Regularized Parametric Regression for High-dimensional Survival Analysis

Yan Li* Kevin S. Xu† Chandan K. Reddy‡

Abstract

Survival analysis aims to predict the occurrence of specific events of interest at future time points. The presence of incomplete observations due to *censoring* brings unique challenges in this domain and differentiates survival analysis techniques from other standard regression methods. In many applications where the distribution of the survival times can be explicitly modeled, parametric survival regression is a better alternative to the commonly used Cox proportional hazards model for this problem of censored regression. However, parametric survival regression suffers from model overfitting in high-dimensional scenarios. In this paper, we propose a *unified* model for regularized parametric survival regression for an arbitrary survival distribution. We employ a generalized linear model to approximate the negative log-likelihood and use the elastic net as a sparsity-inducing penalty to effectively deal with high-dimensional data. The proposed model is then formulated as a penalized iteratively reweighted least squares and solved using a cyclical coordinate descent-based method. We demonstrate the performance of our proposed model on various high-dimensional real-world microarray gene expression benchmark datasets. Our experimental results indicate that the proposed model produces more accurate estimates compared to the other competing state-of-the-art methods.

Keywords: survival analysis; censored data; parametric regression; sparse methods; high-dimensional data.

1 Introduction

Survival analysis aims at modeling time-to-event data, which is typically collected in longitudinal studies that start from a particular time and last until a certain *event of interest* has occurred [12]. However, the event of interest may not always be observed during the study period due to time limitations or losing data traces. This phenomenon is known as *censoring* and makes survival analysis different from and more challenging than standard regression. For the data instances where the event of interest has been observed, the time to the event of interest is known as the *survival time*, while for the other instances (censored instances), the last observed time is known as the *censored time*. The most common form of censoring that occurs in real-world scenarios is *right censoring*, where the survival time of a censored instance is longer than or equal to the censored time, but its precise value is unknown. In the rest of this paper, for the sake of simplicity, we refer to right censored data as censored data, unless otherwise specified.

Parametric regression is one of the fundamental tools in statistics and data analysis. In survival analysis, both the Cox proportional hazards model and parametric censored regression models are important foundational techniques for survival time prediction. Although not as widely studied as the Cox model, parametric censored regression has several advantages compared with the Cox model.

First, parametric censored regression models are more easily interpreted than the Cox model. In parametric censored regression, the probability of occurrence of an event of interest at a certain time is directly described by the density function of the selected distribution, and the probability of non-occurrence of the event of interest until a certain time is represented straightforwardly by a survival function (one minus the distribution function). On the other hand, the Cox model does not model the probability of occurrence directly but learns it by maximizing the hazard ratio between the censored instances and their corresponding risk set.

Second, parametric censored regression is more efficient than the Cox model when tied observations (when survival times of multiple instances are exactly the same) occur during the study. Parametric censored regression can be directly used without any modification, while the Cox model has to use some approximation methods that suffer from either inducing bias (Breslow's approximation and Efron's approximation [7]) or bad scalability (Discrete method [19]).

To enable regularized parametric censored regression to handle high-dimensional censored datasets, in this paper, we propose the "**URPCR**" model, which stands for "**U**nified model for **R**egularized **P**arametric

*Department of Computer Science, Wayne State University, Detroit, MI. E-mail: rock_liyan@wayne.edu

†Electrical Engineering and Computer Science Department, University of Toledo, Toledo, OH. E-mail: kevin.xu@utoledo.edu

‡Department of Computer Science, Wayne State University, Detroit, MI. E-mail: reddy@cs.wayne.edu

Censored **R**egression". Our proposed model unifies the learning process of regularized parametric censored regression with different probability distributions; thus it improves the efficiency of model learning on an arbitrary probability distribution. This efficiency is important because the performance of parametric censored regression is highly dependent on the choice of distribution.

In our proposed URPCR model, the elastic net is employed as the regularization term because it can both induce a sparse coefficient vector and handle correlated features. To unify the learning process of the proposed model with different distributions, we use a second-order Taylor expansion to approximate the log-likelihood; in this way, the URPCR model can be solved as a penalized iteratively reweighted least squares (IRLS). However, different from the standard linear model, a bias scale parameter has to be learned in addition to the coefficient vector in our proposed generalized linear model. Motivated by coordinate descent, in our learning scheme, this scale parameter is viewed as one coordinate and is iteratively updated based on Newton's method. Finally, the model is learned via a cyclical coordinate descent scheme.

In our empirical evaluation using several real-world high-dimensional cancer gene expression survival benchmark datasets, our model attains very competitive C-index values and outperforms most of the competing methods available in the literature of survival analysis. Additionally, we also demonstrate that our model outperforms most of the competing methods for the task of classifying whether or not a subject is alive at various time points in the observed study period. This is accomplished by our URPCR model without the need to re-train a new classifier for each time point, which is one of the main advantages of this work.

The rest of the paper is organized as follows. In Section 2, related data mining approaches for survival analysis are discussed. In Section 3, the basic concepts of survival regression are introduced. Our proposed approach is explained in detail in Section 4. Section 5 demonstrates our experimental results on several real-world datasets while Section 6 concludes our work.

2 Related Work

In this section, we present some related works in survival analysis and highlight the differences and relationships between our proposed model and existing literature.

The Cox proportional hazards model [5] is one of the earliest and most widely used survival analysis methods. It has obtained significant interest from researchers in both the statistics and data mining communities. To deal with high-dimensional data, some regularization methods have been integrated with it. These methods include LASSO-COX [20], which introduces the L_1 norm penalty in the Cox log-likelihood function, Elastic-Net Cox (EN-COX) [26], which uses the elastic net penalty term, and kernel elastic net penalized Cox regression [22].

