Computational Statistics (2022) 37:507–533
https://doi.org/10.1007/s00180-021-01098-z

ORIGINAL PAPER

GSDAR: a fast Newton algorithm for ℓ_0 regularized generalized linear models with statistical guarantee

Jian Huang[1] · Yuling Jiao[2] · Lican Kang[2] · Jin Liu[3] · Yanyan Liu[2] · Xiliang Lu[2]

Received: 8 July 2020 / Accepted: 13 March 2021 / Published online: 29 March 2021

Abstract

We propose a fast Newton algorithm for ℓ_0 regularized high-dimensional generalized linear models based on support detection and root finding. We refer to the proposed method as GSDAR. GSDAR is developed based on the KKT conditions for ℓ_0-penalized maximum likelihood estimators and generates a sequence of solutions of the KKT system iteratively. We show that GSDAR can be equivalently formulated as a generalized Newton algorithm. Under a restricted invertibility condition on the likelihood function and a sparsity condition on the regression coefficient, we establish an explicit upper bound on the estimation errors of the solution sequence generated by GSDAR in supremum norm and show that it achieves the optimal order in finite iterations with high probability. Moreover, we show that the oracle estimator can be recovered with high probability if the target signal is above the detectable level. These results directly concern the solution sequence generated from the GSDAR algorithm, instead of a theoretically defined global solution. We conduct simulations and real data analysis to illustrate the effectiveness of the proposed method.

Keywords High-dimensional generalized linear models · Sparse learning · ℓ_0-penalty · Support detection · Estimation error

✉ Jian Huang
jian-huang@uiowa.edu

✉ Yanyan Liu
liuyy@whu.edu.cn

1 Department of Statistics and Actuarial Science, University of Iowa, Iowa City, Iowa 52242, USA

2 School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

3 Center of Quantitative Medicine Duke-NUS Medical School, Singapore, Singapore

1 Introduction

Generalized linear models (GLMs) are an important class of statistical models that have wide applications in practice (Nelder and Wedderburn 1972; McCullagh 2019). In GLMs, the conditional distribution of the response variable $Y \in \mathbb{R}$, given the value of the vector of the covariates $\mathbf{x} \in \mathbb{R}^p$, follows an exponential family distribution with the density function

$$f(y; \theta) = \exp[y\theta - c(\theta) + d(y)],$$

where $c(\cdot)$ and $d(\cdot)$ are known functions, $\theta = \mathbf{x}^T \boldsymbol{\beta}^*$, $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is the vector of underlying regression coefficients. Suppose we have a random sample $\{(\mathbf{x}_i, y_i) : i = 1, ..., n\}$ that are i.i.d copies of $(\mathbf{x}, Y)$. Let $E(y_i|\mathbf{x}_i) = \mu_i$, where μ_i is related to the linear function of the predictors $\theta_i = \mathbf{x}_i^T \boldsymbol{\beta}$ through a monotone and differentiable link function $g = (\dot{c})^{-1}$ such that

$$g(\mu_i) = \theta_i.$$

The GLMs include several important special models, including linear regression, logistic regression and Poisson regression.

When the number of predictors p exceeds the sample size n, it is often reasonable to assume that the model is sparse in the sense that the number of predictors that are truly related to the response is much smaller than n. Many researchers have proposed penalized methods for variable selection and estimation in high-dimensional GLMs. Park and Hastie (2007) and Van de Geer et al. (2008) extended the Lasso method (Tibshirani 1996) from linear regression to GLMs. Meier et al. (2008) proposed the group lasso for logistic regression. Friedman et al. (2010) developed coordinate descent to solve the elastic net (Zou and Hastie 2005) penalized GLMs. Path following proximal gradient descent (Nesterov 2013) was adopted in Wang et al. (2014) and Loh and Wainwright (2015) to solve the SCAD (Fan and Li 2001) and MCP (Zhang 2010) regularized GLMs. Li et al. (2017) proposes a DC proximal Newton (DCPN) method to solve GLMs with sparsity promoting nonconvex penalties such as SCAD and MCP. Recently, several authors considered Newton type algorithm for solving sparse GLMs (Wang et al. 2019; Yuan et al. 2017; Shen and Li 2017).

In addition, there is a large body of work on variable selection using ℓ_0 penalties. Many researchers have developed methods that are modifications of the original Bayes information criterion (BIC) (Schwarz et al. 1978), including mBIC for controlling FWER (Bogdan et al. 2004, 2008) and other modifications of BIC for controlling false discovery rate (Frommlet et al. 2012; Żak-Szatkowska and Bogdan 2011). Variable selection methods based on these criteria have been applied to high-dimensional problems such as genomewide association studies (GWAS) using heuristic search methods (Dolejsi et al. 2014; Frommlet et al. 2012). Another interesting algorithmic approach for selection with ℓ_0 penalties is discussed in Frommlet and Nuel (2016). Furthermore, the extended BIC (EBIC) (Chen and Chen 2008, 2012) is also an important method for model selection with ℓ_0 penalties, and its relevant theoretical properties have been studied (Abramovich et al. 2006; Birgé and Massart 2001). Finally, in the

context of genetic association studies, Frommlet et al. (2016) proposed the method for high-dimensional model selection with ℓ_0 penalties.

In this paper, we consider the problem of variable selection and estimation in high-dimensional GLMs based on the ℓ_0-penalized minimization problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathcal{L}(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_0, \tag{1}$$

where $\mathcal{L}(\boldsymbol{\beta}) = -\frac{1}{n}\sum_{i=1}^n \left[y_i \mathbf{x}_i^T\boldsymbol{\beta} - c(\mathbf{x}_i^T\boldsymbol{\beta})\right]$ is the negative log-likelihood function, $\|\boldsymbol{\beta}\|_0$ is the number of nonzero elements of $\boldsymbol{\beta}$, and $\lambda \geq 0$ is a tuning parameter.

It is well known that the ℓ_0-penalized minimization problem (1) is NP-hard (Chen et al. 2014; Natarajan 1995). Therefore, it is infeasible or extremely difficult to compute the exact solution to this minimization problem in high-dimensional settings. We propose a computational approach to approximate the solution to (1) based on a nontrivial extension of the support detection and rooting finding (SDAR) algorithm Huang et al. (2018), developed in the context of ℓ_0 penalized linear regression models. GSDAR is a computational algorithm motivated from the KKT conditions. It generates a sequence of solutions $\{\boldsymbol{\beta}^k\}_k$ iteratively, based on support detection using primal and dual information. We show that GSDAR can be equivalently formulated as a generalized Newton algorithm for finding the root of the KKT systems. Under a restricted invertibility condition on the likelihood function and a sparsity condition on the regression coefficient $\boldsymbol{\beta}^*$, we derive an explicit upper bound on the estimation errors of the solution sequence in supremum norm and show that it achieves optimal order in finite iterations. Moreover, we show that the oracle estimator can be recovered with high probability if the target signal is over the detectable level. These results directly concern the solution sequence generated from the GSDAR algorithm, instead of a theoretically defined global. Therefore, there is no disconnection between our theoretical results and computation algorithm.

The rest of this paper is organized as follows. In Sect. 2 we derive the GSDAR algorithm based on an appropriate formulation of the KKT conditions. We also show that GSDAR can be equivalently formulated as a semismooth Newton algorithm. In Sect. 3 we present an upper bound on the estimation error of the solution sequence generated from GSDAR. In Sect. 4, we extend GSDAR algorithm to AGSDAR, an adaptive version of GSDAR. In Sect. 5 we evaluate the performance of GSDAR and AGSDAR on simulated and real data and compare it with several state-of-the-art methods. We conclude in Sect. 6. Proofs of the theorems are given in the Appendix.

2 Derivation of GSDAR

First, we introduce some notation used throughout the paper. We write $n \gtrsim \log(p)$ to mean that $n \geq c\log(p)$ for some universal constant $c \in (0, \infty)$, where p diverges as n goes to infinity. Let $\|\boldsymbol{\beta}\|_q = (\sum_{i=1}^p |\beta_i|^q)^{\frac{1}{q}}$, $q \in [1, \infty]$, denote the q-norm of a vector $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^T \in \mathbb{R}^p$. Denote the support of $\boldsymbol{\beta}$ by $\text{supp}(\boldsymbol{\beta}) = \{i : \beta_i \neq 0,\ i = 1, ..., p\}$ and $A^* = \text{supp}(\boldsymbol{\beta}^*)$. Let $|A|$ be the size of the set A. Let $\boldsymbol{\beta}_A = (\beta_i, i \in A) \in \mathbb{R}^{|A|}$ and let $\boldsymbol{\beta}|_A \in \mathbb{R}^p$ with its i-th element $(\boldsymbol{\beta}|_A)_i = \beta_i 1(i \in A)$,

where $\mathbf{1}(\cdot)$ is the indicator function. Denote $\mathbf{X}_A = (\mathbf{x}_j, j \in A) \in \mathbb{R}^{n\times|A|}$, where $\mathbf{x}_j$ is j-th column of the covariate matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$. Let $\|\boldsymbol{\beta}\|_{T,\infty}$ and $\|\boldsymbol{\beta}\|_{\min}$ be the T-th largest element (in absolute value) and the minimum absolute value of $\boldsymbol{\beta}$, respectively. Let $\nabla\mathcal{L}$ and $\nabla^2\mathcal{L}$ be the gradient and Hessian of function $\mathcal{L}$, respectively.

The following lemma gives the KKT conditions for (1).

Lemma 1 *If $\widehat{\boldsymbol{\beta}}$ is a minimizer of* (1)*, then $\widehat{\boldsymbol{\beta}}$ satisfies:*

$$\begin{cases} \widehat{\mathbf{d}} = -\nabla\mathcal{L}(\widehat{\boldsymbol{\beta}}), \\ \widehat{\boldsymbol{\beta}} = H_\lambda(\widehat{\boldsymbol{\beta}} + \widehat{\mathbf{d}}), \end{cases} \tag{2}$$

where $H_\lambda(\cdot)$ is the hard thresholding operator whose i-th element is defined by

$$(H_\lambda(\boldsymbol{\beta}))_i = \begin{cases} 0, & |\beta_i| < \sqrt{2\lambda}, \\ \beta_i, & |\beta_i| \geq \sqrt{2\lambda}. \end{cases}$$

Conversely, if $\widehat{\boldsymbol{\beta}}$ and $\widehat{\mathbf{d}}$ satisfy (2)*, then $\widehat{\boldsymbol{\beta}}$ is a local minimizer of* (1)*.*

The proof of Lemma 1 is given in Appendix A.1.

