bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Fenchel duality of Cox partial likelihood and its application in survival kernel learning

Christopher M. Wilson[a,*], Kaiqiao Li[b,*], Qiang Sun[c], Pei Fen Kuan[b,**], and Xuefeng Wang[a,**]

[a] *Department of Biostatistics and Bioinformatics, H. Lee Moffitt Cancer Center & Research Institute, Tampa, FL 33612, USA*
[b] *Department of Applied Math & Statistics, Stony Brook University, Stony Brook, NY 11794, USA*
[c] *Department of Statistical Sciences, University of Toronto, Ontario M5S 3G3, Canada*

Abstract

The Cox proportional hazard model is the most widely used method in modeling time-to-event data in the health sciences. A common form of the loss function in machine learning for survival data is also mainly based on Cox partial likelihood function, due to its simplicity. However, the optimization problem becomes intractable when more complicated regularization is employed with the Cox loss function. In this paper, we show that a convex conjugate function of Cox loss function based on Fenchel Duality exists, and this provides an alternative framework to optimization based on the primal form. Furthermore, the dual form suggests an efficient algorithm for solving the kernel learning problem with censored survival outcomes. We illustrate the application of the derived duality form of Cox partial likelihood loss in the multiple kernel learning setting

Keywords: Convex Conjugate; Cox model; Convex Optimization; Multiple Kernel learning; Fenchel Dual; Survival data

1. Introduction

The two most widely utilized models for censored survival data are the Cox proportional hazard (PH) model [1] and accelerated failure time (AFT), due

[*] These authors contributed equally to this work.
[**] Please address correspondence to xuefeng.wang@moffitt.org (Wang)

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

to their flexibility and efficiency [2]. Cox PH models are the most widely used model in health and clinical sciences, while the AFT model is more popular in areas like engineering. The Cox model is a semi-parametric regression method where no assumptions imposed on the baseline hazard function. The parametric regression coefficients quantify the effect size of each covariate and the exponential of the coefficient is interpreted as the unit increase of the hazard ratio. The Cox model works well in practice because it can tolerate a modest deviation from the PH assumption. The partial likelihood in the Cox model is defined as the probability that one individual will experience the event at a time t among those who have survived longer than t. It was shown that maximizing partial likelihood provides an asymptotically efficient estimation of regression coefficients [3]. The Cox model has been successfully extended to various high-dimensional settings, where the number of features is more than the number of samples. Additionally, the log partial likelihood (LPL) is differentiable and convex. Thus the combination of Cox LPL, and l1 (lasso), l2 (ridge), or elastic net penalties can be directly solved by standard Newton-Rapshon method. Li and Luan [4] pioneered methods for kernel Cox regression in the framework the penalization framework from the view of function estimation in reproducing kernel Hilbert spaces. Furthermore, the convex combined loss function often guarantees the convergence of efficient optimization algorithms including coordinate descent [5], which is the core algorithm implemented in a popular R package for penalized regression **glmnet**. The Cox LPL has also been adopted in many machine learning approaches as a loss function to the survival setting. For example, Ridgeway [6] adapted the gradient boosting method for the Cox model, which is implemented in the R package **gbm**. Li and Luan [7] also considered a boosting procedure using smoothing splines to estimate the proportional hazards models. More recently, the application of neural network-based deep learning techniques to the Cox PH model has begun to receive attention [8]. These machine learning techniques generalize the Cox model to include non-linear effects and to better address heterogeneous effects.

In this paper, we derive the Fenchel dual form of Cox partial likelihood,

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

which is a key step in the implementation of machine learning approaches for survival outcomes that can incorporate multiple high throughput data sources . Duality approach is a basic tool in machine learning and is commonly used in nonlinear programming and convex optimization to provide a lower bound approximation for the primal problem. It is often easier to optimize the lower bound via the dual form and it has fewer variables in the high dimensional setting. The main property of the resulting function, called Fenchel conjugate, is always convex regardless of the convexity of the original function. Note that the Lagrangian and Fenchel dual are defined under different contexts, even though many Lagrangian duals can be derived from Fenchel conjugate functions and in many cases, both of them are referred to as a "dual problem". Lagrangian duality form is defined within the context of the optimization problem (often with constraints), while the Fenchel form is more general and is defined for a function. Our work is motivated by the need to bridge the gap between modern machine learning techniques and survival models. To apply methods like SVM, survival data are often dichotomized with a cutoff time point. Such a method will yield biased results because censored data points are excluded from the analysis, additionally, the results will be affected by different cutoff values.

The remainder of the paper is organized as follows. In Section 2, we review the Cox proportional hazard model and Fenchel duality. In Section 3, we derive a conjugate function for the Cox model. We perform simulations in Section 4 to demonstrate the usage of the derived form in the multiple kernel learning. We analyze both Skin Cutaneous Melanoma (SKCM) gene and miRNA expression data from The Cancer Genome Atlas (TCGA). Finally we conclude with a discussion in Section 6.

