Survival Analysis on Rare Events Using Group-Regularized Multi-Response Cox Regression

Ruilin Li*[1], Yosuke Tanigawa[2], Johanne M. Justesen[2], Jonathan Taylor[3], Trevor Hastie[2,3],
Robert Tibshirani†[2,3] and Manuel A. Rivas‡[2]

[1]Institute for Computational and Mathematical Engineering, Stanford University
[2]Department of Biomedical Data Science, Stanford University
[3]Department of Statistics, Stanford University

Abstract

We propose a Sparse-Group regularized Cox regression method to improve the prediction performance of large-scale and high-dimensional survival data with few observed events. Our approach is applicable when there is one or more other survival responses that 1. has a large number of observed events; 2. share a common set of associated predictors with the rare event response. This scenario is common in the UK Biobank (Sudlow et al. 2015) dataset where records for a large number of common and rare diseases of the same set of individuals are available. By analyzing these responses together, we hope to achieve higher prediction performance than when they are analyzed individually. To make this approach practical for large-scale data, we developed an accelerated proximal gradient optimization algorithm as well as a screening procedure inspired by Qian et al. (2019). We provide a software implementation of the proposed method and demonstrate its efficacy through simulations and applications to UK Biobank data.

Cox proportional hazard model; Sparse-Group Lasso; Multi-response regression.

1 Introduction

1.1 Cox Proportional Hazard Model

Cox model (Cox 1972) provides a flexible mathematical framework that describes the relationship between the predictors and a time-to-event response. For each individual we observe a triple $\{O, X, T\}$, where $X \in \mathbb{R}^d$ are the features and $O \in \{0, 1\}$ is a status indicator. If $O = 1$, then T is the actual time-to-event. If $O = 0$, then we only know that the time-to-event is at least T. The hazard function according to the Cox model can be written as

$$h(t|X) = h_0(t)\exp(X^T\beta),$$

where $\beta \in \mathbb{R}^d$ is the coefficients vector that measures the strength of association between X and the response. This hazard function is equivalent to the cumulative distribution function:

$$P(T \leq t|X) = 1 - \exp\left(-\int_0^t h_0(s)e^{\beta^T X}ds\right).$$

*Corresponding author: ruilinli@stanford.edu
†Corresponding author: tibs@stanford.edu
‡Corresponding author: mrivas@stanford.edu

Here $h_0 : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is the baseline hazard function. In our applications we are interested in the relationship between the features and the responses, so the baseline hazard function is a nuisance variable. We can estimate the parameters β directly without knowing the baseline hazard by maximizing the log partial likelihood function (Cox 1972):

$$\begin{aligned} l(\beta|Data) &= \log\left[\prod_{i:O_i=1} \frac{\exp(X_i^T\beta_k)}{\sum_{j:T_j\geq T_i}\exp(X_j^T\beta)}\right] \\ &= \sum_{i=1}^n O_i\left[X_i^T\beta - \log\left(\sum_{j:T_j\geq T_i}\exp(X_j^T\beta)\right)\right] \end{aligned}$$

When the number of observed events is small relative to n, estimating β becomes challenging. This could happen, for example, when the time-to-event response is the age of diagnosis of a rare disease. In particular, if O_i are i.i.d Bernoulli random variables with probability p, then the information matrix is proportional to p and thus the asymptotic variance of the maximum partial likelihood estimate is inversely proportional to p.

We evaluate a fitted survival model using the concordance index, or the C-index. For a parameter estimate $\hat{\beta}$, the C-index is defined as

$$C(\hat{\beta}) = \frac{\sum_{i=1}^n O_i[|\{j : \hat{\beta}^T X_i > \hat{\beta}^T X_j\}| + |\{j : \hat{\beta}^T X_i = \hat{\beta}^T X_j\}|/2]}{\sum_{i=1}^n O_i|\{j : T_j > T_i\}|}. \tag{1}$$

For more details on C-index, see Harrell et al. (1982), Li et al. (2020).

1.2 Sparse-Group Lasso

The Lasso method (Tibshirani 1996) makes the assumption that only a small subset of predictors are associated with the response. In other words, it assumes that β has only a small number of non-zero entries. A sparse solution can be obtained by optimizing an L_1-regularized objective function.

Sparse-Group Lasso (Simon et al. 2013) assumes not only that many individual elements of β are 0, but also that many groups of variables have coefficients 0 simultaneously. For example, in a single-response Cox model with d-dimensional features, if groups of variables $\mathcal{G} = \{g : g \subseteq [d]\}$ are believed to have sparse-group structure, then the sparse-group Lasso minimizes the following objective function:

$$-\sum_{i=1}^n O_i\left[X_i^T\beta - \log\left(\sum_{j:T_j\geq T_i}\exp(X_j^T\beta)\right)\right] + \lambda\|\beta\|_1 + \sum_{g\in\mathcal{G}}\lambda_g\|\beta_g\|_2. \tag{2}$$

2 Methods

2.1 Preliminaries

In this section we define the notations and the key model assumptions that we will use in the subsequent sections. For an integer n, define $[n] = \{1, 2, \cdots, n\}$ and define $x_+ = \max\{x, 0\}$ for all $x \in \mathbb{R}$.