Let $\widehat{A} = \text{supp}(\widehat{\boldsymbol{\beta}})$ and $\widehat{I} = (\widehat{A})^c$. By the definition of $H_\lambda(\cdot)$ and (2), we have

$$\widehat{A} = \{i : |\widehat{\beta}_i + \widehat{d}_i| \geq \sqrt{2\lambda}\}, \quad \widehat{I} = \{i : |\widehat{\beta}_i + \widehat{d}_i| < \sqrt{2\lambda}\},$$

and

$$\begin{cases} \widehat{\boldsymbol{\beta}}_{\widehat{I}} = \mathbf{0} \\ \widehat{\mathbf{d}}_{\widehat{A}} = \mathbf{0} \\ \widehat{\boldsymbol{\beta}}_{\widehat{A}} \in \underset{\boldsymbol{\beta}_{\widehat{A}}}{\text{argmin}}\ \widetilde{\mathcal{L}}(\boldsymbol{\beta}_{\widehat{A}}) \\ \widehat{\mathbf{d}}_{\widehat{I}} = [-\nabla\mathcal{L}(\widehat{\boldsymbol{\beta}})]_{\widehat{I}}, \end{cases}$$

where

$$\widetilde{\mathcal{L}}(\boldsymbol{\beta}_{\widehat{A}}) = \mathcal{L}(\boldsymbol{\beta}|_{\widehat{A}}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i\mathbf{x}_{i(\widehat{A})}^T\boldsymbol{\beta}_{\widehat{A}} - c\big(\mathbf{x}_{i(\widehat{A})}^T\boldsymbol{\beta}_{\widehat{A}}\big)\right].$$

Let $\{\boldsymbol{\beta}^k, \mathbf{d}^k\}$ be the output of k-th iteration in GSDAR algorithm. If $\{\boldsymbol{\beta}^k, \mathbf{d}^k\}$ approximates $\{\widehat{\boldsymbol{\beta}}, \widehat{\mathbf{d}}\}$ well, then $\{A^k, I^k\}$ also approximates $\{\widehat{A}, \widehat{I}\}$ well, where $\{A^k, I^k\}$ is defined as

$$A^k = \{i : |\beta_i^k + d_i^k| \geq \sqrt{2\lambda}\}, \quad I^k = \{i : |\beta_i^k + d_i^k| < \sqrt{2\lambda}\}. \tag{3}$$

We obtain a new approximation pair $\{\boldsymbol{\beta}^{k+1}, \mathbf{d}^{k+1}\}$ as follows:

$$\begin{cases} \boldsymbol{\beta}_{I^k}^{k+1} = \mathbf{0} \\ \mathbf{d}_{A^k}^{k+1} = \mathbf{0} \\ \boldsymbol{\beta}_{A^k}^{k+1} \in \underset{\boldsymbol{\beta}_{A^k}}{\operatorname{argmin}} \widetilde{\mathcal{L}}(\boldsymbol{\beta}_{A^k}) \\ \mathbf{d}_{I^k}^{k+1} = [-\nabla \mathcal{L}(\boldsymbol{\beta}^{k+1})]_{I^k}, \end{cases} \tag{4}$$

where

$$\widetilde{\mathcal{L}}(\boldsymbol{\beta}_{A^k}) = \mathcal{L}(\boldsymbol{\beta}|_{A^k}) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \mathbf{x}_{i(A^k)}^T \boldsymbol{\beta}_{A^k} - c\left(\mathbf{x}_{i(A^k)}^T \boldsymbol{\beta}_{A^k}\right)\right].$$

If the minimizer $\boldsymbol{\beta}_{A^k}^{k+1}$ of (4) is not unique, we choose the one with the smallest value in ℓ_∞-norm. If we have the prior information that $\|\boldsymbol{\beta}^*\|_0 \leq T$, then we set

$$\sqrt{2\lambda} = \| \boldsymbol{\beta}^k + \mathbf{d}^k \|_{T,\infty} \tag{5}$$

in (3). With this choice of λ, we have $|A^k| = T$ in every iteration. Let $\boldsymbol{\beta}^0$ be an initial value, then we obtain a sequence of solutions $\{\boldsymbol{\beta}^k, k \geq 1\}$ by using (3) and (4) with the λ in (5).

We give a detailed description of the GSDAR algorithm in Algorithm 1.

Algorithm 1 GSDAR

1: Input: $\boldsymbol{\beta}^0$, T, $\mathbf{d}^0 = -\nabla \mathcal{L}(\boldsymbol{\beta}^0)$; $k = 0$
2: **for** $k = 0, 1, \ldots,$ **do**
3: $A^k = \{j : |\beta_j^k + d_j^k| \geq \|\boldsymbol{\beta}^k + \mathbf{d}^k\|_{T,\infty}\}$, $I^k = (A^k)^c$.
4: $\boldsymbol{\beta}_{I^k}^{k+1} = \mathbf{0}$.
5: $\mathbf{d}_{A^k}^{k+1} = \mathbf{0}$.
6: $\boldsymbol{\beta}_{A^k}^{k+1} = \underset{\boldsymbol{\beta}_{A^k}}{\operatorname{argmin}} \widetilde{\mathcal{L}}(\boldsymbol{\beta}_{A^k})$.
7: $\mathbf{d}_{I^k}^{k+1} = [-\nabla \mathcal{L}(\boldsymbol{\beta}^{k+1})]_{I^k}$.
8: **if** $A^k = A^{k+1}$, **then**
9: Stop and denote the last iteration $\boldsymbol{\beta}_{\widehat{A}}$, $\boldsymbol{\beta}_{\widehat{I}}$, $\mathbf{d}_{\widehat{A}}$, $\mathbf{d}_{\widehat{I}}$.
10: **else**
11: $k = k + 1$
12: **end if**
13: **end for**
14: Output: $\widehat{\boldsymbol{\beta}} = (\boldsymbol{\beta}_{\widehat{A}}^{\mathrm{T}}, \boldsymbol{\beta}_{\widehat{I}}^{\mathrm{T}})^{\mathrm{T}}$ as the estimates of $\boldsymbol{\beta}^*$.

In Algorithm 1, we usually set the initial value $\boldsymbol{\beta}^0 = \mathbf{0}$. We terminate GSDAR when $A^k = A^{k+1}$ for some k, because the sequences generated by GSDAR will not change. In Sect. 3, we will prove that under some regularity conditions on $\mathbf{X}$ and $\boldsymbol{\beta}^*$,

with high probability $A^* = A^k = A^{k+1}$ in finite steps, i.e., the GSDAR will stop and the oracle estimator will be recovered.

2.1 GSDAR as a generalized Newton algorithm

The proposed GSDAR is derived in an intuitive way from the suitably formulated KKT conditions for the ℓ_0 penalized log-likelihood. We show that the GSDAR Algorithm 1 can be interpreted as a Newton-type method for finding roots of the KKT system (2) even though the original problem (1) is nonconvex and nonsmooth. Let $\mathbf{w} = (\boldsymbol{\beta}; \mathbf{d}) \in \mathbb{R}^p \times \mathbb{R}^p$ and

$$F(\mathbf{w}) = \begin{pmatrix} F_1(\mathbf{w}) \\ F_2(\mathbf{w}) \end{pmatrix} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^{2p},$$

where $F_1(\mathbf{w}) = \boldsymbol{\beta} - H_\lambda(\boldsymbol{\beta} + \mathbf{d})$ and $F_2(\mathbf{w}) = n\mathbf{d} + n\nabla\mathcal{L}(\boldsymbol{\beta})$.

Proposition 1 *The iteration in (4) can be equivalently reformulated as*

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \left(\mathcal{H}^k\right)^{-1} F\left(\mathbf{w}^k\right), \tag{6}$$

where

$$\mathcal{H}^k = \begin{pmatrix} \mathcal{H}_1^k & \mathcal{H}_2^k \\ n\nabla^2\mathcal{L}(\boldsymbol{\beta}^k) & n\mathbf{I} \end{pmatrix}$$

with

$$\mathcal{H}_1^k = \begin{pmatrix} \mathbf{0}_{A^kA^k} & \mathbf{0}_{A^kI^k} \\ \mathbf{0}_{I^kA^k} & \mathbf{I}_{I^kI^k} \end{pmatrix} \quad \textit{and} \quad \mathcal{H}_2^k = \begin{pmatrix} -\mathbf{I}_{A^kA^k} & \mathbf{0}_{A^kI^k} \\ \mathbf{0}_{I^kA^k} & \mathbf{0}_{I^kI^k} \end{pmatrix}.$$

The proof of this proposition is given in Appendix A.2. We remark that, although the iteration (6) has exactly the same format of a Newton type algorithm, it does not imply the superlinear convergence property from the semismooth Newton method theory (Qi and Sun (1993); Qi (1993); Chen et al. (2000)). This is because the hard thresholding operator in (2) is not Newton differentiable. A recent work Wang et al. (2019) proved GSDAR with an approximate step size achieves fast local convergence to stationary points for ℓ_0 constraint high-dimensional logistic regression model. In the following section, we establish an error bound of $\boldsymbol{\beta}^k$ as an estimator of the underlying target $\boldsymbol{\beta}^*$.

3 Theoretical properties

In this section, we establish the ℓ_∞ error bound for the GSDAR estimator. Under appropriate conditions, we show that $\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty$ achieves sharp estimation error rate. Furthermore, if the minimum value of target signal is detectable, GSDAR will

recover the oracle estimator in finite steps if T is greater than the true model size K. We assume the following conditions.

(C1) There exist constants $0 < L < U < \infty$ such that, for all $\boldsymbol{\beta}_1 \neq \boldsymbol{\beta}_2$ with $\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_0 \leq 2T$,

$$0 < L \leq \frac{(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2)^T \cdot \nabla^2 \mathcal{L}(\widetilde{\boldsymbol{\beta}}) \cdot (\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2)}{\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_1 \|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_\infty} \leq U < \infty,$$

where $\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}_1 + \nu(\boldsymbol{\beta}_2 - \boldsymbol{\beta}_1)$ for any $\nu \in (0, 1)$.

(C2) $\|\boldsymbol{\beta}^*_{A^*}\|_{\min} \geq \frac{3c_1}{L}\sqrt{\frac{\log(p)}{n}}$, where c_1 is a universal numerical constant.

