Gradient Boosting Survival Tree with Applications in Credit Scoring

Miaojun Bai, Yan Zheng and Yun Shen
baimiaojun-jk, zhengyan, shenyun-jk@360jinrong.net
360 Finance Inc.

ARTICLE HISTORY
Compiled August 12, 2019

ABSTRACT

Credit scoring (Thomas et al., 2002) plays a vital role in the field of consumer finance. Survival analysis (Banasik et al., 1999) provides an advanced solution to the credit-scoring problem by quantifying the probability of survival time. In order to deal with highly heterogeneous industrial data collected in Chinese market of consumer finance, we propose a nonparametric ensemble tree model called gradient boosting survival tree (GBST) that extends the survival tree models (Gordon and Olshen, 1985; Ishwaran et al., 2008) with a gradient boosting algorithm (Friedman, 2001). The survival tree ensemble is learned by minimizing the negative log-likelihood in an additive manner. The proposed model optimizes the survival probability simultaneously for each time period, which can reduce the overall error significantly. Finally, as a test of the applicability, we apply the GBST model to quantify the credit risk with large-scale real market data. The results show that the GBST model outperforms the existing survival models measured by the concordance index (C-index), Kolmogorov-Smirnov (KS) index, as well as by the area under the receiver operating characteristic curve (AUC) of each time period.

1. Introduction

Chinese consumer finance market, with the rapid development of internet finance and e-commerce, has entered a period of market eruption in recent years. The consumer credit hit 8.45 trillion RMB, the historical high in October 2018, according to a research report released by Tsinghua University[1]. However, only approximately one third of Chinese consumers have credit ratings, in spite of the rising needs for financing, and therefore, the lack of credit infrastructure has made a critical challenge to Chinese credit providers and it calls for more advanced credit scoring techniques.

Credit scoring has been a commonly used risk measurement tool for decades (see e.g. Thomas et al. (2002) for a comprehensive introduction) and it has already been widely adopted by credit agencies all over the world. Credit scoring is traditionally solved by classification methods, e.g., logistic regression, decision tree, and more recently, gradient boosting decision tree (GBDT) and other boosting models (see e.g. XGBoost, Chen and Guestrin, 2016). Studies on the Chinese consumer finance market (see e.g. Li et al. (2019)) indicate that ensemble methods like GBDT and XGBoost outperform the traditional methods, due to the fact that credit data in Chinese market are usually high-dimensional and extremely heterogeneous, some of which have high missing rate.

Survival analysis is a statistical method to deal with the analysis of lifetime data,

---

[1] Chinese consumer credit market research 2018 (in Chinese), Institute for Chinese Economic Practice and Thinking, Tsinghua University, `https://att.dahecube.com/4a5ab7de-1aa3-4754-a4d7-6e707d943d69`.

where the time to the occurrence of an event is of particular interest. In the context of credit scoring, the event is usually defined as “default”. Therefore, the primary advantage of applying survival analysis is that it can not only model whether to default, but also predict the probability of time when default occurs. The idea to use survival analysis in the field of credit scoring was first introduced by Narain (1992). Since 2000, several studies (see e.g., Cao et al. (2009); Dirick et al. (2017) and references therein) apply survival analysis to quantify the credit risk and the results show a better performance on real credit data, which are collected by banks in developed countries like the UK and Belgium (Dirick et al., 2017), than the traditional credit scoring methods. Among others, the most widely applied model in survival analysis is the Cox proportional hazards model (Blakely and Cox, 1972) with (generalized) linear functions (Cao et al., 2009) or with splines (Dirick et al., 2017). These models may not be the best choice for high-dimensional heterogeneous credit data in Chinese market.

In the field of survival analysis, tree-structured survival models, e.g., survival trees (Gordon and Olshen, 1985) and random survival forests (RSF, Ishwaran et al., 2008) are proposed to quantify survival probabilities in medical studies. However, studies show that boosting methods outperform random forests on many classification benchmarks (see e.g. Chen and Guestrin, 2016).

This inspires us to develop a gradient boosting algorithm for survival tree models. Hence, we propose in this paper a nonparametric ensemble tree model called gradient boosting survival tree (GBST) that extends the survival tree models with a gradient boosting algorithm. The proposed model optimizes the survival probability of each time period simultaneously, which can reduce the overall error of the model significantly.

The remainder of this paper is organized as follows. Section 2 introduces the methodology of gradient boosting survival tree (GBST). We start with a brief introduction of survival analysis, followed by a calculation of negative log-likelihood loss function based on the hazard function represented by a survival tree ensemble. The loss function is optimized by a gradient boosting approach and the corresponding algorithms are derived. In Section 3, empirical experiments are conducted on the data collected in Chinese consumer finance market. The results show that GBST outperforms other candidate models, Cox, random survival forest and XGBoost, measured by the concordance index (C-index), Kolmogorov-Smirnov (KS) index, as well as by the area under the receiver operating characteristic curve (AUC) of each time period. Finally, some discussions and future work are presented in Section 4.

2. Gradient boosting survival tree

In this section, we introduce the method of *gradient boosting survival tree* (GBST) that extends the traditional survival analysis models with a gradient boosting tree algorithm.

2.1. Survival analysis

The survival analysis aims, in general, for quantifying the distribution of time until some random event happens. In the field of consumer finance, the event is that a borrower defaults. Let T denote the default time, i.e., the time period from the moment that an borrower obtains the money to the event that he defaults. Note that in general T is a random variable and we assume it follows a probability distribution $\mathbb{P}$. Then for any time t, the survival function $S(t)$ is defined as follows,

$$S(t) := \mathbb{P}(T > t),$$

which quantifies the probability that the borrower "survives" longer than time t. Comparing with the traditional credit scoring methods, the survival analysis is able to predict not only the probability whether a borrowers defaults, but also the probability of time when he defaults.

