Solving the Cox Proportional Hazards Model and Its Applications

Jessica Ko

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-110
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-110.html

May 20, 2017

Solving the Cox Proportional Hazards Model and Its Applications

by Jessica Ko

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Laurent El Ghaoui
Research Advisor

5/8/2017
(Date)

* * * * * * *

Professor Adityanand Guntuboyina
Second Reader

5/9/2017
(Date)

Solving the Cox Proportional Hazards Model and Its Applications

by

Jessica Ko

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Laurent El Ghaoui, Chair
Professor Aditya Guntuboyina

Spring 2017

Solving the Cox Proportional Hazards Model and Its Applications

Abstract

Solving the Cox Proportional Hazards Model and Its Applications

by

Jessica Ko

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Laurent El Ghaoui, Chair

The Cox proportional hazards model allows data to be analyzed with a concept of survival and death over time. Unlike a lot of other traditional models, there is a clear relationship of how the risk of death is affected by time and the features of the data. The model is equivalent to a generalized linear model, and the ℓ_1 regularized Cox model can be solved by coordinate descent. In addition, a condition to eliminate features is explored to save computational time in solving the maximization of the partial log likelihood. The Cox model can be applied to many tasks because of its unique survival aspect. The probabilities of surviving past a certain time are used to predict loan defaults. Understanding which characteristics correlate with survival for dogs and cats in animal shelters is also possible through creating survival curves.

Acknowledgments

I would like to thank my advisor, Professor El Ghaoui, for being supportive, kind, and patient during my time here at Berkeley. I appreciate all the help and guidance he has given me throughout my undergraduate and graduate careers. I am glad to have had the opportunity to work with him as I have learned a lot about machine learning and optimization. I thank him for his valuable advice and comments while I was working on this thesis. I would also like to thank Professor Guntuboyina for reading this work.

I owe endless thanks to my parents, sister, and relatives for their unwavering love, support, and encouragement through this arduous journey. This would not have been possible without my parents, as they have demonstrated how much can be accomplished solely by hard work and passion. I am thankful for my sister who has been a great friend, always believing in me. As clichè as this sounds, there are not enough words to describe how grateful I am for how much my family has done for me.

Contents

Chapter 1

Introduction

Survival analysis is a field dedicated to analyzing the time to the occurrence of an event of interest. Traditionally, survival analysis is used in bio-statistics to determine the chances of a patient surviving after undergoing some treatment. For example, this can be applied to analyzing cancer patients after receiving chemotherapy. Data is recorded from the patient over time, and the outcome after the study is noted. Some possible outcomes are dying, being cured, or exiting the study. However, survival analysis can be applied to other areas to analyze customers leaving over time and recidivism of prisoners once they were released.

One method used in survival analysis is the Cox proportional hazards model or Cox model, which uniquely quantifies the risk of the event of interest occurring over time [7]. Throughout this work, survival will be considered as when the event of interest did not occur. In addition, the model can define the effect of features on survival and can determine how likely the outcome will occur after a certain time for predicting whether an event will occur. Moreover, features can be investigated to determine if there is a correlation for being more likely to occur. In Chapter 2, the theory behind the Cox model will be described.

Throughout this work, the ℓ_1 norm is added to the Cox proportional hazards model because regularization encourages sparsity and prevents overfitting. Then, cyclic coordinate descent is used to solve this problem. Previous work has been done on using coordinate descent for solving the Cox proportional hazards model with elastic net, which is implemented in R. Cyclic coordinate descent is shown to be successful for convex problems with ℓ_1, ℓ_2, or elastic net penalties because it exploits the sparsity of the model and has an explicit form for each coordinate-wise maximization [11]. Furthermore, cyclic coordinate descent is an efficient algorithm for the regularized Cox model [20]. Coordinate descent has been proven to be useful for solving other models such as elastic-net penalized regression models [13, 26]. In Chapter 3, a cyclic coordinate descent method for solving a regularized Cox model will be explained, and its higher accuracy compared to existing methods will be discussed.

In addition, safe feature elimination (SAFE) is applied to the model to speed up the computation of solving the model in Chapter 4. Applying SAFE to the Lasso problem has shown to reduce running time [9]. The computational effort of the feature elimination step is negligible compared to solving the Lasso problem. Furthermore, SAFE can be applied on other ℓ_1 penalized convex problems like the Cox model.

After the Cox model is solved, the model can be used for a variety of applications. Chapter 5 provides applications of the Cox model for two scenarios. Predicting defaults on loans and analyzing characteristics of animals with different outcomes in animal shelters are successful using the Cox model.

Chapter 2

Cox Proportional Hazards Model

The Cox proportional hazards model accurately depicts interactions between the features and risk in the *hazard function* [7]. Time-dependent features can also be easily used in the model to account for features that may change with time. Even though time-dependent features are not considered in this work, they are powerful for creating a model that precisely describes the interactions of the features. Given a vector $x \in \mathbb{R}^d$ of d features and a parameter $\beta \in \mathbb{R}^d$, the *hazard function* is defined as

$$\lambda(t|x) = \lambda_0(t)e^{\beta^\top x}.$$

The *baseline hazard function* $\lambda_0(t)$ does not need to be specified for the Cox model, making it semi-parametric. This is advantageous because the Cox model will be robust and have fewer restrictions. The baseline hazard function is appropriately named because it describes the risk at a certain time when $x = 0$, which is when the features are not incorporated. The hazard function describes the relationship between the baseline hazard and features of a specific sample to quantify the hazard or risk at a certain time.

The model only needs to satisfy the proportional hazard assumption, which is that the hazard of one sample is proportional to the hazard of another sample [6]. This property can be checked by using p-values of the Cox model as described in Chapter 5. Two samples x_1 and x_2 satisfy this assumption when the ratio is not dependent on time as shown below.

$$\frac{\lambda(t|x_1)}{\lambda(t|x_2)} = \frac{\lambda_0(t)e^{\beta^\top x_1}}{\lambda_0(t)e^{\beta^\top x_2}} = \frac{e^{\beta^\top x_1}}{e^{\beta^\top x_2}}$$

Also, more generally the *relative risk* to the average risk of the training data is defined below for sample x_k and sample mean $\hat{x}$.

$$\frac{\lambda(t|x_k)}{\lambda(t|\hat{x})} = \frac{\lambda_0(t)e^{\beta^\top x_k}}{\lambda_0(t)e^{\beta^\top \hat{x}}} = \frac{e^{\beta^\top x_k}}{e^{\beta^\top \hat{x}}}$$

2.1 β Estimation by Partial Likelihood

The parameter β can be found by maximizing the partial likelihood because the hazard function is not specified. The following sections explain the formulation of the optimization problem. In order to be used for the Cox model, each sample i needs:

- x_i a feature vector;
- T_i time when event occurred or the censoring time, which is the last time the sample is observed if the event of interest did not occur in the time period;
- D_i death indicator where 1 is for when death occurred and 0 is for censoring

2.1.1 Equivalence to Poisson Regression

The partial likelihood of the Cox model can be fitted by the likelihood of Poisson regression, a generalized linear model, because the likelihoods are proportional to each other [25]. The advantage of the estimates of β being the same is that it can be fitted using software for generalized linear models like in R. Alternatively, the estimate from the Cox model can be used for Poisson regression. In Chapter 3, a coordinate descent method is proposed for solving the maximum partial likelihood of the Cox model.

