report_update5.pdf

by Yangjun LU

Submission date: 20-Jul-2024 09:59PM (UTC+0800)

Submission ID: 2419603834

File name: report_update5.pdf (719.06K)

Word count: 7928

Character count: 40195

City University of Hong Kong

PROJECT REPORT

Divide-and-Conquer for Massive Survival Data

Supervisor：Prof. Su Wen

Author：Lu Yangjun,

Yang Haotian,

Yue Rongqi,

Wu Cheng

Department of Biostatistics

July 20, 2024

Abstract

We presents an application of the Divide-and-Conquer (DAC) algorithm , combined with the adaptive LASSO penalty for the Cox proportional hazards model. We aim to improve computational efficiency in analyzing large-scale biomedical data. The $\mathrm{DAC}_{\mathrm{lin}}$ approach uses a linearization technique to simplify model estimation, greatly reducing the computational burden compared to traditional methods. Simulation studies show that the DAC algorithm performs well in various data settings, cutting computation time significantly while maintaining accuracy. We also applied the DAC algorithm to a COVID-19 dataset, identifying key factors affecting patient outcomes after data preprocessing and feature engineering. The results demonstrate the DAC algorithm's innovation and value in practical research.

Keywords: Divide-and-Conquer Algorithm; Cox Proportional Hazards Model; Adaptive LASSO

INDEX

1.Introduction

The field of biomedical research has experienced an exponential growth in data availability, primarily attributed to technological advancements in data collection and storage. The computational challenges inherent in analyzing right-censored survival data within the scope of large datasets have driven the need for methodological strategies. Sparse regression models effectively eliminate irrelevant predictors and estimate the impact of those that contribute valuable information. It is especially useful for analyzing time-to-event data affected by censoring. A straightforward method for achieving a sparse risk prediction model is fitting a Cox proportional hazards model with regularization, utilizing penalties like the adaptive LASSO, as introduced by Liu (2007). In this study, we examine the deployment of the DAC strategy, a novel methodology designed to fit aLASSO penalized Cox proportional hazards model. The algorithm can further decrease the computational challenges inherent in the analysis of massive datasets and offers a viable solution for handling large sample sizes and high-dimensional predictors.

Survival analysis has been significantly advanced by the development of non-parametric models , parametric models, and semi-parametric models, which offer flexibility in handling the uncertainty and randomness inherent in time-to-event data. Semi-parametric models have diversified over time, offering various methodological options. The Cox proportional hazards model, introduced by Cox (1972,1975), has been a cornerstone due to its ability to estimate the relative risk of events over time without specifying the baseline hazard function. This model has been extensively used and refined. The accelerated failure time model, alternatively, assumes a direct relationship between covariates and the survival time, offering a different perspective on the effect of predictors (Wei, 1992; Zhang & Davidian, 2001). This model has been particularly useful in settings where the proportional hazards assumption is not satisfied. Lastly, the additive hazards model, which posits a linear relationship between covariates and the log of the hazard function, has gained traction for its simplicity and interpretability (Aalen, 1989; Lin & Ying, 1994). Recent methodological advancements have aimed at addressing the challenges of estimation and inference with this model in the presence of high-dimensional data (Huang &Wang, 2012; Wang et al., 2018).

The Cox proportional hazards model, introduced by Cox (1972,1975), allows for the estimation of hazard ratios while accommodating censored data, making it a versatile tool in biomedical research. Over the years, numerous studies have expanded its applications and theoretical underpinnings. Recent advancements have focused on enhancing its predictive accuracy and computational efficiency. For instance, Zhang et al. (2019) proposed a Bayesian approach to improve the model's robustness, while Li and Zhao (2020) developed a penalized likelihood method to handle high-dimensional data. The integration of machine learning techniques, as seen in the work of Wang et al. (2021), has also been explored to refine the model's predictive capabilities. These developments underscore the ongoing efforts to adapt the Cox model to modern data challenges.

The Divide-and-Conquer (DAC) algorithm has emerged as a strategic approach to address the computational challenges in large-scale data analysis. Recent studies have demonstrated its effectiveness in fitting penalized Cox models. For example, Liu and Li (2018) proposed a DAC-based, significantly reducing the time required for model fitting. Further, the work of Zhao et al. (2020) showed how DAC can be integrated with distributed computing frameworks to handle massive datasets. The method's scalability and robustness have been highlighted in the literature, with applications ranging from genomics to epidemiological studies. Recent advancements, such as the work of Chen et al. (2022), have focused on optimizing DAC for cloud computing environments, further extending its applicability in the era of big data.

A critical review of the literature reveals a persistent trade-off between computational feasibility and statistical accuracy. As datasets expand in size and complexity, there is an urgent need for a method that can efficiently handle large sample sizes and high-dimensional predictors without compromising the precision of estimates. To address the research gaps, this study addresses the question that how DAC methodology be effectively applied to analyze massive survival datasets under the Cox proportional hazards model, while maintaining statistical accuracy and computational efficiency.

In this study, we introduce an innovative DAC algorithm, termed $\mathrm{DAC}_{\mathrm{lin}}$, which employs a linearization technique to estimate the aLASSO penalized Cox proportional hazards models. This approach significantly alleviates the computational burden when compared to traditional DAC methods." This study aims to provide a comprehensive application of the DAC approach for the analysis of massive survival data, offering a viable solution to a critical issue in the field.

The subsequent structure of this study is outlined below. In Section 2, we will focus on the theory of DAC algorithm, starting from Cox Proportional Hazard model. Firstly, we will discuss how to estimate coefficients with unpenalized likelihood. Then we will add adaptive LASSO penalty term and explain how it works. After that, the importance of DAC algorithm will be shown. Steps of the algorithm will be displayed, and we will show some instinct idea about some expression. Section 3 uses simulation to evaluate the performance of DAC algorithm in handling different data configuration, comparing it with the full sample-based aLASSO estimator for Cox models, demonstrating efficiency and computational advantages. Section 4 primarily details the application of data cleaning and feature engineering within a specific dataset related to COVID-19 information. It also examines the implementation of algorithms discussed in the Section 3. The study finds that feature information was effectively extracted during data cleaning and feature engineering. In terms of algorithm application, the DAC algorithm significantly enhances computational efficiency, particularly in scenarios involving mixed binary and continuous covariates.

2. Theory

2.1 Notations and Settings

We will let T represent an individual's time of the period of surviving. $Z(\cdot)$ is a p vector consists of factors that might have significant effect on the survival time T. There are situations that $Z(\cdot)$ is related to survival time T, called time-dependent factor. But in this report we will only consider time-independent case. C records the time the individual is censored. Δ is an indicator that shows the censor status of the individual. If an individual is uncensored, Δ_i of the individual equals to 1. Now we let $X = \min(T, C)$, combining the survival time and the censor time to an individual's survival time during the study period. So we have a complete survival data for each individual: $D_i = (X_i, \Delta_i, Z_i(\cdot))$. We assume there are n individuals, so the size of data is n. By stacking all data and turning it into a matrix, we have $D = \begin{bmatrix} D_1 \\ \vdots \\ D_n \end{bmatrix}$. Since $Z(\cdot)$ is a p vector, so β is also a p vector, with $\beta = (\beta_1, \ \beta_2, \ldots, \ \beta_p)$. β here is assumed to be sparse, and $n >> p$.