Throughout this paper, we consider a finite set of observation time, denoted by $\{\tau_j, j = 0, 1, 2, \ldots, J\}$ satisfying $0 = \tau_0 < \tau_1 < \ldots < \tau_J < \infty$. In the context of consumer finance, the observation time could be, e.g., the monthly payment due date. The discrete-time hazard function is defined as

$$h(\tau_j) := \mathbb{P}(\tau_{j-1} < T \leq \tau_j | T > \tau_{j-1}), \quad j = 1, 2, \ldots,$$

which quantifies the probability of the event that occurs within the time period $(\tau_{j-1}, \tau_j]$ for the first time. By definition, the survival function S can be represented by the hazard function h. Indeed, for each $j = 1, 2, \ldots$, we have

$$\begin{aligned} S(\tau_j) =& \mathbb{P}(T > \tau_j) = \mathbb{P}(T > \tau_j | T > \tau_{j-1}) \mathbb{P}(T > \tau_{j-1} | T > \tau_{j-2}) \cdots \mathbb{P}(T > \tau_1) \\ =& (1 - h(\tau_j))(1 - h(\tau_{j-1})) \cdots (1 - h(\tau_1)) = \prod_{l=1}^{j} (1 - h(\tau_l)). \end{aligned} \tag{1}$$

Finally, the probability of the event that occurs within the time period $(\tau_{j-1}, \tau_j]$ is given by

$$\mathbb{P}(\tau_{j-1} < T \leq \tau_j) = h(\tau_j) S(\tau_{j-1}) = h(\tau_j) \prod_{l=1}^{j-1} \left(1 - h(\tau_l)\right). \tag{2}$$

Suppose that $h(t) \in (0, 1)$ for each t and we consider a logarithmic transformation of h such as

$$f(t) := \log \frac{h(t)}{1 - h(t)}, \text{ satisfying} \tag{3}$$

$$h(t) = \frac{1}{1 + e^{-f(t)}} \text{ and } 1 - h(t) = \frac{1}{1 + e^{f(t)}}. \tag{4}$$

By defining the following two mappings:

$$y_j(t) := \begin{cases} -1, & \text{if } t > \tau_j, \\ 1, & \text{if } t \leq \tau_j, \end{cases}, \text{ and} \tag{5}$$

$$J(t) := \begin{cases} j, & \text{if } t \in (\tau_{j-1}, \tau_j], \\ J + 1, & \text{if } t > \tau_J. \end{cases}$$

Eq. (2) implies

$$\mathbb{P}(T=t)=\prod_{j=1}^{J(t)}\frac{1}{1+e^{-y_j(t)f(\tau_j)}}, \forall t \leq \tau_J.$$

On the other hand, if $t > \tau_J$, i.e., beyond the total observation period, its probability could be quantified as

$$\mathbb{P}(T=t>\tau_J)=\prod_{j=1}^{J}\frac{1}{1+e^{f(\tau_j)}}=\prod_{j=1}^{J}\frac{1}{1+e^{-y_j(t)f(\tau_j)}}.$$

Combining the above two cases, we have that for any observed default time $t \in [0,\infty)$, its probability is given by

$$\mathbb{P}(T=t)=\prod_{j=1}^{J(t)\wedge J}\frac{1}{1+e^{-y_j(t)f(\tau_j)}}, \tag{6}$$

where $J(t) \wedge J := \min(J(t), J)$.
Remark. In many real applications, the exact "death" time might not be observable. Instead, with a discrete series of observation (sampling) time, one can usually determine in which period $(\tau_{j-1}, \tau_j]$ one individual is "dead". Then the output of $y_j(t)$ and $J(t)$ are observable without knowing the exact "death" time.

2.2. Learning objective

Let $\boldsymbol{x} \in \mathbb{R}^n$ denote the feature of each individual. In the field of consumer finance, it quantifies the characteristics of each borrower, such as age, sex, education level, etc. The fundamental task of survival analysis is then to investigate the optimal survival function for each individual $S(t;\boldsymbol{x})$. As shown in Eq. (1), it is sufficient to finding out the best hazard function $h(t;\boldsymbol{x})$ and equivalently its logarithmic representation $f(t;\boldsymbol{x})$.

Now we state the major assumption of this paper: f is assumed to be approximated by $\hat{f} : [0,\infty) \times \mathbb{R}^n \to \mathbb{R}$ with a tree ensemble model such as

$$f(t;\boldsymbol{x}) \cong \hat{f}(t;\boldsymbol{x}) := \sum_{k=1}^{K} f_k(t;\boldsymbol{x}), f_k \in \mathcal{F}, \tag{7}$$

where for each k, f_k is the output of kth *survival tree*, and

$$\mathcal{F} := \{f(t;\boldsymbol{x}) = w(t, q(\boldsymbol{x})) | q : \mathbb{R}^n \to L, w : [0,\infty) \times L \to \mathbb{R}\}$$

is the space of survival trees with leaf nodes as functionals of time t. Here q represents the structure of each tree with L leaf nodes that maps a variable $\boldsymbol{x}$ to its corresponding leaf index. Each f_k represents an independent tree structure q_k and leaf weights functional $w^{(k)}$. The difference between a survival tree and a conventional regression tree is that the leaf nodes of a survival tree are functionals of time t, whereas the leaf nodes of a regression tree are simply real values.