Parametric censored regression is another important branch of survival analysis. Parametric censored regression methods assume that the survival times of all instances in a dataset follow a particular distribution, and that there exists a linear relationship between either the survival time or the logarithm of the survival time and the features [6, 12]. Thus, these regression models can be viewed as generalized linear models. The Tobit model [21] is the earliest attempt of extending linear regression with the Gaussian distribution for data analysis with censored observations. Then, several other distributions such as Weibull, extreme value distributions [1], and log-logistic distribution [2, 13] have been successfully implemented for parametric censored regression in the early 1980s. Buckley and James [3] have proposed an algorithm, known as BJ regression, which incorporates the Kaplan-Meier (K-M) estimator [10] as the baseline distribution. To handle high-dimensional survival analysis, Wang et al. applied the elastic net penalty to the BJ regression (EN-BJ) [23], a weighted linear regression is proposed in [14], and a L_1 norm regularized accelerated failure time (AFT) model is sloved via bootstrap approach in [9].

In this paper, we propose the URPCR model to handle survival prediction with censored instances in high-dimensional data. We develop a unified learning scheme for learning a regularized parametric censored regression of an arbitrary survival distribution. In addition, the objective function is regularized using the elastic net penalty, which can induce the required sparsity and efficiently handle the challenge of high dimensionality. Compared to the BJ and EN-BJ estimators, our model does not need to use the Kaplan-Meier estimator to approximate the survival time of censored instances during the training process.

3 Preliminaries

In this section, we first introduce some basic notations and concepts of a survival regression model, and then we briefly review the basic formulation of parametric regression for survival analysis.

3.1 Terminologies and Notations In survival analysis, for each data instance, we observe either a survival time (O_i) or a censored time (C_i), but not both. The dataset is said to be right-censored if and only if $y_i = \min(O_i, C_i)$ can be observed during the study [17]. An instance in the survival data is usually represented

by a triplet (X_i, T_i, δ_i), where X_i is a $1 \times p$ feature vector; δ_i is the censoring indicator, i.e. $\delta_i = 1$ for an uncensored instance, and $\delta_i = 0$ for a censored instance; and T_i denotes the *observed time* and is equal to the survival time O_i for uncensored instances and C_i otherwise, i.e.

$$T_i = \begin{cases} O_i & \text{if} \quad \delta_i = 1 \\ C_i & \text{if} \quad \delta_i = 0 \end{cases} \tag{3.1}$$

For censored instances, O_i is a latent value, and the goal of survival analysis is to model the relationship between X_i and O_i by using the triplets (X_i, T_i, δ_i) for censored and uncensored instances.

One of the most important concepts in modeling such censored data is the *survival function* $S(t) = Pr(O \geq t)$, which is the probability that the time to the event of interest is no earlier than time t [11]. In contrast, the *cumulative death distribution function* $F(t)$ is defined as $F(t) = 1 - S(t)$ and represents the probability that the time to the event of interest is less than t. The *death density function* $f(t)$ is defined as $f(t) = \frac{F(t+\Delta t) - F(t)}{\Delta t}$, where Δt is a short time interval.

3.2 Parametric Regression for Survival Analysis Parametric methods for estimating the survival probability are efficient and accurate when survival times follow a particular distribution. Unlike the Cox proportional hazard model, in parametric methods, a complete likelihood function can be solved directly, and the parameters can be estimated using maximum-likelihood estimation (MLE) [12]. We now discuss the generic MLE procedure [6] used for survival data with censored observations.

Consider a set of N instances out of which there are c censored observations and $(N - c)$ uncensored observations. For convenience, we use the general notation $\mathbf{b} = (b_1, b_2, \cdots, b_p)$ to represent a set of parameters and assume that the survival times follow a probability distribution with survival function $S(t, \mathbf{b})$ and death density function $f(t, \mathbf{b})$. If the i^{th} instance is a censored observation, then it is not possible to obtain the actual survival time; however, it can be concluded that the event of interest did not happen until the censored time C_i, so $S(C_i, \mathbf{b})$ should be a probability value that is close to 1. On the contrary, if the i^{th} instance is an uncensored observation with survival time O_i, then $f(O_i, \mathbf{b})$ should be a high probability value. Thus, we can use $\prod_{\delta_j=1} f(T_i, \mathbf{b})$ to represent the joint probability of the $(N - c)$ uncensored observations and $\prod_{\delta_j=0} S(T_i, \mathbf{b})$ to represent the joint probability of the c right-censored observations. Therefore, the likelihood function of all N instances is given by

$$L(\mathbf{b}) = \prod_{\delta_i=1} f(T_i, \mathbf{b}) \prod_{\delta_i=0} S(T_i, \mathbf{b}) \tag{3.2}$$

Note that $\mathbf{b}$ is not the feature coefficient vector but the parameters of the assumed distribution.

4 Proposed Model

In this section, we will discuss the proposed URPCR model in detail, along with an efficient optimization approach to learn the model. The URPCR employs the basic notion of generalized linear models (GLMs) and the framework of cyclical coordinate descent to solve the elastic net penalized parametric censored regression. Thus, the URPCR enables the parametric censored regressions to perform feature selection and handle high-dimensional data sets in survival analysis.

4.1 Objective Function The URPCR aims at learning the relationship between the feature vectors and the target value in the same manner as done in the generalized linear model, which can be formulated as follows:

$$v = X\beta + \sigma\varepsilon, \quad \varepsilon \sim f \tag{4.3}$$

for some distribution f, where β is the coefficient vector, ε is the error term, and $\sigma > 0$ is an unknown bias scalar. For the i^{th} instance, v_i can either be the survival time ($v_i = O_i$) or the logarithm of the survival time ($v_i = \log(O_i)$). When $v_i = O_i$, Eq.(4.3) becomes an extended linear regression with a self-selected bias distribution; when $v_i = \log(O_i)$, then Eq.(4.3) represents an AFT model [25], which is a commonly used prediction method in survival analysis. Thus, the URPCR encompasses these two models within a unified framework, which can be solved with exactly the same learning process. This is the primary novel aspect of the proposed work.

