Computational Statistics
https://doi.org/10.1007/s00180-024-01496-z

ORIGINAL PAPER

PDAS: a Newton-type method for L_0 regularized accelerated failure time model

Ning Su[1] · **Yanyan Liu**[1] · **Lican Kang**[1]

Received: 10 May 2023 / Accepted: 29 March 2024

Abstract

Regularization methods are commonly utilized in survival analysis to address variable selection and estimation problems. Although most of the penalties can be regarded as variations of L_0 regularization to handle computational challenges, they may not always be efficient or effective in sparse recovery scenarios with massive amounts of data. To address this concern, this paper proposes a method for L_0 regularized estimation in the high-dimensional accelerated failure time (AFT) model, called the Primal Dual Active Set (PDAS) algorithm. Our approach introduces a tuning parameter to select active sets based on primal and dual information and performs root finding using the Karush-Kuhn-Tucker (KKT) conditions. To generate a sequence of solutions iteratively, this work also presents a sequential Primal Dual Active Set (SPDAS) algorithm that incorporates the PDAS algorithm in each iteration. Our approach can be classified as a Newton-type method to address the L_0 regularization problem directly. Extensive analysis, including simulations and real data studies, demonstrates that our approach provides competitive performance in terms of computational efficiency and predictive accuracy compared with existing methods for sparse recovery.

Keywords L_0 regularization · KKT conditions · Sparse recovery · High dimension · AFT model

✉ Lican Kang
kanglican@whu.edu.cn

Ning Su
suning@whu.edu.cn

Yanyan Liu
liuyy@whu.edu.cn

[1] School of Mathematics and Statistics, Wuhan University, Wuhan, China

Published online: 21 April 2024

1 Introduction

Survival analysis (Klein and Moeschberger 2003; Fleming and Harrington 2011) is a statistical technique used to analyze time-to-event data such as the time to death, relapse of a disease, or failure of a machine. A commonly used method in survival analysis is the accelerated failure time (AFT) modeling technique (Koul et al. 1981; Wei 1992), which can estimate the effect of one or more covariates on the survival time of an event. AFT models assume a linear relationship between the logarithm of the survival time and covariates. AFT models are a critical alternative to the Cox proportional hazard model (Cox 1972) when the proportional hazard assumption is violated. AFT models can model a wide range of survival curves. However, the AFT model has a fundamental drawback of being unable to provide a meaningful solution for large indefinite linear systems through the least square method. In the presence of high-dimensional data, variable selection is crucial to constructing a parsimonious and interpretable model.

The problem of variable selection is a well-established and constantly evolving challenge in the field of data analysis. Regularization techniques have been widely used to address this problem, particularly in large-scale regression models, to identify important variables for predicting future responses. In the past two decades, numerous penalty methods have been proposed to address this issue. One of the most popular techniques is the least absolute shrinkage and selection operator (Lasso) (Tibshirani 1996), which uses constraints on the L_1 norm to formulate a convex optimization problem. The L_1 penalty was introduced into the AFT model by Huang et al. (2006) based on the Stute estimator. Despite its computational feasibility, the Lasso may produce inconsistent estimates without appropriate conditions on the covariates, as shown by Zhang and Huang (2008). Another line of regularization involves non-convex penalties, such as the smoothly clipped absolute deviation (SCAD) (Fan and Li 2001), the minimax concave penalty (MCP) (Zhang 2010), and L_0 penalty (Huang et al. 2018a). Johnson (2008) applied the SCAD penalty in the AFT model based on a rank-based estimator, and Johnson et al. (2008) utilized the SCAD and Buckley-James estimator to achieve variable selection in a semi-parametric model. Hu and Chai (2013) considered the penalized weighted least square estimation in the AFT model based on the MCP. Cheng et al. (2022) extended the support detection and root finding algorithm (Huang et al. 2018a) from linear regression models to the AFT model with L_0 penalization.

In this work, we investigate the L_0 regularized AFT model in high-dimensional and sparse cases. The L_0 penalty, which is a discontinuous function, is used to penalize the number of non-zero components, making it challenging to solve the problem in the high-dimensional context. To tackle this problem, inspired by the unified primal dual set algorithm (Huang et al. 2020) and a semismooth Newton algorithm for pathwise optimization (Huang et al. 2018b), we develop a novel method based on the Karush-Kuhn-Tucker (KKT) conditions that employ a feasible iterative algorithm to search for a steady solution to the optimization problem under a fixed tuning parameter. We refer to our method as PDAS (primal dual active set) algorithm. We further introduce a sequential PDAS algorithm (abbreviated

as SPDAS), in which the PDAS algorithm is integrated into each iteration step, to generate a sequence of solutions, rather than minimizing the L_0 penalized least squares criterion directly. We use a natural approach to adjust the tuning parameter to construct the SPDAS algorithm, and we use the high-dimensional Bayesian information criterion proposed by Wang et al. (2013) to determine the optimal tuning parameter of the solution path generated by the SPDAS algorithm. We also theoretically prove that the PDAS algorithm is essentially a Newton-type method, even though the optimization problem is non-convex and non-smooth.

The structure of this paper is arranged as follows. In Sect. 2, we weight the loss function through the Kaplan-Meier estimator and simplify the L_0 penalized least squares criterion to formulate an optimization target for the AFT model. Section 3 focuses on the development of the PDAS algorithm based on KKT conditions and the demonstration of its classification as a Newton-type method. In Sect. 4, we establish SPDAS algorithm to obtain a solution path. In Sect. 5, we assess the performance of our method through numerical simulations and real data analysis. Proofs for all lemmas and theorems are provided in Appendix A.

2 Methodology

We first introduce some notations used throughout the paper. We denote $\|\boldsymbol{\xi}\|_q = (\sum_{i=1}^p |\xi_i|^q)^{\frac{1}{q}}$ as q ($q \in [1, \infty]$) norm of a vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_p)^\top \in \mathbb{R}^p$. Let $S = \{1, 2, \ldots, p\}$. For any set $A \subset S$ with size $|A|$, denote $\boldsymbol{\xi}_A = (\xi_i, i \in A) \in \mathbb{R}^{|A|}$. Denote $\mathbf{X}_A = (\mathbf{X}_j, j \in A) \in \mathbb{R}^{n \times |A|}$, where $\mathbf{X}_j$ is jth column of the design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. $\lfloor \cdot \rfloor$ means floor operation that rounds a real number down to the nearest integer less than or equal to the given number.

In the AFT model, the relationship between the covariates and the survival time is described by the following equation:

$$\ln(T_i) = \mathbf{x}_i^\top \boldsymbol{\beta}_0 + \epsilon_i, \quad i = 1, \ldots, n,$$

where n is the sample size, $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ is the underlying regression coefficient vector, and ϵ_i's are random error terms. We consider right censored survival time in this work. Let C_i denote the potential censoring time. We have access to data $\{Y_i, \delta_i, \mathbf{x}_i\}$, where $Y_i := \min\{\ln(T_i), \ln(C_i)\}$, and $\delta_i := 1_{\{T_i \le C_i\}}$ represents the censoring indicator. Let $Y_{(1)}, \ldots, Y_{(n)}$ denote the order statistics of Y_i's, and $\delta_{(1)}, \ldots, \delta_{(n)}$ denote the associated censoring indicators, $\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(n)}$ are the associated covariates. We introduce a weight function

$$\begin{aligned} w_{(1)} &= \frac{\delta_{(1)}}{n}, \\ w_{(i)} &= \frac{\delta_{(i)}}{n-i+1} \cdot \prod_{j=1}^{i-1} \left(\frac{n-j}{n-j+1} \right)^{\delta_{(j)}}, \ i = 2, \ldots, n. \end{aligned}$$

$w_{(i)}$'s are the jumps in Kaplan-Meier estimator (Kaplan and Meier 1958) based on $(Y_{(i)}, \delta_{(i)})$, $i = 1, \ldots, n$. Therefore, the estimator $\hat{\boldsymbol{\beta}}$ is a minimizer of

$$\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^{n} w_{(i)} \left(Y_{(i)} - \mathbf{X}_{(i)}^{\top} \boldsymbol{\beta} \right)^2.$$

We directly consider the L_0 regularized method for variable selection and estimation in the high-dimensional AFT model based on the weighted least squares loss. The L_0 penalized estimator is given by

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\operatorname{argmin}} \, \mathcal{L}(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_0, \tag{2.1}$$

where $\lambda \geq 0$ is a tuning parameter, and $\|\boldsymbol{\beta}\|_0$ denotes the number of nonzero elements of $\boldsymbol{\beta}$.