Suppose we observe samples $\{\boldsymbol{x}_i, t_i\}, i = 1, 2, \ldots, N$, where t_i denotes the default time. Under the i.i.d. assumption, we can calculate the negative log-likelihood as in Eq. (6) as follows,

$$\begin{aligned}
\mathcal{L}(\phi) &= -\log\left(\prod_{i=1}^{N} \mathbb{P}(t_i; \boldsymbol{x}_i)\right) = -\log\left(\prod_{i=1}^{N} \prod_{j=1}^{J(t_i)\wedge J} \frac{1}{1+e^{-y_j(t_i)f(\tau_j;\boldsymbol{x}_i)}}\right) \\
&\cong -\log\left(\prod_{i=1}^{N} \prod_{j=1}^{J(t_i)\wedge J} \frac{1}{1+e^{-y_j(t_i)\hat{f}(\tau_j;\boldsymbol{x}_i)}}\right) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{J(t_i)\wedge J} \log\left(1+\exp\left\{-y_j(t_i)\hat{f}(\tau_j;\boldsymbol{x}_i)\right\}\right). \qquad (8)
\end{aligned}$$

For each $j = 1, 2, \ldots, J$, the set of samples that survive longer than τ_{j-1} (in other words, $J(t_i) \geq j$), N_j, is defined as follows:

$$N_j := \{i \in \{1, 2, \ldots, N\} | J(t_i) \geq j\}. \qquad (9)$$

Then it is easy to verify that Eq. (8) implies

$$\mathcal{L}(\phi) = \sum_{j=1}^{J} \sum_{i \in N_j} \log\left(1+\exp\left(-y_j(t_i)\hat{f}(\tau_j;\boldsymbol{x}_i)\right)\right).$$

Finally, in order to avoid over-fitting, one l^2 regularization term is added to penalize the complexity of the model:

$$\mathcal{L}(\phi) = \sum_{j=1}^{J} \sum_{i \in N_j} \log\left(1+\exp\left(-y_j(t_i)\hat{f}(\tau_j;\boldsymbol{x}_i)\right)\right) + \frac{\lambda}{2}\sum_{k} \|w^{(k)}\|^2, \qquad (10)$$

where $\lambda > 0$ is a hyperparameter controls how large the penalization is and $w^{(k)}$ is the leaf weight functionals for kth survival tree. It is remarkable that since we consider in this paper merely the discrete observation time $\{\tau_j\}$, the l^2-norm of a leaf weight functional w is calculated as

$$\|w\|^2 := \sum_{j=1}^{J} \sum_{l=1}^{L} w_l^2(\tau_j).$$

2.3. Gradient tree boosting

Inspired by the gradient tree boosting method (Friedman, 2001), the optimization problem stated in Eq. (10) is solved in an additive manner. More specifically, one starts with an initial *survival tree* f_0 and for $m = 1, 2, \ldots,$

$$f_m = \underset{f \in \mathcal{F}}{\operatorname{argmin}} \mathcal{L}^{(m)} := \sum_{j=1}^{J} \sum_{i \in N_j} \log \left(1 + \exp\left\{-y_j(t_i) \left(\hat{f}^{(m-1)}(\tau_j; \boldsymbol{x}_i) + f(\tau_j; \boldsymbol{x}_i)\right)\right\}\right) + \frac{\lambda}{2}\|w\|^2 \tag{11}$$

and $\hat{f}$ is updated by

$$\hat{f}^{(m)}(t; \boldsymbol{x}) = \hat{f}^{(m-1)}(t; \boldsymbol{x}) + f_m(t; \boldsymbol{x}).$$

Define the logarithmic loss function $\varrho : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as

$$\varrho(y, f) := \log(1 + \exp(-y \cdot f)).$$

Then $\mathcal{L}^{(m)}$ can be approximated by a Taylor expansion up to the second order:

$$\mathcal{L}^{(m)}(f) \cong \sum_{j=1}^{J} \sum_{i \in N_j} \left(\underbrace{\varrho\left(y_j(t_i), \hat{f}^{(m-1)}(\tau_j; \boldsymbol{x}_i)\right)}_{\text{independent of } f} + r_{i,j}^{(m-1)} f(\tau_j; \boldsymbol{x}_i) + \frac{1}{2}\sigma_{i,j}^{(m-1)} f^2(\tau_j; \boldsymbol{x}_i) \right) + \frac{\lambda}{2}\|w\|^2, \tag{12}$$

where the first and second order derivative $r_{i,j}^{(m-1)}$ and $\sigma_{i,j}^{(m-1)}$ are given by

$$\begin{aligned} r_{i,j}^{(m-1)} &:= \frac{\partial \varrho}{\partial f}\left(y_j(t_i), \hat{f}^{(m-1)}(\tau_j; \boldsymbol{x}_i)\right), \\ \sigma_{i,j}^{(m-1)} &:= \frac{\partial^2 \varrho}{\partial f^2}\left(y_j(t_i), \hat{f}^{(m-1)}(\tau_j; \boldsymbol{x}_i)\right). \end{aligned}$$

Removing the constant independent of f in Eq. (12), the objective becomes

$$\tilde{\mathcal{L}}^{(m)}(f) := \sum_{j=1}^{J} \sum_{i \in N_j} \left(r_{i,j}^{(m-1)} f(\tau_j; \boldsymbol{x}_i) + \frac{1}{2}\sigma_{i,j}^{(m-1)} f^2(\tau_j; \boldsymbol{x}_i) \right) + \frac{\lambda}{2}\|w\|^2, \tag{13}$$

Note that since $y_j(t_i)$ takes value either 1 or -1 (see Eq. (5)), for the logarithmic loss function ϱ, it is easy to verify that

$$\frac{\partial \varrho}{\partial f}(y, f) = \begin{cases} -(1 - h), & \text{if } y = 1 \\ h, & \text{if } y = -1 \end{cases}, \text{ and} \tag{14}$$

$$\frac{\partial^2 \varrho}{\partial f^2}(y, f) = h(1 - h), \text{ if } y = 1 \text{ or } -1, \tag{15}$$

where $h := \frac{1}{1+e^{-f}}$ can be interpreted as the hazard probability (cf. Eq. (4)).