Under this linear hypothesis, the likelihood function of all N instances can be represented as

$$L = \prod_{\delta_i=1} f(\varepsilon_i/\sigma) \prod_{\delta_i=0} (1 - F(\varepsilon_i)) \tag{4.4}$$

where $\varepsilon_i = \frac{y_i - X_i\beta}{\sigma}$, $y_i = T_i$ for extended linear regression, and $y_i = \log(T_i)$ for AFT model. The log-likelihood can be written in the form

$$ll = \sum_{\delta_i=1} g_1(\varepsilon_i) - \log(\sigma) + \sum_{\delta_i=0} g_2(\varepsilon_i) \tag{4.5}$$

where $g_1 = \log(f(\cdot))$ and $g_2 = \log(1 - F(\cdot))$.

To avoid overfitting the model, it may not be appropriate to build a prediction model that includes all of the features. This becomes even more important when the feature dimension (p) is either close to or larger than the sample size (N). Sparsity-inducing penalization is an effective method which can perform

model estimation and feature selection simultaneously. The elastic net [27] is one of the most commonly used penalty terms in the data mining and machine learning communities and consists of a mixture of the L_1 (lasso) and L_2 (ridge regression) penalties. Therefore, it can obtain both sparsity in the coefficients and handle correlated feature spaces simultaneously. Mathematically, it is defined as follows:

$$P(\beta) = \alpha\|\beta\|_1 + \frac{1-\alpha}{2}\|\beta\|_2^2 \tag{4.6}$$

where $0 \leq \alpha \leq 1$ is used to adjust the weights of the L_1 and L_2 norm penalties. Hence, the Lagrangian of the penalized negative log-likelihood becomes

$$\min_{\beta,\sigma}\left[-\frac{2}{N}\left(\sum_{\delta_i=1} g_1(\varepsilon_i) - \log(\varepsilon_i) + \sum_{\delta_i=0} g_2(\varepsilon_i)\right) + \lambda P(\beta)\right] \tag{4.7}$$

where $\lambda \geq 0$ is the Lagrange multiplier.

4.2 Optimization To minimize the objective function proposed in Eq.(4.7), we use a second-order Taylor expansion to approximate the log-likelihood and a cyclical coordinate descent-based method, which solves a penalized iteratively reweighted least squares (IRLS) problem in each iteration [18], to solve the generalized linear model.

If we treat σ as fixed and let $\eta = X\beta$, a two-term Taylor series expansion of the log-likelihood centered at $\tilde{\beta}$ has the following form.

$$\begin{aligned} ll(\beta) &\approx ll(\tilde{\beta}) + (\beta-\tilde{\beta})^T ll^{'}(\tilde{\beta}) + \frac{(\beta-\tilde{\beta})^T ll^{''}(\tilde{\beta})(\beta-\tilde{\beta})}{2} \\ &= ll(\tilde{\beta}) + (X\beta-\tilde{\eta})^T ll^{'}(\tilde{\eta}) + \frac{(X\beta-\tilde{\eta})^T ll^{''}(\tilde{\eta})(X\beta-\tilde{\eta})}{2} \end{aligned} \tag{4.8}$$

where $\tilde{\eta} = X\tilde{\beta}$; $ll^{'}(\tilde{\beta})$, $ll^{''}(\tilde{\beta})$, $ll^{'}(\tilde{\eta})$, and $ll^{''}(\tilde{\eta})$ denote the gradient and Hessian of the log-likelihood with respect to $\tilde{\beta}$ and $\tilde{\eta}$, respectively. By some simple algebra, we obtain

$$ll(\beta) \approx \frac{1}{2}(z(\tilde{\eta})-X\beta)^T ll^{'}(\tilde{\eta})(z(\tilde{\eta})-X\beta)+C(\tilde{\eta},\tilde{\beta}), \tag{4.9}$$

where $z(\tilde{\eta}) = \tilde{\eta} - ll^{'}(\tilde{\eta})/ll^{''}(\tilde{\eta})$ is the adjusted dependent variable, and $C(\tilde{\eta},\tilde{\beta})$ is a constant term that does not depend on β. To speed up the algorithm, rather than using the full $N \times N$ $ll^{''}(\tilde{\eta})$ matrix, we use the diagonal elements of $ll^{''}(\tilde{\eta})$ in our algorithm. The i^{th} diagonal element is denoted as $ll^{''}(\tilde{\eta})_i$. We define $z(\tilde{\eta})_i = \tilde{\eta}_i - ll^{'}(\tilde{\eta})_i/ll^{''}(\tilde{\eta})_i$. Therefore, Eq.(4.7) can be simplified as a penalized IRLS:

$$\min_{\beta} -\frac{1}{N}\sum_{i=1}^{N} ll^{''}(\tilde{\eta})_i(z(\tilde{\eta})_i - X_i\beta)^2 + \lambda P(\beta) \tag{4.10}$$

The partial derivative of the IRLS with respect to the k^{th} coordinate, $k = 1, 2, \cdots, p$, can be calculated as:

$$-\frac{1}{N}\sum_{i=1}^{N} ll^{''}(\tilde{\eta})_i x_{ik}(z(\tilde{\eta})_i - X_i\beta) + \lambda\alpha \cdot sgn(\beta_k) + \lambda(1-\alpha)\beta_k \tag{4.11}$$

where $sgn(\cdot)$ is the signum function. Hence, the coordinate-wise update of the penalized IRLS will take the following form:

$$\hat{\beta}_k = \frac{S(-\frac{1}{N}\sum_{i=1}^{N} ll^{''}(\tilde{\eta})_i x_{ik}(z(\tilde{\eta})_i - \sum_{j\neq k} x_{ij}\beta_j), \lambda\alpha)}{-\frac{1}{N}\sum_{i=1}^{N} ll^{''}(\tilde{\eta})_i x_{ik}^2 + \lambda(1-\alpha)} \tag{4.12}$$

where $S(Z,\ \gamma) = sgn(Z) \cdot (|Z| - \gamma)_+$ is the soft-thresholding operation, and $ll^{'}(\tilde{\eta})_i$ and $ll^{''}(\tilde{\eta})_i$ can be calculated as follows:

$$ll^{'}(\tilde{\eta})_i = \begin{cases} \frac{\partial g_1}{\partial \tilde{\eta}_i} = -\frac{1}{\sigma} \cdot \frac{f^{'}(\varepsilon_i)}{f(\varepsilon_i)} & \text{if} \quad \delta_i = 1 \\ \frac{\partial g_2}{\partial \tilde{\eta}_i} = -\frac{1}{\sigma} \cdot \frac{-f(\varepsilon_i)}{1-F(\varepsilon_i)} & \text{if} \quad \delta_i = 0 \end{cases}$$

$$ll^{''}(\tilde{\eta})_i = \begin{cases} \frac{\partial^2 g_1}{\partial \tilde{\eta}_i^2} = \frac{1}{\sigma^2} \cdot \frac{f^{''}(\varepsilon_i)}{f(\varepsilon_i)} - \left(\frac{\partial g_1}{\partial \tilde{\eta}_i}\right)^2 & \text{if} \quad \delta_i = 1 \\ \frac{\partial^2 g_2}{\partial \tilde{\eta}_i^2} = \frac{1}{\sigma^2} \cdot \frac{-f^{'}(\varepsilon_i)}{1-F(\varepsilon_i)} - \left(\frac{\partial g_2}{\partial \tilde{\eta}_i}\right)^2 & \text{if} \quad \delta_i = 0 \end{cases}$$

where $f(\cdot)$ is the density function of the selected distribution, $F(\cdot)$ is the corresponding cumulative distribution function, and $f^{'}(\cdot)$ and $f^{''}(\cdot)$ denote the gradient and Hessian of the density function [19], respectively. In this paper, we choose the Gaussian distribution, Logistic distribution, and Extreme value distribution as the baseline distributions. It should be noted that, besides these three distributions that are being described in this paper, our framework is suitable for all other parametric distributions once the corresponding functions for the distributions are calculated.

All the analysis until this point has assumed a fixed σ. We will also vary the value of σ in our learning scheme by making σ another coordinate that is updated once all of the coefficient variables are updated. In the proposed algorithm, we use the Newton-Raphson method to update $\log\sigma$, which can be written in the following form:

$$\log\sigma = \log\tilde{\sigma} - ll^{'}(\log\tilde{\sigma})/ll^{''}(\log\tilde{\sigma}) \tag{4.13}$$

where $\tilde{\sigma}$ is learned in the previous iteration, $ll^{'}(\log\tilde{\sigma}) = \frac{1}{N}\sum ll^{'}(\log\tilde{\sigma})_i$, and $ll^{''}(\log\tilde{\sigma}) = \frac{1}{N}\sum ll^{''}(\log\tilde{\sigma})_i$. Additionally, $ll^{'}(\log\tilde{\sigma})_i$ and $ll^{''}(\log\tilde{\sigma})_i$ can be calculated as follows:

$$ll^{'}(\log\tilde{\sigma})_i = \begin{cases} \frac{\partial g_1}{\partial \log\tilde{\sigma}_i} = -\frac{\varepsilon_i f^{'}(\varepsilon_i)}{f(\varepsilon_i)} & \text{if} \quad \delta_i = 1 \\ \frac{\partial g_2}{\partial \log\tilde{\sigma}_i} = -\frac{-\varepsilon_i f(\varepsilon_i)}{1-F(\varepsilon_i)} & \text{if} \quad \delta_i = 0 \end{cases}$$

$ll^{''}(\log\tilde{\sigma})_i =$

$$\begin{cases} \frac{\partial^2 g_1}{\partial(\log\tilde{\sigma}_i)^2} = \frac{\varepsilon_i^2 f^{''}(\varepsilon_i)+\varepsilon_i f^{'}(\varepsilon_i)}{f(\varepsilon_i)} - \left(\frac{\partial g_1}{\partial \log\tilde{\sigma}_i}\right)^2 & \text{if } \delta_i = 1 \\ \frac{\partial^2 g_2}{\partial(\log\tilde{\sigma}_i)^2} = \frac{-\varepsilon_i^2 f^{'}(\varepsilon_i)}{1-F(\varepsilon_i)} - \frac{\partial g_1}{\partial \log\tilde{\sigma}_i}\cdot\left(1+\frac{\partial g_1}{\partial \log\tilde{\sigma}_i}\right) & \text{if } \delta_i = 0 \end{cases}$$

Good initial values of the coefficients and σ turn out to be vital for successful optimization, especially in a high-dimensional data set. In coordinate descent, the coefficient vector usually starts with the zero vector because the L_1 norm penalty induces lot of zero elements in the coefficient vector. For σ, a clever starting point is introduced in [19], where the model is fit starting with the mean and variance of each feature. As the normalization makes the values of each feature and the target value in the data set have zero mean and unit variance, the initial values of the iteration only depend on the mean and variance of the selected distribution, denoted as μ and s^2, respectively. The μ and s^2 of the selected baseline distributions can be found in the Appendix.

4.3 The URPCR Algorithm Algorithm 1 outlines the overall steps involved in the proposed model including the main optimization method. In lines 1-5, the dependent variable is calculated based on the user setting. In line 6, the coefficient vector and $\hat{\sigma}$ are initialized by a zero vector and s (the standard deviation of the selected distribution), respectively. In lines 9-14, the weight and adjusted dependent variables of the IRLS for each training instance are calculated. In line 15, one coordinate is updated based on coordinate descent. In lines 17-21, the updated formulas are calculated after all the p coefficients have been updated. Finally, in lines 22-24, the σ is updated based on these new updated equations. Note that, at each time, only one variable in the vector $\tilde{\beta}$ is being updated, and hence $\tilde{\eta}_i$ can be updated based on the previous iteration's result in O(1). Thus, one complete cycle of coordinate descent through all p variables costs O(Np) operations, and the σ can be updated in O(N) operations. Hence the total computation cost for each optimization step of the proposed algorithm is O(Np).