In this report, we adopt a semi-parametric model: Cox proportional hazard model (Cox 1972, 1975) to predict the survival time of individuals. The Cox proportional hazard model describes the relationship among hazard function, baseline hazard function and covariates vector $Z(\cdot)$ in the following expression:

$$h(t \mid Z) = h_0(t) \cdot e^{\beta Z} \tag{2.1}$$

After fitting Cox proportional hazard model, we can interpret the coefficient through relative risk. Suppose we have a binary covariate Z, and we can interpret the coefficient of Z by the following function:

$$\frac{h(t \mid Z = 1)}{h(t \mid Z = 0)} = e^{\beta(1-0)} = e^{\beta} \tag{2.2}$$

2.2 The estimation of origin Cox's proportional hazards model

Now lets consider the estimation of original Cox proportional hazard model. We can first derive the likelihood function as follows:

$$L(\beta,h_0(\cdot))=\prod_i\left(f(x_i\mid z_i)^{\Delta_i}\cdot S(x_i\mid z_i)^{1-\Delta_i}\right)\tag{2.3}$$

The function above can be explained by dividing individuals into two groups. Individuals in the dataset are seperated according to the status of censor. If the event the researchers are interested in happened before the individual is censored for an individual, the individual belongs to group 1. In this case we use the $f(\cdot)$ to describe the probability for this situation. Otherwise, the individual belongs to group 2, which means the patient survives through the whole study. So we use the survival function $S(\cdot)$ to express this probability.

By concluding the input of the likelihood function (3), we have the following expression that explains the likelihood function relies on three components: data, β and hazard function $h(\cdot)$:

$$L(\beta,h_0(\cdot))=function\left(data,\beta,h_0(\cdot)\right)\tag{2.4}$$

From expression (4), to acquire the maximum likelihood estimator, we need data and baseline hazard function $h_0(\cdot)$. However, since Cox proportional hazard model is a semi-parametric model, there is no arbitrary form for $h_0(\cdot)$. As a result, the baseline hazard function $h_0(\cdot)$ will be excluded from the likelihood function, since we have no information about it:

$$L(\beta,h_0(\cdot))=L_1(\beta)\cdot L_2(\beta,h_0(\cdot))\tag{2.5}$$

After separation, we have $L_1(\beta)$ that does not depend on baseline hazard function, which is also called partial likelihood PL. It is feasible to estimate coefficients through PL, and the estimator has been proven to be consistent to the true parameter β. It also enjoys the asymptotic normal property (Cox 1972, 1975). There are also drawbacks when using partial likelihood, for example, the estimator we have may be inefficient.

2.3 Adding penalty term to the semi-parametric model

However, if we are tackling with lots of covariates, modification of the likelihood function must be done. A penalty term is what we want to help us select proper factors. First we need to apply log transformation to partial likelihood function $L_1(\beta)$:

$$l_n(\beta)=\sum_{i=1}^{n}\Delta_i\left[z_i\beta-\log\left(\sum_{j=1}^{n}I\left(X_j\geq X_i\right)e^{z_j\beta}\right)\right] \tag{2.6}$$

Define $J(\beta_j)=|\beta_j|$ and add the LASSO penalty term to function (6):

$$-\frac{1}{n}l_n(\beta)+\lambda\sum_{j=1}^{p}|\beta_j| \tag{2.7}$$

We have LASSO penalized Cox's proportional hazard model's objective function (7), where λ is the tuning parameter. It controls the degree of filtering. Large λ will lead to great bias between estimated coefficients and true coefficients. On the other hand, Small λ provides trivial power to filter out factors with small coefficients.

In (7), we provide all covariates with a same weight λ. We can improve the LASSO penalty term by adjusting the weight automatically according to their coefficient, and upgrade the objective function (7):

$$\min_{\beta}\left(-\frac{1}{n}l_n(\beta)+\lambda\sum_{j=1}^{p}\frac{|\beta_j|}{|\tilde{\beta}_j|}\right) \tag{2.8}$$

The constant factor $|\tilde{\beta}_j|$ added in the penalty term aims to automatically change the weight for different factors. It is chosen adaptively according to the data we collected, and we simply define $\tilde{\beta}=\arg\max_{\beta}\left(\frac{1}{n}l_n(\beta)\right)$. With the adaptive penalty term, factors with large coefficient have smaller weight, and they are more easier to be kept. Factors with smaller coefficient have larger weight, and they are more likely to be filtered out (Zhang & Lu, 2007).

2.4 Divide-and-conquer algorithm

2.4.1 Notations

Many algorithms can be applied to solve function (7), such as interior point algorithm (Boyd & Vandenberghe, 2004) and modified shooting algorithm (Zhang & Lu, 2007). However, there are several reasons we need to use divide-and-conquer algorithm: In real world, the data are not stored in the same place, and it is hard to gather the data together and analyze it as a whole dataset; Besides, it is computationally challenging when we are facing a large dataset. For example, there might be a million data and hundreds of covariates in a clinical dataset. Insisting on applying traditional algorithms is time-consuming.

In order to apply the fast divide-and-conquer algorithm (Wang & Hong, 2019) to data, changes of notations are required to be done. The data $D = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_{n-1} \\ D_n \end{bmatrix}$ is randomly divided into K subsets. For each subset, the index space of data that are included is noted as Ω_k. Similarly, for the whole set, we have $\Omega = \Omega_{full}$. For simplicity, we define $l_{\Omega_{full}} = \frac{1}{n} l_n(\beta)$, and we note the estimation from (8) as $\hat{\beta}_{full}$:

$$\hat{\beta}_{full} = \arg\max_{\beta} \left(l_{\Omega_{full}}(\beta) - \lambda \sum_{j=1}^{p} \frac{\left|\beta_j\right|}{\left|\tilde{\beta}_j\right|} \right) \qquad (2.9)$$

2.4.2 Least square approximation

First, we need to apply least square approximation (LSA) by Wang and Leng (2007) to the likelihood function $l_{\Omega_{full}}$. To transform the likelihood function, we need to apply Taylor series expansion around the estimator $\tilde{\beta} = \arg\max_{\beta} \left(\frac{1}{n} l_n(\beta) \right)$:

$$l_{\Omega_{full}}(\beta) \approx l_{\Omega_{full}}\left(\tilde{\beta}\right) + \dot{l}_{\Omega_{full}}\left(\tilde{\beta}\right)^{T}\left(\beta-\tilde{\beta}\right) + \frac{1}{2}\left(\beta-\tilde{\beta}\right)^{T}\left(\ddot{l}_{\Omega_{full}}\left(\tilde{\beta}\right)\right)\left(\beta-\tilde{\beta}\right) \qquad (2.10)$$

which $\dot{l}_{\Omega_{full}}(\beta) = \frac{\partial l_{\Omega_{full}}(\beta)}{\partial \beta}$ and $\ddot{l}_{\Omega_{full}}(\beta) = \frac{\partial^{2} l_{\Omega_{full}}(\beta)}{\partial \beta^{2}}$ are respectively the first order and second order derivatives of the likelihood function $l_{\Omega_{full}}$.