For a survival tree with L leaf nodes, f can be represented as

$$f(\tau_j; \boldsymbol{x}_i) = \sum_{l=1}^{L} w_l(\tau_j)\mathbf{1}(i \in I_l)$$

where we define $I_l := \{i \in \{1, 2, \ldots, N\} | q(\boldsymbol{x}_i) = l\}$ and q represents the structure of the tree. Eq. (13) becomes

$$\tilde{\mathcal{L}}^{(m)}(w) := \sum_{j=1}^{J}\sum_{l=1}^{L}\left\{\sum_{i\in N_j\cap I_l}\left(r_{i,j}^{(m-1)}w_l(\tau_j) + \frac{1}{2}\sigma_{i,j}^{(m-1)}w_l^2(\tau_j)\right) + \frac{\lambda}{2}w_l^2(\tau_j)\right\}.$$

As shown in Eq. (15), $\sigma_{i,j}^{(m-1)} > 0, \forall i, j$, and $\tilde{\mathcal{L}}^{(m)}$ is therefore a strictly convex function with respect to w. Hence, given a fixed structure q, for each j and l, the optimal $w_l^{(m)}(\tau_j)$ is

$$w_l^{(m)}(\tau_j) = -\frac{\sum_{i\in N_j\cap I_l} r_{i,j}^{(m-1)}}{\sum_{i\in N_j\cap I_l}\sigma_{i,j}^{(m-1)} + \lambda}, \tag{16}$$

and the corresponding $\tilde{\mathcal{L}}^{(m)}$ is

$$\tilde{\mathcal{L}}^{(m)} = -\frac{1}{2}\sum_{j,l}\frac{\left(\sum_{i\in N_j\cap I_l} r_{i,j}^{(m-1)}\right)^2}{\sum_{i\in N_j\cap I_l}\sigma_{i,j}^{(m-1)} + \lambda}, \tag{17}$$

where the explicit form of $r_{i,j}^{(m-1)}$ and $\sigma_{i,j}^{(m-1)}$ is

$$\begin{aligned} r_{i,j}^{(m-1)} &= \begin{cases} -(1 - h_{i,j}^{(m-1)}), & \text{if } y_j(t_i) = 1 \\ h_{i,j}^{(m-1)}, & \text{if } y_j(t_i) = -1 \end{cases} \\ \sigma_{i,j}^{(m-1)} &= (1 - h_{i,j}^{(m-1)})h_{i,j}^{(m-1)}, \end{aligned}$$

with

$$h_{i,j}^{(m-1)} := \frac{1}{1 + \exp(-\hat{f}^{(m-1)}(\tau_j; \boldsymbol{x}_i))}$$

being the hazard probability estimated for $\boldsymbol{x}_i$ at time τ_j up to $(m-1)$th iteration.

Eq. (17) measures how well a survival tree q fits the observed data. Based on it, we develop a greedy algorithm for constructing the optimal survival tree, since usually it is impossible to explore all possible tree structures. We start with a single leaf node and iteratively split it into two sets. Suppose that one set I is split into two separate sets I_L and I_R. Then the corresponding loss reduction is given by

$$\tilde{\mathcal{L}}_{split} = \frac{1}{2}\sum_j\left[\frac{\left(\sum_{i\in N_j\cap I_L} r_{i,j}^{(m-1)}\right)^2}{\sum_{i\in N_j\cap I_L}\sigma_{i,j}^{(m-1)} + \lambda} + \frac{\left(\sum_{i\in N_j\cap I_R} r_{i,j}^{(m-1)}\right)^2}{\sum_{i\in N_j\cap I_R}\sigma_{i,j}^{(m-1)} + \lambda} - \frac{\left(\sum_{i\in N_j\cap I} r_{i,j}^{(m-1)}\right)^2}{\sum_{i\in N_j\cap I}\sigma_{i,j}^{(m-1)} + \lambda}\right].$$

The pseudo code of the proposed greedy split algorithm is shown in Algorithm 1.

Algorithm 1 Split algorithm for GBST.

```
Input: $I$: Set of individuals in current node; $\{\boldsymbol{x}_i \in \mathbb{R}^n\}$: the features of individuals;
    $N_j, j = 1, 2, \ldots, J$ defined in Eq. (9)
for $k = 1$ to $K$ do
    for $j = 1$ to $J$ do
        $W_j \leftarrow \sum_{i \in N_j \cap I} r_{i,j}$, $V_j \leftarrow \sum_{i \in N_j \cap I} \sigma_{i,j}$
    end for
    for $s$ in sorted($I$, by $\boldsymbol{x}_{sk}$) do
        for $j = 1$ to $J$ do
            if $s \in N_j$ then
                $W_{jL} \leftarrow W_{jL} + r_{s,j}$, $V_{jL} \leftarrow V_{jL} + \sigma_{s,j}$
                $W_{jR} = W_j - W_{jL}$, $V_{jR} = V_j - V_{jL}$
            end if
        end for
        $score \leftarrow \max(score, \sum_j [\frac{G_{jL}^2}{V_{jL}+\lambda} + \frac{G_{jR}^2}{V_{jR}+\lambda} - \frac{G_j^2}{V_j+\lambda}])$
    end for
end for
Output: Split with max score, splitting schema
```

To initialize the tree, one has to give an initial estimation of the hazard probability $h_{i,j}^{(0)}$. Without any prior knowledge, we assume each individual follows the same hazard distribution over time with the Kaplan-Meier estimator (Kaplan and Meier, 1958). More specifically, let d_j be the number of default individuals at $(\tau_{j-1}, \tau_j]$ and n_j be the number of individuals whose survival time is larger than τ_{j-1}, i.e., formally,

$$d_j := \sum_{i=1}^{N} \mathbf{1}(J(t_i) = j), \quad n_j := \sum_{i=1}^{N} \mathbf{1}(J(t_i) \geq j),$$

where t_i is the default time of ith individual. Then, the hazard probability by the Kaplan-Meier estimator is calculated as

$$\hat{h}_j = \frac{d_j}{n_j}, \tag{18}$$

which is used as our initial estimation $h_{i,j}^{(0)} = \hat{h}_j, \forall i, j$.