Usually, in the learning process, the model has to be trained based on a series of values for λ, and the best λ is selected via cross-validation. In this paper, we build a pathwise solution similar to the approach given in [18]; initialize λ to a sufficiently large number, which forces β to a zero vector, and then gradually decrease λ in each learning iteration. For a new λ, the initial values of β and σ are the estimated β and σ learned from the previous λ as a warm start, so the initial values of β and σ are not far from the optimal value, and the algorithm can converge in few iterations. The convergence of the Newton step in the algorithm is not guaranteed; it may become unstable if the initial parameter is far from the optimal value. However, in a pathwise solution, the warm start is not far from the optimal value, so it solves the convergence problem to a large extent.

Algorithm 1: URPCR Algorithm

Input: Training data (X, T, δ), Regularization parameter λ, Adjustment Weight α, Selected Distribution, flag AFT
Output: $\hat{\beta}$, $\hat{\sigma}$

1 **if** *AFT==TRUE* **then**
2 $y = \log(T)$;
3 **else**
4 $y = T$;
5 **end**
6 **Initialize**: $\hat{\beta} \leftarrow \mathbf{0}$, $\hat{\sigma} \leftarrow s$;
7 **repeat**
8 **for** $k = 1$ *to* p **do**
9 **for** $i = 1$ *to* N **do**
10 Calculate $\tilde{\eta}_i = X_i\tilde{\beta}$, $\varepsilon_i = \frac{y_i-\tilde{\eta}_i}{\tilde{\sigma}}$;
11 Calculate $f(\varepsilon_i)$, $F(\varepsilon_i)$, $f^{'}(\varepsilon_i)$, and $f^{''}(\varepsilon_i)$;
12 Calculate $ll^{'}(\tilde{\eta})_i$ and $ll^{''}(\tilde{\eta})_i$;
13 Update $z(\tilde{\eta})_i = \tilde{\eta}_i - ll^{'}(\tilde{\eta})_i/ll^{''}(\tilde{\eta})_i$;
14 **end**
15 $\tilde{\beta}_k \leftarrow \frac{S(-\frac{1}{N}\sum_{i=1}^{N} ll^{''}(\tilde{\eta})_i x_{ik}(z(\tilde{\eta})_i - \sum_{j\neq k} x_{ij}\tilde{\beta}_j),\lambda\alpha)}{-\frac{1}{N}\sum_{i=1}^{N} ll^{''}(\tilde{\eta})_i x_{ik}^2+\lambda(1-\alpha)}$;
16 **end**
17 **for** $i = 1$ *to* N **do**
18 Calculate $\tilde{\eta}_i = X_i\tilde{\beta}$, $\varepsilon_i = \frac{y_i-\tilde{\eta}_i}{\tilde{\sigma}}$;
19 Calculate $f(\varepsilon_i)$, $F(\varepsilon_i)$, $f^{'}(\varepsilon_i)$, and $f^{''}(\varepsilon_i)$;
20 Calculate $ll^{'}(\log\tilde{\sigma})_i$ and $ll^{''}(\log\tilde{\sigma})_i$;
21 **end**
22 Calculate $ll^{'}(\log\tilde{\sigma})$ and $ll^{''}(\log\tilde{\sigma})$;
23 Update $\log\sigma$ based on Eq.(4.13);
24 $\tilde{\sigma} = exp(\log\sigma)$;
25 **until** *Convergence of* $\tilde{\beta}$ *and* $\tilde{\sigma}$;
26 $\hat{\beta} \leftarrow \tilde{\beta}$, $\hat{\sigma} \leftarrow \tilde{\sigma}$;

5 Experimental Results

In this section, we will first describe the datasets used in our evaluation and then provide the performance results along with the implementation details.

5.1 Dataset Description For our evaluation, we used several publicly available high-dimensional gene expression cancer survival benchmark datasets[1]. The datasets we used in our experiments are as follows:

- The Norway/Stanford Breast Cancer Data (NSBCD) contains gene expression measurements of 115 women

[1] http://user.it.uu.se/~liuya610/download.html

with breast cancer. The missing values are imputed using 10-nearest neighbor imputation (which is a common practice in the biomedical domain).

- Lung adenocarcinoma (Lung) is a dataset containing observations of 86 early-stage lung adenocarcinoma patients.
- The Dutch Breast Cancer Data (DBCD) contains information on 4919 gene expression levels of a series of 295 women with breast cancer. Measurements were taken from the fresh-frozen-tissue bank of the Netherlands Cancer Institute.
- Diffuse Large B-Cell Lymphoma (DLBCL) is a dataset that contains Lymphochip DNA microarrays from 240 biopsy samples of DLBCL tumors.

All of these datasets measure cancer survival using gene expression levels. Table 1 provides the details of the datasets that are being used in this paper. In this table, the column titled "# Censored" corresponds to the number of censored instances in each dataset. We used 5-fold cross validation when the number of instances is greater than 150 and 3-fold cross validation otherwise.

Table 1: Details of the datasets used in this paper.

Dataset	**# Instances**	**# Features**	**# Censored**
NSBCD	115	549	77
Lung	86	7129	62
DBCD	295	4919	216
DLBCL	240	7399	102

5.2 Evaluation Metrics The concordance index (C-index), or *concordance probability*, is used to measure the performance of prediction models in survival analysis [8]. Let us consider a pair of bivariate observations $(y_1, \hat{y}_1)$ and $(y_2, \hat{y}_2)$, where y_i is the actual observation, and $\hat{y}_i$ is the predicted one. The concordance probability is defined as

$$c = Pr(\hat{y}_1 > \hat{y}_2 | y_1 \geq y_2). \tag{5.14}$$

By definition, the C-index has the same scale as the area under the ROC (AUC) in binary classification, and if y_i is binary, then the C-index is same as the AUC. In the hazard ratio-based regression models, the instances with a low hazard rate should survive longer, and the C-index is calculated as follows:

$$c = \frac{1}{num} \sum_{i \in \{1 \cdots N | \delta_i = 1\}} \sum_{y_j > y_i} I[X_i \hat{\beta} > X_j \hat{\beta}]$$

where num denotes the number of comparable pairs and $I[\cdot]$ is the indicator function. The C-index in other censored regression methods, which directly target the survival time, should be calculated as:

$$c = \frac{1}{num} \sum_{i \in \{1 \cdots N | \delta_i = 1\}} \sum_{y_j > y_i} I[S(\hat{y}_j | X_j) > S(\hat{y}_i | X_i)]$$

where $S(\hat{y}_i | X_i)$ is the predicted target value for X_i.

