UCLA
Research Reports

Title

Scalable Sparse Cox's Regression for Large-Scale Survival Data via Broken Adaptive Ridge

Permalink
https://escholarship.org/uc/item/0gs54401

Authors
Kawaguchi, Eric S
Suchard, Marc A
Liu, Zhenqiu
et al.

Publication Date
2018-11-07

eScholarship.org Powered by the California Digital Library
University of California

Scalable Sparse Cox's Regression for Large-Scale Survival Data via Broken Adaptive Ridge

Eric S. Kawaguchi, Marc A. Suchard, Zhenqiu Liu, Gang Li [*]

Abstract

This paper develops a new sparse Cox regression method for high-dimensional massive sample size survival data. Our method is an L_0-based iteratively reweighted L_2-penalized Cox regression, which inherits some appealing properties of both L_0 and L_2-penalized Cox regression while overcoming their limitations. We establish that it has an oracle property for selection and estimation and a grouping property for highly correlated covariates. We develop an efficient implementation for high-dimensional massive sample size survival data, which exhibits up to a 20-fold speedup over a competing method in our numerical studies. We also adapt our method to high-dimensional small sample size data. The performance of our method is illustrated using simulations and real data examples.

Keywords: Censoring; Cox's proportional hazards model; High-dimensional covariates; Massive sample size; Penalized regression.

[*] University of California, Los Angeles, CA 90095-1772, USA. (E-mail: vli@ucla.edu).

1 Introduction

Advancing medical informatics tools and high-throughput biological experimentation are making large-scale data routinely accessible to researchers, administrators, and policy-makers. This "data deluge" poses new challenges and critical barriers for quantitative researchers as existing statistical methods and software grind to a halt when analyzing these large-scale datasets, and calls for a need of methods that can readily fit large-scale data. This paper primarily concerns survival analysis of *sparse high-dimensional massive sample size* (sHDMSS) data, a particular type of large-scale data with the following characteristics: 1) high-dimensional with large number (p_n in thousands or tens of thousands) of covariates, 2) massive in sample-size (n in thousands to hundreds of millions), 3) sparse in covariates with only a very small portion of covariates being nonzero for each subject, and 4) rare in event rate. A typical example of sHDMSS data is the pediatric trauma mortality data (**?**) from the National Trauma Databank (NTDB) maintained by the American College of Surgeons (**?**). This data set includes 210,555 patient records of injured children under 15 collected over 5 years from 2006 -2010. Each patient record includes 125,952 binary covariates that indicate the presence, or absence, of an attribute (ICD9 Codes, AIS codes, etc.) as well as their two-way interactions. The data matrix is extremely sparse with less than 1% of the covariates being non zero. The event rate is also very low at 2%. Another application domain where sHDMSS data are common is drug safety studies that use massive patient-level databases such as the U.S. FDA's Sentinel Initiative (*https://www.fda.gov/safety/fdassentinelinitiative/ucm2007250.htm*) and the Observational Health Data Sciences and Informatics (OHDSI) program (*https://ohdsi.org/*) to study rare adverse events with hundreds of millions of patient records and tens of thousands of patient attributes that are sparse in the covariates.

sHDMSS survival data presents multiple challenges to quantitative researchers. First, not all of the thousands of covariates are expected to be relevant to an outcome of interest. Traditionally, researchers hand-pick subject characteristics to include in an analysis. However, hand picking can introduce not only bias, but also a source of variability between researchers and studies. Moreover, it would become impractical and infeasible in large-scale evidence generation when hundreds or thousands of analyses are to be performed (**?**). Hence, automated sparse regression methods are desired. Secondly, the massive sample size presents a critical barrier to the application of existing sparse survival regression methods in a high-dimensional setting. While there are available many sparse survival regression methods including **???????**, current methods and standard software become inoperable for large datasets due to high computational costs and large memory requirements. **?** presented tools for fitting penalized Cox's regression on sHDMSS data with L_2 (ridge) and L_1 (LASSO) penalties. However, it is well known that ridge regression is not sparse and that although L_1-penalized regression produces a sparse solution, it tends to select too many noise variables and is biased for estimation. Lastly, the commonly used "divide and conquer" strategy for massive size data is deemed inappropriate for sHDMSS data since each of the divided data would typically be too sparse for a meaningful analysis. Improved scalable sparse regression methods for sHDMSS data are critically needed.

The purpose of this paper is to develop a new sparse Cox regression method, named Cox broken adaptive ridge (CoxBAR) regression, using L_0-based iteratively reweighted L_2-

penalized Cox regression, rigorously study its statistical properties for variable selection and estimation, develop efficient implementation strategies to make it scalable to sHDMSS survival data, and discuss its use for ultrahigh dimensional settings with small sample size. The CoxBAR estimator can be viewed as a Cox ridge estimator whose small coefficients are shrunk towards zero by iteratively applying a reweighted ridge regression that aims to approximate an L_0-penalized regression. There are several considerations that motivated us to study this approach. First, the CoxBAR algorithm aims to approximates an L_0-penalized Cox regression. It is well known that L_0-penalized regression is natural for variable selection, but has some limitations such as being unstable (?) and not scalable to high-dimensional settings. Second, our CoxBAR method uses the Cox ridge estimator as its initial value, which is not sparse, but has desirable predictive and grouping properties. In Lemma 2 of the appendix, we show that the CoxBAR algorithm essentially leaves the large values of the initial Cox ridge estimator almost unchanged, but shrinks its small values to zero using an approximate L_0-penalized regression. Consequently, the CoxBAR estimator enjoys the best of ridge and L_0-penalized regression while overcoming their limitations. For example, we will show that the CoxBAR estimator is selection consistent (an L_0-penalized regression property), estimation consistent (a ridge estimator property), stable (a ridge estimator property), and has a grouping property for highly correlated covariates (a ridge estimator property). Third, the CoxBAR algorithm only involves repeatedly performing reweighted L_2-penalized convex optimization, which allows one to take advantage of existing efficient algorithms for large scale L_2-penalized optimization with only minor modifications. Lastly, we illustrate in Section 3 that CoxBAR does not require costly data-driven tuning parameter selection, which is a big advantage over other sparse regression methods for fitting large-scale survival data.

Unlike other penalized regression methods that produce a sparse solution in a single step, the CoxBAR method is not sparse per se at each iteration and only achieves sparsity at its limit. This makes it difficult to study its statistical properties. A key innovation of this paper is to rigorously develop asymptotic theory for the CoxBAR estimator, establishing its consistency in identifying the sparse structure of the model and in parameter estimation as well as its grouping property for highly correlated covariates, with diverging dimension. To this end, we point out that ?? independently introduced similar algorithms for variable selection in linear models and generalized linear models, but only investigated their statistical properties using empirical studies. The idea of iteratively reweighted penalizations dates back at least to the well-known Lawson's algorithm (?) in classical approximation theory, which has been applied to various applications including L_d $(0 < d < 1)$ minimization (?), sparse signal reconstruction (?), compressive sensing (?????), and variable selection for linear models and generalized linear models (??). However, previous theoretical studies of this approach have focused only on algorithm convergence properties. This paper not only extends this approach to Cox's regression, but also for the first time provides rigorously justified large sample statistical theory. The second key contribution of this paper is an efficient implementation of CoxBAR on sHDMSS data by taking advantage of existing efficient large-scale L_2-penalized Cox regression tools, the sparsity in the data and in the Cox partial likelihood, and showing that CoxBAR does not require costly data-driven tuning parameter selection. It is also worth noting that the performance of an iteratively reweighted penalization method depends highly on its initial value, but there is not available an explicit guidance on how to choose the initial value in the literature. Our CoxBAR algorithm explicitly uses a Cox ridge regression estimate as its initial value. This seemingly obvious choice has important implications on the stability of the CoxBAR method and is responsible for its asymptotic consistency as well as its grouping property for highly correlated covariates as revealed in our theoretical derivations. Finally, in addition to its application to sHDMSS survival data, our large sample theory implies that CoxBAR can be combined with a sure screening method to yield a new two-stage survival regression procedure that is consistent for simultaneous variable selection and parameter estimation in ultrahigh dimensional settings with relatively small sample size.

In Section 2, we formally define the CoxBAR method, state our theoretical results on its selection and estimation consistency as well as its grouping property, describe an efficient implementation of CoxBAR on sHDMSS survival data, and discuss how to adapt it as a post-screening sparse regression method for ultrahigh dimensional covariates with relatively small sample size. Simulation studies are presented in Section 3 to demonstrate the performance of the CoxBAR estimator with both moderate and massive sample size in various low and high-dimensional settings. Real data examples including an application of CoxBAR on the pediatric trauma mortality data (**?**) are given in Section 4. Closing remarks and discussion are given in Section 5. Proofs of the theoretical results and regularity conditions needed for the derivations are collected in the appendix. An R package for CoxBAR is available at *https://github.com/OHDSI/BrokenAdaptiveRidge*.

2 Methodology

2.1 Cox's broken adaptive ridge regression and its large sample properties

2.1.1 The Estimator

Suppose that one observes a random sample of right-censored survival data consisting of n independent and identically distributed triplets, $(\tilde{T}_i, \delta_i, \mathbf{x}_i)$, $i = 1, \ldots, n$, where for subject i, $\tilde{T}_i = \min(T_i, C_i)$ is the observed time, $\delta_i = I(T_i \leq C_i)$ is the censoring indicator, T_i is a survival time of interest, and C_i is a censoring time that is conditionally independent of T_i given a p_n-dimensional covariate vector $\mathbf{x_i}$.

Assume the Cox (**?**) proportional hazard model

$$h(t|\mathbf{x}) = h_0(t)\exp(\mathbf{x}^T\boldsymbol{\beta}), \tag{1}$$

where $h(t|\mathbf{x})$ is the conditional hazard function of a survival time t given a p_n-dimensional covariate vector $\mathbf{x}$, $h_0(t)$ is an unspecified baseline hazard function, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{p_n})$ is a vector of regression coefficients. Adopting the counting process notation of **?**, the log-partial likelihood for the Cox model is defined as

$$l_n(\boldsymbol{\beta}) = \sum_{i=1}^{n}\int_0^1 \boldsymbol{\beta}^T\mathbf{x}_i(s)dN_i(s) - \int_0^1 \ln\left[\sum_{j=1}^{n} Y_j(s)\exp\{\boldsymbol{\beta}^T\mathbf{x}_j(s)\}\right] d\bar{N}(s). \tag{2}$$

Here $Y_i(s) = I(\tilde{T}_i \geq s)$ is the at-risk process, $N_i(s) = I(T_i \leq s, T_i \leq C_i)$, and $\bar{N} = \sum_{i=1}^n N_i$. Without loss of generality, we work on the time interval $s \in [0, 1]$ as in **?**, which can be extended to the time interval $[0, \tau]$ for $0 < \tau < \infty$ without difficulty.

Our Cox's broken adaptive ridge (CoxBAR) estimation of $\boldsymbol{\beta}$ starts with an initial Cox ridge regression estimator (**?**)

$$\hat{\boldsymbol{\beta}}^{(0)} = \arg\min_{\boldsymbol{\beta}} \{-2l_n(\boldsymbol{\beta}) + \xi_n \sum_{j=1}^{p_n} \beta_j^2\}, \tag{3}$$

which is updated iteratively by a reweighed L_2-penalized Cox regression estimator

$$\hat{\boldsymbol{\beta}}^{(k)} = \arg\min_{\boldsymbol{\beta}} \left\{ -2l_n(\boldsymbol{\beta}) + \lambda_n \sum_{j=1}^{p_n} \frac{\beta_j^2}{(\hat{\beta}_j^{(k-1)})^2} \right\}, \quad k \geq 1. \tag{4}$$

where ξ_n and λ_n are non-negative penalization tuning parameters. The CoxBAR estimator is defined as

$$\hat{\boldsymbol{\beta}} = \lim_{k \to \infty} \hat{\boldsymbol{\beta}}^{(k)}. \tag{5}$$

Remark 1 *(Computation of CoxBAR) For small to moderate size data, one may simply calculate the CoxBAR estimator through iterating a Newton-Raphson type procedure as in* **?***, who outlined an iterative reweighted least squares algorithm for generalized linear models. However, the procedure calls for calculating both the gradient and Hessian at each iteration, which will become infeasible in large-scale settings with large n and p_n due to high computational costs, high memory requirements, and numerical instability. In the Section 2.2, we will discuss an efficient algorithm that adapts efficient algorithms for L_2-penalized Cox regression and accounts for the sparsity in sHDMSS data and in the partial likelihood to make CoxBAR scalable to sHDMSS data.*

Remark 2 *Since L_2 penalization yields a non-sparse solution, defining the CoxBAR estimator as the limit is necessary to produce sparsity. Although λ_n is fixed at each iteration, it is weighted inversely by the square of the ridge regression estimates from the previous iteration. Consequently, coefficients whose true values are zero will have larger penalties in the next iteration; whereas penalties for truly non-zero coefficients will converge to a constant. Theorem 1 below proves that the estimates of the truly zero coefficients shrink closer towards zero while the estimates of the truly non-zero coefficients converge to their true values.*

2.1.2 Oracle properties

Below we establish the oracle properties for the CoxBAR estimator for simultaneous variable selection and parameter estimation. Define $\boldsymbol{\beta}_0 = \left(\boldsymbol{\beta}_{01}^T, \boldsymbol{\beta}_{02}^T\right)^T$ as the true parameter values of the model where, without loss of generality, $\boldsymbol{\beta}_{01} = (\beta_{01}, ..., \beta_{0q})$ is a vector of q non-zero values and $\boldsymbol{\beta}_{02} = \mathbf{0}$ is a $p_n - q$ dimensional vector of zeros. Let $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2$ be the first q and the remaining $p_n - q$ components of the CoxBAR estimator, $\hat{\boldsymbol{\beta}}$, respectively.