To be precise, the weighted least squares loss is rewritten as a standard least squares loss as follows. Let the design matrix be $\mathbf{X} = \left(\mathbf{X}_{(1)}, \ldots, \mathbf{X}_{(n)} \right)^{\top}$ and response variable be $\mathbf{y} = \left(Y_{(1)}, \ldots, Y_{(n)} \right)^{\top}$. Define

$$\begin{aligned} \mathbf{X}^{\diamond} &= \operatorname{diag}\left(\sqrt{w_{(1)}}, \ldots, \sqrt{w_{(n)}} \right) \cdot \mathbf{X}, \\ \tilde{\mathbf{y}} &= \operatorname{diag}\left(\sqrt{w_{(1)}}, \ldots, \sqrt{w_{(n)}} \right) \cdot \mathbf{y}. \end{aligned}$$

Without loss of generality, assume that $\|\mathbf{X}_j^{\diamond}\|_2 > 0$, $j = 1, \ldots, p$ hold throughout this paper, where $\mathbf{X}_j^{\diamond}$ is the jth column of $\mathbf{X}^{\diamond}$. Let

$$\Lambda = \operatorname{diag}\left(\frac{\sqrt{n}}{\|\mathbf{X}_1^{\diamond}\|_2}, \ldots, \frac{\sqrt{n}}{\|\mathbf{X}_p^{\diamond}\|_2} \right).$$

Define $\boldsymbol{\xi} = \Lambda^{-1} \boldsymbol{\beta}$ and $\tilde{\mathbf{X}} = \mathbf{X}^{\diamond} \Lambda$. Then each column of $\tilde{\mathbf{X}}$ is $\sqrt{n}$-length and supp($\boldsymbol{\xi}$)=supp($\boldsymbol{\beta}$), where supp($\boldsymbol{\beta}$)=$\{j : \beta_j \neq 0, j = 1, \ldots, p\}$. Let

$$\tilde{\mathcal{L}}(\boldsymbol{\xi}) = \frac{1}{2n} \left\| \tilde{\mathbf{y}} - \tilde{\mathbf{X}} \boldsymbol{\xi} \right\|_2^2.$$

Define

$$\boldsymbol{\xi}^* \in \underset{\boldsymbol{\xi} \in \mathbb{R}^p}{\operatorname{argmin}} \, \tilde{\mathcal{L}}(\boldsymbol{\xi}) + \lambda \|\boldsymbol{\xi}\|_0. \tag{2.2}$$

Then the L_0 penalized estimator of $\boldsymbol{\beta}^*$ defined in (2.1) can be obtained as $\boldsymbol{\beta}^* = \Lambda \boldsymbol{\xi}^*$.

3 Derivation of PDAS algorithm

We firstly obtain the KKT condition of (2.2), a necessary condition for the global solution, which is a basis for deriving the PDAS algorithm and shown in the following lemma.

Lemma 3.1 *If $\boldsymbol{\xi}^*$ is the global minimizer of (2.2), then it satisfies*

$$\begin{cases} \mathbf{d}^* = \tilde{\mathbf{X}}^\top(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}^*)/n, \\ \boldsymbol{\xi}^* = \Gamma_\lambda(\boldsymbol{\xi}^* + \mathbf{d}^*), \end{cases} \tag{3.1}$$

where the ith element of $\Gamma_\lambda(\cdot)$ is defined by

$$(\Gamma_\lambda(\boldsymbol{\xi}))_i = \begin{cases} 0, & |\boldsymbol{\xi}_i| \le \sqrt{2\lambda}, \\ \boldsymbol{\xi}_i, & |\boldsymbol{\xi}_i| > \sqrt{2\lambda}. \end{cases} \tag{3.2}$$

Conversely, if $\boldsymbol{\xi}^*$ and $\mathbf{d}^*$ satisfy (3.1) and (3.2), then $\boldsymbol{\xi}^*$ is a local minimizer of (2.2), and $\boldsymbol{\beta}^* = \Lambda\boldsymbol{\xi}^*$ is a local minimizer of (2.1).

Denote $A^* = \mathrm{supp}(\boldsymbol{\xi}^*)$ and $I^* = (A^*)^c$. By the definition of $\boldsymbol{\xi}^*$ and $\mathbf{d}^*$ in (3.1), and $\Gamma_\lambda(\cdot)$ in (3.2), we have

$$A^* = \{i \in S : |\xi_i^* + d_i^*| > \sqrt{2\lambda}\}, \quad I^* = \{i \in S : |\xi_i^* + d_i^*| \le \sqrt{2\lambda}\},$$

and

$$\begin{cases} \boldsymbol{\xi}^*_{I^*} = \mathbf{0} \\ \mathbf{d}^*_{A^*} = \mathbf{0} \\ \boldsymbol{\xi}^*_{A^*} = (\tilde{\mathbf{X}}^\top_{A^*}\tilde{\mathbf{X}}_{A^*})^{-1}\tilde{\mathbf{X}}^\top_{A^*}\mathbf{y} \\ \mathbf{d}^*_{I^*} = \tilde{\mathbf{X}}^\top_{I^*}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{A^*}\boldsymbol{\xi}^*_{A^*})/n \\ \boldsymbol{\beta}^* = \Lambda\boldsymbol{\xi}^*. \end{cases}$$

The main idea of PDAS algorithm is to mimic the above display iteratively. To be precise, let $\{\boldsymbol{\xi}^k, \mathbf{d}^k\}$ be the kth iteration of PDAS algorithm, the corresponding active set A^k and inactive set I^k can be calculated by

$$A^k = \{i \in S : |\xi_i^k + d_i^k| > \sqrt{2\lambda}\}, \quad I^k = \{i \in S : |\xi_i^k + d_i^k| \le \sqrt{2\lambda}\}.$$

Then, we update a new approximation pair $\{\boldsymbol{\xi}^{k+1}_{I^k}, \mathbf{d}^{k+1}_{A^k}, \boldsymbol{\xi}^{k+1}_{A^k}, \mathbf{d}^{k+1}_{I^k}\}$ as follows:

$$\begin{cases} \boldsymbol{\xi}^{k+1}_{I^k} = \mathbf{0} \\ \mathbf{d}^{k+1}_{A^k} = \mathbf{0} \\ \boldsymbol{\xi}^{k+1}_{A^k} = (\tilde{\mathbf{X}}^\top_{A^k}\tilde{\mathbf{X}}_{A^k})^{-1}\tilde{\mathbf{X}}^\top_{A^k}\mathbf{y} \\ \mathbf{d}^{k+1}_{I^k} = \tilde{\mathbf{X}}^\top_{I^k}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{A^k}\boldsymbol{\xi}^{k+1}_{A^k})/n \\ \boldsymbol{\beta}^{k+1} = \Lambda\boldsymbol{\xi}^{k+1}. \end{cases} \tag{3.3}$$

We summarize the above derivation of PDAS in the following Algorithm 1.

Algorithm 1 PDAS Algorithm

1: Input: $\boldsymbol{\xi}^0$, $\mathbf{d}^0$, λ, K
2: **for** $k = 0, 1, \ldots, K$, **do**
3: $A^k = \{j \in S : |\xi_j^k + d_j^k| > \sqrt{2\lambda}\}$, $I^k = (A^k)^c$.
4: $\boldsymbol{\xi}_{I^k}^{k+1} = \mathbf{0}$.
5: $\mathbf{d}_{A^k}^{k+1} = \mathbf{0}$.
6: $\boldsymbol{\xi}_{A^k}^{k+1} = (\tilde{\mathbf{X}}_{A^k}^\top \tilde{\mathbf{X}}_{A^k})^{-1} \tilde{\mathbf{X}}_{A^k}^\top \mathbf{y}$.
7: $\mathbf{d}_{I^k}^{k+1} = \tilde{\mathbf{X}}_{I^k}^\top (\tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{A^k} \boldsymbol{\xi}_{A^k}^{k+1})/n$.
8: **if** $A^k = A^{k+1}$ or $k \geq K$, **then**
9: Stop and denote the items in the last iteration as $\boldsymbol{\xi}_{\widehat{A}}$, $\boldsymbol{\xi}_{\widehat{I}}$, $\mathbf{d}_{\widehat{A}}$, $\mathbf{d}_{\widehat{I}}$.
10: **else**
11: $k = k + 1$
12: **end if**
13: **end for**
14: Output: $\widehat{\boldsymbol{\xi}}(\lambda) = \left(\boldsymbol{\xi}_{\widehat{A}}^\top, \boldsymbol{\xi}_{\widehat{I}}^\top\right)^\top$, $\widehat{\mathbf{d}}(\lambda) = (\mathbf{d}_{\widehat{A}}^\top, \mathbf{d}_{\widehat{I}}^\top)^\top$ and $\widehat{\boldsymbol{\beta}}(\lambda) = \Lambda \widehat{\boldsymbol{\xi}}(\lambda)$ as the estimation at λ.

In line 3 of Algorithm 1, the active set A^k and inactive set I^k are determined by combining the primal and dual information in the previous step. In line 6 of Algorithm 1, a least square estimator restricted to the selected active set A^k is calculated to update $\boldsymbol{\xi}$. The stop criterion in line 8 indicates that the proposed PDAS algorithm will terminate if the estimated support set at current iteration coincides with the previous one or the iteration number exceeds the given maximum tolerance number K. Next we show that the proposed PDAS algorithm can be interpreted as a generalized Newton method for finding the root of the KKT system (3.1).