We also evaluate performance by re-formulating the problem into a binary classification problem where we choose a particular time point, and each patient is given a label corresponding to whether the patient is alive at that time point or not. For this binary classification task, we evaluate the prediction performance using AUC [4].

5.3 Implementation Details The proposed model is implemented using C++ with the Eigen library[2], and in each iteration, the weight updates for all N instances (lines 9-14 and lines 17-21 of Algorithm 1) are calculated in parallel.

All of the methods used in our comparisons are implemented in R. The Cox and unregularized parametric censored regression are obtained from the *survival* package [19]. In the *survival* package, the *coxph* function is employed to train the Cox model. The Tobit regression is trained using the *survreg* function. The parametric censored regressions are trained using the *survreg* function with Normal, Log-normal, Logistic, Log-logistic, and Weibull distributions. Three sparse regression methods, namely LASSO-COX, EN-COX, and EN-BJ, which are penalized versions using lasso and elastic net penalty terms, are also used for our comparisons. LASSO-COX and EN-COX are built using the *cocktail* function in the *fastcox* package [26], while EN-BJ is implemented using the *bujar* package [24].

Boosting concordance index (BoostCI) [15] for survival data is an approach where the concordance index metric is modified to an equivalent smoothed criterion using the sigmoid function. In addition to the above survival methods, we also compared our methods with ordinary least squares (OLS) because URPCR is a generalized linear model. In our experiments, the Gaussian distribution, Logistic distribution, and Extreme value distribution are chosen as the baseline distributions. For each dataset, the validation data is used to select the appropriate distribution and to decide whether the dependent variable y should be the observed time T or the logarithm of the observed time $\log(T)$.

5.4 Results and Discussion Table 2 provides the C-index values obtained by various regression methods on the real-world high-dimensional micro-array cancer

[2] `http://eigen.tuxfamily.org/`

Table 2: Performance comparison of the proposed URPCR method and seven other existing related methods using C-index values (along with their standard deviations).

DataSet	COX	LASSO-COX	EN-COX	BoostCI	OLS	Tobit	EN-BJ	URPCR
NSBCD	0.441 (0.059)	0.591 (0.109)	0.605 (0.100)	0.626 (0.083)	0.633 (0.111)	0.373 (0.021)	0.622 (0.092)	**0.693** (**0.056**)
Lung	0.514 (0.137)	0.668 (0.087)	0.664 (0.066)	0.571 (0.088)	0.572 (0.061)	0.470 (0.132)	0.663 (0.128)	**0.771** (**0.039**)
DBCD	0.529 (0.063)	0.685 (0.042)	0.719 (0.030)	0.705 (0.038)	0.560 (0.072)	0.487 (0.078)	0.718 (0.040)	**0.735** (**0.027**)
DLBCL	0.510 (0.029)	0.624 (0.042)	**0.637** (**0.036**)	0.595 (0.017)	0.505 (0.089)	0.492 (0.052)	0.623 (0.061)	0.631 (0.056)