We analyze $K \geq 1$ time-to-event responses on n individuals. For example, the responses could be time from birth to K different diseases. The data we observed are in the format:

$$\mathcal{D} = \{X_i, T_i^1, \cdots, T_i^K, O_i^1, \cdots, O_i^K\}_{i=1}^n.$$

Here $X_i \in \mathbb{R}^d$ are ith individual's features. Denote the full features matrix $\boldsymbol{X} = [X_1, X_2, \cdots, X_n]^T \in \mathbb{R}^{n\times d}$. For $k = 1, \cdots, K$, $O_i^k = 1$ implies that T_i^k is the true time until the event k and for the ith individual, and $O_i^k = 0$ implies that the true time until the event k is right-censored at T_i^k. We assume each response follows a Cox proportional hazard model:

$$h_k(t|X) = h_{0,k}(t)\exp(X^T\beta_k). \tag{3}$$

where $h_{0,k} : \mathbb{R}^+ \mapsto \mathbb{R}^+$ is the baseline hazard function of the kth response. Let

$$n_k = \sum_{i=1}^n O_i^k$$

be the number of observed event k.

We make the assumption that not only β_k is sparse for all $k \in [K]$, but they also have a small common support. That is $\{j \in [d] : \beta_k^j \neq 0, k \in [K]\}$ is a small set relative to d. In human genetics applications, the first assumption means that each response is associated with a small set of genetic variants, and the second assumption implies that there are significant overlap among the genetic variants that are associated with each response. This belief is the main driver for prediction performance improvements on rare diseases, and it translates to the following regularized partial likelihood objective function:

$$\min_{\beta_1,\cdots,\beta_K} \sum_{k=1}^K \frac{1}{n_k}\left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log\left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j)\right)\right] + \lambda\left(\sum_{j=1}^d \|\beta^j\|_1 + \alpha\|\beta^j\|_2\right). \tag{4}$$

Here the first term is the sum of the K negative log-partial-likelihood, normalized by the number of observed events. The second term is the regularization. $\|\beta^j\|_1$, $\|\beta^j\|_2$ are the 1-norm and 2-norm of the coefficients for the jth variable. That is, if we put all the parameters into a matrix $B = [\beta_1, \beta_2, \cdots, \beta_K] \in \mathbb{R}^{d\times K}$, then β^j is the jth row of B and β_k is the kth column. Note that when $\alpha = 0$, the objective function decouples for each β_k and they can be optimized separately. In our implementation we solve a slightly more general problem:

$$\min_{\beta_1,\cdots,\beta_K} \left\{\sum_{k=1}^K \frac{1}{n_k}\left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log\left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j)\right)\right] + \lambda\left[\sum_{j=1}^d w_j(\|\beta^j\|_1 + \alpha\|\beta^j\|_2)\right]\right\},$$

where $\{w_j\}_{j=1}^d$ are user provided penalty factor for each variables, which may be useful in the setting where protein-truncating or protein-altering variants should be prioritized (Rivas et al. 2015, DeBoever et al. 2018). Just like in Yuan & Lin (2006), our implementation by default fixes α at $\sqrt{K}$. The solution is computed on a pre-defined sequence of λs: $\lambda_1 > \lambda_2 > \cdots > \lambda_L$, where λ_1 is chosen so that the solution just become non-zero.

To simplify the notation, for $j \in [d]$ denote

$$g_j = g_j(\beta_1, \cdots, \beta_k) := \frac{\partial}{\partial \beta^j} \sum_{k=1}^K \frac{1}{n_k}\left[\sum_{i:O_i^k=1} -\beta_k^T X_i + \log\left(\sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j)\right)\right]. \tag{5}$$

Here $g_j \in \mathbb{R}^K$ is the partial derivative of the smooth part of (4) with respect to the coefficients of the variable j. Finally, let $S_1(\cdot;\lambda) : \mathbb{R}^K \mapsto \mathbb{R}^K$ be the element-wise soft-thresholding function:

$$S_1(v;\lambda)_k = sign(v_k)(|v_k| - \lambda)_+ \text{ for all } k \in [K].$$

2.2 Optimization Method

We use a Nesterov-accelerated (Nesterov 1983) proximal gradient method (Daubechies et al. 2004, Beck & Teboulle 2009) to optimize the objective function (4). Proximal gradient algorithm is particularly suitable when the objective function is the sum of a smooth function and a simple function. In our case the smooth function is the sum of the negative log-partial-likelihood functions, and the simple function is the regularization term. The algorithm alternates between two steps until convergence criteria is met:

1. A gradient step that decreases the smooth part of the objective:

$$\beta^j \leftarrow \beta^j - tg_j,$$

   where t is the step size that we determine using backtracking line search.

