Electronic Journal of Statistics
Vol. 4 (2010) 1055–1096
ISSN: 1935-7524
DOI: 10.1214/10-EJS578

Sparse regression with exact clustering

Yiyuan She

Department of Statistics
Florida State University
Tallahassee, FL 32306-4330
e-mail: yshe@stat.fsu.edu

Abstract: This paper studies a generic sparse regression problem with a customizable sparsity pattern matrix, motivated by, but not limited to, a supervised gene clustering problem in microarray data analysis. The clustered lasso method is proposed with the l_1-type penalties imposed on both the coefficients and their pairwise differences. Somewhat surprisingly, it behaves differently than the lasso or the fused lasso – the exact clustering effect expected from the l_1 penalization is rarely seen in applications. An asymptotic study is performed to investigate the power and limitations of the l_1-penalty in sparse regression. We propose to combine data-augmentation and weights to improve the l_1 technique. To address the computational issues in high dimensions, we successfully generalize a popular iterative algorithm both in practice and in theory and propose an 'annealing' algorithm applicable to generic sparse regressions (including the fused/clustered lasso). Some effective accelerating techniques are further investigated to boost the convergence. The accelerated annealing (AA) algorithm, involving only matrix multiplications and thresholdings, can handle a large design matrix as well as a large sparsity pattern matrix.

AMS 2000 subject classifications: Primary 62J07, 62H30.
Keywords and phrases: Sparsity, clustering, thresholding, lasso.

Received January 2010.

1. Background

This paper assumes a regression setup

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \tag{1.1}$$

where $\boldsymbol{y}$ is the observed response vector and $\boldsymbol{X}$ is the regression (design) matrix of size n-by-p. The main goal is to recover $\boldsymbol{\beta}$ under some sparsity assumptions. One typical assumption is that $\boldsymbol{\beta}$ is sparse in the sense that many of its components are zero (referred to as the *zero-sparsity* in this paper), where the lasso [26] by solving the following convex optimization problem is a popular method

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

or stated in a multi-objective way [6]

$$\min(\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2, \|\boldsymbol{\beta}\|_1). \tag{1.2}$$

The test error is yet always the first concern in fitting a regression model, which is assumed throughout the paper. One advantage of the lasso lies in its computational feasibility even for large-scale data. For some concrete computation procedures, we refer to the LARS (Efron *et al.* [13]), the homotopy method (Osborne *et al.* [22]), and a recently re-discovered iterative algorithm (Fu [15], Daubechies *et al.*, Friedman *et al.* [14], Wu & Lange [29]) among others. There are numerous theoretical works on the zero-sparsity, Bunea *et al.* [8], Zhang & Huang [31], Candès and Tao [9], to name a few.

On the other hand, motivated by the intuition that the l_1-norm is a convex relaxation of the l_0-norm, researchers have tried far more l_1-type penalties to capture various types of sparsity, especially in the field of signal processing and computer vision. Unfortunately, there is not much theoretical analysis in the literature, and there is a lack of scalability of current computational algorithms in very high dimensions. This paper aims to shed light on a range of issues related to l_1 sparsity recovery in a general setup.

The rest of the paper is organized as follows. Motivated by a gene clustering problem, Section 2 proposes a clustered lasso method, and provides a generic sparse regression framework with customizable sparsity patterns. A theoretical study is performed on the power and limitations of the l_1-penalty in Section 3. Improving techniques of data-augmentation and weights are also investigated. Section 4 tackles the computation problem in high-dimensional space by developing an iterative algorithm with theoretical justifications. It can be seen as a generalization of the popular coordinate descent algorithm. All technical details are left to the Appendices.

2. Clustered lasso

The motivation of this paper is to perform a microarray study to discover aging-related genes. The microarray dataset consists of large-scale gene expression data of 133 human kidney samples. The gene expression matrix $\boldsymbol{X}$ is of size $133 \times 44{,}928$, and the responses, $\boldsymbol{y}$, are the ages of the 133 subjects. After normalizing the data, one can run the lasso to classify the large number of genes as relevant and irrelevant factors in response to age. The number of relevant genes is limited by n. To deeply study the gene effects and to obtain possibly more relevant variables in the model, a reasonable idea is to make the nonzero coefficients come out **equal** in clusters. As a form of regularization, this is much more interpretative (in terms of average expression values) than the ridge regression. One can construct group-based predictors that are measured more accurately and are less sensitive to noise. Later, if some gene expression values are missing in the microarrays, these groups can be used to impute the missing values. Finally, the gene groups may provide some biological insights to identifying the functionally-related genes that are coexpressed in response to age. (See Dettling and Bühlmann [11] or Jörnsten and Yu [17] for a detailed description of this biological motivation.) In consideration of these benefits, we would like to identify and group relevant variables based on their effects (coefficients).

The proposed problem requires combined regression and clustering analysis. One possible way is to directly apply some clustering procedure to the estimated coefficients, which often results in an *ad-hoc* algorithm. The estimate from the fitting stage may not be stable. To carry out the clustering task in the second stage, one needs to specify a similarity measure and the number of clusters. Typically, the standard error information of the estimate is discarded in this step. More importantly, the clustering criterion is different than the test error, so the obtained clusters may not improve model fitting at all. For high-dimensional data, this two-stage approach is unstable and inaccurate. Alternatively, a more ambitious and more trustworthy means is to take the clusters into account when fitting the regression model, which can be achieved by integrating a penalty for improper clustering into the objective function. We refer to it as *sparse regression with exact clustering*. The notion of "exactness" is necessary in proper clustering because without the standard error information, statisticians cannot determine how close two estimates say $\hat{\beta}_i$ and $\hat{\beta}_j$ are, even if the gap between them is small; however, once getting a gap estimate *exactly* equal to 0, one usually has enough confidence to put gene i and gene j into the same group. This exactness also enhances model parsimony (in comparison to the ridge regression or the lasso) – the number of degrees of freedom of the model is essentially the number of nonzero clusters.

In the language of multi-objective optimization [6], the problem can be formulated into

$$\min(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2, \|\boldsymbol{\beta}\|_0, \sum_{i<j} 1_{\beta_i\neq\beta_j}). \tag{2.1}$$

Two types of sparsity are desirable: *zero-sparsity* and *equi-sparsity*, achieved by minimizing $\|\boldsymbol{\beta}\|_0$ and $\sum_{i<j} 1_{\beta_i\neq\beta_j}$, respectively. The problem is a combinatorial optimization and is NP-hard [1]. Motivated by the fact that the l_1-penalty is a convex approximation of the l_0-penalty in optimization, we may try to minimize

$$(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2, \|\boldsymbol{\beta}\|_1, \sum_{i<j} |\beta_i-\beta_j|),$$

or equivalently,

$$\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\sum_{i<j}|\beta_i-\beta_j|, \tag{2.2}$$

referred to as the **clustered lasso**.

Remark. Though similar in form, the clustered lasso (2.2) is different than the *fused lasso* (Tibshirani *et al.* [27]) in that it does not require the regression features to be ordered and so the clustering problem is much more challenging. In fact, the clustered lasso organizes all features and thus can be used as a pre-processing step for the fused lasso. This idea is used later in Section 3.3 in the algorithm design. It is also worthwhile to make a comparison between the clustered lasso and the *grouped lasso* methods [30, 32].

Grouped lasso assumes the grouping of features (predictors) is known, arising naturally from the underlying background, such as the dummy variables introduced for a multi-level factor. The coefficients within the same predictor group are not necessarily equal. The clustered lasso performs **supervised** clustering and groups the predictors taking into account both $\boldsymbol{X}$ and $\boldsymbol{y}$. Bondell and Reich's OSCAR [5] is close in spirit in this sense which minimizes $\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda_1\|\boldsymbol{\beta}\|_1+\lambda_2\sum_{i<j}|\beta_i|\vee|\beta_j|$. It is not difficult to see that the objective function can be written as $\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda_1'\|\boldsymbol{\beta}\|_1+\lambda_2\sum_{i<j}(|\beta_i|-|\beta_j|)$, and thus OSCAR seeks zero-sparsity and equi-sparsity in $|\boldsymbol{\beta}|$.

It is more convenient to introduce a general framework for sparse regression where the objective is to obtain a regression estimate with $\boldsymbol{T}$-sparsity, i.e.,

$$\min(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2,\|\boldsymbol{T}\boldsymbol{\beta}\|_0), \tag{2.3}$$

where $\boldsymbol{T}$ is the sparsity pattern matrix specified by the user. A feasible alternative to overcome the NP-hardness is to solve

$$\min(\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2,\|\boldsymbol{T}\boldsymbol{\beta}\|_1). \tag{2.4}$$

The $\boldsymbol{T}$ matrix can be used to characterize *coding sparsity* in $\boldsymbol{\beta}$, not only the zero-sparsity in the narrow sense. Some examples are presented as follows.

Example 2.1 (Mixed $\boldsymbol{T}$)**.** *Suppose a priori knowledge of $\boldsymbol{\beta}$ is available: the successive differences of $(\beta_1,\beta_2,\beta_3)$ are equal, β_3 equals β_4, and β_5 is zero. Then $\boldsymbol{T}$ may include rows of*

$$\begin{bmatrix} 1 & -2 & 1 & & \\ & & 1 & -1 & \\ & & & & 1 \end{bmatrix}$$

to capture all sparsity in fitting the regression model.

Example 2.2 (Clustered/Fused lasso)**.** *In our clustered lasso problem,*

$$\boldsymbol{T}=\begin{bmatrix} \boldsymbol{I} \\ \lambda\boldsymbol{F} \end{bmatrix}, \tag{2.5}$$

where $\boldsymbol{F}$ is a matrix including all pairwise differences (see (4.28)). And the fused lasso [27] replaces the $\boldsymbol{F}$ in (2.5) by a neighboring difference matrix (see (4.26)).

Example 2.3 (Dense $\boldsymbol{T}$)**.** *$\boldsymbol{T}$ can be given by a wavelet transformation matrix (possibly overcomplete), which is useful in signal denoising and compression.*

Example 2.4 (Spatial $\boldsymbol{T}$)**.** *In the field of computer vision and image processing, there exist many meaningful choices for $\boldsymbol{T}$, which can be constructed from various spatial operators, such as the following* Laplacian of Gaussian *used in edge detection*

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

The two-dimensional fused lasso [14] also makes an example here.

In summary, the customizable $\boldsymbol{T}$ represents the sparsity requirement posed in regression analysis. Unless otherwise specified, our studies in the rest of the paper are toward an arbitrarily given $\boldsymbol{T}$ matrix.

3. Limitations and improvements of the clustered lasso

Somewhat surprisingly, the plain clustered lasso (2.2) suffers some serious problems. Its test error is often not small although it has two regularization parameters. More importantly, it barely clusters the predictors properly in experiments. We demonstrate an example as follows. Let $\boldsymbol{\beta}^T = (\{0\}^3, \{4\}^5, \{-4\}^5, \{2\}^2, \{-8\}^1)$, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n]^T$, $\boldsymbol{x}_i$ i.i.d. $\sim$ MVN$(\boldsymbol{0}, \sigma^2\boldsymbol{\Sigma})$, where $\sigma = 5$ and $\Sigma_{ij} = (-1)^{(i-j)}0.8$. This is a parsimonious model with only 4 degrees of freedom (4 nonzero clusters). The training sample size is 100. To make the clustered lasso less affected by various parameter tuning strategies, we generate a large enough validation dataset (of size 1000) to find the optimal λ_1 and λ_2. Ideally, $\boldsymbol{T\beta}$ have 27 zeros, 3 corresponding to the zero-sparsity, and 24 corresponding to the equi-sparsity. But for the clustered lasso estimate $\hat{\boldsymbol{\beta}}$, $\boldsymbol{T}_z\hat{\boldsymbol{\beta}}$ hardly shows exact-clustering effects, as demonstrated by Figure 1. Although setting the regularization parameters in the objective function to be large results in more zeros in $\boldsymbol{T}\hat{\boldsymbol{\beta}}$, the fitted model is often poor in both estimation and prediction. The problem exists for other ad-hoc parameter tunings. Increasing the sample size does not resolve the issue, either. Our theorem below reveals that this bizarre

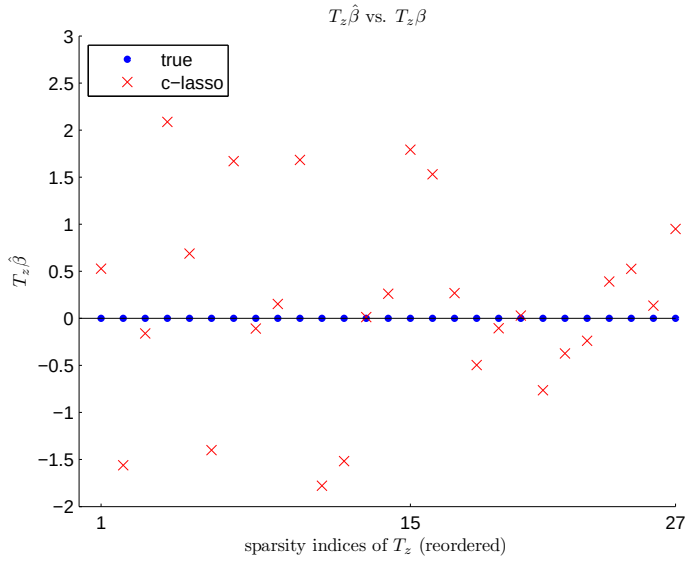

FIG 1. *The plain clustered lasso does not show enough exact-clustering effect. The true* $\boldsymbol{\beta}$ *is sparse in that* $\boldsymbol{T\beta}$ *has 27 exact zeros.*

behavior is in fact due to the l_1 relaxation (see Theorem 3.1). In this section, we study the limitations of the l_1 technique in $\boldsymbol{T}$-sparsity recovery and propose some effective improvements.

3.1. Power and limitations of the L_1-penalty in sparse regression

It is widely known that the l_1-norm penalty is a convex approximation to the l_0-norm penalty in optimization. For instance, the variable selection problem can be formulated as an l_0-minimization and discovering the best subset of predictors is NP-hard. The lasso replaces the l_0-norm with the l_1-norm in the criterion and offers a computationally feasible way to tackle this problem. However, it may not be selection consistent for coherent designs [33, 34]. For a general $\boldsymbol{T}$, the nature of this l_1 approximation is worth careful study in theory.

For clarity, we adopt the generalized sign notation. Introduce $\widetilde{\mathrm{Sgn}}(\boldsymbol{v}) = \{\boldsymbol{s} : s_i = 1 \text{ if } v_i > 0, s_i = -1 \text{ if } v_i < 0, \text{ and } s_i \in [-1,1] \text{ if } v_i = 0\}$, and $\widetilde{\mathrm{sgn}}(\boldsymbol{v})$ is used to denote a specific element in $\widetilde{\mathrm{Sgn}}(\boldsymbol{v})$. The usual sign vector is defined as $\mathrm{sgn}(\boldsymbol{v}) = \{\boldsymbol{s} : s_i = 1 \text{ if } v_i > 0, s_i = -1 \text{ if } v_i < 0, \text{ and } s_i = 0 \text{ if } v_i = 0\}$. Let $\hat{\boldsymbol{\beta}}$ be an optimal solution to the generic sparse regression

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{T}\boldsymbol{\beta}\|_1. \tag{3.1}$$

$\boldsymbol{T}$ may not have full rank. By the KKT optimality conditions [25] (the nonsmooth version), $\hat{\boldsymbol{\beta}}$ is an optimal solution if and only if $\hat{\boldsymbol{\beta}}$ satisfies

$$\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}) + \lambda \boldsymbol{T}^T \widetilde{\mathrm{sgn}}(\boldsymbol{T}\boldsymbol{\beta}) = \boldsymbol{0},$$

for some $\widetilde{\mathrm{sgn}}(\boldsymbol{T}\boldsymbol{\beta})$. We work in a classical setting ($\mathcal{C}$): assume that $p, \boldsymbol{\beta}$ are fixed and $n \to \infty$; $\boldsymbol{\Sigma} \triangleq \boldsymbol{X}^T\boldsymbol{X}/n \to \boldsymbol{C}$, a positive definite matrix, with probability 1. Throughout this paper, given a matrix $\boldsymbol{A}$, we use $\boldsymbol{A}_I$ to denote the submatrix of $\boldsymbol{A}$ composed of the rows indexed by I, such that $\boldsymbol{A}_I\boldsymbol{\alpha} = (\boldsymbol{A}\boldsymbol{\alpha})_I, \forall \boldsymbol{\alpha}$. Given two matrices $\boldsymbol{A}, \boldsymbol{B}$, $\boldsymbol{B} \subset \boldsymbol{A}$ means that $\boldsymbol{B}$ is a submatrix of $\boldsymbol{A}$, composed of certain rows of $\boldsymbol{A}$. Hence $\boldsymbol{A}_I \subset \boldsymbol{A}$.

Proposition 3.1. *If $\lambda = o(n)$, then $\hat{\boldsymbol{\beta}} \xrightarrow{P} \boldsymbol{\beta}$, and so $\boldsymbol{T}\hat{\boldsymbol{\beta}} \xrightarrow{P} \boldsymbol{T}\boldsymbol{\beta}$.*

Consistency is a weak requirement, placing no restrictions on $\boldsymbol{\Sigma}$ or $\boldsymbol{T}$. It can be easily achieved by a properly chosen λ. Yet in using the l_1 penalty, we expect something more in sparsity recovery. In this paper, we are more interested in another notion of consistency.

Definition 3.1. *(Sign consistency, Zhao and Yu [33]) Let $\hat{\boldsymbol{\theta}}$ be a sequence of estimators of $\boldsymbol{\theta}$. Then $\hat{\boldsymbol{\theta}}$ is defined to be sign-consistent if $P(sgn(\hat{\boldsymbol{\theta}}) = sgn(\boldsymbol{\theta})) \to 1$.*

Note that consistency implies nonzero sign consistency. For example, from Proposition 3.1, $P(\mathrm{sgn}(\hat{\boldsymbol{\beta}}_I) = \mathrm{sgn}(\boldsymbol{\beta}_I)) \to 1$ for $I = \{i : \boldsymbol{\beta}_i \neq 0\}$, and $P(\mathrm{sgn}((\boldsymbol{T}\hat{\boldsymbol{\beta}})_{I'}) = \mathrm{sgn}((\boldsymbol{T}\boldsymbol{\beta})_{I'})) \to 1$ for $I' = \{i : (\boldsymbol{T}\boldsymbol{\beta})_i \neq 0\}$.

Definition 3.2. *(Zero s-consistency) Let $\hat{\boldsymbol{\theta}}$ be a sequence of estimators of $\boldsymbol{\theta}$ satisfying $\boldsymbol{A}\boldsymbol{\theta} = \mathbf{0}$ for some matrix $\boldsymbol{A}$. $\hat{\boldsymbol{\theta}}$ is defined to be zero s-consistent with respect to $\boldsymbol{A}$ if $P(\boldsymbol{A}\hat{\boldsymbol{\theta}} = \mathbf{0}) \to 1$.*

The zero-s consistency is a key notion used to characterize sparsity recovery. For example, in the clustered lasso, zero-s consistency means successfully discovering all the true groups asymptotically. Returning to our $\boldsymbol{T}$-sparsity problem, define $z = z(\boldsymbol{T}, \boldsymbol{\beta}) = \{i : (\boldsymbol{T}\boldsymbol{\beta})_i = 0\}$, $nz = nz(\boldsymbol{T}, \boldsymbol{\beta}) = \{i : (\boldsymbol{T}\boldsymbol{\beta})_i \neq 0\}$. Then we have the following result.