Remark 1 Condition (C1) extends the weak cone invertibility condition in Ye and Zhang (2010). This kind of restricted strong convexity conditions is needed in bounding the estimation error in high-dimensional models Zhang et al. (2012). Condition (C2) is needed to guarantee the target signal to be detectable.

3.1 ℓ_∞ Error bounds

Theorem 1 *Assume (C1) holds with* $0 < U < \frac{1}{T}$. *Set* $K \leq T$ *and* $\boldsymbol{\beta}^0 = \boldsymbol{0}$ *in Algorithm 1.*

(i) Before Algorithm 1 terminates, we have

$$\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k \|\boldsymbol{\beta}^*\|_\infty + \frac{2}{L}\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty,$$

where $\xi = 1 - \frac{2L(1-TU)}{T(1+K)} \in (0, 1)$.

(ii) Assume the rows of $\mathbf{X}$ *are i.i.d. sub-Gaussian with* $n \gtrsim \log(p)$, *then there exist universal constants* $\{c_1, c_2, c_3\}$ *with* $0 < c_i < \infty$, $i = 1, 2, 3$, *such that with probability at least* $1 - c_2 \exp(-c_3 \log(p))$,

$$\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k \|\boldsymbol{\beta}^*\|_\infty + \frac{2c_1}{L}\sqrt{\frac{\log(p)}{n}}.$$

It follows that

$$\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \mathcal{O}\left(\sqrt{\frac{\log(p)}{n}}\right)$$

with high probability if $k \geq \mathcal{O}\left(\log_{\frac{1}{\xi}} \frac{n}{\log(p)}\right)$.

The proof of this theorem is given in Appendix A.4.

Remark 2 The requirement $U < \frac{1}{T}$ is not essential since we can always rescale the loss function $\mathcal{L}$ to make it hold. This rescaling is equivalent to multiplying a step size to the dual variable in the the GSDAR algorithm. Let τ be this step size satisfying $0 < \tau < \frac{1}{TU}$. Then, Theorem 1 still holds by replacing ξ with $1 - \frac{2\tau L(1-\tau TU)}{T(1+K)} \in (0, 1)$.

3.2 Support recovery

The following theorem establishes the support recovery property of GSDAR.

Theorem 2 *Assume (C1) and (C2) hold with* $0 < U < \frac{1}{T}$, *and the rows of* $\mathbf{X}$ *are i.i.d. sub-Gaussian with* $n \gtrsim \log(p)$. *Set* $K \leq T$ *in Algorithm 1. Then with probability at least* $1 - c_2 \exp(-c_3 \log(p))$, $A^* \subseteq A^k$ *if* $k > \log_{\frac{1}{\xi}} 9(T + K)(1 + \frac{U}{L})r^2$, *where* $r = \frac{\|\boldsymbol{\beta}^*\|_\infty}{\|\boldsymbol{\beta}^*_{A^*}\|_{\min}}$ *is the range of* $\boldsymbol{\beta}^*$.

The proof of this theorem is given in Appendix A.5.

Remark 3 Theorem 2 shows that the estimated support via GSDAR can recover the true support with the cost at most $\mathcal{O}(\log(T))$ number of iteration if the minimum signal strength of $\boldsymbol{\beta}^*$ is above the detectable threshold $\mathcal{O}(\sqrt{\frac{\log(p)}{n}})$. Support recovery for sparse GLMs has also been studied in Li et al. (2017); Yuan et al. (2017) and Shen and Li (2017). In Li et al. (2017), the authors propose a DC proximal Newton (DCPN) method to solve GLMs with nonconvex sparse promoting penalties such as MCP/SCAD. They derive an estimation error in ℓ_2 norm with order $\mathcal{O}(\sqrt{\frac{K \log p}{n}})$ under an assumption similar to (C1). They show that the true support can be reconverted under the requirement $\|\boldsymbol{\beta}^*_{A^*}\|_{\min} \geq \mathcal{O}(\sqrt{\frac{K \log(p)}{n}})$, which is stronger than our assumption (C2). The computational complexity of DCPN is worse than GSDAR since the DCPN is based on the multistage convex relaxation scheme to transform the original nonconvex optimizations into a sequence of LASSO regularized GLMs, therefore, a Lasso inner solver is called at each stage Ge et al. (2019). In Yuan et al. (2017) and Shen and Li (2017), Gradient hard thresholding pursuit is shown to recover the true support under the requirement $\|\boldsymbol{\beta}^*_{A^*}\|_{\min} \geq \mathcal{O}(\sqrt{\frac{K \log(p)}{n}})$, which is also stronger than our assumption (C2). If we set $T = K$ in GSDAR, then the stopping criterion $A^k = A^{k+1}$ holds if $k \geq \mathcal{O}(\log(K))$ since the estimated support coincides with the true support. As a consequence, the oracle estimator will be recovered in $\mathcal{O}(\log(K))$ steps. However, Yuan et al. (2017) or Shen and Li (2017) did not prove that the stopping condition of gradient hard thresholding pursuit can be satisfied. Meanwhile, the iteration complexity of Gradient hard thresholding pursuit analyzed by Shen and Li (2017) is $\mathcal{O}(K)$, which is worse than the complexity bound established here.

4 Adaptive GSDAR

In practice, the sparsity level of the true parameter value $\boldsymbol{\beta}^*$ is unknown. So we regard T as a tuning parameter. Let T increase from 0 to Q, a given positive integer. We

compute a set of solutions: $\{\widehat{\boldsymbol{\beta}}(T) : T = 0, 1, ..., Q\}$, where $\widehat{\boldsymbol{\beta}}(0) = \mathbf{0}$. We take $Q = \alpha n/\log(n)$ as suggested by Fan and Lv (2008), where α is a positive and finite constant. In our numerical studies, we set $\alpha = 1$. We can use a data-driven method such as HBIC (Wang et al. 2013), mBIC (Bogdan et al. 2004, 2008) or mBIC2 (Żak-Szatkowska and Bogdan 2011) to determine $\widehat{T}$, the choice of T. Then we take $\widehat{\boldsymbol{\beta}}(\widehat{T})$ as the final estimator of $\boldsymbol{\beta}^*$.

We summarize the adaptive GSDAR in Algorithm 2.

Algorithm 2 AGSDAR

1: Input: $\boldsymbol{\beta}^0$, $\mathbf{d}^0$=$-\nabla\mathcal{L}(\boldsymbol{\beta}^0)$, an integer ϑ, an integer Q.
2: **for** $k = 1, \ldots,$ **do**
3: Run Algorithm 1 with $T = \vartheta k$ and with initial value $\boldsymbol{\beta}^{k-1}$, $\mathbf{d}^{k-1}$. Denote the output by $\boldsymbol{\beta}^k$, $\mathbf{d}^k$.
4: **if** $T > Q$, **then**
5: stop
6: **else**
7: $k = k + 1$
8: **end if**
9: **end for**
10: Output: $\widehat{\boldsymbol{\beta}}(\widehat{T})$ as the estimates of $\boldsymbol{\beta}^*$.

5 Simulation studies and real data analysis

In this section, we conduct simulation studies to evaluate the performance of the proposed method in the context of logistic regression with a binary response and use real data to illustrate its applications. First, we compare AGSDAR with Lasso, MCP and the stepwise selection method in terms of accuracy, efficiency and classification accuracy rate. Then, we further compare AGSDAR with these methods on the effects of model parameters, including sample size n, model dimension p and correlation level among the predictors. Third, we evaluate the computational efficiency of GSDAR by examining the average number of iterations needed for GSDAR to converge. Finally, we illustrate the application of GSDAR/AGSDAR on several real datasets.

Our implementation of Lasso and MCP is according to the `R` package `ncvreg` developed by Breheny and Huang (2011). The stepwise selection method is implemented in the `R` package `bigstep` (Bogdan et al. 2004, 2008). In the implementation of AGSDAR, we set $Q = n/\log(n)$, and use HBIC to choose the value of T. The `R` code of GSDAR is available on GitHub at https://github.com/jian94/GSDAR.

5.1 Accuracy, efficiency and classification accuracy rate

We generate the design matrix $\mathbf{X}$ as follows. First, we generate a $n \times p$ random Gaussian matrix $\overline{\mathbf{X}}$, whose entries are i.i.d. $\sim N(0, 1)$, and normalize its columns to the $\sqrt{n}$ length. Then the design matrix $\mathbf{X}$ is generated with $\mathbf{x}_1 = \overline{\mathbf{x}}_1$, $\mathbf{x}_p = \overline{\mathbf{x}}_p$, and $\mathbf{x}_j = \overline{\mathbf{x}}_j + \rho(\overline{\mathbf{x}}_{j+1} + \overline{\mathbf{x}}_{j-1})$, $j = 2, ..., p-1$. The underlying regression coefficient $\boldsymbol{\beta}^*$ with K nonzero coefficients is generated such that the K nonzero coefficients in $\boldsymbol{\beta}^*$ are uniformly distributed in (m_1, m_2), where $m_1 = 5\sqrt{2\log p/n}$ and $m_2 =$

Table 1 Numerical results (the averaged relative error, CPU time, the average classification accuracy rate by prediction) on data set with $n = 300$, $p = 5000$, $K = 10$, $\rho = 0.2{:}0.2{:}0.8$

ρ	Method	AREE	Time(s)	ACRP
0.2	Lasso	0.99	6.03	86.68%
	MCP	0.95	11.93	93.95%
	Stepwise	4.34	120.39	93.81%
	AGSDAR	0.95	1.42	91.15%
0.4	Lasso	0.99	6.11	86.62%
	MCP	0.95	11.07	94.37%
	Stepwise	4.62	111.55	94.03%
	AGSDAR	0.97	1.33	88.73%
0.6	Lasso	0.99	6.33	86.55%
	MCP	0.96	11.47	93.85%
	Stepwise	1.66	111.87	93.46%
	AGSDAR	0.98	1.41	89.80%
0.8	Lasso	1.00	6.28	86.43%
	MCP	0.97	11.47	93.38%
	Stepwise	1.87	109.10	93.40%
	AGSDAR	0.98	1.44	89.75%

$100 \cdot m_1$. The K nonzero coefficients are randomly assigned to the K components of $\boldsymbol{\beta}^*$. The response variable is generated according to $y_i \sim \text{Binomial}(1, p_i)$, where $p_i = \exp(\mathbf{x}_i^T \boldsymbol{\beta}^*)/[1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta}^*)]$, $i = 1, \ldots, n$.