Theorem 1 (Oracle Properties) *Assume the regularity conditions (C1) - (C6) from Appendix A.1 hold. Then, with probability tending to 1,* *(a) the CoxBAR estimator* $\hat{\boldsymbol{\beta}} = \left(\hat{\boldsymbol{\beta}}_1^T, \hat{\boldsymbol{\beta}}_2^T\right)^T$ *exists and is unique with* $\hat{\boldsymbol{\beta}}_2 = \mathbf{0}$ *and* $\hat{\boldsymbol{\beta}}_1$ *being the unique fixed point of* $f(\boldsymbol{\beta}_1)$*, where* $f(\boldsymbol{\beta}_1)$ *is a solution to* $\dot{Q}_{n1}(\boldsymbol{\theta}) = \mathbf{0}$ *for*

$$Q_{n1}(\boldsymbol{\theta}_1) = -2l_{n1}(\boldsymbol{\theta}_1) + \lambda_n \boldsymbol{\theta}_1^T D_1(\boldsymbol{\beta}_1)\boldsymbol{\theta}_1,$$

with $D_1(\boldsymbol{\beta}_1) = diag(\beta_1^{-2}, \beta_2^{-2}, \ldots, \beta_q^{-2})$ *and* $l_{n1}(\boldsymbol{\theta}_1)$ *being the first* q *components of* $l_n(\boldsymbol{\theta})$*.*

(b) $\sqrt{n}(\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{01}) \xrightarrow{D} N(\mathbf{0}, I_1(\boldsymbol{\beta})^{-1})$*, where* $I_1(\boldsymbol{\beta})^{-1}$ *is the leading* $q \times q$ *submatrix of* $I(\boldsymbol{\beta})^{-1}$*.*

2.1.3 The grouping property

When the true model has a group structure, it is desirable for a variable selection method to either retain or drop all variables that are clustered within the same group. Ridge regression has a grouping property, and it is intuitive to conjecture that the CoxBAR method would as well since the estimator is based on an iterative ridge regression. The following theorem states the grouping property of the CoxBAR estimator for highly correlated covariates.

Theorem 2 *Let* λ_n*,* $\{(\tilde{T}_i, \delta_i, \mathbf{x}_i)\}_{i=1}^n$*, and the CoxBAR estimator* $\hat{\boldsymbol{\beta}}$ *be given, and assume that* $X = (\mathbf{x}_i^T, \ldots \mathbf{x}_n^T)$ *is standardized. That is, for all* $j = 1, \ldots, p_n$*,* $\sum_{i=1}^n x_{ij} = 0$*,* $\mathbf{x}_{[,j]}^T \mathbf{x}_{[,j]} = n-1$*, where* $\mathbf{x}_{[,j]}$ *is the* j^{th} *column of* X*. Then for any* $\hat{\beta}_i \neq 0$ *and* $\hat{\beta}_j \neq 0$*,*

$$|\hat{\beta}_i^{-1} - \hat{\beta}_j^{-1}| \leq \frac{1}{\lambda_n}\sqrt{2\{(n-1)-(n-1)r_{ij}\}}\sqrt{n(1+d)^2}, \tag{6}$$

with probability tending to one, where $d_n = \sum_{i=1}^n \delta_i$*, and* $r_{ij} = \frac{1}{n-1}\mathbf{x}_{[,i]}^T \mathbf{x}_{[,j]}$ *is the sample correlation of* $\mathbf{x}_{[,i]}$ *and* $\mathbf{x}_{[,j]}$*.*

We can see that as $r_{ij} \to 1$, the absolute difference between $\hat{\beta}_i$ and $\hat{\beta}_j$ approaches 0 implying that the estimated coefficients of two highly correlated variables will be similar in magnitude.

2.1.4 Selection of tuning parameters

The CoxBAR method depends on two tuning parameters: ξ_n for the initial ridge estimator in (3) and λ_n for the iterative ridge step in (4). Commonly used data-driven methods such as k-fold cross-validation (**?**), Akaike information criterion (AIC) (**?**), Bayesian information criterion (BIC) (**??**) could be used to search for the optimal pair, (ξ_n, λ_n). While the data-driven selection methods work for moderate-size data, they are computationally costly for large-scale data. Because the iteratively reweighted ridge regression aims to approximate L_0-penalized Cox regression, it is natural to consider fixing λ_n at $\ln(n)$ or $\ln(d_n) \equiv \ln(\text{number of uncensored events})$, which correspond to the regular and censored BIC penalty, respectively (**???**).

Our simulations in Section 3 suggest that while fixing λ_n at $\ln(n)$ or $\ln(d_n)$, the CoxBAR estimator is very stable to the choice of ξ_n over a wide range interval (Figure 1). Furthermore, both $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ work well in practice for the $p_n < n$ setting (Table 1). Thus, costly data-driven determination of the tuning parameters can be avoided by using pre-specified values for λ_n and ξ_n, which is crucial to reducing the computational burden for fitting large-scale survival data as discussed in the next two subsections.

2.2 CoxBAR for sparse high-dimensional massive sample size (sHDMSS) data

As mentioned earlier, the Newton-Raphson algorithm used for each iteration of the CoxBAR algorithm will become infeasible in large-scale settings with large n and p_n due to high computational costs, high memory requirements, and numerical instability. Because CoxBAR only involves fitting a reweighted Cox's ridge regression at each of its iteration step, it allows us to take advantage of existing efficient algorithms for large-scale Cox ridge regression with only minor modifications as detailed below.

2.2.1 Adaptation of existing algorithms for fitting massive L_2-penalized Cox's regression

? developed an efficient implementation of the massive Cox's ridge regression for sHDMSS data using a Bayesian framework, under which finding the maximum a posteriori (MAP) estimates of $\boldsymbol{\beta}$ based on the joint penalized partial likelihood $L_p(\boldsymbol{\beta}) \propto L_n(\boldsymbol{\beta}|\mathbf{D})\pi(\boldsymbol{\beta})$ is equivalent to fitting a Cox ridge regression with tuning parameter ϕ, where $L_n(\boldsymbol{\beta}|\mathbf{D}) = \exp\{l_n(\boldsymbol{\beta})\}$, $\mathbf{D} = \{(\tilde{T}_i, \delta_i, \mathbf{x}_i) : i = 1, \dots n\}$, and $\pi(\boldsymbol{\beta})$ is the prior distribution of $\boldsymbol{\beta}$ with independent marginal normal distributions

$$\pi(\beta_j|\phi) \sim N(0, \phi^{-1}), \quad \phi > 0 \tag{7}$$

for $j = 1, \dots, p_n$ (**??**). For parameter estimation, **?** adopted the column relaxation with logistic loss (CLG) algorithm of **?**, which is a type of cyclic coordinate descent algorithm that estimates the coefficients using 1D updates. The CLG easily scales to high-dimensional data (**???**) and has been recently implemented for fitting massive ridge and LASSO penalized generalized linear models (**?**), parametric survival models (**?**), and Cox 's model (**?**). When fitting this Cox ridge regression model, the CLG algorithm involves finding $\beta_j^{(new)}$, the value of the j^{th} entry of $\boldsymbol{\beta}$, that minimizes $-l_p(\boldsymbol{\beta})$, assuming that the other values of β_j's are held constant at their current values. Using the prior from Equation (7) and ignoring the constants in the posterior partial likelihood, finding $\beta_j^{(new)}$ is equivalent to finding the z that minimizes,

$$g(z) = -z\sum_{i=1}^{n}\delta_i x_{ij} + \sum_{i=1}^{n}\delta_i \ln\left\{\sum_{y\in R(\tilde{T}_i)}\exp\left(\sum_{k=1,k\neq j}^{p_n}\beta_k x_{yk} + z x_{yj}\right)\right\} + \frac{z^2}{2\phi}, \tag{8}$$

where $R(\tilde{T}_i) = \{j : \tilde{T}_j > \tilde{T}_i\}$ is the risk set for observation i. Even for this 1D problem, an optimization procedure needs to be used since there is no closed form solution. Using a Taylor series approximation at the current β_j, one can approximate $g(\cdot)$ through

$$g(z) \approx g(\beta_j) + g'(\beta_j)(z-\beta_j) + \frac{1}{2}g''(\beta_j)(z-\beta_j)^2, \tag{9}$$

where

$$g'(\beta_j) = \frac{d}{dz}g(z)\bigg|_{z=\beta_j} = -\sum_{i=1}^{n}x_{ij}\delta_i + \sum_{i=1}^{n}\delta_i\frac{\sum_{y\in R(\tilde{T}_i)}x_{yj}\exp(\boldsymbol{\beta}^T\mathbf{x}_y)}{\sum_{y\in R(\tilde{T}_i)}\exp(\boldsymbol{\beta}^T\mathbf{x}_y)} + \frac{\beta_j}{\phi}, \tag{10}$$

and

$$g''(\beta_j) = \frac{d^2}{dz^2}g(z)\bigg|_{z=\beta_j} = \sum_{i=1}^{n} \delta_i \frac{\sum_{y \in R(\tilde{T}_i)} x_{yj}^2 \exp(\boldsymbol{\beta}^T \mathbf{x}_y)}{\sum_{y \in R(\tilde{T}_i)} \exp(\boldsymbol{\beta}^T \mathbf{x}_y)} - \left(\sum_{i=1}^{n} \delta_i \frac{\sum_{y \in R(\tilde{T}_i)} x_{yj} \exp(\boldsymbol{\beta}^T \mathbf{x}_y)}{\sum_{y \in R(\tilde{T}_i)} \exp(\boldsymbol{\beta}^T \mathbf{x}_y)} \right) + \frac{1}{\phi}. \tag{11}$$

Consequently, the Taylor series approximation in Equation (9) has its minimum at

$$\beta_j^{(new)} = \beta_j + \Delta\beta_j = \beta_j - \frac{g'(\beta_j)}{g''(\beta_j)}. \tag{12}$$

Furthermore, the above algorithm of **?** adopts multiple aspects of the work by **?** and **?**. For CLG, a trust region approach is implemented so that $|\Delta\beta_j|$ is not allowed to be too large on a single iteration. This prevents large updates in regions where a quadratic is a poor approximation to the objective. Second, rather than iteratively updating $\beta_j^{(new)} = \beta_j + \Delta\beta_j$ until convergence, CLG does this only once before going on to the next variable. Since the optimal value of $\beta_j^{(new)}$ depends on the current value of the other β_j's, there is little reason to tune each $\beta_j^{(new)}$ with high precision. Instead, we simply want to decrease $-l_p(\boldsymbol{\beta})$ before going on to the next β_j.

The above discussed techniques for fitting massive Cox's ridge regression is adapted directly within each step of our CoxBAR algorithm with the following priors: for $j = 1, \ldots, p_n$

$$\pi(\beta_j^{(0)}|\xi_n) \sim N(0, \xi_n^{-1}), \tag{13}$$

for $k = 0$, and for $k \geq 1$

$$\pi(\beta_j^{(k)}|\lambda_n, \hat{\beta}^{(k-1)}) \sim N(0, (\hat{\beta}_j^{(k-1)})^2/\lambda_n^{-1}). \tag{14}$$

We iterate with respect to k until convergence is attained.

2.2.2 Efficient computing and storage by accounting for sparsity in the covariate structure and partial likelihood

Recall that the design matrix X for sHDMSS data has few non-zero entries for each subject. Storing such a sparse matrix as a dense matrix is inefficient and may increase computation time and/or cause a standard software to crash due to insufficient memory allocation. To the best of our knowledge, popular penalization packages such as `glmnet` (**?**) and `ncvreg` (**?**) do not support a sparse data format as an input for right-censored survival models, although the former supports the input for other generalized linear models. For sHDMSS data, we propose to use specialized, column-data structures as in **?** and **?**. The advantage of this structure is two-fold: it significantly reduces the memory requirement needed to store the covariate information, and performance is enhanced when employing cyclic coordinate descent. For example when updating β_j, efficiency is gained when computing and storing the inner product $r_i = \boldsymbol{\beta}^T \mathbf{x}_i$ using a low-rank update $r_i^{(new)} = r_i + x_{ij} + \Delta\beta_j$ for all i (**?????**).

Furthermore, as seen in equations (10) and (11) , one would need to calculate the series of cumulative sums introduced through the risk set $R(\tilde{T}_i) = \{j : \tilde{T}_j > \tilde{T}_i\}$ for each subject i. These cumulative sums would need to be calculated when updating each parameter estimate in the optimization routine. This can prove to be computationally costly, especially when both n and p_n are large. By taking advantage of the sparsity of the design matrix, one can reduce the computational time needed to calculate these cumulative sums by entering into this operation only if at least one observation in the risk set has a non-zero covariate value along dimension j and embarking on the scan at the first non-zero entry rather than from the beginning, which have been implemented for massive Cox's ridge regression by **?** and for a conditional Poisson regression model by **?**.

Our CoxBAR implementation naturally exploits the sparsity in the data matrix and in the partial likelihood by imbedding an adaptive version of **?**'s massive Cox's ridge regression within each iteration of the iteratively reweighted Cox's ridge regression.

We finally highlight that our CoxBAR method uses pre-specified tuning parameters as discussed in Section 2.1.4, which provides huge computation savings.