The Cox model can be interpreted in terms of a Poisson regression. Given the cumulative hazard $\Lambda(t)$ and sample i, the estimates of β can be obtained by treating the death indicator D_i as Poisson distributed with mean $\mu_i = \Lambda(t_i)e^{\eta_i}$ where $\eta = \beta^\top x$. The link function is modified to be $\beta^\top x = \log(\mu_i) - \log(\Lambda(t_i))$ [16]. More information about the cumulative hazard is in Chapter 2.2.

2.1.2 Partial Likelihood

In order to formulate the partial likelihood, the f unique failure times are ordered increasingly $t_1 < \cdots < t_f$ and $j(i)$ is the index of the sample failing at time t_i. When at most one sample failed at each time, the partial likelihood for the Cox model can be written as

$$\begin{aligned}\mathcal{L}(\beta) &= \prod_{i=1}^{f} \frac{\lambda_0(t_i)e^{\beta^\top x_{j(i)}}}{\sum_{j\in R_i}\lambda_0(t_i)e^{\beta^\top x_j}} \\ &= \prod_{i=1}^{f} \frac{e^{\beta^\top x_{j(i)}}}{\sum_{j\in R_i} e^{\beta^\top x_j}}\end{aligned}$$

where the risk set R_i is the set of indices of samples with death or censor times occurring after t_i or $R_i = \{k | T_k \geq t_i\}$. This represents the probability of failure occurring to a sample at time t_i among those at risk at time t_i. The semi-parametric property can be exhibited here because the baseline hazard λ_0 gets canceled out.

However, the partial likelihood above does not take tied events into account, so the probabilities are not as accurate. Tied events occur if the number of deaths d_i at time t_i is greater than 1. Breslow introduces a different partial likelihood function to deal with the ties [5]. Given that $I(i)$ is the set of indices where a sample fails at time t_i or $I(i) = \{k | D_k = 1 \, and \, T_k = t_i\}$, the partial likelihood can be redefined as shown below.

$$\mathcal{L}(\beta) = \prod_{i=1}^{f} \frac{e^{\left(\sum_{s \in I(i)} \beta^\top x_s\right)}}{\left(\sum_{j \in R_i} e^{\beta^\top x_j}\right)^{d_i}}$$

This work will refer to the Breslow ties version as the partial likelihood because ties often occur in the datasets.

2.1.3 Minimize Negative Partial Log Likelihood

The parameter β can be found by minimizing the negative partial log likelihood $\ell(\beta)$, which is defined below.

$$\begin{aligned} \ell(\beta) &= \log(\mathcal{L}(\beta)) \\ &= \sum_{i=1}^{f} \log \frac{e^{\sum_{s \in I(i)} \beta^\top x_s}}{\left(\sum_{j \in R_i} e^{\beta^\top x_j}\right)^{d_i}} \\ &= \sum_{i=1}^{f} \left[\left(\sum_{s \in I(i)} \beta^\top x_s \right) - d_i \log \sum_{j \in R_i} e^{\beta^\top x_j} \right] \end{aligned}$$

The minimization of the negative log likelihood with ℓ_1 regularization is formed below in the optimization problem with objective $f(\beta)$.

$$f(\beta) = -\ell(\beta) + \lambda \|\beta\|_1$$

$$\min_{\beta} f(\beta) \tag{2.1}$$

Regularization is included because there are many benefits such as being more accurate than stepwise selection and yielding interpretable models [22]. In addition, the regularization prevents degenerate behavior when there are more predictors than observations.

2.2 Survival Function of Cox Model

The *survival function* obtained from the Cox model can be used to make predictions on a sample surviving because it's the probability of a sample surviving after time t. The survival function is defined as

$$S(t) = \exp\left(-\Lambda(t)\right).$$

The *cumulative hazard* or *cumulative risk* $\Lambda(t)$ is defined as

$$\Lambda(t) = \int_0^t \lambda(s)ds$$

where $\lambda(t)$ is the hazard function, the instantaneous probability of death at time t, given survival until t [18]. It can also be rewritten as

$$\lambda(t) = -\frac{d}{dt}\log S(t).$$

The survival function can be rewritten at time t for a given sample x

$$\begin{aligned} S(t|x) = \exp(-\Lambda(t)) &= \exp\left(-\int_0^t \lambda(s|x)ds\right) \\ &= \exp\left(-\int_0^t \lambda_0(s)e^{\beta^\top x}ds\right) \\ &= \exp\left(-e^{\beta^\top x}\int_0^t \lambda_0(s)ds\right) \\ &= S_0(t)^{e^{\beta^\top x}} \end{aligned}$$

where the *cumulative baseline hazard* is $\Lambda_0(t) = \int_0^t \lambda_0(s)ds$ and the *baseline survival function* is $S_0(t) = e^{-\Lambda_0(t)}$ [19] . The survival function $S(t|x)$ or probability of survival after time t is defined by $S_0(t)^{e^{\beta^\top x}}$. The parameter β can be recovered by a coordinate descent method discussed in Chapter 3. Although the hazard function is not needed for the parameter estimation by partial likelihood, it is necessary to find the survival function for prediction. In the following sections, two methods for estimating the cumulative baseline hazard $\Lambda_0(t)$ will be discussed.

Breslow Estimator

The Breslow estimator of the cumulative baseline hazard is defined below [15].

$$\Lambda_0(t) = \sum_{i:T_i\leq t} \frac{d_i}{\sum_{j\in R_i} e^{\beta^\top x_j}} \tag{2.2}$$

The cumulative hazard is estimated by using the expected number of failures in a time period $(t, t + \delta t)$.

$$d_i \approx \delta t \sum_{j \in R_i} \lambda_0(t) e^{\beta^\top x_j}$$

$$\delta t \lambda_0(t_i) \approx \frac{d_i}{\sum_{j \in R_i} e^{\beta^\top x_j}}$$

By summing over the times, the cumulative hazard function is derived to show Equation 2.2 [27].

