at	−0.04222970	1.48853993	0.22244314
X1562309_s_at	−0.01698123	0.13654472	0.71174048
X244546_at	−0.06535564	2.13214249	0.14423902
X210600_s_at	−0.07831158	3.69695166	0.05451197
X1561140_at	−0.10121657	4.06587337	0.04375805
X231478_at	−0.02512684	0.21626282	0.64190234
GLOBAL		180.64060816	0.00000142

can choose an arbitrary value of learning rate between 0 and 1. The algorithm separates two classes remarkably well, reaching 82.32% of accuracy in leave-one-out study. At each fold, an importance matrix is obtained by measuring the frequency of chosen features. One thing worth mentioning is that the feature selection is stable, which means the algorithm deriving the same importance matrix at each fold. In (Figure 5.4), the bottom panel shows the importance of a covariates based on its selected frequency, whereas the top panel in the relative importance defined by the fraction of frequency of a covariate over the most importance covariate (X_217320_at). Secondly, we fit the Cox's regression model on the top 100 covariates to assess graphical results from Schoenfeld residuals as a function time, as well as the goodness-of-fit test. Table 5.2 shows first 10 results (out of 100) of the goodness-of-fit test on proportional assumption. As we can see, there actually are several probe sets

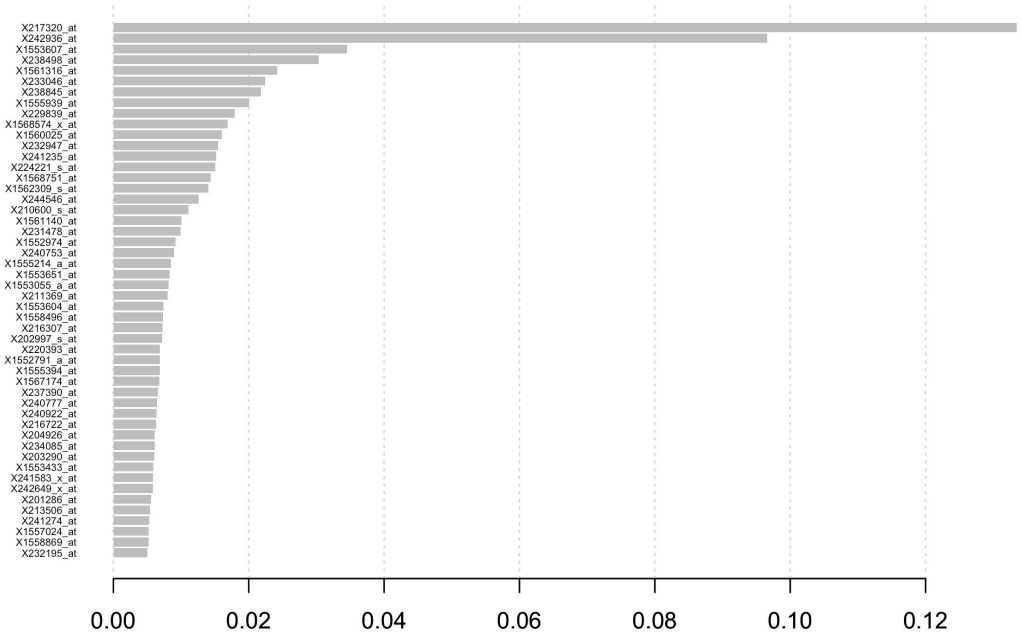

Figure 5.4: Importance ranked by the frequency of to be chosen by XGboost algorithm, ordered from top to bottom.

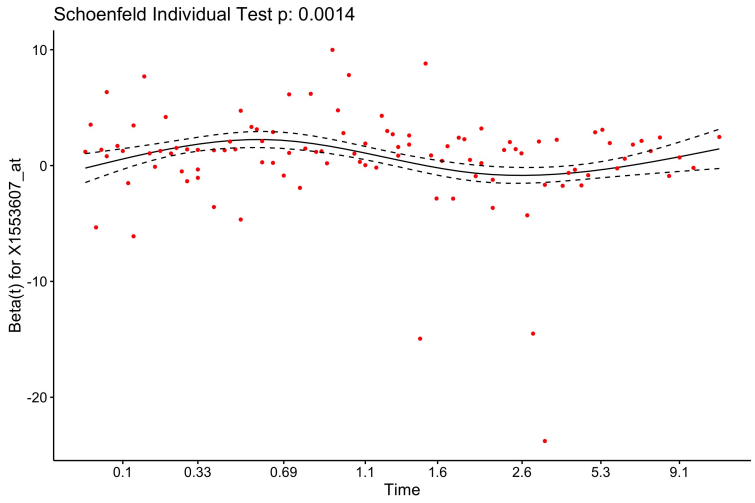

Figure 5.5: Schoenfeld Residuals Test Results: The solid lines smooth the spread of Schoenfeld residuals in neighborhood of 0.

satisfying the proportional hazard assumption, such as X217320_at, X242936_at, X238498_at,

Et cetera.