2.3 CoxBAR for Ultrahigh-Dimensional Data

We now discuss how to CoxBAR to the ultrahigh dimensional survival data setting where the number of covariates far exceeds the number of observations in the dataset. When p_n is much higher than n, it is crucial to first employ some screening procedure to reduce the number of covariates to be less than the sample size while simultaneously having the true model be nested within the screened out model, before applying CoxBAR. There are a number of screening methods for right-censored survival data, which include marginal screening methods (**????**) and joint screening methods (**?**). For example, the sure independent screening method of **?**, SIS measures the importance of the covariates based on the marginal partial likelihood, which is fast, but may overlook important covariates that are jointly correlated, but not marginally correlated, with the observed survival time. The sure joint screening method of **?** is based on the joint partial likelihood of potentially important covariates using a sparsity-restricted maximum partial likelihood estimate. Under certain regularity conditions, these methods have been shown to possess the sure screening property in the sense that the subset of retained covariates includes the true model with probability tending to one.

Our theoretical results in Section 2.1 guarantee that our CoxBAR estimator has the oracle property and grouping property when applied after a sure screening procedure. For example, let SJS-CoxBAR denote the two-step estimator obtained by first performing the SJS method of **?** and then applying CoxBAR postscreening. The following theorem proves that SJS-CoxBAR is an oracle estimator under certain conditions.

Theorem 3 *Let $p_n = O(n^d)$, $0 \leq d < 1$. For the full model suppose that Conditions (C1) - (C3) from Appendix A.1 and Conditions (D1) - (D7) from* **?** *hold. Furthermore, suppose Conditions (C4) - (C6) from Appendix A.1 hold for the sub model, $\hat{s}$, obtained after sure joint screening. Define $l_{\hat{s}}(\boldsymbol{\beta})$ as the log-partial likelihood of the model corresponding to $\hat{s}$. Then with probability tending to one,* *(a) the CoxBAR estimator* $\hat{\boldsymbol{\beta}} = \left(\hat{\boldsymbol{\beta}}_1^T, \hat{\boldsymbol{\beta}}_2^T\right)^T$ *exists and is unique with* $\hat{\boldsymbol{\beta}}_2 = \mathbf{0}$ *and* $\hat{\boldsymbol{\beta}}_1$ *being the unique fixed point of* $f(\boldsymbol{\beta}_1)$*, where* $f(\boldsymbol{\beta}_1)$ *is a solution to* $\dot{Q}_{\hat{s}1}(\boldsymbol{\theta}) = \mathbf{0}$ *for*

$$Q_{\hat{s}1}(\boldsymbol{\theta}_1) = -2l_{\hat{s}1}(\boldsymbol{\theta}_1) + \lambda_n \boldsymbol{\theta}_1^T D_1(\boldsymbol{\beta}_1)\boldsymbol{\theta}_1,$$

with $D_1(\boldsymbol{\beta}_1) = diag(\beta_1^{-2}, \beta_2^{-2}, \ldots, \beta_q^{-2})$ *and* $l_{\hat{s}1}(\boldsymbol{\theta}_1)$ *being the first* q *components of* $l_{\hat{s}}(\boldsymbol{\theta})$*.*

(b) $\sqrt{n}(\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{01}) \xrightarrow{D} N(\mathbf{0}, I_1(\boldsymbol{\beta})^{-1})$*, where* $I_1(\boldsymbol{\beta})^{-1}$ *is the leading* $q \times q$ *submatrix of* $I(\boldsymbol{\beta})^{-1}$*.*

3 Simulations

In this section we demonstrate the performance of the CoxBAR estimator using simulations. We first show, in Section 3.1, that our method is robust to the selection of ξ_n, thus suggesting that data-driven methods such as cross-validation for selecting ξ_n are unnecessary. In Section 3.2, we evaluate and compare the variable selection and parameter estimation performance of our CoxBAR method to some popular variable selection methods under various $p_n \leq n$ settings. Section 3.3 presents a large-scale sparse data simulation. Finally, Section 3.4 presents results of the two-stage CoxBAR procedure for ultrahigh dimensional data.

With the exception of Section 3.3 we use the same structure to simulate the data for the empirical studies. Observations for each of the 100 Monte Carlo samples are drawn from the following exponential hazard model,

$$h(t_i|\mathbf{x}_i) = \exp(\mathbf{x_i}^T\boldsymbol{\beta^*}) \quad i = 1, \ldots, n, \tag{15}$$

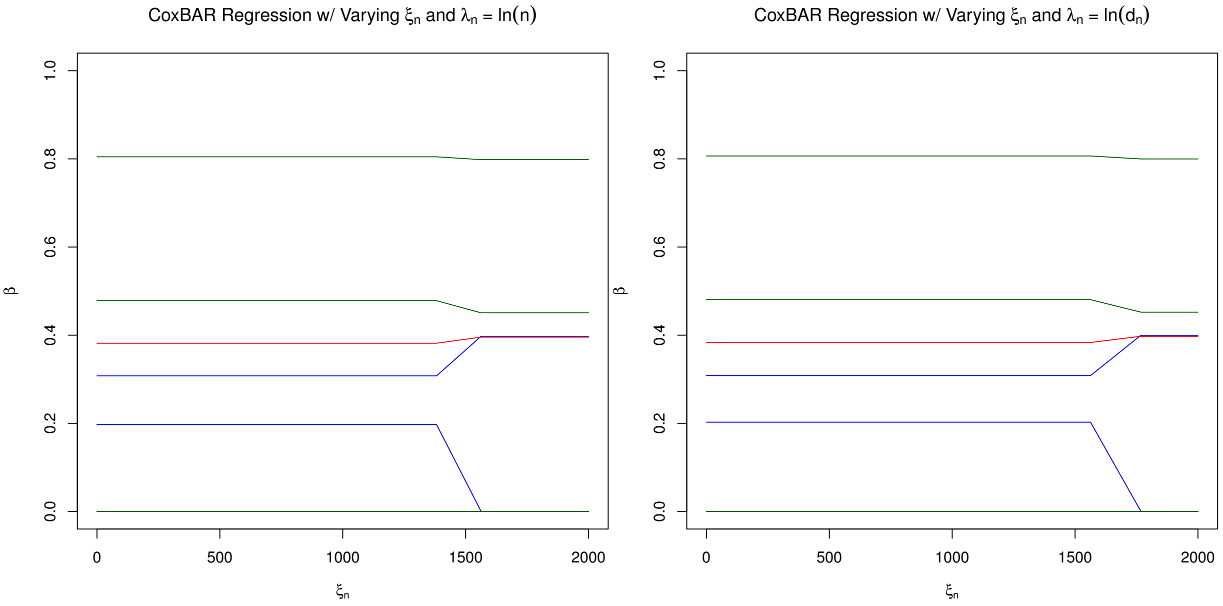
where $\boldsymbol{\beta}^* = (0.2, 0.2, 0, 0.5, 0.5, 0, 0, 0.7, 0.7, \mathbf{0}_{p_n-9})^T$. The design matrix $X = (\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T)$ was generated from a p_n-dimensional normal distribution with mean zero and covariance matrix $\Sigma = (\sigma_{ij})$ with an autoregressive structure such that $\sigma_{ij} = 0.5^{|i-j|}$. To simulate independent censoring we let $\delta_i \overset{iid}{\sim} Bern(.8)$, which corresponds to 20% censoring.

3.1 Sensitivity of CoxBAR with respect to ξ_n

Recall that the CoxBAR method relies on two tuning parameters: ξ_n for the initial Cox ridge regression and λ_n for the subsequent iteratively reweighted ridge regression. While fixing λ_n at $\ln(n)$ or $\ln(d_n)$, as discussed in Section 2.1.4, we illustrate below that CoxBAR is not sensitive to ξ_n. Using Model (15) we simulated a random sample of size $n = 300$ and $p_n = 10$. We fix $\lambda_n = \ln(n)$ or $\ln(d_n)$, and vary ξ_n over $[10^{-2}, 2000]$. The solution path plots are displayed in Figure 1. It is seen that the CoxBAR estimator is essentially unchanged over a large interval of ξ_n, suggesting that any pre-specified value in this interval for ξ_n would produce a reasonable CoxBAR estimate.

3.2 Model selection and parameter estimation

In this simulation study, we numerically compare the model selection and parameter estimation performance of CoxBAR with three popular sparse survival regression methods: LASSO



(a) Path plot for CoxBAR with $\lambda_n = \ln(n)$. (b) Path plot for CoxBAR with $\lambda_n = \ln(d_n)$.

Figure 1: Path plot for CoxBAR regression with varying ξ_n and fixed $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ for a random sample of size $n = 300$ and $p_n = 10$.

(**?**), SCAD (**?**) , and adaptive LASSO (**?**). Estimation bias is summarized through the mean sum of squared bias (SSB), $E\{\sum_{i=1}^{p_n}(\hat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i^*)^2\}$. Variable selection performance is measured by the mean number of false positives (FP), the mean number of false negatives (FN), and the mean number of misclassified coefficients (MC). All simulations were conducted using R. We use the R packages `glmnet` for LASSO (Cox-LASSO), and adaptive LASSO (Cox-ALASSO), and `ncvreg` for SCAD (Cox-SCAD) in our simulations. As mentioned earlier, Cox-ALASSO requires initial estimates for its weights. Since Section 3.1 yields evidence that cross validating over ξ_n is not needed for the CoxBAR method, we fixed $\xi_n = 20$. Five-fold cross validation is used for selecting the optimal tuning parameter for Cox-LASSO, Cox-SCAD, and Cox-ALASSO. For the CoxBAR method we propose to look at $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$, which are labeled CoxBAR$_1$ and CoxBAR$_2$, respectively. We simulate data from Model (15) by fixing $n = 300$, and varying $p_n = 10, 50$, and 100. We simulated 100 Monte Carlo samples for each scenario. The results are summarized in Table 1.

It is seen from Table 1 that for variable selection, both CoxBAR$_1$ and CoxBAR$_2$ tend to have a much smaller MC with a much smaller FP and a slightly larger FN than other methods. For estimation, all considered methods with the exception of Cox-LASSO, have similar SSB. In summary, in comparison with other methods, CoxBAR$_1$ and CoxBAR$_2$ tend to yield more sparse and accurate models, with comparable estimation performance.

Table 1: (Low dimensional, moderate sample size) Simulated estimation and variable selection performance of CoxBAR, Cox-LASSO, Cox-SCAD, and Cox-ALASSO (CoxBAR$_1$ and CoxBAR$_2$ denote CoxBAR with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ respectively; SSB=mean sum of squared bias; FP=mean number of false positives; FN=mean number of false negatives; MC=mean number of misclassified coefficients (MC); Each entry is based on 100 Monte Carlo samples of size $n = 300$)

p_n	Method	SSB	FP	FN	MC
10	CoxBAR$_1$	0.09	0.01	1.08	1.09
	CoxBAR$_2$	0.09	0.01	1.07	1.08
	Cox-LASSO	0.07	2.00	0.05	2.05
	Cox-SCAD	0.07	1.30	0.23	1.53
	Cox-ALASSO	0.06	0.86	0.24	1.10
50	CoxBAR$_1$	0.10	0.04	1.16	1.20
	CoxBAR$_2$	0.10	0.04	1.14	1.18
	Cox-LASSO	0.12	8.56	0.03	8.59
	Cox-SCAD	0.10	3.46	0.29	3.75
	Cox-ALASSO	0.10	5.14	0.21	5.35
100	CoxBAR$_1$	0.10	0.08	1.09	1.17
	CoxBAR$_2$	0.09	0.11	1.04	1.15
	Cox-LASSO	0.14	10.68	0.15	10.83
	Cox-SCAD	0.10	5.04	0.34	5.38
	Cox-ALASSO	0.11	8.09	0.25	8.34

3.3 Large-scale sparse data

In this large scale data simulation, we generated a sHDMSS dataset with $n = 60000$ and $p_n = 20000$. Observed survival times are generated from an exponential model similar to Model (15) with $\boldsymbol{\beta}^* = (\mathbf{0.7}_{10}, -\mathbf{0.7}_{10}, \mathbf{0.5}_{10}, -\mathbf{0.5}_{10}, \mathbf{0}_{p_n-40})$. We allow the censoring rate to be 90% and the sparseness level to be 95%, such that each row of X has, on average, only 5% of the entries being assigned a non-zero value. We compared our CoxBAR method with the massive sparse Cox's regression for LASSO (mCox-LASSO) using the `Cyclops` package (**??**) which, to the best of our knowledge, is the only software available today that exploits the sparsity of the large-scale survival data for efficient computing. For mCox-LASSO, cross validation, combined with an efficient noncovex optimization technique instead of a grid search, was used to find the optimal value for the tuning parameter. For the CoxBAR method, we considered $\lambda_n = \ln(n)$ (CoxBAR$_1$) and $\lambda_n = \ln(d_n)$ (CoxBAR$_2$) while fixing $\xi_n = 20$. The results are summarized in Table 2.

Table 2: (High dimensional, massive sample size) Runtime, estimation, and variable selection results of CoxBAR and the massive Cox regression with LASSO penalty (mCox-LASSO, **?**) for a simulated sHDMSS dataset with $n = 60,000$ and $p_n = 20,000$. (CoxBAR$_1$ and CoxBAR$_2$ denote CoxBAR with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ respectively; SSB= sum of squared bias; FP= number of false positives; FN= number of false negatives; MC= number of misclassified coefficients (MC).)

Method	Runtime (minutes)	SSB	FP	FN	MC
CoxBAR$_1$	20	0.39	0	0	0
CoxBAR$_2$	26	0.20	1	0	1
mCox-LASSO	400	3.19	20	0	20

As one would expect, mCox-LASSO has retained all 40 true nonzero coefficients, together with 20 noise variables. In contrast, CoxBAR$_1$ selected exactly the true model with 40 correctly retained nonzero coefficients with no noise variable. CoxBAR$_2$ has a similar performance to CoxBAR$_1$ with 1 noise variable. Both CoxBAR$_1$ and CoxBAR$_2$ have much smaller bias (SSB ≈ 0.39 and SSB ≈ 0.20, respectively) than mCox-LASSO (SSB ≈ 3.2). Moreover, although optimized in the `Cyclops` package, mCox-LASSO took over 400 minutes to run, whereas CoxBAR$_1$ or CoxBAR$_2$ took only around 20 minutes that represents a 20-fold speedup.