Weibull Distribution

The hazard function can be estimated by the Weibull distribution $\lambda_0(t) \sim Weibull(\lambda, k)$ [21].

$$\lambda_0(t) = (\lambda k)(\lambda t)^{k-1}$$

The cumulative hazard function can then be written as

$$\Lambda_0(t) = 1 - e^{-(\lambda t)^k}.$$

The hazard or risk is increasing when $k > 1$, so deaths or failures are more likely to occur as time progresses. Similarly, hazard or risk is decreasing when $k < 1$.

After the parameter β and the cumulative baseline hazard is estimated, the probability of surviving after time t or the survival function can be recovered. The probability of survival after time t can be used for predictions by considering samples where $S(t) > 0.5$ as surviving. Chapter 5 shows an application of using survival functions to predict loan defaults.

Chapter 3

Solving the Model by Coordinate Descent

Cyclic coordinate descent is used to solve the minimum partial log likelihood with the ℓ_1 norm. Throughout this chapter, it is assumed that the ℓ_1 norm is included in the optimization problem as shown in Equation 2.1. The separability structure of the cost function, where the partial log likelihood $\ell(\beta)$ is differentiable and convex and $\|\beta\|_1$ is convex, guarantees that the coordinate descent algorithm will converge to the optimal solution [23, 24]. The algorithm cycles between fixing each index and solving the minimization problem. Each individual problem is written as below where f_k is the objective function corresponding to fixing all indices of β except for k.

$$\min_{\beta_k} f_k(\beta_k)$$

Each individual minimization problem for a fixed index is solved by the bisection method. The benefit of this method is that calculating the derivative with respect to one variable is not as costly as calculating the gradient [8].

3.1 Coordinate Descent Bound

The method starts with an interval where the optimal index k of the parameter β_k^* falls under. A lower bound L and upper bound U can be found such that $\beta_k^* \in [L_k, U_k]$ and $L_k \leq U_k$. Using some guidelines, it is possible to find bounds for the optimal parameter.

Property 1. If $f_k'(L_k) < 0 < f_k'(U_k)$, then $\beta_k^* \in [L_k, U_k]$ and $L_k \leq U_k$.

Proof: $f_k(\beta_k)$ is a convex function, so the derivative is monotonically increasing. $L_k \leq U_k$ because $f_k'(L_k) \leq f'(U_k)$. The derivative of a convex function is continuous, so by the Intermediate Value Theorem, $\exists \beta_k^*$ such that $f_k'(\beta_k^*) \approx 0$ and $\beta_k^* \in [L_k, U_k]$. The global minimum is found at this critical point.

When the bound, L_k and U_k, can be found for all indices k of β, Property 1 can be used. The first estimate can be a number closer to 0 because the ℓ_1 regularization will likely cause the parameters to be sparse. A simple guideline is to set the bounds so that $L_k < 0 < U_k$. The bounds can be continually doubled until Property 1 is satisfied.

However, it is possible that $f'_k(\beta_k) < 0$ or $f'_k(\beta_k) > 0$ for all β_i and Property 1 will never be satisfied. In this case, when the bounds get sufficiently large, then a bound cannot be defined precisely and the optimal values likely lie close to infinity. The heuristics used to find a bound for coordinate descent are shown in Algorithm 1.

```
for index $k$ in $\beta$ do
    initialize $L_k, U_k$ where $L_k < 0 < U_k$ ;
    while $U_k - L_k <$ Max Limit and ( $f'_k(L_k) \geq 0$ or $f'_k(U_k) \leq 0$ ) do
        if $f'_k(L_k) \geq 0$ then
            $L_k \leftarrow 2L_k$ ;
        end
        if $f'_k(U_k) \leq 0$ then
            $U_k \leftarrow 2U_k$ ;
        end
    end
end
```

Algorithm 1: Optimal β_k Bound

3.2 Coordinate Descent Algorithm

After finding the bound for the optimal β, coordinate descent and the bisection method can be used on the model. For coordinate descent, all indexes in β are fixed except β_k. The derivative of the objective function, $f'_k(\beta_k)$, with respect to index k where $x_{s,t}$ indicates the t feature of sample s is defined by Equation 3.1 . Because the ℓ_1 is not differentiable at 0, a subgradient is introduced to handle this case [4] . Thus, the subgradient of $\|x\|_1 = g(x)$.

$$g(x) = \begin{cases} +1, & \text{if } x > 0 \\ -1, & \text{otherwise} \end{cases}$$

Using this as the subgradient for ℓ_1 norm, the following can be defined as

$$\begin{aligned} f'_k(\beta_k) &= \frac{d}{d\beta_k}\left[-\ell(\beta) + \lambda\|\beta\|_1\right] \\ &= \sum_{i=1}^{f}\left[\left(\sum_{s\in I(i)} x_{s,k}\right) - \left(\frac{d_i}{\sum_{j\in R_i} e^{\beta^\top x_j}}\right)\left(\sum_{j\in R_i} x_{j,k}e^{\beta^\top x_j}\right)\right] + \lambda g(\beta_k). \end{aligned} \tag{3.1}$$

The bisection method will attempt to find β such that $f'(\beta^*) \approx 0$. The full algorithm is shown in Algorithm 2.

```
initialize $\beta$ ;
while $\beta$ Not Converged do
    for index $k$ in $\beta$ do
        Fix all indexes except $i$ in $\beta$;
        Consider interval $[L_k, U_k] \in \beta_k^*$ ;
        $l \leftarrow L_k$;
        $u \leftarrow U_k$;
        while $\beta_k$ Not Converged do
            $x \leftarrow (l + u)/2$ ;
            if $f_k'(x) < 0$ then
                $l \leftarrow x$ ;
            else
                $u \leftarrow x$ ;
            end
        end
        $\beta_i \leftarrow x$;
    end
end
```

Algorithm 2: Coordinate Descent

In Appendix A, more optimizations can be included to speed up computation time by writing the equations in matrix form and caching values.

3.3 Comparison with Existing Packages

In this section, there is a brief comparison of the coordinate descent algorithm mentioned in this chapter to existing packages in R. The `survival` package in R contains a function `coxph` that also fits data to the Cox model, but the model is not regularized. This R package will not be as good for sparse data and may tend to overfit.

While the `survival` package does not have the regularization term, the `glmnet` package includes the regularization term in the form of elastic net, which includes the ℓ_1 and ℓ_2 norm [12]. In addition, `glmnet` can solve other generalized linear models. Using 12,000 samples with 14 features from the loan dataset that will be explored in Chapter 5, the `glmnet` package performs very fast in less than one second for any value of λ, but the coordinate descent method ranges from a few seconds to about one minute using $L_k = -10$ and $U_k = 10$ for all indices. The exact timings of the coordinate descent method are described in more detail in Table 4.1. However, for `glmnet`, accuracy in the minimizing the negative log likelihood is sacrificed for faster computation times.