Finally, we summarize the algorithm for GBST in Algorithm 2.

3. Experiments

In this section, we conduct empirical experiments by applying the gradient boosting survival tree (GBST) method to the data collected in Chinese market of consumer finance and present the results of the experiments.

Algorithm 2 Algorithm for GBST

```
Input: $\boldsymbol{x} \in \mathbb{R}^n$: the features of individuals; $y_j(t_i), j = 1, ..., J; i = 1, ..., N$: the default labels of individuals at observed times.
Calculate the set $N_j$ for each $j$ defined in Eq. (9).
Initialize $h_{i,j}^{(0)} = \hat{h}_j, \forall i, j$ as in Eq. (18).
for $m = 1$ to $M$ do
    Construct a tree with the split algorithm (see Algorithm 1)
    In each terminal node $l$, calculate $w_l^{(m)}$ as in Eq. (16) for each $l$
    Calculate $f_m(\tau_j; \boldsymbol{x}_i) \leftarrow \sum_l w_l(\tau_j) \mathbf{1}(i \in I_l)$
    Update $\hat{f}^{(m)}(\tau_j; \boldsymbol{x}_i) \leftarrow \hat{f}^{(m-1)}(\tau_j; \boldsymbol{x}_i) + f_m(\tau_j; \boldsymbol{x}_i)$
end for
Calculate the hazard function $h(\tau_j; \boldsymbol{x}_i) \leftarrow \frac{1}{1+\exp\left\{-\hat{f}^{(M)}(\tau_j; \boldsymbol{x}_i)\right\}}$
Output: The Survival function $S(\tau_j; \boldsymbol{x}_i) \leftarrow \prod_{l=1}^{j-1} (1 - h(\tau_l; \boldsymbol{x}_i))$
```

3.1. Dataset

We use the data provided by 360 Finance Inc., who provides credit products mainly in form of installment loans. The loans are expected to be repaid monthly over a period of 3, 6 or 12 months. In order to make a full observation of repayment behavior, we only consider the accounts who borrowed money in January 2018 and March 2018[2] and the loans were scheduled to be repaid in 12-month installments. Among these accounts, 200,000 loans borrowed in January are randomly selected as the training set, and 120,000 in March are sampled as the testing set (see Table 1). In total, 320,000 samples are drawn. An account is defined as a *default* account if on any scheduled repayment due date the borrower is once overdue for at least 10 days, no matter whether he pays off the installment later or not. Hence, the discretized observation time (see Section 2.1) applied in survival analysis is the monthly repayment due date with extra 10 days. It is remarkable that once an account defaults in some repayment month, it remains in the default state in the rest months, regardless of later repayments. Moreover, any early repayment accounts will never be removed during the observation period, which means if someone early repays the loan at some time, he will be regarded as "repaying on time" in the rest time.

dataset	**observation period**	**sample size**
training set	January 2018 to January 2019	200,000
testing set	March 2018 to March 2019	120,000

Table 1.: Summary of datasets.

Therefore, the *default rate* in certain repayment month is the number of accounts who is overdue for at least 10 days in this month or was so in previous months over the total number of accounts, i.e.,

$$\text{default rate}(t) = \frac{\#\text{default accounts up to month } t}{\#\text{total accounts}}. \tag{19}$$

---

[2] The data from February 2018 are excluded because it covers the Chinese New Year Festival which usually has a strong festival effect and results in significant shifts on data distribution.

Note that the default rate defined here is accumulative over time, in accordance with default probability used in survival analysis. Hence, the curve of default rate grows with time, as depicted in Figure 1. In order to comply with the data protection policy of 360 Finance Inc., the exact values of default rates are not plotted. Instead, we plot the values of default rates compared to a baseline, which is the default rate of the sample accounts from the training dataset cummulated to 12th month. It is also remarkable that our definition of default is more strict than the conventional definition and, therefore, the default rate in our definition is higher as well. We also see in Figure 1 that the default rate of the testing data in March 2018 grows faster than that of the training data in January 2018, though two curves do not diverge significantly.

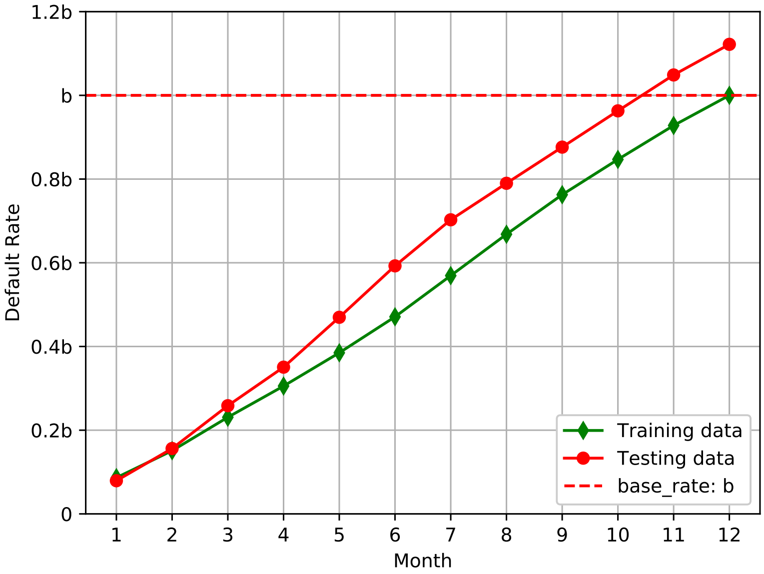


Figure 1.: (Cumulative) default rate (see Eq. (19)) for both training and testing data, compared to the baseline default rate b, which is the default rate of the sample accounts from the training dataset cummulated to 12th month. For instance, $0.2b$ means that the default rate is 20% of the default rate b.