We further compared the solution paths of mCox-LASSO and CoxBAR in Figure 2. The black dashed line in the mCox-LASSO solution path plot (Figure 2(a)) represents the estimates at the optimal tuning parameter obtained via cross validation. We can see that the mCox-LASSO solution path changes rapidly as its tuning parameter varies. Thus it is important to use an optimal value for mCox-LASSO, which has to be selected using a data-driven procedure. mCox-LASSO also tends to keep a substantial number of noise variables with large estimation bias even at its optimal penalty value. In contrast, the CoxBAR solution path plot (Figure 2(b)) with respect to λ_n changes very slowly over a relative large interval that includes $\ln(n)$ (black solid vertical line) and $\ln(d_n)$ (black dotted vertical line), and correctly selects the true model with small estimation bias. For the CoxBAR method, we also made a CoxBAR solution path plot with respect to ξ_n, while fixing $\lambda_n = \ln(n)$ in Figure 2(c). It shows that the CoxBAR estimates are very stable and, in fact, correctly identifies the true model over a large range of ξ_n, which affirms our observation in Section 3.1 with small scale data.

3.4 Ultrahigh dimensional data

This section presents a simulation to illustrate the performance of our two-stage estimator SJS-CoxBAR described in Section 2.3 in ultrahigh dimensional settings where p_n is much larger than n. We generated data from model (15) with different combinations of $n = 300$, 500 and $p_n = 1660$, 5200. The sure joint screening method of **?** was initially used to choose a sub-model of size $m = \lfloor \frac{n}{2\ln(n)} \rfloor$, where$\lfloor \cdot \rfloor$ is the floor function. We then compared the performance of Cox-LASSO (SJS-LASSO), Cox-SCAD (SJS-SCAD), Cox-ALASSO (SJS-ALASSO) and CoxBAR (SJS-CoxBAR) on the screened model. Five-fold cross validation was used for SJS-LASSO, SJS-SCAD, and SJS-ALASSO whereas $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$, and $\xi_n = 1$ was used for SJS-CoxBAR. Part of the simulation results are reported in Table 3.

Similar to Section 3.2, we observe that the SJS-CoxBAR method has much fewer false positives (FP) and misclassifications (MC) than the other methods across all considered scenarios. The false negatives (FN) of SJS-CoxBAR is slightly higher than, but within the same order of magnitude as the other methods.

4 Real data examples

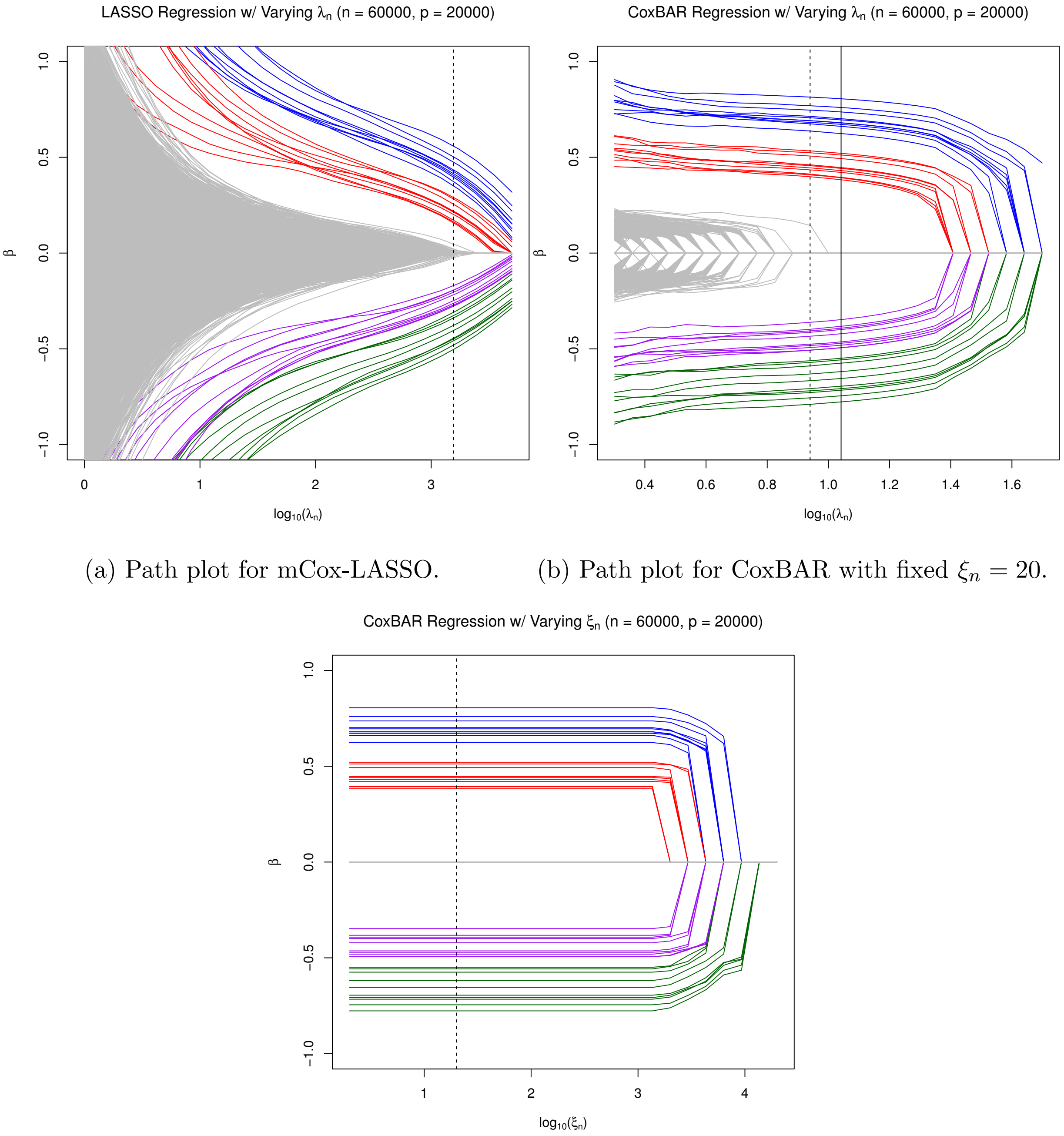
4.1 Primary biliary cirrhosis data

We first provide an example on small data by revisiting the primary biliary cirrhosis dataset (**?**). **?** has a more detailed account of this dataset and we refer the readers to their paper for more information. We analyze the data similarly to **?** and **?** by focusing on the 276 complete cases and all 17 standardized covariates. The number of events, d_n, was 111. Five-fold cross validation was used to select the optimal tuning parameters for Cox-LASSO, Cox-SCAD, and Cox-ALASSO. For the CoxBAR method we fixed $\xi_n = 1$ and compared both $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$. We obtain very similar estimates when using different values of ξ_n, which are omitted from this paper.

Table 4 compares the four sparse Cox regression methods with the maximum partial likelihood estimates. It is seen that our method selects fewer variables with smaller BIC scores compared to Cox-LASSO, Cox-SCAD, Cox-ALASSO.

4.2 Pediatric national trauma data bank (NTDB) data

For an application of CoxBAR regression in the large-scale sparse data setting, we will look at a subset of the National Trauma Data Bank that involves children and adolescents. This



(a) Path plot for mCox-LASSO.

(b) Path plot for CoxBAR with fixed $\xi_n = 20$.

(c) Path plot for CoxBAR with fixed $\lambda_n = \ln(n)$.

Figure 2: Path plots for mCox-LASSO and CoxBAR regression: (a) Path plot for mCox-LASSO regression, where the black dashed line represents the estimates when using cross validation to find the optimal value of the tuning parameter; (b) Path plot for CoxBAR regression with $\xi_n = 20$ and varying λ_n, where the black solid and dashed line represent estimates for $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$, respectively; (c) Path plot for CoxBAR regression with $\lambda_n = \ln(n)$ and varying ξ_n, where the black dashed line represent the estimates for CoxBAR when $\xi_n = 20$.

Table 3: (High dimensional, moderate sample size) Simulated estimation and variable selection performance of SJS-CoxBAR, SJS-LASSO, SJS-SCAD, and SJS-Adaptive LASSO (SJS-ALASSO) (SJS-CoxBAR$_1$ and SJS-CoxBAR$_2$ denote SJS-CoxBAR with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$, respectively; SSB=mean sum of squared bias; FP=mean number of false positives; FN=mean number of false negatives; MC=mean number of misclassified coefficients (MC); Each entry is based on 100 Monte Carlo samples)

n	m	p_n	Method	SSB	FP	FN	MC
300	26	1660	SJS-CoxBAR$_1$	0.12	1.00	1.16	2.16
			SJS-CoxBAR$_2$	0.13	1.11	1.13	2.24
			SJS-LASSO	0.58	20.30	0.80	21.10
			SJS-SCAD	0.94	20.30	0.80	21.10
			SJS-ALASSO	0.65	18.41	0.80	19.21
		5200	SJS-CoxBAR$_1$	0.17	1.78	1.22	3.00
			SJS-CoxBAR$_2$	0.19	2.28	1.23	3.51
			SJS-LASSO	0.58	20.22	1.10	21.32
			SJS-SCAD	0.93	19.85	1.10	20.95
			SJS-ALASSO	0.64	17.39	1.10	18.49
500	40	1660	SJS-CoxBAR$_1$	0.06	0.50	0.70	1.20
			SJS-CoxBAR$_2$	0.07	0.66	0.70	1.36
			SJS-LASSO	0.41	33.11	0.28	33.39
			SJS-SCAD	0.69	33.76	0.28	34.04
			SJS-ALASSO	0.46	29.61	0.28	29.89
		5200	SJS-CoxBAR$_1$	0.09	1.42	0.84	2.26
			SJS-CoxBAR$_2$	0.10	1.70	0.81	2.51
			SJS-LASSO	0.49	33.31	0.47	33.78
			SJS-SCAD	0.82	33.64	0.47	34.11
			SJS-ALASSO	0.56	29.61	0.47	30.08

dataset was previously analyzed by **?** as an example for efficiently massive Cox regression with LASSO (mCox-LASSO) and ridge regression to high-dimensional and massive sample size (HDMSS) data. The dataset includes 210,555 patient records of injured children under 15 that was collected over 5 years (2006 -2010). Each patient record includes 125,952 binary covariates which indicate the presence, or absence, of an attribute (ICD9 Codes, AIS codes, etc.) as well as the two-way interactions between attributes. The outcome of interest is mortality after time of injury. The data is extremely sparse, with less than 1% of the covariates being non-zero and has a censoring rate of 98%. Since the data is too large to fit other popular oracle procedures, we compare the CoxBAR method, with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ and with $\xi_n = 20$, to mCox-LASSO with cross validation. We run both models on the full dataset and record the partial log-likelihood, number of non-zero covariates, BIC score, and computing time in Table 5. As shown in Table 5, the two CoxBAR methods selects fewer covariates than mCox-LASSO with a four or five-fold speedup in computing time. Further, the BIC score for the two CoxBAR methods are approximately 1500 or 2000 less than that of the mCox-LASSO method.

Table 4: (PBC data) Selected variables, estimated regression coefficients, and BIC scores of Cox-LASSO, Cox-SCAD, Cox-ALASSO, and CoxBAR regression for the PBC data. (CoxBAR$_1$ and CoxBAR$_2$ denote CoxBAR with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$, respectively; asterisk denotes non-zero values < 0.01)

	MPLE	Cox-LASSO	Cox-SCAD	Cox-ALASSO	CoxBAR$_1$	CoxBAR$_2$
trt	-0.12	-	-	-	-	-
age	0.03	0.01	0.01	0.01	-	0.02
sex	-0.37	-	-	-	-	-
ascites	0.09	0.11	0.13	-	-	-
hepato	0.03	-	-	-	-	-
spiders	0.10	-	-	-	-	-
edema	1.01	0.63	0.30	0.62	-	0.69
bili	0.08	0.08	0.12	0.10	0.11	0.10
chol	0.00*	-	-	-	-	-
albumin	-0.74	-0.53	-0.38	-0.50	-0.88	-0.69
copper	0.00*	0.00*	0.00*	0.00*	0.00*	0.00*
alk.phos	0.00*	-	-	-	-	-
ast	0.00*	0.00*	-	-	-	
trig	-0.00*	-	-	-	-	-
platelet	0.00*	-	-	-	-	-
protime	0.23	0.11	0.04	0.05	-	-
stage	0.45	0.25	0.22	0.22	0.47	0.41
λ_n	-	0.07	0.10	0.30	4.93	4.02
ξ_n	-	-	-	0.42	1.00	1.00
# Selected	17	9	8	7	4	6
BIC Score	1028.21	1002.52	1013.48	999.18	984.31	981.94

4.3 Diffuse Large-B-Cell lymphoma data

For an application of SJS-CoxBAR in the ultrahigh dimensional setting, we analyze the microarray diffuse large-B-cell lymphoma data (?) that was also analyzed in ?. The dataset consists of $n = 240$ patients and $p = 7399$ cDNA microarray expressions. The censoring rate was around 43%. Similar to ? we remove the 5 patients with observed survival times close to 0, standardize the genes, and use sure joint screening to reduce the number of genes from 7399 to 43. We then apply our CoxBAR method with $\xi_n = 1$ and both $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$, to the same 43 genes and compare our method to what was reported in Table 3.5 of ?. These results are provided in Table 6. We see that all four methods have comparable BIC scores. However, using SJS-CoxBAR with $\lambda_n = \ln(d_n)$ only includes 8 genes whereas SJS-SCAD includes 30 genes with a negligible difference in BIC score.