The objective function values shown in Table 3.1 are obtained by solving the Cox model with `glmnet` or the coordinate descent method and computing the objective function values using the optimal parameter. The `glmnet` package seems to aggressively zero out many indices, which may lead to its reduced accuracy. This effect is noticeable in the table because the objective function value is the same for $\lambda \geq 0.5$.

Table 3.1: Compare Objective Function of `glmnet` and Coordinate Descent

λ	`glmnet`	Coordinate Descent
0	51695.84	51120.60
0.5	51536.44	51121.23
10	51536.44	51128.52
50	51536.44	51153.70
100	51536.44	51179.85

The coordinate descent method achieves better accuracy by getting a smaller value for the objective function at optimum at the expense of a longer computation time. The longer computation time can be justified because the model only needs to be fit to the data once for every λ value.

Chapter 4

Safe Feature Elimination

By forming the dual of the Cox model, safe feature elimination (SAFE) can be applied to the model to eliminate features that are not present after solving the optimization problem. Performing the feature elimination has many computation benefits because it can greatly reduce the time needed to solve the optimization problem. The feature elimination step can be parallelized because each feature can be screened independently of each other. In addition, each elimination step requires significantly little computation to perform. This method of feature elimination is shown to be very beneficial for ℓ_1-penalized least-square regression problems [9].

4.1 Dual of Cox Model

In order to define the dual, the optimization problem for the Cox model can be rewritten, assuming the data is in a matrix $X = [x_1, \dots, x_n] \in \mathbb{R}^{d \times n}$ for n samples and $\beta \in \mathbb{R}^d$. A failure times matrix can be defined as $\Delta := (\delta_{ij}) \in \{0,1\}^{f \times n}$ where f is the number of unique failure times and $\delta_{ij} = 1$ if $j \in R_i$.

Assuming $Z = \mathbf{1}\beta^\top X \in \mathbb{R}^{f \times n}$ such that $Z_{ij} = \beta^\top x_j$ for every i where $\mathbf{1}$ is a vector of ones in $\mathbb{R}^f$, the maximization problem from Chapter 2 can be rewritten as

$$\begin{aligned} p^* &= \max_{\beta, Z} c^\intercal \beta - \sum_{i=1}^{f} d_i \log \left(\sum_{j \in R_i} e^{Z_{ij}} \right) - \lambda \|\beta\|_1 \\ &= \max_{\beta, Z} c^\intercal \beta - \sum_{i=1}^{f} d_i \log \left(\sum_{j=1}^{n} \delta_{ij} e^{Z_{ij}} \right) - \lambda \|\beta\|_1 \end{aligned}$$

where $c = \displaystyle\sum_{\{i | D_i = 1\}} x_i \in \mathbb{R}^d$.

Using a dual variable $U \in \mathbb{R}^{f \times n}$, the dual can be written as

$$p^* = \min_U \max_{\beta, Z} c^\intercal \beta - \sum_{i=1}^{f} d_i \log \left(\sum_{j=1}^{n} \delta_{ij} e^{Z_{ij}} \right) - \lambda \|\beta\|_1 + \operatorname{Tr} U^\top (Z - \mathbf{1}\beta^\top X). \tag{4.1}$$

Using $U^\top = [u_1, \ldots, u_f]$ and $Z^\top = [z_1, \ldots, z_f]$ where $u_i, z_i \in \mathbb{R}^n$, the trace can be rewritten

$$\operatorname{Tr} U^\top Z = \sum_{i=1}^{f} u_i^\intercal z_i.$$

The dual problem can be rewritten as

$$p^* = \min_U \sum_{i=1}^{f} \max_{z_i} u_i^\intercal z_i - d_i \log \left(\sum_{j=1}^{n} \delta_{ij} e^{Z_{ij}} \right) : \|XU^\top \mathbf{1} - c\|_\infty \leq \lambda.$$

For each i, consider each optimization problem

$$\max_{z_i} u_i^\intercal z_i - d_i \log \left(\sum_{j=1}^{n} \delta_{ij} e^{Z_{ij}} \right) : \|XU^\top \mathbf{1} - c\|_\infty \leq \lambda.$$

The solution is

$$= \begin{cases} d_i \sum_{j=1}^{n} U_{ij} \log U_{ij} & \text{if } u_i \geq 0, \mathbf{1}^\top u_i = 1, \forall\, j \ : \ u_{ij}(1 - \Delta_{ij}) = 0, \\ +\infty & \text{otherwise} \end{cases}$$

where Δ_j is the jth column of Δ. The dual can then be written as

$$p^* = \min_U \sum_{i=1}^{f} d_i \sum_{j=1}^{n} U_{ij} \log U_{ij} : \|XU^\top \mathbf{1} - c\|_\infty \leq \lambda, U\mathbf{1} = \mathbf{1}, U \geq 0, U \circ \Delta = 0$$

where $\circ$ represents element-wise multiplication.

4.2 Feature Elimination

After the dual is formed, the criteria for the safe feature elimination can be derived. For each feature $k = 1, \ldots, d$, if the following holds, then $\beta_k =$ at optimum where f_k is the kth row of X.

$$\lambda > \max_U |f_k^\top U^\top \mathbf{1} - c_k| : U\mathbf{1} = \mathbf{1}, U \geq 0, U \circ \Delta = 0 \tag{4.2}$$

This can be shown by looking at Equation 4.1 to get $\beta_k^* = 0$ the following must be true at optimum

$$-\lambda |\beta_k^*| + (f_k^\top U^\top \mathbf{1} - c_k)\beta_k^* < 0. \tag{4.3}$$

If all U in the feasible set satisfies Equation 4.3, then the feature can be eliminated. The following shows how the equation can be related to Equation 4.2.

$$\max_{U,\beta_k}(f_k^\top U^\top \mathbf{1} - c_k)\beta_k = \max_{U,\beta_k} |(f_k^\top U^\top \mathbf{1} - c_k)\beta_k| < \lambda|\beta_k|$$

By dividing both sides by $|\beta_k^*|$ and dropping the maximization of β_k, the equivalence is shown. In addition, the absolute value can be added to the objective function because β_k is a scalar and the signs of $f_k^\top U^\top \mathbf{1} - c_k$ and β_k will be matching when maximized.