3.1 PDAS as a generalized Newton algorithm

In this subsection, we derive the PDAS algorithm from the Newton-type method for finding roots of the KKT system (3.1) even though the original minimization problem (2.2) is non-convex and non-smooth. Let $\mathbf{w} = \begin{pmatrix} \boldsymbol{\xi} \\ \mathbf{d} \end{pmatrix}$ and $F(\mathbf{w}) = \begin{bmatrix} F_1(\mathbf{w}) \\ F_2(\mathbf{w}) \end{bmatrix}$: $\mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^{2p}$, where

$$F_1(\mathbf{w}) = \boldsymbol{\xi} - \Gamma_\lambda(\boldsymbol{\xi} + \mathbf{d}), \quad F_2(\mathbf{w}) = n\mathbf{d} - \tilde{\mathbf{X}}^\top(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}).$$

The next theorem shows that the PDAS algorithm is actually a form of the generalized Newton method. This theoretically illustrates that our proposed method is superior to some first-order optimization algorithms (Breheny and Huang 2011;

Friedman et al. 2010; Simon et al. 2011), and it is also reflected in our simulation studies, see Sect. 5.

Theorem 3.1 *The iteration in* (3.3) *is equivalent to*

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \left(\mathcal{H}^k\right)^{-1} F\left(\mathbf{w}^k\right), \tag{3.4}$$

where

$$\mathcal{H}^k = \begin{pmatrix} \mathcal{H}_1^k & \mathcal{H}_2^k \\ \tilde{\mathbf{X}}_k^\top \tilde{\mathbf{X}}_k & n\mathbf{I} \end{pmatrix}, \quad \tilde{\mathbf{X}}_k^\top \tilde{\mathbf{X}}_k = \begin{pmatrix} \tilde{\mathbf{X}}_{A^k}^T \tilde{\mathbf{X}}_{A^k} & \tilde{\mathbf{X}}_{A^k}^T \tilde{\mathbf{X}}_{I^k} \\ \tilde{\mathbf{X}}_{I^k}^T \tilde{\mathbf{X}}_{A^k} & \tilde{\mathbf{X}}_{I^k}^T \tilde{\mathbf{X}}_{I^k} \end{pmatrix},$$
$$\mathcal{H}_1^k = \begin{pmatrix} \mathbf{0}_{A^k A^k} & \mathbf{0}_{A^k I^k} \\ \mathbf{0}_{I^k A^k} & \mathbf{I}_{I^k I^k} \end{pmatrix}, \quad \mathcal{H}_2^k = \begin{pmatrix} -\mathbf{I}_{A^k A^k} & \mathbf{0}_{A^k I^k} \\ \mathbf{0}_{I^k A^k} & \mathbf{0}_{I^k I^k} \end{pmatrix}.$$

Remark 3.1 In Theorem 3.1, it is demonstrated that the PDAS algorithm can be considered as a generalized form of the Newton algorithm. It is crucial to emphasize the invertibility of the Hessian matrix $\mathcal{H}^k$. Specifically, the rank of $\mathcal{H}^k$ is primarily determined by $\tilde{\mathbf{X}}_{A^k}^T \tilde{\mathbf{X}}_{A^k}$. In high-dimensional sparse models, the small size of the active set ensures the invertibility of $\mathcal{H}^k$. Furthermore, to address situations where $\mathcal{H}^k$ may be non-invertible, we can introduce a small perturbation of the identity matrix to render $\mathcal{H}^k$ invertible.

4 Sequential PDAS algorithm

The PDAS algorithm only yields an estimator for a fixed tuning parameter λ, but typically, we are more interested in obtaining a solution path. To address this, we propose a Sequential PDAS (SPDAS) algorithm in this section. However, two practical issues need to be considered to obtain the desirable solution path: determining the initial value $(\boldsymbol{\xi}^0, \mathbf{d}^0)$ in Algorithm 1 and selecting an appropriate regularization parameter λ.

To address the first issue, we use Lemma 3.1 to set the initial value $\lambda_0 = \frac{\|\tilde{\mathbf{X}}^\top \tilde{\mathbf{y}}/n\|_\infty^2}{2}$, which ensures that $\hat{\boldsymbol{\xi}}(\lambda_0) = \mathbf{0}$ and $\hat{\mathbf{d}}(\lambda_0) = \frac{\tilde{\mathbf{X}}^\top \tilde{\mathbf{y}}}{n}$. We then use a decreasing sequence of regularization parameters $\lambda_m = \lambda_0 \alpha^m$, $\alpha \in (0, 1)$, and apply Algorithm 1 on the sequence λ_m with the solution $(\hat{\boldsymbol{\xi}}(\lambda_m), \hat{\mathbf{d}}(\lambda_m))$ being the initial guess for the λ_{m+1} -problem.

To address the second issue and obtain the optimal λ, we can use a data-driven method such as cross-validation, modified Bayesian information criterion (Kim et al. 2012; Wang et al. 2013), or the voting method (Huang et al. 2020) without any additional computational overhead. The SPDAS algorithm combines the PDAS algorithm with the warm-start and continual strategy to provide good initial guesses and simultaneously output a solution path. The algorithm is described in Algorithm 2. We can stop the SPDAS algorithm and obtain a solution path until $\|\hat{\boldsymbol{\xi}}(\lambda_m)\|_0 > \lfloor \frac{n}{\log p} \rfloor$ for some m.

Consequently, the SPDAS algorithm combines the PDAS algorithm with the warm-start and continual strategy, providing good initial estimates while simultaneously generating a solution path. See Algorithm 2 for a description of the SPDAS algorithm.

Algorithm 2 SPDAS Algorithm

1: Input: $\widehat{\boldsymbol{\xi}}(\lambda_0) = \mathbf{0}$, $\widehat{\mathbf{d}}(\lambda_0) = \tilde{\mathbf{X}}^\top \tilde{\mathbf{y}}/n$, $\lambda_0 = \frac{\|\tilde{\mathbf{X}}^\top \tilde{\mathbf{y}}/n\|_\infty^2}{2}$, M, α.
2: **for** $m = 1, \ldots, M$ **do**
3: $\quad \lambda = \lambda_m = \lambda_0 \alpha^m$, $\boldsymbol{\xi}^0 = \widehat{\boldsymbol{\xi}}(\lambda_{m-1})$, $\mathbf{d}^0 = \widehat{\mathbf{d}}(\lambda_{m-1})$.
4: $\quad$ Run Algorithm 1 to get $\widehat{\boldsymbol{\xi}}(\lambda_m)$ and $\widehat{\mathbf{d}}(\lambda_m)$.
5: $\quad$ if $\|\widehat{\boldsymbol{\xi}}(\lambda_m)\|_0 > \lfloor \frac{n}{\log p} \rfloor$, stop.
6: **end for**
7: Output: $\left\{\widehat{\boldsymbol{\beta}}(\lambda_0), \widehat{\boldsymbol{\beta}}(\lambda_1), \ldots, \widehat{\boldsymbol{\beta}}(\lambda_M)\right\} = \Lambda \cdot \left\{\widehat{\boldsymbol{\xi}}(\lambda_0), \widehat{\boldsymbol{\xi}}(\lambda_1), \ldots, \widehat{\boldsymbol{\xi}}(\lambda_M)\right\}$.

Remark 4.1 The stopping rule in line 5 of Algorithm 2 results in the termination of the SPDAS algorithm when the number of non-zero elements within an iteration exceeds a predetermined threshold. This threshold is determined by a fixed quantity denoted as $\frac{n}{\log(p)}$. This approach is particularly suitable when $n \ll p$. Such an approach enables the estimation of sparsity for L_1 penalties with an order of magnitude of $O\left(\frac{n}{\log(p)}\right)$ (Candes et al. 2006; Candes and Tao 2006). Moreover, in the context of high-dimensional linear regression with non-convex penalty, the L_2 error bound between the estimated and true coefficient vectors can be approximated as $O\left(\sqrt{\|\boldsymbol{\beta}_0\|_0 \frac{\log(p)}{n}}\right)$ (Zhang and Zhang 2012). Alternatively, an alternative stopping rule as proposed by Fan and Lv (2008), such as $O\left(\frac{n}{\log(n)}\right)$, can be adopted. This particular stopping rule establishes an upper bound on the maximum feasible model that can be reliably estimated given a specific sample size n.