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

2. Methods

2.1. Fenchel duality

Suppose we have function $f(x)$ on R^n, then the Fenchel convex conjugate of $f(x)$ is defined in terms of the supremum by

$$f^*(\rho) = \sup_{x \in R^n} \left(\rho^T x - f(x)\right).$$

The mapping from $f(.)$ to $f^*(.)$ defined above is also known as the Legendre-Fenchel transform. The convex conjugate function measures the maximum gap between line function $\rho^T x$ and original function $f(x)$, where each pair of $\left(\rho,\ f^T(\rho)\right)$ corresponds to a tangent line of the original function $f(x)$. The resulting function f^* has the nice property to be always convex, because it is the supremum of an affine function. Figure 1 illustrates how the conjugate dual for a classic example $f(x) = |x|$. The conjugate function offers one important option to build a dual problem that might be more tractable or computationally efficient than the primal problem. Note in Figure 1A, when $|\rho| > 1$ that as $x \to \infty$ then $|\rho|x \to \infty$, thus $\sup_x\{\rho x - |x|\} = \infty$. Alternative, when $|\rho| \le 1$ that $|\rho|x \le x$ for all x, hence the largest value for $\sup_x\{\rho x - |x|\} = 0$, when $x = 0$. The complex conjugate is illustrated in Figure 1B, note that is a convex function.

By Fenchel-Moreau theorem, $f = f^{**}$ if only and only if f is a convex and and lower-semi continuous function which holds for Cox proportional hazards model. Therefore, we can convert the problem into dual problem with f^* to obtain $\widehat{\rho}$ and map it back to our primal and obtain final solution. We define the relative interior of a set C as

$$ri(C) = \{x \in C |\text{ for all } y \in C \text{ there exists } \lambda > 1 \text{ such that } \lambda x + (1-\lambda)y \in C\},$$

in other words for an point $x \in C$ there exists a ball that is entirety contained in C. Additionally, using Fenchel duality theorem[9] (Theorem 31.1), we have the following statement. If $ri(dom(f)) \cap ri(dom(g)) \neq \emptyset$, and $f(\cdot)$ and $g(\cdot)$

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

are convex, we have

$$\inf_x \left(f\left(x\right) + g\left(x\right)\right) = \sup_y \left(-f^*\left(y\right) - g^*\left(-y\right)\right),$$

note that minimizing $h(x) = f(x) + g(x)$ occurs when $\nabla h = \nabla f + \nabla g = 0$ or when $\nabla f = -\nabla g$. Hence, Fenchel duality theorem shows us that minimizing the summation of two convex function can be reduced to the problem of maximizing the gap between their parallel tangent lines since it is the lower bound of primary problem. In our case, both the Cox loss function and EN regularizer are both convex, so we can apply this theorem that our target problem which reduces to the following problem

$$\max\left(-L^*\left(-\rho\right) - \delta_C\left(\|\rho\|_{K_m}\right)\right)$$

where L^* and ϕ are the conjugate function of L, K_m Hilbert space that is generated by the reporducing kernel K_m, and ϕ.

2.2. Cox proportional hazard model and partial likelihood

The Cox PH model [1] relates the covariates to the hazard function of the outcome at time t using the following equation,

$$h_i(t) = h_0(t)\exp\{\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle\},$$

where $h_0(t)$ is the baseline hazard function at time t and $\mathbf{x}_i$ is the vector of predictor variables for the i^{th} subject. An appealing feature of the Cox model is that, as shown in the partial likelihood

$$PL = \prod_{i\in D} \frac{\exp(\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)}{\sum_{l\in R}\exp(\langle \mathbf{x}_l, \boldsymbol{\beta}\rangle)} = \prod_{i\in D} \frac{\exp(\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle)}{\sum_{l\in R} I(t_l \geq t_j)\exp(\langle \mathbf{x}_l, \boldsymbol{\beta}\rangle)}, \qquad (2.1)$$

estimates of regression coefficients are obtained without parametric assumptions about the baseline hazard function. Here D is the set of uncensored subjects and R is the set of the observations at risk at time t. The PL can be understood as constructing the conditional probability that the event occurs to a particular subject at time t. Typically, we optimize the negative log of Cox PL 2.1,

$$\mathcal{L} = -\log(PL) = -\sum_{i=1}^{n} \delta_i \left(\langle \mathbf{x}_i, \boldsymbol{\beta}\rangle - \log\sum_{l\in R}\exp\left(\langle \mathbf{x}_l, \boldsymbol{\beta}\rangle\right)\right),$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