However, the global test indicates the violation of proportional hazard assumption, $\chi^2 = 180.64$ at significance of 0.00000142. Furthermore, the plots of Schoenfeld residuals as a function time again indicate non-proportionality, where non-linear relationship appears in plots of X1553697_at and X1561316_at.

6 Simulation Study

A simulated study was conducted in order to evaluate and compare boosting methods. For simplicity, the simulated data ($n = 1000$) is assumed to be exponentially distributed, due to the constant hazard. We also set the ground truth, which contains the first five active covariates with coefficients of $0.5(\{\beta_i\}_{i=1}^{5} = 0.5)$ and the remaining 995 is non-active covariates. The baseline PDF of survival time (Figure 6.1) is generated from

$$f(t) = 0.5e^{-0.5t}$$

and

$$\log\left[\frac{\lambda(t|Z_i)}{\lambda_0(t)}\right] = 0.5 \times z_1 + 0.5 \times z_2 + 0.5 \times z_3 + 0.5 \times z_4 + 0.5 \times z_5$$

First of all, we repeat the parametric analysis to obtain the baseline distribution of survival time. The estimated distribution of baseline survival probability is

$$\hat{S}_0(t) = e^{-0.508517t}$$

Since the data is simulated, we are certain that the proportional hazard is satisfied. One of the most versatile methods for survival analysis is Lasso Cox regression (L-Cox). Based on L_1-norm penalty, L-Cox shrinks the coefficients of non-significant covariates exactly equal to zero, performing feature selection by finding minimum penalty. We performed 5-fold cross

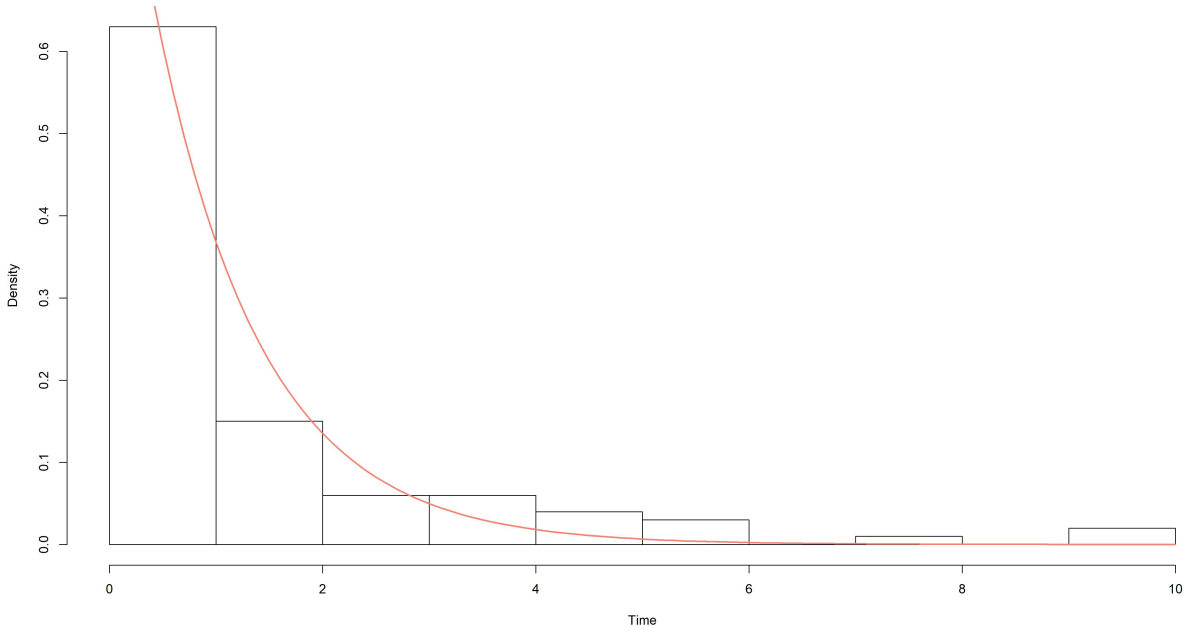

Figure 6.1: Baseline survival probability density function of simulated data, using exponential survival function.