Since $\tilde{\beta}$ is the estimation that maximize the likelihood function, naturally we have $\dot{l}_{\Omega_{full}}\left(\tilde{\beta}\right) = 0$. As a result, expression (9) can be simplified as following:

$$l_{\Omega_{full}}(\beta) \approx l_{\Omega_{full}}\left(\tilde{\beta}\right) + \frac{1}{2}\left(\beta-\tilde{\beta}\right)^{T}\left(\ddot{l}_{\Omega_{full}}\left(\tilde{\beta}\right)\right)\left(\beta-\tilde{\beta}\right) \qquad (2.11)$$

Because the constant term $l_{\Omega_{full}}\left(\tilde{\beta}\right)$ does not affect the estimation of β, it can be ignored. We note the covariance matrix of $\tilde{\beta}$ as Σ, and the inverse of covariance matrix Σ is found to have an natural estimator $\ddot{l}_{\Omega_{full}}\left(\tilde{\beta}\right)$. Thus we can replace $\ddot{l}_{\Omega_{full}}\left(\tilde{\beta}\right)$ with $\hat{\Sigma}^{-1}$, and we have the following function acting as an approximation of the likelihood function:

$$l_{\Omega_{full}}(\beta) \approx \frac{1}{2}\left(\beta-\tilde{\beta}\right)^{T}\left(\hat{\Sigma}^{-1}\right)\left(\beta-\tilde{\beta}\right) \qquad (2.12)$$

Plug (12) into function (9), we have:

$$\hat{\beta}_{full,lin} \approx \arg\max_{\beta}\left(\frac{1}{2}\left(\beta-\tilde{\beta}\right)^{T}\left(\hat{\Sigma}^{-1}\left(\tilde{\beta}\right)\right)\left(\beta-\tilde{\beta}\right) - \lambda\sum_{j=1}^{p}\frac{\left|\beta_{j}\right|}{\left|\tilde{\beta}_{j}\right|}\right) \qquad (2.13)$$

The expression (13) can become more close to the least square estimation. Since covariance matrix Σ is positive-definite, we can apply Cholesky decomposition to the inverse of covariance matrix Σ^{-1}:

$$\Sigma^{-1} = L_{DAC}\left(\tilde{\beta}\right)L_{DAC}\left(\tilde{\beta}\right)^{T} \qquad (2.14)$$

Plug (14) into (13), and we have done the application of least square approximation on the objective function:

$$\hat{\beta}_{DAC} \approx \arg\max_{\beta}\left(\frac{1}{2}\left(Y\left(\tilde{\beta}\right)-X\left(\tilde{\beta}\right)\beta\right)^{T}\left(Y\left(\tilde{\beta}\right)-X\left(\tilde{\beta}\right)\beta\right)-\lambda\sum_{j=1}^{p}\frac{\left|\beta_{j}\right|}{\left|\tilde{\beta}_{j}\right|}\right) \tag{2.15}$$

where $Y\left(\tilde{\beta}\right)=\hat{L}_{DAC}\left(\tilde{\beta}\right)\tilde{\beta}$ and $X\left(\tilde{\beta}\right)=\hat{L}_{DAC}\left(\tilde{\beta}\right)$. Here $\hat{L}_{DAC}\left(\tilde{\beta}\right)$ is a $p\times p$ matrix.

2.4.3 Divide-and-conquer algorithm

Next, we need to use divide-and-conquer algorithm to get $\tilde{\beta}$. To solve (15), we need to solve $\tilde{\beta}$ through $\tilde{\beta}=\arg\max_{\beta}\left(l_{\Omega_{full}}(\beta)\right)$. But as we have mentioned above, we may face different challenges when trying to solve it directly. Under this circumstance, divide-and-conquer is our primary choice. We will first solve $\tilde{\beta}$ through divide-and-conquer algorithm, and finally solve for the final estimation of $\hat{\beta}_{DAC}$ in (15).

We will derive an estimation of $\tilde{\beta}$ called $\tilde{\beta}_{DAC}$ in the following steps:

Step (1) $$\tilde{\beta}_{DAC}^{[0]}=\arg\max_{\beta} l_{\Omega_1}(\beta)$$

In step (1), we directly solve for the estimation of $\tilde{\beta}$ on the first subset.

Step (2) $$\tilde{\beta}_{DAC}^{[i]}=\frac{1}{K}\sum_{j=1}^{K}\tilde{\beta}_{\Omega_{j,lin}}\left(\tilde{\beta}_{DAC}^{[i-1]}\right) \text{, for } i=1,\ldots,I$$

The goal of step (2) is to update the $\tilde{\beta}_{DAC}^{[i]}$, according to the method provided by Wang and Hong (2019). We can define the iteration times I we want to update $\tilde{\beta}_{DAC}^{[i]}$. After I times iterations, the result we get $\tilde{\beta}_{DAC}=\tilde{\beta}_{DAC}^{[I]}$ is the DAC approximation to $\tilde{\beta}$.

Step(3) $$\hat{\beta}_{DAC} \approx \arg\max_{\beta}\left(\frac{1}{2}\left(Y\left(\tilde{\beta}_{DAC}\right)-X\left(\tilde{\beta}_{DAC}\right)\beta\right)^{T}\left(Y\left(\tilde{\beta}_{DAC}\right)-X\left(\tilde{\beta}_{DAC}\right)\beta\right)-\lambda\sum_{j=1}^{p}\frac{\left|\beta_{j}\right|}{\left|\tilde{\beta}_{DAC,j}\right|}\right)$$

In step (3) we plug the estimation into expression (15). The estimator is gained by solving the optimization problem in this step. In simulation, we found that the computation time is much faster than directly fitting the model using objective function (9). Besides, the property of estimator is proven to be similar to that in (9), such as asymptotic distribution and consistency (Wang & Hong, 2019).

2.5 Tuning parameter

Previously we have mentioned how does the tuning parameter affect the estimator. If we want to pick out the best model, the optimization of tuning parameter λ is required. In this report, Bayesian information criterion (BIC) is used in choosing tuning parameter, but there is modification of BIC specially designed for survival model (Volinsky & Raftery, 2000). In the modified BIC, the number of uncensored cases $u = \sum_i \Delta_i$ is combined with the penalty term. Initial λ is chosen by researchers, and the optimized λ will be calculated automatically while fitting the model. The modified BIC is defined as:

$$BIC_{\text{mod}} = -2\sum_i l_i\left(\beta_{\lambda,\Omega_{full}}\right) + \left(\log u\right) df_\lambda$$

The specific tuning parameter that produces the minimum BIC will be chose and plugged into (15), and we can solve for the estimation with the lowest BIC.

3. Simulation

3.1 Simulation objective

The primary objective of our simulation study is to create various data scenarios to assess the performance of both the DAC algorithm and the full sample-based aLASSO algorithm in order to show that the estimator $\hat{\beta}_{DAC}$ achieves the same performance as $\hat{\beta}_{full}$ but can be computed very efficiently when handling massive datasets.

3.2 Simulation settings

We examine the efficacy of the full sample-based aLASSO estimator in contrast to the Cox model's β_{full}. In this analysis, the primary focus is on scenarios with a large sample size $n_0 = 10^6$, subsets *K*=100, each with a subset sample size n= 10^4. Additionally, we investigate the influence of the number of iterations, considering *I*= 1, 2, and 3, to understand how iterations affect the performance of the proposed estimator.

We additionally considered n_0 = 10^5, 2×10^5, 5×10^5, 10^6, and 2×10^6 to assess the influence of varying initial values of n_0 on the comparative effectiveness of various methods, particularly in computational efficiency.

We created simulation survival data to fit Cox proportional hazards model using the SIM.FUN function in R package *divideconquer* (Yan Wang & Tianxi Cai 2024), which is based on exponential distribution and Weibull distribution.