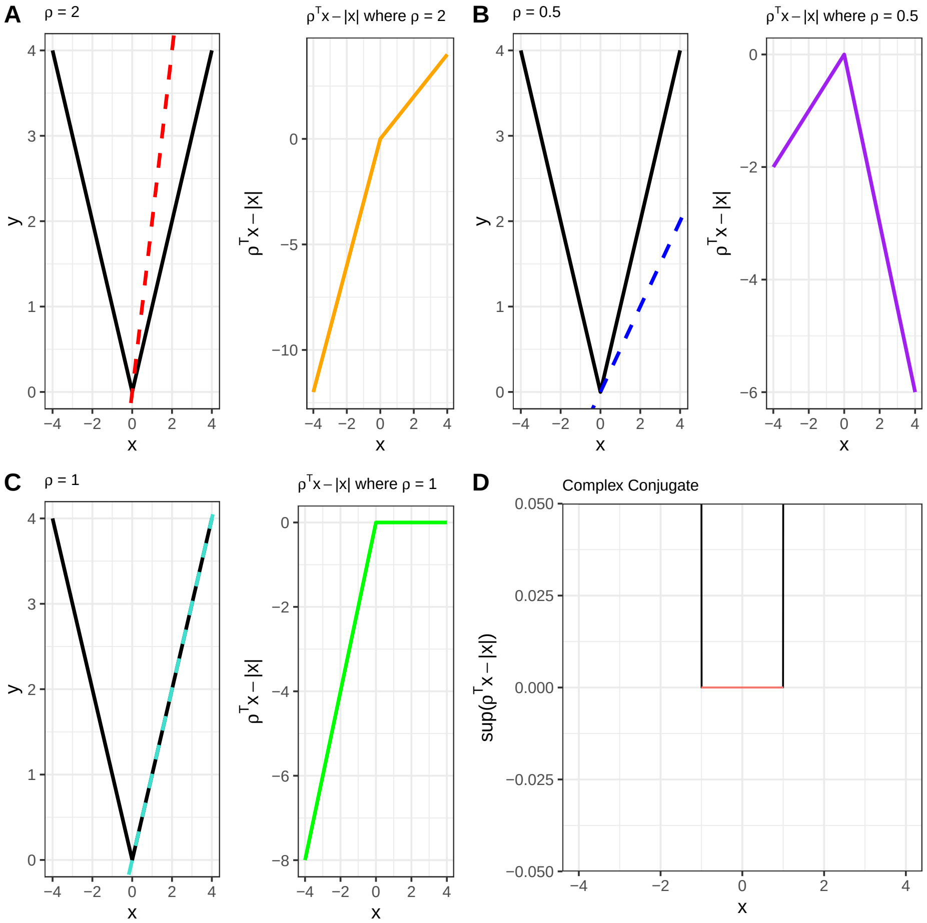

Figure 1: “Visual derivation" of the complex conjugate function is computed. (A)-(C) Shows the properties of $\rho x - f(x)$, where $f(x) = |x|$ for particular values for $\rho = 2, 1, 0.5$. Notice that the difference between ρx and $f(x)$ remains less than infinity only when $|\rho| \leq 1$, while when $\rho > 0$, ρx increases faster than $f(x)$. The final form of the conplex conjugate is displayed in (D).

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

where δ is the event indicator. It can be shown that $\mathcal{L}$ is convex and thus obtaining the regression coefficients that minimize $\mathcal{L}$ can be conducted by using gradient based methods. This framework can be extended to penalized regression by adding a regularization term to $\mathcal{L}$. For example, the lasso solution of regression coefficients corresponds to $\widehat{\beta} = \text{argmin}_{\beta}(\mathcal{L}(\beta) + R(\beta))$, where $R(\beta)$ denotes the regularization terms applied to constrain coefficients, such as the l1/l2 or group lasso penalty terms [10].

2.3. Multiple Kernel Learning

In the traditional Cox regression framework, we assume a linear relationship between our predictors and survival time. However, in reality, the relationship is far more complex. Meanwhile, nonlinear models are often hard to analyze and interpret. Kernel methods are non-parametric methods that utilize reproducing kernel Hilbert space (RKHS) [11, 12], and provide a useful alternative to linear or nonlinear models. Kernel functions map a predictor matrix from $n \times p$ to $n \times n$. This allows us to focus on the similarity between subjects dramatically reducing the complexity of the predictor space to finding a linear relationship between a similarity measure. For instance, using polynomial kernel, we can map a circle boundary problem to a linear boundary problem, which dramatically reduces our computation cost.

The exact relationships between predictions and outcomes are unknown, hence selecting an optimal kernel function presents a challenge. There are no clear rules for selecting a single optimal kernel, but cross-validation is usually implemented[13, 14, 15]. An interesting property of kernels is that a linear combination of two kernel functions results in another kernel function [11]. There have been many works that utilize this fact and have shown that convex combinations of multiple kernels can provide more accurate classifiers than single kernels[13, 16, 17]. Learning the optimal kernel weights is referred to as multiple kernel learning (MKL).

Under the MKL framework, we can denote our target problem as follow

$$\underset{\alpha}{\text{argmin}} \left(L \left(\bar{K} \alpha \right) + \phi_C \left(\alpha \right) \right) \tag{2.2}$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

where $L(\cdot)$ is the loss function of Cox proportional hazards model, $\bar{K} = (K_1, \cdots, K_M)$ are a set of kernels, α is the coefficient matrix for each kernels, and $\phi_C(\cdot)$ is the regularized term for coefficients matrix. In this paper we used elastic net (EN) penalty which can be written as

$$\phi_C(\alpha) = \sum_{m=1}^{M} \phi_C^{(m)}(\alpha_m) = C \sum_{m=1}^{M} \left[(1-\lambda) \|\alpha_m\|_{K_m} + \frac{\lambda}{2} \|\alpha_m\|_{K_m}^2 \right]. \tag{2.3}$$

where $\lambda \in [0,1]$ defines the amount of weight is assigned to the $l2$ and $l1$ regularizer, and C is a multiplier that changes the impact prediction error on the objective function in 2.2.The EN penalty allows us strike a balance between the $l1$ and $l2$ regularization. In other words, it allows for sparse selection that the coefficients for non-informative kernels will be shrunk to zeros, and the coefficients for similar kernels tend to be close, so called group property, which returns us a consistent result [18, 17]. The elastic net penalty is non-smooth, we can use the theory of Moreau's envelope function obtain the approach to solve this problem.