2. A proximal step that keeps the regularization term small:

$$\beta^j \leftarrow prox_t(\beta^j).$$

   Here the proximal operator $prox_t : \mathbb{R}^K \mapsto \mathbb{R}^K$ is defined as

$$prox_t(x) := \arg\min_z \frac{1}{2t}\|z - x\|_2^2 + \lambda\|z\|_1 + \lambda\alpha\|z\|_2.$$

   To simplify the notation we omit the dependency of $prox_t(x)$ on λ, α. Simple calculation shows that

$$prox_t(x) = S_2(S_1(x; t\lambda); t\alpha\lambda).$$

   where $S_2(\cdot; t\alpha\lambda) : \mathbb{R}^K \mapsto \mathbb{R}^K$ is a group soft-thresholding operator:

$$S_2(v; t\alpha\lambda) = \begin{cases} 0 & \text{if } \|v\|_2 \le t\alpha\lambda \\ \frac{v}{\|v\|_2}(\|v\|_2 - t\alpha\lambda) & \text{otherwise} \end{cases}.$$

We describe the optimization algorithm in pseudocode, including details about Nesterov acceleration and backtracking line search in algorithm 1.

2.3 Variable Screening for Lasso Path

In many of our applications the data is large-scale and high-dimensional. For example, the UK Biobank dataset (Sudlow et al. 2015) contains millions of genetic variants and over $500,000$ participants. Reading the feature matrix of the UK Biobank dataset into R requires more than 4 terabytes of memory, which is much larger than the RAM size of most computers. While memory mapping (Kane et al. 2013) allows users to access out-of-memory data with ease, it requires lots of disk Input/Output operations, which is much slower than in-memory operation. This becomes even more problematic for iterative optimization algorithms that use the entire feature matrix every iteration.

To reduce the frequency of reading the full data matrix, here we derive a version of variable screening method following similar ideas of the strong rule (Tibshirani et al. 2012) and the Batch Screening Iterative Lasso (Qian et al. 2019).

For each $\alpha > 0, v \in \mathbb{R}^K$, define the α^* norm of v to be

$$\|v\|_{\alpha^*} := \sup\{u^T v : u \in \mathbb{R}^K, \|u\|_1 + \alpha\|u\|_2 \le 1\}. \tag{6}$$

We use the following results to get an optimality condition for the solutions $\beta_1, \cdots, \beta_K$. The proofs are given in section 6.

Algorithm 1: Proximal Gradient Method for (4)

```
Set line search parameter $\eta > 1$;
Denote the parameter matrix $B \in \mathbb{R}^{d \times K}$. Initialize $B^{(0)}$;
Write the smooth part of the objective as:
    $$f(B) = \sum_{k=1}^{K} \frac{1}{n_k} \left[ \sum_{i:O_i^k=1} -\beta_k^T X_i + \log \left( \sum_{j:T_j^k \geq T_i^k} \exp(\beta_k^T X_j) \right) \right].$$
Set iteration count $i = 0$; Set initial step-size $t = 1$; Set Nesterov weights $w_0, w_1 = 1$;
while $B$ has not converged do
    Nesterov acceleration: $w_1 \leftarrow (1 + \sqrt{1 + 4w_0^2})/2$;
    $B^{(i+0.5)} \leftarrow B^{(i)} + (w_0 - 1)(B^{(i)} - B^{(i-1)})/w_1$; $w_0 \leftarrow w_1$;
    Compute the gradient $g_j = \frac{\partial}{\partial \beta^j} f(B^{(i+0.5)})$ for $j \in [d]$;
    Start backtracking line search:
    repeat
        $B \leftarrow B^{(i+0.5)} - t\nabla f(B^{(i+0.5)})$;
        Denote $\beta^j$ the $j$th row of $B$ for $j \in [d]$;
        Apply proximal step: $\beta^j \leftarrow prox_t(\beta^j)$ for $j \in [d]$;
        if $f(B) \leq f(B^{(i+0.5)}) + \langle B - B^{(i+0.5)}, \nabla f(B^{(i+0.5)}) \rangle + \|B - B^{(i+0.5)}\|_2^2/(2t)$ then
            break;
        end
        $t \leftarrow t/\eta$;
    until the break condition above is satisfied;
    $B^{(i+1)} \leftarrow B$; $i \leftarrow i + 1$;
    Check convergence based on objective value change or parameter change.
end
return $B^{(i)}$
```

Proposition 1. *For any $\lambda > 0$, the gradients defined at* (5) *at the optimal solution to* (4) *satisfies:*

$$\|g_j\|_{\alpha^*} \begin{cases} \leq \lambda & \text{if the optimal } \beta^j = 0 \\ = \lambda & \text{if the optimal } \beta^j \neq 0 \end{cases} \text{for all } j \in [d]. \tag{7}$$

This result motivates us to first fit a model (solving (4)) using a small number of variables whose gradient has the largest α^* norm, assuming the coefficients for the rest of the variables are all zero. Then to verify the validity of the assumption we check that $\|g_j\|_{\alpha^*} \leq \lambda$ for variables assumed to have zero coefficients. We refer to this step as KKT checking. Note that based on its definition (6), it's not clear how we can compute $\|v\|_{\alpha^*}$. Here we give a more explicit characterization.

Proposition 2. $\|v\|_{\alpha^*} \leq \lambda$ *if and only if* $\|S_1(v;\lambda)\|_2 \leq \alpha\lambda$.