Proposition 3.2. *If $\lambda = O(\sqrt{n})$, i.e., $\limsup_{n\to\infty} \lambda/\sqrt{n} < \infty$, then $\hat{\boldsymbol{\beta}}$ is* not *zero s-consistent with respect to $\boldsymbol{T}_z$.*

In the following studies, the *joint* zero s-consistency will be our main concern. Namely, we study the conditions for zero s-consistency (with respect to some $\boldsymbol{T}_1 \subset \boldsymbol{T}_z$) under the consistency assumption. This is because in practice although blindly increasing λ would bring into play the thresholding power of the l_1-penalty, we prefer a tuned value of λ with small test error (like the one obtained from cross-validation). The consistency requirement complies with the usual tuning criteria.

Theorem 3.1. *Assume the classical setup (C); $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1 \\ \boldsymbol{T}_2 \end{bmatrix}$, $\boldsymbol{T}_1\boldsymbol{\beta} = \mathbf{0}$; $\lambda/n \to 0$, $\lambda/\sqrt{n} \to \infty$. We use $\boldsymbol{A}^+$ to denote the Moore-Penrose inverse of $\boldsymbol{A}$. Then a necessary condition for $\hat{\boldsymbol{\beta}}$ to be zero s-consistent w.r.t. $\boldsymbol{T}_1$ is*

$$\begin{aligned} \exists \widetilde{sgn}(\boldsymbol{T}_2\boldsymbol{\beta}) \text{ s.t. } &\|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_2^T) \cdot \widetilde{sgn}(\boldsymbol{T}_2\boldsymbol{\beta})\|_\infty \\ &\leq \|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)\|_\infty, \end{aligned} \tag{3.2}$$

and a sufficient condition is given by

$$\|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_2^T) \cdot \widetilde{sgn}(\boldsymbol{T}_2\boldsymbol{\beta})\|_\infty < 1, \forall \widetilde{sgn}(\boldsymbol{T}_2\boldsymbol{\beta}). \tag{3.3}$$

For a concrete example, suppose $\boldsymbol{T}_z$ has full row rank and substitute $\boldsymbol{T}_z$ for $\boldsymbol{T}_1$, and $\boldsymbol{T}_{nz}$ for $\boldsymbol{T}_2$. Then, (3.2) and (3.3) become

$$\|(\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_z^T)^{-1}(\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T) \cdot \operatorname{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta})\|_\infty \leq 1. \tag{3.4}$$

and

$$\|(\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_z^T)^{-1}(\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T) \cdot \operatorname{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta})\|_\infty < 1, \tag{3.5}$$

respectively. Thus the sufficient condition is pretty strong. Simple algebra also shows that they further reduce to the *irrepresentable conditions* [33, 34] in the lasso case where $\boldsymbol{T} = \boldsymbol{I}$.

As another interesting example, suppose $(\boldsymbol{T}_1, \boldsymbol{T}_2)$ is 'separable' in the sense that $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1 \\ \boldsymbol{T}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_{11} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{T}_{22} \end{bmatrix}$. We write $\boldsymbol{C}$ in a corresponding block form $\begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{12}^T & \boldsymbol{C}_2 \end{bmatrix}$, and assume that $\boldsymbol{C}_2$ is nonsingular. Then the LHS of (3.3) becomes

$$\|(\boldsymbol{T}_{11}\boldsymbol{S}^{-1}\boldsymbol{T}_{11}^T)^+\boldsymbol{T}_{11}\boldsymbol{S}^{-1}\boldsymbol{C}_{12}\boldsymbol{C}_2^{-1}\boldsymbol{T}_{22}^T \cdot \widetilde{\operatorname{sgn}}(\boldsymbol{T}_2\boldsymbol{\beta})\|_\infty,$$

where $\boldsymbol{S} = \boldsymbol{C}_1 - \boldsymbol{C}_{12}\boldsymbol{C}_2^{-1}\boldsymbol{C}_{12}^T$. Therefore, if the entries of $\boldsymbol{C}_{12}$ are small enough (in absolute value), the zero s-consistency w.r.t. $\boldsymbol{T}_1$ naturally follows. (Note that $(\boldsymbol{T}_{11}\boldsymbol{S}^{-1}\boldsymbol{T}_{11}^T)^+$ is a continuous function of $\boldsymbol{C}_{12}$ since the rank of $\boldsymbol{T}_{11}\boldsymbol{S}^{-1}\boldsymbol{T}_{11}^T$ is preserved.) This conclusion coincides with the lasso studies where $\boldsymbol{T} = \boldsymbol{I}$ (see, e.g., Zhao and Yu [33]). Unfortunately, the clustered lasso does not fall into this class because the rows of the $\boldsymbol{T}$ encompass all pairwise differences and thus never result in a separable $(\boldsymbol{T}_z, \boldsymbol{T}_{nz})$.

Theorem 3.1 indicates that in contrast to consistency, zero s-consistency imposes further constraints on $\boldsymbol{\Sigma}$ (the data) aside from the controllable regularization parameter λ. Without going into the mathematical details, these conditions intuitively mean that one should have good control over $(\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_z^T)^+ \cdot (\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T)$ to ensure the l_1 penalty is effective for sparsity recovery. For instance, if we consider the joint zero s-consistency with respect to $\boldsymbol{T}_1$ for all signals satisfying $\boldsymbol{T}_1\boldsymbol{\beta} = \boldsymbol{0}$, the sufficiency condition (3.3) becomes

$$\|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_2^T)\|_\infty < 1. \tag{3.6}$$

Hence the magnitude of the entries of $(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{T}_1\boldsymbol{C}^{-1} \cdot \boldsymbol{T}_2^T$ plays an important role. Given $\boldsymbol{\beta}$, $\boldsymbol{T}_1$, and $\boldsymbol{C}$, (3.6) makes a huge difference between the fused lasso and the clustered lasso: the $\boldsymbol{T}_2$ of the clustered lasso contains up to $O(p^2)$ more rows in addition to the $\boldsymbol{T}_2$ of the fused lasso. Recalling that the matrix infinity norm is the maximum of the l_1-norms of all rows, we see the clustered lasso is certainly more likely to break (3.6).

We illustrate the conditions with the previous example in Figure 1 for both clustered lasso and fused lasso (which uses the correct ordering from the true $\boldsymbol{\beta}$). For convenience, we ignore the zero-sparsity constraint and concentrate on the equi-sparsity. Substituting $\boldsymbol{T}_z$ for $\boldsymbol{T}_1$ in (3.2), the LHS equals 0.6 and the RHS equals 1 for the fused lasso, but these quantities are 3.0 and 1.6 respectively for the clustered lasso. In (3.6), the LHS is only 1.7 for the fused lasso, but 31.2 for the clustered lasso. The fused lasso and the clustered lasso (though similar in form) thus show remarkable difference in the behavior of the l_1-penalty, the latter much more difficult to recover the true sparsity even asymptotically.

This explains the dilemma we encountered earlier. No matter how we devise a scheme to tune the regularization parameters, the design criterion favors the models with small generalization error. Therefore, the tuned regularization parameters cannot be very large seen from Proposition 3.1 (if we do not want our estimate to be inconsistent). But Proposition 3.2 and Theorem 3.1 then limit the l_1's ability to enforce sparsity. Although this requirement on $\boldsymbol{\Sigma}$ might not be very restrictive for the lasso or even for the fused lasso, it becomes so stringent for the clustered lasso that the expected exact-clustering effect is seldom seen strong enough in applications. In the next subsection, we propose different means to improve the naïve l_1-penalty to gain exact clustering.

3.2. Improving techniques

3.2.1. Weights

To further improve the sparsity weights can be added into the l_1 norm. Zou [34] shows that asymptotically, this weighted form of lasso (adaptive lasso) is sign consistent and enjoys the oracle properties. This technique applies to the generic sparse regression (3.1). According to Theorem 3.1, if we could rescale the rows of $\boldsymbol{T}$ in an ideal way

$$\boldsymbol{DT} = \begin{bmatrix} \boldsymbol{I} & \\ & \varepsilon \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{T}_z \\ \boldsymbol{T}_{nz} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_z \\ \varepsilon \boldsymbol{T}_{nz} \end{bmatrix} \triangleq \boldsymbol{T}',$$

then, the LHSs of (3.2) and (3.3) may be reduced significantly for ε small enough, while the RHSs remain unchanged. In fact, one of the advantages of the fused lasso (see example 2.2) is that the two regularization parameters provide adaptive weights for the components of $\boldsymbol{T\beta}$. This weight construction is based on different types of sparsity. For a general $\boldsymbol{T}$, however, it may not be possible to supply this information. Furthermore, it is really between the zero components ($\boldsymbol{T}_z\boldsymbol{\beta}$) and nonzero components ($\boldsymbol{T}_{nz}\boldsymbol{\beta}$) that the weights should make a big difference. We introduce weights for each individual component of $\boldsymbol{T\beta}$ and consider the weighted sparse regression of the form

$$\min \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \lambda \sum w_i |(\boldsymbol{T\beta})_i|. \tag{3.7}$$

where w_i are positive.

Theorem 3.2. *Assume the classical setup (C). Suppose*

$$w_i^{-1} = O_p\left(A(n)\right), \forall j \in z, \quad w_i = O_p\left(B(n)\right), \forall j \in nz.$$

Then for some properly chosen $\lambda(n)$, the optimal solution $\hat{\boldsymbol{\beta}}$ to (3.7) is both zero s-consistent with respect to $\boldsymbol{T}_z$ and $\sqrt{n}$-consistent as long as

$$A(n)B(n) \to 0. \tag{3.8}$$

For example, $w_i^{-1} = O_p(1/\sqrt{n}), \forall j \in z$, and $w_i = O_p\left(1\right), \forall j \in nz$.

(3.8) is a broad condition. Essentially it only requires

$$\max\{w_{nz}\}/\min\{w_z\} \xrightarrow{P} 0, \tag{3.9}$$

and so provides a flexible way for weight construction. We can use $1/w_i = |(\boldsymbol{T}\hat{\boldsymbol{\beta}}_{wts})_i|$ with any consistent estimate $\hat{\boldsymbol{\beta}}_{wts}$. This can be viewed as a generalization of Zou [34]. (In fact, $\hat{\boldsymbol{\beta}}_{wts}$ does not even have to be an estimator of $\boldsymbol{\beta}$ seen from (3.9), which also justifies the use of one-step SCAD weights [36].) On the other hand, one should be aware that Theorem 3.2 is an asymptotic study with p fixed. Therefore, although the weighted l_1-penalty can increase model

sparsity, careful consideration must be given to the practical weight construction especially in large-p applications. Another issue is that adding the weights does not improve the test error very much. It can even hurt the goodness-of-fit to some extent. This is undesirable in statistical modeling and we would like to combine it with the following data augmentation technique.

3.2.2. Data augmentation

It has been recognized that the l_1 with a data-augmentation modification, such as the elastic net (eNet for short), can achieve much smaller test error and can resolve singularities and the 'grouping' issue [35]. To introduce the data-augmentation (DA), we facilitate our discussion by focusing on the zero-sparsity in this subsection. The technique carries over to a customizable $\boldsymbol{T}$ as will shown in Section 3.3.

If one cares about prediction accuracy only, the ridge regression is a good alternative to the lasso. In view of data augmentation, it considers an augmented problem with the design and response given by

$$\begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda}\boldsymbol{I} \end{bmatrix} \to \begin{bmatrix} \boldsymbol{y} \\ \times \end{bmatrix}. \tag{3.10}$$

where $\sqrt{\lambda}\boldsymbol{I}$ decorrelates the predictor columns. The $\times$ part is often $\mathbf{0}$, but data-dependent choices might be better such as using the univariate estimate $\sqrt{\lambda}\hat{\boldsymbol{\beta}}_{uni}$:

$$\hat{\boldsymbol{\beta}}_{uni} = \left[\boldsymbol{x}_i^T\boldsymbol{y}/(\boldsymbol{x}_i^T\boldsymbol{x}_i)\right]_{p\times 1}.$$

In fact, following this idea, we can give the elastic net a natural characterization and explain why an extra factor comes in to correct the naïve eNet [35]. In [35], the naïve eNet is introduced as a combination of the lasso and ridge regression by imposing both the l_1 penalty and the l_2 penalty on $\boldsymbol{\beta}$. Then, to guard against double shrinkage, Zou and Hastie gave an *empirical* way to improve this naïve estimate by multiplying it by a factor of $1+\lambda_2$. The resulting eNet estimate, according to Theorem 2 of [35], is defined by

$$\hat{\boldsymbol{\beta}}^{(eNet)}_{\lambda_1,\lambda_2} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T \frac{\boldsymbol{X}^T\boldsymbol{X}+\lambda_2\boldsymbol{I}}{1+\lambda_2}\boldsymbol{\beta} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} + \lambda_1\|\boldsymbol{\beta}\|_1, \tag{3.11}$$

assuming $\boldsymbol{x}_i^T\boldsymbol{x}_i = 1$ for $i = 1, \cdots, p$. We show the following result.

Theorem 3.3. *Given $\lambda_1, \lambda_2 > 0$, define*

$$\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} = \arg\min \left\| \begin{bmatrix} \boldsymbol{y} \\ \sqrt{\lambda_2}\boldsymbol{X}^T\boldsymbol{y} \end{bmatrix} - \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{I} \end{bmatrix} \boldsymbol{\beta} \right\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1. \tag{3.12}$$

Then $\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} = \hat{\boldsymbol{\beta}}^{(eNet)}_{\frac{\lambda_1}{1+\lambda_2},\lambda_2}$.

In addition, since the eNet uses a search strategy of first picking a grid of values for λ_2, then searching over the λ_1-space for each λ_2 in the grid, the tuned $\hat{\boldsymbol{\beta}}_{\lambda_1^*,\lambda_2^*}$ based on (3.12) coincides with the tuned eNet estimate. In short, the eNet solves the lasso problem of

$$\begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda}\boldsymbol{I} \end{bmatrix} \to \begin{bmatrix} \boldsymbol{y} \\ \sqrt{\lambda}\hat{\boldsymbol{\beta}}_{uni} \end{bmatrix}. \tag{3.13}$$

This indicates the power of the DA: even using a not so accurate estimator ($\hat{\boldsymbol{\beta}}_{uni}$ is not even consistent for nonorthogonal designs) can still effectively reduce the test error.

In the next, we propose a *nondiagonal* manner of data augmentation. To see the motivation we ignore the l_1-constraint for the moment and consider the augmented data fitting

$$\hat{\boldsymbol{\beta}} = \arg\min \left\| \begin{bmatrix} \boldsymbol{y} \\ \sqrt{\lambda_2}\hat{\boldsymbol{\beta}}_{uni} \end{bmatrix} - \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{I} \end{bmatrix} \boldsymbol{\beta} \right\|_2^2.$$

Let $\boldsymbol{X} = \boldsymbol{U}\mathrm{diag}\{d_i\}\boldsymbol{V}^T$ be the SVD. Then $\hat{\boldsymbol{y}} \triangleq \boldsymbol{X}\hat{\boldsymbol{\beta}}$ is given by $\hat{\boldsymbol{y}} = (1+\lambda_2)\cdot \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X}+\lambda_2\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \sum_{i=1}^{p} \frac{d_i^2(1+\lambda_2)}{d_i^2+\lambda_2}(\boldsymbol{u}_i^T\boldsymbol{y})\boldsymbol{u}_i$. For comparison, the OLS fit is $\hat{\boldsymbol{y}}_{ols} = \sum(\boldsymbol{u}_i^T\boldsymbol{y})\boldsymbol{u}_i$. When $d_i < 1$, the projection of $\boldsymbol{y}$ on $\boldsymbol{u}_i$ is shrunk; when $d_i > 1$, the projection on $\boldsymbol{u}_i$ is extended. The reference value '1' is not data-dependent. We would rather replace it by an adaptive scale parameter. One way is to solve the following problem *jointly* for $\boldsymbol{\beta}$ and s

$$\min_{(\boldsymbol{\beta},s)} \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_2\|\boldsymbol{\beta}-s\hat{\boldsymbol{\beta}}_{uni}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1,$$

or equivalently (by optimizing over s),

$$\min \|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_2 \left\| \boldsymbol{\beta} - \frac{\sum \hat{\beta}_{uni,i}\boldsymbol{\beta}_i}{\sum \hat{\beta}_{uni,i}^2}\hat{\boldsymbol{\beta}}_{uni} \right\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1.$$

In general, given an initial estimate $\hat{\boldsymbol{\beta}}_{aug}$, we construct a matrix $\boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}}_{aug})$

$$\boldsymbol{\Lambda}(\boldsymbol{\beta}) \triangleq \begin{cases} \boldsymbol{I} - \boldsymbol{\beta}\boldsymbol{\beta}^T/\|\boldsymbol{\beta}\|_2^2, & \boldsymbol{\beta} \neq \boldsymbol{0} \\ \boldsymbol{I}, & \boldsymbol{\beta} = \boldsymbol{0} \end{cases} \tag{3.14}$$

and propose the non-diagonal data augmentation by solving

$$\min \left\| \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\cdot\boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}}_{aug}) \end{bmatrix} \boldsymbol{\beta} \right\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1. \tag{3.15}$$

Suppose $\hat{\boldsymbol{\beta}}_{aug} \neq \boldsymbol{0}$. The eigenvalues of $\boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}}_{aug})$ are 1's with multiplicity $p-1$, and 0 with multiplicity 1, and the eigenvector corresponding to 0 is $\hat{\boldsymbol{\beta}}_{aug}/\|\hat{\boldsymbol{\beta}}_{aug}\|_2$. It is not difficult to show that the whole input matrix

$$\begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{\Lambda} \end{bmatrix} \tag{3.16}$$

is rank-deficient if and only if all components of $\boldsymbol{X}\hat{\boldsymbol{\beta}}_{aug}$ are exactly equal to 0. Therefore, this resulting nondiagonal DA from introducing a data-dependent scale parameter is still able to decorrelate the covariates in real-world applications.

Interestingly, from an empirical Bayesian point of view, (3.16) corresponds to a multivariate Gaussian prior with a nondiagonal and *degenerate* covariance matrix ($rank(\Lambda) = p - 1$). Therefore, not all information provided by $\hat{\boldsymbol{\beta}}_{aug}$ is used as the prior knowledge. The new data augmentation is more robust and accommodates a less accurate initial estimate. Indeed, the tuning of the regularization parameter λ_2 makes it possible to save one degree of freedom in the construction of $\boldsymbol{\Lambda}$.

3.3. Algorithm design for supervised exact-clustering

The data-augmentation and weights can be combined in sparse regression to reduce test error and increase model sparsity simultaneously. We discuss the practical algorithm design for the supervised exact clustering problem to demonstrate this point.