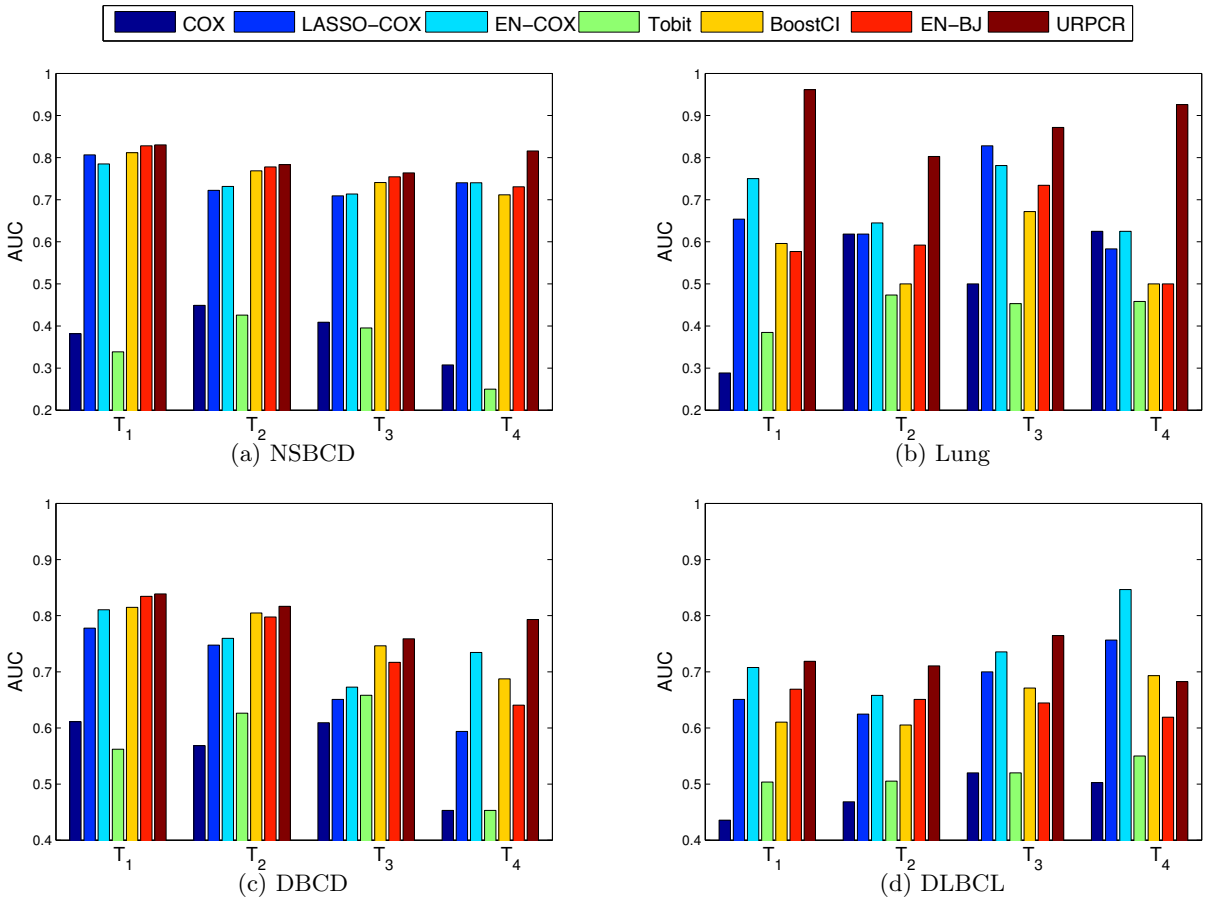


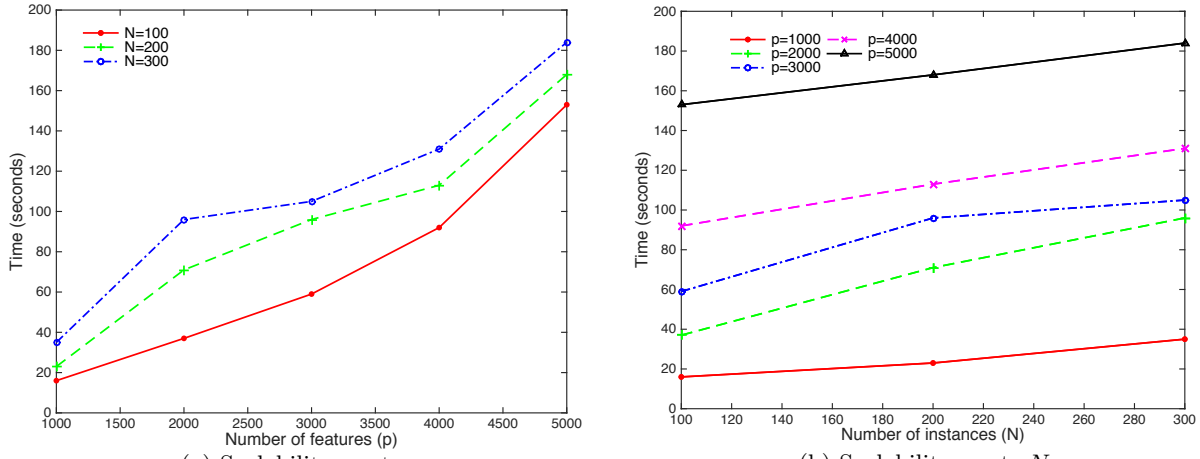
Figure 1: AUC values for binary classification of survival times for four different time thresholds. The URPCR is compared to six different survival regression methods. For each plot, T_1, T_2, T_3, and T_4 are the time thresholds corresponding to the timepoints at which 25%, 50%, 75%, and 100% of events have occurred, respectively.

datasets. The results show that our proposed URPCR model obtains higher C-index in most of the datasets.

Figure 1 provides histogram plots of the AUC values for the binary classification task on each dataset with four different time splits corresponding to the time points when 25%, 50%, 75%, and 100% of events have occurred. The AUC values for our proposed models are higher than those of the existing survival prediction methods in all but one task, which further reinforces the accuracy of our proposed model compared to the other survival prediction methods; we exclude OLS in the plots since it is not a survival regression method. These results demonstrate that our proposed model is able to predict temporal event occurrence at different

Table 3: Performance comparison of the proposed regularized censored regressions and unregularized censored regressions with different distributions using C-index values (along with their standard deviations).

	Normal		Log-normal		Logistic		Log-logistic		Weibull	
	original	URPCR	original	URPCR	original	URPCR	original	URPCR	original	URPCR
NSBCD	0.373 (0.021)	**0.667 (0.065)**	0.444 (0.054)	**0.682 (0.039)**	0.379 (0.020)	**0.693 (0.056)**	0.238 (0.050)	**0.667 (0.042)**	0.304 (0.153)	**0.688 (0.074)**
Lung	0.470 (0.132)	**0.736 (0.028)**	0.411 (0.075)	**0.712 (0.020)**	0.566 (0.095)	**0.771 (0.039)**	0.587 (0.066)	**0.762 (0.041)**	0.428 (0.101)	**0.762 (0.068)**
DBCD	0.487 (0.078)	**0.716 (0.030)**	0.491 (0.057)	**0.735 (0.027)**	0.490 (0.088)	**0.721 (0.063)**	0.527 (0.025)	**0.723 (0.059)**	0.458 (0.104)	**0.708 (0.036)**
DLBCL	0.492 (0.052)	**0.626 (0.057)**	0.320 (0.078)	**0.625 (0.056)**	0.491 (0.044)	**0.498 (0.279)**	0.431 (0.125)	**0.581 (0.099)**	0.396 (0.084)	**0.631 (0.056)**



(a) Scalability w.r.t. p (b) Scalability w.r.t. N

Figure 2: *Scalability results*: Plots of the runtimes of URPCR with the extreme value distribution. The times denote total runtimes for ten λ values averaged over five trials.

time points effectively without the need to re-train a new classifier at each time point.

Table 3 provides the C-index values obtained from the original censored regression and the URPCR regularized parametric censored regression methods based on different distributions, where Log-normal and Log-logistic denote that the logarithm of the observed time is assumed to follow the normal distribution and logistic distribution, respectively. It should be noted that Weibull distribution is a special case of the generalized extreme value distribution. The results show that, with sparsity-inducing penalization, our proposed model is able to improve the prediction performance of the parametric censored regression on the high-dimensional datasets for different kinds of distributions.