Table 5: (Pediatric NTDB data) Comparison of mCox-LASSO and CoxBAR regression for the pediatric NTDB data. (CoxBAR$_1$ and CoxBAR$_2$ denote CoxBAR with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ respectively)

Method	Runtime (in hours)	Log-Likelihood	# Non-Zero	BIC Score
mCox-LASSO	76	-32408.73	253	67918.61
CoxBAR$_1$	16	-32789.64	57	66377.96
CoxBAR$_2$	19	-32475.10	84	65979.83

Table 6: (BLCA data) Comparison of SJS-LASSO, SJS-SCAD, SJS-ALASSO, and SJS-CoxBAR regression for the BLCA data. (CoxBAR$_1$ and CoxBAR$_2$ denote CoxBAR with $\lambda_n = \ln(n)$ and $\lambda_n = \ln(d_n)$ respectively; SJS-LASSO and SJS-SCAD results are from **?**)

Method	Log-Likelihood	# Selected	BIC Score
SJS-SCAD	-546.1902	30	1256.168
SJS-LASSO	-542.9862	36	1282.518
SJS-CoxBAR$_1$	-624.1901	5	1275.678
SJS-CoxBAR$_2$	-607.2283	8	1258.133

5 Discussion

We have presented a new scalable sparse Cox regression method and adapted it to both sHDMSS and ultrahigh dimensional right-censored survival data. The concept of CoxBAR regression is based on an iterative Cox ridge regression where the penalty is adaptively reweighted; which not only results in an estimator that inherits the nice properties of both L_0 and L_2-penalized Cox regression while avoiding their pitfalls, but also allows us to take advantage of existing efficient algorithms and software (**?**) for large scale L_2-penalized Cox regression. We have shown that CoxBAR regression enjoys the oracle properties and a grouping property for highly correlated covariates. Our numerical studies showed that compared to some competing methods, the CoxBAR method generally has less false positives and misclassifications, with a small trade-off of slightly increased false negatives for variable selection, and is as good or better for estimation in both low and high dimensional settings. In addition, as illustrated in Section 3 that the CoxBAR estimator can avoid costly data driven tuning parameter selection with pre-specified values, which highlights a huge advantage for large scale studies. For instance, it demonstrated a 20-fold speedup over the L_1-penalized Cox regression on a simulated sHDMSS survival data in Section 3.3. Furthermore, our developed theory for CoxBAR guarantees that it can be combined with a sure screening procedure to obtain an oracle two-stage sparse regression method for high (or ultrahigh) dimensional small sample size data. Finally, our L_0-based CoxBAR method and theory can be easily extended to an L_d-based CoxBAR method for any $d \in [0, 1]$, by replacing $(\hat{\beta}_j^{(k-1)})^2$ with $|\hat{\beta}_j^{(k-1)}|^{2-d}$ in (4). We have observed from simulations that as d increases towards 1, the resulting estimator becomes less sparse, and the number of false positives and misclassifications as well as estimation bias tend to increase especially for larger p_n, while the number of false negatives tends to decrease.

Acknowledgement

Gang Li's research was supported in part by National Institute of Health Grants P30 CA- 16042, UL1TR000124-02, and P01AT003960.

A Appendix

A.1 Regularity Conditions for Theorem 1.

Due to **?**, we can rewrite the log-partial likelihood as

$$l_n(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[\int_0^1 \boldsymbol{\beta}^T \mathbf{x}_i(s) - \ln\{S^{(0)}(\boldsymbol{\beta}, s)\} \right] dN_i(s), \tag{16}$$

where

$$S^{(k)}(\boldsymbol{\beta}, s) = \frac{1}{n} \sum_{i=1}^{n} Y_i(s) \mathbf{x}_i(s)^{\otimes k} \exp\{\boldsymbol{\beta}^T \mathbf{x}_i(s)\},$$

for $k = 0, 1, 2$ and where $\mathbf{x}^{\otimes k} = 1, \mathbf{x}, \mathbf{x}\mathbf{x}^T$ for $k = 0, 1, 2$, respectively. Further, let the corresponding martingale for $N_i(t)$ be defined as

$$M_i(t) = N_i(t) - \int_0^1 h_0(t) \exp\{\boldsymbol{\beta}_0^T \mathbf{x}_i(t)\} dt.$$

Define $\|\cdot\|$ as the Euclidean norm (for vectors) and spectral norm (for matrices). The regularity conditions for the oracle properties of the CoxBAR estimator are as follows:

(C1) $\int_0^1 h_0(t) dt < \infty$;

(C2) There exists some compact neighborhood, $\mathcal{B}_0$, of the true value $\boldsymbol{\beta}_0$ such that: For $k = 0, 1, 2$, there exists a bounded scalar, vector, and matrix function $s^{(k)}(\boldsymbol{\beta}; t)$ defined on $\mathcal{B}_0 \times [0, 1]$ that are absolutely continuous for $t \in [0, 1]$ for $\boldsymbol{\beta} \in \mathcal{B}_0$ such that

$$\sup_{t \in [0,1], \boldsymbol{\beta} \in \mathcal{B}_0} \left\| S^{(k)}(\boldsymbol{\beta}; t) - s^{(k)}(\boldsymbol{\beta}; t) \right\| \to \mathbf{0}$$

in probability and, furthermore, $s^{(0)}(\boldsymbol{\beta}; t)$ is bounded away from zero on $\mathcal{B}_0 \times [0, 1]$;

(C3) Using the notation defined in (C2), define $e(\boldsymbol{\beta}, t) = s^{(1)}(\boldsymbol{\beta}, t)/s^{(0)}(\boldsymbol{\beta}, t)$, $v(\boldsymbol{\beta}, t) = s^{(2)}(\boldsymbol{\beta}, t)/s^{(0)}(\boldsymbol{\beta}, t) - e(\boldsymbol{\beta}, t)^{\otimes 2}$ and let

$$I(\boldsymbol{\beta}) = \int_0^1 v(\boldsymbol{\beta}, t) s^{(0)}(\boldsymbol{\beta}_0, t) h_0(t) dt.$$

Let $H_n(\boldsymbol{\beta}) = -n^{-1} \ddot{l}_n(\boldsymbol{\beta})$, where $\ddot{l}_n(\boldsymbol{\beta})$ is the second derivative of $l_n(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$. Then there exists some compact neighborhood, $\mathcal{B}_0$, of the true value $\boldsymbol{\beta}_0$, such that

$$\sup_{\boldsymbol{\beta} \in \mathcal{B}_0} \|H_n(\boldsymbol{\beta}) - I(\boldsymbol{\beta}_0)\| \xrightarrow{a.s.} \mathbf{0}, \tag{17}$$

for some positive-definite $p_n \times p_n$ matrix $I(\boldsymbol{\beta}_0)$. Assume further the existence of some constant $C > 1$ such that $C^{-1} < \rho_{\min}(I(\boldsymbol{\beta}_0)) \le \rho_{\max}(I(\boldsymbol{\beta}_0)) < C$, for sufficiently large n, and where $\rho_{\min}(Q)$ and $\rho_{\max}(Q)$ represent the smallest and largest eigenvalues of the matrix Q respectively;

(C4) Let $D_i = \int_0^1 \{\mathbf{x}_i(t) - e(\boldsymbol{\beta}_0;t)\}\, dM_i(t)$. Then for all $1 \le j,\ l \le p_n$, there exists a constant K such that

$$\sup_{1\le i\le n} E(D_{ij}^2 D_{il}^2) < K < \infty,$$

where D_{ij} is the j^{th} element of D_i;

(C5) As $n \to \infty$ we have that $p_n^2 q/\sqrt{n} \to 0$, $\xi_n/\sqrt{n} \to 0$, $\lambda_n/\sqrt{n} \to 0$, $\lambda_n^2/(p_n\sqrt{n}) \to \infty$, and $\lambda_n\sqrt{q}/\sqrt{n} \to 0$;

(C6) There exists constants $0 < a_0 < a_1 < \infty$ such that for all $j \in \{1,\dots,q\}$, $|\beta_{0j}| \in [a_0, a_1]$.

? have also required conditions (C1) - (C5) for diverging p_n and showed that given

$$\hat{\boldsymbol{\beta}}_{pen} = \arg\min_{\boldsymbol{\beta}} \left\{ -l_n(\boldsymbol{\beta}) + n\sum_{j=1}^{p_n} p_{\lambda_n}(\beta_j) \right\},$$

where $p_{\lambda_n}(\beta)$ is a non-negative penalty function, then under certain conditions for $p_{\lambda_n}(\beta_j)$, $||\hat{\boldsymbol{\beta}}_{pen} - \boldsymbol{\beta}_0|| = O_p(\sqrt{p_n/n})$. Condition (C6) is needed to prove that for diverging p_n and with $\xi_n/\sqrt{n} \to 0$, $||\hat{\boldsymbol{\beta}}_{ridge} - \boldsymbol{\beta}_0|| = O_p(\sqrt{p_n/n})$, where $\hat{\boldsymbol{\beta}}_{ridge}$ is the Cox ridge estimator defined in Equation (3).

A.2 Proof of Theorem 1.

Lemma 1 (Consistency of Ridge Estimator) *Let $\hat{\boldsymbol{\beta}}_{ridge}$ be the Cox ridge estimator defined in Equation (3). Given Conditions (C1) - (C6),*

$$||\hat{\boldsymbol{\beta}}_{ridge} - \boldsymbol{\beta}_0|| = O_p(\sqrt{p_n/n}). \tag{18}$$

The proof follows from Theorem 1 of **?**. The condition that $p_n^4/n \to 0$ from **?** can be easily derived from Condition (C5) where we assume $p_n^2 q/\sqrt{n} \to 0$. We need to further show that $a_n \to 0$ and $b_n \to 0$ where

$$a_n = \max_{1\le j\le q}\{|p'_{\xi_n}(\beta_{0j})| : \beta_{0j} \ne 0\},$$
$$b_n = \max_{1\le j\le q}\{|p''_{\xi_n}(\beta_{0j})| : \beta_{0j} \ne 0\}.$$

Following the notation of **?**, for ridge regression we can see that $p_{\xi_n}(\beta_{0j}) = (\xi_n/n)\beta_{0j}^2$ for $j = 1,\dots,p_n$. Thus $p'_{\xi_n}(\beta_{0j}) = (2\xi_n/n)\beta_{0j}$ and $p''_{\xi_n}(\beta_{0j}) = 2\xi_n/n$. From Conditions (C5) and (C6) we have that $\xi_n/\sqrt{n} \to 0$ and we have that $\beta_{0j} \in [a_0, a_1]$ for all $j \in \{1,\dots,q\}$. Therefore,

$$a_n = \max_{1\le j\le q}\{|p'_{\xi_n}(\beta_{0j})| : \beta_{0j} \ne 0\} \le \frac{2\xi_n a_1}{n} = \frac{\xi_n}{\sqrt{n}}\frac{2a_1}{\sqrt{n}} = o(n^{-1/2}),$$

and

$$b_n = \max_{1\le j\le q}\{|p''_{\xi_n}(\beta_{0j})| : \beta_{0j} \ne 0\} \le \frac{2\xi_n}{n} = \frac{\xi_n}{\sqrt{n}}\frac{2}{\sqrt{n}} = o(n^{-1/2}).$$

Therefore $a_n \to 0$ and $b_n \to 0$. Thus from **?**, we have that $||\hat{\boldsymbol{\beta}}_{ridge} - \boldsymbol{\beta}_0|| = O_p(\sqrt{p_n/n})$.