In the case of penalized methods, the tuning parameter was chosen with the Bayesian Information Criterion (BIC) as detailed in Section 2. The aLASSO penalized estimator was implemented using the glmnet function from the R package glmnet (Friedman et al., 2010; Simon et al., 2011), with the parameter γ set to 1.

The DAC algorithms can achieve greater efficiency through the utilization of parallel computing techniques, however, to ensure fair comparison with other algorithms, the simulation was carried out as a single-core job on AMD Ryzen 9 5950X 16-Core Processor 3.40 GHz.

3.3 Time-independent covariates

For the covariates, we considered p = 50 and p = 150. We generated Z from a multivariate normal distribution with mean 0 and variance $\Sigma = [\mathrm{I}(\mathrm{l} = \mathrm{l}') + \mathrm{vI}(\mathrm{l} \neq \mathrm{l}')]_{l=1,\ldots,p}^{l'=1,\ldots,p}$, and we also considered correlation denoted as v= 0.55, and 0.85 to represent moderate, and strong correlations among the covariates respectively. For a given Z_i, i = 1, · · ·, n_0, we generated T_i from a Weibull distribution with a shape parameter of 2 and a scale parameter of $\{0.5\, exp(\beta_0^T Z_i)\}^{-0.5}$, where we considered one choice of β_0 to reflect different degrees of sparsity and signal strength:

$$\beta_0^{(I)} = (0.8, 0.75\,_2^T, 0.55\,_2^T, 0.35\,_2^T, 0.05\,_2^T, 0\,_{p-9}^T)^T$$

We created a variable C derived from an exponential distribution with an exp(0.5) rate parameter for censoring, which led to a censoring rate ranging from 68% to 76% across various configurations. For each configuration, we generated random datasets and subsequently ran each combination of study settings a total of M=100 times.

3.4 Measures of performance

Measures of performance: We report the observed values of the following performance measures.

Time (seconds): the mean execution time of code in different configurations

Bias ($\times 10^{-3}$): the average deviation of $\hat{\beta}$ from the true value;

MSE ($\times 10^{-5}$): mean squared error;

GMSE ($\times 10^{-5}$): global mean squared error, defined as:

$$\mathrm{GMSE} = \frac{1}{M}\sum_{m=1}^{M}(\hat{\beta}_{DAC} - \beta_0)\,\Sigma(\hat{\beta}_{DAC} - \beta_0)$$

3.5 Simulation Results

Table 1 presents the mean computation time and the Global Mean Squared Error (GMSE) for the unpenalized estimators, denoted as $\hat{\beta}_{DAC}$ and $\hat{\beta}_{\Omega\text{full}}$. The table shows that the DAC estimator $\hat{\beta}_{DAC}$, when iterated twice (I = 2), achieves a GMSE that is on par with the estimator based on the full dataset $\hat{\beta}_{\Omega\text{full}}$, while also cutting the computation time by over 50%. For example, when p=150, and v=0.55, the mean computation time of $\hat{\beta}_{DAC}$ is 33.42 (I = 2) much smaller than the mean computation time of $\hat{\beta}_{\Omega\text{full}}$ 125.55, but very close GMSE 50.7 and 46.1 representing similar efficiency. Moreover, the GMSE of $\hat{\beta}_{DAC}$ with two iterations (I = 2) is comparable to that with three iterations (I = 3). In all scenarios, the outcomes for $\hat{\beta}_{DAC}$ are virtually the same whether Iteration is set to 2 or 3. As a result, we will only conclude the results for $\hat{\beta}_{DAC}$ with I = 2 in the subsequent discussion.

The mean computation times for the time-independent survival data are summarized in Table 2 (where p=50) and Table 3 (where p=150). These times correspond to the aLASSO penalized estimator using the true parameter vector β_0, which is defined as $\beta_0^{(I)} = (0.8, 0.75_2^T, 0.55_2^T, 0.35_2^T, 0.05_2^T, 0_{p-9}^T)^T$. In various scenarios, the mean computation duration for $\hat{\beta}_{DAC}$ varies from 2.34 to 7.04 seconds with p=50 and from 16.32 to 54.21 seconds with p=150, with nearly the entire duration dedicated to the calculation of the unpenalized estimator $\hat{\beta}_{DAC}$. Conversely, $\hat{\beta}_{\Omega\text{full}}$ demands a more extended computation period, averaging from 271.75 to 276.77 seconds for p=50 and from 686.7 to 706.8 seconds for p=150. This indicates that the computation time for $\hat{\beta}_{DAC}$ is approximately 2.5% of that for the full sample estimator at p=50 and around 7% at p=150.

Table 1 Examinations of the estimators $\hat{\beta}_{DAC}$(I = 1, 2, 3) and $\hat{\beta}_{\Omega\text{full}}$ are conducted focusing on the average computation time in seconds and the global mean squared error (GMSE $\times 10^{-5}$) for the estimation of β_0.

P	Estimator	V	Iteration	Time	GMSE
50	β_{DAC}	0.55	1	2.34	18.2
			2	4.59	18.3
			3	6.91	18.3
	β_{full}			24.87	16.1
50	β_{DAC}	0.85	1	2.37	18.4
			2	4.68	18.9
			3	7.04	18.9
	β_{full}			25.155	17.8
150	β_{DAC}	0.55	1	16.32	50.9
			2	33.42	50.7
			3	50.73	50.7
	β_{full}			125.55	46.1
150	β_{DAC}	0.85	1	17.95	54.0
			2	36.16	55.4
			3	54.21	55.4
	β_{full}			139.07	43.1

On the other hand, based on the performances in Tables 2 and 3, we can observe that $\hat{\beta}_{DAC}$ and $\hat{\beta}_{\Omega\text{full}}$ show consistent results concerning GMSE and BIAS, indicating that the DAC algorithm is highly effective. In terms of the storage space required for computation, the full sample estimator $\hat{\beta}_{\Omega\text{full}}$ is only feasible to calculate under conditions where an extensive amount of memory is available, particularly when $n_0 = 2\times 10^6$. In contrast, the $\hat{\beta}_{\Omega\text{full}}$ estimator can be processed with a substantially more modest memory requirement.

Table 2 When estimating β_0 with p=50, we evaluate the performance of the estimators $\hat{\beta}_{DAC}$ (for I=1,2,3 iterations) and $\hat{\beta}_{\Omega\text{full}}$ across several indicator: average computation time in seconds, GMSE ($\times10^{-5}$), the empirical probability of coefficient selection, bias ($\times10^{-3}$), MSE ($\times10^{-5}$).