3.2. Preprocessing of data and feature selection

For each account, over 400 raw variables are collected at application, including

- personal information provided by the borrower, e.g., sex, age, education level;
- credit report retrieved from People's Bank of China (PBC), e.g., income score, credit score;
- device information collected by the 360 Finance App which is authorized by the borrower, e.g., location of application, device brand, memory size of the mobile phone;
- credit report from third-party credit rate agencies, e.g., number of loans applied in other platforms, travel intensity;
- social network information.

We first exclude the variables with a missing rate higher than 80%. The categorical variables, e.g. sex and education level, are represented by numerical values with one-hot encoding. For each variable, the missing values are filled up with a default value and an additional binary variable is generated to indicate whether the value is missing. After preprocessing, ca. 1000 features are generated. In order to reduce the computational complexity and to speed up the optimization procedure, we then apply XGBoost (XGB, Chen and Guestrin, 2016) to do feature selection. More specifically, we train an XGB model with the training dataset to solve a classification problem, where the samples are labeled as "bad" if the account defaults within the total observation period (12 months), and are labeled as "good" otherwise. The hyperparameters of XGB model, such as max tree depth, are chosen based on a 5-fold cross validation. We finally select the 50 features with the highest importance (for details see Section 10.13, Friedman et al., 2001). Table 2 shows some of the most important features and their data sources.

source	feature
PBC report	income score
	credit score
	overdue information of credit cards
personal information	age
	sex
	education level
device information	location
third-party rate agency	no. of loans in other lending platforms
	travel intensity
other information	whether possessing a car
	application channel

Table 2.: List of important features, which are selected according to the feature importance generated by an XGBoost model trained with the training dataset.

3.3. Convergence

At each iteration of the GBST algorithm (see Algorithm 2), a survival tree is constructed by minimizing the loss function stated in Eq. (10) on the training dataset. This turns out to be very time-costly when the dataset is large. Hence, we apply a sampling approach as proposed in stochastic gradient boosting (Friedman, 2002) to reduce the computational cost. Instead of using the entire training dataset, we randomly sample a subset of the training dataset and apply them to train the model, with certain sampling rate α. Note that the loss defined in Eq. (10) measures how well the proposed GBST model fits the observed samples. To test the convergence behavior of the GBST algorithm with random sampling, we run the algorithm 1000 times independently. In this experiment, we use the sampling rate $\alpha = 20\%$, and the regularization parameter $\lambda = 0.001$, and the max tree depth is set to 6. The latter two hyperparameters are chosen empirically based on the hyperparameters selected for the XGB model that is applied in feature selection (see the last subsection 3.2).

Figure 2 plots the distribution of loss values over iterations of these 1000 independent runs. For each iteration, we calculate the minimal, maximal, 25% quantile, median and 75% quantile of the 1000 independent loss values and plot in Figure 2 by a whisker plot. Figure 2 shows that the loss values decrease very quickly over iterations and converges uniformly to a certain level near 0, as the iteration grows. Hence, the GBST algorithm with random sampling shows a promising property of convergence. It shows also that 30 iterations are sufficient to obtain the optimal solution. Hence, by training, we stop at the 30th iteration.

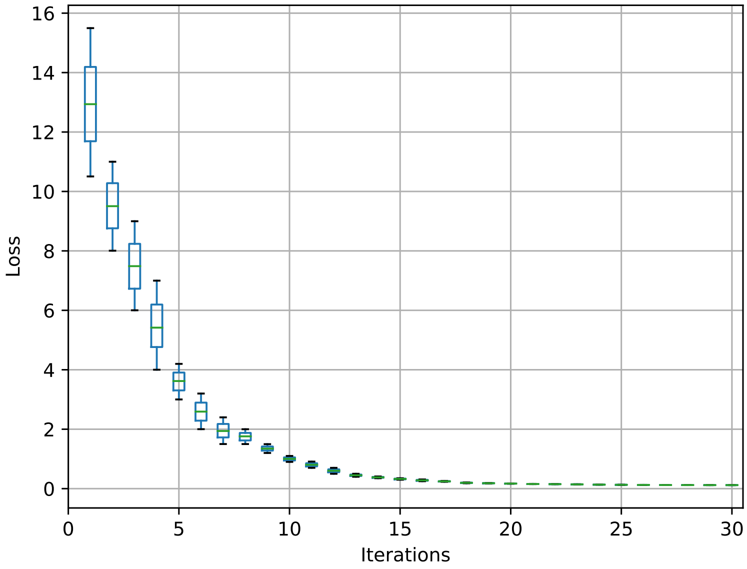


Figure 2.: Distribution of loss values defined in Eq. (10) over iterations with 1000 independent runs.

3.4. Performance

After training with the features $\boldsymbol{x}$ generated in the dataset of January 2018, we obtain the optimal GBST model and apply it to calculate the predicted survival probability of each account in the testing dataset of March 2018 for every repayment month. For every repayment month, the accounts are sorted by the probability in ascending order, and evenly divided into 20 bins. Therefore, the first group contains the accounts with the lowest predicted survival probability, i.e., the highest predicted default rate, and vice versa. Figure 3 exhibits the plots of the real default rate of each group across the whole period of loan. The primary findings are as follows:

- The predicted probability of survival is negatively related to the real default rate in all the repayment months: groups with higher (lower) predicted survival probability generally have lower (higher) real default rates.
- The slope of the real default rate line becomes much steeper in the interval between the 7th group and the 9th group, which implies that the survival probability in this interval has stronger discriminating power of default.