For any $\boldsymbol{\beta}$, define $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T\right)^T$ where $\boldsymbol{\beta}_1^T$ and $\boldsymbol{\beta}_2^T$ correspond to the first q and remaining $p_n - q$ components of $\boldsymbol{\beta}$, respectively. Let

$$Q_n(\boldsymbol{\theta}; \boldsymbol{\beta}) = -2l_n(\boldsymbol{\theta}) + \lambda_n \boldsymbol{\theta}^T D(\boldsymbol{\beta})\boldsymbol{\theta}, \tag{19}$$

where $D(\boldsymbol{\beta}) = diag(\beta_1^{-2}, \beta_2^{-2}, \ldots, \beta_q^{-2}, \beta_{q+1}^{-2}, \ldots, \beta_{p_n}^{-2})$. For simplicity, let us define $Q_n(\boldsymbol{\theta}; \boldsymbol{\beta})$ as $Q_n(\boldsymbol{\theta})$. Let $\dot{Q}_n(\boldsymbol{\theta})$ and $\ddot{Q}_n(\boldsymbol{\theta})$ be the first and second derivatives of $Q(\boldsymbol{\theta})$, respectively. Therefore

$$\dot{Q}_n(\boldsymbol{\theta}) = -2\dot{l}_n(\boldsymbol{\theta}) + 2\lambda_n D(\boldsymbol{\beta})\boldsymbol{\theta}, \tag{20}$$

$$\ddot{Q}_n(\boldsymbol{\theta}) = -2\ddot{l}_n(\boldsymbol{\theta}) + 2\lambda_n D(\boldsymbol{\beta}). \tag{21}$$

Lemma 2 *Suppose* $g(\boldsymbol{\beta}) = \left(g_1(\boldsymbol{\beta})^T, g_2(\boldsymbol{\beta})^T\right)^T$ *is a solution to* $\dot{Q}_n(\boldsymbol{\theta}) = \mathbf{0}$. *Furthermore, define* $\mathcal{H}_n \equiv \{\boldsymbol{\beta} = \left(\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T\right)^T : |\boldsymbol{\beta}_1| = (|\beta_1|, \ldots, |\beta_q|)^T \in [1/M_0, M_0]^q, \|\boldsymbol{\beta}_2\| \leq \delta_n\sqrt{p_n}/\sqrt{n}\}$, *where* $M_0 > 1$ *is a constant such that* $\boldsymbol{\beta}_{01} \in [1/M_0, M_0]^q$, *and* δ_n *is a sequence of positive real numbers. Suppose* $\delta_n \to \infty$ *and* $p_n\delta_n^2/\lambda_n \to 0$. *Then under the regularity conditions (C1) - (C6), with probability tending to 1, we have that:*

(a) $g(\cdot)$ *is a mapping from* $\mathcal{H}_n$ *to itself;*

(b) For some constant $C_0 > 1$,

$$\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \frac{\|g_2(\boldsymbol{\beta})\|}{\|\boldsymbol{\beta}_2\|} < \frac{1}{C_0} \tag{22}$$

By first-order Taylor expansion for $\dot{Q}(\boldsymbol{\beta})$ at $\boldsymbol{\beta}_0$ in a neighborhood $g(\boldsymbol{\beta})$ we have that,

$$\dot{Q}_n(\boldsymbol{\beta}_0) = \dot{Q}_n(g(\boldsymbol{\beta})) + \ddot{Q}_n(\boldsymbol{\beta}^*)(\boldsymbol{\beta}_0 - g(\boldsymbol{\beta})), \tag{23}$$

where $\boldsymbol{\beta}_0$ is the true parameter vector, and $\boldsymbol{\beta}^* \in [\boldsymbol{\beta}_0, g(\boldsymbol{\beta})]$. Rearranging terms,

$$\ddot{Q}_n(\boldsymbol{\beta}^*)g(\boldsymbol{\beta}) = -\dot{Q}_n(\boldsymbol{\beta}_0) + \ddot{Q}_n(\boldsymbol{\beta}^*)\boldsymbol{\beta}_0, \tag{24}$$

since $\dot{Q}_n(g(\boldsymbol{\beta})) = 0$. Using (20) and (21) we can rewrite (24) as,

$$\begin{aligned}\left\{-2\ddot{l}_n(\boldsymbol{\beta}^*) + 2\lambda_n D(\boldsymbol{\beta})\right\} g(\boldsymbol{\beta}) &= -\left\{-2\dot{l}_n(\boldsymbol{\beta}_0) + 2\lambda_n D(\boldsymbol{\beta})\boldsymbol{\beta}_0\right\} + \left\{-2\ddot{l}_n(\boldsymbol{\beta}^*) + 2\lambda_n D(\boldsymbol{\beta})\right\} \boldsymbol{\beta}_0 \\ &= 2\dot{l}_n(\boldsymbol{\beta}_0) - 2\ddot{l}_n(\boldsymbol{\beta}^*)\boldsymbol{\beta}_0.\end{aligned}$$

Hence,

$$\left\{-\frac{1}{n}\ddot{l}_n(\boldsymbol{\beta}^*) + \frac{\lambda_n}{n}D(\boldsymbol{\beta})\right\} g(\boldsymbol{\beta}) = -\frac{1}{n}\ddot{l}_n(\boldsymbol{\beta}^*)\boldsymbol{\beta}_0 + \frac{1}{n}\dot{l}_n(\boldsymbol{\beta}_0). \tag{25}$$

Define $H_n(\boldsymbol{\beta}^*) = -n^{-1}\ddot{l}_n(\boldsymbol{\beta}^*)$. By Condition (C3), $H_n(\boldsymbol{\beta}^*)^{-1}$ exists and can be partitioned into,

$$H_n(\boldsymbol{\beta}^*)^{-1} = \begin{bmatrix} A & B \\ B^T & G \end{bmatrix}$$

Therefore by pre-multiplying both sides of (25) by $H_n(\boldsymbol{\beta}^*)^{-1}$ we get

$$\{g(\boldsymbol{\beta}) - \boldsymbol{\beta}_0\} + \frac{\lambda_n}{n} H_n(\boldsymbol{\beta}^*)^{-1} D(\boldsymbol{\beta}) g(\boldsymbol{\beta}) = \frac{1}{n} H_n(\boldsymbol{\beta}^*)^{-1} \dot{l}_n(\boldsymbol{\beta}_0) \tag{26}$$

Also $D(\boldsymbol{\beta})$ can be partitioned into,

$$D(\boldsymbol{\beta}) = \left[\begin{array}{ll} D_1(\boldsymbol{\beta}_1) & \mathbf{0} \\ \mathbf{0}^T & D_2(\boldsymbol{\beta}_2) \end{array} \right]$$

where $D_1(\boldsymbol{\beta}_1) = \text{diag}(\beta_1^{-2}, ..., \beta_q^{-2})$ and $D_2(\boldsymbol{\beta}_2) = \text{diag}(\beta_{q+1}^{-2}, ..., \beta_{p_n}^{-2})$.
Since $\boldsymbol{\beta}_0 = (\boldsymbol{\beta}_{01}^T, \mathbf{0}^T)$, (26) can be re-written as

$$\left(\begin{array}{c} g_1(\boldsymbol{\beta}) - \boldsymbol{\beta}_{01} \\ g_2(\boldsymbol{\beta}) \end{array} \right) + \frac{\lambda_n}{n} \left(\begin{array}{c} AD_1(\boldsymbol{\beta}_1) g_1(\boldsymbol{\beta}) + BD_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \\ B^T D_1(\boldsymbol{\beta}_1) g_1(\boldsymbol{\beta}) + GD_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \end{array} \right) = \frac{1}{n} H_n(\boldsymbol{\beta}^*)^{-1} \dot{l}_n(\boldsymbol{\beta}_0). \tag{27}$$

By Theorem 1 of **?**, Conditions (C1) - (C5) guarantee that $n^{-1}\dot{l}_n(\boldsymbol{\beta}_0) = O_p(\sqrt{p_n/n})$ and by Condition (C3), we have that $n^{-1}H_n(\boldsymbol{\beta}^*)^{-1}\dot{l}_n(\boldsymbol{\beta}_0) = O_p(\sqrt{p_n/n})$. Therefore,

$$\sup_{\boldsymbol{\beta} \in \mathcal{H}_n} \left\| g_2(\boldsymbol{\beta}) + \frac{\lambda_n}{n} B^T D_1(\boldsymbol{\beta}_1) g_1(\boldsymbol{\beta}) + \frac{\lambda_n}{n} GD_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\| = O_p(\sqrt{p_n/n}). \tag{28}$$

Note that $|\boldsymbol{\beta}_1| \in [1/M_0, M_0]^q$ and $||g_1(\boldsymbol{\beta})|| \leq ||g(\boldsymbol{\beta})|| \leq ||\hat{\boldsymbol{\beta}}_{mle}|| = O_p(\sqrt{p_n})$. Furthermore,

$$\left\|BB^T\right\| - \left\|A^2\right\| \leq \left\|A^2 + BB^T\right\| \leq \left\|H_n(\boldsymbol{\beta}^*)^{-2}\right\| < C^2.$$

So $\left\|B^T\right\| \leq \sqrt{2}C$ and $\|B\| \leq \sqrt{2C}$, and therefore by Condition (C3),

$$\begin{aligned} \sup_{\boldsymbol{\beta} \in \mathcal{H}_n} \left\| \frac{\lambda_n}{n} B^T D_1(\boldsymbol{\beta}_1) g_1(\boldsymbol{\beta}) \right\| &\leq \frac{\lambda_n}{n} \sup_{\boldsymbol{\beta} \in \mathcal{H}_n} \left\|B^T\right\| \|D_1(\boldsymbol{\beta}_1)\| \|g_1(\boldsymbol{\beta})\| && (29) \\ &\leq \frac{\lambda_n}{n} \sqrt{2} C M_0^2 \sup_{\boldsymbol{\beta} \in \mathcal{H}_n} \|g_1(\boldsymbol{\beta})\| && (30) \\ &= o_p(\sqrt{p_n/n}). && (31) \end{aligned}$$

As a result, we can rewrite (28) as

$$\sup_{\boldsymbol{\beta} \in \mathcal{H}_n} \left\| g_2(\boldsymbol{\beta}) + \frac{\lambda_n}{n} GD_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\| = O_p(\sqrt{p_n/n}). \tag{32}$$

Since G is positive definite, by the singular value decomposition, $G = \sum_{i=1}^{p_n - q} r_{2i} \mathbf{u}_{2i} \mathbf{u}_{2i}^T$, where r_{2i} and $\mathbf{u}_{2i}$ are the eigenvalues and eigenvectors of G, respectively. Since Condition

(C3) assumes that for all $i = 1, \ldots, p_n - q$, $r_{2i} \in (1/C, C)$ for some $C > 1$, we have that

$$\begin{aligned}
\frac{\lambda_n}{n} \left\| G D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\| &= \frac{\lambda_n}{n} \left\| \sum_{i=1}^{p_n - q} r_{2i} \mathbf{u}_{2i} \mathbf{u}_{2i}^T D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\| \\
&= \frac{\lambda_n}{n} \left(\sum_{i=1}^{p_n - q} r_{2i}^2 \left\| \mathbf{u}_{2i}^T D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\|^2 \right)^{1/2} \\
&\geq \frac{\lambda_n}{n} \frac{1}{C} \left(\sum_{i=1}^{p_n - q} \left\| \mathbf{u}_{2i}^T D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\|^2 \right)^{1/2} \\
&= \frac{1}{C} \left\| \frac{\lambda_n}{n} D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\|. \qquad (33)
\end{aligned}$$

Therefore, with probability tending to one,

$$\frac{1}{C} \left\| \frac{\lambda_n}{n} D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\| - \| g_2(\boldsymbol{\beta}) \| \leq \delta_n \sqrt{p_n/n}. \qquad (34)$$

Let $\mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} = (g_2(\beta_{q+1})/\beta_{q+1}, \ldots, g_2(\beta_{p_n})/\beta_{p_n})^T$. So $D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) = D_2(\boldsymbol{\beta}_2)^{1/2} \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2}$ and $g_2(\boldsymbol{\beta}) = D_2(\boldsymbol{\beta})^{-1/2} \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2}$. Since $||\boldsymbol{\beta}_2|| \leq \delta_n \sqrt{p_n/n}$, we have that

$$\frac{1}{C} \left\| \frac{\lambda_n}{n} D_2(\boldsymbol{\beta}_2) g_2(\boldsymbol{\beta}) \right\| = \frac{1}{C} \frac{\lambda_n}{n} \left\| D_2(\boldsymbol{\beta}_2)^{1/2} \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| \geq \frac{1}{C} \frac{\lambda_n}{n} \frac{\sqrt{n}}{\delta_n \sqrt{p_n}} \left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\|, \qquad (35)$$

and

$$\| g_2(\boldsymbol{\beta}) \| = \left\| D_2(\boldsymbol{\beta}_2)^{-1/2} \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| \leq \frac{\delta_n \sqrt{p_n}}{\sqrt{n}} \left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\|, \qquad (36)$$

with probability tending to one. Hence it follows from (33), (35), and (36), that with probability tending to one,

$$\frac{1}{C} \frac{\lambda_n}{n} \frac{\sqrt{n}}{\delta_n \sqrt{p_n}} \left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| - \frac{\delta_n \sqrt{p_n}}{\sqrt{n}} \left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| \leq \delta_n \sqrt{p_n/n}. \qquad (37)$$

Therefore with probability tending to one,

$$\left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| \leq \frac{1}{\lambda_n/(C p_n \delta_n^2) - 1} < \frac{1}{C_0}, \qquad (38)$$

for some constant $C_0 > 1$ provided that $\lambda_n/(p_n \delta_n^2) \to \infty$ as $n \to \infty$. Now from (38), we have that

$$\| g_2(\boldsymbol{\beta}) \| \leq \left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| \max_{q_n+1 \leq j \leq p_n} g_2(\beta_j) \leq \left\| \mathbf{m}_{g_2(\boldsymbol{\beta})/\boldsymbol{\beta}_2} \right\| \| \boldsymbol{\beta}_2 \| \leq \frac{1}{C_0} \| \boldsymbol{\beta}_2 \|, \qquad (39)$$

with probability tending to one. Thus

$$\Pr \left(\sup_{\boldsymbol{\beta} \in \mathcal{H}_n} \frac{\| g_2(\boldsymbol{\beta}) \|}{\| \boldsymbol{\beta}_2 \|} < \frac{1}{C_0} \right) \to 1 \quad \text{as } n \to \infty \qquad (40)$$

and (b) is proven. Since $C_0 > 1$ we also have that, with probability tending to one, $\|g_2(\boldsymbol{\beta})\| \leq \delta_n\sqrt{p_n/n}$. Hence for any $|g_1(\boldsymbol{\beta})| \in [1/M_0, M_0]^q$, $g(\boldsymbol{\beta}) = \big(g_1(\boldsymbol{\beta})^T, g_2(\boldsymbol{\beta})^T\big)^T$ is a mapping of $g_2(\boldsymbol{\beta})$ to itself and shrinks the norm of $g_2(\boldsymbol{\beta})$ towards 0 at each iterative step. To finish the proof of (i) we need to show that $|g_1(\boldsymbol{\beta})| \in [1/M_0, M_0]^q$ with probability tending to 1. Likewise with (28) and (29), we know that

$$\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \left\|(g_1(\boldsymbol{\beta}) - \boldsymbol{\beta}_{01}) + \frac{\lambda_n}{n}AD_1(\boldsymbol{\beta}_1)g_1(\boldsymbol{\beta}) + \frac{\lambda_n}{n}BD_2(\boldsymbol{\beta}_2)g_2(\boldsymbol{\beta})\right\| = O_p(\sqrt{p_n/n}), \qquad (41)$$