We randomly choose 80% of the samples as the training set and the remaining 20% as the test set in calculating the classification accuracy rate. We take $n = 300$, $p = 5000$, $K = 10$ and $\rho = 0.2{:}0.2{:}0.8$.

Table 1 presents the simulation results, including the average of relative estimation error (AREE) of $\widehat{\boldsymbol{\beta}}$ defined as $\text{AREE} = \frac{1}{100} \sum_{j=1}^{100} \|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^*\|/\|\boldsymbol{\beta}^*\|$, CPU time in seconds (Time) and average classification accuracy rate (ACAR) based on 100 independent replications.

We see that AGSDAR has about the same AREE values as Lasso, MCP, while the stepwise method has the largest AREE values. In terms of the speed, AGSDAR is about 5, 8 and 80 times faster than Lasso, MCP and the stepwise method, respectively. For the average classification accuracy rate, AGSDAR has smaller ACRP values than MCP and the stepwise method but higher ACRP values than Lasso. The simulation results reported below demonstrate that AGSDAR tends to perform better than the other methods in terms of model selection.

5.2 Influence of the model parameters

We now consider the effects of each of the model parameters on the performance of AGSDAR, Lasso, MCP and the stepwise method. We generate each row the $n \times p$ design matrix X from $N(0, \Sigma)$, where $\Sigma_{ij} = \rho^{|i-j|}$, $1 \le i, j \le p$. The underlying regression coefficient vector $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is generated in such a way that the K

nonzero coefficients in $\boldsymbol{\beta}^*$ are uniformly distributed in $(1, R)$, and the support A^* is a randomly chosen subset of $\{1, ..., p\}$ with $|A^*| = K < n$. Then the response $y_i \sim \text{Binomial}(1, p_i)$, where $p_i = \exp(\mathbf{x}_i^T \boldsymbol{\beta}^*)/[1 + \exp(\mathbf{x}_i^T \boldsymbol{\beta}^*)]$, $i = 1, ..., n$.

We compare the performance of the methods considered in terms of average positive discovery rate (APDR), average false discovery rate (AFDR) and average combined discovery rate (ADR) Luo and Chen (2014) defined as follows.

$$\begin{aligned} \text{APDR} &= \frac{1}{100} \sum \frac{|\widehat{A} \bigcap A^*|}{|A^*|}, \\ \text{AFDR} &= \frac{1}{100} \sum \frac{|\widehat{A} \bigcap A^{*c}|}{|\widehat{A}|}, \\ \text{ADR} &= \text{APDR} + (1 - \text{AFDR}), \end{aligned}$$

where $\widehat{A}$ denotes the estimated support set. The simulation results are based on 100 independent replications.

5.2.1 Influence of the sample size *n*

Table 2 shows the influence of the sample size n on APDR, AFDR and ADR. We set $p = 500$, $K = 6$, $R = 10$, $\rho = 0.3$ and let n vary from 100 to 400 by step 50.

We see that as the sample size n increases, Lasso always has the highest values of APDR. However, Lasso also has the largest values of AFDR for each n, which is only a little smaller than APDR when $n > 100$. This indicates that Lasso tends to over select variables that are not in the support of the regression coefficient. AGSDAR always has the smallest values of AFDR and highest values of ADR when $n > 100$, and its APDR values are also not small. Therefore, AGSDAR avoids selecting the erroneous variable while selecting as many relevant variables as possible into the model, especially when the sample size n increases. MCP and the stepwise method are similar to AGSDAR in terms of variable selection. However, MCP and the stepwise method tend to select more irrelevant variables than AGSDAR. Overall, AGSDAR always selects more relevant variables and fewer irrelevant variables.

5.2.2 Influence of the variable dimension *p*

Table 3 shows the influence of the model dimension p on the APDR, AFDR and ADR. We set $n = 100$, $K = 6$, $R = 10$, $\rho = 0.2$, and take $p = 100$ to 700 with a step size 100.

Table 3 shows that Lasso has the largest values on APDR and AFDR, and lowest values on ADR. Meanwhile, the AFDR values of Lasso are greater than 0.5 and higher than those of APDR when $p > 400$ which suggests that Lasso selects more irrelevant variables than relevant variables. AGSDAR, MCP and the stepwise method have almost the same APDR values, especially when $p < 500$, indicating that AGSDAR, MCP and the stepwise method have similar ability to select relevant variables. Besides, AGSDAR has the better AFDR and ADR values than Lasso and MCP, and is comparable with the stepwise method with respect to AFDR and ADR. Hence,

Table 2 Numerical results (APDR, AFDR, ADR) on the data $p = 500$, $K = 6$, $R = 10$, $\rho = 0.3$ and $n = 100{:}50{:}400$

n	method	APDR	AFDR	ADR
100	Lasso	0.83	0.84	0.99
	MCP	0.79	0.36	1.43
	Stepwise	0.75	0.15	1.60
	AGSDAR	0.72	0.19	1.53
150	Lasso	0.92	0.87	1.05
	MCP	0.90	0.22	1.68
	Stepwise	0.86	0.17	1.69
	AGSDAR	0.85	0.15	1.70
200	Lasso	0.95	0.88	1.07
	MCP	0.93	0.19	1.74
	Stepwise	0.91	0.19	1.72
	AGSDAR	0.90	0.12	1.78
250	Lasso	0.97	0.89	1.08
	MCP	0.93	0.16	1.77
	Stepwise	0.95	0.18	1.77
	AGSDAR	0.93	0.06	1.87
300	Lasso	0.98	0.89	1.09
	MCP	0.95	0.15	1.80
	Stepwise	0.95	0.16	1.79
	AGSDAR	0.96	0.06	1.90
350	Lasso	0.99	0.89	1.10
	MCP	0.95	0.16	1.79
	Stepwise	0.97	0.18	1.79
	AGSDAR	0.96	0.05	1.91
400	Lasso	0.99	0.89	1.10
	MCP	0.97	0.15	1.82
	Stepwise	0.98	0.14	1.84
	AGSDAR	0.98	0.05	1.93

AGSDAR tends to select fewer irrelevant variables and thus reduce the complexity of the model.

5.2.3 Influence of the correlation ρ

Table 4 presents the influence of the correlation ρ on APDR, AFDR and ADR. We set $n = 150$, $p = 500$, $K = 6$, $R = 10$ and $\rho = 0.1$ to 0.9 with an increasing step size 0.1.

We see from Table 4 that Lasso has the best APDR values and worst AFDR and ADR values for every ρ. AGSDAR, MCP and the stepwise method have nearly the same APDR values for each ρ. AGSDAR always has the best AFDR and ADR values when $\rho < 0.7$, and it is still comparable to the stepwise method in terms of AFDR

Table 3 Numerical results (APDR, AFDR, ADR) on the data $n = 100$, $K = 6$, $R = 10$, $\rho = 0.2$ and $p = 100{:}100{:}700$

p	Method	APDR	AFDR	ADR
100	Lasso	0.92	0.77	1.15
	MCP	0.83	0.20	1.63
	Stepwise	0.83	0.15	1.68
	AGSDAR	0.82	0.16	1.66
200	Lasso	0.88	0.81	1.07
	MCP	0.83	0.23	1.60
	Stepwise	0.80	0.16	1.64
	AGSDAR	0.80	0.17	1.63
300	Lasso	0.89	0.82	1.07
	MCP	0.82	0.29	1.53
	Stepwise	0.77	0.16	1.61
	AGSDAR	0.80	0.21	1.59
400	Lasso	0.84	0.84	1.00
	MCP	0.79	0.34	1.45
	Stepwise	0.74	0.18	1.56
	AGSDAR	0.75	0.20	1.55
500	Lasso	0.83	0.85	0.98
	MCP	0.78	0.35	1.43
	Stepwise	0.70	0.17	1.53
	AGSDAR	0.74	0.20	1.54
600	Lasso	0.79	0.85	0.94
	MCP	0.77	0.39	1.38
	Stepwise	0.70	0.19	1.51
	AGSDAR	0.70	0.22	1.48
700	Lasso	0.80	0.85	0.95
	MCP	0.77	0.37	1.40
	Stepwise	0.68	0.20	1.48
	AGSDAR	0.70	0.25	1.45

and ADR when $\rho \geq 0.7$. Therefore AGSDAR can simultaneously select the relevant variables and avoid the irrelevant variables for a wide spectrum of the values of ρ.

5.3 Number of iterations

To further evaluate the numerical convergence of GSDAR, we conduct simulations to examine the number of iterations for GSDAR to converge with $T = K$ in Algorithm 1. We generate data in the same way as described in Sect. 5.2. We look at the influence of the correlation level ρ. We record the average number of iterations for different values of ρ. Figure 1 shows the average number of iterations of GSDAR based on 100 independent replications with $n = 500$, $p = 1000$, $K = 2{:}2{:}50$, $R = 3$ and $\rho = 0.1{:}0.2{:}0.7$.