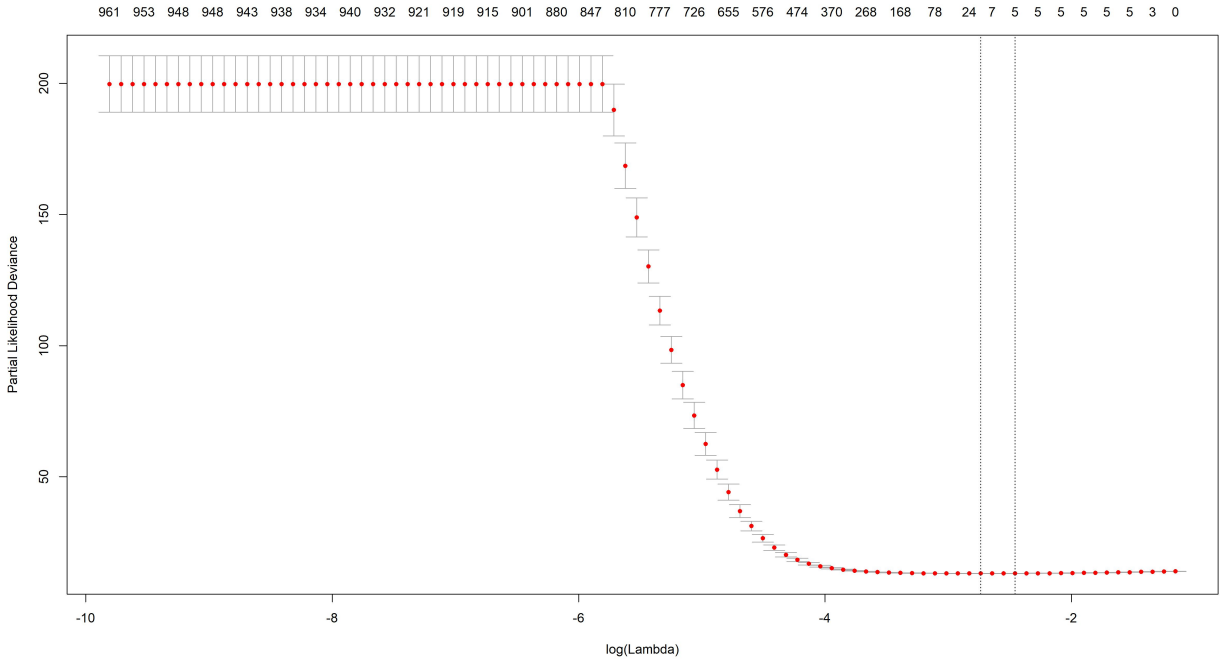

Figure 6.2: 5-fold cross-validation of L-Cox model, using partial likelihood deviance as scoring criteria for parameter tuning ($\log(\lambda)$)

Table 6.1: Comparison of gradient boosting algorithms in large simulated data ($p = 1000$).

	L-Cox	gbL	gbC	gbG	gbB
Number of selected features	10(5)	10(5)	11(5)	5(5)	12(5)
Train Brier's score	0.5322	0.5642	0.6723	0.9063	0.2763
Test Brier's score	0.9458	1.2834	1.3385	1.5032	0.5398
Test C-index	0.7112	0.6908	0.8977	0.6811	0.6673

validation on the simulated data, in order to obtain the best penalty λ, which minimizes the partial likelihood deviance. Figure 6.2 shows that $\lambda = 0.06479103$ gives us the best L-Cox model with 10 selected covariates. 5 out of 10 selected covariates are correct and relatively closed to the ground truth.

Table 6.2: Comparison of gradient boosting algorithms in smaller simulated data ($p = 10$).

	L-Cox	gbL	gbC	gbG	gbB
Train Brier's score	1.2036	1.4108	1.7166	1.9862	0.5589
Test Brier's score	2.0143	2.4234	2.6523	2.9087	0.9887
Test C-index	0.6267	0.5802	0.7419	0.5792	0.5017

Table 6.3: Estimate of coefficients from simulated data with optimally tuned parameter by implementation of grid search.

	β_1	β_2	β_3	β_4	β_5
gbL	0.405619	0.197475	0.150577	0.479283	0.501966
gbG	0.457558	0.195512	0.115215	0.562434	0.547816
L-Cox	0.460840	0.252616	0.208977	0.529942	0.553262
gbB	0.487321	0.398092	0.467893	0.367896	0.401365
gbC	0.384246	0.492172	0.501973	0.439576	0.497782

We suspect that the number of non-active covariates ($p = 995$) is the reason for the over-fitted number of covariates. Thus, we created a smaller data set with only 10 covariates. The ground truth is remained the same as the first simulated data. We now apply four gradient boosting algorithms on the simulated data, which are gradient boosting via optimization of partial log-likelihood (gbL), modified Brier score (gbB), smoothed C-index and Gehan loss function (gbG). As a result, all of the algorithms select the correct set of active covariates. In Table 6.1, gbC produced the best C-index, while the remaining algorithms have a slightly smaller C-index. However, the Brier's score of gbC is significant higher than gbB algorithm,

which is only 0.2763 and 0.5398 in train and test set, respectively. On the other hand, gbG selected the exact number of active covariates, while the other algorithms seem to be over-fitted. Table 6.2 is the results of applying these gradent boosting on the smaller data set. Again, the gbC has the largest C-index of 0.7419, significant higher than other. Nevertheless, gbB outperforms other algorithms with aspect of minimizing the Brier's score, achieving the smallest Brier's score of 0.5017 in test set and 0.9887 in train set.