5 Simulation studies and real data analysis

In this section, we thoroughly evaluate the SPDAS algorithm, implemented using the R programming language. The R code of PDAS and SPDAS algorithms are available at https://github.com/suningningning/SPDAS.git. We also compare its performance with other widely used penalized methods, Lasso, MCP, and SCAD. We utilize specific R packages, *ncvreg* (Breheny and Huang 2011) and *glmnet* (Friedman et al. 2010; Simon

Table 1 100 independent replications with $n = 300$, $p = 5000$, $\rho = 0.2 : 0.2 : 0.8$, $\sigma = 1$, $R = 10$, CR $= 0.25, 0.5$, $T = 15$ and $\mathbf{X}$ follows (I)

ρ	CR	Method	AE	RE(10^{-2})	RP(10^{-2})	MSES	Time(s)
0.2	0.25	Lasso	0.71	17.10	0	105.75	1.93
		MCP	0.16	2.74	74	15.70	2.96
		SCAD	0.16	2.70	68	15.97	3.00
		SPDAS	0.15	2.62	100	15.00	2.40
	0.5	Lasso	1.26	30.98	0	59.47	2.39
		MCP	0.19	3.36	64	16.03	3.44
		SCAD	0.19	3.45	43	17.95	3.51
		SPDAS	0.19	3.25	99	15.01	2.92
0.4	0.25	Lasso	0.72	16.78	0	100.54	1.99
		MCP	0.15	2.67	74	15.65	3.02
		SCAD	0.15	2.66	69	15.92	3.03
		SPDAS	0.15	2.58	100	15.00	2.43
	0.5	Lasso	1.37	32.89	0	52.45	2.42
		MCP	0.18	3.37	63	16.44	3.44
		SCAD	0.18	3.38	44	17.98	3.43
		SPDAS	0.18	3.28	98	15.02	3.16
0.6	0.25	Lasso	0.71	16.83	0	102.36	1.89
		MCP	0.15	2.75	79	15.40	2.97
		SCAD	0.16	2.76	72	15.67	2.98
		SPDAS	0.15	2.69	100	15.00	2.60
	0.5	Lasso	1.32	31.67	0	56.46	2.43
		MCP	0.18	3.29	63	15.88	3.42
		SCAD	0.19	3.34	44	17.34	3.44
		SPDAS	0.18	3.23	99	15.01	3.99
0.8	0.25	Lasso	0.72	16.78	0	100.54	2.56
		MCP	0.15	2.67	74	15.65	3.56
		SCAD	0.15	2.66	69	15.92	3.59
		SPDAS	0.15	2.58	100	15.00	3.11
	0.5	Lasso	1.22	30.42	0	63.74	2.41
		MCP	0.25	4.41	57	16.21	3.42
		SCAD	0.20	3.49	45	17.52	3.46
		SPDAS	0.26	4.54	91	15.07	6.34

et al. 2011). These packages provide established implementations of Lasso, MCP, and SCAD, ensuring a fair and reliable comparison.

We consider the following AFT model:

$$\ln(T_i) = \mathbf{x}_i^\top \boldsymbol{\beta}_0 + \epsilon_i, \quad i = 1, \ldots, n,$$

where ϵ's are independent and identically distributed from $N(0, \sigma^2)$, the support A^* of parameter $\boldsymbol{\beta}_0$ is chosen uniformly from S with $|A^*| = T < n$, the non-zero elements of $\boldsymbol{\beta}_0$ are generated via $\beta_{0i} = \theta_i R^{\kappa_i}$, θ_i are i.i.d. Bernoulli random variables with parameter 0.5, κ_i are i.i.d. uniform random variables in [0, 1] and $R > 1$. The

censoring variable C_i's are generated independently from the uniform distribution $U(0, \eta)$, where η controls the censoring rate (CR). For the $n \times p$ covariate matrix $\mathbf{X}$, we consider the following two settings:

(I) Method1 The rows of $\mathbf{X}$ are independent and identically distributed from $N(0, \Sigma)$, where $\Sigma_{i,j} = \rho^{|i-j|}$ for $1 \le i, j \le p$, and ρ is the correlation parameter.
(II) We first generate a $n \times p$ random Gaussian matrix $\mathbf{Z}$ whose entries are independent and identically distributed from $N(0, 1)$. Then the covariates matrix $\mathbf{X}$ is generated with $\mathbf{X}_1 = \mathbf{z}_1$, $\mathbf{X}_p = \mathbf{z}_p$, and $\mathbf{X}_j = \mathbf{z}_j + \rho(\mathbf{z}_{j+1} + \mathbf{z}_{j-1})$, $j = 2, \ldots, p-1$. Here ρ is a measure of the correlation among covariates.

5.1 Accuracy and efficiency

To assess the performance of SPDAS relative to Lasso, MCP, and SCAD, we conduct a comprehensive comparison using several key metrics. These metrics include the average L_∞ absolute error (AE), the average L_2 relative error (RE), and the average CPU time (Time) measured in seconds. Additionally, we evaluate the support recovery capabilities of all methods, which are quantified by the mean size of the estimated supports (MSES) and the exact support recovery probability (RP). Let J denote the number of independent replications. The above metrics can be defined as

$$\text{AE} = \frac{1}{J}\sum_{j=1}^{J} \|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}_0\|_\infty,$$

$$\text{RE} = \frac{1}{J}\sum_{j=1}^{J} \|\hat{\boldsymbol{\beta}}^{(j)} - \boldsymbol{\beta}_0\|_2 / \|\boldsymbol{\beta}_0\|_2,$$

$$\text{RP} = \frac{1}{J}\sum_{j=1}^{J} \mathbf{1}(\hat{A}^{(j)} = A^*),$$

$$\text{MSES} = \frac{1}{J}\sum_{j=1}^{J} |\hat{A}^{(j)}|,$$

$$\text{Time} = \frac{1}{J}\sum_{j=1}^{J} t^{(j)},$$

where $\hat{\boldsymbol{\beta}}^{(j)}$ is the estimator at jth simulation, $\hat{A}^{(j)}$ is the estimated support, and $t^{(j)}$ is the jth running time. We consider the following two scenarios:

- $\mathbf{X}$ is generated according to (I), and $n = 300$, $p = 5000$, $\rho = 0.2 : 0.2 : 0.8$, $\sigma = 1$, $R = 10$, $\text{CR} = 0.25, 0.5$, $T = 15$.
- $\mathbf{X}$ is generated according to (II), and $n = 600$, $p = 10000$, $\rho = 0.2 : 0.2 : 0.8$, $\sigma = 1$, $R = 10$, $\text{CR} = 0.25, 0.5$, $T = 30$.

Table 2 1000 independent replications $n = 600$, $p = 10000$, $\rho = 0.2 : 0.2 : 0.8$, $\sigma = 1$, $R = 10$, CR $= 0.25, 0.5$, $T = 30$ and $\mathbf{X}$ follows (II)

ρ	CR	Method	AE	RE(10^{-2})	RP(10^{-3})	MSES	Time(s)
0.2	0.25	Lasso	0.42	5.49	0	155.17	6.50
		MCP	0.11	0.97	1000	30.00	12.21
		SCAD	0.14	1.07	1000	30.00	13.40
		SPDAS	0.11	0.96	1000	30.00	7.05
	0.5	Lasso	0.93	11.79	0	102.01	6.24
		MCP	0.16	1.28	1000	30.00	12.43
		SCAD	0.25	1.86	1000	30.00	13.37
		SPDAS	0.14	1.22	999	30.00	7.52
0.4	0.25	Lasso	0.39	5.03	0	148.04	6.26
		MCP	0.10	0.87	1000	30.00	12.25
		SCAD	0.12	0.97	1000	30.00	13.32
		SPDAS	0.10	0.86	1000	30.00	7.18
	0.5	Lasso	0.85	10.84	0	102.10	6.41
		MCP	0.14	1.18	999	30.01	12.18
		SCAD	0.27	1.83	999	30.00	12.58
		SPDAS	0.13	1.11	996	30.01	6.89
0.6	0.25	Lasso	0.36	4.63	0	134.47	6.87
		MCP	0.09	0.77	1000	30.00	12.18
		SCAD	0.11	0.86	1000	30.00	13.29
		SPDAS	0.09	0.76	1000	30.00	7.4
	0.5	Lasso	0.79	9.95	0	98.42	6.84
		MCP	0.13	1.08	996	30.01	11.47
		SCAD	0.24	1.62	1000	30.00	12.90
		SPDAS	0.19	1.39	975	30.07	9.71
0.8	0.25	Lasso	0.32	4.46	0	118.85	6.85
		MCP	0.08	0.66	1000	30.00	11.87
		SCAD	0.10	0.77	1000	30.00	12.53
		SPDAS	0.08	0.66	1000	30.00	7.59
	0.5	Lasso	0.73	9.22	0	93.56	6.82
		MCP	0.11	0.89	1000	30.00	11.94
		SCAD	0.23	1.50	1000	30.00	12.54
		SPDAS	0.10	0.84	999	30.00	9.40

Based on independent replications, we get the simulation results shown in the following Tables 1 - 2.

Table 1 shows the 100 independent replications result of scenario (I) with $n = 300$, $p = 5000$, $\rho = 0.2 : 0.2 : 0.8$, $\sigma = 1$, $R = 10$, CR $= 0.25, 0.5$, $T = 15$. When comparing the performance of SPDAS with that of Lasso, MCP, and SCAD, noteworthy differences emerge. SPDAS, MCP, and SCAD demonstrate superior accuracy, as indicated by smaller absolute error and relative error values. Furthermore, even under high censoring rates, SPDAS outperforms the other methods in terms of recovery probability. Concerning computational efficiency, SPDAS

manifests a higher computational time demand than Lasso, while showcasing lower computational time requirements relative to the other two methods in most scenarios. The SPDAS exhibits increased sensitivity to the correlation of covariates compared to alternative algorithms. Specifically, in scenario (I) distinguished by a pronounced correlation and a high censoring rate, the computational speed of the SPDAS algorithm is slower when compared with other methods.