Using the above we can check if $\|v\|_{\alpha*} \leq \lambda$ quite easily and compute $\|v\|_{\alpha*}$ using binary search in $\mathcal{O}(K \log \|v\|_\infty)$ to any fixed precision (since $\|v\|_{\alpha^*} \leq \|v\|_\infty$).

Now we are ready to state the overall structure of our algorithm with variable screening. Suppose valid solutions for $\lambda_1, \cdots, \lambda_l$ have been obtained. Next we follow these steps:

1. (**Screening**) In the last iteration, we cache the full gradient $\{g^j\}_{j=1}^d$ evaluated at the solution at λ_l. In the fitting step we include two types of variables:
   - We include the ever-active variables $\mathcal{A} := \{j \in [d] : \hat{\beta}^j \neq 0$ for any previously obtained $\hat{\beta}\}$.
   - Top M variables with the largest $\|g_j\|_{\alpha*}$ that are also not ever-active.

   We denote the set of variables used to fit (4) as the strong set $\mathcal{S} \subseteq [d]$.
2. (**Fitting**) In this step, we solve the problem (4) for the next few λs using only variables in $\mathcal{S}$, assuming $\beta^j = 0$ for all $j \notin \mathcal{S}$. This is done using proximal gradient descent (algorithm 1). To speed-up the computation we initialize the variables at the previous valid solution (warm start).
3. (**KKT Checking**) After obtaining the solution from the fitting step. We compute the full gradient. This is the only step we will need the full data matrix $\boldsymbol{X}$. We check if the KKT conditions (7) are satisfied for all variables then go back to the screening step at the first λ value where the KKT condition fails. We also cache the full gradient at the last valid solutions for the screening step.
4. (**Early Stopping**) We keep a separate validation set to evaluate the current estimated parameters. We choose the optimal λ as the one that gives the highest validation C-index. The optimal λ might be different for different responses. In this paper we focus only on the prediction accuracy of one rare event, so it is reasonable to stop when the validation C-index of this response starts to decrease, regardless if the optimal λ for other responses has been reached.

These steps are described in algorithm 2.

2.4 Software

We implemented the proximal gradient method in section 2.1 as an R package, available at `https://github.com/rivas-lab/multisnpnet-Cox`. In this package we also implement the screening procedure described in this section for genetics data in Plink2 format (Chang et al. 2014).

Algorithm 2: Iterative Screening for Lasso Path

```
Initialize ever active set $\mathcal{A}^{(0)} = \emptyset$ and construct the regularization parameters $\lambda_1, \cdots, \lambda_L$;
Initialize a short list of initial regularization parameters $\Lambda^{(0)} = \{\lambda_1, \cdots, \lambda_{L^{(0)}}\}$;
Initialize the parameters $\hat{\beta}_1^{(0)}, \cdots, \hat{\beta}_K^{(0)} = 0$; Set the iteration counter $i = 0$;
while $\hat{\beta}(\lambda_L)$ not computed do
    Set $v$ to be the largest number such that the solutions for $\lambda_1, \cdots, \lambda_v$ are obtained;
    Screening:
    Compute (or use the cached) full gradient (5) $g_1, \cdots, g_d$ at the last solutions.
    Set $\mathcal{E}_M^{(i)}$ to be the $M$ variables in $[d] \setminus \mathcal{A}^{(i)}$ with the largest $\|g_j\|_{\alpha*}$;
    Set $\mathcal{S}^{(i)} = \mathcal{A}^{(i)} \cup \mathcal{E}_M^{(i)}$. This will be the variables used in the fitting step;
    Fitting:
    for $\lambda \in \Lambda^{(i)}$ do
        Obtain parameter estimates by solving (4) using only variables in $\mathcal{S}^{(i)}$;
        The optimization algorithm is described in algorithm 2;
        Coefficients for variables not in $\mathcal{S}^{(i)}$ are set to be zero.;
    end
    Checking:
    Compute the full gradients $g_1(\lambda), \cdots, g_d(\lambda)$ at solutions obtained with regularization
     parameters $\lambda \in \Lambda^{(i)}$;
    Find the smallest $\lambda \in \Lambda^{(k)}$ such that the KKT condition (7) is satisfied:

        $$\overline{\lambda}^{(i)} = \min\left\{\lambda \in \Lambda^{(i)} : \max_{j \in [d] \setminus \mathcal{S}^{(i)}} \|g_j(\lambda)\|_{\alpha*} < \lambda\right\}$$

    Update ever active set $\mathcal{A}^{(i+1)} = \mathcal{A}^{(i)} \cup \{j : \hat{\beta}^j(\overline{\lambda}^{(i)}) \neq 0\}$ ;
    Update $\Lambda^{(i+1)} = \{\lambda \in \Lambda^{(i)} : \lambda < \overline{\lambda}^{(i)}\}$. Extend $\Lambda^{(i+1)}$ if it is too short.;
    For $\lambda \in \{\lambda \in \Lambda^{(i)} : \lambda \geq \overline{\lambda}^{(i)}\}$, we obtain valid solutions; Set $i = i + 1$;
end
return $\hat{\beta}(\lambda_1), \cdots, \hat{\beta}(\lambda_L)$
```