5.5 Scalability Experiments We also empirically evaluate the scalability of the proposed algorithm with respect to sample size (N) and the number of features (p). All synthetic datasets are generated using the function "simple.surv.sim" in *survsim* package [16] with different sample size and feature dimensionality. All the features are generated based on the uniform distribution, and each of them have a different randomly set interval. The coefficient vector is also randomly generated and remain within $[-1, 1]$. The observed time is assumed to follow a Log-logistic distribution, and time to censorship follows a Weibull distribution. All timing calculations are carried out on an Intel Xeon 3 GHz processor with 16 cores (32 threads). Figure 2(a) shows runtimes for fixed N and varying p, and Figure 2(b) shows runtimes for fixed p and varying N. These two plots suggest that the runtime of URPCR is close to being *linear in both N and p*. Notice that the the lines in Figure 2(b) increase more slowly than the lines in Figure 2(a), which indicates that our proposed URPCR model has better scalability with respect to N than with respect to p. This is because, in our implementation, the weight updates of all N instances (lines 9-14 and lines 17-21 of Algorithm 1) are calculated in parallel.

6 Conclusion

In this paper, we developed a unified model for regularized parametric censored regression that is able to efficiently handle high-dimensional (right) censored data. The elastic net penalty is used to induce sparseness into the resulting coefficients, thus avoiding over-fitting the data, especially in high-dimensional scenarios. In order to unify the learning scheme for various popular distributions, we used Taylor expansion to approximate the objective function as a generalized linear model and solved the penalized iterative reweighted least squares problem via a cyclical coordinate descent-based method. We compared the performance of the proposed URPCR algorithm with several state-of-the-art censored regression methods using various publicly available high-dimensional microarray gene expression survival benchmark datasets. We also demonstrated the linear scalability of the proposed model with respect to both the number of samples and the number of features. Our results also show that the proposed unified regularized model significantly outperforms original unregularized variants of the parametric methods when the same underlying distributions are used for modeling. We plan to extend this work in the future by using other recent structured regularization terms such as group lasso and tree lasso in the context of survival analysis.

Acknowledgments This work was supported in part by the US National Science Foundation grants IIS-1527827 and IIS-1231742.

References

[1] M. Aitkin and D. Clayton, *The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM*, Applied Statistics, 29 (1980), pp. 156–163.

[2] S. Bennett, *Log-logistic regression models for survival data*, Applied Statistics, (1983), pp. 165–171.

[3] J. Buckley and I. James, *Linear regression with censored data*, Biometrika, 66 (1979), pp. 429–436.

[4] L. E. Chambless and G. Diao, *Estimation of time-dependent area under the ROC curve for long-term risk prediction*, Statistics in Medicine, 25 (2006), pp. 3474–3486.

[5] D. R. Cox, *Regression models and life-tables*, Journal of the Royal Statistical Society. Series B (Methodological), 34 (1972), pp. 187–220.

[6] M. J. Crowther and P. C. Lambert, *A general framework for parametric survival analysis*, Statistics in Medicine, 33 (2014), pp. 5280–5297.

[7] B. Efron, *The efficiency of Cox's likelihood function for censored data*, Journal of the American Statistical Association, 72 (1977), pp. 557–565.

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

[9] J. Huang, S. Ma, and H. Xie, *Regularized estimation in the accelerated failure time model with high-dimensional covariates*, Biometrics, 62 (2006), pp. 813–820.

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

[11] J. P. Klein and M.-J. Zhang, *Survival analysis, software*, Wiley Online Library, 2005.

[12] E. T. Lee and J. Wang, *Statistical methods for survival data analysis*, vol. 476, Wiley, 2003.

[13] Y. Li, V. Rakesh, and C. K. Reddy, *Project success prediction in crowdfunding environment*, in Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM), ACM, 2016.

[14] Y. Li, B. Vinzamuri, and C. K. Reddy, *Regularized weighted linear regression for high-dimensional censored data*, in In Proceedings of SIAM International Conference on Data Mining, SIAM, 2016.

[15] A. Mayr and M. Schmid, *Boosting the concordance index for survival data–a unified framework to derive and evaluate biomarker combinations*, PLoS ONE, 9 (2014), p. e84483.

[16] D. Morina and A. Navarro, *The R package survsim for the simulation of simple and complex survival data*, Journal of Statistical Software, 59 (2014), pp. 1–20.

[17] C. K. Reddy and Y. Li, *A review of clinical prediction models*, in Healthcare Data Analytics, C. K. Reddy and C. C. Aggarwal, eds., Chapman and Hall/CRC Press, 2015.

[18] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for Cox's proportional hazards model via coordinate descent*, Journal of Statistical Software, 39 (2011), pp. 1–13.

[19] T. Therneau, *A package for survival analysis in S. R package version 2.37-4*, URL http://CRAN.R-project.org/package=survival, (2013).

[20] R. Tibshirani et al., *The lasso method for variable selection in the Cox model*, Statistics in Medicine, 16 (1997), pp. 385–395.

[21] J. Tobin, *Estimation of relationships for limited dependent variables*, Econometrica: Journal of the Econometric Society, 26 (1958), pp. 24–36.

[22] B. Vinzamuri, Y. Li, and C. K. Reddy, *Active learning based survival regression for censored data*, in Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, 2014, pp. 241–250.

[23] S. Wang, B. Nan, J. Zhu, and D. G. Beer, *Doubly penalized Buckley-James method for survival data with high-dimensional covariates*, Biometrics, 64 (2008), pp. 132–140.

[24] Z. Wang and C. Wang, *Buckley-James boosting for survival analysis with high-dimensional biomarker data*, Statistical Applications in Genetics and Molecular Biology, 9 (2010), pp. 1–33.

[25] L. Wei, *The accelerated failure time model: a useful alternative to the cox regression model in survival analysis*, Statistics in Medicine, 11 (1992), pp. 1871–1879.

[26] Y. Yang and H. Zou, *A cocktail algorithm for solving the elastic net penalized Cox's regression in high dimensions*, Statistics and its Interface, 6 (2012), pp. 167–173.

[27] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2005), pp. 301–320.