Owing to the superior computational efficiency of the data generation method outlined in scenario (II) as compared to that in scenario (I), we opt to increase the number of simulations employing the second approach to 1000. Based on 1000 independent replications, Table 2 presents a numerical results of scenario (II) with the following parameter settings: $n = 600$, $p = 10000$, $\rho = 0.2 : 0.2 : 0.8$, $\sigma = 1$, $R = 10$, CR $= 0.25$, 0.5, and $T = 30$. In terms of computational efficiency, it is readily apparent that the SPDAS surpasses both the SCAD and the MCP in terms of speed. Notably, even in scenarios characterized by a high frequency of censoring, SPDAS maintains a notable advantage over MCP and SCAD in computational swiftness. However, when assessing the performance of these methods based on metrics such as relative error, and recovery probability, SPDAS exhibits a modest decrement in performance compared to the MCP specifically when the censoring rate reaches 0.5.

5.2 Influence of the model parameters

In this subsection, we consider how the model parameters, including sample size n, covariates dimension p, correlation ρ, censoring rate CR and the size of support set T influence the performance of SPDAS algorithm and other alternative methods in terms of computational speed (Time), exact support recovery probability (RP), relative error (RE) and mean size of estimated supports (MSES). Let $\mathbf{X}$ be generated according to scenario (I). The sample size n, the covariates dimension p, the correlation ρ, the censoring rate CR, the size of support set T and others are set as following:

- $n = 100 : 50 : 500$, $p = 600$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.3$, $T = 10$.
- $n = 200$, $p = 500 : 200 : 1700$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.3$, $T = 10$.
- $n = 200$, $p = 600$, $\rho = 0.1 : 0.1 : 0.9$, $\sigma = 1$, $R = 5$, CR $= 0.3$, $T = 10$.
- $n = 200$, $p = 600$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.1 : 0.1 : 0.7$, $T = 10$.
- $n = 400$, $p = 600$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.3$, $T = 5, 10 : 10 : 60$.

The evaluation measures how Time, RP, RE, and MSES change concerning n, p, ρ, CR, T based on 100 replications.

5.2.1 Influence of the sample size *n*

The top left panel of Fig. 1 presents the impact of sample size n on the estimated supports. The results indicate that SPDAS has a more stable and appropriate mean size of estimated supports. At the same time, Lasso produces significantly larger

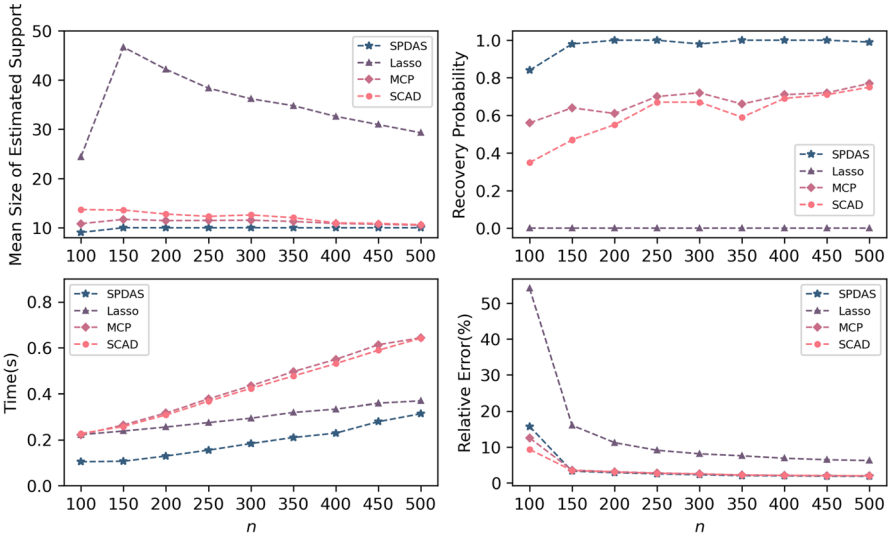

Fig. 1 The numerical results of the influence of sample size n ($n = 100 : 50 : 500$, $p = 600$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.3$, $T = 10$)

estimates, and the estimates of MCP and SCAD are slightly larger. The top right panel of Fig. 1 depicts the impact of sample size n on the recovery probability. The results reveal that SPDAS has a recovery probability close to 1, whereas Lasso fails to recover the true support, and the recovery probability of MCP and SCAD is less than 0.8. The bottom left panel of Fig. 1 shows that the average CPU time of all the algorithms increases with the growing sample size. The bottom right panel of Fig. 1 illustrates the influence of sample size n on the relative error. The numerical results suggest that SPDAS, MCP, and SCAD exhibit analogous levels of accuracy, demonstrating superior performance in comparison to Lasso.

5.2.2 Influence of the variable dimension p

The line graph depicted in Fig. 2 displays the numerical results of the impact of variable dimension p. With increasing variable dimension p, the SPDAS is observed to maintain the recovery probability at 1, while Lasso, MCP, and SCAD exhibit considerably less than 1. The top left panel and the bottom right panel of Fig. 2 suggest that SPDAS can estimate values closer to the true size of the support set and yield a smaller relative error.

5.2.3 Influence of the correlation ρ

The top left panel of Fig. 3 presents the results of the mean size of the estimated support sets obtained by SPDAS, Lasso, MCP, and SCAD with varying correlation values. The results show that SPDAS tends to estimate the true values of T, while Lasso tends to produce larger estimates. The top right panel of Fig. 3 displays

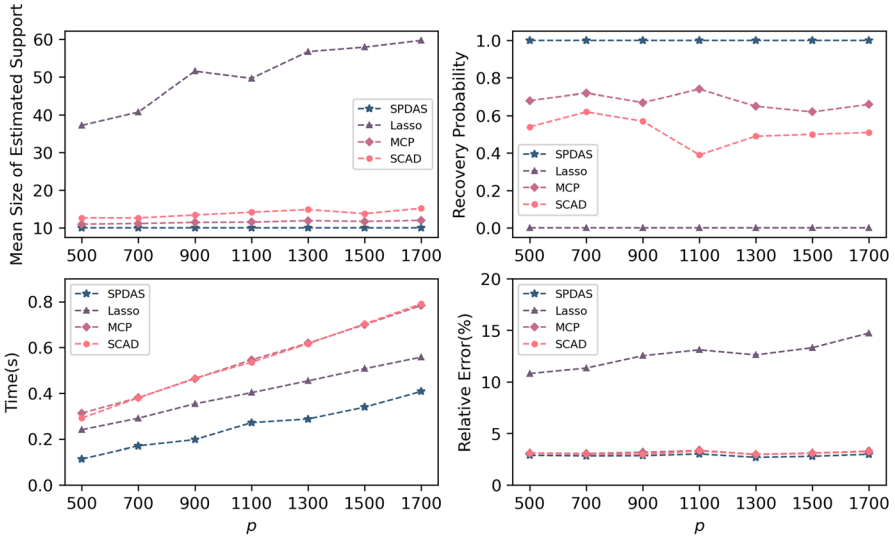

Fig. 2 The numerical results of the influence of variable dimension p ($n = 200$, $p = 500 : 200 : 1700$, $\rho = 0.5, \sigma = 1, R = 5, \text{CR} = 0.3, T = 10$)

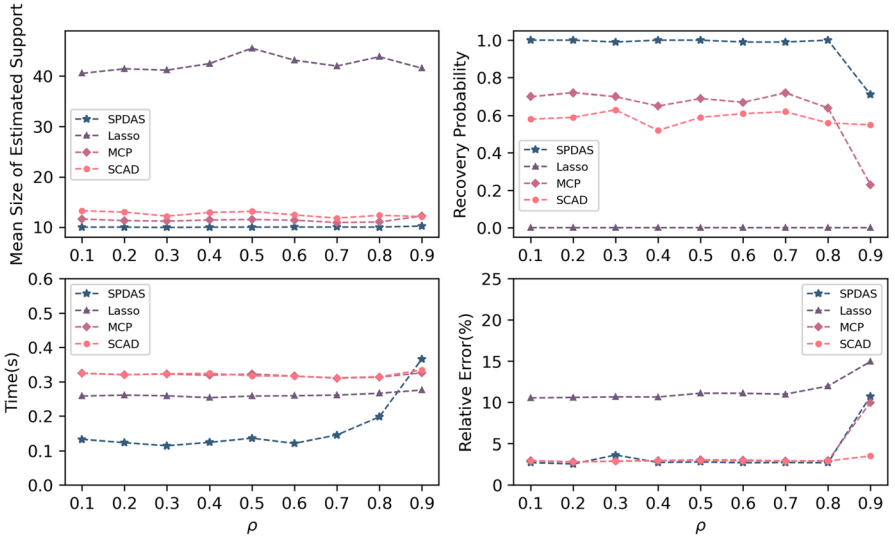

Fig. 3 The numerical results of the influence of correlation ρ ($n = 200$, $p = 600$, $\rho = 0.1 : 0.1 : 0.9$, $\sigma = 1, R = 5, \text{CR} = 0.3, T = 10$)

the impact of correlation ρ on the recovery probability. The recovery probability of SPDAS remains stable and close to 1 when $\rho \leq 0.8$, but declines rapidly otherwise. The bottom left panel of Fig. 3 illustrates the influence of correlation ρ on CPU time. The time consumed by Lasso and MCP gradually increases with ρ, while SPDAS