Table 4 Numerical results (APDR, AFDR, ADR) on the data $n = 150$, $p = 500$, $K = 6$, $R = 10$ and $\rho = 0.1{:}0.1{:}0.9$

ρ	Method	APDR	AFDR	ADR
0.1	Lasso	0.92	0.87	1.05
	MCP	0.87	0.22	1.65
	Stepwise	0.86	0.18	1.68
	AGSDAR	0.85	0.15	1.70
0.2	Lasso	0.92	0.87	1.05
	MCP	0.89	0.21	1.68
	Stepwise	0.85	0.18	1.67
	AGSDAR	0.85	0.15	1.70
0.3	Lasso	0.92	0.87	1.05
	MCP	0.90	0.23	1.67
	Stepwise	0.87	0.17	1.70
	AGSDAR	0.88	0.13	1.75
0.4	Lasso	0.91	0.87	1.04
	MCP	0.87	0.23	1.64
	Stepwise	0.84	0.20	1.64
	AGSDAR	0.84	0.15	1.69
0.5	Lasso	0.90	0.86	1.04
	MCP	0.85	0.26	1.59
	Stepwise	0.85	0.18	1.67
	AGSDAR	0.83	0.16	1.67
0.6	Lasso	0.90	0.87	1.03
	MCP	0.88	0.22	1.66
	Stepwise	0.85	0.19	1.66
	AGSDAR	0.84	0.16	1.68
0.7	Lasso	0.90	0.86	1.04
	MCP	0.83	0.26	1.57
	Stepwise	0.81	0.18	1.63
	AGSDAR	0.80	0.22	1.58
0.8	Lasso	0.88	0.86	1.02
	MCP	0.75	0.31	1.44
	Stepwise	0.79	0.22	1.57
	AGSDAR	0.75	0.26	1.49
0.9	Lasso	0.82	0.84	0.98
	MCP	0.55	0.48	1.07
	Stepwise	0.59	0.39	1.20
	AGSDAR	0.58	0.44	1.14

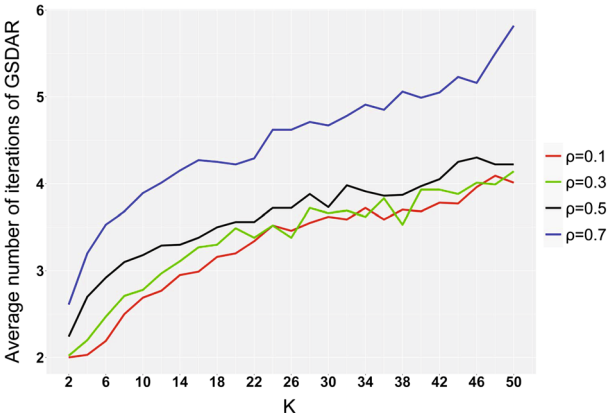

Fig. 1 The average number of iterations of GSDAR as K increases

As shown in Fig. 1, the average number of iterations of GSDAR increases as the sparsity level increases from 2 to 50 for every ρ. But even when the sparsity level K is 50, the average number of iterations is only 4 when $\rho = 0.1, 0.3,$ and$0.5,$ and is about 5.5 when $\rho = 0.7$. This indicates that GSDAR converges fast.

5.4 Real data examples

We illustrate the application of GSDAR to five data sets: duke breast-cancer, gisette, leukemia, madelon and splice, which are described in Table 5. These datasets are available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. The duke breast-cancer and leukemia data sets have been standardized such that the mean of each predictor is 0 and variance is 1. The response variable takes the value $y = 1$ if the subject has the disease and $y = 0$ otherwise. We fit the logistic regression model to these data sets and compare the classification accuracy rate of the proposed methods with Lasso, MCP and the stepwise method. Set $T = n/\log(n)$ in GSDAR. We implement the AGSDAR, Lasso, MCP and the stepwise method as described in Sect. 5. The results are given in Table 6, which show that the classification accuracy rates of GSDAR and AGSDAR are comparable with those of Lasso, MCP and the stepwise method. Moreover, we denote the number of selected variables by $\widehat{T}$, which is showed in Table 7.

In summary, our simulation results and application to data examples demonstrated that the proposed GSDAR algorithm performs better than or comparably with other penalized methods such as Lasso and MCP and the stepwise selection method.

6 Conclusion

GSDAR is a generalized Newton algorithm for fitting sparse, high-dimensional GLMs. It iteratively solves the KKT system for the ℓ_0-penalized likelihood for the GLMs.

Table 5 Description of four real data sets

Data name	n Samples	p Features	Training size n_1	Testing set n_2
Duke breast-cancer	42	7129	38	4
Gisette	7000	5000	6000	1000
Leukemia	72	7129	38	34
Madelon	2600	500	2000	600
Splice	3175	60	1000	2175

Table 6 Classification accuracy rate

Data name	GSDAR	AGSDAR	Lasso	MCP	Stepwise
Duke breast-cancer	1	1	1	25%	1
Gisette	54.10%	56.30%	51.30%	59.90%	49.40%
Leukemia	91.18%	94.12%	91.17%	94.11%	44.12%
Madelon	57.83%	59.83%	61.50%	61.50%	49.00%
Splice	84.23%	85.05%	85.70%	84.91%	51.63%

Table 7 The number of selected variables($\widehat{T}$)

Data name	GSDAR	AGSDAR	Lasso	MCP	Stepwise
Duke breast-cancer	10	5	23	5	2
Gisette	344	60	507	49	48
Leukemia	5	14	13	4	1
Madelon	6	3	4	2	2
Splice	30	25	40	26	14

We establish an optimal ℓ_∞ error bound for the sequence generated by GSDAR algorithm under appropriate regularity and sparsity conditions. Furthermore, we show that the oracle estimator can be recovered with high probability if the target signal is detectable. We also propose the AGSDAR algorithm, an adaptive version of GSDAR, to handle the problem of unknown sparsity level. Numerical results on simulated and real data demonstrate that GSDAR/AGSDAR algorithm is fast, stable and accurate. Therefore, the proposed GSDAR algorithm is a useful addition to the literature on variable selection in high-dimensional GLMs.

For the future research, it would be interesting to generalize GSDAR to solve structured sparsity learning problems (Breheny and Huang 2015; Jiao et al. 2017) with general convex losses or to problems related to deep neural networks (Scardapane et al. 2017; Louizos et al. 2018; Ma et al. 2019).

Acknowledgements We wish to thank two anonymous reviewers for their constructive and helpful comments that led to significant improvements in the paper. The work of J. Huang is supported in part by the U.S. National Science Foundation grant DMS-1916199. The work of Y. Jiao is supported in part by the National Science Foundation of China grant 11871474 and by the research fund of KLATASDSMOE of China. The

research of J. Liu is supported by Duke-NUS Graduate Medical School WBS: R-913-200-098-263 and MOE2016-T2-2-029 from Ministry of Eduction, Singapore. The work of Y. Liu is supported in part by the National Science Foundation of China grant 11971362. The work X. Lu is supported by National Science Foundation of China Grants 11471253 and 91630313.

A Appendix

In the appendix, we prove Lemma 1, Proposition 1 and Theorems 1 and 2.

A.1 Proof of Lemma 1

Proof Let $L_\lambda(\boldsymbol{\beta}) = \mathcal{L}(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_0$. Assume $\widehat{\boldsymbol{\beta}}$ is a global minimizer of $L_\lambda(\boldsymbol{\beta})$. Then by Theorem 10.1 in Rockafellar and Wets (2009), we have

$$\mathbf{0} \in \nabla\mathcal{L}(\widehat{\boldsymbol{\beta}}) + \lambda\partial\|\widehat{\boldsymbol{\beta}}\|_0, \tag{7}$$

where $\partial\|\widehat{\boldsymbol{\beta}}\|_0$ denotes the limiting subdifferential (see Definition 8.3 in Rockafellar and Wets (2009)) of $\|\cdot\|_0$ at $\widehat{\boldsymbol{\beta}}$. Let $\widehat{\mathbf{d}} = -\nabla\mathcal{L}(\widehat{\boldsymbol{\beta}})$ and define $G(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta} - (\widehat{\boldsymbol{\beta}} + \widehat{\mathbf{d}})\|^2 + \lambda\|\boldsymbol{\beta}\|_0$. We recall that, from the definition of the limiting subdifferential of Definition 8.3 in Rockafellar and Wets (2009)), $\partial\|\widehat{\boldsymbol{\beta}}\|_0$ satisfies that $\|\boldsymbol{\beta}\|_0 \geq \|\widehat{\boldsymbol{\beta}}\|_0 + \langle\partial\|\widehat{\boldsymbol{\beta}}\|_0, \boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\rangle + o(\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|)$ for any $\boldsymbol{\beta} \in \mathbb{R}^p$. (7) is equivalent to

$$\mathbf{0} \in \widehat{\boldsymbol{\beta}} - (\widehat{\boldsymbol{\beta}} + \widehat{\mathbf{d}}) + \lambda\partial\|\widehat{\boldsymbol{\beta}}\|_0.$$

Moreover, $\widetilde{\boldsymbol{\beta}}$ being the minimizer of $G(\boldsymbol{\beta})$ is equivalent to $\mathbf{0} \in \partial G(\widetilde{\boldsymbol{\beta}})$. Obviously, $\widehat{\boldsymbol{\beta}}$ satisfies $\mathbf{0} \in \partial G(\widehat{\boldsymbol{\beta}})$. Thus we deduce that $\widehat{\boldsymbol{\beta}}$ is a KKT point of $G(\boldsymbol{\beta})$. Then $\widehat{\boldsymbol{\beta}} = H_\lambda(\widehat{\boldsymbol{\beta}} + \widehat{\mathbf{d}})$ follows from the result that the KKT points of G coincide with its coordinate-wise minimizer (Huang et al. 2021). Conversely, suppose $\widehat{\boldsymbol{\beta}}$ and $\widehat{\mathbf{d}}$ satisfy (2), then $\widehat{\boldsymbol{\beta}}$ is a local minimizer of $L_\lambda(\boldsymbol{\beta})$. To show $\widehat{\boldsymbol{\beta}}$ is a local minimizer of $L_\lambda(\boldsymbol{\beta})$, we can assume $\mathbf{h}$ is small enough and $\|\mathbf{h}\|_\infty < \sqrt{2\lambda}$. Then we will show $L_\lambda(\widehat{\boldsymbol{\beta}} + \mathbf{h}) \geq L_\lambda(\widehat{\boldsymbol{\beta}})$ in two cases respectively.

First, we denote

$$\widehat{A} = \{i : |\widehat{\beta}_i + \widehat{d}_i| \geq \sqrt{2\lambda}\}, \quad \widehat{I} = \{i : |\widehat{\beta}_i + \widehat{d}_i| < \sqrt{2\lambda}\}.$$

By the definition of $H_\lambda(\cdot)$ and (2), we can conclude that $|\widehat{\beta}_i| \geq \sqrt{2\lambda}$ when $i \in \widehat{A}$ and $\widehat{\boldsymbol{\beta}}_{\widehat{I}} = 0$. Thus it yields that $\mathrm{supp}(\widehat{\boldsymbol{\beta}}) = \widehat{A}$. Moreover, we also have $\widehat{\mathbf{d}}_{\widehat{A}} = [-\nabla\mathcal{L}(\widehat{\boldsymbol{\beta}})]_{\widehat{A}} = 0$, which is equivalent to $\widehat{\boldsymbol{\beta}}_{\widehat{A}} \in \underset{\boldsymbol{\beta}_{\widehat{A}}}{\mathrm{argmin}}\, \widetilde{\mathcal{L}}(\boldsymbol{\beta}_{\widehat{A}})$.