To obtain the feature elimination rule, the maximization problem needs to be solved. First, consider the below expression for each f_k

$$S_+(f_k) = \max_U f_k^\top U^\top \mathbf{1} : U\mathbf{1} = \mathbf{1}, U \geq 0, U \circ \Delta = 0.$$

Using duality, a lower bound is found

$$\begin{aligned}
S_+(f_k) &= \min_Z \max_{U\geq 0,\, U\mathbf{1}=\mathbf{1}} f_k^\top U^\top \mathbf{1} + \operatorname{Tr} Z^\top((\Delta - \mathbf{1}\mathbf{1}^\top) \circ U) \\
&= \min_Z \max_{U\geq 0,\, U\mathbf{1}=\mathbf{1}} \operatorname{Tr} U^\top((\Delta - \mathbf{1}\mathbf{1}^\top) \circ Z + \mathbf{1} f_k^\top) \\
&= \min_Z \sum_{i=1}^{f} \max_{1\leq j\leq n} ((\delta_{ij} - 1)Z_{ij} + f_{kj}) \\
&= \sum_{i=1}^{f} \min_z \max_{1\leq j\leq n} (f_{kj} + (\delta_{ij} - 1)z_j) \\
&= \sum_{i=1}^{f} \min_z \max\left(\max_{j\,:\,\delta_{ij}=1} f_{kj}, \max_{j\,:\,\delta_{ij}=0} f_{kj} - z_j\right) \\
&\geq \sum_{i=1}^{f} \max_{j\,:\,\delta_{ij}=1} f_{kj}.
\end{aligned}$$

It can also be shown that $S_+(f_k) \leq \sum_{i=1}^f \max_{j\,:\,\delta_{ij}=1} f_{kj}$ by choosing $Z_{ij} = 0$ for $\delta_{ij} = 1$, $Z_{ij} = \max_{h\,:\,\delta_{ih}=0} f_{ih} - \max_{h\,:\,\delta_{ih}=1} f_{ih}$ otherwise. As a result, the expression can be written as

$$S_+(f_k) = \sum_{i=1}^{f} \max_{j\,:\,\delta_{ij}=1} f_{kj}.$$

This implies

$$\begin{aligned}
S_-(f_k) &= \min_U f_k^\top U^\top \mathbf{1} : U\mathbf{1} = \mathbf{1}, U \geq 0, U \circ \Delta = 0 \\
&= -S_+(-f_k) \\
&= \sum_{i=1}^{f} \min_{j\,:\,\delta_{ij}=1} f_{kj}.
\end{aligned}$$

Using these two expressions, Equation 4.2 can be written as

$$\lambda > \max\left(|S_+(f_k) - c_k|, |S_-(f_k) - c_k|\right)$$

$$\lambda > \max\left(c_k - \sum_{i=1}^{f} \min_{j\,:\,\delta_{ij}=1} X_{kj}, \sum_{i=1}^{f} \max_{j\,:\,\delta_{ij}=1} X_{kj} - c_k\right). \tag{4.4}$$

If the kth feature satisfies the SAFE condition (Equation 4.4), then $\beta_k = 0$ at optimum.

4.3 Speed Ups Using SAFE

The feature elimination is computationally faster than not using SAFE because many computations are saved. The SAFE condition only needs to be checked once for every feature in the beginning of the optimization. In addition, the SAFE condition is very quick to check because of the form of the expression. An experiment was run on 12,000 samples with 14 features from the loan dataset that will be explored in Chapter 5. The results of the timings with and without the SAFE condition are shown in Table 4.1. For these times, the coordinate descent bound is chosen to be $[-10, 10]$ for each index. The computations are run on a machine with 8 GB of memory and an 2.6GHz dual-core Intel Core i5 processor.

With smaller values of λ, the model does not eliminate as many features, so using SAFE is only 16% faster than not using SAFE for solving the minimization problem. However, when the regularization weight is larger and more features are eliminated, there is significant computation advantage. As shown in the table, the largest regularization weight performs 442% faster when using the SAFE conditions. Using safe feature elimination will speed up the time it takes to solve the optimization problem especially when the regularization weights are larger.

Table 4.1: Timing Computations with and without SAFE in Seconds

λ	No SAFE	SAFE	% Faster Using SAFE	# Features Satisfying SAFE Condition
0.0001	131.0	109.70	19.42%	2
0.5	116.29	99.77	16.56%	2
10	107.5	92.49	16.23%	2
50	77.42	65.84	17.59%	3
100	71.35	33.85	110.78%	4
200	36.48	15.79	131.03%	7
300	30.41	5.61	442.07%	9

Chapter 5

Applications

In this chapter, the Cox model will be applied to two different datasets: predicting loan defaults and investigating the relationship of characteristics of animals in shelters and survival. In order for the Cox model to be applicable to a dataset, there needs to be a clear definition of elapsed time and what event is considered to be death or failure. The following datasets have both properties, so the Cox model can be applied to these tasks. Both of these applications solve the regularized Cox model by using coordinate descent with SAFE.

5.1 Predicting Loan Defaults

Deciding whether or not to approve a loan is important for banks because issuing loans that are likely to default are risky and decrease the bank's profitability. By using the Cox model to aid in predicting loan defaults, banks can make a better decision about issuing loans. In addition, this will help borrowers financially plan by allowing them to understand how likely they can get a loan. Furthermore, this can be applied to loans on Lending Club to help investors discover good investments by finding loans that are less likely to default. Lending Club is one of the world's largest online credit marketplace that allows for peer-to-peer lending.

In the following sections, the dataset on loans issued from 2007 to 2016 is used for predicting loan defaults, a binary task [14]. The dataset contains about one million samples and 110 features, but only 15,000 samples and a subset of features will be used for the predictions. The training, validation, and testing sets were split so that the two classes are balanced.

Data

The time period was calculated by finding the number of months that have passed since the loan was issued and when the last payment was received. Even though loans are issued on different dates, all the start dates can be assumed to be the same time without loss of generality [25]. Failure of the loan is defined as 'Charged Off', 'Default', or 'Does not meet the credit policy. Status:Charged Off'.

The loan default dataset contains many features, and a subset of these features was selected to use for prediction. First, the features were preprocessed. Features with continuous values were normalized into their z-scores like annual income, and categorical variables were one-hot encoded like home ownership. The following features were selected and their descriptions from the dataset are shown [14]

- 'annual_inc': Self-reported annual income.
- 'dti': A ratio calculated using the borrowers' total monthly debt payments on the total debt obligations, excluding mortgage and the requested Lending Club loan, divided by the borrowers self-reported monthly income.
- 'emp_length': Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
- 'funded_amnt': The total amount committed to that loan at that point in time.
- 'grade': Lending Club assigned loan grade ('A', 'B', 'C', 'D', 'E', 'F', or 'G')
- 'home_ownership' : Home ownership status of the borrower such as own, rent, mortgage, or other.
- 'int_rate': Interest rate of loan
- 'loan_amnt' : Listed amount of the loan applied for by the borrower. The credit department can reduce the loan amount.
- 'mort_acc': Number of mortgage accounts.
- 'num_bc_sats': Number of satisfactory bankcard accounts.
- 'num_bc_tl': Number of bankcard accounts.
- 'pub_rec_bankruptcies': Number of public record bankruptcies.
- 'revol_bal': Total credit revolving balance.
- 'term': Number of payments on the loan can either be 36 or 60 months.