2.4. Moreau Envelope and Elastic Net

The Moreau envelope function allows us to approximate a non-smooth function with a smooth function leading to simpler optimization task. Let $g : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ be a function, for every $\gamma > 0$ we define the Moreau envelope as

$$e_{\gamma g}(x) = \inf_y \left\{ g(y) + \frac{1}{2\gamma} ||x-y||^2 \right\}. \tag{2.4}$$

The Moreau envelope strikes a balance between function approximation and smoothness through the parameter γ. Note that Moreau envelope of $g(y) = -\rho^T y$ is the negative of the convex conjugate of $f(x) = \frac{1}{2\gamma}||x-y||^2$. Additionally, if $e_{\gamma g}$ is smooth derivative given by

$$\nabla e_{\gamma g} = prox\,(y|g) = \operatorname*{argmin}_y \left\{ g(y) + \frac{1}{2\gamma} ||x-y||^2 \right\} \tag{2.5}$$

A simple example of the Moreau envelope is shown in Figure 2A. Note that as γ increases the approximation of the function becomes worse and the shape of the function around $x = 0$ is increasingly rounded.

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

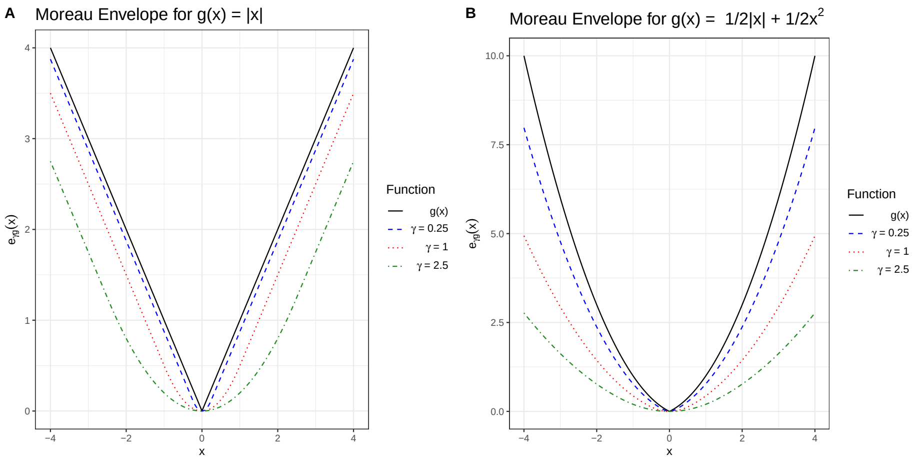

Figure 2: Moreau envelope for lasso regularizer $g(x) = |x|$ for several values of smoothing parameter γ, and elastic net regularizer $g(x) = 1/2|x| + 1/2x^2$. Notice that as the value for γ increases we obtain a function that is more smooth, but a worse approximation of $g(x)$.

Now we apply the concept of the Moreau envelope to the EN. We set $x = \alpha_m$, $y = \alpha'_m$, and $\gamma = 1$ from 2.5 resulting in

$$prox\left(\alpha_m|\phi_C^{(m)}\right) = \underset{\alpha'_m \in R^N}{\operatorname{argmin}}\left(Cg\left(\left\|\alpha'_m\right\|^2_{K_m}\right) + \frac{1}{2}\left\|\alpha'_m - \alpha_m\right\|^2_{K_m}\right), \tag{2.6}$$

where $g\left(x\right) = \left(1-\lambda\right)\sqrt{x} + \frac{\lambda}{2}x$.

Using Cauchy-Schwarz inequality we have

$$\widetilde{g}_C\left(\left\|\alpha'_m\right\|_{K_m}\right) + \frac{1}{2}\left\|\alpha'_m - \alpha_m\right\|^2_{K_m} \geq \widetilde{g}_C\left(\left\|\alpha'_m\right\|_{K_m}\right) + \frac{1}{2}\left(\left\|\alpha'_m\right\|_{K_m} - \left\|\alpha_m\right\|_{K_m}\right)^2.$$

Therefore, we can obtain the minimum solution that

$$\begin{aligned}\left\|\alpha'_m\right\|_{K_m} &= \underset{x \geq 0}{\operatorname{argmin}}\left(\widetilde{g}_C\left(x\right) + \frac{1}{2}\left(x - \left\|\alpha_m\right\|_{K_m}\right)^2\right)\\ &= prox\left(\left\|\alpha_m\right\|_{K_m}|\widetilde{g}_C\right) = \begin{cases} 0 & \left\|\alpha_m\right\|_{K_m} \leq C\left(1-\lambda\right)\\ \frac{\left\|\alpha_m\right\|_{K_m} - C(1-\lambda)}{(C\lambda+1)\left\|\alpha_m\right\|_{K_m}}\alpha_m & else,\end{cases}\end{aligned}$$

where $prox\left(\cdot\right)$ is known as the soft operator, see 2B.

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

3. SpicyMKL algorithm for Cox Proportional Hazard

SpicyMKL was introduced as an efficient implementation of MKL what could learn the best convex combination of potentially 1000 candidate kernels. In order to accomplish this the MKL problem was reformulated using the Moreau envelope and the convex conjugate to ensure that the sum of the loss and regularization penalties is a smooth and convex function [17]. SpicyMKL was introduced for multiple loss function and regularization function, but was not extended to the survival setting.