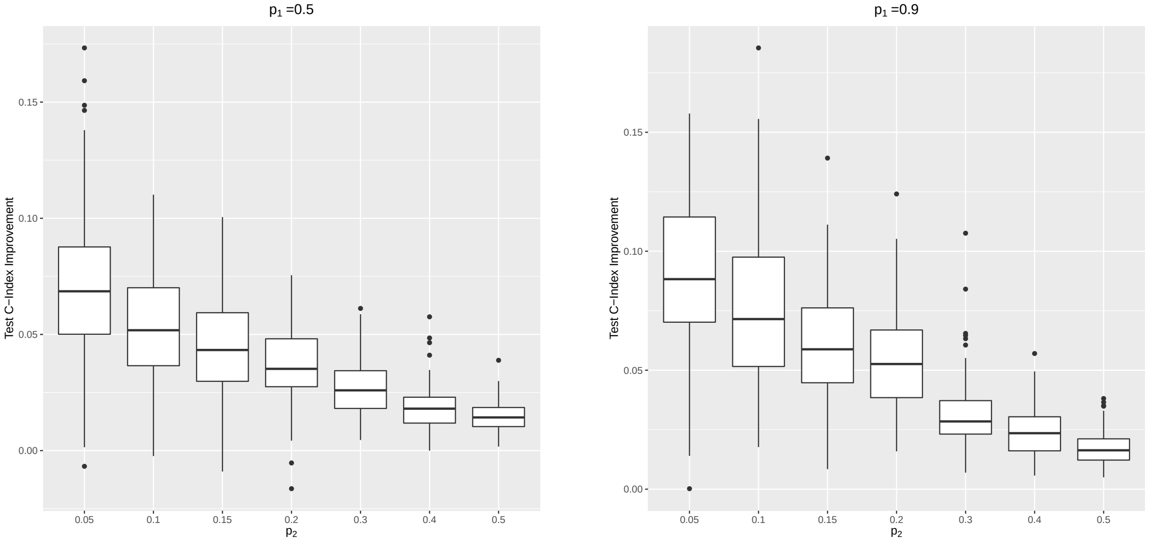

Figure 1: Absolute improvements in test C-index of the rare event response when two responses are trained together. p_1 is the proportion of uncensored events of the first response. p_2 (rare) is the proportion of uncensored events of the second response. The true coefficients have exactly the same support.

The major computational bottleneck in the proximal gradient method are matrix-matrix multiplications. These operations have high arithmetic intensity and are particularly suitable for GPU acceleration. Therefore we also provide a GPU implementation of algorithm 1 on CUDA-enabled device, available at `https://github.com/rivas-lab/multisnpnet-Cox_gpu`. With $n = d = 10000, K = 20$, the GPU implementation achieves almost 10× speedup in solving a path of 50 λs (9.7 seconds vs 92 seconds) on a Tesla V100 GPU than the CPU implementation on an Intel Xeon 6528R processor with 28 threads, accelerated using Intel's Math Kernel Library.

3 Simulations

In this section we compare the performance of the proposed approach against a simple Lasso, where multiple responses are fitted independently. Here we simulate two responses ($K = 2$), $n = 400, d = 5000$, the entries of the predictor matrix are i.i.d random signs $\{-1, 1\}$ with probability 0.5 each, and the time-to-event responses are exponential distributed with rate $\exp(X_i^T \beta)$, which satisfies proportional hazard. The parameters β_1, β_2 have a common support of size 35. We use a large (5000 samples) and uncensored validation set to select the optimal λ. α is fixed at $\sqrt{2}$. The parameter estimate corresponding to the best λ is then evaluated at a large, uncensored test set. We use the C-index as both the validation and test metric. The censoring for both responses are randomly chosen and independent from everything else. The simulations are done for multiple combinations of censoring proportions, each repeated 100 times. Here we report the improvement in C-index of the response with rare events. See Figure 1.

The assumption that the coefficients for different responses have the same support can come from domain knowledge (such as biology). In practice this is usually not exactly satisfied. Here we use simulation to examine the robustness of our approach. The setup is the same as the previous simulations, except now the overlap of the support might be smaller than 35 (the number of non-zero entries of β_1, β_2). See the left panel of Figure 2.