		V=0.55				V=0.85			
		β_{DAC}			β_{full}	β_{DAC}			β_{full}
	I =	1	2	3		1	2	3	
	TIME	2.34	4.59	6.91	271.75	2.37	4.68	7.04	276.77
	GMSE	4.94	5.03	5.03	3.57	5.74	6.20	6.20	2.13
	%zero	0	0	0	0	0	0	0	0
$\beta_{0j} = 0.8$	Bias	3.32	3.37	3.37	2.83	3.54	3.62	3.62	3.37
	MSE	1.38	1.41	1.41	1.38	1.74	1.77	1.77	1.74
	%zero	0	0	0	0	0	0	0	0
$\beta_{0j} = 0.75$	Bias	-0.89	-0.89	-0.89	-0.63	-1.21	-1.16	-1.16	-1.03
	MSE	0.65	0.66	0.66	0.65	0.92	0.94	0.94	0.92
	%zero	0	0	0	0	0	0	0	0
$\beta_{0j} = 0.55$	Bias	1.08	1.11	1.11	0.96	-0.15	-0.09	-0.09	-0.09
	MSE	0.44	0.44	0.44	0.44	0.55	0.56	0.56	0.55
	%zero	0	0	0	0	0	0	0	0
$\beta_{0j} = 0.35$	Bias	1.77	1.78	1.78	1.32	3.41	3.46	3.46	2.83
	MSE	1.09	1.09	1.09	1.09	2.50	2.54	2.54	2.50
	%zero	0	0	0	0	0	0	0	0
$\beta_{0j} = 0.05$	Bias	0.34	0.32	0.32	0.28	-2.41	-2.37	-2.37	-2.12
	MSE	0.11	0.10	0.10	0.10	1.55	1.54	1.54	1.54
	%zero	100	100	100	100	100	100	100	100
$\beta_{0j} = 0$	Bias	0	0	0	0	0	0	0	0
	MSE	0	0	0	0	0	0	0	0

Table 3 When estimating β_0 with p=150, we evaluate the performance of the estimators $\hat{\beta}_{DAC}$ (for I=1,2,3 iterations) and $\hat{\beta}_{\Omega\text{full}}$ across several indicator: average computation time in seconds, GMSE (×10^{-5}), the empirical probability of coefficient selection, bias (×10^{-3}), MSE (×10^{-5}).

		V=0.55				V=0.85			
		β_{DAC}			β_{full}	β_{DAC}			β_{full}
	I=	1	2	3		1	2	3	
	TIME	16.32	33.42	50.73	686.7	17.95	36.16	54.21	706.8
	GMSE	4.81	4.98	4.98	4.27	5.80	6.23	6.23	5.36
$\beta_{0j} = 0.8$	%zero	0	0	0	0	0	0	0	0
	Bias	3.46	3.53	3.53	3.24	3.42	3.52	3.52	3.23
	MSE	1.45	1.48	1.48	1.45	1.66	1.72	1.72	1.66
$\beta_{0j} = 0.75$	%zero	0	0	0	0	0	0	0	0
	Bias	-0.89	-0.84	-0.84	-0.77	-1.19	-1.12	-1.12	-1.08
	MSE	0.62	0.62	0.62	0.62	0.89	0.88	0.88	0.88
$\beta_{0j} = 0.55$	%zero	0	0	0	0	0	0	0	0
	Bias	0.98	1.01	1.01	0.94	-1.66	-0.96	-0.96	-0.63
	MSE	0.37	0.38	0.38	0.37	0.51	0.51	0.51	0.51
$\beta_{0j} = 0.35$	%zero	0	0	0	0	0	0	0	0
	Bias	1.67	1.67	1.67	1.62	3.58	3.59	3.59	3.42
	MSE	1.00	1.00	1.00	1.00	2.68	2.67	2.67	2.67
$\beta_{0j} = 0.05$	%zero	0	0	0	0	0	0	0	0
	Bias	0.48	0.45	0.45	0.38	-2.35	-2.31	-2.31	-2.06
	MSE	0.11	0.10	0.10	0.10	1.37	1.35	1.35	1.35
$\beta_{0j} = 0$	%zero	100	100	100	100	100	100	100	100
	Bias	0	0	0	0	0	0	0	0
	MSE	0	0	0	0	0	0	0	0

Table 4 summarizes the observed computational times based on 50 replications for n = (10^5 , 2×10^5 , 5×10^5 , 10^6 , 2×10^6) and p = (50, 150) with v = 0.55, and censoring rate 68% ~ 76%. When the sample size varies from $n_0 = 10^5$ to n_0=2×10^6 , the computation time increases for all methods but the increase is more drastic for $\hat{\beta}_{\Omega\text{full}}$. It is evident that our proposed DAC method demands less computational time compared to the full data-based evaluation in every scenario, with the speed advantage escalating as the sample size enlarges and the dimensionality increases. In practical terms, the evaluation of each subset via the DAC method would be conducted in parallel on separate machines, thereby diminishing the computational load.

Table 4 Comparisons of $\hat{\beta}_{DAC}$ (I =2), and $\hat{\beta}_{\Omega\text{full}}$ for estimating β_0 when p=50, v=0.55 with respect to average computation time in seconds and Difference (The temporal discrepancy between $\hat{\beta}_{DAC}$ and $\hat{\beta}_{\Omega\text{full}}$).

V=0.55		P=50			P=150	
N	DAC	Full	Difference	DAC	Full	Difference
100000	0.46	23.12	22.66	3.17	62.97	59.8
200000	0.86	48.45	47.59	6.39	129.39	123
500000	2.22	124.69	122.47	15.86	314.03	298.17
1000000	4.32	271.75	267.43	31.125	686.72	655.595
2000000	8.88	602.44	593.56	62.65	1456.23	1393.58

3.6 Statistical Performance

Generally speaking, the $\hat{\beta}_{\text{DAC}}$ estimator is capable of attaining a statistical performance on par with that of $\hat{\beta}_{\Omega\text{full}}$, particularly in terms of Global Mean Squared Error (GMSE), as well as the assessment of variable selection, bias, and Mean Squared Error (MSE) for individual coefficients. For example, as depicted in Table 2, the GMSEs ($\times10^{-5}$) for $\hat{\beta}_{\text{DAC}}$ and $\hat{\beta}_{\Omega\text{full}}$ are 5.03 and 3.57 for p=50 and v=0.55 respectively; and 4.98 and 4.27 for p=150 and v=0.55 respectively. The comparative performance of these methods exhibits similar trends across varying degrees of correlation v among the covariates. When the signals are notably strong and sparse, the performance of $\hat{\beta}_{\mathrm{DAC}}$ remains competitive with $\hat{\beta}_{\Omega\mathrm{full}}$, particularly for the strongest signals with a value of 0.80, both the DAC estimator $\hat{\beta}_{\mathrm{DAC}}$ and $\hat{\beta}_{\Omega\mathrm{full}}$ exhibit minimal bias and manage to achieve perfect variable selection. This indicates that in scenarios where the signal strength is high and the number of influential variables is limited, both estimators perform very well in bias and variable selection accuracy. Moreover, both penalized estimators for the weakest signal level (0.05) exhibit a minor bias at a correlation of v=0.55 and a more pronounced bias at v=0.85. Such biases in the estimation of weaker signals are expected with shrinkage estimators (Menelaos et al., 2016), especially in conditions of high correlation among predictors. It is noteworthy, however, that the performance of $\hat{\beta}_{\mathrm{DAC}}$ and $\hat{\beta}_{\Omega\mathrm{full}}$ is almost indistinguishable, which suggests that the DAC approach adds minimal additional approximation errors. The relative performance of $\hat{\beta}_{\mathrm{DAC}}$ and $\hat{\beta}_{\Omega\mathrm{full}}$ is consistent across various sample sizes. As n_0 ranges from 10^5 to 2×10^6, $\hat{\beta}_{\mathrm{DAC}}$ consistently outperforms as the most precise estimator, matching the accuracy of $\hat{\beta}_{\Omega\mathrm{full}}$ while also providing a substantial gain in computational efficiency.