In our problem, we denote $L^*(\cdot)$ as the conjugate function of the Cox loss function

$$
\begin{aligned}
L^*(-\rho) &= \sup_{z\in R^n}\left((-\rho)^T z - L(z)\right) \\
&= \begin{cases}
h(\rho) & \begin{array}{l} \rho_i < \delta_i,\ i = 1, \dots, n, \text{ and} \\ \sum_{j=i}^{n} \rho_j < 0,\ i = 2, \dots, n, \text{ and} \\ \sum_{j=1}^{i} \rho_j = 0 \end{array} \\
 & \\
+\infty & \text{else,}
\end{cases}
\end{aligned}
$$

where

$$
\begin{aligned}
h(\rho) = \sum_{i=1}^{n} (\delta_i - \rho_i)\log(\delta_i - \rho_i) - \sum_{i=1}^{n-1} \rho_i \log\left(\frac{\prod_{j=i+1}^{n}\left(\delta_j - \sum_{k=j}^{n}\rho_k\right)}{\prod_{j=i+1}^{n}\left(-\sum_{k=j}^{n}\rho_k\right)}\right) \\
- \sum_{i=1}^{n} \delta_i \log\left(\delta_i - \sum_{j=i}^{n}\rho_j\right),
\end{aligned}
$$

and $g_C = \sum_{m=1}^{M} g_C^{(m)}$ is the conjugate function of the soft operator of the regularizer,

$$
g_C^{(m)}\left(\|\rho\|_{K_m}\right) = \begin{cases} 0 & \|\rho\|_{K_m} \le C(1-\lambda) \\ \frac{\left[\|\rho\|_{K_m} - C(1-\lambda)\right]^2}{2C\lambda} & \text{else.} \end{cases}
$$

A full derivation of the results can be found in the supplemental materials. We

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

can see that both $L^*$and δ_C are secondarily differentiable, we can use Newton method to solve above questions.

The conjugate function of L is secondarily differentiable where the first derivative is given by

$$L_i^{*'} = \begin{cases} \log\left(\frac{\prod_{j=i+1}^{n}\left(-\sum_{k=j}^{n}\rho_k\right)}{(\delta_i-\rho_i)\prod_{j=i+1}^{n}\left(\delta_j-\left(\sum_{k=j}^{n}\rho_k\right)\right)}\right) & i<n \\ -\log\left(\delta_n-\rho_n\right) & i=n \end{cases}. \tag{3.1}$$

We can see that if $j < i$, the derivation respected to the i^{th} component does not contain j, so the $L_{ij}^{*'} = 0$. If $j > i$, $L_{ij}^{*'} = L_{ji}^{*'} = 0$. Hence the Hessian matrix of the conjugate function is diagonal matrix, then we can obtain by taking the second derivative,

$$L_{ij}^{*''} = \begin{cases} \frac{1}{\delta_i-\rho_i} & j=i \\ 0 & j\neq i. \end{cases} \tag{3.2}$$

We can calculate gradient and Hessian matrix of $\delta_C^{(m)}$ using (3.1) and (3.2) which are given by

$$g_C^{(m)'}\left(\|\rho\|_{K_m}\right) = \begin{cases} 0 & \|\rho\|_{K_m} \leq C\left(1-\lambda\right) \\ \frac{\|\rho\|_{K_m}-C(1-\lambda)}{C\lambda} & \text{else.} \end{cases}$$

$$g_C^{(m)''}\left(\|\rho\|_{K_m}\right) = \begin{cases} 0 & \|\rho\|_{K_m} \leq C\left(1-\lambda\right) \\ \frac{1}{C\lambda} & \text{else.} \end{cases}$$

We can see that the conjugate function is feasible if $\lambda > 0$, which means we can only use the smooth dual form for elastic net but not block one norm penalty. The block one norm penalty is a kernelized version of group lasso [19, 20].

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

From derivation of the SpicyMKL algorithm, we have

$$\begin{aligned}\nabla_\rho \phi^*\left(\|\rho\|_{K_m}\right) &= \frac{K_m\rho}{\|\rho\|_{K_m}} g_C^{(m)'}\left(\|\rho\|_{K_m}\right)\\ \nabla\nabla_\rho^{'}\phi^*\left(\|\rho\|_{K_m}\right) &= \left(\frac{K_m}{\|\rho\|_{K_m}} - \frac{K_m\rho\rho^{'}K_m}{\|\rho\|_{K_m}^3}\right) g_C^{(m)'}\left(\|\rho\|_{K_m}\right)\\ &\quad + \frac{K_m\rho\rho^{'}K_m}{\|\rho\|_{K_m}^2} g_C^{(m)''}\left(\|\rho\|_{K_m}\right).\end{aligned}$$

Using Newton algorithm we can obtain the optimal $\widehat{\rho}$. As shown in the supplemental material, the step size of Newton update is given by the size that will not make the update of ρ goes beyond the domain of L^*. To satisfy the $\sum_{i=1}^n \rho_i = 0$ constraint for L^*, we added a penalty function $\frac{10^5}{2}\left(\mathbf{1}^{'}\rho\right)^2$. Using Rockafellar[9] (Theorem 31.3), we have

$$\overline{K}\widehat{\alpha} = -\nabla_\rho L^*\left(-\widehat{\rho}\right). \tag{3.3}$$

Under Karush–Kuhn–Tucker (KKT) condition.