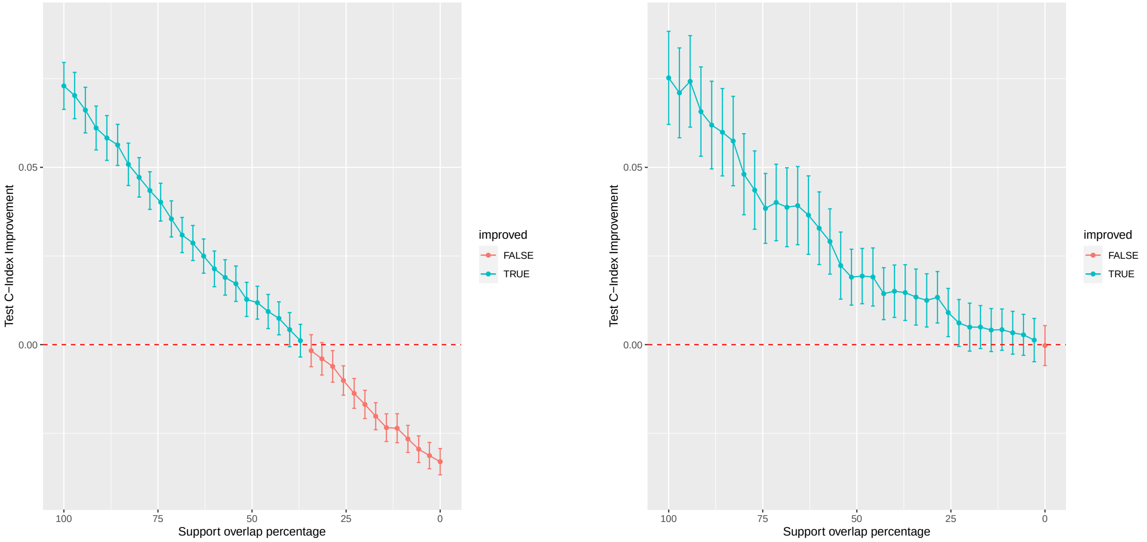

Figure 2: Absolute improvements in Test C-index of the rare event response when two responses are trained together. 60% of the events for the first response are uncensored, and 5% of the events for the second are uncensored. The horizontal axis is the overlap proportion of the support of β_1 and β_2, in other words $|\{i \in [p] : \beta_1^i \neq 0, \beta_2^i \neq 0\}| \cdot 100/35$. The left panel shows the result when α is fixed at $\sqrt{2}$, and λ is selected from a large uncensored validation set. The right panel shows the result when α is selected between $\{0, \sqrt{2}\}$ using a small validation set of size 200 with 5% uncensored second event. The whiskers indicates 95% confidence intervals.

We can see that, when the support overlap percentage is less than around 40% the prediction performance actually becomes worse when the two responses are trained together. One solution to this problem is to also treat α as a hyperparameter and use the validation set to determine it. For large data set having a two-dimensional hyperparameter could be quite cumbersome. In our simulation and real data application, we only choose α from two values $\{0, \sqrt{K}\}$, although in principle one can use a large set of α candidates at a higher computational cost. The right panel of Figure 2 shows the C-index improvements when we use a small validation set to determine whether α should be 0 or $\sqrt{K}$. All other settings are the same as above.

4 Application to UK Biobank Data

In this section we apply the proposed method to UK Biobank data. We focus on thyroiditis, which has 808 observed events in the study population ($337,129$ white British participants). We randomly assign 70% of the samples to the training set, 10% to the validation set, and 20% to the test set. The predictors here are ~ 1 million genetic variants, as well as 11 covariates (sex, and 10 principal components of the genetic variants). We first fit a baseline model using only the 11 covariates, without using any genetic variants. This gives a baseline test C-index at 0.649. We then fit a single-response Lasso Cox regression, where λ is selected using based on the validation C-index. At the best λ value, the test C-index is 0.679. To apply our approach, we pair thyroiditis with 6 other more common endocrine diseases that we believe share common genetic factors with thyroiditis. These diseases are listed in the table below. Here we report the test C-indices when thyroiditis is paired with one other disease and when all 7 responses are trained together. We also report the test C-indices when we use the validation set to determine both the optimal λ and whether to set $\alpha = 0$ or $\alpha = \sqrt{K}$.

Table 1 shows clear increase in prediction performance on thyroiditis when all 7 responses are trained together, and when thyroiditis is paired with thyrotoxicosis only. On the other hand, all multi-response solutions have test C-index comparable with the single response solution.

Paired response(s)	Test C-index	Test C-index (validate α)
thyroiditis (single response)	0.679	-
other hypothyroidism	**0.682**	**0.682**
other non-toxic goitre	**0.681**	0.679
thyrotoxicosis (hyperthyroidism)	**0.686**	**0.686**
other disorders of thyroid	0.679	0.679
insulin-dependent diabetes mellitus	**0.683**	**0.683**
non-insulin-dependent diabetes mellitus	0.679	0.679
all 7 responses trained together	**0.688**	**0.688**

Table 1: Test C-index of thyroiditis when this response is paired with other ones. The second column is obtained when α is fixed at $\sqrt{K}$ (which is 1 in the first row, $\sqrt{7}$ in the last row, and $\sqrt{2}$ in the rest). The third column is obtained when we use the validation set to choose α between 0 (single-response) and $\sqrt{K}$. Improved C-indices are given in boldface. Baseline test C-index of thyroiditis, when only the 11 covariates are used for fitting, is 0.649.

5 Conclusion and Discussion

We developed a regularized regression method for multiple survival responses. This method is particularly suitable when we can pair a response with few observed events with others having a larger number of events, such that these responses have a same set of useful predictors. We demonstrate the improvements in the prediction accuracy for the rare events responses through simulation and real data applications. We also provide efficient implementation for the proposed method.