4. Application

4.1 Data Resource

The dataset utilized in this study was sourced from the publicly available COVID-19 epidemiological data curated by the Open COVID-19 Data Curation Group (Xu et al., 2020). This dataset, which was last updated on April 1, 2021, encompasses epidemiological records of over 3 million COVID-19 patients reported between January 6, 2020, and February 16, 2021, across more than 60 countries or territories. The dataset includes detailed characteristics of individuals, such as age, sex, date of symptoms onset, date of COVID-19 confirmation, presence of chronic diseases (binary variable), type of chronic disease, geographic location, travel history, hospital admission date, specific event date (death, hospital discharge, or last recording), and outcome (death, discharge, or censored) (Xu et al., 2020).

4.2 Data Preprocessing and Feature Engineering

We addressed all missing values and transformed all categorical variables using One-Hot Encoding. Specifically, we applied natural language processing (NLP) techniques to the textual information within the chronic disease variable, resulting in the creation of over 60 binary variables.

To fit Cox models, the dataset was transformed into a time-to-event structure. The event variable was constructed such that if the outcome variable contained the keywords "discharge" or "recover," the event variable was set to 1; otherwise, it was set to 0 (Kvaløy & Lindqvist, 2003).

Nemati (2020) similarly provide method for constructing time variables The time variable is constructed by:

$$Time_1 = date\ of\ \ confirmation - \ date\ of\ symptom\ onset$$

$$Time_2 = \text{specific event date} - \ date\ of\ symptom\ onset$$

$$Time\ = \max(Time_1\ , Time_2)$$

And based time variable constructing, the censor variable is also determined:

$$Censor = \begin{cases} 1, & Time_2 \leq \ Time_1 \\ 0, & Otherwise \end{cases}$$

We cannot consider all the covariates into the model in this Section due to multicollinearity and other relevant factors. Kvaløy & Lindqvist (2003) suggest multicollinearity may cause to problems such as non-convergence in Cox models. To mitigate these issues, we implemented a feature engineering approach that involved principal component analysis (PCA) and the elimination of highly correlated variables (Rahmat et al., 2024).

Since variables of latitude and longitude do not accurately represent the direct Euclidean distances between individuals, we transformed these coordinates using the following method (Kumar et al., 2022).

$$Location\_x = R \times \cos(latitude) \times \ \cos(longitude)$$

$$Location\_y = R \times \cos(latitude) \times \ \sin(longitude)$$

$$Location\_z = R \times \sin(latitude)$$

$$R \ = radius\ of\ equator$$

4.3 Variable Description

After feature engineering, dataset remains 21 variables involving 2 response variables and 19 covariates. The main variables are shown in the table 6 at appendix.

Descriptive statistics for all continuous variables are shown in the table 5 below.

Table 5 Descriptive statistics for continuous variables

Variable	Mean	Median	Min	Max	Sample Size
Time	4.34	2	1	41	1867368
Age	46.76	47	1	106	1867368
Density	1460	146	7	7894	1867368
GDP	14342.903	10875.680	3912.680	48598.76	1867368
Per_GDP	9.36	0.49	8.13	10.30	1867368
Location_x	424.90	236.90	-6320.98	6369.05	1867368
Location_y	946	480.60	-6368.24	6370.68	1867368
Location_z	-278.70	102.20	-6371.45	6366.67	1867368

Except for some economic indicators, most of the variables do not have significant anomalous distributions (Beck & Levine, 2004).

4.4 Design of Application

We provide the following two steps to apply algorithmic design of Section (3) to this dataset.

(i) Verify the existence of multivariate effect in the dataset. Therefore, two hypotheses are proposed here:

$$\begin{cases} \mathrm{H}_{i0}: \text{ Nonexistence of multivariate effect} \\ \mathrm{H}_{i1}: \text{ Existence of multivariate effect} \end{cases}$$

(ii) If H_{i0} is rejected, we consider that multivariate effect may exist in the dataset. Execute the algorithm in the Section (3) and perform the same comparison method in terms of computation time as

Section (3) and provide regression results for Cause-Specific Cox models based on the DAC algorithm.

4.5 Outcome and Conclusion

Step (i):

To construct the Cox Multivariate model, we chose a simple but not robust method to determine whether it has a multivariate effect or not. If Kaplan-Meier curve cross, then the model may exist multivariate effect (Dormuth et al., 2022).

The following figure 1 shows the KM curve with chronic_disease (binary variable) as a covariate, and it is clear to see that the two curves intersect. Therefore, the H_{i0} hypothesis is rejected and the H_{i1} hypothesis is remained.

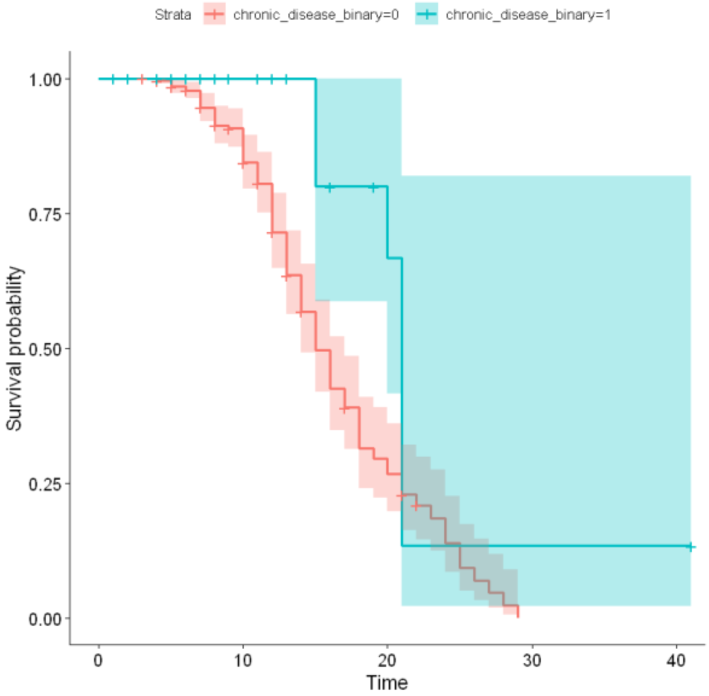

Figure 1 The Kaplan-Meier curve of chronic_disease covariate.

Step (ii):

Based on step (i), We applied DAC_{lin} algorithm with K = 10 and paralleled DAC_{lin} on this dataset. Computing $\hat{\beta}_{DAC}$ with L =2 took 227.57 seconds, which is 193.78% optimized compared to Computing

$\hat{\beta}_{\text{full}}$ which took 463.55 seconds.