$$\begin{aligned}\nabla L^*\left(-\rho\right) &= -\sum_{m=1}^M \nabla_\rho g_C^{(m)}\left(\|\widehat{\rho}\|_{K_m}\right)\\ &= -\sum_{m=1}^M \frac{K_m\widehat{\rho}}{\|\widehat{\rho}\|_{K_m}} g_C^{(m)'}\left(\|\widehat{\rho}\|_{K_m}\right),\end{aligned}$$

and the solution to (3.3) is

$$\Rightarrow \widehat{\alpha}_m = \frac{\widehat{\rho}}{\|\widehat{\rho}\|_{K_m}} g_C^{(m)'}\left(\|\widehat{\rho}\|_{K_m}\right)$$

4. Simulation Data

To evaluate the performance of Multiple Kernel Cox regression (MKCox), we simulate data that are generated with different relationships between the feature and the hazard function. Our simulations were inspired by Katzman [8]. We simplify these simulations we conducted to benchmark and explore the properties of MKCox. We simulate the features from a bivariate normal distribution, $\mathbf{X} = [X_1, X_2]$, with $\mu = 0$, and $\sigma_{X_1} = \sigma_{X_2} = 1$ and a range of

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

values of correlation, and we consider the following hazard function:

$$h(X) = X_1 + 2 * X_2, \text{ and} \tag{4.1}$$

$$h(X) = \log(\lambda) \exp\left(-\frac{(X_1^2 + X_2^2)}{2 * r^2}\right) \tag{4.2}$$

where $\lambda = 5$, and $r = 1/2$. Then the survival times are generated by:

$$T = -\frac{\log(u)}{\exp(h(X))}, \text{ where } u \sim U(0,1),$$

a censoring time is established, so that approximately 50% of patients have observed an event. We used two kernels in for our illustration K_1 is the radial basis function with hyperparameter $\sigma = 2$, and K_2 is a linear kernel. We compare MKCox to the following methods random survival forest ([21]) (RF) implemented in *randomForestSRC* (2.9.0), and stochastic gradient boosted ([22]) Cox regression (GBM) implemented in *gbm* (2.1.5). All our analyses were performed in R (3.6.0).

We aim to obtain a good estimate of the hazard function, as well as, maintain a good prediction of survival time. In Figures 3 and 4, we see that all methods can capture the structure of the hazard function when the underlying relationship is linear. Additionally, MKCox can recover the underlying patterns in the hazard function better than Cox regression and GBM, while both RSF and MKCox both capture the nonlinear pattern for accurately. To evaluate the performance of MKCox will compute popular metrics for survival models such as the concordance index. These results are shown in Table 1. Notice in the linear case that all methods provide similar concordance indices, while in the nonlinear case Cox performs substantially worse and MKCox slightly better than other machine learning methods. These simulations illustrate that MKCox can produce similar or better results under different underlying relationships between the hazard ratio and the features.

5. Case Study

The Cancer Genome Atlas (TCGA) project is a large initiative to study the multiomics effect of gene expression RNAseq and stem loop expression on

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

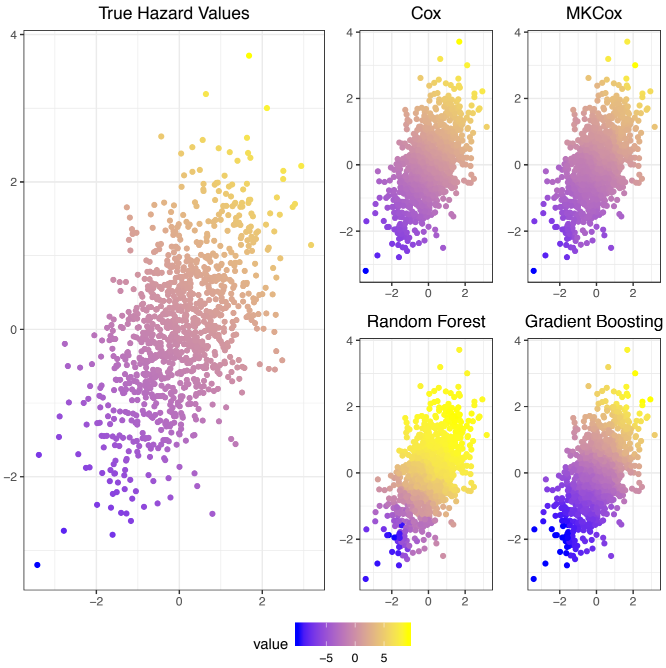

Figure 3: ((A) The values of the linear hazard function used in (4.1), $h(x_1, x_2) = x_1 + 2x_2$. (B) Predicted hazard values by Cox proportional hazard, and the three machine learning techniques.

160 patients' survival time[23, 24]. We downloaded the data from Genomic Data Commons (GDC) Data Portal. The latest survival data were downloaded using the *TCGAbiolinks* ([25]) package in R. The gene expression and miRNA expression data were downloaded from University of California at Santa Cruz (UCSC) Xena ([26]) (`https://xena.ucsc.edu/`) database. For gene expression, we used 165 the fragments per kilobase of transcript per million mapped reads upper quartile (FPKM-UQ) with $\log_2(x+1)$ transformation on mRNA via high-throughput sequencing (HTseq) ([27]) technique with gencode v22, while for stem loop expression, we used the per million mapped reads (RPM) with $\log_2(x+1)$ transformation via miRNA expression quantification technique aligned to GRCh38. 170 In total, we have 235 dead and 215 alive (450 in total) patients.