Here are two directions for future studies. When there are more than two responses, the relationship between the prediction performance and the degree of overlapping in the coefficients support is yet to be understood. On the practical side, it is reasonable to have different types of models (not just Cox model), or even different response types (such as binary or count) to boost the accuracy of survival analysis on rare events. In principle the same type of regularization could still be applied.

For the GPU implementation, one challenge is the limited amount of memory available on GPUs. In modern computer clusters it is common to have machines with hundreds of Gigabytes of system memory, but most GPUs have less than 32GB of memory. In our implementation we use single precision floating point numbers to store the data matrix, which alleviates memory burden by a factor of two. However, for UK Biobank scale data this is still sometimes insufficient. For example, when the number of participants in the training data is $250,000$, we are only able to fit a model with up to $32,000$ variables on a Tesla V100. One possible solution is to use multiple GPUs, but inter-GPU communication might become the bottleneck. For genetics data, another solution is to utilize their 2-bit representation, which can significantly reduce memory requirement (2 bits per entry vs 32). We leave these ideas for future studies.

6 Proof of Propositions

6.1 Proof of Proposition 1

We show a slightly more general result from convex analysis. Let $f : \mathbb{R}^K \mapsto \mathbb{R}$ be continuously differentiable, convex, and bounded from below. Let $\|\cdot\|$ be a norm on $\mathbb{R}^K$, and $\|\cdot\|_*$ be its corresponding dual norm. Let $\lambda > 0$, and set

$$x^* := \arg\min_x f(x) + \lambda\|x\| \tag{8}$$

We will show that

$$\|\nabla f(x^*)\|_* \begin{cases} \leq \lambda & \text{if } x^* = 0 \\ = \lambda & \text{if } x^* \neq 0 \end{cases}. \tag{9}$$

It is clear that (8) is equivalent to the constrained optimization problem

$$\min_{x,y} f(x) + \lambda\|y\| \text{ such that } x = y. \tag{10}$$

The Lagrangian is

$$\mathcal{L}(x, y, z) = f(x) + \lambda\|y\| + z^T(y - x). \tag{11}$$

The Lagrangian dual is

$$g(z) := \inf_{x,y} \mathcal{L}(x, y, z) = \inf_y \lambda\|y\| + z^T y + \inf_x f(x) - z^T x. \tag{12}$$

Using the definition of dual norm, when $\|z\|_* > \lambda$, the infimum of the first term above is $-\infty$, and when $\|z\|_* \leq \lambda$, the infimum is 0. Therefore

$$g(z) = \begin{cases} -\infty & \text{if } \|z\|_* > \lambda \\ \inf_x f(x) - z^T x & \text{if } \|z\|_* \leq \lambda \end{cases} \tag{13}$$

Therefore the dual solution $z^* := \arg\max_z g(z)$ must satisfy $\|z\|_* \leq \lambda$. Now we go back to the Lagrangian. Since the primal objective is convex and Slaters condition holds, the solution to the primal problem can be obtained through minimizing

$$\mathcal{L}(x, y, z^*) = f(x) + \lambda\|y\| + {z^*}^T(y - x). \tag{14}$$

which implies that, at the optimal x^* we must have $\nabla f(x^*) = z^*$, so

$$\|\nabla f(x^*)\|_* = \|z^*\|_* \leq \lambda, \text{ and } \lambda\|x^*\| + \nabla f(x^*)^T x^* = 0 \tag{15}$$

If $x^* = 0$, then the second equality is already satisfied, so we only need $\|\nabla f(x^*)\|_* \leq \lambda$. If $x^* \neq 0$, then by Holder's inequality

$$\lambda\|x^*\| = |\nabla f(x^*)^T x^*| \leq \|\nabla f(x^*)\|_* \|x^*\| \leq \lambda\|x^*\|, \tag{16}$$

so we must have $\|\nabla f(x^*)\|_* = \lambda$. Proposition 1 is a direct consequence of this result.

6.2 Proof of Proposition 2

We prove the claim for $\lambda = 1$. The regularization term can be written as

$$\|u\|_1 + \alpha\|u\|_2 = \sup_{\|v_1\|_\infty \leq 1} v_1^T u + \sup_{\|v_2\|_2 \leq \alpha} v_2^T u = \sup_{v \in \mathcal{B}_\alpha^*} v^T u \tag{17}$$

where $\mathcal{B}_\alpha^*$ is the unit dual ball

$$\mathcal{B}_\alpha^* := \{v_1 \in \mathbb{R}^K : \|v_1\|_\infty \leq 1\} \oplus \{v_2 \in \mathbb{R}^K : \|v_2\|_2 \leq \alpha\}. \tag{18}$$

That is $\|v\|_{\alpha^*} \leq 1$ if and only if $v \in \mathcal{B}_\alpha^*$, which by definition means that $v = v_1 + v_2$ for some $\|v_1\|_\infty \leq 1, \|v_2\|_2 \leq \alpha$. We must have (and it is sufficient to have)

$$\inf_{\|v_1\|_\infty < 1} \|v - v_1\|_2 \leq \alpha. \tag{19}$$

The infimum on the left-hand side is achieved when $v_1 = v - S_1(v, 1)$, which proves the claim.