Case 1: $\mathbf{h}_{\widehat{I}} \neq 0$.

$$\|\widehat{\boldsymbol{\beta}} + \mathbf{h}\|_0 = \|\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}}\|_0 + \|\mathbf{h}_{\widehat{I}}\|_0,$$
$$\lambda\|\widehat{\boldsymbol{\beta}} + \mathbf{h}\|_0 - \lambda\|\widehat{\boldsymbol{\beta}}\|_0 = \lambda\|\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}}\|_0 + \lambda\|\mathbf{h}_{\widehat{I}}\|_0 - \lambda\|\widehat{\boldsymbol{\beta}}_{\widehat{A}}\|_0.$$

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

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

Therefore, we get

$$\begin{aligned}
&L_\lambda(\widehat{\boldsymbol{\beta}} + \mathbf{h}) - L_\lambda(\widehat{\boldsymbol{\beta}})\\
&= \sum_{i=1}^{n}[c(\mathbf{x}_i^T(\widehat{\boldsymbol{\beta}} + \mathbf{h})) - c(\mathbf{x}_i^T\widehat{\boldsymbol{\beta}})] - \mathbf{y}^T\mathbf{X}\mathbf{h} + \lambda\|\mathbf{h}_{\widehat{I}}\|_0\\
&> \sum_{i=1}^{n}[c(\mathbf{x}_i^T(\widehat{\boldsymbol{\beta}} + \mathbf{h})) - c(\mathbf{x}_i^T\widehat{\boldsymbol{\beta}})] - \mathbf{y}^T\mathbf{X}\mathbf{h} + \lambda\\
&> 0.
\end{aligned}$$

Let $m(\mathbf{h}) = \sum_{i=1}^{n}[c(\mathbf{x}_i^T(\widehat{\boldsymbol{\beta}}+\mathbf{h})) - c(\mathbf{x}_i^T\widehat{\boldsymbol{\beta}})] - \mathbf{y}^T\mathbf{X}\mathbf{h}$, so $m(\mathbf{h})$ is a continuous function about $\mathbf{h}$. As $\mathbf{h}$ is small enough and $\|\mathbf{h}\|_\infty < \sqrt{2\lambda}$, then $m(\mathbf{h}) + \lambda > 0$. Thus the last inequality holds.

Case 2: $\mathbf{h}_{\widehat{I}} = 0$.

$$\lambda\|\widehat{\boldsymbol{\beta}} + \mathbf{h}\|_0 - \lambda\|\widehat{\boldsymbol{\beta}}\|_0 = \lambda\|\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}}\|_0 - \lambda\|\widehat{\boldsymbol{\beta}}_{\widehat{A}}\|_0.$$

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

$$\lambda\|\widehat{\boldsymbol{\beta}} + \mathbf{h}\|_0 - \lambda\|\widehat{\boldsymbol{\beta}}\|_0 = \lambda\|\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}}\|_0 - \lambda\|\widehat{\boldsymbol{\beta}}_{\widehat{A}}\|_0 = 0,$$

and

$$\begin{aligned}
&L_\lambda(\widehat{\boldsymbol{\beta}} + \mathbf{h}) - L_\lambda(\widehat{\boldsymbol{\beta}})\\
&= \sum_{i=1}^{n}[c(\mathbf{x}_i^T(\widehat{\boldsymbol{\beta}} + \mathbf{h})) - c(\mathbf{x}_i^T\widehat{\boldsymbol{\beta}})] - \mathbf{y}^T\mathbf{X}\mathbf{h}\\
&= \sum_{i=1}^{n}[c(\mathbf{x}_{i(\widehat{A})}^T(\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}})) - c(\mathbf{x}_{i(\widehat{A})}^T\widehat{\boldsymbol{\beta}}_{\widehat{A}})] - \mathbf{y}^T\mathbf{X}_{\widehat{A}}\mathbf{h}_{\widehat{A}}\\
&= \sum_{i=1}^{n}[c(\mathbf{x}_{i(\widehat{A})}^T(\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}}))] - \mathbf{y}^T\mathbf{X}_{\widehat{A}}(\widehat{\boldsymbol{\beta}}_{\widehat{A}} + \mathbf{h}_{\widehat{A}})\\
&\quad - \sum_{i=1}^{n}[c(\mathbf{x}_{i(\widehat{A})}^T\widehat{\boldsymbol{\beta}}_{\widehat{A}})] + \mathbf{y}^T\mathbf{X}_{\widehat{A}}\widehat{\boldsymbol{\beta}}_{\widehat{A}}\\
&\geq 0.
\end{aligned}$$

As known that $\widehat{\boldsymbol{\beta}}_{\widehat{A}} \in \operatorname{argmin}_{\boldsymbol{\beta}_{\widehat{A}}} \widetilde{\mathcal{L}}(\boldsymbol{\beta}_{\widehat{A}})$, so the last inequality holds. In summary, $\widehat{\boldsymbol{\beta}}$ is a local minimizer of $L_\lambda(\widehat{\boldsymbol{\beta}})$. □

A.2 Proof of Proposition 1

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

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

can be recast as

$$\mathcal{H}^k D^k = -F\left(\mathbf{w}^k\right), \tag{8}$$

$$\mathbf{w}^{k+1} = \mathbf{w}^k + D^k. \tag{9}$$

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

$$\mathbf{w}^k = \begin{pmatrix} \boldsymbol{\beta}^k_{A^k} \\ \boldsymbol{\beta}^k_{I^k} \\ \mathbf{d}^k_{A^k} \\ \mathbf{d}^k_{I^k} \end{pmatrix}, \quad D^k = \begin{pmatrix} D^{\boldsymbol{\beta}}_{A^k} \\ D^{\boldsymbol{\beta}}_{I^k} \\ D^{\mathbf{d}}_{A^k} \\ D^{\mathbf{d}}_{I^k} \end{pmatrix}, \tag{10}$$

$$F\left(\mathbf{w}^k\right) = \begin{bmatrix} -\mathbf{d}^k_{A^k} \\ \boldsymbol{\beta}^k_{I^k} \\ n[\nabla \mathcal{L}(\boldsymbol{\beta}^k)]_{A^k} + n\mathbf{d}^k_{A^k} \\ n[\nabla \mathcal{L}(\boldsymbol{\beta}^k)]_{I^k} + n\mathbf{d}^k_{I^k} \end{bmatrix}. \tag{11}$$

Substituting (10), (11) and $\mathcal{H}^k$ into (8), we have

$$\left(\mathbf{d}^k_{A^k} + D^{\mathbf{d}}_{A^k}\right) = \mathbf{0}_{A^k}, \tag{12}$$

$$\boldsymbol{\beta}^k_{I^k} + D^{\boldsymbol{\beta}}_{I^k} = \mathbf{0}_{I^k}, \tag{13}$$

$$n\nabla^2 \mathcal{L}(\boldsymbol{\beta}^k) \begin{pmatrix} D^{\boldsymbol{\beta}}_{A^k} \\ D^{\boldsymbol{\beta}}_{I^k} \end{pmatrix} + n \begin{pmatrix} D^{\mathbf{d}}_{A^k} \\ D^{\mathbf{d}}_{I^k} \end{pmatrix} = n[\nabla \mathcal{L}(\boldsymbol{\beta}^k)] + n\mathbf{d}^k. \tag{14}$$

It follows from (9) that

$$\begin{pmatrix} \boldsymbol{\beta}^{k+1}_{A^k} \\ \boldsymbol{\beta}^{k+1}_{I^k} \\ \mathbf{d}^{k+1}_{A^k} \\ \mathbf{d}^{k+1}_{I^k} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}^k_{A^k} + D^{\boldsymbol{\beta}}_{A^k} \\ \boldsymbol{\beta}^k_{I^k} + D^{\boldsymbol{\beta}}_{I^k} \\ \mathbf{d}^k_{A^k} + D^{\mathbf{d}}_{A^k} \\ \mathbf{d}^k_{I^k} + D^{\mathbf{d}}_{I^k} \end{pmatrix}. \tag{15}$$

Substituting (15) into (12)–(14), we get (4) of Algorithm 1. This completes the proof. □

A.3 Preparatory lemmas

The proofs of Theorems 1 and 2 are built on the following lemmas.

Lemma 2 *Assume (C1) holds and* $\|\boldsymbol{\beta}^*\|_0 = K \le T$. *Denote* $B^k = A^k \backslash A^{k-1}$. *Then,*

$$\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_1\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_\infty \ge 2L\zeta[\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*)],$$

where $\zeta = \frac{|B^k|}{|B^k|+|A^*\backslash A^{k-1}|}$.

Proof Obviously, this lemma holds if $A^k = A^{k-1}$ or $\mathcal{L}(\boldsymbol{\beta}^k) \le \mathcal{L}(\boldsymbol{\beta}^*)$. So we only prove the lemma by assuming $A^k \ne A^{k-1}$ and $\mathcal{L}(\boldsymbol{\beta}^k) > \mathcal{L}(\boldsymbol{\beta}^*)$. The condition (C1) indicates

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\beta}^*) - \mathcal{L}(\boldsymbol{\beta}^k) - \langle\nabla\mathcal{L}(\boldsymbol{\beta}^k), \boldsymbol{\beta}^* - \boldsymbol{\beta}^k\rangle\\ &\quad\ge \frac{L}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_1\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_\infty.\end{aligned}$$

Hence,

$$\begin{aligned}&\langle-\nabla\mathcal{L}(\boldsymbol{\beta}^k), \boldsymbol{\beta}^* - \boldsymbol{\beta}^k\rangle\\ &\quad= \langle\nabla\mathcal{L}(\boldsymbol{\beta}^k), -\boldsymbol{\beta}^*\rangle\\ &\quad\ge \frac{L}{2}\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_1\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_\infty + \mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*)\\ &\quad\ge \sqrt{2L}\sqrt{\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_1\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_\infty}\sqrt{\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*)}.\end{aligned}$$

From the definition of A^k and A^*, it is known that B^k contains the first $|B^k|$-largest elements (in absolute value) of $\nabla\mathcal{L}(\boldsymbol{\beta}^k)$, and $\text{supp}(\nabla\mathcal{L}(\boldsymbol{\beta}^k)) \bigcap \text{supp}(\boldsymbol{\beta}^*) = A^*\backslash A^{k-1}$. Thus, we have

$$\begin{aligned}\langle\nabla\mathcal{L}(\boldsymbol{\beta}^k), -\boldsymbol{\beta}^*\rangle &\le \frac{1}{\sqrt{\zeta}}\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_2\|\boldsymbol{\beta}^*_{A^*\backslash A^{k-1}}\|_2\\ &= \frac{1}{\sqrt{\zeta}}\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_2\|(\boldsymbol{\beta}^* - \boldsymbol{\beta}^k)_{A^*\backslash A^{k-1}}\|_2\\ &\le \frac{1}{\sqrt{\zeta}}\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_2\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_2\\ &\le \frac{1}{\sqrt{\zeta}}\sqrt{\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_1\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_\infty}\\ &\quad\times\sqrt{\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_1\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^k\|_\infty}.\end{aligned}$$