Since we have two features sources (gene expression and stem loop expres-

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

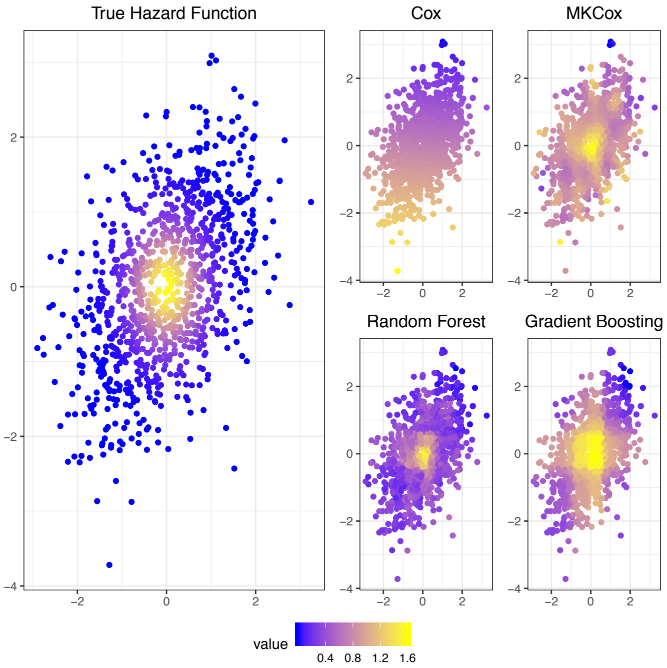

Figure 4: IA) The values of the nonlinear hazard function used in (4.1), $h(x_1, x_2) = 5 * \exp((x_1^2 + x_2^2)/2)$. (B) Predicted hazard values by Cox proportional hazard, and the three machine learning techniques.

sion), we kernelized the two feature sources separately using pathway kernels with $K_i = X_i L_i^{-1} X_i$ where K_i, X_i and L_i are the kernel matrix, feature matrix and standardized Laplacian matrix for each feature source, respectively. So our model can be written as

$$l\left(\text{survival time}, \sum_{i=1}^{2} K_i \alpha_i\right) + C\left((1-\lambda)\sum_{i=1}^{2} \|\alpha_i\|_{K_i} + \frac{\lambda}{2}\sum_{i=1}^{2} \|\alpha_i\|_{K_i}^2\right)$$

which is equivalent to a linear grouped network regularized model

$$l\left(\text{survival time}, \sum_{i=1}^{2} X_i \beta_i\right) + C^{'}\left(\left(1-\lambda^{'}\right)\sum_{i=1}^{2} \|\beta_i\|_{L_i} + \frac{\lambda^{'}}{2}\sum_{i=1}^{2} \|\beta_i\|_{L_i}^2\right)$$

where $\beta_i = L_i^{-1}\alpha_i$ so that we can obtain the coefficient of feature i using this transformation. In this case the kernel learning method has strong interpretability. The Laplacian matrices were estimated empirically by neighbor network and coexpression network method proposed by [28].

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Dataset	Cox	MKL	RF	GBM
Simulated Linear Data	0.886	0.884	0.878	**0.887**
Simulated Non-linear Data	0.511	**0.653**	0.599	0.634
SKCM		**0.640**	0.606	0.536

Table 1: Concordance Index for Each Model

To evaluate the performance we split that data into 301 training and 149 test samples, stratified by survival status (dead versus alive). The models we compared were all trained on training data and the results were obtained on test data. The parameters for our MKL model and GBM models were tuned by 5-fold cross-validation. From Table 1 we can see that our proposed multiple kernel learning using network kernels worked the best. Though it was a linear model, it achieved a higher concordance index than nonlinear tree-based models like the random forest or stochastic gradient boosting machine. Due to the flexibility and efficiency of MKCox can incorporate many pathways under different kernel representations.

6. Conclusion

In this paper, we derived an efficient multiple kernel learning algorithm for survival prediction models and the convex conjugate function for Cox proportional hazards loss function. A challenge of deriving efficient algorithms for proportional hazard models is that the Hessian is not a diagonal matrix. However, through the convex conjugate function we can utilize the diagonal property to achieve a time competitive algorithm. Therefore, the Cox proportional hazards loss function can be more easily implemented than other machine learning methods.

Both the simulation and case study in our paper showed a robust performance of our proposed method in likelihood function estimation and out of data prediction, even compared to tree-based methods which often showed a strong predictive power. As we can see that MKL method was shown to have superior

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

performance in cancer genomic studies ([29]). Future studies include extending the model to other more complex survival problems including competing risks.

7. Acknowledgements

This work was supported in part by Institutional Research Grant number 14-189-19 from the American Cancer Society, and Moffitt Foundation Methodological Development Pilot Grant.

[1] D. R. Cox, Models and life-tables regression, JR Stat. Soc. Ser. B 34 (1972) 187–220.

[2] M. Newby, Accelerated failure time models for reliability data analysis, Reliability Engineering & System Safety 20 (3) (1988) 187 – 197.

[3] B. Efron, The efficiency of cox's likelihood function for censored data, Journal of the American statistical Association 72 (359) (1977) 557–565.