7 Gradient Boosting for Analyzing the DLBCL data

We compare four methods based on the component-wise gradient boosting algorithms, which include the gradient boosting via optimization of the partial log-likelihood (gbL), the modified Brier score (gbB), the smoothed C-index and the Gehan loss function (gbG). To avoid overfitting, we performed 5-fold cross-validation with a log-logistic baseline, to obtain the average training and testing errors for both the Brier score and the concordance C-index. (Table 7.1).

First of all, a gbB model is built on entire data set, to select 211 most important covariates, then the model with selected covariates is used to perform the cross validation. Based of the modified Brier's score loss, the gbB outperforms the remaining algorithm, with the smallest test error of 0.2861, in compare to 0.5230 and 0.9726 of gbL and gbG, respectively. On the other hand, the log-likelihood of gbB is higher than those of gbL and gbG. Additionally, the C-index of the gbB is 0.7402, which is remarkably smaller than C-indices of the gbL and gbG (0.92) approaches. The stability of feature selection of gbG is outstanding, since the algorithm picks up the same number of probe sets (204) after 1000 iterations and the in-bag log-likelihood converges at approximately 11.648. On the other hand, gbL does not seem to be converging on the log-likelihood, which has resulted in a continuous increase in number of parameters as the number of boosting steps increases (*). Thus, we decided to use only results produced by gbB, gbC and gbG for further analysis.

The two algorithms select a set of 20 common covariates, which includes four probe sets

Table 7.1: Comparison of gradient boosting algorithms with Brier's score as train-test cross validation.

	gbL	gbB	gbG	gbC
Number of selected features	176-295*	211	204	220
Train Brier's score	0.5230	0.0933	0.5379	0.6011
Test Brier's score	0.8466	0.2861	0.9726	0.8473
Test C-index	0.9106	0.7402	0.9173	0.9711

identified by the twin boosting from [1].

The probe sets selected seem to be matching with medical facts after consulting with subject matter experts. First of all, we would like to briefly interpret the biological relevance of some common probe sets selected by the two algorithms. The most remarkable probe set found is 239672_at, which possesses gene symbol BLID. A BH3-like motif, which is encoded by this gene, contains protein associated to cell death. Localizing in both the mitochondrion and cytoplasm, the protein may be reasonable for apoptosis. The cell death inducer BH3-like Motif Containing and breast cancer cell protein are aliases for BLID. Besides, HS6ST2 (1552766_at) is heparan sulfate proteoglycans, which plays an important role in cell growth and migration. Likewise, ADRA1A (237390_at), alpha-1-adrenergic receptors, create mitogenic responses and regulate growth, as well as induce cell's proliferation. The gene may bleed prostatic hypertrophy, which strongly related to prostate cancer. As we know, cancers are diseases related to abnormal growth of cells. Hence, it is not surprising our results indicate the two above genes might impact survival probability of a patient. Additionally, probe set 155899_x_at, which has gene symbol LOC283922/PDPR, is represented as Pyruvate Dehydrogenase Phosphatase Regulatory Subunit Pseudogene. This probe set was also found in [1] with negative coefficient. SLC17A1 gene (1560884_at) is also a protein coding gene, which might be a reason for gout and hyperuricemia. Its pathways are transportation of glucose and other compounds like sugars, salts, metal ions and acids. 237797_at is DNM1L gene, which encodes an element of the dynamin superfamily (or known as protein coding gene). In case of dysfunction, several neurological disorders might be induced, such as Alzheimer's disorder. Some diseases related to DNLM1L are encephalopathy induced by

mitochondrial defection and peroxisomal fission 1. The pathways of DNLM1L include CDK-mediated phosphorylation and removal Cdc6, the same as ABCA13 (1553605_a_at), which is also discovered by gbB and gbG. Schizophrenia is also mental disease, induced by ABCA13.