Given $\hat{\boldsymbol{\beta}}_{aug}$ and $\hat{\boldsymbol{\beta}}_{wts}$, we perform the nondiagonal DA and introduce weights into the l_1 penalty, which amounts to solving the following optimization problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \left\| \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_{aug} \end{bmatrix} - \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{X}_{aug} \end{bmatrix} \boldsymbol{\beta} \right\|_2^2 + \lambda \sum w_i |(T\boldsymbol{\beta})_i|, \tag{3.17}$$

where λ, τ are two regularization parameters, $\boldsymbol{X}_{aug} = \sqrt{\tau}\boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}}_{aug})$, $\boldsymbol{y}_{aug} = \boldsymbol{0}$, and $w_i = 1/|(\boldsymbol{T}\hat{\boldsymbol{\beta}}_{wts})_i|$. (3.17) will be referred to as the DAW version of the sparse regression. In particular, the DAW version of the clustered lasso will be called DAW-CLASSO for convenience.

For the supervised exact clustering problem, we propose to improve the plain clustered lasso estimate (denoted by $\hat{\boldsymbol{\beta}}_{c\text{-}lasso}$) as follows. (Figure 2 plots the outline of the procedure.)

(i) Fit a fused lasso model with the covariates ordered according to $\hat{\boldsymbol{\beta}}_{c\text{-}lasso}$, the estimate denoted by $\hat{\boldsymbol{\beta}}_{f\text{-}lasso}$.

(ii) Substituting $\hat{\boldsymbol{\beta}}_{c\text{-}lasso}$ for $\hat{\boldsymbol{\beta}}_{aug}$ and $\hat{\boldsymbol{\beta}}_{f\text{-}lasso}$ for $\hat{\boldsymbol{\beta}}_{wts}$, fit the DAW-CLASSO, the estimate denoted by $\hat{\boldsymbol{\beta}}_{daw\text{-}classo^1}$.

Then, we can repeat (i) with the covariates re-ordered according to the last estimate $\hat{\boldsymbol{\beta}}_{daw\text{-}classo^1}$, to obtain an updated fused lasso estimate, and then repeat (ii) with $\hat{\boldsymbol{\beta}}_{daw\text{-}classo^1}$ used for data-augmentation and the updated fused lasso estimate for weight construction. The new estimate is denoted by $\hat{\boldsymbol{\beta}}_{daw\text{-}classo^2}$. The experience shows the improvement brought by $\hat{\boldsymbol{\beta}}_{daw\text{-}classo^2}$ is already significant enough, although ideally one can repeat the DAW process within the allowed time.

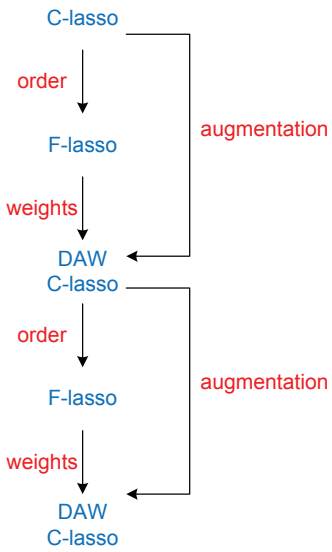

FIG 2. *Algorithm design for the supervised exact clustering.*

3.4. Simulation studies

We carried out simulation experiments to empirically study the improvement brought by DAW. Each simulation dataset contains training data, validation data, and test data, the numbers of observations denoted by # ="$\cdot/\cdot/\cdot$" respectively as follows. The rows of $\boldsymbol{X}$ are independently drawn from $N(\mathbf{0}, \boldsymbol{\Sigma})$. We use $(\{a_1\}^{n_1}, \cdots, \{a_k\}^{n_k})$ to denote a column vector made by n_1 a_1's, $\cdots$, n_k a_k's consecutively.

Example 3.1 (**Many small clusters, overlap likely to occur**). # $= 20/100/100$, $\boldsymbol{\beta} = (\{0\}^2, \{-1.5\}^2, \{-2\}^2, \{0\}^2, \{1\}^2, \{4\}^3)$, $\sigma = 5$, $\Sigma_{ij} = \rho^{|i-j|}$ *with* $\rho = 0.5$. *The degrees of freedom (number of nonzero clusters) of the true model is 4.*

Example 3.2 (Same as the Example 3.1, **much more correlated**). # $= 20/100/100$, $\boldsymbol{\beta} = (\{0\}^2, \{-1.5\}^2, \{-2\}^2, \{0\}^2, \{1\}^2, \{4\}^3)$, $\sigma = 5$, $\Sigma_{ij} = \rho^{|i-j|}$ *with* $\rho = 0.9$.

Example 3.3 (**Big clusters coexist with small clusters, negatively correlated design**). # $= 30/100/100$, $\boldsymbol{\beta} = (\{0\}^3, \{4\}^5, \{-4\}^5, \{2\}^2, \{-8\}^1)$, $\sigma = 5$, $\Sigma_{ij} = (-1)^{(i-j)}0.8$. *The df of the model is 4.*

Example 3.1 and 3.2 demonstrate a situation of many small clusters in the coefficients, where overlap is likely to occur. The design matrix of the second is much more correlated than that of the first. Example 3.3 assumes a more challenging negatively correlated design and the coefficients have big clusters

TABLE 1
Performance comparison of the clustered lasso and its DAW versions in terms of test error and proper sparsity

	Example 3.1		Example 3.2		Example 3.3	
	Test-err	p-Spar	**Test-err**	p-Spar	**Test-err**	p-Spar
C-LASSO	**45.0**	**15.8**%	**22.1**	**22.1**%	**69.3**	**5.3**%
DAW-CLASSO^1	40.2	30.9%	16.3	39.2%	63.2	10.5%
DAW-CLASSO^2	**35.3**	**30.8**%	**15.0**	**37.5**%	**60.5**	**12.5**%

as well as small clusters. The signal to noise variance ratio of this example is only about 1. Note that it is not the goal of this study to propose or advocate a best means for parameter tuning. We set aside a separate validation dataset to tune the parameter. This large validation tuning ensures fair and stable performance comparisons. For those with multiple regularization parameters, we use the *alternative* search strategy [24] which has been shown to be fast and efficacious.

Each model is simulated 50 times. Then we measure the performance of each algorithm by the test error and the proper sparsity. The test error is characterized by the scaled MSE (SMSE) $100 \cdot (\sum_{i=1}^{N}(\hat{y}_i - y_i)^2/(N\sigma^2) - 1)$ on the test data. The proper sparsity is defined as $100\% \cdot |\{i : (\boldsymbol{T}_z\hat{\boldsymbol{\beta}})_i = 0\}|/|z|$ which represents the percentage of *proper* zeros in $\boldsymbol{T}\hat{\boldsymbol{\beta}}$. It is a very sensitive measure for the clustering problem since it takes into account all pairwise comparisons within each cluster.

Table 1 compares the performance of the clustered lasso (C-LASSO) and its DAW versions – DAW-CLASSO^1 and DAW-CLASSO^2 (by performing the DAW process once and twice, see Section 3.3). The clustered lasso does not exhibit enough exact-clustering even in the highly correlated Example 3.2. The inadequate proper sparsity indicates the great challenge of supervised clustering, especially with insufficient training data. A similar phenomenon is observed in the fused lasso studies (Tibshirani *et al.* [27]) where the ordering of the covariates is available.

The combined data-augmented weighted l_1 technique significantly improves the performance of the clustered lasso in finite samples: the test error is reduced effectively, as a result of (nondiagonal) data-augmentation; simultaneously, the proper sparsity is enhanced after introducing weights into the l_1-penalty, by 70% at the minimum. Both improvements are effective regardless of correlation strength and cluster size.

On the other hand, in spite of the encouraging simulation results we noticed that the computational difficulties cannot be underestimated. The interior point methods based semidefinite programming (SDP) solvers (like SeDuMi or SDPT3) cannot even handle the clustered lasso for a moderate value of p. Therefore, we are in great need of a fast algorithm with good scalability to apply the proposed methodology in high dimensions.

4. A fast algorithm for solving the sparse regression

In applying the (improved) clustered lasso to the microarray data, we encounter insurmountable difficulty with all optimization procedures (to date), mainly due to the fact that $\boldsymbol{T}$ has p columns and $O(p^2)$ rows. Our experience shows that it is already extremely difficult or infeasible to carry out the supervised clustering for p just greater than 110. In this section, we propose a simple but *scalable* algorithm to solve the generic sparsity problem in practical applications. It is motivated by the popular coordinate descent algorithm for computing the lasso solution path.

4.1. Motivation

We start with the lasso which minimizes

$$\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{\beta}\|_1. \tag{4.1}$$

By the KKT optimality conditions [25], $\hat{\boldsymbol{\beta}}$ is an optimal solution if and only if $\hat{\boldsymbol{\beta}}$ satisfies the equation

$$\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y})+\lambda\widetilde{\mathrm{sgn}}(\boldsymbol{\beta})=\boldsymbol{0},\ \text{or}\ \lambda\widetilde{\mathrm{sgn}}(\boldsymbol{\beta})=\boldsymbol{X}^T\boldsymbol{y}-\boldsymbol{\Sigma}\boldsymbol{\beta}, \tag{4.2}$$

where the same generalized sign notation is used to denote a subgradient of $\|\boldsymbol{\beta}\|_1$. There is a simple but important fact about $\widetilde{\mathrm{sgn}}$: *Given an arbitrary* $\widetilde{sgn}(\boldsymbol{\beta})\in\widetilde{Sgn}(\boldsymbol{\beta})$, *let* $\boldsymbol{\xi}=\boldsymbol{\beta}+\lambda\widetilde{sgn}(\boldsymbol{\beta})$, *then*

$$\boldsymbol{\beta}=\Theta_S(\boldsymbol{\xi};\lambda),$$

where $\Theta_S(\cdot;\lambda)$ *(or* $\Theta(\cdot;\lambda)$, *for simplicity) is the soft-thresholding operator using* λ *as the threshold value.* Rewriting (4.2) as

$$\boldsymbol{\beta}+\lambda\widetilde{\mathrm{sgn}}(\boldsymbol{\beta})=\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{I}-\boldsymbol{\Sigma})\boldsymbol{\beta} \tag{4.3}$$

motivates an iterative design to solve (4.1)

$$\boldsymbol{\xi}^{(j+1)}=\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{I}-\boldsymbol{\Sigma})\boldsymbol{\beta}^{(j)},\quad \boldsymbol{\beta}^{(j+1)}=\Theta(\boldsymbol{\xi}^{(j+1)};\lambda). \tag{4.4}$$

If $\|\boldsymbol{\Sigma}\|_2<1$, this nonlinear process can be shown to *converge*[1] to an optimal point even if $\boldsymbol{\Sigma}$ is singular (in which case (4.4) is not a contraction, but only a *nonexpansive* mapping) [10].

(4.4) has been proposed in different forms [10, 14, 29] and is strongly advocated for large-data problems. In particular, Daubechies *et al.* [10] proved nice theoretical results on its convergence in a functional framework; Friedman *et al.* [14] demonstrated its amazing performance in terms of the computation

[1] This theoretical achievement is considerably stronger than an 'every accumulation point' argument often seen in numerical analysis [4].

time compared to the homotopy method and the LARS. This iterative algorithm is simple to implement and has very good scalability.

Now we consider the generic sparsity problem

$$\min \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\lambda\|\boldsymbol{T}\boldsymbol{\beta}\|_1, \tag{4.5}$$

where $\boldsymbol{T}$ is a given sparsity pattern matrix that can be specified by users in different situations. In this section, we assume $\boldsymbol{T}$ has full column rank (and thus is a square or 'thin' matrix). The optimal $\hat{\boldsymbol{\beta}}$ satisfies the equation

$$\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\beta}-\boldsymbol{y})+\lambda\boldsymbol{T}^T\widetilde{\operatorname{sgn}}(\boldsymbol{T}\boldsymbol{\beta})=\boldsymbol{0}.$$

Similar to the derivation of (4.4), we can get

$$\boldsymbol{T}^T\cdot\Theta(\boldsymbol{T}\boldsymbol{\beta};\lambda)=\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{T}^T\boldsymbol{T}-\boldsymbol{\Sigma})\boldsymbol{\beta}. \tag{4.6}$$

The difficulty is, however, $\boldsymbol{T}^T$ has no left inverse in the case of a 'thin' $\boldsymbol{T}$. For example, in the fused lasso,

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{I}\\ \lambda_2\boldsymbol{F}\end{bmatrix}\ \text{with}\ \boldsymbol{F}=\begin{bmatrix}1&-1&&&\\&1&-1&&\\&&\cdots&\cdots&\\&&&1&-1\end{bmatrix}.$$

$\boldsymbol{T}$ does not have any right inverse. Accordingly, it is difficult to develop a proper iterative algorithm based on (4.6).

This poses a great challenge in generalizing the coordinate optimization from the lasso to the fused lasso or the clustered lasso. In [14], introducing descent cycles, fusion cycles, and smoothing cycles, Friedman *et al.* gave an ad-hoc design for solving the *diagonal* fused lasso. There is no guarantee that the procedure converges or provides a solution to the original problem.

We reparameterize (4.5) by introducing $\boldsymbol{H}$ satisfying $\boldsymbol{H}\boldsymbol{T}=\boldsymbol{I}$. Assuming that the SVD decomposition of $\boldsymbol{T}$ is given by $\boldsymbol{T}=\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, we take $\boldsymbol{H}=\boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^T$ throughout this section. The generic sparsity regression problem (4.5) is equivalent to the following *constrained* lasso problem:

$$\min f(\boldsymbol{\gamma})\triangleq\frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{H}\cdot\boldsymbol{\gamma}\|_2^2+\lambda\|\boldsymbol{\gamma}\|_1\quad\text{s.t. }\boldsymbol{T}\boldsymbol{H}\boldsymbol{\gamma}=\boldsymbol{\gamma}. \tag{4.7}$$

This suggests a simple iterative way to solve (4.5)

$$\begin{cases}\boldsymbol{\gamma}^{(j)}=\Theta(\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{I}-\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H})\boldsymbol{\gamma}^{(j-1)};\lambda)\\ \boldsymbol{\gamma}^{(j+1)}=\boldsymbol{T}\boldsymbol{H}\boldsymbol{\gamma}^{(j)}\end{cases} \tag{4.8}$$

and

$$\boldsymbol{\beta}^{(j)}=\boldsymbol{H}\boldsymbol{\gamma}^{(j)}. \tag{4.9}$$

We can prove that the sequence of iterates defined by (4.8) must converge under some mild conditions. Yet a further challenge is that it does not converge to the right solution.

4.2. The 'annealing' algorithm

Observe that the original optimization problem (4.5) is also equivalent to

$$\min \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}\|_2^2+\frac{\lambda}{k}\|(\boldsymbol{T}\cdot k)\boldsymbol{\beta}\|_1.$$

for *any* k positive. We get a variant of (4.8) and (4.9)

$$\begin{cases}\tilde{\boldsymbol{\gamma}}^{(j)}=\Theta\left(\frac{1}{k}\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{I}-\frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H})\tilde{\boldsymbol{\gamma}}^{(j-1)};\frac{\lambda}{k}\right)\\ \tilde{\boldsymbol{\gamma}}^{(j+1)}=\boldsymbol{T}\boldsymbol{H}\tilde{\boldsymbol{\gamma}}^{(j)},\end{cases}\tag{4.10}$$

$$\boldsymbol{\beta}^{(j)}=\boldsymbol{H}\tilde{\boldsymbol{\gamma}}^{(j)}/k.\tag{4.11}$$

Let $\boldsymbol{\gamma}^{(j)}=\tilde{\boldsymbol{\gamma}}^{(j)}/k$. Since

$$\tilde{\boldsymbol{\gamma}}^{(j)}+\frac{\lambda}{k}\widetilde{\mathrm{sgn}}(\tilde{\boldsymbol{\gamma}}^{(j)})=k\boldsymbol{\gamma}^{(j)}+\frac{\lambda}{k}\widetilde{\mathrm{sgn}}(\boldsymbol{\gamma}^{(j)}\cdot k)=k\boldsymbol{\gamma}^{(j)}+\frac{\lambda}{k}\widetilde{\mathrm{sgn}}(\boldsymbol{\gamma}^{(j)}),$$

we obtain

$$\begin{cases}\boldsymbol{\gamma}^{(j)}=\Theta\left(\frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{I}-\frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H})\boldsymbol{\gamma}^{(j-1)};\frac{\lambda}{k^2}\right)\\ \boldsymbol{\gamma}^{(j+1)}=\boldsymbol{T}\boldsymbol{H}\boldsymbol{\gamma}^{(j)},\end{cases}\tag{4.12}$$

$$\boldsymbol{\beta}^{(j)}=\boldsymbol{H}\boldsymbol{\gamma}^{(j)}.\tag{4.13}$$

For clarity, we write (4.12) as even and odd updates

$$\begin{cases}\boldsymbol{\gamma}_e^{(j)}=\Theta\left(\frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y}+(\boldsymbol{I}-\frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H})\boldsymbol{\gamma}_o^{(j-1)};\frac{\lambda}{k^2}\right),\\ \boldsymbol{\gamma}_o^{(j)}=\boldsymbol{T}\boldsymbol{H}\boldsymbol{\gamma}_e^{(j)}.\end{cases}\tag{4.14}$$

Theorem 4.1. *The following results hold for the sequence of iterates defined by (4.14):*

1. *Convergence. There exists a $k_0>0$ such that for any $k>k_0$, $\boldsymbol{\gamma}_e^{(j)},\boldsymbol{\gamma}_o^{(j)},\boldsymbol{\beta}^{(j)}$ converge given any initial value in (4.14). That is, as $j\to\infty$, we have*

$$\boldsymbol{\gamma}_e^{(j)}(k)\to\boldsymbol{\gamma}_e(k),\boldsymbol{\gamma}_o^{(j)}(k)\to\boldsymbol{\gamma}_o(k),\boldsymbol{\beta}^{(j)}(k)\to\boldsymbol{\beta}(k).$$

2. *Optimality. As $k\to\infty$, every limit point of $\boldsymbol{\beta}(k)$ (or $\boldsymbol{\gamma}_e(k)$, $\boldsymbol{\gamma}_o(k)$) is an optimal solution to (4.5) (or (4.7)).*
3. *Rate. Let $\Delta(k)=\boldsymbol{\gamma}_e(k)-\boldsymbol{\gamma}_o(k)$, f_{opt} be the optimal value in (4.7). Then,*[2]

$$\|\Delta(k)\|\le\frac{C}{k^2},\tag{4.15}$$

and

$$0\le f_{opt}-f(\boldsymbol{\gamma}_e(k))\le\frac{C}{k^2}.\tag{4.16}$$

[2]In this paper, we use C to denote a positive constant. These C's may not take the same value.