[4] H. LI, Y. LUAN, Kernel cox regression models for linking gene expression profiles to censored survival data, in: Pacific Symposium on Biocomputing 2003: Kauai, Hawaii, 3-7 January 2003, World Scientific, 2002, p. 65.

[5] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for cox's proportional hazards model via coordinate descent, Journal of statistical software 39 (5) (2011) 1.

[6] G. Ridgeway, The state of boosting, Computing Science and Statistics (1999) 172–181.

[7] H. Li, Y. Luan, Boosting proportional hazards models using smoothing splines, with applications to high-dimensional microarray data, Bioinformatics 21 (10) (2005) 2403–2409.

[8] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, Y. Kluger, Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network, BMC medical research methodology 18 (1) (2018) 24.

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[9] R. T. Rockafellar, Convex analysis, Princeton Mathematical Series, Princeton University Press, Princeton, N. J., 1970.

[10] R. TIBSHIRANI, The lasso method for variable selection in the cox model, Statistics in Medicine 16 (4) (1997) 385–395. doi:10.1002/(SICI)1097-0258(19970228)16:4<385::AID-SIM380>3.0.CO;2-3.

[11] N. Aronszajn, Theory of reproducing kernels, Transactions of the American Mathematical Society 68 (3) (1950) 337–404.
URL http://dx.doi.org/10.2307/1990404

[12] B. Scholkopf, A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2001.

[13] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, Simplemkl, Journal of Machine Learning Research 9 (2008) 2491–2521.

[14] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, W. Xu, Applications of support vector machine (svm) learning in cancer genomics, Cancer genomics & proteomics 15 (1) (2018) 41–51. doi:10.21873/cgp.20063.
URL https://pubmed.ncbi.nlm.nih.gov/29275361

[15] R. Gholami, N. Fakhari, P. Samui, S. Sekhar, V. E. Balas, Chapter 27 - Support Vector Machine: Principles, Parameters, and Applications, Academic Press, 2017, pp. 515–535. doi:https://doi.org/10.1016/B978-0-12-811318-9.00027-2.
URL http://www.sciencedirect.com/science/article/pii/B9780128113189000272

[16] Z. Xu, R. Jin, H. Yang, I. King, M. R. Lyu, Simple and efficient multiple kernel learning by group lasso, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, USA, 2010, pp. 1175–1182.
URL http://dl.acm.org/citation.cfm?id=3104322.3104471

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

[17] T. Suzuki, R. Tomioka, Spicymkl: a fast algorithm for multiple kernel learning with thousands of kernels, Machine Learning 85 (1) (2011) 77–108. doi:10.1007/s10994-011-5252-9.
URL https://doi.org/10.1007/s10994-011-5252-9

[18] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (2) (2005) 301–320. arXiv:https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2005.00503.x, doi:10.1111/j.1467-9868.2005.00503.x.
URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00503.x

[19] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society Series B 68 (2006) 49–67. doi:10.1111/j.1467-9868.2005.00532.x.

[20] F. R. Bach, Consistency of the group lasso and multiple kernel learning, J. Mach. Learn. Res. 9 (2008) 1179–1225.
URL http://dl.acm.org/citation.cfm?id=1390681.1390721

[21] H. Ishwaran, M. Lu, Random survival forests, Wiley StatsRef: Statistics Reference Online (2008) 1–13.

[22] J. H. Friedman, Stochastic gradient boosting, Computational statistics & data analysis 38 (4) (2002) 367–378.

[23] K. Tomczak, P. Czerwińska, M. Wiznerowicz, The cancer genome atlas (tcga): an immeasurable source of knowledge, Contemporary oncology (Poznan, Poland) 19 (1A) (2015) A68–A77. doi:10.5114/wo.2014.47136.
URL https://pubmed.ncbi.nlm.nih.gov/25691825

[24] J. Guan, R. Gupta, F. V. Filipp, Cancer systems biology of tcga skcm: Efficient detection of genomic drivers in melanoma, Scientific Reports 5 (1)

bioRxiv preprint doi: https://doi.org/10.1101/2020.05.04.077263; this version posted May 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

(2015) 7857. doi:10.1038/srep07857.
URL https://doi.org/10.1038/srep07857

[25] A. Colaprico, T. C. Silva, C. Olsen, L. Garofano, C. Cava, D. Garolini, T. S. Sabedot, T. M. Malta, S. M. Pagnotta, I. Castiglioni, et al., Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data, Nucleic acids research 44 (8) (2015) e71–e71.

[26] M. Goldman, B. Craft, M. Hastie, K. Repecka, A. Kamath, F. McDade, D. Rogers, A. Brooks, J. Zhu, D. Haussler, The ucsc xena platform for public and private cancer genomics data visualization and interpretation. biorxiv, 326470.

[27] S. Anders, P. T. Pyl, W. Huber, Htseq-a python framework to work with high-throughput sequencing data, Bioinformatics 31 (2) (2015) 166–169.

[28] K. Li, X. Wang, P. F. Kuan, Mixture network regularized generalized linear model with feature selection, bioRxiv (2019) 678029.

[29] J. C. Costello, L. M. Heiser, E. Georgii, M. Gönen, M. P. Menden, N. J. Wang, M. Bansal, P. Hintsanen, S. A. Khan, J.-P. Mpindi, et al., A community effort to assess and improve drug sensitivity prediction algorithms, Nature biotechnology 32 (12) (2014) 1202.