Therefore,

$$\begin{aligned}&\sqrt{2L}\sqrt{\mathcal{L}(\boldsymbol{\beta}^k)-\mathcal{L}(\boldsymbol{\beta}^*)}\\&\leq\frac{1}{\sqrt{\xi}}\sqrt{\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_1\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_\infty}.\end{aligned}$$

In summary,

$$\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_1\|\nabla_{B^k}\mathcal{L}(\boldsymbol{\beta}^k)\|_\infty\geq 2L\xi[\mathcal{L}(\boldsymbol{\beta}^k)-\mathcal{L}(\boldsymbol{\beta}^*)].$$

□

Lemma 3 *Assume (C1) holds with* $0<U<\frac{1}{T}$, *and* $K\leq T$ *in Algorithm 1. Then before Algorithm 1 terminates,*

$$\mathcal{L}(\boldsymbol{\beta}^{k+1})-\mathcal{L}(\boldsymbol{\beta}^*)\leq\xi[\mathcal{L}(\boldsymbol{\beta}^k)-\mathcal{L}(\boldsymbol{\beta}^*)],$$

where $\xi=1-\frac{2L(1-TU)}{T(1+K)}\in(0,1)$.

Proof Let $\Delta^k=\boldsymbol{\beta}^k-\nabla\mathcal{L}(\boldsymbol{\beta}^k)$. The condition of (C1) indicates

$$\begin{aligned}\mathcal{L}(\Delta^{k+1}|_{A^{k+1}})-\mathcal{L}(\boldsymbol{\beta}^{k+1})&\leq\langle\nabla\mathcal{L}(\boldsymbol{\beta}^{k+1}),\Delta^{k+1}|_{A^{k+1}}-\boldsymbol{\beta}^{k+1}\rangle\\&+\frac{U}{2}\left\|\Delta^{k+1}|_{A^{k+1}}-\boldsymbol{\beta}^{k+1}\right\|_1\left\|\Delta^{k+1}|_{A^{k+1}}-\boldsymbol{\beta}^{k+1}\right\|_\infty.\end{aligned}$$

On the one hand, by the definition of $\boldsymbol{\beta}^{k+1}$ and $\nabla\mathcal{L}(\boldsymbol{\beta}^{k+1})$, we have

$$\begin{aligned}&\langle\nabla\mathcal{L}(\boldsymbol{\beta}^{k+1}),\Delta^{k+1}|_{A^{k+1}}-\boldsymbol{\beta}^{k+1}\rangle\\&=\langle\nabla\mathcal{L}(\boldsymbol{\beta}^{k+1}),\Delta^{k+1}|_{A^{k+1}}\rangle\\&=\langle\nabla_{A^{k+1}}\mathcal{L}(\boldsymbol{\beta}^{k+1}),\Delta^{k+1}_{A^{k+1}}\rangle\\&=\langle\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1}),\Delta^{k+1}_{A^{k+1}\setminus A^k}\rangle.\end{aligned}$$

Further, we also have

$$\begin{aligned}&\left\|\Delta^{k+1}|_{A^{k+1}}-\boldsymbol{\beta}^{k+1}\right\|_1\\&=\left\|\Delta^{k+1}|_{A^{k+1}\setminus A^k}+\Delta^{k+1}|_{A^{k+1}\bigcap A^k}\right.\\&\quad\left.-\boldsymbol{\beta}^{k+1}|_{A^{k+1}\bigcap A^k}-\boldsymbol{\beta}^{k+1}|_{A^k\setminus A^{k+1}}\right\|_1\\&=\left\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\right\|_1+\left\|\Delta^{k+1}_{A^{k+1}\bigcap A^k}-\boldsymbol{\beta}^{k+1}_{A^{k+1}\bigcap A^k}\right\|_1\\&\quad+\left\|\boldsymbol{\beta}^{k+1}_{A^k\setminus A^{k+1}}\right\|_1\\&=\left\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\right\|_1+\left\|\boldsymbol{\beta}^{k+1}_{A^k\setminus A^{k+1}}\right\|_1,\end{aligned}$$

and

$$\begin{aligned}&\left\|\Delta^{k+1}|_{A^{k+1}}-\boldsymbol{\beta}^{k+1}\right\|_\infty\\&=\left\|\Delta^{k+1}|_{A^{k+1}\setminus A^k}+\Delta^{k+1}|_{A^{k+1}\bigcap A^k}\right.\\&\quad\left.-\boldsymbol{\beta}^{k+1}|_{A^{k+1}\bigcap A^k}-\boldsymbol{\beta}^{k+1}|_{A^k\setminus A^{k+1}}\right\|_\infty\\&=\left\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\right\|_\infty\bigvee\left\|\boldsymbol{\beta}^{k+1}_{A^k\setminus A^{k+1}}\right\|_\infty,\end{aligned}$$

where $a\bigvee b=\max\{a,b\}$. By the definition of A^k, A^{k+1} and $\boldsymbol{\beta}^{k+1}$, we know that

$$|A^k\setminus A^{k+1}|=|A^{k+1}\setminus A^k|,\quad \Delta^{k+1}_{A^k\setminus A^{k+1}}=\boldsymbol{\beta}^{k+1}_{A^k\setminus A^{k+1}}.$$

By the definition of A^{k+1}, we can conclude that

$$\begin{aligned}&\|\Delta^{k+1}_{A^k\setminus A^{k+1}}\|_1=\|\boldsymbol{\beta}^{k+1}_{A^k\setminus A^{k+1}}\|_1\le\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\|_1,\\&\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\|_\infty\bigvee\|\boldsymbol{\beta}^{k+1}_{A^k\setminus A^{k+1}}\|_\infty=\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\|_\infty.\end{aligned}$$

Due to $-\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1})=\Delta^{k+1}_{A^{k+1}\setminus A^k}$ and $U<\frac{1}{T}$, hence we can deduce that

$$\begin{aligned}&\mathcal{L}(\Delta^{k+1}|_{A^{k+1}})-\mathcal{L}(\boldsymbol{\beta}^{k+1})\\&\le\langle\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1}),\Delta^{k+1}_{A^{k+1}\setminus A^k}\rangle+U\left\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\right\|_1\left\|\Delta^{k+1}_{A^{k+1}\setminus A^k}\right\|_\infty\\&\le-(1/T-U)\left\|\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1})\right\|_1\times\left\|\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1})\right\|_\infty.\end{aligned}$$

By the definition of $\boldsymbol{\beta}^{k+1}$, we have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}^{k+1})-\mathcal{L}(\boldsymbol{\beta}^k)&\le\mathcal{L}(\Delta^k|_{A^k})-\mathcal{L}(\boldsymbol{\beta}^k)\\&\le-(1/T-U)\left\|\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1})\right\|_1\times\left\|\nabla_{A^{k+1}\setminus A^k}\mathcal{L}(\boldsymbol{\beta}^{k+1})\right\|_\infty.\end{aligned}$$

Moreover, $\frac{|A^*\setminus A^{k-1}|}{|B^k|}\le K$. By Lemma 2, we have

$$\mathcal{L}(\boldsymbol{\beta}^{k+1})-\mathcal{L}(\boldsymbol{\beta}^k)\le-\frac{2L(1-TU)}{T(1+K)}[\mathcal{L}(\boldsymbol{\beta}^k)-\mathcal{L}(\boldsymbol{\beta}^*)].$$

Therefore, we have

$$\mathcal{L}(\boldsymbol{\beta}^{k+1})-\mathcal{L}(\boldsymbol{\beta}^*)\le\xi[\mathcal{L}(\boldsymbol{\beta}^k)-\mathcal{L}(\boldsymbol{\beta}^*)],$$

where $\xi=1-\frac{2L(1-TU)}{T(1+K)}\in(0,1)$. □

Lemma 4 *Assume $\mathcal{L}$ satisfies (C1) and*

$$\mathcal{L}(\boldsymbol{\beta}^{k+1})-\mathcal{L}(\boldsymbol{\beta}^*)\le\xi[\mathcal{L}(\boldsymbol{\beta}^k)-\mathcal{L}(\boldsymbol{\beta}^*)]$$

for all $k \geq 0$. *Then,*

$$\begin{aligned}\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k \|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_\infty \\ + \frac{2}{L}\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty.\end{aligned} \tag{16}$$

Proof If $\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty < \frac{2\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty}{L}$, then (16) holds, so we only consider the case that $\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \geq \frac{2\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty}{L}$. On the one hand, $\mathcal{L}$ satisfies (C1), then

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*) \\ &\geq \langle \nabla \mathcal{L}(\boldsymbol{\beta}^*), \boldsymbol{\beta}^k - \boldsymbol{\beta}^* \rangle + \frac{L}{2}\left\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\right\|_1 \left\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\right\|_\infty \\ &\geq -\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_1 + \frac{L}{2}\left\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\right\|_1 \left\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\right\|_\infty.\end{aligned}$$

Due to $\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \geq \frac{2\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty}{L}$, then

$$(\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_1 - \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty)\left(\frac{L}{2}\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty - \|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty\right) \geq 0.$$

Further, we can get

$$\frac{L}{2}\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty^2 - \|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty \|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty - [\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*)] \leq 0,$$

which is univariate quadratic inequality about $\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty$. Thus, by simple computation, we can get

$$\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{\frac{2\max\{\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*), 0\}}{L}} + \frac{2\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty}{L}. \tag{17}$$

On the other hand, because $\mathcal{L}$ satisfies (C1), then

$$\begin{aligned}&\mathcal{L}(\boldsymbol{\beta}^0) - \mathcal{L}(\boldsymbol{\beta}^*) \\ &\leq \langle \nabla \mathcal{L}(\boldsymbol{\beta}^*), \boldsymbol{\beta}^0 - \boldsymbol{\beta}^* \rangle + \frac{U}{2}\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_1 \left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty \\ &\leq \left\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\right\|_\infty \left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_1 + \frac{U}{2}\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_1 \left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty \\ &\leq (K+T)\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty (\|\nabla \mathcal{L}(\boldsymbol{\beta}^*)\|_\infty + \frac{U}{2}\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty).\end{aligned}$$