In order to use the Cox model, the proportional hazard assumption must hold for the data. The assumption can be verified with a p-value < 0.05 for the features selected using `coxph` in R. Using this assumption, there are 14 features selected : 'dti', 'emp_length', 'funded_amnt', 'grade = C', 'grade = D', 'grade = F', 'home_ownership=OWN', 'home_ownership=OTHER', 'int_rate', 'mort_acc', 'num_bc_sats', 'num_bc_tl', 'pub_rec_bankruptcies', and 'term= 36 months'.

Results

Using the survival function of the Cox model, the failure of a loan can be predicted when the probability of surviving after time t is less than 0.5 or $S(t) < 0.5$. In addition, if the relative risk to the average of the samples is greater than one, then the sample may be considered to fail. Using different λ values for regularization and different methods of determining failure, the test set accuracy can be compared in Table 5.1. Accuracy is defined as the number of correctly classified samples divided by the total number of samples. Overall, the predictions based on survival probabilities were better than predictions based on relative risk. The Breslow estimator seems to perform very poorly as most of the accuracies are worse than randomly guessing. Accuracies using the Weibull estimator are slightly better.

Table 5.1: Accuracies from Survival Probabilities and Relative Risk

λ	Survival Prob. Breslow Estimator	Survival Prob. Weibull Estimator	Relative Risk
0	**0.48933**	0.5	0.36
0.5	0.489	0.5793	0.35933
10	0.48833	0.57267	0.361
50	0.485	0.57633	0.36033
100	0.48766	**0.58133**	0.361
150	0.48667	0.58	**0.45067**

Even though the survival probabilities and relative risk were not good for predicting loan failures, properties from the Cox model can be used to enhance predictions by combining them with the original features and using them on different machine learning algorithms. The test accuracies of using different machine learning models and feature sets are shown in Table 5.2. Hyperparameters of each model are chosen by tuning them based on the accuracy of the validation set. In addition, the neural network with two hidden layers had the activation function, optimization solver, learning rate, and layer sizes adjusted according to the performance on the validation set. The first column is the accuracies for only training on the original features mentioned earlier. The other three columns contain not only the original features but also features from the Cox model like relative risk.

Table 5.2: Accuracies of Models Using Different Features

Model	Original Features Only	Relative Risk from Cox Model	Survival Prob. from Cox Model	Risk and Prob. from Cox Model
Logistic Regr.	0.66367	0.66067	0.664	0.66367
SVM	0.588	0.591	0.589	0.59167
Decision Tree	**0.67667**	**0.68967**	**0.68767**	**0.66867**
Random Forest	0.66067	0.66633	0.66833	0.66367
Neural Network	0.65633	0.65367	0.65733	0.65667

Overall, the features from the Cox model, relative risk and survival probabilities, slightly increased the accuracy of predicting the loan failures. Even though the survival probabilities were bad predictors on their own, combining them with the original features improved the accuracy. This is shown when comparing the accuracies of only using the original features with the accuracies of also using Cox model features. When comparing the different models, the decision tree performs the best across the different feature sets. Further improvements in the accuracy can possibly be made by using the Cox model in the loss function of a neural network [2].

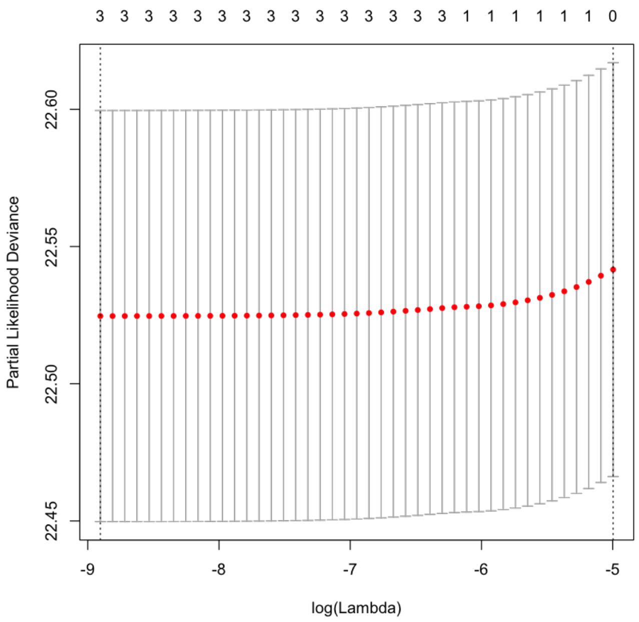
5.2 Survival Analysis on Dogs and Cats in Animal Shelters

Animal shelters are often overcrowded with animals because they lack the resources to care for homeless pets and more people disown their pets than adopt them. In this section, correlations between animal characteristics and survival once entering an animal shelter over time will be explored. Understanding these correlations can help improve animal shelters and the well-being of these animals. Many different factors can affect the survival of animals in pet shelters. In order to understand the correlations between these features like color, breed, and etc. on survival, an appropriate model like the Cox model must be used. Survival analysis is applied on the Austin Animal Shelter dataset to explore survival of animals in shelters [1]. The Austin Animal Shelter operates the largest No Kill municipal animal shelter in the United States. Even though the animals are not killed for population control, overcrowding can still be a problem if there are too many animals in the shelter and not enough resources to care for them in the facilities.

The dataset contains 49,970 animals recorded from 2013 to 2016 such as outcome time, outcome type (adoption, transfer, euthanasia, or death), and etc. Even though the animals have different dates for when they entered the animal shelter, the start times can be interpreted as starting at the same with no loss of generality [25] . The time is measured as the difference between intake time and outcome time. The animal shelter collected a variety of categorical features like intake type (Owner, Stray, or PublicAssist), injured/sick or not, pregnant or not, dog or cat, spayed/neutered or not, gender, purebred or mix, and black colored or not. Similar to Chapter 5.1, features satisfying the Cox proportionality assumption can only be used, so this reduces the feature set to only spayed/neutered or not, gender, and black colored or not. For this dataset, survival is defined as an animal being adopted or transferred to another center. Failure, or death, is defined as when an animal dies or is euthanized.