4. k_0*. Finally,*

$$k_0 \le \frac{1}{\sqrt{2}} \frac{\sigma_{\max}(\boldsymbol{X})}{\sigma_{\min}(\boldsymbol{T})}, \tag{4.17}$$

where $\sigma_{\max}(\sigma_{\min})$ *denotes the largest (smallest) singular value of the corresponding matrix.*

See Appendix C for the details of the proof by use of a generalization of Daubechies *et al.*'s convergence theorem [10]. In the following, we abbreviate the subscripts of $\boldsymbol{\gamma}_e^{(j)}(k)$ and $\boldsymbol{\gamma}_e(k)$ for simplicity. We summarize more findings in the case that $\boldsymbol{\Sigma}$ is nonsingular:

Proposition 4.1. *Suppose* $\boldsymbol{\Sigma}$ *is nonsingular. Then*

$$\boldsymbol{\gamma}_{opt} \triangleq \arg\left(\min_{\boldsymbol{\gamma}} f(\boldsymbol{\gamma}) \ s.t. \ \boldsymbol{T}\boldsymbol{H}\boldsymbol{\gamma} = \boldsymbol{\gamma}\right)$$

is unique. On the convergence of $\boldsymbol{\gamma}^{(j)}(k)$ $(k > k_0)$*, we have*

$$\|\boldsymbol{\gamma}^{(j)}(k) - \boldsymbol{\gamma}(k)\| \le \left(1 - \frac{\rho_0}{k^2}\right)^j \|\boldsymbol{\gamma}^{(0)}(k) - \boldsymbol{\gamma}(k)\|, \tag{4.18}$$

where $\rho_0 = \lambda_{\min}^{+}(\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H})$*, the smallest positive eigenvalue of* $\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H}$*; and*

$$\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}_{opt}\| \le \frac{C}{k^2}. \tag{4.19}$$

Moreover, sign consistency can be achieved by finite k*. Specifically,*

$$(\boldsymbol{\gamma}(k))_z = \boldsymbol{0},$$

for any k *large enough, where the index set* z *satisfies* $(\boldsymbol{\gamma}_{opt})_z = \boldsymbol{0}$*.*

From (4.18), with $\delta \triangleq \|\boldsymbol{\gamma}^{(0)}(k) - \boldsymbol{\gamma}(k)\|$, we have $\|\boldsymbol{\gamma}^{(j)}(k) - \boldsymbol{\gamma}(k)\| \le \epsilon_0$ if $\left(1 - \frac{\rho_0}{k^2}\right)^j \delta \le \epsilon_0$ or $j \le \frac{\log(\delta/\epsilon_0)}{\log(1-\rho_0/k^2)} \approx k^2 \cdot \frac{1}{\rho_0}\log(\delta/\epsilon_0)$, which indicates that the number of iterations required at k is $O(k^2)$. On the other hand, from (4.15) or (4.19), the error is of order $1/k^2$. In general, for a small value of k, $\boldsymbol{\beta}^{(j)}(k)$ converges fast but to an inaccurate solution, while when k gets larger, $\boldsymbol{\beta}^{(j)}(k)$ converges more slowly but to a more accurate point.

We can adopt an 'annealing' design (not the simulated annealing) with k acting as the inverse temperature parameter. Run (4.14) for some k till convergence, then use the estimate as the initial point to move on to a new iteration associated with a larger k. The outline for our annealing algorithm is given as follows. The details of the design are given in the next subsection.

1. Initialization. Set the starting values of $\boldsymbol{\gamma}_o^{(0)}$, k, etc.
2. Iteration.
   - Update $\boldsymbol{\gamma}_e^{(j)}$, $\boldsymbol{\gamma}_o^{(j)}$, and $\boldsymbol{\beta}^{(j)}$ according to (4.14) and (4.13).
   - If $\|\boldsymbol{\beta}^{(j)} - \boldsymbol{\beta}^{(j-1)}\|$ is small, then

  - If $\|\boldsymbol{\gamma}_e^{(j)} - \boldsymbol{\gamma}_o^{(j)}\|$ is small enough, exit.
  - Otherwise, increase k to a larger value.
- Let $j \leftarrow j+1$; go to the next iteration.

Both the inner j-convergence and the outer k-convergence can be slow. The convergence rates may not be geometric (see, e.g., (4.19)) caused by the non-expansive operators. Some effective techniques are needed to boost the convergences.

4.3. Accelerated annealing

It is natural to think of updating k at each iteration j. In this *inhomogeneous* updating, the 'cooling schedule', i.e., the growing manner of $k(j)$, is crucial to guarantee an optimal convergent point that solves (4.7).

Theorem 4.2. *Assume* $\boldsymbol{\Sigma}$ *is nonsingular. If* $k(j)$ *satisfies*

$$\sum_{j=1}^{\infty} \frac{1}{k^2(j)} = \infty, \text{ and } k(j) \to \infty \text{ as } j \to \infty, \tag{4.20}$$

then the inhomogeneous chain must converge to the optimal solution.

We can take $k(j) = \sqrt{j}$ for instance. A detailed proof of Theorem 4.2 is provided in Appendix C, based on a useful decomposition for inhomogeneous chains due to Wrinkler [28]. In theory, a valid cooling schedule should be no faster than the $k(j)$ satisfying (4.20). Theorem 4.2 also implies that it essentially takes polynomial time to yield a good solution, in contrast to the exponential time in the simulated annealing [20]. But $\sqrt{j}$ might still be too slow in practice. In most applications, we are only interested in obtaining a good enough solution, thereby allowing for an even faster cooling schedule. Empirically, we recommend the *stagewise homogenous* updating – run a sequence of homogenous chains, each at a fixed k. The trick is to run the chains for small values of k first to complete the major improvements over the initial point, but not till convergence since $\boldsymbol{\gamma}(k)$ may not be close to $\boldsymbol{\gamma}_{opt}$; the fine adjustments are left to large-k iterations. An illustration of the cooling schedule is given in Figure 3. In implementation, k is doubled once a certain stopping criterion is met. We find the following type of relative error

$$\frac{\|\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}\|}{\|\Delta^{(j)}(k)\|}$$

makes a good stopping criterion, where $\Delta^{(j)}(k) \triangleq \boldsymbol{\gamma}_e^{(j)}(k) - \boldsymbol{\gamma}_o^{(j)}(k)$. Due to (4.15), we may also use

$$k^2 \cdot \|\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}\|. \tag{4.21}$$

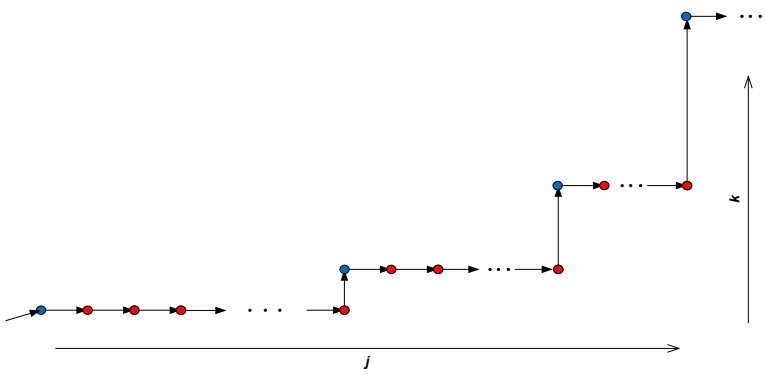

FIG 3. *Stagewise homogenous updating in AA.*

Accelerating the inner j-convergence is even trickier because the iteration here is nonlinear and nonsmooth. Introduce $\boldsymbol{K}$ satisfying

$$\boldsymbol{K}^T\boldsymbol{K} = \frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H} + \boldsymbol{U}_\perp\boldsymbol{U}_\perp^T,$$

where $\boldsymbol{U}_\perp$ is the orthogonal complement of $\boldsymbol{U}$, and let $\boldsymbol{\alpha} = \boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y}/k^2$. We represent the updating kernel (4.14) as (see Appendix C)

$$\boldsymbol{\xi}^{(j+1)} = (\boldsymbol{I} - \boldsymbol{K}^T\boldsymbol{K})\boldsymbol{\gamma}^{(j)} + \boldsymbol{\alpha}, \boldsymbol{\gamma}^{(j+1)} = \Theta(\boldsymbol{\xi}^{(j+1)}; \frac{\lambda}{k^2}). \tag{4.22}$$

We consider two forms of relaxation parameterized by ω for the above nonlinear process:

$$\text{(I)} \qquad \boldsymbol{\xi}^{(j+1)} = (1-\omega)\boldsymbol{\xi}^{(j)} + \omega((\boldsymbol{I} - \boldsymbol{K}^T\boldsymbol{K})\boldsymbol{\gamma}^{(j)} + \boldsymbol{\alpha}), \boldsymbol{\gamma}^{(j+1)} = \Theta(\boldsymbol{\xi}^{(j+1)}; \frac{\lambda}{k^2}), \tag{4.23}$$

$$\text{(II)} \quad \boldsymbol{\xi}^{(j+1)} = (1-\omega)\boldsymbol{\gamma}^{(j)} + \omega((\boldsymbol{I} - \boldsymbol{K}^T\boldsymbol{K})\boldsymbol{\gamma}^{(j)} + \boldsymbol{\alpha}), \boldsymbol{\gamma}^{(j+1)} = \Theta(\boldsymbol{\xi}^{(j+1)}; \omega\cdot\frac{\lambda}{k^2}). \tag{4.24}$$

Both relaxations seem to converge in practice and yield an optimal solution when $0 < \omega < 2$. When $\omega = 1$, they degenerate to the nonrelaxation form (4.22). Before proceeding, we introduce some more operators $T_k, \Theta_k, \widetilde{T}_k, \overline{T}_k$: for any vector $\boldsymbol{v}$, $T_k \circ \boldsymbol{v} = \boldsymbol{J}\boldsymbol{v} + \boldsymbol{\alpha}$, $\forall \boldsymbol{v}$, with $\boldsymbol{J} = \boldsymbol{I} - \omega\boldsymbol{K}^T\boldsymbol{K}$, $\Theta_k \circ \boldsymbol{v} = \Theta(\boldsymbol{v}; \lambda/k^2)$; $\widetilde{T}_k = \Theta_k \circ T_k$, $\overline{T}_k = T_k \circ \Theta_k$.

Proposition 4.2. *For Relaxation (II), given any $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{\gamma}^{(j)}(k)$ converges to a fixed point of $\widetilde{T}_k$ as $j \to \infty$, provided $0 < \omega < 2$. All conclusions in Theorem 4.1 hold under this condition, except that the last statement becomes*

$$k_0 \le \sqrt{\frac{\omega}{2}}\frac{\sigma_{\max}(\boldsymbol{X})}{\sigma_{\min}(\boldsymbol{T})}. \tag{4.25}$$

The cooling schedule Theorem (Theorem 4.2) also applies for $0 < \omega < 2$.

Convergence analysis is more difficult for Relaxation (I). Currently, we have obtained the following result.

Proposition 4.3. *For Relaxation (I), given any* $\boldsymbol{\gamma}^{(0)}$, $\boldsymbol{\gamma}^{(j)}(k)$ *converges to a fixed point of* $\widetilde{T}_k$ *as* $j \to \infty$, *provided* $0 < \omega \leq 1$. *If* $2\overline{T}_k - I$ *is nonexpansive, the same conclusion is true for* $1 < \omega < 2$.

The proof presented in Appendix C is motivated by Browder and Petryshyn's reasonable wanderer [7]. We claim that $\boldsymbol{\gamma}^{(j)}$ converges for $0 < \omega \leq 2$ (based on extensive experience). Relaxation (I) with $\omega = 2$ is of particular interest: in this situation, $\boldsymbol{\xi}^{(j)}$ does not converge (possessing two accumulation points), but $\boldsymbol{\gamma}^{(j)}$ converges and the limit depends on $\boldsymbol{U}_\perp \boldsymbol{U}_\perp^T \boldsymbol{\xi}^{(0)}$ — if $\boldsymbol{U}_\perp \boldsymbol{U}_\perp^T \boldsymbol{\xi}^{(0)} = \boldsymbol{0}$, this limit is an optimal solution, or a fixed point of $\widetilde{T}_k$. For an inaccurate initial point, the relaxation with $\omega = 2$ can reduce the number of iterations by 40% or so in comparison to $\omega = 1$.

We now state the full procedure for the accelerated annealing (AA) algorithm. Suppose $\boldsymbol{X}$, $\boldsymbol{y}$, λ, $\boldsymbol{T}(= \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T)$, and $\boldsymbol{H}$ are known. In the initialization stage, we set a starting value of $\boldsymbol{\beta}^{(cur)}$ and construct $\boldsymbol{\gamma}^{(e)}$. The initial k is given by (4.25). The iteration (starting with $j = 0$) is specified below with ε_{outer}, $\varepsilon_{inner,a}$, $\varepsilon_{inner,b}$ as prescribed error tolerances.

AA Iteration

- $\boldsymbol{\xi}^{(new)} \leftarrow (\boldsymbol{I} - \frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H})\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma}^{(e)} + \frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y}$.
- `If` $j > 0$, $\boldsymbol{\xi}^{(new)} \leftarrow (1-\omega)\boldsymbol{\xi}^{(cur)} + \omega\boldsymbol{\xi}^{(new)}$.
- $\boldsymbol{\gamma}^{(e)} \leftarrow \Theta(\boldsymbol{\xi}^{(new)}; \frac{\lambda}{k^2})$.
- $\boldsymbol{\beta}^{(new)} \leftarrow \boldsymbol{H}\boldsymbol{\gamma}^{(e)}, \boldsymbol{\gamma}^{(o)} \leftarrow \boldsymbol{T}\boldsymbol{\beta}^{(new)}$.
- `If` $\|\boldsymbol{\beta}^{(cur)} - \boldsymbol{\beta}^{(new)}\|_\infty < \max\left(\varepsilon_{inner,a}/k^2, \varepsilon_{inner,b}\right)$
  - `If` $\|\boldsymbol{\gamma}^{(o)} - \boldsymbol{\gamma}^{(e)}\|_\infty < \varepsilon_{outer}$, `exit.`
  - `Otherwise let` $k \leftarrow 2k$, $j \leftarrow 0$.
- $\boldsymbol{\beta}^{(cur)} \leftarrow \boldsymbol{\beta}^{(new)}$, $\boldsymbol{\xi}^{(cur)} \leftarrow \boldsymbol{\xi}^{(new)}$.
- $j \leftarrow j + 1$; `go to the next iteration.`

This AA algorithm is very simple to implement, and can solve the sparse regression for any $\boldsymbol{T}$. There are no complicated operations such as matrix inversion involved in the iteration. In addition, since the problem is convex, a pathwise algorithm with warm starts is preferred, where the estimate associated with the current value of λ is used as the initial point in AA for the next value of λ. An even more effective trick is to construct the initial estimate via the linear extrapolation of the last two estimates.

The computational cost of the AA algorithm is primarily due to matrix multiplication and thresholding. Although an SVD for $\boldsymbol{T}$ is required, it only needs a one-time calculation. Furthermore, for some regularly patterned sparsity matrix, like the fused lasso and the clustered lasso, we are able to provide explicit analytical solutions.

Let

$$\boldsymbol{T}_1 = \begin{bmatrix} \boldsymbol{I} \\ \lambda \boldsymbol{F}_1 \end{bmatrix} \text{ with } \boldsymbol{F}_1 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \cdots & \cdots & \\ & & & 1 & -1 \end{bmatrix} \tag{4.26}$$

be the sparsity matrix for the fused lasso, and let

$$\boldsymbol{T}_2 = \begin{bmatrix} \boldsymbol{I} \\ \lambda \boldsymbol{F}_2 \end{bmatrix} \tag{4.27}$$

denote the sparsity matrix for the clustered lasso, where $\boldsymbol{F}_2$ is a pairwise difference matrix that can be defined by

$$\boldsymbol{F}_2(i,j) = \begin{cases} 1, & \text{if } j = \alpha_i \\ -1, & \text{if } j = \beta_i \\ 0, & \text{otherise} \end{cases} \tag{4.28}$$

for $i = 1, \cdots, d(d-1)/2$, with $\{(\alpha_i, \beta_i)\}$ enumerating all possible pairwise combinations of $\{1, 2, \cdots, d\}$. Without loss of generality, assume $\alpha_{d(d-1)/2-2} = d-2$, $\beta_{d(d-1)/2-2} = d-1$, $\alpha_{d(d-1)/2-1} = d-2$, $\beta_{d(d-1)/2-1} = d$, $\alpha_{d(d-1)/2} = d-1$, $\beta_{d(d-1)/2-2} = d$; that is, the bottom right 3-by-3 submatrix of $\boldsymbol{F}_2$ is $\begin{bmatrix} 1 & -1 & \\ 1 & & -1 \\ & 1 & -1 \end{bmatrix}$.

Proposition 4.4. *The following formulas provide the SVDs for the fused lasso and the clustered lasso, with* $\boldsymbol{F}_1 = \boldsymbol{U}_1 \boldsymbol{D}_1 \boldsymbol{V}_1^T$, $\boldsymbol{T}_1 = \tilde{\boldsymbol{U}}_1 \tilde{\boldsymbol{D}}_1 \tilde{\boldsymbol{V}}_1^T$, $\boldsymbol{F}_2 = \boldsymbol{U}_2 \boldsymbol{D}_2 \boldsymbol{V}_2^T$, *and* $\boldsymbol{T}_2 = \tilde{\boldsymbol{U}}_2 \tilde{\boldsymbol{D}}_2 \tilde{\boldsymbol{V}}_2^T$:

1. $\boldsymbol{U}_1 = \sqrt{\frac{2}{d}} \left[\sin\left(\frac{ij\pi}{n}\right)\right]_{(d-1)\times(d-1)}$, $\boldsymbol{D}_1 = diag\{2\sin\left(\frac{i\pi}{2d}\right)\}_{(d-1)\times(d-1)}$, $\boldsymbol{V}_1 = \sqrt{\frac{2}{d}} \left[\cos\left(\frac{(2i-1)j\pi}{2n}\right)\right]_{d\times(d-1)}$.
2. $\tilde{\boldsymbol{U}}_1 = \begin{bmatrix} \frac{1}{\sqrt{d}}\boldsymbol{1}_{d\times 1} & \boldsymbol{V}_1(\boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2)^{-\frac{1}{2}} \\ \boldsymbol{0}_{(d-1)\times 1} & \boldsymbol{U}_1 \cdot \lambda \boldsymbol{D}_1 (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2)^{-\frac{1}{2}} \end{bmatrix}$, $\tilde{\boldsymbol{D}}_1 = \begin{bmatrix} 1 & \\ & (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2)^{\frac{1}{2}} \end{bmatrix}$, $\tilde{\boldsymbol{V}}_1 = \begin{bmatrix} \frac{1}{\sqrt{d}}\boldsymbol{1}_{d\times 1} & \boldsymbol{V}_1 \end{bmatrix}$.
3. $\boldsymbol{U}_2 = \begin{bmatrix} \boldsymbol{u}_{21} & \frac{1}{\sqrt{d}}\boldsymbol{F}_2 \boldsymbol{V}_1 \end{bmatrix}$, $\boldsymbol{D}_2 = diag\{0, \sqrt{d}, \cdots, \sqrt{d}\}$, $\boldsymbol{V}_2 = \tilde{\boldsymbol{V}}_1$, $\forall d \geq 3$, *where* $\boldsymbol{u}_{21} = \frac{1}{\sqrt{3}} \begin{bmatrix} 0 & \cdots & 0 & 1 & -1 & 1 \end{bmatrix}^T$.
4. $\tilde{\boldsymbol{U}}_2 = \begin{bmatrix} \frac{1}{\sqrt{d}}\boldsymbol{1}_{d\times 1} & \frac{1}{\sqrt{1+\lambda^2 d}}\boldsymbol{V}_1 \\ \boldsymbol{0} & \frac{\lambda}{\sqrt{1+\lambda^2 d}}\boldsymbol{F}_2 \boldsymbol{V}_1 \end{bmatrix}$, $\tilde{\boldsymbol{D}}_2 = diag\{1, \sqrt{1+\lambda^2 d}, \cdots, \sqrt{1+\lambda^2 d}\}$, $\tilde{\boldsymbol{V}}_2 = \boldsymbol{V}_2 = \tilde{\boldsymbol{V}}_1$.