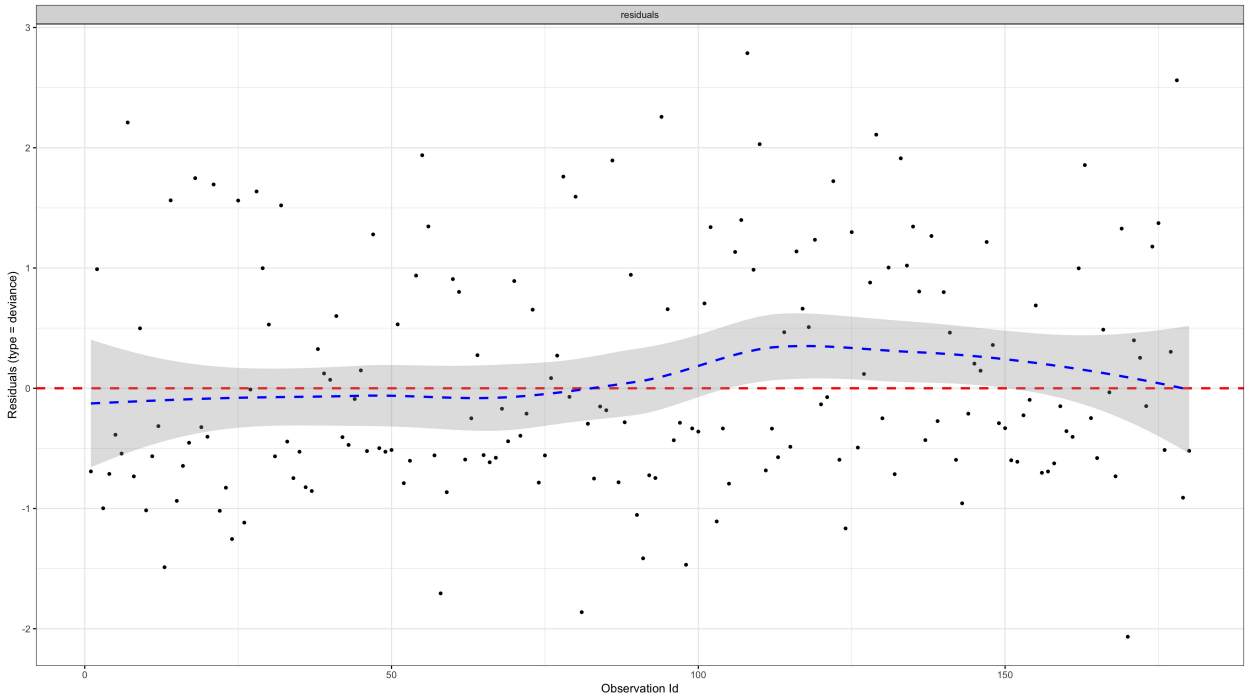

Figure 7.1: Plot of deviance residuals with smoothed line indicating the spread around zero (shaded region is 95% confident interval.)

We now conduct model diagnostics on the gbB model, which includes 220 probe sets. First of all, we validate the proportional hazards assumption by using the Schoenfeld residuals test. The test result in a test statistic $\chi^2 = 7.41$ with a global significance level of 0.0065, which is not highly practically significant based on our experience. Thus, there seems no strong evidence of non-proportional hazards. Second, as we discussed in the previous section, the gbB algorithm is expected to outperfom other algorithms when the Brier score is selected as the comparison criterion. Particularly, the gbB model achieves a Brier's score of 0.03524, which is remarkably small among existing competitors. In addition, the C-index from the gbB model is 0.745, which is also considerably smaller than the C-indices of the gbL and gbG models. In addition, the plot of deviance residuals as shown in Figure 7.1 is generally