To examine the model performance more rigorously, we adapt the concordance index (C-Index) proposed by Harrell et al. (1982) to measure the goodness of fit for the survival model, as the statistics provides a global assessment of the model for the continuous event. Notably, our dataset also guarantees the C-Index not to be doomed by the censoring. As shown in Table 3, computed in every repayment month, the values of C-index stay around 0.8, while slightly moving down with time, representing that

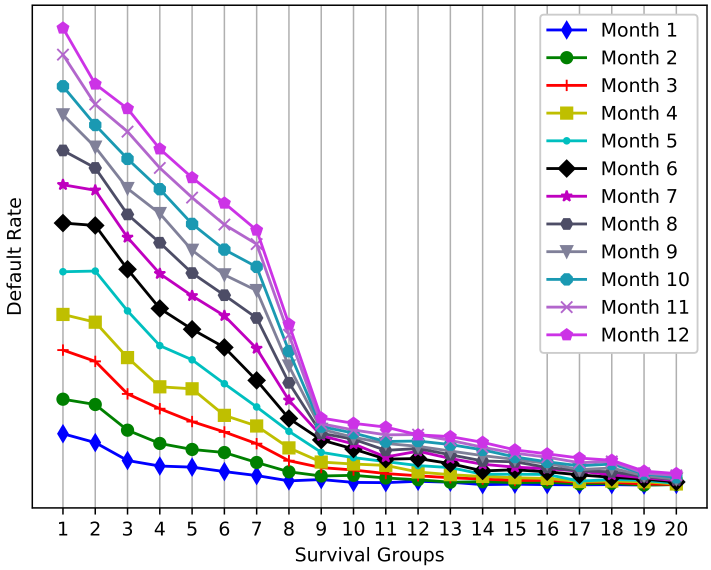


Figure 3.: Default rate curves of different survival groups in the testing dataset over repayment months. Accounts in the testing dataset (March 2018) are ascendingly sorted by the survival probability predicted by all candidate models, GBST, COX, RSF and XGB.

the survival model has strong discrimination[3]. The downward trend of the C-Index over months reflects the fact that the prediction of survival probability becomes less accurate over time, since features $\boldsymbol{x}$ used in the prediction are collected at application and will not be renewed during months, whereas $\boldsymbol{x}$ might shift in value over time, due to possible macroeconomic or personal economic changes. For instance, a borrower might become unemployed after the application of installment loans and, therefore, fails to repay the installment at later time. However, this information cannot be captured and be foreseen at application.

Month	1	2	3	4	5	6
C-Index	0.8121	0.8064	0.8027	0.7992	0.7979	0.7951
Month	7	8	9	10	11	12
C-Index	0.7935	0.7928	0.7923	0.7908	0.7899	0.7889

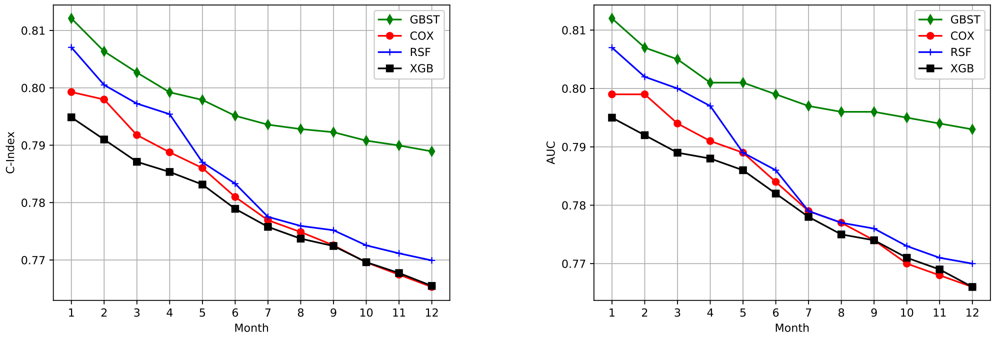
Table 3.: C-Indices of repayment months calculated for accounts in the testing dataset.

3.5. Comparison with existing models

In this section, we compare the performance of the proposed gradient boosting survival tree (GBST) model with the performance of alternative models, including Cox proportional hazards (COX) model, random survival forest (RSF) and XGBoost (XGB). We apply the C-index introduced above as well as the other two statistics,

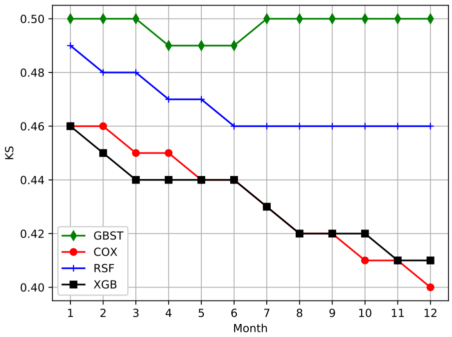
---

[3] The C-Index takes values ranging from 0.5 (no discrimination) to 1.0 (perfect discrimination).



(a) C-Index

(b) AUC



(c) KS-Index

Figure 4.: C-Index, AUC and KS-Index of all repayment months for all candidate models, GBST, proportional hazards (COX), random survival forest (RSF) and XGBoost (XGB), calculated in the testing dataset.

Kolmogorov-Smirnov (KS) index and the area under the receiver operating characteristics curve (AUC) to measure the goodness of model fit. The three panels in Figure 4 illustrate the curve of statistics in every repayment month respectively. All the results are obtained from the test set, based on the model trained in the training set.

All of the models we tested have shown the discriminating power to some extent, since the values of AUC and C-Index are all above 0.7, and the KS indices are generally higher than 0.4. However, the curve of GBST consistently lies above the other three curves of alternative models regardless of the statistics, which indicates that GBST outperforms the other models in terms of higher values of AUC , KS and C-Index. Most importantly, the discriminating power of the GBST model decays much more slowly with time than the counterparts, indicating that the GBST model tends to be more time-consistent.