In order to graph the survival curves, the best regularization weight λ needs to be chosen. Unlike classification tasks, there is not a well defined accuracy metric to pick the best λ value, so another metric must be used. The λ with the lowest cross-validated deviance is chosen for the model [17]. The deviance is calculated by doing 10-fold cross validation. Figure 5.1 shows how the deviances slowly increases as $\log(\lambda)$ increases, so $\log(\lambda) = -8.9$ is chosen.

Figure 5.1: Picking the Best λ



After finding β by coordinate descent and SAFE, the survival curves can now be graphed to compare the difference in the survival probabilities for certain features. For the feature of interest, the data is separated by the feature values. For example, the data is separated by male and female for gender. After, the mean $\hat{x}$ of each feature is calculated, and the index of the feature is marked as the respective value. For example, if gender is feature i, then $\hat{x_1}[i] = 1$ for female and $\hat{x_0}[i] = 0$ for male. The two mean vectors, one for each binary class, are then graphed using the survival function $S(t|\hat{x_0})$ or $S(t|\hat{x_1})$ for all failure times with the Breslow estimator for the cumulative baseline hazards. The survival curves for the three features are shown in Figures 5.2, 5.3, and 5.4. Each downward step in the graphs indicates an event of failure.

Figure 5.2:

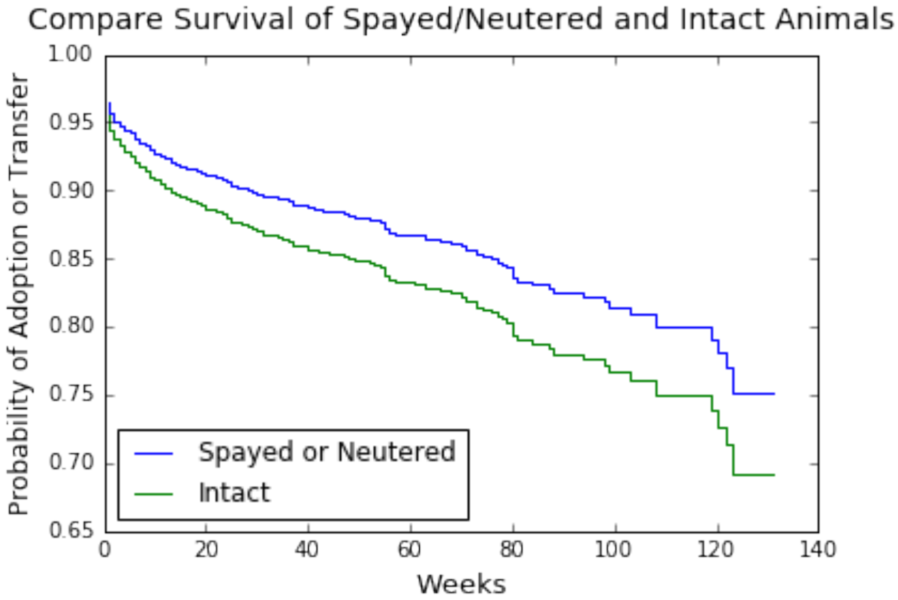


Figure 5.3:

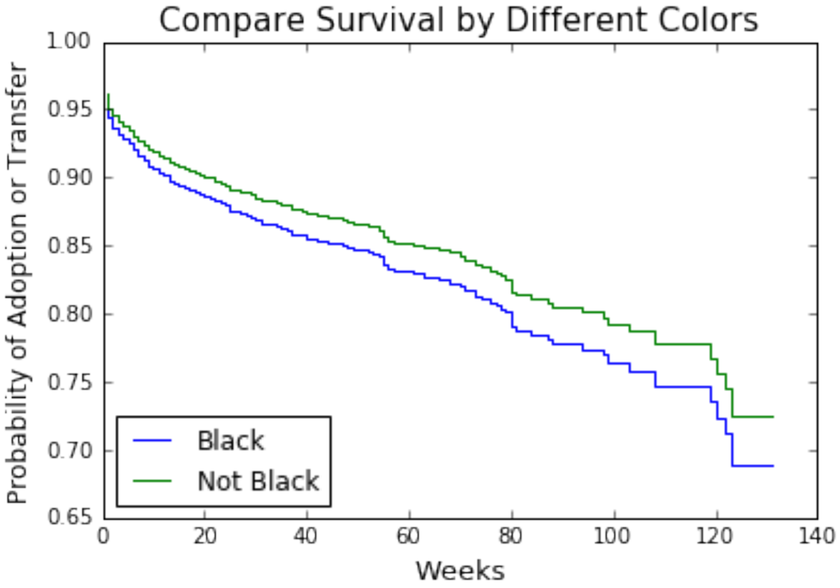


Figure 5.4:

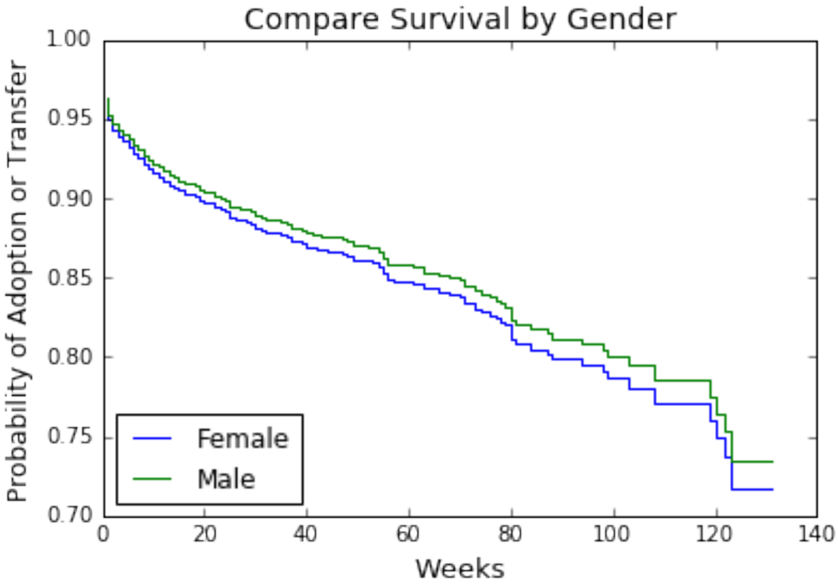


Figure 5.2 shows that spayed or neutered animals correlate more with survival. People may be more likely to adopt a pet who has undergone this procedure because it can save costs and help curb overpopulation. Figure 5.3 shows that black colored animals correlate with lower survival. Black cats are sometimes symbolized as bad luck, and darker colored dogs are often portrayed as being aggressive in media. This observation may be related to black dog syndrome, which is a tendency for black dogs to be adopted less frequently. Compared to the other factors, gender does not seem to correlate with survival as greatly as shown in Figure 5.4.