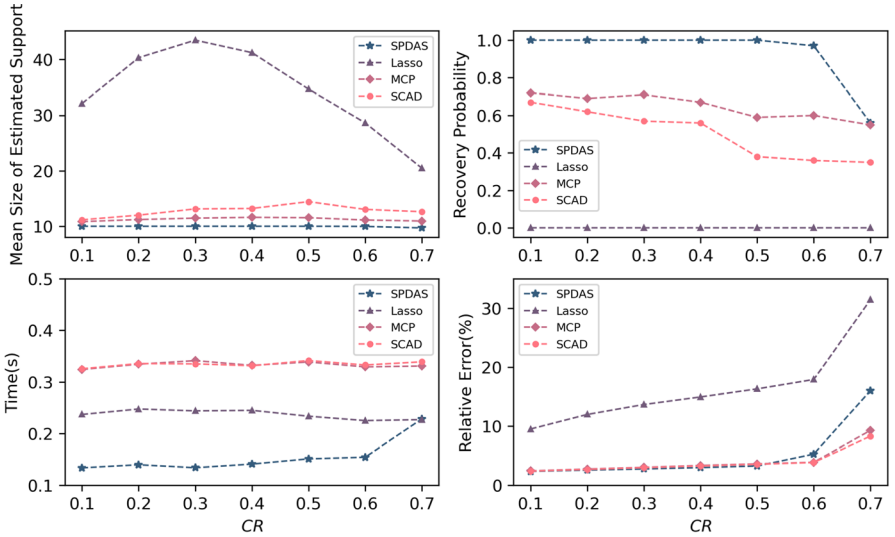

Fig. 4 The numerical results of the influence of censoring rate CR ($n = 200$, $p = 600$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.1 : 0.1 : 0.7$, $T = 10$)

exhibits a swift upward trend when $\rho > 0.7$. Finally, the bottom right panel of Fig. 3 shows that the relative error of SPDAS surges significantly when $\rho > 0.8$.

5.2.4 Influence of the censoring rate CR

The top left panel of Fig. 4 shows the effect of the censoring rate CR on the mean size of estimated support sets. While MCP and SCAD yield a slightly larger estimate than the exact support, SPDAS maintains stability at the true size. The recovery probability of SPDAS gradually decreases from 1 to 0.6 when CR > 0.5. The bottom left panel of Fig. 4 demonstrates that the CPU time of SPDAS gently increases with the growing censoring rate when CR < 0.6 and is larger than Lasso when CR $= 0.7$. The bottom right panel of Fig. 4 indicates that all methods exhibit a poor relative error when the CR $= 0.7$.

5.2.5 Influence of the size of support set *T*

Simulation results about the effect of support size T, depicted in Fig. 5, encompass a range of size of support T spanning from 5 to 60. The top right panel of Fig. 5 visually demonstrates that the SPDAS consistently achieves near-perfect selection accuracy for the correct support set, irrespective of the sparsity level. In contrast, other algorithms approach a recovery probability close to 1 as the size of support T reaches 60. Furthermore, we can notice that at lower sparsity levels (larger T), the performance differences among the algorithms are not pronounced. However, as the sparsity level increases, these differences gradually become more prominent.

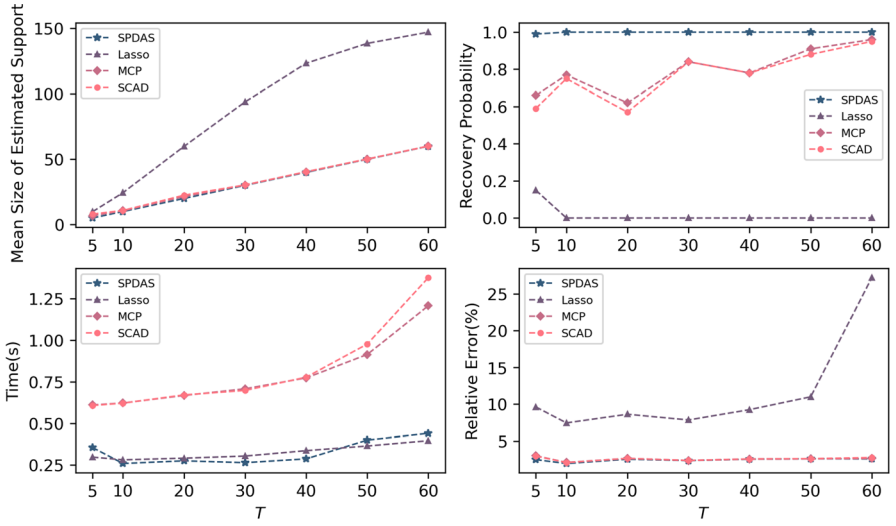

Fig. 5 The numerical results of the influence of the size of support set T ($n = 400$, $p = 600$, $\rho = 0.5$, $\sigma = 1$, $R = 5$, CR $= 0.3$, $T = 5, 10 : 10 : 60$)

This indicates that the SPDAS exhibits greater robustness and accuracy compared to other algorithms when dealing with highly sparse data.

5.3 Real data example

This section entails the application of the proposed methodology to conduct a rigorous analysis of real-world data. Specifically, the ovarian cancer dataset, denoted as *TCGA_eset*, is obtained from the R package *curatedOvarianData* (Ganzfried et al. 2013). The dataset encompasses a total of 578 samples, consisting of 13104 gene expression features, as well as various clinical factors. Our primary focus centers around the gene expression features and the survival time of the patients.

To ensure data integrity, we eliminate 21 samples that contain missing data. Subsequently, we employ the DC-SIS (Li et al. 2012) method to perform a screening of the gene expression features. The primary objective is to condense the 13104 gene expression features into a more manageable subset of 5000 gene expressions.

The resulting censoring rate approximates 47.94%. In our analysis, we employ the AFT model to capture the relationship between the censoring time and covariates, as described by the equation:

$$\ln(T) = \mathbf{X}^T \boldsymbol{\beta} + \epsilon.$$

Here, T, the response variable, represents the time to death and is subject to right censoring. The covariates, denoted as $\mathbf{X}$, are comprised of 5000 expression data points, and ϵ signifies the noise component.

The results presented in Table 3 indicate that the signs of the variable estimates selected by Lasso and SPDAS are the same. The overlap of variable selections

Table 3 Estimation results of *TCGA_eset*

Gene name	Lasso	SPDAS	MCP	SCAD
IL2RG	−0.29	–	–	–
HCP5	−0.13	–	–	–
OR7A10	0.25	1.73	–	–
CCDC22	−0.17	–	–	–
PLA2G2D	−0.41	−1.92	–	–
TBC1D22B	−1.50	−3.40	−0.82	−0.61
ITK	−0.11	−1.71	–	–
RBMX2	−0.11	–	–	–
TIMP4	−0.34	–	–	–
RNF186	–	-	0.32	0.22
MRPS22	–	−1.47	–	–
S100A5	–	−1.40	–	–
UCP3	0.11	–	–	–
HNRNPA1P31	0.18	–	–	–
HNF1A	0.02	–	–	–
CCL7	−0.40	–	–	–
IFNAR1	−0.24	–	–	−0.04
ENPP2	−0.01	–	–	–
ZNF354A	−0.03	–	–	–
S100A5	−0.14	–	–	–
PTPRN	0.06	0.92	–	–
ATP5G3	–	1.27	–	–
RAB3B	–	-	–	0.03

between Lasso and SPDAS is predominantly comprised of those Lasso coefficient estimates that possess larger magnitudes. Both the MCP and SCAD penalties exhibit a propensity for selecting variables of notable significance at a relatively modest rate. Furthermore, it is noteworthy that all methods concur in selecting the TBC1D22B gene, as its estimates exhibit significantly greater magnitudes compared to the estimates of other coefficients. Our focus narrows down to standard variable selection methodologies, adopting a sequential approach to variable selection. However, the regulatory mechanisms of human genes are intricately complex. In addition to its high dimensionality, gene expression data exhibits another distinctive characteristic, namely the formation of gene variable groups. Conventional variable selection methodologies may inadvertently overlook significant group effects and exhibit suboptimal performance (Zeng and Xie 2012). Consequently, it becomes crucial to conscientiously incorporate the intricate nuances inherent in gene data analysis. Therefore, it is of paramount importance for future investigations to expand our algorithmic framework to accommodate more intricate and sophisticated models.

6 Conclusion

We develop the SPDAS algorithm, which is an efficient solution for the L_0 regularized problem in the high-dimensional AFT model. By utilizing the weighted least squares method, we derived the PDAS algorithm by establishing a necessary condition for the global solution of the L_0 penalized optimization target. We have also proved that the PDAS algorithm is a Newton-type algorithm. Furthermore, we embed the PDAS algorithm in each iteration to generate a sequence of solutions, resulting in the SPDAS algorithm. Our simulation studies and real data analysis results demonstrate the favorable performance of SPDAS in variable selection and estimation. Furthermore, SPDAS exhibits notable computational efficiency across various censorship rates and proves to be a viable approach in high-dimensional settings. Therefore, SPDAS could be a valuable addition to the toolkit for addressing the L_0 penalized high-dimensional AFT model.

However, our present approach assumes the independence of failure time and censoring time to employ the Kaplan-Meier estimator to construct the optimization target. Consequently, it is imperative for future research to advance the SPDAS algorithm to address survival time data more properly. One promising avenue for improvement involves exploring the conditional independence of failure and censoring time given covariates, as well as the potential incorporation of an L_0 penalized likelihood criterion.