To apply Proposition 4.4 to the DAW-CLASSO (3.17), we need to generalize our algorithms and results to the weighted version of (4.5)

$$\min \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \|\boldsymbol{\Lambda}\boldsymbol{T}\boldsymbol{\beta}\|_1,$$

where $\mathbf{\Lambda} = \text{diag}\{\lambda_i\}$ with $\lambda_i > 0$. This is trivial though: replacing the universal threshold λ by the componentwise thresholds λ_i, we find all conclusions and proofs carry over.

4.4. Practical performance

As previously mentioned, the task of supervised clustering is quite challenging in computation. Since the clustered lasso is a convex optimization problem, the general-purpose solvers such as SeDuMi and SDPT3 can be applied. These solvers are usually based on interior point (IP) methods. Empirically, we find that they are more appropriate for small scale problems and are more accurate than the proposed accelerated annealing algorithm if feasible. The SDP solvers typically fail when p is much greater than 100, due to high computational complexity and memory requirements. The size of the sparsity matrix $\boldsymbol{T}$ or its left inverse $\boldsymbol{H}$ can be huge ($O(p^3)$) even for a medium value of p. For the kidney microarray data described in Section 2, although we can reduce the problem size by gene filtering — for example, FDR < 0.05 yields about 800 genes — we could only manage to run the clustered lasso with general-purpose convex optimization softwares for p less than 110. These seriously restrict the use of the clustered lasso in real-world applications.

By contrast, the AA iterations only involve low-complexity operations like matrix-vector multiplications and componentwise thresholdings, which provides good algorithm scalability. Moreover, statisticians usually have the need to compute the whole solution path to tune the regularization parameters, and so warms starts (or our *extrapolated* warm starts) are particularly effective to speed the computation (due to the convexity of the problem). This, however, does not apply to the SDP solvers which compute the solution path as a series of independent optimization problems. Furthermore, there is no need to compute or store $\boldsymbol{H}$ in AA seen from Proposition 4.4. In fact, $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ are the only dense matrices needed in calculating all matrix-vector multiplications, and they are of order $p \times p$. This reduces the storage needs to $O(p^2)$. We would also like to point out that the cooling schedule can be used to provide a speed-accuracy tradeoff. In a limited time situation, one may use a faster cooling scheme to obtain a greedy solution. Though not necessarily optimal, it may serve the needs of some applications.

Finally, we present the results of the AA-implemented DAW-CLASSO on the kidney data. Supervised clustering was applied to the 800 most correlated genes after FDR filtering. Five-fold cross-validation was used to tune the parameters. Figure 4 demonstrates the results. The coefficient estimates are successfully clustered and gene groups are formed directly due to the exact-clustering effect. It seems that some of them might be tricky to be found by a two-stage procedure (modeling $\rightarrow$ clustering, see Section 2) based on studying the differences between the coefficient estimates only. In addition, different than many *ad hoc* clustering procedures, the supervised clustering optimizes the clusters during the model fitting process and automatically selects the number of clusters and cluster sizes.

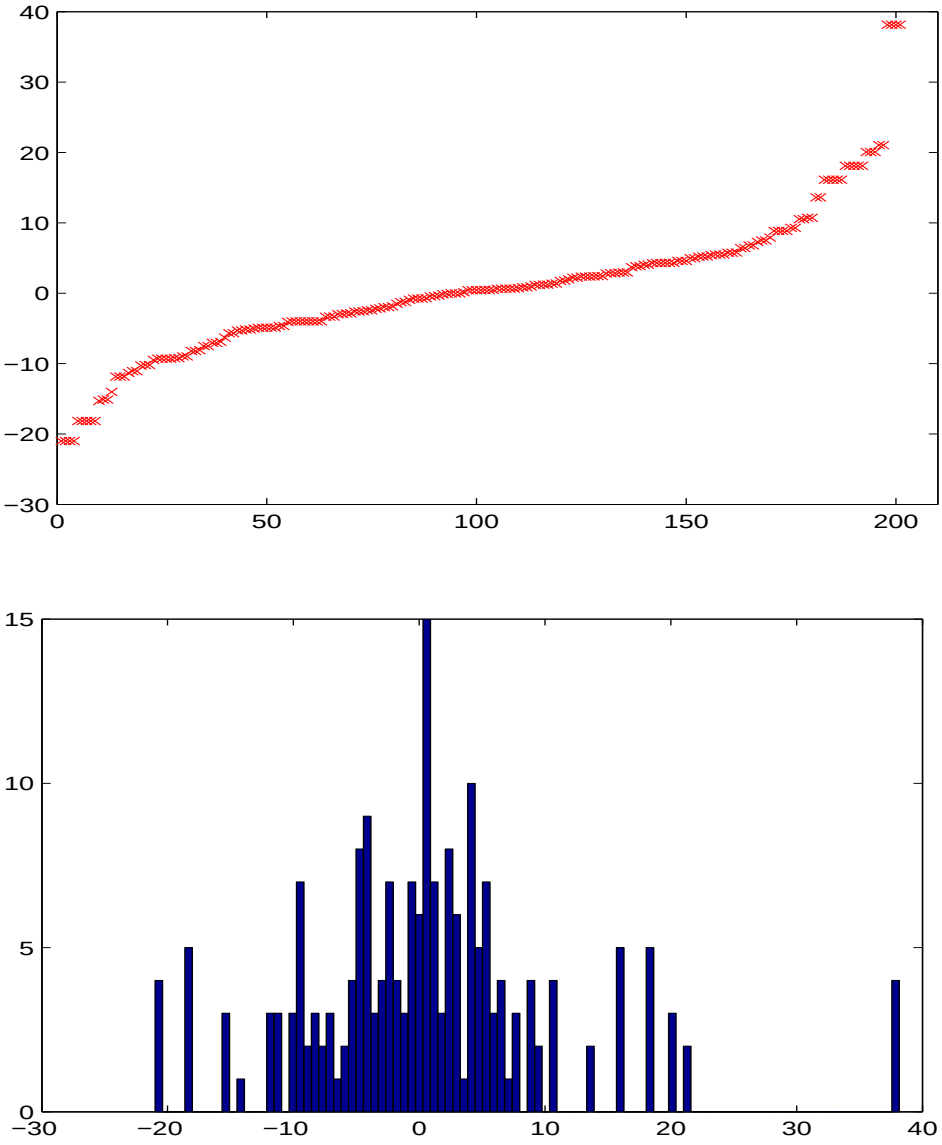

FIG 4. *Clustering results on the kidney data. Only the nonzero clusters are shown. The upper panel plots the nonzero coefficient estimates (reordered), and the lower panel is a histogram of all nonzero clusters. The dominant zero cluster (consisting of about 600 genes) is not shown for a better view of overall clustering results.*

5. Summary

We studied a generic sparse regression problem with a customizable sparsity pattern matrix $\boldsymbol{T}$, motivated by a supervised gene clustering problem. Interestingly, we found both in practice and in theory that the granted power of the l_1-penalty to approximate the l_0-penalty can be rather poor (even for large samples), say, when $\boldsymbol{T}_{nz}$ is large and $(\boldsymbol{T}_z, \boldsymbol{T}_{nz})$ is not 'separable' (see Theorem 3.1). This causes serious trouble for the clustered lasso to achieve exact-clustering.

We further proposed using data-augmentation and weights to reduce the test error and increase the model parsimony simultaneously. From an empirical Bayesian point of view, our nondiagonal DA amounts to a degenerate multivariate Gaussian prior, where one degree of freedom is saved in the covariance matrix construction to better accommodate a less accurate initial estimate. Regarding the weighting technique, Theorem 3.2 generalizes the asymptotic lasso results [34, 36] and provides a broad condition for weight construction. To the best of our knowledge, there are no nonasymptotic results available even in the lasso setting. (For adaptive weights used in orthogonal models, we refer to Zou [34] and She [24] (which also gave a correction to [34]) for some oracle results.) Hence a finite-sample theoretical analysis is an important topic of future research. Another different idea to achieve the same goal of joint accuracy and parsimony in finite samples is to use *nonconvex* penalizations [24]. The computational cost can be even more expensive. On the other hand, we can show (proof omitted) that substituting an appropriate thresholding operator (such as hard-thresholding or hybrid hard-ridge thresholding) for Θ, the accelerated annealing still applies for nonconvex penalized clustering models.

Finally, the scalable AA algorithm also raises some interesting open problems, such as the analysis of relaxation (I) and the rate of convergence studies. These problems are nontrivial in numerical analysis due to the nonexpansive nature and nonlinearity of the underlying operators.

Acknowledgements

The author is grateful to the anonymous referee and the associate editor for their careful comments and useful suggestions to improve the presentation of the paper. Most of this paper is based on [23], supported by NSF grant DMS-0604939. The author would like to thank Art Owen for his valuable guidance and for his help in revising an early version of this manuscript. The author also greatly appreciates valuable discussions with Bala Rajaratnam on the subject of this paper.

Appendix A: Proofs of Proposition 3.1, Proposition 3.2, Theorem 3.1, and Theorem 3.2

For the optimization problem

$$\min \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{T}\boldsymbol{\beta}\|_1,$$

by the KKT optimality conditions [25], $\hat{\boldsymbol{\beta}}$ is an optimal solution if and only if there exists a $\widetilde{\operatorname{sgn}}(\boldsymbol{T}\hat{\boldsymbol{\beta}})$ such that

$$\boldsymbol{X}^T(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{y})+\lambda\boldsymbol{T}^T\widetilde{\operatorname{sgn}}(\boldsymbol{T}\hat{\boldsymbol{\beta}})=\boldsymbol{0}. \tag{A.1}$$

Equivalently,

$$\hat{\boldsymbol{\beta}}=\frac{1}{n}\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}^T\boldsymbol{y}-\lambda\boldsymbol{T}^T\widetilde{\operatorname{sgn}}(\boldsymbol{T}\hat{\boldsymbol{\beta}})),$$

or

$$\hat{\boldsymbol{\beta}}=\boldsymbol{\beta}+\frac{1}{n}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon}-\frac{\lambda}{n}\boldsymbol{\Sigma}^{-1}\boldsymbol{T}^T\widetilde{\operatorname{sgn}}(\boldsymbol{T}\hat{\boldsymbol{\beta}})). \tag{A.2}$$

• **Proof of Proposition 3.1**

The proof is obvious by noticing that

$$\frac{1}{n}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon}\sim N(\boldsymbol{0},\frac{\sigma^2}{n}\boldsymbol{\Sigma}^{-1})=O_p(\frac{1}{\sqrt{n}})=o_p(1)$$

and

$$\frac{\lambda}{n}\boldsymbol{\Sigma}^{-1}\boldsymbol{T}^T\widetilde{\operatorname{sgn}}(\boldsymbol{T}\hat{\boldsymbol{\beta}}))=\frac{\lambda}{n}O_p(1)=o_p(1).$$

□

• **Proof of Proposition 3.2**

Assume for the moment

$$\lambda/\sqrt{n}\to\lambda_0\geq 0. \tag{A.3}$$

We first develop a $\sqrt{n}$-consistent result similar to Knight and Fu [18] but in a general situation:

Lemma A.1. *Under the assumptions in the Proposition 3.2 and (A.3),* $\sqrt{n}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\Rightarrow\boldsymbol{\delta}_o$, *where* $\boldsymbol{\delta}_o$ *is defined by*

$$\arg\min_{\boldsymbol{\delta}}\frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{C}\boldsymbol{\delta}-\boldsymbol{r}^T\boldsymbol{\delta}+\lambda_0\left(sgn(\boldsymbol{T}_{nz}\boldsymbol{\beta})^T\boldsymbol{T}_{nz}\boldsymbol{\delta}+\|\boldsymbol{T}_z\boldsymbol{\delta}\|_1\right),$$

with $z=\{i:(\boldsymbol{T}\boldsymbol{\beta})_i=0\}$, $nz=\{i:(\boldsymbol{T}\boldsymbol{\beta})_i\neq 0\}$, *and* $\boldsymbol{r}\sim N(\boldsymbol{0},\sigma^2\boldsymbol{C})$.

In fact, from the KKT equation (A.2), $\hat{\boldsymbol{\delta}}\triangleq\sqrt{n}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})$ satisfies

$$\hat{\boldsymbol{\delta}}=\frac{1}{\sqrt{n}}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon}-\frac{\lambda}{\sqrt{n}}\boldsymbol{\Sigma}^{-1}\boldsymbol{T}^T\widetilde{\operatorname{sgn}}(\frac{1}{\sqrt{n}}\boldsymbol{T}\hat{\boldsymbol{\delta}}+\boldsymbol{T}\boldsymbol{\beta}).$$

So $\hat{\boldsymbol{\delta}}$ solves $\frac{1}{2}\|\frac{1}{\sqrt{n}}\boldsymbol{X}\boldsymbol{\delta}-\boldsymbol{\epsilon}\|_2^2+\lambda\|\frac{1}{\sqrt{n}}\boldsymbol{T}\boldsymbol{\delta}+\boldsymbol{T}\boldsymbol{\beta}\|_1$, or

$$\frac{1}{2}\|\frac{1}{\sqrt{n}}\boldsymbol{X}\boldsymbol{\delta}-\boldsymbol{\epsilon}\|_2^2-\frac{1}{2}\|\boldsymbol{\epsilon}\|_2^2+\lambda\|\frac{1}{\sqrt{n}}\boldsymbol{T}\boldsymbol{\delta}+\boldsymbol{T}\boldsymbol{\beta}\|_1-\lambda\|\boldsymbol{T}\boldsymbol{\beta}\|_1\triangleq f(\boldsymbol{\delta}).$$

Noticing that $f(\boldsymbol{\delta})\Rightarrow g(\boldsymbol{\delta})$, $\sqrt{n}(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})\Rightarrow\boldsymbol{\delta}_o$ follows by Geyer [16].

We need to show $\limsup_{n\to\infty} P(\boldsymbol{T}_z\hat{\boldsymbol{\beta}} = \boldsymbol{0}) < 1$. Observing $\{\boldsymbol{\beta} : \boldsymbol{T}_z\boldsymbol{\beta} = \boldsymbol{0}\}$ is a closed set, $\limsup_{n\to\infty} P(\boldsymbol{T}_z\hat{\boldsymbol{\beta}} = \boldsymbol{0}) \le P(\boldsymbol{T}_z\boldsymbol{\delta}_o = \boldsymbol{0}) \triangleq p_0$. $\boldsymbol{\delta}_o$ satisfies

$$\boldsymbol{C}\boldsymbol{\delta}_o - \boldsymbol{r} + \lambda_0\boldsymbol{T}_z^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_z\boldsymbol{\delta}_o) + \lambda_0\boldsymbol{T}_{nz}^T\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta}) = \boldsymbol{0}.$$

Clearly, $p_0 < 1$ if $\lambda_0 = 0$. Suppose $\lambda_0 > 0$. $\boldsymbol{T}_z\boldsymbol{\delta}_o = \boldsymbol{0}$ means

$$\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_z^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_z\boldsymbol{\delta}_o) = \frac{1}{\lambda_0}\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{r} - \boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta}),$$

which implies

$$\begin{aligned}&\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_z^T\cdot\boldsymbol{s} = \frac{1}{\lambda_0}\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{r} - \boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T\cdot\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta})\\ &\text{is solvable in the solution space } \{\boldsymbol{s} : \|\boldsymbol{s}\|_\infty \le 1\}.\end{aligned} \tag{A.4}$$

Lemma A.2. *Let $\boldsymbol{A}$ be a positive semi-definite matrix with the spectral decomposition given by, say, $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T = \sum d_i\boldsymbol{u}_i\boldsymbol{u}_i^T$. Define $z' = \{i : d_i = 0\}$, $nz' = \{i : d_i \ne 0\}$, and the generalized inverse $\boldsymbol{A}^+ = \boldsymbol{U}\boldsymbol{D}^+\boldsymbol{U}^T = \boldsymbol{U}_{nz'}\boldsymbol{D}_{nz'}^{-1}\boldsymbol{U}_{nz'}^T$. Then $\boldsymbol{A}\boldsymbol{s} = \boldsymbol{\alpha}$ if and only if (i) $\boldsymbol{s} = \boldsymbol{A}^+\boldsymbol{\alpha} + \boldsymbol{U}_{z'}\boldsymbol{\eta}$ for some $\boldsymbol{\eta}$ and (ii) $\boldsymbol{U}_{z'}^T\boldsymbol{\alpha} = \boldsymbol{0}$.*

The proof is omitted.