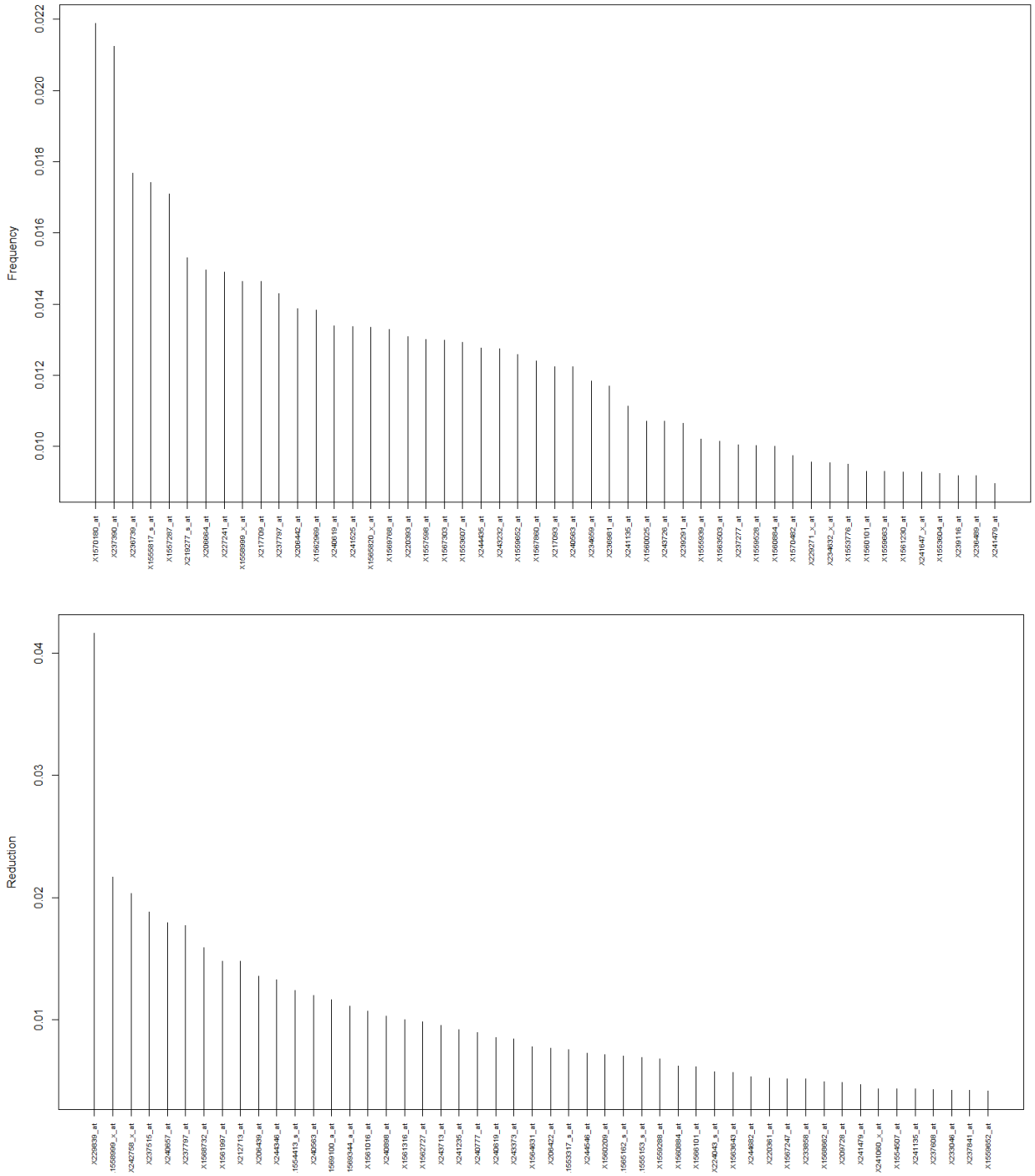

Figure 7.2: Importance plot of top 50 probe sets. Top panel: gbB algorithm, bottom panel: gbG algorithm.

symmetric around the horizontal line at $x = 0$. In addition, 7 out of 180 might be considered as influential observations, since the absolute values of deviance excess 2.