Appendix A

In this appendix, we prove Lemma 3.1 and Theorem 3.1.

Proof of Lemma 3.1

Proof Let $\tilde{L}_\lambda(\boldsymbol{\xi}) = \frac{1}{2n}\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_0$, and $L_\lambda(\boldsymbol{\beta}) = \frac{1}{2n}\sum_{i=1}^n w_{(i)}\big(Y_{(i)} - \mathbf{X}_{(i)}^T\boldsymbol{\beta}\big)^2 + \lambda\|\boldsymbol{\beta}\|_0$. Suppose $\boldsymbol{\xi}^*$ is a minimizer of $\tilde{L}_\lambda$, then

$$
\begin{aligned}
&\xi_i^* \in \underset{t\in\mathbb{R}}{\operatorname{argmin}}\ \tilde{L}_\lambda(\xi_1^*, \ldots, \xi_{i-1}^*, t, \xi_{i+1}^*, \ldots, \xi_p^*)\\
&\Rightarrow \xi_i^* \in \underset{t\in\mathbb{R}}{\operatorname{argmin}}\ \frac{1}{2n}\left\|\tilde{\mathbf{X}}\boldsymbol{\xi}^* - \tilde{\mathbf{y}} + (t-\xi_i^*)\tilde{\mathbf{X}}_i\right\|_2^2 + \lambda\|t\|_0\\
&\Rightarrow \xi_i^* \in \underset{t\in\mathbb{R}}{\operatorname{argmin}}\ \frac{1}{2}(t-\xi_i^*)^2 + (t-\xi_i^*)\tilde{\mathbf{X}}_i^\top(\tilde{\mathbf{X}}\boldsymbol{\xi}^* - \tilde{\mathbf{y}})/n + \lambda\|t\|_0\\
&\Rightarrow \xi_i^* \in \underset{t\in\mathbb{R}}{\operatorname{argmin}}\ \frac{1}{2}[t - (\xi_i^* + \tilde{\mathbf{X}}_i^\top(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}^*)/n)]^2 + \lambda\|t\|_0.
\end{aligned}
$$

Let $\mathbf{d}^* = \tilde{\mathbf{X}}^\top(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}^*)/n$. By the definition of $\Gamma_\lambda(\cdot)$ in (3.2), we have

$$
\xi_i^* = \Gamma_\lambda(\xi_i^* + d_i^*) \qquad \text{for} \quad i = 1, \ldots, p,
$$

which shows (3.1) holds.

Conversely, if $\boldsymbol{\xi}^*$ and $\mathbf{d}^*$ satisfy (3.1) and (3.2), then we will show that $\boldsymbol{\xi}^*$ is a local minimizer of (2.2), and $\boldsymbol{\beta}^* = \Lambda\boldsymbol{\xi}^*$ is a local minimizer of (2.1) too. We can assume $\mathbf{h}$ is small enough and $\|\mathbf{h}\|_\infty < \sqrt{2\lambda}$. Then we will show $\tilde{L}_\lambda(\boldsymbol{\xi}^* + \mathbf{h}) \geq \tilde{L}_\lambda(\boldsymbol{\xi}^*)$ in two case respectively.

Case1: $\mathbf{h}_{I^*} \neq 0$.

$$\begin{aligned}\left\|\boldsymbol{\xi}^* + \mathbf{h}\right\|_0 =& \left\|\boldsymbol{\xi}^*_{A^*} + \mathbf{h}_{A^*}\right\|_0 + \left\|\mathbf{h}_{I^*}\right\|_0,\\ \lambda\left\|\boldsymbol{\xi}^* + \mathbf{h}\right\|_0 - \lambda\left\|\boldsymbol{\xi}^*\right\|_0 =& \lambda\left\|\boldsymbol{\xi}^*_{A^*} + \mathbf{h}_{A^*}\right\|_0 + \lambda\left\|\mathbf{h}_{I^*}\right\|_0 - \lambda\left\|\boldsymbol{\xi}^*_{A^*}\right\|_0.\end{aligned}$$

Because $|\boldsymbol{\xi}^*_i| \geq \sqrt{2\lambda}$ for $i \in A^*$ and $\|\mathbf{h}\|_\infty < \sqrt{2\lambda}$, we have

$$\begin{aligned}&\lambda\|\boldsymbol{\xi}^*_{A^*} + \mathbf{h}_{A^*}\|_0 - \lambda\|\boldsymbol{\xi}^*_{A^*}\|_0 = 0,\\ &\lambda\|\boldsymbol{\xi}^* + \mathbf{h}\|_0 - \lambda\|\boldsymbol{\xi}^*\|_0 = \lambda\|\mathbf{h}_{I^*}\|_0 > \lambda.\end{aligned}$$

Therefore, we get

$$\begin{aligned}&\tilde{L}_\lambda(\boldsymbol{\xi}^* + \mathbf{h}) - \tilde{L}_\lambda(\boldsymbol{\xi}^*)\\ &= \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}(\boldsymbol{\xi}^* + \mathbf{h})\right\|_2^2 - \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}^*\right\|_2^2 + \lambda\|\mathbf{h}_{I^*}\|_0\\ &= \frac{1}{2n}\left[\left\|\tilde{\mathbf{X}}\mathbf{h}\right\|_2^2 - 2(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}^*)^\top\tilde{\mathbf{X}}\mathbf{h}\right] + \lambda\|\mathbf{h}_{I^*}\|_0\\ &\geq \lambda - \langle\mathbf{d}^*, \mathbf{h}\rangle.\end{aligned}$$

The last inequality $\lambda - \langle\mathbf{d}^*, \mathbf{h}\rangle \geq 0$ holds for any small enough vector $\mathbf{h}$, so we obtain $\tilde{L}_\lambda(\boldsymbol{\xi}^* + \mathbf{h}) - \tilde{L}_\lambda(\boldsymbol{\xi}^*) \geq 0$.

Case2: $\mathbf{h}_{I^*} = 0$.

$$\lambda\|\boldsymbol{\xi}^* + \mathbf{h}\|_0 - \lambda\|\boldsymbol{\xi}^*\|_0 = \lambda\|\boldsymbol{\xi}^*_{A^*} + \mathbf{h}_{A^*}\|_0 - \lambda\|\boldsymbol{\xi}^*_{A^*}\|_0.$$

As $|\boldsymbol{\xi}^*_i| \geq \sqrt{2\lambda}$ for $i \in A^*$ and $\|\mathbf{h}_{A^*}\|_\infty < \sqrt{2\lambda}$, then we have

$$\lambda\|\boldsymbol{\xi}^* + \mathbf{h}\|_0 - \lambda\|\boldsymbol{\xi}^*\|_0 = \lambda\|\boldsymbol{\xi}^*_{A^*} + \mathbf{h}_{A^*}\|_0 - \lambda\|\boldsymbol{\xi}^*_{A^*}\|_0 = 0.$$

Due to $\mathbf{d}^*_{A^*} = \tilde{\mathbf{X}}^\top_{A^*}(\tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{A^*}\boldsymbol{\xi}^*_{A^*})/n = 0$, then we can get

$$\boldsymbol{\xi}^*_{A^*} \in \underset{\boldsymbol{\xi}_{A^*}}{\operatorname{argmin}} \frac{1}{2n}\left\|\tilde{\mathbf{X}}_{A^*}\boldsymbol{\xi}_{A^*} - \tilde{\mathbf{y}}\right\|_2^2.$$

Thus, we conclude that

$$
\begin{aligned}
&\tilde{L}_\lambda(\boldsymbol{\xi}^* + \mathbf{h}) - \tilde{L}_\lambda(\boldsymbol{\xi}^*) \\
&= \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}(\boldsymbol{\xi}^* + \mathbf{h})\right\|_2^2 - \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\xi}^*\right\|_2^2 \\
&= \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{A^*}(\boldsymbol{\xi}^*_{A^*} + \mathbf{h}_{A^*})\right\|_2^2 - \frac{1}{2n}\left\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{A^*}\boldsymbol{\xi}^*_{A^*}\right\|_2^2 \\
&\geq 0.
\end{aligned}
$$

In summary, $\boldsymbol{\xi}^*$ is a local minimizer of $\tilde{L}_\lambda$. Let $\tilde{\mathbf{h}} = \Lambda\mathbf{h}$, then $L_\lambda(\boldsymbol{\beta}^* + \tilde{\mathbf{h}}) - L_\lambda(\boldsymbol{\beta}^*) \geq 0$ holds if the vector $\mathbf{h}$ is sufficiently small, thus $\boldsymbol{\beta}^*$ is also a local minimizer of (2.1).