Apply Lemma A.2 to the problem of (A.4) with $\boldsymbol{A} = \boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_z^T$, and $\boldsymbol{\alpha} = \frac{1}{\lambda_0}\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{r} - \boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta})$. (Note that condition (ii) is naturally satisfied, because

$$\boldsymbol{U}_{z'}^T\boldsymbol{A} = \boldsymbol{0} \Rightarrow \boldsymbol{U}_{z'}^T\boldsymbol{A}\boldsymbol{U}_{z'} = \boldsymbol{0} \Rightarrow \boldsymbol{U}_{z'}^T\boldsymbol{T}_z\boldsymbol{C}^{-1/2} \Rightarrow \boldsymbol{U}_{z'}^T\boldsymbol{T}_z = \boldsymbol{0},$$

and so $\boldsymbol{U}_{z'}^T\boldsymbol{\alpha} = \boldsymbol{0}$.) Then (A.4) implies $\exists\boldsymbol{\eta}$ s.t. $\|\boldsymbol{A}^+\boldsymbol{\alpha} + \boldsymbol{U}_{z'}\boldsymbol{\eta}\|_\infty \le 1$, or

$$\left\|\begin{bmatrix}\boldsymbol{U}_{z'} & \boldsymbol{U}_{nz'}\end{bmatrix}\begin{bmatrix}\boldsymbol{\eta}\\ \boldsymbol{D}_{nz'}^{-1}\boldsymbol{U}_{nz'}^T\boldsymbol{\alpha}\end{bmatrix}\right\|_\infty \le 1.$$

Observing that $\begin{bmatrix}\boldsymbol{U}_{z'} & \boldsymbol{U}_{nz'}\end{bmatrix}$ is an orthonormal matrix, say, of size m-by-m, we know

$$\|\boldsymbol{D}_{nz'}^{-1}\boldsymbol{U}_{nz'}^T\boldsymbol{\alpha}\|_\infty \le \sqrt{m}.$$

Consequently, given $\boldsymbol{r} \sim N(\boldsymbol{0}, \sigma^2\boldsymbol{I})$,

$$p_0 \le P(\|\boldsymbol{D}_{nz'}^{-1}\boldsymbol{U}_{nz'}^T(\frac{1}{\lambda_0}\boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{r} - \boldsymbol{T}_z\boldsymbol{C}^{-1}\boldsymbol{T}_{nz}^T\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta}))\|_\infty \le \sqrt{m}) < 1.$$

For the general case $\lambda = O(n)$, if $\hat{\boldsymbol{\beta}}$ were zero s-consistent w.r.t. $\boldsymbol{T}_z$ for some sequence $\lambda(n)$, there must exist a subsequence $\lambda(n_k)$ with $\lambda(n_k)/n_k \to \lambda_0$ for some $\lambda_0 \ge 0$, such that $\boldsymbol{\beta}(n_k)$ is zero s-consistent w.r.t $\boldsymbol{T}_z$. This contradicts the above argument. □

• **Proof of Theorem 3.1**

First, it is easy to derive an asymptotic result similar to Lemma A.1:

$$\frac{n}{\lambda}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \Rightarrow \boldsymbol{\delta}_o, \tag{A.5}$$

where $\boldsymbol{\delta}_o$ is nonrandom, defined by

$$\arg\min_{\boldsymbol{\delta}} \frac{1}{2}\boldsymbol{\delta}^T \boldsymbol{C}\boldsymbol{\delta} + (\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta}))^T \boldsymbol{T}_{nz}\boldsymbol{\delta} + \|\boldsymbol{T}_z\boldsymbol{\delta}\|_1.$$

So the KKT equation for $\boldsymbol{\delta}_o$ is

$$\boldsymbol{C}\boldsymbol{\delta}_o + \boldsymbol{T}_z^T \widetilde{\mathrm{sgn}}(\boldsymbol{T}_z\boldsymbol{\delta}_o) + \boldsymbol{T}_{nz}^T \mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta}) = \boldsymbol{0}. \tag{A.6}$$

Recall that $\hat{\boldsymbol{\beta}}$ is an optimal solution if and only if (A.2) holds. Therefore,

$$\begin{aligned}
\boldsymbol{T}_1\hat{\boldsymbol{\beta}} &= \boldsymbol{T}_1\Big(\frac{1}{n}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon}\Big) - \frac{\lambda}{n}\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}\hat{\boldsymbol{\beta}}) \\
&= \boldsymbol{T}_1\Big(\frac{1}{n}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon}\Big) - \frac{\lambda}{n}\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_1\hat{\boldsymbol{\beta}}) - \frac{\lambda}{n}\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_2^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}}).
\end{aligned}$$

It follows that

$$\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_1\hat{\boldsymbol{\beta}}) = -\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_2^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}}) + \frac{\sqrt{n}}{\lambda}\boldsymbol{\delta}' - \frac{n}{\lambda}\boldsymbol{T}_1\hat{\boldsymbol{\beta}}, \tag{A.7}$$

where $\boldsymbol{\delta}' = \boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon}/\sqrt{n} \sim N(\boldsymbol{0}, \boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)$. Now apply Lemma A.2 with

$$\boldsymbol{A} = \boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T, \boldsymbol{\alpha} = -\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_2^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}}) + \frac{\sqrt{n}}{\lambda}\boldsymbol{\delta}' - \frac{n}{\lambda}\boldsymbol{T}_1\hat{\boldsymbol{\beta}}.$$

Again, condition (ii) is naturally satisfied because $\boldsymbol{U}_{z'}\boldsymbol{T}_1 = \boldsymbol{0}$. (A.7) is equivalent to $\widetilde{\mathrm{sgn}}(\boldsymbol{T}_1\hat{\boldsymbol{\beta}}) = (\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha} + \boldsymbol{U}_{z'}\boldsymbol{\eta}$ for some $\boldsymbol{\eta}$. It is important to point out that even the original KKT equation (A.2) does not resolve the ambiguity of $\boldsymbol{\eta}$ (since $\boldsymbol{T}_1^T\boldsymbol{U}_{z'}\boldsymbol{\eta} = \boldsymbol{0}\cdot\boldsymbol{\eta} = \boldsymbol{0}$). Hence a sufficient condition for $\boldsymbol{T}_1\hat{\boldsymbol{\beta}} = \boldsymbol{0}$ is given by

$$\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha}\|_\infty < 1, \tag{A.8}$$

and a necessary condition for $\boldsymbol{T}\hat{\boldsymbol{\beta}} = \boldsymbol{0}$ is $\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha} + \boldsymbol{U}_{z'}\boldsymbol{\eta}\|_\infty \le 1$ for some $\boldsymbol{\eta}$. On the other hand,

$$\begin{aligned}
\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha}\|_\infty &= \|\boldsymbol{U}_{nz'}\boldsymbol{U}_{nz'}^T(\boldsymbol{U}_{nz'}\boldsymbol{D}_{nz'}^{-1}\boldsymbol{U}_{nz'}^T\boldsymbol{\alpha} + \boldsymbol{U}_{z'}\boldsymbol{\eta})\|_\infty \\
&= \|\boldsymbol{U}_{nz'}\boldsymbol{U}_{nz'}^T((\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha} + \boldsymbol{U}_{z'}\boldsymbol{\eta})\|_\infty \\
&\le \|\boldsymbol{U}_{nz'}\boldsymbol{U}_{nz'}^T\|_\infty = \|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)\|_\infty,
\end{aligned}$$

that is,

$$\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha}\|_\infty \le \|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)\|_\infty. \tag{A.9}$$

Now we study the asymptotics.

Necessity. If $\hat{\boldsymbol{\beta}}$ is zero s-consistent w.r.t. $\boldsymbol{T}_1$, then from (A.5), $\boldsymbol{T}_1\boldsymbol{\delta}_o = \mathbf{0}$, and so $\frac{n}{\lambda}\boldsymbol{T}_1\hat{\boldsymbol{\beta}} \xrightarrow{P} 0$. In addition, $\frac{\sqrt{n}}{\lambda}\boldsymbol{\delta}' = o_p(1)$. Hence

$$\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_2^T\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}})\|_\infty \le \|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)\|_\infty + \epsilon$$

with probability tending to 1, for any $\epsilon > 0$. Observe that $\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}})$ is bounded. There exists a subsequence indexed by n_k such that $\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}}_{n_k}) \to \boldsymbol{s}$ with probability 1. By Proposition 3.1, we immediately know $\boldsymbol{s} \in \widetilde{\mathrm{Sgn}}(\boldsymbol{T}_2\boldsymbol{\beta})$. Thus

$$\|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_2^T\boldsymbol{s}\|_\infty \le \|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)\|_\infty + \epsilon$$

with probability 1, for any $\epsilon > 0$. The necessary condition follows.

Sufficiency. Our goal is to show $P(\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha}\|_\infty < 1) \to 1$ given (3.3). Suppose $\liminf_{n\to\infty} P(\|(\boldsymbol{T}_1\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{\alpha}\|_\infty \ge 1) > 0$. First, since $\boldsymbol{T}_1 \subset \boldsymbol{T}_z$, if we write $\boldsymbol{T}_z$ as $\begin{bmatrix} \boldsymbol{T}_1 \\ \boldsymbol{T}_{2z} \end{bmatrix}$ with $\boldsymbol{T}_{2z} \subset \boldsymbol{T}_2$, $\widetilde{\mathrm{sgn}}(\boldsymbol{T}_{2z}\boldsymbol{\delta}_o) \in \widetilde{\mathrm{Sgn}}(\boldsymbol{T}_{2z}\boldsymbol{\beta})$. Repeating the argument for (A.8), we know (3.3) is sufficient to get $\boldsymbol{T}_1\boldsymbol{\delta}_o = \mathbf{0}$ from the KKT equation (A.6).

Likewise, we can find a subsequence indexed by n_k such that $\widetilde{\mathrm{sgn}}(\boldsymbol{T}_2\hat{\boldsymbol{\beta}}_{n_k}) \to \boldsymbol{s} \in \widetilde{\mathrm{Sgn}}(\boldsymbol{T}_2\boldsymbol{\beta})$, $\frac{n}{\lambda}\boldsymbol{\delta}' \to \mathbf{0}$, $\frac{n}{\lambda}\boldsymbol{T}_1\hat{\boldsymbol{\beta}} \to \mathbf{0}$, and $\boldsymbol{\Sigma}_{n_k} \to \boldsymbol{C}$ with probability 1. So we get $P(\|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_2^T\boldsymbol{s}\|_\infty \ge 1) > 0$, i.e.,

$$\|(\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_1^T)^+\boldsymbol{T}_1\boldsymbol{C}^{-1}\boldsymbol{T}_2^T\boldsymbol{s}\|_\infty \ge 1,$$

which contradicts the assumption. □

• **Proof of Theorem 3.2**

Define $\boldsymbol{W} = \mathrm{diag}\{w_i\}$, $\boldsymbol{W}_z = \mathrm{diag}\{w_i\}_{j\in z}$, $\boldsymbol{W}_{nz} = \mathrm{diag}\{w_i\}_{j\in nz}$. Then the weighted sparse regression (3.7) just replaces the $\boldsymbol{T}$ in (3.1) by $\boldsymbol{W}\boldsymbol{T}$. Define $\hat{\boldsymbol{\delta}} = a(n,\lambda)(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})$ for some sequence $a(n,\lambda)$. Similar to the derivation of Lemma A.1, $\hat{\boldsymbol{\delta}}$ solves

$$\min \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{\Sigma}\boldsymbol{\delta} - \frac{a}{\sqrt{n}}\left(\frac{1}{\sqrt{n}}\boldsymbol{x}^T\boldsymbol{\epsilon}\right)^T\boldsymbol{\delta} + \frac{\lambda a}{n}\|\boldsymbol{W}_z\boldsymbol{T}_z\boldsymbol{\delta}\|_1 + \frac{\lambda a}{n}\mathrm{sgn}(\boldsymbol{T}_{nz}\boldsymbol{\beta})^T\boldsymbol{W}_{nz}\boldsymbol{T}_{nz}\boldsymbol{\delta}.$$

Following the lines of [34], one can prove that if (i) $\lim\frac{a}{\sqrt{n}}$ exist (say equal to a_0), (ii) $\frac{n}{\lambda a}A \to 0$, and (iii) $\frac{\lambda a}{n}B \to 0$, then

$$\hat{\boldsymbol{\delta}} = a(n,\lambda)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \Rightarrow \arg\left(\min_{\boldsymbol{\delta}} \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{C}\boldsymbol{\delta} - a_0\boldsymbol{r}^T\boldsymbol{\delta},\ \text{s.t. } \boldsymbol{T}_z\boldsymbol{\delta} = \mathbf{0}\right), \tag{A.10}$$

where $\boldsymbol{r} \sim N(\mathbf{0}, \sigma^2\boldsymbol{C})$. To guarantee that such $a(n,\lambda)$ exists, it is enough to have $\frac{nA}{\lambda} \ll \frac{n}{\lambda B}$ and $\frac{nA}{\lambda} \ll \sqrt{n}$, where $P \ll Q$ means $\lim P/Q \to 0$. That is, if

$$\frac{\sqrt{n}}{\lambda}A(n) \to 0, A(n)B(n) \to 0, \tag{A.11}$$

then $\hat{\boldsymbol{\beta}}$ is $a(n,\lambda)$-consistent, for any a satisfying (i), (ii), & (iii).

On the other hand, substituting $\boldsymbol{W}_z\boldsymbol{T}_z$ for $\boldsymbol{T}_1$ and $\boldsymbol{W}_{nz}\boldsymbol{T}_{nz}$ for $\boldsymbol{T}_2$ in (A.8), we obtain a sufficient condition for $\boldsymbol{T}_z\hat{\boldsymbol{\beta}} = \boldsymbol{0}$:

$$\left\| \boldsymbol{W}_z^{-1}(\boldsymbol{T}_z\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_z^T)^+ \left(-\boldsymbol{T}_z\boldsymbol{\Sigma}^{-1}\boldsymbol{T}_{nz}^T\boldsymbol{W}_{nz}\widetilde{\mathrm{sgn}}(\boldsymbol{T}_{nz}\hat{\boldsymbol{\beta}}) + \frac{\sqrt{n}}{\lambda}\frac{\boldsymbol{T}_z\boldsymbol{\Sigma}^{-1}\boldsymbol{x}^T\boldsymbol{\epsilon}}{\sqrt{n}} - \frac{n}{\lambda a(n,\lambda)}a(n,\lambda)\boldsymbol{T}_z\hat{\boldsymbol{\beta}} \right) \right\|_\infty < 1.$$

Clearly, by (A.10), (A.11), and (ii), this holds with probability tending to 1.

For the special case $a = \sqrt{n}$, it suffices to show λ satisfying $\sqrt{n}A/\lambda \to 0$, $\lambda B/\sqrt{n} \to 0$ exists. $\lambda = \sqrt{nA/B}$ is one possible choice. □

Appendix B: Proof of Theorem 3.3

For any given $\lambda_1, \lambda_2 > 0$, we know that

$$\begin{aligned} & \left\| \begin{bmatrix} \boldsymbol{y} \\ \sqrt{\lambda_2}\boldsymbol{X}^T\boldsymbol{y} \end{bmatrix} - \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{I} \end{bmatrix} \boldsymbol{\beta} \right\|_2^2 + \lambda_1\|\boldsymbol{\beta}_1\|_1 \\ = \; & \boldsymbol{\beta}^T(\boldsymbol{X}^T\boldsymbol{X} + \lambda_2\boldsymbol{I})\boldsymbol{\beta} - 2(\lambda_2+1)\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} + \lambda_1\|\boldsymbol{\beta}\|_1 + \left(\boldsymbol{y}^T\boldsymbol{y} + \lambda_2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{y}\right). \end{aligned}$$

Therefore,

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2} &= \arg\min \boldsymbol{\beta}^T(\boldsymbol{X}^T\boldsymbol{X} + \lambda_2\boldsymbol{I})\boldsymbol{\beta} - 2(\lambda_2+1)\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} + \lambda_1\|\boldsymbol{\beta}\|_1 \\ &= \arg\min \boldsymbol{\beta}^T\left(\frac{\boldsymbol{X}^T\boldsymbol{X} + \lambda_2\boldsymbol{I}}{1+\lambda_2}\right)\boldsymbol{\beta} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} + \frac{\lambda_1}{1+\lambda_2}\|\boldsymbol{\beta}\|_1. \end{aligned}$$

Comparing $\hat{\boldsymbol{\beta}}_{\lambda_1,\lambda_2}$ to the definition of $\hat{\boldsymbol{\beta}}^{(eNet)}_{\lambda_1,\lambda_2}$ yields the conclusion in Theorem 3.3. □

Appendix C: Proofs of Theorem 4.1, Proposition 4.1, Theorem 4.2, Proposition 4.2, Proposition 4.3, and Proposition 4.4

• Some Basic Facts

Before our formal proofs, let's state some basic facts. Recall that $\boldsymbol{T} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T$, $\boldsymbol{H} = \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}^T$, and $\boldsymbol{I} - \boldsymbol{T}\boldsymbol{H} = \boldsymbol{U}_\perp\boldsymbol{U}_\perp$ where $\boldsymbol{U}_\perp$ is via expanding $\boldsymbol{U}$ to get an orthonormal matrix $\tilde{\boldsymbol{U}} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{U}_\perp^T \end{bmatrix}$; C is used to denote a positive constant, but not necessarily the same even in a single formula. The subscripts of $\boldsymbol{\gamma}_e^{(j)}(k)$ and $\boldsymbol{\gamma}_e(k)$ are omitted for short.

From (4.14), $\boldsymbol{\gamma}_e(k)$, or $\boldsymbol{\gamma}(k)$, satisfies

$$\boldsymbol{\gamma}(k) + \frac{\lambda}{k^2}\widetilde{\mathrm{sgn}}(\boldsymbol{\gamma}(k)) = \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma}(k) + \frac{1}{k^2}(\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H}\boldsymbol{\gamma}(k)),$$

i.e.,

$$\begin{aligned}\boldsymbol{\gamma}(k) &= \frac{1}{k}\arg\min\frac{\lambda}{k}\|\boldsymbol{\gamma}\|_1 + \frac{1}{2}\|\boldsymbol{y} - (\boldsymbol{X}\boldsymbol{H}/k)\cdot\boldsymbol{\gamma}\|_2^2 + \frac{1}{2}\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}\|_2^2 \\ &= \arg\min\lambda\|\boldsymbol{\gamma}\|_1 + \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{H}\cdot\boldsymbol{\gamma}\|_2^2 + \frac{k^2}{2}\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}\|_2^2 \qquad \text{(C.1)}\end{aligned}$$

Let

$$f(\boldsymbol{\gamma}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{H}\cdot\boldsymbol{\gamma}\|_2^2 + \lambda\|\boldsymbol{\gamma}\|_1, \qquad \text{(C.2)}$$

$$F_k(\boldsymbol{\gamma}) = f(\boldsymbol{\gamma}) + \frac{k^2}{2}\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}\|_2^2, \qquad \text{(C.3)}$$

$$\Phi_k(\boldsymbol{\gamma}) = \frac{1}{k^2}F_k(\boldsymbol{\gamma}). \qquad \text{(C.4)}$$

Fact 1) For any k, $\boldsymbol{\gamma}^{(j)}(k)$ $(j = 0, 1, \cdots)$ defined by (4.14) is the sequence of iterates solving the lasso problem $\min_{\boldsymbol{\gamma}}\Phi_k(\boldsymbol{\gamma})$, in the way of (4.4).

This gives another explanation of our approach from the *penalty functions* and we immediately know that (see, e.g., [3])

Fact 2) $f(\boldsymbol{\gamma}(k))\uparrow$, $f(\boldsymbol{\gamma}(k)) \le f_{opt}$.

From Fact 2), $\lambda\|\boldsymbol{\gamma}(k)\|_1 \le f_{opt}$. We have

Fact 3) $\|\boldsymbol{\gamma}(k)\|$ is uniformly bounded.

The KKT equation yields

$$\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}(k) = \frac{1}{k^2}(\boldsymbol{H}^T\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H}\boldsymbol{\gamma}(k) - \lambda\widetilde{\text{sgn}}(\boldsymbol{\gamma}(k))).$$

It follows from Fact 3) that

Fact 4) $\|\Delta(k)\| = \|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}(k)\| = O(\frac{1}{k^2})$ and $\|\Delta(k)\|_2 \downarrow 0$.

The latter result is due to the penalty function again.

• Generalization of Daubechies *et al.*'s Convergence Theorem

Although we have Fact 1), Daubechies *et al.*'s convergence theorem [10], which makes use of Opial's conditions [21] in studying the nonexpansive operators, can *not* be directly applied, because the 2-norm of the operator $\boldsymbol{K}$ satisfying

$$\boldsymbol{K}^*\boldsymbol{K} = \frac{1}{k^2}\boldsymbol{H}^T\boldsymbol{\Sigma}\boldsymbol{H} + \boldsymbol{U}_\perp\boldsymbol{U}_\perp^T = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{U}_\perp\end{bmatrix}\begin{bmatrix}\frac{1}{k^2}\boldsymbol{A} & \\ & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{U}^T \\ \boldsymbol{U}_\perp^T\end{bmatrix} \qquad \text{(C.5)}$$

where

$$\boldsymbol{A} \triangleq \boldsymbol{D}^{-1}\boldsymbol{V}^T\boldsymbol{\Sigma}\boldsymbol{V}\boldsymbol{D}^{-1}, \qquad \text{(C.6)}$$

is exactly 1, whereas Theorem 3.1 [10] requires $\|\boldsymbol{K}\|_2 < 1$. Therefore, we need a generalization of Daubechies *et al.*'s (weak) convergence result.