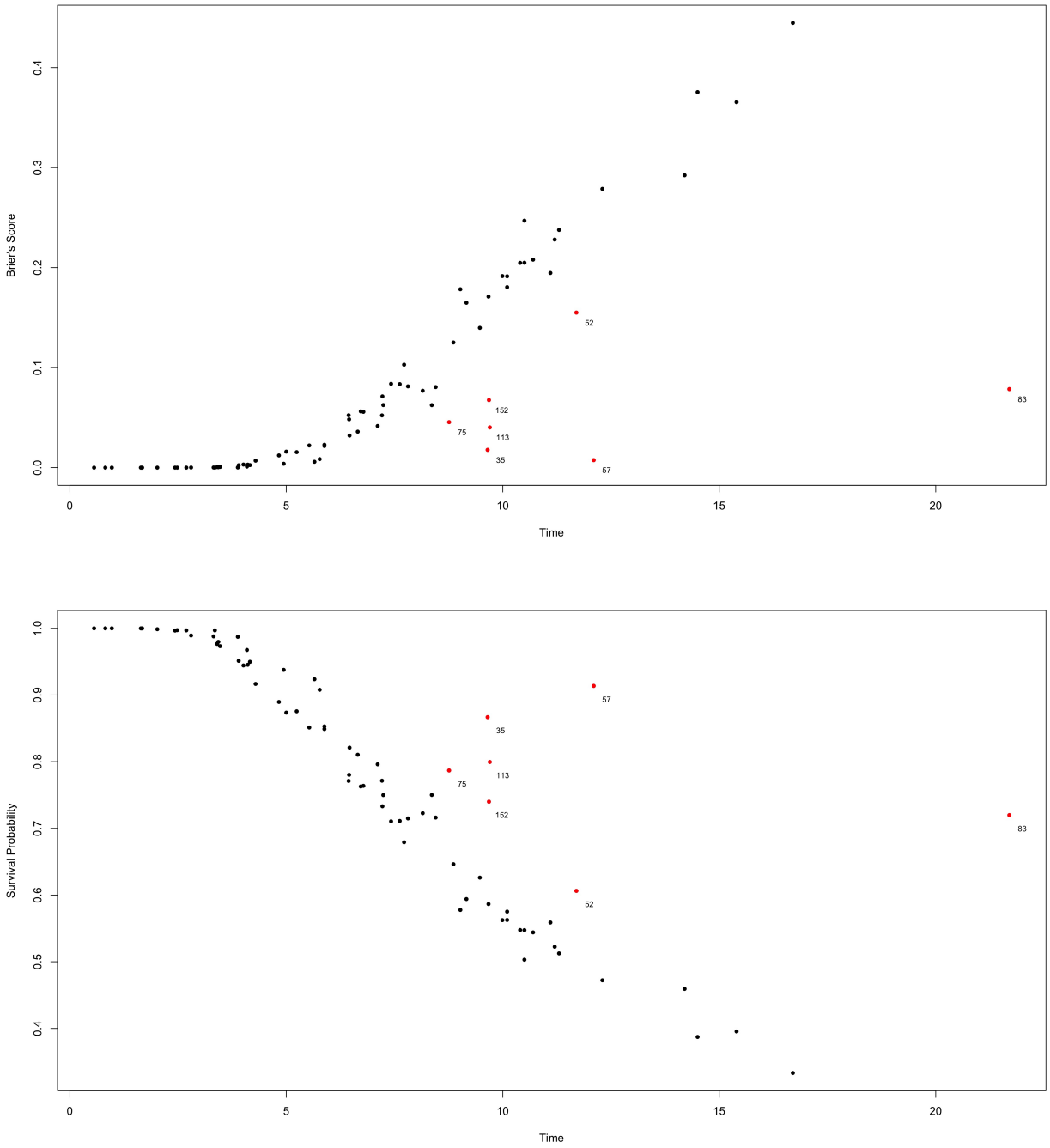

Figure 7.3: Top panel: Brier's score of patients at risk. Bottom panel: Predicted survival probability of patients at risk.

The importance matrices of the covariates from the gbG and gbB methods are evaluated based on the frequency that a covariate is selected by the algorithms. As we can see, the probe set X_229839_at is the dominant key variable in the importance matrix of the gbG

algorithm. While in the gbB importance matrix, the dominant probe sets are X_1570180_at and X_237390_at. What is worthy mentioning is that probe set X_1558999_x_at, which appeared [1] in the top ten important covariates from both algorithms.

Figure 7.3 illustrates the survival probability of patients at risk and time (in months) and the corresponding Brier score. The first impression from the plot is that the survival probability within the first 5 years is very high, above 0.8736. After that it gradually decreases, which matches the inherent nature of the survival probability. Including the covariates was helpful for improving the prediction of survival experience. On the other hand, the Brier scores of cases with the occurrence of events observed in the first 5 years are relatively small (below 0.03), indicating there is higher prediction efficiency for earlier failures. However, there are still a few observations (such as patients 35, 52, 57, 75, 83, 113 and 152) with relatively large Brier score, which indicates some important probe sets might still be missing from the fitted model for predicting the survival experience of these group patients.

8 Concluding Remarks

This thesis has applied a number of component-wise gradient boosting algorithms for analyzing the DLBCL data. In addition, it has developed a new gradient boosting algorithm based on optimizing the Brier score, and demonstrate the new method outperforms many existing boosting algorithms on multiple reliability metrics through a simulation study. Although the gradient boosting algorithms are built upon statistical models, they outperform the traditional statistical methods when exploring high dimensional feature space by leveraging the computing power of the machine learning algorithms. The new gradient boosting method is demonstrated to minimize the predictive Brier score and the C-Index, as well as selects the correct features from the simulation study. In addition, the new boosting algorithm also selected a set of covariates from the DLBCL data, which matches with medical or biological facts and offers insights on a deeper understanding of diffuse large B-cell lymphoma.

The implementation of gradient boosting based on optimizing the Brier score with the XGboost algorithm has a great potential to further improve the computational efficiency and accuracy, as the XGboost is much more powerful boosting algorithm than the gradient boosting. By taking into account the Hessian matrix of the loss function, the future study on applying XGboosting for optimizing the Brier score can be both challenging and also highly promising. Another objective of the future study is to create a R package for implementing the gradient boosting and the XGboosting with the Brier score, which can facilitate broad

applications of the new methods in survival analysis of cancer and other medical data.

References

[1] Zhu Wang and CY Wang. Buckley-james boosting for survival analysis with high-dimensional biomarker data. *Statistical Applications in Genetics and Molecular Biology*, 9(1), 2010.

[2] NCI. Cancer fact and statistics. *National Cancer Institute*, 2019.

[3] E.L Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *J. Amer. Statist. Assoc*, 1958.

[4] John P Klein. Small sample moments of some estimators of the variance of the kaplan-meier and nelson-aalen estimators. *Scandinavian Journal of Statistics*, pages 333–340, 1991.

[5] Sidney J Cutler and Fred Ederer. Maximum utilization of the life table method in analyzing survival. *Journal of chronic diseases*, 8(6):699–712, 1958.

[6] David R Cox. The cox proportional hazards model. 1972.

[7] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of statistical software*, 39(5):1, 2011.

[8] Tim Hesterberg, Nam Hee Choi, Lukas Meier, Chris Fraley, et al. Least angle and 1 penalized regression: A review. *Statistics Surveys*, 2:61–93, 2008.

[9] Lloyd D Fisher and Danyu Y Lin. Time-dependent covariates in the cox proportional-hazards regression model. *Annual review of public health*, 20(1):145–157, 1999.

[10] Simon N Wood and Nicole H Augustin. Gams with integrated model selection using penalized regression splines and applications to environmental modelling. *Ecological modelling*, 157(2-3):157–177, 2002.