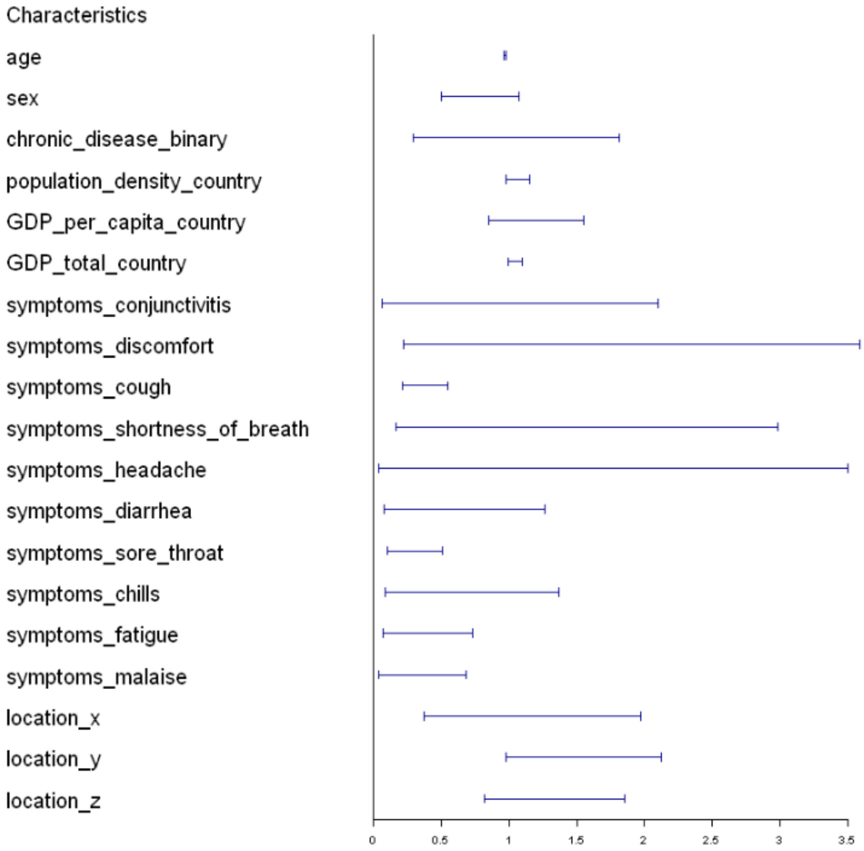

Figure 2 The forest plot of hazard ratios by DAC algorithm

The following figure 2 shows the hazard ratios for all covariates based on the DAC_{lin} for predicting discharge of COVID-19 admission individuals.

As can be seen from the above figure, all symptoms related variables have negative correlation on risk. Of these, the most significant impacts are fatigue symptoms. The results suggest that individuals without fatigue symptoms had a 5-times (CI: 1.24-times~14.3-times) changing of risk compared to individuals with fatigue symptoms. In the analysis of continuous covariates, all point estimates for the economic variables exceed 1, indicating that an increase in economic factors is associated with a higher risk of discharge. Conversely, the point estimate for age is 0.97, suggesting that an increase in age is associated with a lower risk of discharge.

5. Conclusion

In this report we have demonstrated the theory of divide-and-conquer algorithm combined with the modified Cox's proportional hazards model with special penalty term, the benefits of using the DAC algorithm and its application in real clinical data about Covid-19. In Section(2), the related theory about the modified Cox's proportional hazards model with special penalty term and the divide-and-conquer algorithm is explained. We start from the origin Cox's proportional hazards model and adaptive LASSO penalty term. Then backgrounds about why the divide-and-conquer algorithm is provided, with steps on using the algorithm. In Section (3), we created data with different configurations, and the performance indexes, such as MSE, bias and computation time are used to make comparison between methodologies with and without divide-and-conquer algorithm. Those indexes show that divide-and-conquer algorithm reduces the computational time, and provides a good estimation about coefficients. In Section (4), we apply the methodology to the real Covid-19 epidemiology data. Feature engineering is done before and while fitting the model, and table of selected covariates is then provided. We use the Kaplan-Meier plot to confirm that there is multivariate effect between covariates. A figure of hazards ratios for all covariates is then provided to interpret the effect of different covariates.

6. Reference

Aalen, O. O. (1989). A linear regression model for the analysis of life times. *Statistics in Medicine, 8(4)*, 907-925.

Beck, T., & Levine, R. (2004). Stock markets, banks, and growth: Panel evidence. *J. Bank. Financ., 28*(3), 423-442. doi:10.1016/s0378-4266 (02)00408-9

Chen, X., & Zhao, L. (2022). Cloud-based divide-and-conquer algorithm for high-dimensional survival analysis. *Computational Statistics & Data Analysis, 171*, 107406.

Cox, D.R. (1972, 1975). Regression Models and Life-Tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, *34(2)*, pp.187–202. doi:10.1111/j.2517-6161.1972.tb00899.x.

Dormuth, I., Liu, T., Xu, J., Yu, M., Pauly, M., & Ditzhaus, M. (2022). Which test for crossing survival curves? A user's guideline. *BMC Med. Res. Methodol., 22*(1). doi:10.1186/s12874-022-01520-0

Huang, Y., & Wang, H. (2012). A new approach to semiparametric regression analysis of survival data. *Statistical Methods in Medical Research*, *21(1)*, 55-68.

Kumar, N., Qi, S.-A., Kuan, L.-H., Sun, W., Zhang, J., & Greiner, R. (2022). Learning accurate personalized survival models for predicting hospital discharge and mortality of COVID-19 patients. *Sci. Rep., 12*(1). doi:10.1038/s41598-022-08601-6

Kvaløy, J. T., & Lindqvist, B. H. (2003). Estimation and inference in nonparametric cox-models: Time transformation methods. *Comput. Stat., 18*(2), 205-221. doi:10.1007/s001800300141

Liu, J., & Li, H. (2018). A divide-and-conquer method for efficient computation of the LASSO penalty. *Journal of Computational and Graphical Statistics, 27(3)*, 550-564.

Lin, D. Y., & Ying, Z. (1994). Semiparametric and nonparametric regression analysis of survival data. *Statistical Science*, *9(1)*, 32-47.

Li, R., & Zhao, L. C. (2020). Penalized likelihood estimation in high-dimensional Cox regression models. *Journal of the American Statistical Association, 115(529)*, 171-184.

Menelaos, Pavlou, Gareth, Ambler, Shaun, Seaman, Maria, DeIorio and Omar Rumana, Z. (2016, March). Review and evaluation of penalised regression methods for risk prediction in lowdimensional data with few events. Statist. Med. 35(7), 1159–1177.

Nemati, M., Ansary, J., & Nemati, N. (2020). Machine-learning approaches in COVID-19 survival analysis and discharge-time likelihood prediction using clinical data. *Patterns (N. Y.), 1*(5), 100074. doi:10.1016/j.patter.2020.100074

Rahmat, F., Zulkafli, Z., Ishak, A. J., Abdul Rahman, R. Z., Stercke, S. D., Buytaert, W., . . . Ismail, M. (2024). Supervised feature selection using principal component analysis. *Knowl. Inf. Syst., 66*(3), 1955-1995. doi:10.1007/s10115-023-01993-5

Wang, Y., Hong, C., Palmer, N., Di, Q., Schwartz, J., Kohane, I., Cai, T. (2019). A Fast Divide-and-Conquer Sparse Cox Regression. Biostatistics.

Wang, H. and Leng, C. (2007). Unified LASSO Estimation by Least Squares Approximation. *Journal of the American Statistical Association*, *102(479)*, pp.1039–1048. doi:10.1198/016214507000000509.

Wei, L. J. (1992). The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. *The Statistician*, *41(3)*, 187-194.

Wang, H., & Li, G. (2021). Machine learning approaches for survival analysis: A review. *Journal of the American Statistical Association, 116(536)*, 1645-1660.

Wang, H., Huang, Y., & Chen, S. X. (2018). Regularized semiparametric regression for high-dimensional survival data. *Statistics in Medicine, 37(7)*, 1130-1146.

Xu, B., Kraemer, M. U. G., Xu, B., Gutierrez, B., Mekaru, S., Sewalk, K., . . . Kraemer, M. (2020). Open access epidemiological data from the COVID-19 outbreak. *Lancet Infect. Dis., 20*(5), 534. doi:10.1016/s1473-3099(20)30119-5

Zhang, H.H. and Lu, W. (2007). Adaptive Lasso for Cox's proportional hazards model. *Biometrika*, *94(3)*, pp.691–703. doi:10.1093/biomet/asm037.