and similarly

$$\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \left\|\frac{\lambda_n}{n}AD_1(\boldsymbol{\beta}_1)g_1(\boldsymbol{\beta})\right\| = o_p(\sqrt{p_n/n}), \qquad (42)$$

since $|\boldsymbol{\beta}_1| \in [1/M_0, M_0]^q$ and $||g_1(\boldsymbol{\beta})|| < ||\hat{\boldsymbol{\beta}}_{mle}|| = O_p(\sqrt{p_n})$. From (34) and Condition (C3) we have that

$$\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \left\|\frac{\lambda_n}{n}BD_2(\boldsymbol{\beta}_2)g_2(\boldsymbol{\beta})\right\| \leq \frac{\lambda_n}{n}\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \|D_2(\boldsymbol{\beta}_2)g_2(\boldsymbol{\beta})\| \cdot \|B\| \leq 2\sqrt{2}C^2\delta_n\sqrt{p_n/n}, \qquad (43)$$

with probability tending to one. Therefore,

$$\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \|g_1(\boldsymbol{\beta}) - \boldsymbol{\beta}_{01}\| \leq \frac{(2\sqrt{2}C^2+1)\delta_n\sqrt{p_n}}{\sqrt{n}}, \qquad (44)$$

with probability tending to one. By assumption $\delta_n\sqrt{p_n/n} \to 0$ as $n \to \infty$, and hence for all $\epsilon > 0$, $\Pr(\sup_{\boldsymbol{\beta}\in\mathcal{H}_n} \|g_1(\boldsymbol{\beta}) - \boldsymbol{\beta}_{01}\| < \epsilon) \to 1$. As a consequence, since $|\boldsymbol{\beta}_{01}| \in [1/M_0, M_0]^q$, we have that $|g_1(\boldsymbol{\beta})| \in [1/M_0, M_0]^q$ for sufficiently large n. Since $|g_1(\boldsymbol{\beta})| \in [1/M_0, M_0]^q$ and $\|g_2(\boldsymbol{\beta})\| \leq \delta_n/\sqrt{n}$ with probability tending to one, we have shown that

$$\Pr(g(\boldsymbol{\beta}) \in \mathcal{H}_n) \to 1, \quad \text{as } n \to \infty, \qquad (45)$$

and hence $g(\cdot)$ is a mapping from $\mathcal{H}_n$ to itself. This completes Lemma 2.

Remark 3 *Since $\boldsymbol{\beta}_{02} = \mathbf{0}$, we can now express the objective function of this reduced model as*

$$Q_{n1}(\boldsymbol{\theta}_1) = -2l_{n1}(\boldsymbol{\theta}_1) + \lambda_n\boldsymbol{\theta}_1^T D_1(\boldsymbol{\beta}_1)\boldsymbol{\theta}_1, \qquad (46)$$

which is similar to that of (19).

Lemma 3 *Let $f(\boldsymbol{\beta}_1)$ be a solution to $\dot{Q}_{n1}(\boldsymbol{\theta}_1) = 0$. Provided that Conditions (C1) - (C6) from are satisfied, then with probability tending to one:*

(a) $f(\boldsymbol{\beta}_1)$ is a contraction mapping from $[1/M_0, M_0]^q$ to itself;

(b) $\sqrt{n}H_{n1}(\boldsymbol{\beta}_1^*)^{1/2}(\hat{\boldsymbol{\beta}}_1^\circ - \boldsymbol{\beta}_{01}) \xrightarrow{D} N(\mathbf{0}, I_q)$, where $\hat{\boldsymbol{\beta}}_1^\circ$ is the unique fixed point of $f(\boldsymbol{\beta}_1)$.

First we want to show that $f(\cdot)$ is a mapping from $[1/M_0, M_0]^q$ to itself with probability tending to one. Again through a first order Taylor expansion, we have that

$$(f(\boldsymbol{\beta}_1)-\boldsymbol{\beta}_{01})+\frac{\lambda_n}{n}H_{n1}(\boldsymbol{\beta}_1^*)^{-1}D_1(\boldsymbol{\beta}_1)f(\boldsymbol{\beta}_1)=\frac{1}{n}H_{n1}(\boldsymbol{\beta}_1^*)^{-1}\dot{l}_{n1}(\boldsymbol{\beta}_{01}), \tag{47}$$

where $H_{n1}(\boldsymbol{\beta}_1^*)=-n^{-1}\ddot{l}_{n1}(\boldsymbol{\beta}_1^*)$ exists and is invertible and for $\boldsymbol{\beta}_1^*\in[\boldsymbol{\beta}_{01}, f(\boldsymbol{\beta}_1)]$.
Since $n^{-1}\dot{l}_{n1}(\boldsymbol{\beta}_{01})=O_p(\sqrt{q/n})$, we have that

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\left\|f(\boldsymbol{\beta}_1)-\boldsymbol{\beta}_{01}+\frac{\lambda_n}{n}H_{n1}(\boldsymbol{\beta}_1^*)^{-1}D_1(\boldsymbol{\beta}_1)f(\boldsymbol{\beta}_1)\right\|=O_p(\sqrt{q/n}). \tag{48}$$

From Condition (C3) and that $||f(\boldsymbol{\beta}_1)||\le||\hat{\boldsymbol{\beta}}_{mle}||=O_p(\sqrt{q})$, we have

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\left\|\frac{\lambda_n}{n}H_{n1}(\boldsymbol{\beta}_1^*)^{-1}D_1(\boldsymbol{\beta}_1)f(\boldsymbol{\beta}_1)\right\|=o_p(\sqrt{q/n}). \tag{49}$$

Therefore, with probability tending to one

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\|f(\boldsymbol{\beta}_1)-\boldsymbol{\beta}_{01}\|\le\delta_n\sqrt{q/n}. \tag{50}$$

Since it is assumed that $\delta_n\sqrt{q/n}\to 0$ as $n\to\infty$, we get that

$$\Pr(f(\boldsymbol{\beta}_1)\in[1/M_0,M_0]^q)\to 1 \tag{51}$$

as $n\to\infty$. Hence $f(\cdot)$ is a mapping from the region $[1/M_0, M_0]^q$ to itself. To prove that $f(\cdot)$ is a contraction mapping we need to further show that

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\left\|\dot{f}(\boldsymbol{\beta}_1)\right\|=o_p(1). \tag{52}$$

Since $f(\boldsymbol{\beta}_1)$ is a solution to $\dot{Q}_{n1}(\boldsymbol{\theta}_1)=0$, we have that

$$-\frac{1}{n}\dot{l}_{n1}(f(\boldsymbol{\beta}_1))=-\frac{\lambda_n}{n}D_1(\boldsymbol{\beta}_1)f(\boldsymbol{\beta}_1). \tag{53}$$

Taking the derivative of (53) with respect to $\boldsymbol{\beta}_1^T$ and rearranging terms, we have that

$$\left\{H_{n1}(f(\boldsymbol{\beta}_1))+\frac{\lambda_n}{n}D_1(\boldsymbol{\beta}_1)\right\}\dot{f}(\boldsymbol{\beta}_1)=\frac{2\lambda_n}{n}diag\{f_1(\boldsymbol{\beta}_1)/\beta_1^3,\ldots,f_q(\boldsymbol{\beta}_1)/\beta_q^3\}f(\boldsymbol{\beta}_1).$$

Since $\lambda_n/\sqrt{n}\to 0$ as $n\to\infty$,

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\frac{2\lambda_n}{n}\left\|diag\{f_1(\boldsymbol{\beta}_1)/\beta_1^3,\ldots,f_q(\boldsymbol{\beta}_1)/\beta_q^3\}\right\|=o_p(1), \tag{54}$$

and as a consequence,

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\left\|\left\{H_{n1}(f(\boldsymbol{\beta}_1))+\frac{\lambda_n}{n}D_1(\boldsymbol{\beta}_1)\right\}\dot{f}(\boldsymbol{\beta}_1)\right\|=o_p(1). \tag{55}$$

For sufficiently large n and (C3), we have that

$$\left\|H_{n1}(f(\boldsymbol{\beta}_1))\dot{f}(\boldsymbol{\beta}_1)\right\| \geq \frac{1}{C}\left\|\dot{f}(\boldsymbol{\beta}_1)\right\|, \tag{56}$$

and

$$\frac{\lambda_n}{n}\left\|D_1(\boldsymbol{\beta}_1)\dot{f}(\boldsymbol{\beta}_1)\right\| \geq \frac{\lambda_n}{n}\frac{1}{M_0^2}\left\|\dot{f}(\boldsymbol{\beta}_1)\right\|. \tag{57}$$

Therefore

$$\sup_{|\boldsymbol{\beta}_1|\in[1/M_0,M_0]^q}\left\|\dot{f}(\boldsymbol{\beta}_1)\right\| = o_p(1), \tag{58}$$

since $\lambda_n/n \to 0$ as $n \to \infty$. Thus, from (58), we have showed that $f(\cdot)$ is a contraction mapping from $[1/M_0, M_0]^q$ to itself with probability tending to one. Therefore, as $n \to \infty$ there exists a unique fixed solution $\hat{\boldsymbol{\beta}}_1^\circ$ for $f(\boldsymbol{\beta}_1)$ with probability tending to one.

We can algebraically manipulate (47) so that we may obtain

$$f(\boldsymbol{\beta}_1) = \left\{H_{n1}(\boldsymbol{\beta}_1^*) + \frac{\lambda_n}{n}D_1(\boldsymbol{\beta}_1)\right\}^{-1}\left\{H_{n1}(\boldsymbol{\beta}_1^*)\boldsymbol{\beta}_{01} + \frac{1}{n}\dot{l}_{n1}(\boldsymbol{\beta}_{01})\right\}. \tag{59}$$

Now,

$$\begin{aligned}\sqrt{n}H_{n1}(\boldsymbol{\beta}_1^*)^{1/2}(\hat{\boldsymbol{\beta}}_1^\circ - \boldsymbol{\beta}_{01}) &= \sqrt{n}H_{n1}(\boldsymbol{\beta}_1^*)^{1/2}\left[\left\{H_{n1}(\boldsymbol{\beta}_1^*) + \frac{\lambda_n}{n}D_1(\hat{\boldsymbol{\beta}}_1^\circ)\right\}^{-1}H_{n1}(\boldsymbol{\beta}_1^*) - I_q\right]\boldsymbol{\beta}_{01}\\ &\quad + \sqrt{n}H_{n1}(\boldsymbol{\beta}_1^*)^{1/2}\left[\left\{H_{n1}(\boldsymbol{\beta}_1^*) + \frac{\lambda_n}{n}D_1(\hat{\boldsymbol{\beta}}_1^\circ)\right\}^{-1}\frac{1}{n}\dot{l}_{n1}(\boldsymbol{\beta}_{01})\right]\\ &= I_1 + I_2. \end{aligned} \tag{60}$$

For conformable invertible matrices, Φ and Ψ, we have that

$$(\Phi+\Psi)^{-1} = \Phi^{-1} - \Phi^{-1}\Psi(\Phi+\Psi)^{-1}.$$

Therefore we can rewrite I_1 as

$$\begin{aligned} I_1 &= \sqrt{n}H_{n1}(\boldsymbol{\beta}_1^*)^{1/2}\left[\left\{H_{n1}(\boldsymbol{\beta}_1^*) + \frac{\lambda_n}{n}D_1(\hat{\boldsymbol{\beta}}_1^\circ)\right\}^{-1}H_{n1}(\boldsymbol{\beta}_1^*) - I_q\right]\boldsymbol{\beta}_{01}\\ &= -\frac{\lambda_n}{\sqrt{n}}H_{n1}(\boldsymbol{\beta}_1^*)^{-1/2}D_1(\hat{\boldsymbol{\beta}}_1^\circ)\left\{H_{n1}(\boldsymbol{\beta}_1^*) + \frac{\lambda_n}{n}D_1(\hat{\boldsymbol{\beta}}_1^\circ)\right\}^{-1}H_{n1}(\boldsymbol{\beta}_1^*)\boldsymbol{\beta}_{01} \end{aligned} \tag{61}$$

Using Condition (C3) and the assumption that $\lambda_n\sqrt{q}/\sqrt{n} \to 0$ from Condition (C6), we have that

$$\|I_1\| \leq \frac{M_0^2\lambda_n}{\sqrt{n}}\left\|H_{n1}(\boldsymbol{\beta}_1^*)^{-1/2}\right\|\|\boldsymbol{\beta}_{01}\| = O_p(\lambda_n\sqrt{q}/\sqrt{n}) \to 0.$$

Similarly, we can rewrite I_2 as

$$\begin{aligned}
I_2 &= \sqrt{n} H_{n1}(\boldsymbol{\beta}_1^*)^{1/2} \left[\left\{ H_{n1}(\boldsymbol{\beta}_1^*) + \frac{\lambda_n}{n} D_1(\hat{\boldsymbol{\beta}}_1^\circ) \right\}^{-1} \frac{1}{n} \dot{l}_{n1}(\boldsymbol{\beta}_{01}) \right] \\
&= H_{n1}(\boldsymbol{\beta}_1^*)^{-1/2} \frac{1}{\sqrt{n}} \dot{l}_{n1}(\boldsymbol{\beta}_{01}) - \frac{\lambda_n}{\sqrt{n}} H_{n1}(\boldsymbol{\beta}_1^*)^{-1/2} D_1(\hat{\boldsymbol{\beta}}_1^\circ) \left\{ H_{n1}^{-1}(\boldsymbol{\beta}_1^*) + D_1(\hat{\boldsymbol{\beta}}_1^\circ) \right\}^{-1} \frac{1}{n} \dot{l}_{n1}(\boldsymbol{\beta}_{01}) \\
&= H_{n1}(\boldsymbol{\beta}_1^*)^{-1/2} \frac{1}{\sqrt{n}} \dot{l}_{n1}(\boldsymbol{\beta}_{01}) + o_p(1). \qquad (62)
\end{aligned}$$