[11] Matthias Schmid and Torsten Hothorn. Flexible boosting of accelerated failure time models. *BMC bioinformatics*, 9(1):269, 2008.

[12] Louis Gordon and Richard A Olshen. Tree-structured survival analysis. *Cancer treatment reports*, 69(10):1065–1069, 1985.

[13] Larry W Kwak, Jerry Halpern, Richard A Olshen, and Sandra J Horning. Prognostic significance of actual dose intensity in diffuse large-cell lymphoma: results of a tree-structured survival analysis. *Journal of Clinical Oncology*, 8(6):963–977, 1990.

[14] G Landoni, T Greco, G Biondi-Zoccai, C Nigro Neto, D Febres, M Pintaudi, L Pasin, L Cabrini, Gabriele Finco, and A Zangrillo. Anaesthetic drugs and survival: a bayesian network meta-analysis of randomized trials in cardiac surgery. *British journal of anaesthesia*, 111(6):886–896, 2013.

[15] Elia Biganzoli, Patrizia Boracchi, Luigi Mariani, and Ettore Marubini. Feed forward neural networks for the analysis of censored survival data: a partial logistic regression approach. *Statistics in medicine*, 17(10):1169–1186, 1998.

[16] Y-JLOL Mangasarian and WH Wolberg. Breast cancer survival and chemotherapy: a support vector machine analysis. In *Discret Math Probl with Med Appl DIMACS Work Discret Math Probl with Med Appl December 8–10, 1999, Volume 55*, page 1. 2000.

[17] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, Michael S Lauer, et al. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

[18] Torsten Hothorn, Berthold Lausen, Axel Benner, and Martin Radespiel-Tröger. Bagging survival trees. *Statistics in medicine*, 23(1):77–91, 2004.

[19] Torsten Hothorn, Peter Bühlmann, Sandrine Dudoit, Annette Molinaro, and Mark J Van Der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2005.

[20] Greg Ridgeway. The state of boosting. *Computing Science and Statistics*, pages 172–181, 1999.

[21] Rahul Paul, Samuel H Hawkins, Yoganand Balagurunathan, Matthew B Schabath, Robert J Gillies, Lawrence O Hall, and Dmitry B Goldgof. Deep feature transfer learning in combination with traditional features predicts survival among patients with lung adenocarcinoma. *Tomography*, 2(4):388, 2016.

[22] Yan Li, Jie Wang, Jieping Ye, and Chandan K Reddy. A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1715–1724. ACM, 2016.

[23] Brent A. Johnson and Qi Long. Survival ensembles by the sum of pairwise differences with application to lung cancer microarray studies. *The Annals of Applied Statistics*, Aug 2011.

[24] Yifei Chen, Zhenyu Jia, Dan Mercola, and Xiaohui Xie. A gradient boosting algorithm for survival analysis via direct optimization of concordance index. *Computational and mathematical methods in medicine*, 2013, 2013.

[25] G Lenz, G Wright, and S.S Dave. Stromal gene signatures in large-b-cell lymphomas. *The New England Journal of Medicine*, Nov 2008.

[26] Andreas Rosenwald, George Wright, Wing C Chan, Joseph M Connors, Elias Campo, Richard I Fisher, Randy D Gascoyne, H Konrad Muller-Hermelink, Erlend B Smeland, Jena M Giltnane, et al. The use of molecular profiling to predict survival after

chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346(25):1937–1947, 2002.

[27] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society*, 1972.

[28] Anastasios A Tsiatis et al. A large sample study of cox's regression model. *The Annals of Statistics*, 9(1):93–108, 1981.

[29] Per Kragh Andersen and Richard D Gill. Cox's regression model for counting processes: a large sample study. *The annals of statistics*, pages 1100–1120, 1982.

[30] David R Cox. Some remarks on the analysis of survival data. *The First Seattle Symposium of Biostatistics: Survival Analysis*, 1997.

[31] Kalbfleisch and Prentice. The statistical analysis of failure time data (2nd ed.). 2002.

[32] John D Kalbfleisch and Ross L Prentice. *The statistical analysis of failure time data*, volume 360. John Wiley & Sons, 2011.

[33] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[34] Robert E Schapire. The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer, 2003.

[35] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[36] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[37] He et. al Kevin. Component-wise gradient boosting and false discovery control in survival analysis with high-dimensional covariates. *Oxford Academic Journal*, Aug 2015.

[38] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[39] Bühlmann Peter. Boosting for high-dimensional linear models. *The Annals of Statistics*, 2006.

[40] Andreas Mayr, Harald Binder, Olaf Gefeller, and Matthias Schmid. The evolution of boosting algorithms. *Methods of information in medicine*, 53(06):419–427, 2014.

[41] Robert I Jennrich and PF Sampson. Newton-raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, 18(1):11–17, 1976.

[42] Matthias Schmid and Torsten Hothorn. Flexible boosting of accelerated failure time models. *BMC Bioinformatics*, June 2008.