Because other features did not satisfy the proportional hazards assumption, they could not be analyzed with the model and limited what kind of features could be explored, even though there are several other features available in the dataset like breed and intake type. The modified version, the stratified general Cox, could adjust those features that do not satisfy the proportional hazards assumption [3]. More work can be done in the future to incorporate more features for the model.

Chapter 6

Conclusion

In this work, the Cox proportional hazards model with ℓ_1 regularization is solved by coordinate descent with the bisection method. Through experiments, this method is more accurate but slower in computations compared to other methods. The safe feature elimination step reduces the running time of solving the model without introducing long computations to perform the step.

The Cox model may be used for many applications because of the relationship between the risk of an event over time and features of the sample. Predicting loan defaults is explored with the Cox model by using the survival function. Even though the survival function itself performed poorly, using the Cox model for feature engineering such as the relative risk and survival function is effective in increasing the classification accuracy compared to models that do not use these features. In addition, the Cox model is used to explore how features affect the survival of animals in animals shelters. Certain characteristics of animals such as not being black colored and being spayed or neutered correlate to survival.

While solving the regularized Cox proportional hazards model with coordinate descent for its applications was successful, there are still many improvements and explorations to be made. Converging safe regions are shown to lead to faster convergence for Lasso and can possibly be explored for the Cox model [10]. To deal with features that do not satisfy the proportional hazards assumption, the stratified Cox model can be used, so the feature set can be expanded [3]. The Cox model can also be incorporated into the loss function of a neural network to further improve accuracies [2].

Appendix A

Additional Optimizations

In order to save addition computation, the model can be written in matrix form. The data is assumed to be $X \in \mathbb{R}^{n \times d}$ for n samples and d features and $\beta \in \mathbb{R}^d$. $\Delta_R \in \mathbb{R}^{f \times n}$ is a matrix where $\Delta_{R_{ij}} = 1$ if sample j failed at the time corresponding to index i and f is the number of unique times a sample failed. $\Delta_F \in \mathbb{R}^{n \times 1}$ is a matrix where $\Delta_{F_j} = 1$ if the sample j failed. $T \in \mathbb{R}^{f \times 1}$ is a vector where $T_i =$ number of ties at the time corresponding to index i.

The partial log likelihood is shown below where $\mathbf{1}_i$ is a one vector of shape $i \times 1$.

$$\Delta_F^\top X\beta - \left(\log(\Delta_R \exp(X\beta))\right) \mathbf{1}_f^\top$$

The gradient can be written as shown below for index k where component-wise division is used.

$$\Delta_F^\top x_k - \left(\frac{T \circ (\Delta_R(x_k \circ \exp(X\beta)))}{\Delta_R \exp(X\beta)} \right) \mathbf{1}_f^\top$$

During the bisection method for index k, some values are saved to prevent calculating the same value multiple times. For each k, set $\beta_k = 0$ first and save $\beta_{k=0} = e^{X\beta}$. When using the bisection method, the value at β_k changes at each step, so to get the correct value perform the following

$$\exp X\beta = \beta_{k=0} \circ \exp \beta_k X_k$$

where $X_k \in \mathbb{R}^{n \times 1}$ is the kth column of X.

Bibliography

[1] *Austin Animal Center*. 2016. URL: `http://www.austintexas.gov/department/animal-services`.

[2] Bart Baesens et al. "Neural network survival analysis for personal loan data". In: *Journal of the Operational Research Society* 56.9 (2005), pp. 1089–1098.

[3] Lisa Borsi, Marc Lickes, and Lovro Soldo. *The stratified Cox Procedure*. 2011.

[4] S. Boyd. *Notes for EE364b, Stanford University*. Winter 2006-07.

[5] Norman Breslow. "Covariance analysis of censored survival data". In: *Biometrics* (1974), pp. 89–99.

[6] T Caye. *Evaluating Proportional Hazards Assumption*.

[7] D. R. Cox and D. Oakes. *Analysis of survival data*. Vol. 21. CRC Press, 1984.

[8] A. D'Aspremont. *Convex Optimization*. URL: `https://www.di.ens.fr/~aspremon/PDF/MVA/FirstOrderMethodsPartTwo.pdf`.

[9] Laurent El Ghaoui, Vivian Viallon, and Tarek Rabbani. *Safe feature elimination in sparse supervised learning*. 2010.

[10] Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. "Mind the duality gap: safer rules for the lasso". In: *arXiv preprint arXiv:1505.03410* (2015).

[11] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent". In: *Journal of statistical software* 33.1 (2010), p. 1.

[12] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1 (2010), pp. 1–22. URL: `http://www.jstatsoft.org/v33/i01/`.

[13] Jerome Friedman et al. "Pathwise coordinate optimization". In: *The Annals of Applied Statistics* 1.2 (2007), pp. 302–332.

[14] Nathan George. *All Lending Club loan data*. URL: `https://www.kaggle.com/wordsforthewise/lending-club`.

[15] Mary Lunn. *Proportional Hazards with a semi-parametric model called Cox regression*. URL: `http://www.stats.ox.ac.uk/~mlunn/lecturenotes2.pdf`.

[16] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall CRC, 1989.

[17] Carl M O'Brien. *Statistical Learning with Sparsity: The Lasso and Generalizations*. 2016.

[18] G. Rodriguez. *Lecture Notes on Generalized Linear Models*. 2007. URL: `http://data.princeton.edu/wws509/notes/`.

[19] Germán Rodrıguez. *Non-parametric estimation in survival models*. 2005.

[20] Noah Simon et al. "Regularization paths for Cox's proportional hazards model via coordinate descent". In: *Journal of statistical software* 39.5 (2011), p. 1.

[21] Lu Tian. *Survival Distributions, Hazard Functions, Cumulative Hazards*. URL: `https://web.stanford.edu/~lutian/coursepdf/unit1.pdf`.

[22] Robert Tibshirani et al. "The lasso method for variable selection in the Cox model". In: *Statistics in medicine* 16.4 (1997), pp. 385–395.

[23] Paul Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *Journal of optimization theory and applications* 109.3 (2001), pp. 475–494.

[24] Paul Tseng et al. "Coordinate ascent for maximizing nondifferentiable concave functions". In: (1988).

[25] John Whitehead. "Fitting Cox's regression model to survival data using GLIM". In: *Applied Statistics* (1980), pp. 268–275.

[26] Tong Tong Wu and Kenneth Lange. "Coordinate descent algorithms for lasso penalized regression". In: *The Annals of Applied Statistics* (2008), pp. 224–244.

[27] Li Yi. *More on Cox Model*. URL: `http://www-personal.umich.edu/~yili/lect5notes.pdf`.