From Condition (C3), as $n \to \infty$ we have that $H_{n1}(\boldsymbol{\beta}_1^*) \to I_1(\boldsymbol{\beta})$, and similarly to **?**,

$$H_{n1}(\boldsymbol{\beta}_1^*)^{-1/2} \frac{1}{\sqrt{n}} \dot{l}_{n1}(\boldsymbol{\beta}_{01}) \xrightarrow{D} N(\mathbf{0}, I_q). \qquad (63)$$

Now we have that

$$\sqrt{n} H_{n1}(\boldsymbol{\beta}_1^*)^{1/2} (\hat{\boldsymbol{\beta}}_1^\circ - \boldsymbol{\beta}_{01}) \xrightarrow{D} N(\mathbf{0}, I_q), \qquad (64)$$

or equivalently

$$\sqrt{n} (\hat{\boldsymbol{\beta}}_1^\circ - \boldsymbol{\beta}_{01}) \xrightarrow{D} N(\mathbf{0}, I_1(\boldsymbol{\beta})^{-1}). \qquad (65)$$

Proof of Theorem 1:

From part (b) of Lemma 2, we have that

$$\Pr\left(\lim_{k \to \infty} g_2(\boldsymbol{\beta}^{(k)}) = \hat{\boldsymbol{\beta}}_2 = \mathbf{0} \right) \to 1 \qquad (66)$$

as $n \to \infty$. Further, we want to show that

$$\Pr\left(\lim_{k \to \infty} \left\| g_1(\boldsymbol{\beta}^{(k)}) - \hat{\boldsymbol{\beta}}_1^\circ \right\| = 0 \right) \to 1, \qquad (67)$$

where $\hat{\boldsymbol{\beta}}_1^\circ$ is the fixed point of $f(\boldsymbol{\beta}_1)$ defined in Lemma 3. We have that $g(\boldsymbol{\beta})$ is a solution to

$$-\frac{1}{n} D(\boldsymbol{\beta})^{-1} \dot{l}_n(\boldsymbol{\theta}) + \frac{1}{n} \lambda_n \boldsymbol{\theta} = \mathbf{0}, \qquad (68)$$

where $D(\boldsymbol{\beta})^{-1} = diag\{\beta_1^2, \ldots, \beta_q^2, \beta_{q+1}^2, \ldots, \beta_{p_n}\}$. It is clear that $g_2(\boldsymbol{\beta}) = \mathbf{0}$ if $\boldsymbol{\beta}_{02} = \mathbf{0}$. We can further break down (68) into the following components: $g_1(\boldsymbol{\beta})$ is a solution to

$$-\frac{1}{n} D_1^{-1}(\boldsymbol{\beta}_1) \dot{l}_{n1}(\boldsymbol{\theta}_1) + \frac{1}{n} \lambda_n \boldsymbol{\theta}_1 = \mathbf{0} \qquad (69)$$

and $g_2(\boldsymbol{\beta})$ is a solution to

$$-\frac{1}{n} D_2^{-1}(\boldsymbol{\beta}_2) \dot{l}_{n2}(\boldsymbol{\theta}_2) + \frac{1}{n} \lambda_n \boldsymbol{\theta}_2 = \mathbf{0}. \qquad (70)$$

From (70) we can see that

$$\lim_{\boldsymbol{\beta}_2 \to 0} g_2(\boldsymbol{\beta}; \boldsymbol{\beta}_1, \boldsymbol{\beta}_2) = \mathbf{0} \qquad (71)$$

and from (69) we can see that

$$\lim_{\boldsymbol{\beta}_2 \to 0} g_1(\boldsymbol{\beta}; \boldsymbol{\beta}_1, \boldsymbol{\beta}_2) = f(\boldsymbol{\beta}_1). \tag{72}$$

Hence, $g(\cdot)$ is continuous for all $\boldsymbol{\beta} \in \mathcal{H}_n$.
Note that for any $\hat{\boldsymbol{\beta}}_2^{(k)}$, $g(\boldsymbol{\beta}; \boldsymbol{\beta}_1, \hat{\boldsymbol{\beta}}_2^{(k)})$ is a mapping of $\boldsymbol{\beta}_1$. Since g is continuous and with $\hat{\boldsymbol{\beta}}_2^{(k)} \to \mathbf{0}$ with probability tending to one, we have that

$$\omega_k \equiv \sup_{g_1(\boldsymbol{\beta}) \in [1/M_0, M_0]^q} \left\| f(\boldsymbol{\beta}_1) - g_1(\boldsymbol{\beta}; \boldsymbol{\beta}_1, \hat{\boldsymbol{\beta}}_2^{(k)}) \right\| \to 0, \tag{73}$$

as $k \to \infty$. Also by (58), for some $C_1 > 1$

$$\left\| f(\hat{\boldsymbol{\beta}}_1^{(k)}) - \hat{\boldsymbol{\beta}}_1^{\circ} \right\| = \left\| f(\hat{\boldsymbol{\beta}}_1^{(k)}) - f(\hat{\boldsymbol{\beta}}_1^{\circ}) \right\| \leq \frac{1}{C_1} \left\| \hat{\boldsymbol{\beta}}_1^{(k)} - \hat{\boldsymbol{\beta}}_1^{\circ} \right\|. \tag{74}$$

Further

$$\left\| \hat{\boldsymbol{\beta}}_1^{(k+1)} - \hat{\boldsymbol{\beta}}_1^{\circ} \right\| \leq \left\| g_1(\hat{\boldsymbol{\beta}}^{(k)}) - \hat{\boldsymbol{\beta}}_1^{\circ} \right\| \leq \left\| g_1(\hat{\boldsymbol{\beta}}^{(k)}) - f(\hat{\boldsymbol{\beta}}_1^{(k)}) \right\| + \left\| f(\hat{\boldsymbol{\beta}}_1^{(k)}) - \hat{\boldsymbol{\beta}}_1^{\circ} \right\|, \tag{75}$$

and thus

$$\left\| \hat{\boldsymbol{\beta}}_1^{(k+1)} - \hat{\boldsymbol{\beta}}_1^{\circ} \right\| \leq \frac{1}{C_1} \left\| \hat{\boldsymbol{\beta}}_1^{(k)} - \hat{\boldsymbol{\beta}}_1^{\circ} \right\| + \omega_k. \tag{76}$$

Letting $a_k = \left\| \hat{\boldsymbol{\beta}}_1^{(k)} - \hat{\boldsymbol{\beta}}_1^{\circ} \right\|$ for all $k \geq 0$, we can rewrite (76) as

$$a_{k+1} \leq \frac{1}{C_1} a_k + \omega_k. \tag{77}$$

From (76), we know that there exists an $N > 0$ such that for all $k > N$ and for all $\epsilon > 0$, we have that $|\omega_k| < \epsilon$. Therefore for $k > N$, we have that as $k \to \infty$

$$\begin{aligned}
a_{k+1} &\leq \frac{1}{C_1} a_k + \omega_k \\
&\leq \frac{a_{k-1}}{C^2} + \frac{\omega_{k-1}}{C} + \omega_k \\
&\leq \frac{a_1}{C^k} + \frac{\omega_1}{C^{k-1}} + \cdots + \frac{\omega_N}{C^{k-N}} + \left(\frac{\omega_{N+1}}{C^{k-N-1}} + \cdots + \frac{\omega_{k-1}}{C} + \omega_k \right) \\
&\leq (a_1 + \omega_1 + ... + \omega_N) \frac{1}{C^{k-N}} + \frac{1 - (1/C)^{k-N}}{1 - 1/C} \epsilon \to 0,
\end{aligned}$$

with probability tending to one. Therefore,

$$\Pr\left(\lim_{k \to \infty} \left\| \hat{\boldsymbol{\beta}}_1^{(k)} - \hat{\boldsymbol{\beta}}_1^{\circ} \right\| = \mathbf{0} \right) = 1, \tag{78}$$

or equivalently

$$\Pr(\hat{\boldsymbol{\beta}}_1 = \hat{\boldsymbol{\beta}}_1^{\circ}) = 1. \tag{79}$$

Thus from (79) and (66), we have that

$$\lim_{k \to \infty} \hat{\boldsymbol{\beta}}^{(k)} = \lim_{k \to \infty} (g_1(\boldsymbol{\beta}^{(k)})^T, g_2(\boldsymbol{\beta}^{(k)})^T)^T = (\hat{\boldsymbol{\beta}}_1^{\circ T}, \mathbf{0}^T)^T, \tag{80}$$

and therefore part (a) of Theorem 1 is complete. Part (b) of Theorem 1 follows from part (b) of Lemma 3.

A.3 Proof of Theorem 2.

We have that $\hat{\boldsymbol{\beta}} = \lim_{k\to\infty} \hat{\boldsymbol{\beta}}^{(k)}$, where

$$\hat{\boldsymbol{\beta}}^{(k+1)} = g(\hat{\boldsymbol{\beta}}^{(k)}) = \arg\min_{\boldsymbol{\beta}} \left\{ -2l_n(\boldsymbol{\beta}) + \lambda_n \sum_{j=1}^{p_n} \frac{1(\beta_j \neq 0)\beta_j^2}{\left(\hat{\beta}_j^{(k)}\right)^2} \right\}.$$

Looking at (20) we have that,

$$D(\hat{\boldsymbol{\beta}}^{(k)})^{-1}\dot{l}_n(\hat{\boldsymbol{\beta}}^{(k+1)}) = \lambda_n \hat{\boldsymbol{\beta}}^{(k+1)} \tag{81}$$

Therefore for any $l = i, j$ where $\hat{\beta}_i \neq 0$, $\hat{\beta}_j \neq 0$,

$$\hat{\beta}_l^{(k+1)} = \frac{(\hat{\beta}_l^{(k)})^2}{\lambda_n}\dot{l}_{nl}(\hat{\boldsymbol{\beta}}^{(k+1)}). \tag{82}$$

From Theorem 1, we also have that as $k \to \infty$, $\hat{\boldsymbol{\beta}}^{(k)} \to \hat{\boldsymbol{\beta}}$ and hence as $k \to \infty$, (82) can be rewritten as

$$\hat{\beta}_l^{-1} = \frac{1}{\lambda_n}\dot{l}_{nl}(\hat{\boldsymbol{\beta}}). \tag{83}$$

Let $\boldsymbol{\eta} = X\boldsymbol{\beta}$ and

$$\zeta(\eta_i) = \frac{\partial}{\partial \eta_i} l_n(\boldsymbol{\beta}) = N_i(1) - \int_0^1 \frac{Y_i(s)\exp(\eta_i)}{\sum_{j=1}^n Y_j(s)\exp(\eta_j)} d\bar{N}(s) \quad i = 1, \ldots, n. \tag{84}$$

Then

$$|\zeta(\hat{\eta}_i)| \leq |N_i(1)| + \left| \int_0^1 \frac{Y_i(s)\exp(\hat{\eta}_i)}{\sum_{j=1}^n Y_j(s)\exp(\hat{\eta}_j)} d\bar{N}(s) \right| \leq 1 + d_n \quad i = 1, \ldots, n, \tag{85}$$

since the integrand is at most one and where $d_n = \sum_{i=1}^n \delta_i$. Hence

$$\|\zeta(\hat{\boldsymbol{\eta}})\| \leq \|\mathbf{1} + d\mathbf{1}\| = \sqrt{n(1+d)^2}. \tag{86}$$

Let $\mathbf{x}_{[,i]}$ denote the i^{th} column of X. Since X is assumed to be standardized, $\mathbf{x}_{[,i]}^T\mathbf{x}_{[,i]} = n-1$ and $\mathbf{x}_{[,i]}^T\mathbf{x}_{[,j]} = (n-1)r_{ij}$, for all $i \neq j$ and where r_{ij} is the sample correlation between $x_{[,i]}$ and $x_{[,j]}$. Since

$$\hat{\beta}_i^{-1} = \frac{1}{\lambda_n}\mathbf{x}_{[,i]}^T\zeta(\hat{\boldsymbol{\eta}}) \quad \text{and} \quad \hat{\beta}_j^{-1} = \frac{1}{\lambda_n}\mathbf{x}_{[,j]}^T\zeta(\hat{\boldsymbol{\eta}}), \tag{87}$$

we have that

$$\begin{aligned}
\left|\hat{\beta}_i^{-1} - \hat{\beta}_j^{-1}\right| &= \left|\frac{1}{\lambda_n}\mathbf{x}_{[,i]}^T\zeta(\hat{\boldsymbol{\eta}}) - \frac{1}{\lambda_n}\mathbf{x}_{[,j]}^T\zeta(\hat{\boldsymbol{\eta}})\right| \\
&= \left|\frac{1}{\lambda_n}(\mathbf{x}_{[,i]} - \mathbf{x}_{[,j]})^T\zeta(\hat{\boldsymbol{\eta}})\right| \\
&\leq \frac{1}{\lambda_n}\left\|(\mathbf{x}_{[,i]} - \mathbf{x}_{[,j]})\right\| \|\zeta(\hat{\boldsymbol{\eta}})\| \\
&\leq \frac{1}{\lambda_n}\sqrt{2\{(n-1) - (n-1)r_{ij}\}}\sqrt{n(1+d)^2} \qquad (88)
\end{aligned}$$

for any $\hat{\beta}_i \neq 0$ and $\hat{\beta}_j \neq 0$.