To better understand the performance variation in the models of interest, as the plotting of Figure 3, we plot the real default rate curves over 20 survival groups for the four models, and we generate the groups of each model by sorting the model predicted survival probability and following the same recipe introduced in the last subsection 3.4. Without loss of generality, Figure 5 depicts the default rate curves of four models in the 12th repayment month alone, compared to the baseline b that is the default rate of the sample accounts from the training dataset cummulated to 12th month. The baseline b can also be approximately viewed as the average default rate of all 20

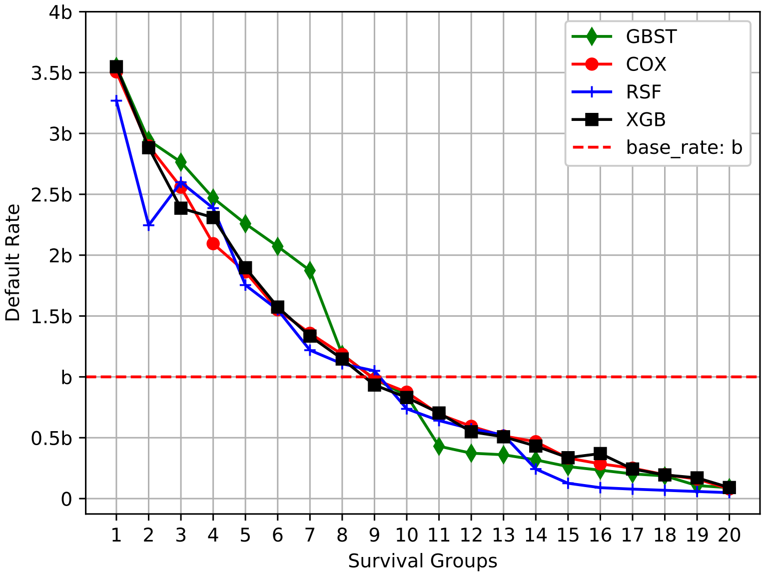


Figure 5.: Default rate curves of different survival groups in the testing dataset in 12th repayment month. b is the same baseline as in Figure 1, i.e., the default rate of the sample accounts from the training dataset cummulated to 12th month. Accounts in the testing dataset (March 2018) are ascendingly sorted by the survival probability predicted by all candidate models, GBST, COX, RSF and XGB.

groups. It is interesting to find that when the survival groups are controlled, the GBST model has higher real default rates in the lower-ranked groups (group $1-8$, which are higher than the baseline) compared to the other models, while it has relatively lower real default rates in the higher-ranked groups, despite several exceptions in the RSF model. This finding proves that the GBST model is more capable of distinguishing the default accounts from the others, which is in accordance with the previous results observed in the goodness-of-fit measures.

4. Discussion and future work

Like traditional scoring methods, the proposed gradient boosting survival tree (GBST) model is based on the attributes of the accounts (called *features* in this paper) measured at the time of application. Yet, many of them change over time, especially when the modeling period is not short. To address the problem of changing attributes, Djeundje and Crook (2019) propose a dynamic parametric survival model with B-splines. In the future, we plan to incorporate time-varying attributes into the GBST model.

In the usual setting of survival analysis, as well as in the setting of GBST, one considers a binary state space, i.e., either the event of interest happens (State = 1) or not (State = -1). However, in many applications (see e.g., Lee et al., 2018), as in the problem of credit scoring, additional events may be of importance as well. For instance, an account with early repayment is viewed as a non-default account. However, with respect to profits, an early-repayment account tends to differ from an account with on-time repayments. This problem can be dealt with by extending the binary state space to the multi-state space, which is left in our future work.

References

Banasik, J., Crook, J. N., and Thomas, L. C. (1999). Not if but when will borrowers default. *Journal of the Operational Research Society*, 50(12):1185–1190.

Blakely, R. J. and Cox, A. (1972). Evidence for short geomagnetic polarity intervals in the early cenozoic. *Journal of Geophysical Research*, 77(35):7065–7072.

Cao, R., Vilar, J. M., and Devia, A. (2009). Modelling consumer credit risk via survival analysis. *Statistics and Operations Research Transactions*, 33(1):0003–30.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

Dirick, L., Claeskens, G., and Baesens, B. (2017). Time to default in credit scoring using survival analysis: a benchmark study. *Journal of the Operational Research Society*, 68(6):652–665.

Djeundje, V. B. and Crook, J. (2019). Dynamic survival models with varying coefficients for credit risks. *European Journal of Operational Research*, 275(1):319–333.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The Elements of Statistical Learning*, volume 1. Springer.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378.

Gordon, L. and Olshen, R. (1985). Tree-structured survival analysis. *Cancer Treatment Reports*, 69(10):1065–1069.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L., and Rosati, R. A. (1982). Evaluating the yield of medical tests. *Jouranl of American Medical Association*, 247(18):2543–2546.

Ishwaran, H., Kogalur, U., Blackstone, E., and Lauer, M. (2008). Random survival forests. *The Annals of Applied Statistics*, 2(3):841–860.

Kaplan, E. L. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282):457–481.

Lee, C., Zame, W. R., Yoon, J., and van der Schaar, M. (2018). Deephit: A deep learning approach to survival analysis with competing risks. In *32 AAAI Conference on Artificial Intelligence.*

Li, W., Ding, S., Wang, H., Chen, Y., and Yang, S. (2019). Heterogeneous ensemble learning with feature engineering for default prediction in peer-to-peer lending in china. *World Wide Web*, pages 1–23.

Narain, B. (1992). Survival analysis and the credit granting decision. *Credit Scoring and Credit Control,* Editors: Thomas L.C., Crook J.N., Edelman D.B., Oxford University Press.

Thomas, L., Edelman, D., and Crook, J. (2002). *Credit Scoring and Its Applications.* SIAM.