□

Proof of Theorem 3.1

Proof Denote $\Lambda^k = -\left(\mathcal{H}^k\right)^{-1} F\left(\mathbf{w}^k\right)$. Then,

$$
\mathbf{w}^{k+1} = \mathbf{w}^k - \left(\mathcal{H}^k\right)^{-1} F\left(\mathbf{w}^k\right)
$$

can be reformulated as

$$
\mathcal{H}^k \Lambda^k = -F\left(\mathbf{w}^k\right), \tag{A.1}
$$

$$
\mathbf{w}^{k+1} = \mathbf{w}^k + \Lambda^k. \tag{A.2}
$$

Partition $\mathbf{w}^k$, Λ^k and $F(\mathbf{w}^k)$ according to A^k and I^k such that

$$
\mathbf{w}^k = \begin{pmatrix} \boldsymbol{\xi}^k_{A^k} \\ \boldsymbol{\xi}^k_{I^k} \\ \mathbf{d}^k_{A^k} \\ \mathbf{d}^k_{I^k} \end{pmatrix}, \quad \Lambda^k = \begin{pmatrix} \Lambda^{\boldsymbol{\xi}}_{A^k} \\ \Lambda^{\boldsymbol{\xi}}_{I^k} \\ \Lambda^{\mathbf{d}}_{A^k} \\ \Lambda^{\mathbf{d}}_{I^k} \end{pmatrix}, \tag{A.3}
$$

$$
F\left(\mathbf{w}^k\right) = \begin{bmatrix} -\mathbf{d}^k_{A^k} \\ \boldsymbol{\xi}^k_{I^k} \\ \tilde{\mathbf{X}}^\top_{A^k}\tilde{\mathbf{X}}_{A^k}\boldsymbol{\xi}^k_{A^k} + \tilde{\mathbf{X}}^\top_{A^k}\tilde{\mathbf{X}}_{I^k}\boldsymbol{\xi}^k_{I^k} + n\mathbf{d}^k_{A^k} - \tilde{\mathbf{X}}^\top_{A^k}\mathbf{y} \\ \tilde{\mathbf{X}}^\top_{I^k}\tilde{\mathbf{X}}_{\mathcal{A}^k}\boldsymbol{\xi}^k_{\mathcal{A}^k} + \tilde{\mathbf{X}}^\top_{I^k}\tilde{\mathbf{X}}_{I^k}\boldsymbol{\xi}^k_{I^k} + n\mathbf{d}^k_{I^k} - \tilde{\mathbf{X}}^\top_{I^k}\tilde{\mathbf{y}} \end{bmatrix}. \tag{A.4}
$$

Substituting (A.3), (A.4) and $\mathcal{H}^k$ into (A.1), we have

$$
\mathbf{d}^k_{A_k} + \Lambda^{\mathbf{d}}_{A_k} = \mathbf{0}_{A^k}, \tag{A.5}
$$

$$
\boldsymbol{\xi}^k_{I_k} + \Lambda^{\boldsymbol{\xi}}_{I_k} = \mathbf{0}_{I^k}, \tag{A.6}
$$

$$\tilde{\mathbf{X}}_{A_k}^T \tilde{\mathbf{X}}_{A_k}(\boldsymbol{\xi}_{A_k}^k + \Lambda_{A_k}^{\boldsymbol{\xi}}) = \tilde{\mathbf{X}}_{A_k}^\top \tilde{\mathbf{y}} - n(\mathbf{d}_{A_k}^k + \Lambda_{A_k}^{\mathbf{d}}) - \tilde{\mathbf{X}}_{A_k}^\top \tilde{\mathbf{X}}_{I_k}(\boldsymbol{\xi}_{I_k}^k + \Lambda_{I_k}^{\boldsymbol{\xi}}), \tag{A.7}$$

$$n(\mathbf{d}_{I_k}^k + \Lambda_{I_k}^{\mathbf{d}}) = \tilde{\mathbf{X}}_{I_k}^\top \tilde{\mathbf{y}} - \tilde{\mathbf{X}}_{I_k}^\top \tilde{\mathbf{X}}_{A_k}(\boldsymbol{\xi}_{A_k}^k + \Lambda_{A_k}^{\boldsymbol{\xi}}) - \tilde{\mathbf{X}}_{I_k}^\top \tilde{\mathbf{X}}_{I_k}(\boldsymbol{\xi}_{I_k}^k + \Lambda_{I_k}^{\boldsymbol{\xi}}). \tag{A.8}$$

It follows from (A.2) that

$$\begin{pmatrix} \mathbf{d}_{A_k}^{k+1} \\ \boldsymbol{\xi}_{I_k}^{k+1} \\ \boldsymbol{\xi}_{A_k}^{k+1} \\ \mathbf{d}_{I_k}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_{A_k}^k + \Lambda_{A_k}^{\mathbf{d}} \\ \boldsymbol{\xi}_{I_k}^k + \Lambda_{I_k}^{\boldsymbol{\xi}} \\ \boldsymbol{\xi}_{A_k}^k + \Lambda_{A_k}^{\boldsymbol{\xi}} \\ \mathbf{d}_{I_k}^k + \Lambda_{I_k}^{\mathbf{d}} \end{pmatrix}. \tag{A.9}$$

Substituting (A.9) into (A.5)-(A.8), we obtain (3.3) in PDAS Algorithm 1. This completes the proof. □

Acknowledgements We extend our sincere thanks to the editor, the associate editor, and the anonymous reviewers for their insightful feedback and constructive criticism, which have significantly contributed to the enhancement of this work. This work is supported by the National Natural Science Foundation of China (Grant No. 12371274 and No. 11971362).

References

Breheny P, Huang J (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann Appl Stat 5(1):232–253

Candes E, Romberg J, Tao T (2006) Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans Inf Theory 52(2):489–509

Candes EJ, Tao T (2006) Near-optimal signal recovery from random projections: universal encoding strategies? IEEE Trans Inf Theory 52(12):5406–5425

Cheng C, Feng X, Huang J, Jiao Y, Zhang S (2022) ℓ0-regularized high-dimensional accelerated failure time model. Comput Stat Data Anal 170:107430

Cox DR (1972) Regression models and life-tables. J Roy Stat Soc: Ser B (Methodol) 34(2):187–202

Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc 96(456):1348–1360

Fan J, Lv J (2008) Sure independence screening for ultrahigh dimensional feature space. J R Stat Soc Ser B Stat Methodol 70(5):849–911

Fleming TR, Harrington DP (2011) Counting processes and survival analysis. Wiley, Hoboken

Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–13

Ganzfried BF, Riester M, Haibe-Kains B, Risch T, Tyekucheva S, Jazic I, Wang XV, Ahmadifar M, Birrer MJ, Parmigiani G, Huttenhower C, Waldron L (2013) curatedOvarianData: clinically annotated data for the ovarian cancer transcriptome. Database 2013:bat013

Hu J, Chai H (2013) Adjusted regularized estimation in the accelerated failure time model with high dimensional covariates. J Multivar Anal 122:96–114

Huang J, Ma S, Xie H (2006) Regularized estimation in the accelerated failure time model with high-dimensional covariates. Biometrics 62(3):813–820

Huang J, Jiao Y, Liu Y, Lu X (2018) A constructive approach to l0 penalized regression. J Mach Learn Res 19:403–439

Huang J, Jiao Y, Lu X, Shi Y, Yang Q (2018b) SNAP: a semismooth newton algorithm for pathwise optimization with optimal local convergence rate and oracle properties. arXiv:1810.03814

Huang J, Jiao Y, Jin B, Liu J, Lu X, Yang C (2020) A unified primal dual active set algorithm for nonconvex sparse recovery. Stat Sci 36(2):215–238
Johnson BA (2008) Variable selection in semiparametric linear regression with censored data. J R Stat Soc Ser B (Stat Methodol) 70(2):351–370
Johnson BA, Lin DY, Zeng D (2008) Penalized estimating functions and variable selection in semiparametric regression models. J Am Stat Assoc 103(482):672–680
Kaplan EL, Meier P (1958) Nonparametric estimation from incomplete observations. J Am Stat Assoc 53(282):457–481
Kim Y, Kwon S, Choi H (2012) Consistent model selection criteria on high dimensions. J Mach Learn Res 13:1037–1057
Klein JP, Moeschberger ML (2003) Survival analysis: techniques for censored and truncated data, vol 1230. Springer, Berlin
Koul H, Susarla V, Ryzin JV (1981) Regression analysis with randomly right-censored data. Ann Stat 9(6):1276–1288
Li R, Zhong W, Zhu L (2012) Feature screening via distance correlation learning. J Am Stat Assoc 107(499):1129–1139
Simon N, Friedman J, Hastie T, Tibshirani R (2011) Regularization paths for cox's proportional hazards model via coordinate descent. J Stat Softw 39(5):1–13
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J Roy Stat Soc: Ser B (Methodol) 58(1):267–288
Wang L, Kim Y, Li R (2013) Calibrating nonconvex penalized regression in ultra-high dimension. Ann Stat 41(5):2505–2536
Wei LJ (1992) The accelerated failure time model: a useful alternative to the cox regression model in survival analysis. Stat Med 11(14–15):1871–1879
Zeng L, Xie J (2012) Group variable selection via SCAD-L_2. Statistics 48(1):49–66
Zhang CH (2010) Nearly unbiased variable selection under minimax concave penalty. Ann Stat 38(2):894–942
Zhang CH, Huang J (2008) The sparsity and bias of the lasso selection in high-dimensional linear regression. Ann Stat 36(4):1567–1594
Zhang CH, Zhang T (2012) A general theory of concave regularization for high-dimensional sparse estimation problems. Stat Sci 27(4):576–593