Acknowledgments

Y.T. is supported by Funai Overseas Scholarship from Funai Foundation for Information Technology and the Stanford University School of Medicine.

M.A.R. is supported by Stanford University and a National Institute of Health center for Multi and Trans-ethnic Mapping of Mendelian and Complex Diseases grant (5U01 HG009080). This work was supported by National Human Genome Research Institute (NHGRI) of the National Institutes of Health (NIH) under awards R01HG010140. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

R.T was partially supported by NIH grant 5R01 EB001988-16 and NSF grant 19 DMS1208164.

T.H. was partially supported by grant DMS-1407548 from the National Science Foundation, and grant 5R01 EB 001988-21 from the National Institutes of Health.

This research has been conducted using the UK Biobank Resource under application number 24983. We thank all the participants in the study. The primary and processed data used to generate the analyses presented here are available in the UK Biobank access management system (`https://amsportal.ukbiobank.ac.uk/`) for application 24983, "Generating effective therapeutic hypotheses from genomic and hospital linkage data" (`http://www.ukbiobank.ac.uk/wp-content/uploads/2017/06/24983-Dr-Manuel-Rivas.pdf`), and the results are displayed in the Global Biobank Engine (`https://biobankengine.stanford.edu`).

Some of the computing for this project was performed on the Sherlock cluster. We would like to thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results.

Conflict of Interest: None declared.

References

Beck, A. & Teboulle, M. (2009), 'A fast iterative shrinkage-thresholding algorithm for linear inverse problems', *SIAM J. Img. Sci.* **2**(1), 183–202.
URL: *https://doi.org/10.1137/080716542*

Chang, C., Chow, C., Tellier, L., Vattikuti, S., Purcell, S. & Lee, J. (2014), 'Second-generation plink: Rising to the challenge of larger and richer datasets', *GigaScience* **4**.

Cox, D. R. (1972), 'Regression models and life-tables', *Journal of the Royal Statistical Society. Series B (Methodological)* **34**(2), 187–220.
URL: *http://www.jstor.org/stable/2985181*

Daubechies, I., Defrise, M. & De Mol, C. (2004), 'An iterative thresholding algorithm for linear inverse problems with a sparsity constraint', *Communications on Pure and Applied Mathematics* **57**(11), 1413–1457.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.20042*

DeBoever, C., Tanigawa, Y., Lindholm, M. E., McInnes, G., Lavertu, A., Ingelsson, E., Chang, C., Ashley, E. A., Bustamante, C. D., Daly, M. J. et al. (2018), 'Medical relevance of protein-truncating variants across 337,205 individuals in the uk biobank study', *Nature communications* **9**(1), 1–10.

Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L. & Rosati, R. A. (1982), 'Evaluating the yield of medical tests', *Jama* **247**(18), 2543–2546.

Kane, M., Emerson, J. & Weston, S. (2013), 'Scalable strategies for computing with massive data', *Journal of Statistical Software, Articles* **55**(14), 1–19.
URL: *https://www.jstatsoft.org/v055/i14*

Li, R., Chang, C., Justesen, J. M., Tanigawa, Y., Qian, J., Hastie, T., Rivas, M. A. & Tibshirani, R. (2020), 'Fast Lasso method for large-scale and ultrahigh-dimensional Cox model with applications to UK Biobank', *Biostatistics* . kxaa038.
URL: *https://doi.org/10.1093/biostatistics/kxaa038*

Nesterov, Y. (1983), A method for solving the convex programming problem with convergence rate $O(1/k^2)$.

Qian, J., Du, W., Tanigawa, Y., Aguirre, M., Tibshirani, R., Rivas, M. A. & Hastie, T. (2019), 'A fast and flexible algorithm for solving the lasso in large-scale and ultrahigh-dimensional problems', *bioRxiv* .
URL: *https://www.biorxiv.org/content/early/2019/05/07/630079*

Rivas, M. A., Pirinen, M., Conrad, D. F., Lek, M., Tsang, E. K., Karczewski, K. J., Maller, J. B., Kukurba, K. R., DeLuca, D. S., Fromer, M. et al. (2015), 'Effect of predicted protein-truncating genetic variants on the human transcriptome', *Science* **348**(6235), 666–669.

Simon, N., Friedman, J., Hastie, T. & Tibshirani, R. (2013), 'A sparse-group lasso', *Journal of Computational and Graphical Statistics* **22**(2), 231–245.
URL: *https://doi.org/10.1080/10618600.2012.681250*

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., Liu, B., Matthews, P., Ong, G., Pell, J., Silman, A., Young, A., Sprosen, T., Peakman, T. & Collins, R. (2015), 'Uk biobank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age', *PLOS Medicine* **12**(3), 1–10.
URL: *https://doi.org/10.1371/journal.pmed.1001779*

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* **58**(1), 267–288.
URL: *http://www.jstor.org/stable/2346178*

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J. & Tibshirani, R. J. (2012), 'Strong rules for discarding predictors in lasso-type problems', *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **74**(2), 245–266.
URL: *http://www.jstor.org/stable/41430939*

Yuan, M. & Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society Series B* **68**, 49–67.