In fact, we can generalize Theorem 3.1 (or Proposition 3.11 to be more specific) in [10] to $\boldsymbol{K}$ satisfying

$$\|\boldsymbol{K}\|_2 < \sqrt{2}, \qquad \text{(C.7)}$$

which may also validate the over-relaxation technique used to speed the convergence. In this part, we will use some notation compatible with [10], with mild changes. (Our thresholding operator $\Theta(\cdot;\lambda)$ uses a threshold value λ instead of $\lambda/2$.)

Let

$$\Phi(\boldsymbol{f}) = \frac{1}{2}\|\boldsymbol{K}\boldsymbol{f}-\boldsymbol{g}\|_2^2+\lambda\|\boldsymbol{f}\|_1, \Phi^{\mathrm{SUR}}(\boldsymbol{f};\boldsymbol{a}) = \Phi(\boldsymbol{f})+\frac{1}{2}(\boldsymbol{f}-\boldsymbol{a})^T\boldsymbol{J}(\boldsymbol{f}-\boldsymbol{a}), \quad \text{(C.8)}$$

and $\boldsymbol{J} \triangleq \boldsymbol{I}-\boldsymbol{K}^*\boldsymbol{K}$. The iterative process can be represented as

$$\boldsymbol{f}^{n+1} = \Theta(\boldsymbol{J}\boldsymbol{f}^n + \boldsymbol{K}^*\boldsymbol{g};\lambda) \quad \text{(C.9)}$$

Since $\|\boldsymbol{K}\|_2 < \sqrt{2}$, $-1 < \mathrm{eig}(\boldsymbol{J}) \le 1$, where $\mathrm{eig}(\boldsymbol{J})$ denotes any eigenvalue of $\boldsymbol{J}$. Note that Φ^{SUR} is still strictly convex in $\boldsymbol{f}$ and Proposition 2.1 [10] holds; in particular, for $\boldsymbol{f}_{opt} = \arg\min_{\boldsymbol{f}} \Phi^{\mathrm{SUR}}(\boldsymbol{f};\boldsymbol{a})$ given $\boldsymbol{a}$,

$$\Phi^{\mathrm{SUR}}(\boldsymbol{f}_{opt}+\boldsymbol{h};\boldsymbol{a}) \ge \Phi^{\mathrm{SUR}}(\boldsymbol{f}_{opt};\boldsymbol{a}) + \|\boldsymbol{h}\|_2^2, \quad \forall \boldsymbol{h}. \quad \text{(C.10)}$$

Let $\boldsymbol{f}^{n+1} = \arg\min_{\boldsymbol{f}} \Phi^{\mathrm{SUR}}(\boldsymbol{f};\boldsymbol{f}^n)$. Then it is easy to get

$$\begin{aligned} & \Phi(\boldsymbol{f}^{n+1}) + \frac{1}{2}(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n)^T\boldsymbol{J}(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n) = \Phi^{\mathrm{SUR}}(\boldsymbol{f}^{n+1};\boldsymbol{f}^n) \\ \le\ & \Phi^{\mathrm{SUR}}(\boldsymbol{f}^n;\boldsymbol{f}^n) - \frac{1}{2}\|\boldsymbol{f}^{n+1}-\boldsymbol{f}^n\|_2^2 = \Phi(\boldsymbol{f}^n) - \frac{1}{2}\|\boldsymbol{f}^{n+1}-\boldsymbol{f}^n\|_2^2 \\ \Longrightarrow\ & \Phi(\boldsymbol{f}^{n+1}) + \frac{1}{2}(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n)^T(\boldsymbol{I}+\boldsymbol{J})(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n) \le \Phi(\boldsymbol{f}^n). \end{aligned}$$

Hence $\Phi(\boldsymbol{f}^n) \downarrow$ and the series $\sum_{n=0}^{\infty}(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n)^T(\boldsymbol{I}+\boldsymbol{J})(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n)$ is convergent. On the other hand, since $\mathrm{eig}(\boldsymbol{J}) > -1$, $\|\boldsymbol{f}^{n+1}-\boldsymbol{f}^n\|_2 \le A \cdot \|(\boldsymbol{I}+\boldsymbol{J})^{1/2}(\boldsymbol{f}^{n+1}-\boldsymbol{f}^n)\|_2$, where A is some strictly positive constant.

With these facts, it is not difficult to write out the full proof for the (weak) convergence of $\boldsymbol{f}^n$ for any $\boldsymbol{K}$ satisfying (C.7), by making corresponding changes in Lemma 3.5 and Lemma 3.7 [10]. The details are left to the readers. □

• Proofs of Theorem 4.1 and Proposition 4.2

Now, with Fact 1) and the above generalization, $\boldsymbol{\gamma}^{(j)}, \boldsymbol{\gamma}_o^{(j)}$ defined by (4.14) must converge given any initial value, because $\|\boldsymbol{K}\|_2 = 1 < \sqrt{2}$. (Recall that $\boldsymbol{K}$ is defined by (C.5).) By Fact 3), $\{\boldsymbol{\gamma}(k)\}$ has at least one accumulation point. Consider a subsequence $\boldsymbol{\gamma}(k_l) \to \boldsymbol{\gamma}_o$ as $l \to \infty$. Then $f(\boldsymbol{\gamma}_o) = \lim_{l\to\infty} f(\boldsymbol{\gamma}(k_l)) \le f_{opt}$ due to Fact 2). So any accumulation point of $\boldsymbol{\gamma}(k)$ is an optimal solution.

The convergence rate of $\|\Delta(k)\|$ is covered by Fact 4).

From Fact 2), $f(\boldsymbol{\gamma}(k)) \uparrow$, $f_{opt} - f(\boldsymbol{\gamma}(k)) \ge 0$. So $f(\boldsymbol{\gamma}(k))$ converges. Note that

$$f(\boldsymbol{\gamma}) \triangleq \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{H}\boldsymbol{\gamma}\|_2^2+\lambda\|\boldsymbol{\gamma}\|_1 \ge \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{H}\boldsymbol{\gamma}\|_2^2+\lambda\|\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma}\|_1-\lambda\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}\|_1.$$

It follows from Fact 4) that

$$f(\boldsymbol{\gamma}(k)) \geq \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{H}\boldsymbol{\gamma}(k)\|_2^2 + \lambda\|\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma}(k)\|_1 - \frac{1}{k^2}C.$$

Let g_{opt} be the optimal value of

$$\min g(\boldsymbol{\gamma}) \triangleq \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{H}\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma}\|_2^2+\lambda\|\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma}\|_1 \text{ s.t. } \|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}\| \leq \|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}(k)\|. \tag{C.11}$$

Then

$$f(\boldsymbol{\gamma}(k)) \geq g_{opt} - \frac{C}{k^2}. \tag{C.12}$$

Observe that for any $\boldsymbol{\gamma}$ minimizing (C.11), $\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\gamma} + \theta\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}$ is an optimal solution, too, for $\forall\theta : 0 \leq \theta \leq 1$. It is enough to consider

$$\min g(\boldsymbol{\gamma}) \text{ s.t. } \|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}\| = 0,$$

which is equivalent to

$$\min f(\boldsymbol{\gamma}) \text{ s.t. } \boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma} = \boldsymbol{0}$$

Thus $\boldsymbol{\gamma}_{opt}$ is always one optimal solution to (C.11) given any k. By (C.12),

$$f(\boldsymbol{\gamma}(k)) \geq g(\boldsymbol{\gamma}_{opt}) - \frac{C}{k^2} = f(\boldsymbol{\gamma}_{opt}) - \frac{C}{k^2}.$$

For Relaxation (II), it is of the same form as (C.9) if we let

$$\boldsymbol{J}(\text{or } \boldsymbol{J}_k) = \boldsymbol{I} - \omega\boldsymbol{K}^T\boldsymbol{K}, \text{ for } 0 < \omega < 2, \tag{C.13}$$

with $\omega = 1$ corresponding to the non-relaxed version (4.14) (or (4.22)). Since $\sqrt{\omega}\cdot\|\boldsymbol{K}\|_2 < \sqrt{2}$, $\boldsymbol{\gamma}^{(j)}$ defined by (4.24) converges. Clearly, the above conclusions and proofs go through.

For the choice of k_0, our generalization guarantees the convergence if (C.7) is satisfied, whereas

$$\omega\|\boldsymbol{A}\|_2 \cdot \frac{1}{k^2} < 2 \Longleftrightarrow k^2 > \frac{\omega}{2}\|\boldsymbol{D}^{-1}\boldsymbol{V}^T\boldsymbol{\Sigma}\boldsymbol{V}\boldsymbol{D}^{-1}\|_2.$$

Since

$$\|\boldsymbol{D}^{-1}\boldsymbol{V}^T\boldsymbol{\Sigma}\boldsymbol{V}\boldsymbol{D}^{-1}\|_2 \leq \frac{\sigma_{\max}^2(\boldsymbol{X})}{\sigma_{\min}^2(\boldsymbol{T})},$$

it is sufficient to have

$$k > \sqrt{\frac{\omega}{2}} \cdot \frac{\sigma_{\max}(\boldsymbol{X})}{\sigma_{\min}(\boldsymbol{T})}.$$

Hence $k_0 \leq \sqrt{\frac{\omega}{2}} \cdot \frac{\sigma_{\max}(\boldsymbol{X})}{\sigma_{\min}(\boldsymbol{T})}$.

The cooling schedule part of Proposition 4.2 is left to the proof of Theorem 4.2. □

• **Proof of Proposition 4.1**

Represent the iteration of $\boldsymbol{\gamma}^{(j)}$ by nonexpansive operators $\widetilde{T}_k, \Theta_k, T_k$:

$$\boldsymbol{\gamma}^{(j+1)} = \widetilde{T}_k \circ \boldsymbol{\gamma}^{(j)} = \Theta_k \circ (T_k \circ \boldsymbol{\gamma}^{(j)}), \tag{C.14}$$

where $\Theta_k \circ \boldsymbol{v} = \Theta(\boldsymbol{v}; \lambda/k^2)$, $T_k \circ \boldsymbol{v} = \boldsymbol{J}_k \boldsymbol{v} + \boldsymbol{\alpha}_k$ with $\boldsymbol{\alpha}_k = \boldsymbol{H}^T \boldsymbol{X}^T \boldsymbol{y}/k^2$. $\Theta_k, T_k, \widetilde{T}_k$ are nonexpansive in that

$$\|\Theta_k \circ \boldsymbol{v} - \Theta_k \circ \boldsymbol{v}'\| \le \|\boldsymbol{v} - \boldsymbol{v}'\|, \quad \|\widetilde{T}_k \circ \boldsymbol{v} - \widetilde{T}_k \circ \boldsymbol{v}'\| \le \|T_k \circ \boldsymbol{v} - T_k \circ \boldsymbol{v}'\| \le \|\boldsymbol{v} - \boldsymbol{v}'\|.$$

(See Lemma 2.2 and Lemma 3.4 of [10].) Define $\overline{T}_k = T_k \circ \Theta_k$ to be used later.

If $\boldsymbol{\Sigma}$ is nonsingular, $\text{eig}(\boldsymbol{K}) > 0$, and thus $\widetilde{T}_k$ becomes a contraction. It is not difficult to show (4.18) since $\lambda_{\min}(\boldsymbol{A}) = \lambda^+_{\min}\left(\tilde{\boldsymbol{U}}\left[\begin{smallmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{smallmatrix}\right]\tilde{\boldsymbol{U}}^T\right) = \lambda^+_{\min}(\boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H})$. Since $f(\boldsymbol{\gamma})$ is strictly convex, $\boldsymbol{\gamma}_{opt}$ is unique.

To prove the finite-k sign consistency we introduce the following result:

Fact 5) Let $\boldsymbol{v}_o$ be the unique optimal solution of the convex optimization $\min f_0(\boldsymbol{v}) \triangleq h(\boldsymbol{v}) + \|\boldsymbol{B}(\boldsymbol{v})\|_1$ with $h, \boldsymbol{B}$ smooth. Define the index sets $z = \{i : (\boldsymbol{B}(\boldsymbol{v}_o))_i = 0\}$, and $nz = \{i : (\boldsymbol{B}(\boldsymbol{v}_o))_i \neq 0\}$. Let $\boldsymbol{v}_{oo}$ be the optimal solution of

$$\min_{\boldsymbol{v}} h(\boldsymbol{v}) + \text{sgn}\,(\boldsymbol{B}(\boldsymbol{v}_o)_{nz})^T \boldsymbol{B}(\boldsymbol{v})_{nz} \text{ s.t. } (\boldsymbol{B}(\boldsymbol{v}))_z = \boldsymbol{0}. \tag{C.15}$$

Then $\boldsymbol{v}_o = \boldsymbol{v}_{oo}$.

In fact, by the generalized KKT (see, e.g., [25]), $\boldsymbol{v}_o$ solves $\min f_0(\boldsymbol{v})$ if and only if

$$\nabla h(\boldsymbol{v}_o) + D\boldsymbol{B}(\boldsymbol{v}_o)^T \widetilde{\text{sgn}}\,(\boldsymbol{B}(\boldsymbol{v}_o)) = \boldsymbol{0}.$$

Let $\boldsymbol{b} = \widetilde{\text{sgn}}\,(\boldsymbol{B}(\boldsymbol{v}_o))$. Then $\min f_0(\boldsymbol{v}) \iff \min f_1(\boldsymbol{v}) \triangleq h(\boldsymbol{v}) + \boldsymbol{b}^T \boldsymbol{B}(\boldsymbol{v})$. And we know $b_i = \pm 1, \forall i \in nz$, $b_i \in [-1, 1], \forall i \in z$. Now consider $\min f_2(\boldsymbol{v}) \triangleq h(\boldsymbol{v}) + \boldsymbol{b}^T_{nz} \cdot (\boldsymbol{B}(\boldsymbol{v}))_{nz}$ s.t. $(\boldsymbol{B}(\boldsymbol{v}))_z = \boldsymbol{0}$ with an optimal solution $\boldsymbol{v}_{oo}$. We have $f_1(\boldsymbol{v}_o) = f_2(\boldsymbol{v}_o) \ge f_2(\boldsymbol{v}_{oo}) = f_1(\boldsymbol{v}_{oo})$. Hence $\boldsymbol{v}_o = \boldsymbol{v}_{oo}$.

Back to our problem, observe that $\boldsymbol{\eta}_k \triangleq \boldsymbol{U}^T \boldsymbol{\gamma}(k), \boldsymbol{\eta}_{opt} \triangleq \boldsymbol{U}^T \boldsymbol{\gamma}_{opt}$ respectively solve

$$\min_{\boldsymbol{\eta}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{V}\boldsymbol{D}^{-1} \cdot \boldsymbol{\eta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{U}\boldsymbol{\eta} + \boldsymbol{U}_\perp \boldsymbol{U}_\perp^T \boldsymbol{\gamma}(k)\|_1,$$

and

$$\min \frac{1}{2} \|\boldsymbol{X}\boldsymbol{V}\boldsymbol{D}^{-1} \cdot \boldsymbol{\eta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{U}\boldsymbol{\eta}\|_1.$$

Define index sets $z = \{i : (\boldsymbol{\gamma}_{opt})_i = 0\}$, $nz = \{i : (\boldsymbol{\gamma}_{opt})_i \neq 0\}$. Given any index set I, we use $\boldsymbol{U}_I$ to denote the submatrix of $\boldsymbol{U}$ composed of its corresponding rows such that $(\boldsymbol{U}\boldsymbol{\alpha})_I = \boldsymbol{U}_I \cdot \boldsymbol{\alpha}, \forall \boldsymbol{\alpha}$. Fact 5) states that $\boldsymbol{\eta}_{opt}$ solves

$$\min \frac{1}{2} \|\boldsymbol{X}\boldsymbol{V}\boldsymbol{D}^{-1} \cdot \boldsymbol{\eta} - \boldsymbol{y}\|_2^2 + \lambda \text{sgn}\left((\boldsymbol{\gamma}_{opt})_{nz}\right)^T \cdot (\boldsymbol{U}\boldsymbol{\eta})_{nz} \text{ s.t. } (\boldsymbol{U}\boldsymbol{\eta})_z = \boldsymbol{0}, \tag{C.16}$$

because $\boldsymbol{U}\boldsymbol{\eta}_{opt} = \boldsymbol{\gamma}_{opt}$. Clearly, $\text{sgn}\left((\boldsymbol{U}\boldsymbol{\eta}_k)_{nz}\right) = \text{sgn}\left((\boldsymbol{\gamma}_{opt})_{nz}\right)$ for k large enough since $\boldsymbol{U}\boldsymbol{\eta}_k \to \boldsymbol{\gamma}_{opt}$. We claim that $(\boldsymbol{U}\boldsymbol{\eta}_k)_z = (\boldsymbol{\gamma}_{opt})_z = \mathbf{0}$ is also true for any k large enough.

Otherwise, noticing $\boldsymbol{\gamma}_{opt}$ is finite dimensional, there must exist some index sets $nzz \subset z$, and $zz = z \backslash nzz$ such that each component of $(\boldsymbol{U}\boldsymbol{\eta}_{k_j})_{nzz}$ is nonzero, and $(\boldsymbol{U}\boldsymbol{\eta}_{k_j})_{zz} = \mathbf{0}$, for some subsequence $\boldsymbol{\eta}_{k_j}$ with $k_j \to \infty$ as $j \to \infty$, which implies $\boldsymbol{U}_\perp \boldsymbol{U}_\perp^T \boldsymbol{\gamma}(k_j) \to \mathbf{0}$. It follows that a further subsequence of $\boldsymbol{\eta}_{k_j}$ asymptotically solves

$$\begin{aligned} \min \frac{1}{2}\|\boldsymbol{X}\boldsymbol{V}\boldsymbol{D}^{-1} \cdot \boldsymbol{\eta} - \boldsymbol{y}\|_2^2 + \lambda \text{sgn}\left((\boldsymbol{\gamma}_{opt})_{nz}\right)^T \cdot (\boldsymbol{U}\boldsymbol{\eta})_{nz} \\ + \lambda \boldsymbol{b}_{nzz}^T \cdot (\boldsymbol{U}_{nzz}\boldsymbol{\eta}) \text{ s.t. } \boldsymbol{U}_{zz}\boldsymbol{\eta} = \mathbf{0} \end{aligned} \tag{C.17}$$

for some sign vector $\boldsymbol{b}_{nzz}$ (with each component ± 1). Obviously, none of the rows of $\boldsymbol{U}_{nzz}$ lies in the (row) space spanned by the row vectors of $\boldsymbol{U}_{zz}$. Excluding the case of degeneracy, the optimization problem does not have the same optimal solution $\boldsymbol{\eta}_{opt}$ as (C.16). Hence the finite-k sign consistency holds. We also know for k large, $\boldsymbol{\eta}_k$ solves

$$\min \frac{1}{2}\|\boldsymbol{X}\boldsymbol{V}\boldsymbol{D}^{-1}\cdot\boldsymbol{\eta}-\boldsymbol{y}\|_2^2+\lambda \text{sgn}\left((\boldsymbol{\gamma}_{opt})_{nz}\right)^T\cdot(\boldsymbol{U}\boldsymbol{\eta})_{nz} \text{ s.t. } (\boldsymbol{U}\boldsymbol{\eta})_z=\left(\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\boldsymbol{\gamma}(k)\right)_z. \tag{C.18}$$

Note that (C.18) is a simple quadratic programming (QP) problem.