Zhang, M. Q., & Davidian, M. (2001). What is the accelerated failure time model?. *In Encyclopedia of Biopharmaceutical Statistics (pp. 5-10).* CRC Press.

Zhang, H., & Liu, Y. (2019). Bayesian Cox proportional hazards model with a mixture of priors. *Statistical Methods in Medical Research, 28(4)*, 1430-1444.

Zhao, L., & Liu, J. (2020). Distributed divide-and-conquer algorithm for large-scale Cox regression. *Journal of the American Statistical Association, 115(530)*, 1101-1113.

7. Appendix

Table 6: the name and definitions & types of each variable

Variable	definition	type
Status	Discharge then Status=1; Otherwise Status=0	binary
Time	Survival Time	numeric
Age	Age of Individual	numeric
Sex	Male then Sex=1; Otherwise Sex=0	binary
chronic_disease	Existence of chronic disease then chronic_disease=1; Otherwise chronic_disease=0	binary
density	Population / (Territorial Areas)	numeric
GDP	GDP (based on PPP theory) of a country	numeric
Per_GDP	GDP / (population of a country)	numeric
Symptoms_conjunctivitis	Existence of conjunctivitis then Symptoms_conjunctivitis=1； otherwise Symptoms_conjunctivitis=0	binary
Symptoms_discomfort	Existence of discomfort then Symptoms_discomfort =1； otherwise Symptoms_ discomfort =0	binary
Symptoms_cough	Existence of cough then Symptoms_ cough =1； otherwise Symptoms_ cough =0	binary
Symptoms_ shortness_of_breath	Existence of shortness of breath then Symptoms_shortness_of_breath =1； otherwise Symptoms_shortnes_of_breath =0	binary
Symptoms_headache	Existence of headache then Symptoms_headache =1； otherwise Symptoms_headache =0	binary
Symptoms_diarrhea	Existence of diarrhea then Symptoms_diarrhea =1； otherwise Symptoms_diarrhea =0	binary
Symptoms_sore_throat	Existence of sore throat then Symptoms_sore_throat =1； otherwise Symptoms_sore_throat =0	binary
Symptoms_chills	Existence of chills then Symptoms_chills =1； otherwise Symptoms_chills =0	binary
symptoms_fatigue	Existence of fatigue then symptoms_fatigue=1； otherwise symptoms_fatigue=0	binary
symptoms_malaise	Existence of malaise then symptoms_malaise =1； otherwise symptoms_malaise =0	binary
Location_x	Transferred Coordinates	numeric
Location_y	Transferred Coordinates	numeric
Location_z	Transferred Coordinates	numeric

report_update5.pdf

ORIGINALITY REPORT

13%	9%	11%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	oup.silverchair-cdn.com Internet Source	3%
2	Wen Su, Guosheng Yin, Jing Zhang, Xingqiu Zhao. "Divide and conquer for accelerated failure time model with massive time-to-event data", Canadian Journal of Statistics, 2022 Publication	1%
3	Yan Wang, Chuan Hong, Nathan Palmer, Qian Di, Joel Schwartz, Isaac Kohane, Tianxi Cai. "A fast divide-and-conquer sparse Cox regression", Biostatistics, 2021 Publication	1%
4	www.sparkasse-re.de Internet Source	1%
5	arxiv.org Internet Source	1%
6	dash.harvard.edu Internet Source	<1%
7	www.nature.com Internet Source	<1%

8	Hernandez Tenorio, Rommy. "Characterization of porous nickel-titanium alloys for medical applications", Proquest, 20111003 Publication	<1 %
9	Hunter, David R, Donald St P Richards, and James L Rosenberger. "Iterative Conditional Maximization Algorithm for Nonconcave Penalized Likelihood", Nonparametric Statistics and Mixture Models, 2011. Publication	<1 %
10	Submitted to RMIT University Student Paper	<1 %
11	www.mdpi.com Internet Source	<1 %
12	Ahmed, S. Ejaz, and S.M. Enayetur Raheem. "Shrinkage and absolute penalty estimation in linear regression models", Wiley Interdisciplinary Reviews Computational Statistics, 2012. Publication	<1 %
13	rucore.libraries.rutgers.edu Internet Source	<1 %
14	theses.lib.polyu.edu.hk Internet Source	<1 %
15	Howard N. Gutnick, Ralph St. John. "A Model for Predicting Clinically Relevant Group	<1 %

	Differences of Open-Response Tests", Journal of Speech, Language, and Hearing Research, 1982 Publication	
16	Submitted to University of Sunderland Student Paper	<1 %
17	hdl.handle.net Internet Source	<1 %
18	www.zbw.eu Internet Source	<1 %
19	Submitted to University of New South Wales Student Paper	<1 %
20	Wang, Wei. "The Divide-and-Combine Approaches for Multivariate Survival Analysis and Multistate Survival Analysis in Big Data.", Rutgers The State University of New Jersey, School of Graduate Studies, 2021 Publication	<1 %
21	Wei Wang, Shou-En Lu, Jerry Q. Cheng, Minge Xie, John B. Kostis. "Multivariate survival analysis in big data: A divide-and-combine approach", Biometrics, 2021 Publication	<1 %
22	www.frontiersin.org Internet Source	<1 %

23	Korlepara, Piyush Kumar. "Fuzzy and Probabilistic Rule-Based Approaches to Identify Fault Prone Files", The University of Western Ontario (Canada), 2023 Publication	<1%
24	Hao Liu, Jing Qin. "Semiparametric Probit Models with Univariate and Bivariate Current-status Data", Biometrics, 2018 Publication	<1%
25	dc.tsinghuajournals.com Internet Source	<1%
26	Castellano, G.. "Knowledge discovery by a neuro-fuzzy modeling framework", Fuzzy Sets and Systems, 20050101 Publication	<1%
27	www.retecivica.novi-ligure.al.it Internet Source	<1%
28	www.ssa.gov Internet Source	<1%
29	Zhixuan Fu, Chirag R. Parikh, Bingqing Zhou. "Penalized variable selection in competing risks regression", Lifetime Data Analysis, 2016 Publication	<1%
30	bora.uib.no Internet Source	<1%

31	Pomsuwan, Tossapol. "New Variants of Random Forest-Based Methods for Survival Analysis and Applications to Biomedical Datasets", University of Kent (United Kingdom), 2024 Publication	<1 %
32	academic.oup.com Internet Source	<1 %
33	backend.orbit.dtu.dk Internet Source	<1 %
34	cdr.lib.unc.edu Internet Source	<1 %
35	cran.ma.imperial.ac.uk Internet Source	<1 %
36	dspace2.lib.nccu.edu.tw Internet Source	<1 %
37	escholarship.org Internet Source	<1 %
38	International Journal of Quality & Reliability Management, Volume 30, Issue 4 (2013-05-27) Publication	<1 %
39	Liu, X., and D. Zeng. "Variable selection in semiparametric transformation models for right-censored data", Biometrika, 2013. Publication	<1 %

40	Nan Qiao, Wangcheng Li, Feng Xiao, Cunjie Lin. "Optimal subsampling for the Cox proportional hazards model with massive survival data", Journal of Statistical Planning and Inference, 2023 Publication	<1 %
41	Neeraj Kumar, Shi-ang Qi, Li-Hao Kuan, Weijie Sun, Jianfei Zhang, Russell Greiner. "Learning accurate personalized survival models for predicting hospital discharge and mortality of COVID-19 patients", Scientific Reports, 2022 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		