Then, we can get

$$\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*) \leq \xi[\mathcal{L}(\boldsymbol{\beta}^{k-1}) - \mathcal{L}(\boldsymbol{\beta}^*)]$$

$$
\begin{aligned}
&\leq \xi^k[\mathcal{L}(\boldsymbol{\beta}^0) - \mathcal{L}(\boldsymbol{\beta}^*)] \\
&\leq \xi^k(K+T)\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty \\
&\quad \times (\left\|\nabla\mathcal{L}(\boldsymbol{\beta}^*)\right\|_\infty + \frac{U}{2}\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty) \\
&\leq \frac{\xi^k(L+U)(K+T)}{2}\left\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\right\|_\infty^2.
\end{aligned}
$$

Hence, by (17), we have

$$
\begin{aligned}
\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k\|\boldsymbol{\beta}^0 - \boldsymbol{\beta}^*\|_\infty \\
+ \frac{2}{L}\|\nabla\mathcal{L}(\boldsymbol{\beta}^*)\|_\infty.
\end{aligned}
$$

□

Lemma 5 (Proof of Corollary 2 in Loh and Wainwright (2015)). *Assume $x_{ij}^{\cdot}s$ are sub-Gaussian and $n \gtrsim \log(p)$, then there exists universal constants (c_1, c_2, c_3) with $0 < c_i < \infty$, $i = 1, 2, 3$ such that*

$$
P\left(\|\nabla\mathcal{L}(\boldsymbol{\beta}^*)\|_\infty \geq c_1\sqrt{\frac{\log(p)}{n}}\right) \leq c_2\exp(-c_3\log(p)).
$$

A.4 Proof of Theorem 1

Proof By Lemma 3, we have

$$
\mathcal{L}(\boldsymbol{\beta}^{k+1}) - \mathcal{L}(\boldsymbol{\beta}^*) \leq \xi[\mathcal{L}(\boldsymbol{\beta}^k) - \mathcal{L}(\boldsymbol{\beta}^*)],
$$

where

$$
\xi = 1 - \frac{2L(1-TU)}{T(1+K)} \in (0,1).
$$

So the conditions of Lemma 4 are satisfied. Taking $\boldsymbol{\beta}^0 = \mathbf{0}$, we can get

$$
\begin{aligned}
&\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \\
&\quad \leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k\|\boldsymbol{\beta}^*\|_\infty + \frac{2}{L}\|\nabla\mathcal{L}(\boldsymbol{\beta}^*)\|_\infty.
\end{aligned}
$$

By Lemma 5, then there exists universal constants (c_1, c_2, c_3) defined in Lemma 5, with at least probability $1 - c_2\exp(-c_3\log(p))$, we have

$$
\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty
$$

$$\leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k \|\boldsymbol{\beta}^*\|_\infty + \frac{2c_1}{L}\sqrt{\frac{\log(p)}{n}}. \quad (18)$$

Some algebra shows that

$$\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \leq \mathcal{O}(\sqrt{\frac{\log(p)}{n}})$$

by taking $k \geq \mathcal{O}(\log_{\frac{1}{\xi}} \frac{n}{\log(p)})$ in (18). Then, the proof is complete. □

A.5 Proof of Theorem 2

Proof (18) and assumption (C2) and some algebra shows that that

$$\begin{aligned}
&\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^*\|_\infty \\
&\leq \sqrt{(K+T)(1+\frac{U}{L})}(\sqrt{\xi})^k \|\boldsymbol{\beta}^*\|_\infty + \frac{2}{3}\|\boldsymbol{\beta}^*_{A^*}\|_{\min} \\
&< \|\boldsymbol{\beta}^*_{A^*}\|_{\min},
\end{aligned}$$

if $k > \log_{\frac{1}{\xi}} 9(T+K)(1+\frac{U}{L})r^2$. This implies that $A^* \subseteq A^k$. □

References

Abramovich F, Benjamini Y, Donoho DL, Johnstone IM et al (2006) Adapting to unknown sparsity by controlling the false discovery rate. Ann Stat 34(2):584–653

Birgé L, Massart P (2001) Gaussian model selection. J Eur Math Soc 3(3):203–268

Bogdan M, Ghosh JK, Doerge R (2004) Modifying the schwarz bayesian information criterion to locate multiple interacting quantitative trait loci. Genetics 167(2):989–999

Bogdan M, Ghosh JK, Żak-Szatkowska M (2008) Selecting explanatory variables with the modified version of the Bayesian information criterion. Qual Reliab Eng Int 24(6):627–641

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

Breheny P, Huang J (2015) Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. Stat Comput 25(2):173–187

Chen J, Chen Z (2008) Extended bayesian information criteria for model selection with large model spaces. Biometrika 95(3):759–771

Chen J, Chen Z (2012) Extended bic for small-n-large-p sparse glm. Stat Sinica 22:555–574

Chen X, Nashed Z, Qi L (2000) Smoothing methods and semismooth methods for nondifferentiable operator equations. SIAM J Numer Anal 38(4):1200–1216

Chen X, Ge D, Wang Z, Ye Y (2014) Complexity of unconstrained l2-lp minimization. Math Program 143(1–2):371–383

Dolejsi E, Bodenstorfer B, Frommlet F (2014) Analyzing genome-wide association studies with an fdr controlling modification of the bayesian information criterion. PloS One 9(7):e103322

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

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

Frommlet F, Nuel G (2016) An adaptive ridge procedure for l 0 regularization. PloS One 11(2):e0148620
Frommlet F, Ruhaltinger F, Twaróg P, Bogdan M (2012) Modified versions of Bayesian information criterion for genome-wide association studies. Comput Stat Data Anal 56(5):1038–1051
Frommlet F, Bogdan M, Ramsey D (2016) Phenotypes and genotypes. Springer, Berlin
Ge J, Li X, Jiang H, Liu H, Zhang T, Wang M, Zhao T (2019) Picasso: A sparse learning library for high dimensional data analysis in R and Python. J Mach Learn Res 20(44):1–5
Huang J, Jiao Y, Liu Y, Lu X (2018) A constructive approach to l 0 penalized regression. J Mach Learn Res 19(1):403–439
Huang J, Jiao Y, Jin B, Liu J, Lu X, Yang C (2021) A unified primal dual active set algorithm for nonconvex sparse recovery. Stat Sci (to appear). arXiv:1310.1147
Jiao Y, Jin B, Lu X (2017) Group sparse recovery via the $\ell^0(\ell^2)$ penalty: theory and algorithm. IEEE Trans Signal Process 65(4):998–1012
Li X, Yang L, Ge J, Haupt J, Zhang T, Zhao T (2017) On quadratic convergence of DC proximal Newton algorithm in nonconvex sparse learning. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30, Curran Associates, Inc
Loh P-L, Wainwright MJ (2015) Regularized m-estimators with nonconvexity: statistical and algorithmic theory for local optima. J Mach Learn Res 16:559–616
Louizos C, Welling M, Kingma DP (2018) Learning sparse neural networks through L_0 regularization. In: *International conference on learning representations*, pp 1–13. URL https://openreview.net/forum?id=H1Y8hhg0b
Luo S, Chen Z (2014) Sequential lasso cum ebic for feature selection with ultra-high dimensional feature space. J Am Stat Assoc 109(507):1229–1240
Ma R, Miao J, Niu L, Zhang P (2019) Transformed ℓ_1 regularization for learning sparse deep neural networks. URL https://arxiv.org/abs/1901.01021
McCullagh P (2019) Generalized linear models. Routledge, Oxfordshire
Meier L, Van De Geer S, Bühlmann P (2008) The group lasso for logistic regression. J R Stat Soc: Ser B (Stat Methodol) 70(1):53–71
Natarajan BK (1995) Sparse approximate solutions to linear systems. SIAM J Comput 24(2):227–234
Nelder JA, Wedderburn RW (1972) Generalized linear models. J R Stat Soc: Ser A (General) 135(3):370–384
Nesterov Y (2013) Gradient methods for minimizing composite functions. Math Program 140(1):125–161
Park MY, Hastie T (2007) L1-regularization path algorithm for generalized linear models. J R Stat Soc: Ser B (Stat Methodol) 69(4):659–677
Qi L (1993) Convergence analysis of some algorithms for solving nonsmooth equations. Math Oper Res 18(1):227–244
Qi L, Sun J (1993) A nonsmooth version of Newton's method. Math Program 58(1–3):353–367
Rockafellar RT, Wets RJ-B (2009) Var Anal. Springer Science & Business Media, Berlin
Scardapane S, Comminiello D, Hussain A, Uncini A (2017) Group sparse regularization for deep neural networks. Neurocomputing 241:81–89
Schwarz G et al (1978) Estimating the dimension of a model. Ann Stat 6(2):461–464
Shen J, Li P (2017) On the iteration complexity of support recovery via hard thresholding pursuit. In: *Proceedings of the 34th international conference on machine learning-volume 70*, pp 3115–3124
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc: Ser B (Methodological) 58(1):267–288
Van de Geer SA et al (2008) High-dimensional generalized linear models and the lasso. Ann Stat 36(2):614–645
Wang L, Kim Y, Li R (2013) Calibrating non-convex penalized regression in ultra-high dimension. Ann Stat 41(5):2505
Wang R, Xiu N, Zhou S (2019) Fast newton method for sparse logistic regression. arXiv preprint arXiv:1901.02768
Wang Z, Liu H, Zhang T (2014) Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. Ann Stat 42(6):2164
Ye F, Zhang C-H (2010) Rate minimaxity of the lasso and dantzig selector for the lq loss in lr balls. J Mach Learn Res 11(Dec):3519–3540
Yuan X-T, Li P, Zhang T (2017) Gradient hard thresholding pursuit. J Mach Learn Res 18:166
Żak-Szatkowska M, Bogdan M (2011) Modified versions of the bayesian information criterion for sparse generalized linear models. Comput Stat Data Anal 55(11):2908–2924

Zhang C-H (2010) Nearly unbiased variable selection under minimax concave penalty. Ann Stat 38(2):894–942
Zhang C-H, Zhang T et al (2012) A general theory of concave regularization for high-dimensional sparse estimation problems. Stat Sci 27(4):576–593
Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc: Ser B (Stat Methodol) 67(2):301–320