Let $rz \subset z$ be one index set such that $\boldsymbol{U}_{rz}$ has full row rank and $\text{rank}(\boldsymbol{U}_{rz}) = \text{rank}(\boldsymbol{U}_z)$. Since $\boldsymbol{\eta}_k$ always exists, the optimization problem (C.18) can be simplified into

$$\min \frac{1}{2}\|\boldsymbol{X}\boldsymbol{V}\boldsymbol{D}^{-1}\cdot\boldsymbol{\eta}-\boldsymbol{y}\|_2^2+\lambda \text{sgn}\left((\boldsymbol{\gamma}_{opt})_{nz}\right)^T\cdot(\boldsymbol{U}_{nz}\boldsymbol{\eta}) \text{ s.t. } \boldsymbol{U}_{rz}\boldsymbol{\eta} = (\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T)_{rz}\boldsymbol{\gamma}(k),$$

or

$$\min \frac{1}{2}\boldsymbol{\eta}^T\boldsymbol{A}\boldsymbol{\eta} - \boldsymbol{\alpha}^T\boldsymbol{\eta} \text{ s.t. } \boldsymbol{U}_{rz}\boldsymbol{\eta} = \boldsymbol{\delta}_k, \tag{C.19}$$

where $\boldsymbol{\alpha} = \boldsymbol{D}^{-1}\boldsymbol{V}^T\boldsymbol{X}^T\boldsymbol{y} - \lambda\boldsymbol{U}^T\text{sgn}((\boldsymbol{\gamma}_{opt})_{nz})$, $\boldsymbol{\delta}_k = (\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T)_{rz}\boldsymbol{\gamma}(k)$.

Solving this QP, we obtain

$$\begin{aligned} \boldsymbol{\eta}_k &= \left\{\boldsymbol{A}^{-1}\boldsymbol{\alpha} - \boldsymbol{A}^{-1}\boldsymbol{U}_{rz}^T(\boldsymbol{U}_{rz}\boldsymbol{A}^{-1}\boldsymbol{U}_{rz}^T)^{-1}\boldsymbol{U}_{rz}\boldsymbol{A}^{-1}\boldsymbol{\alpha}\right\} \\ &\quad + \boldsymbol{A}^{-1}\boldsymbol{U}_{rz}^T(\boldsymbol{U}_{rz}\boldsymbol{A}^{-1}\boldsymbol{U}_{rz}^T)^{-1}\boldsymbol{\delta}_k. \end{aligned} \tag{C.20}$$

Note that since $\boldsymbol{U}_{rz}$ has full row rank, $(\boldsymbol{U}_{rz}\boldsymbol{A}^{-1}\boldsymbol{U}_{rz}^T)^{-1}$ exists. Now it follows immediately that

Lemma C.1. $\|\boldsymbol{U}\boldsymbol{U}^T \cdot (\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k'))\| \le C \cdot \|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T \cdot (\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k'))\|, \quad \forall k, k'.$

Letting $k \to \infty$, we get the convergence rate of $\boldsymbol{\gamma}(k)$: $\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}_{opt}\| = O(1/k^2)$. □

• **Proof of Theorem 4.2**

We prove the theorem for the general relaxed case in the form of (II), where $0 < \omega < 2$, with $\omega = 1$ corresponding to the non-relaxed version; see (C.13). The operators introduced in (C.14) will be used for simplicity, except that Θ_k is redefined by $\Theta_k \circ \boldsymbol{v} = \Theta(\boldsymbol{v}; \omega\lambda/k^2), \forall \boldsymbol{v}$.

First, from

$$\begin{aligned} & \widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(1)} \circ \boldsymbol{\gamma}^{(0)} - \boldsymbol{\gamma}_{opt} \\ = \; & \left(\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ (\widetilde{T}_{k(N-1)} \circ \cdots \circ \widetilde{T}_{k(1)} \circ \boldsymbol{\gamma}^{(0)}) - \widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt}\right) \\ & + (\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt} - \boldsymbol{\gamma}_{opt}), \end{aligned}$$

we get

$$\begin{aligned} & \|\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(1)} \circ \boldsymbol{\gamma}^{(0)} - \boldsymbol{\gamma}_{opt}\| \\ \leq \; & \|\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ (\widetilde{T}_{k(N-1)} \circ \cdots \circ \widetilde{T}_{k(1)} \circ \boldsymbol{\gamma}^{(0)}) - \widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt}\| \\ & + \|\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt} - \boldsymbol{\gamma}_{opt}\| \\ \leq \; & (\|T_{k(n)}\| \cdot \cdots \cdot \|T_{k(N)}\|) \cdot \|\widetilde{T}_{k(N-1)} \circ \cdots \circ \widetilde{T}_{k(1)} \circ \boldsymbol{\gamma}^{(0)} - \boldsymbol{\gamma}_{opt}\| \\ & + \|\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt} - \boldsymbol{\gamma}_{opt}\| \triangleq \mathrm{I} \cdot \mathrm{II} + \mathrm{III}. \end{aligned}$$

That is,

$$\|\widetilde{T}_{k(n)} \circ \cdots \circ \widetilde{T}_{k(1)} \circ \boldsymbol{\gamma}^{(0)} - \boldsymbol{\gamma}_{opt}\| \leq \mathrm{I} \cdot \mathrm{II} + \mathrm{III} \tag{C.21}$$

in short. Moreover,

$$\begin{aligned} & \widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt} - \boldsymbol{\gamma}_{opt} \\ = \; & \left(\widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt} - \widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}(k(N))\right) \\ & + \left(\widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N+1)} \circ \boldsymbol{\gamma}(k(N)) - \boldsymbol{\gamma}_{opt}\right) \\ = \; & \left(\widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}_{opt} - \widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N)} \circ \boldsymbol{\gamma}(k(N))\right) \\ & + \sum_{j=1}^{M} \Big\{\widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N+j)} \circ \boldsymbol{\gamma}(k(N+j-1)) - \\ & \widetilde{T}_{k(N+M)} \circ \cdots \circ \widetilde{T}_{k(N+j)} \circ \boldsymbol{\gamma}(k(N+j))\Big\} + \left(\boldsymbol{\gamma}(k(N+M)) - \boldsymbol{\gamma}_{opt}\right). \end{aligned}$$

Hence

$$\mathrm{III} \leq 2 \sup_{j \geq N} \|\boldsymbol{\gamma}(k(j)) - \boldsymbol{\gamma}_{opt}\| + \sum_{j \geq N} \|\boldsymbol{\gamma}(k(j)) - \boldsymbol{\gamma}(k(j+1))\|. \tag{C.22}$$

Since $\boldsymbol{\Sigma}$ is nonsingular,

$$\mathrm{I} \leq \prod_{j=N}^{n} \left(1 - \frac{\omega\rho_0}{k^2(j)}\right) = \exp\left(\sum_{N}^{n} \log\left(1 - \frac{\omega\rho_0}{k^2(j)}\right)\right) \leq \exp\left(-\sum_{N}^{n} \frac{1}{k^2(j)} \cdot \omega\rho_0\right). \tag{C.23}$$

If we can show

$$\sum_{1}^{\infty} \|\boldsymbol{\gamma}(k(j)) - \boldsymbol{\gamma}(k(j+1))\| \text{ converges }, \tag{C.24}$$

then since $k(j) \to \infty$, $\exists N$ such that $\sup_{j \geq N} \|\boldsymbol{\gamma}(k(j)) - \boldsymbol{\gamma}_{opt}\|$, $\sum_{N}^{\infty} \|\boldsymbol{\gamma}(k(j)) - \boldsymbol{\gamma}(k(j+1))\|$, and thus III, are small enough. For this N, $\exists M$ such that $\sum_{N}^{N+M} \frac{1}{k^2(j)}$ is large enough to guarantee I·II is small enough. So any cooling schedule satisfying

$$\sum_{j=1}^{\infty} \frac{1}{k^2(j)} = \infty, \text{ and } k(j) \to \infty,$$

guarantees the convergence to the optimal point $\boldsymbol{\gamma}_{opt}$.

In the remainder, we will prove (C.24). It is enough to show

Lemma C.2. *$\|\boldsymbol{\gamma}(k) - \boldsymbol{\gamma}(k')\|_2 \leq \left(\frac{1}{k^2} - \frac{1}{k'^2}\right) \cdot C$ for $\forall k, k' : k \leq k'$.*

We still consider the general relaxation form (II), with $0 < \omega < 2$.

$$\begin{aligned}
& \|\boldsymbol{\gamma}(k') - \boldsymbol{\gamma}(k)\|_2 \\
\leq\ & \|\boldsymbol{\gamma}(k') - \widetilde{T}_k \circ \boldsymbol{\gamma}(k') + \widetilde{T}_k \circ \boldsymbol{\gamma}(k') - \boldsymbol{\gamma}(k)\|_2 \\
\leq\ & \|\boldsymbol{\gamma}(k') - \widetilde{T}_k \circ \boldsymbol{\gamma}(k')\|_2 + \|\widetilde{T}_k \circ \boldsymbol{\gamma}(k') - \widetilde{T}_k \circ \boldsymbol{\gamma}(k)\|_2 \\
\leq\ & \|\widetilde{T}_{k'} \circ \boldsymbol{\gamma}(k') - \Theta_k \circ T_{k'} \circ \boldsymbol{\gamma}(k') + \Theta_k \circ T_{k'} \circ \boldsymbol{\gamma}(k') - \widetilde{T}_k \circ \boldsymbol{\gamma}(k')\|_2 \\
& + \|T_k \circ \boldsymbol{\gamma}(k') - T_k \circ \boldsymbol{\gamma}(k)\|_2 \\
\leq\ & \|\Theta_{k'} \circ (T_{k'} \circ \boldsymbol{\gamma}(k')) - \Theta_k \circ (T_{k'} \circ \boldsymbol{\gamma}(k'))\|_2 \\
& + \|\Theta_k \circ (T_{k'} \circ \boldsymbol{\gamma}(k')) - \Theta_k \circ (T_k \circ \boldsymbol{\gamma}(k'))\|_2 + \|T_k \circ \boldsymbol{\gamma}(k') - T_k \circ \boldsymbol{\gamma}(k)\|_2 \\
\leq\ & \|\Theta_{k'} \circ (T_{k'} \circ \boldsymbol{\gamma}(k')) - \Theta_k \circ (T_{k'} \circ \boldsymbol{\gamma}(k'))\|_2 + \|T_{k'} \circ \boldsymbol{\gamma}(k') - T_k \circ \boldsymbol{\gamma}(k')\|_2 \\
& + \|T_k \circ \boldsymbol{\gamma}(k') - T_k \circ \boldsymbol{\gamma}(k)\|_2 \triangleq \mathrm{I}^* + \mathrm{II}^* + \mathrm{III}^*
\end{aligned}$$

That is,

$$\|\boldsymbol{\gamma}(k') - \boldsymbol{\gamma}(k)\|_2 \leq \mathrm{I}^* + \mathrm{II}^* + \mathrm{III}^* \tag{C.25}$$

It is easy to verify

$$|\Theta_{k'}\boldsymbol{v} - \Theta_k \boldsymbol{v}| \preceq \lambda\omega \left(\frac{1}{k^2} - \frac{1}{k'^2}\right),$$

where '$\preceq$' means the component-wise '$\leq$'. Therefore,

$$\mathrm{I}^* \leq C \cdot \left(\frac{1}{k^2} - \frac{1}{k'^2}\right). \tag{C.26}$$

Using Fact 3), we have

$$\mathrm{II}^* = \left\| \left(\frac{1}{k^2} - \frac{1}{k'^2}\right) (\omega \cdot \boldsymbol{U}\boldsymbol{A}\boldsymbol{U}^T \boldsymbol{\gamma}(k') - \boldsymbol{H}^T \boldsymbol{X}\boldsymbol{y}) \right\|_2 \leq \left(\frac{1}{k^2} - \frac{1}{k'^2}\right) \cdot C. \tag{C.27}$$

To control III*, we rewrite it as

$$\begin{aligned}
\text{III}^{*2} &= \|\boldsymbol{J}_k\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\|_2^2 = \left\|\left(\boldsymbol{I}-\omega\boldsymbol{K}^T\boldsymbol{K}\right)\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|_2^2 \\
&= \left\|\left(\boldsymbol{U}(\boldsymbol{I}-\frac{\omega}{k^2}\boldsymbol{A})\boldsymbol{U}^T+(1-\omega)\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\right)\cdot\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|_2^2 \\
&= \left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)^T\cdot\boldsymbol{U}(\boldsymbol{I}-\frac{\omega}{k^2}\boldsymbol{A})^2\boldsymbol{U}^T\cdot\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right) \\
&\quad +(1-\omega)^2\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)^T\cdot\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\cdot\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right).
\end{aligned}$$

Hence,

$$\begin{aligned}
\text{III}^* \leq &\left(\left(1-\omega\frac{\epsilon}{k^2}\right)^2\left\|\boldsymbol{U}\boldsymbol{U}^T\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|_2^2\right. \\
&\left.+(1-\omega)^2\left\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|_2^2\right)^{1/2}, \qquad \text{(C.28)}
\end{aligned}$$

for some $\epsilon>0$, because $\boldsymbol{\Sigma}$ and thus $\boldsymbol{A}$ are nonsingular.

Summarizing (C.26), (C.27), and (C.28), we obtain

$$\sqrt{\tau_1^2+\tau_2^2}-\sqrt{(1-\frac{\omega\epsilon}{k^2})^2\tau_1^2+(1-\omega)^2\tau_2^2}\leq C\cdot\left(\frac{1}{k^2}-\frac{1}{k'^2}\right)$$

where $\tau_1=\left\|\boldsymbol{U}\boldsymbol{U}^T\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|_2$, $\tau_2=\left\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|_2$.

Using Lemma C.1 and the fact that $0<\omega<2$, we get

$$\begin{aligned}
&\sqrt{\tau_1^2+\tau_2^2}-\sqrt{(1-\frac{\omega\epsilon}{k^2})^2\tau_1^2+(1-\omega)^2\tau_2^2} \\
=\ &\frac{\tau_1^2+\tau_2^2-\left(1-\frac{\omega\epsilon}{k^2}\right)^2\tau_1^2-(1-\omega)^2\tau_2^2}{\sqrt{\tau_1^2+\tau_2^2}+\sqrt{\left(1-\frac{\omega\epsilon}{k^2}\right)^2\tau_1^2+(1-\omega)^2\tau_2^2}} \\
\geq\ &\frac{\epsilon'\tau_2^2}{2\sqrt{\tau_1^2+\tau_2^2}}\geq\epsilon''\cdot\tau_2
\end{aligned}$$

for some $\epsilon',\epsilon''>0$. Hence

$$\left\|\boldsymbol{U}_\perp\boldsymbol{U}_\perp^T\left(\boldsymbol{\gamma}(k')-\boldsymbol{\gamma}(k)\right)\right\|\leq C\cdot\left(\frac{1}{k^2}-\frac{1}{k'^2}\right)$$

By Lemma C.1 again, Lemma C.2 is true. Now the proof of Theorem 4.2 is complete. □

• **Proof of Proposition 4.3**

First (4.23) can be rewritten using the introduced operators:

$$\boldsymbol{\xi}^{(j+1)}=((1-\omega)I+\omega\overline{T}_k)\circ\boldsymbol{\xi}^{(j)}. \qquad \text{(C.29)}$$

Obviously, $\overline{T}_k$ is nonexpansive. We claim that the set of fixed points of $\overline{T}_k$, denoted by F, is nonempty. In fact, let $\boldsymbol{\gamma}$ be a minimizer of the convex function Φ_k defined by (C.4). The KKT optimality condition gives

$$\boldsymbol{\gamma} = \Theta_k \circ (\boldsymbol{J}_k\boldsymbol{\gamma} + \boldsymbol{\alpha}_k) = \widetilde{T}_k \circ \boldsymbol{\gamma}.$$

Let $\boldsymbol{\xi} = \boldsymbol{J}_k\boldsymbol{\gamma} + \boldsymbol{\alpha}_k$. Then $\boldsymbol{\xi} = \boldsymbol{J}_k(\Theta_k \circ \boldsymbol{\xi}) + \boldsymbol{\alpha}_k = \overline{T}_k \circ \boldsymbol{\xi}$. So $\overline{T}_k$ has at least one fixed point. In the rest of this proof, all subscripts k are abbreviated for simplicity.

For $0 < \omega < 1$, (C.29) is the Mann iterates [19] introduced for nonexpansive mapping $\overline{T}$. The sequence is known to converge to a fixed point of $\overline{T}$ if F is nonempty; see Opial [21], Browder and Petryshyn [7], or Dotson [12].

Now consider $1 < \omega < 2$. Let $\omega = 1 + \omega'$. So $\omega' \in (0, 1)$ and

$$\boldsymbol{\xi}^{(j+1)} = \omega'(2\overline{T} - I) \circ \boldsymbol{\xi}^{(j)} + (1 - \omega')\overline{T} \circ \boldsymbol{\xi}^{(j)}$$

If $2\overline{T} - I$ is nonexpansive, $(1 - \omega)I + \omega\overline{T}$ is nonexpansive for any $\omega \in (1, 2)$.

Let $\boldsymbol{\xi} \in F$. Clearly, $\overline{T} \circ \boldsymbol{\xi} = \boldsymbol{\xi} = (2\overline{T} - I) \circ \boldsymbol{\xi}$. On the one hand,

$$\begin{aligned}
\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}\|_2^2 &= \left\|\omega'\left((2\overline{T} - I) \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\right) + (1 - \omega')\left(\overline{T} \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\right)\right\|_2^2 \\
&\le \omega'^2\|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\|_2^2 + (1 - \omega')^2\|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\|_2^2 \\
&\quad +2\omega'(1 - \omega')\langle(2\overline{T} - I) \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}, \overline{T} \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\rangle.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
a^2\|\boldsymbol{\xi}^{(j)} - \overline{T} \circ \boldsymbol{\xi}^{(j)}\|_2^2 &= a^2\|(2\overline{T} - I) \circ \boldsymbol{\xi}^{(j)} - \overline{T} \circ \boldsymbol{\xi}^{(j)}\|_2^2 \\
&= a^2\left\|\left((2\overline{T} - I) \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\right) - \left(\overline{T} \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\right)\right\|_2^2 \\
&\le a^2\|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\|_2^2 + a^2\|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\|_2^2 \\
&\quad -2a^2\langle(2\overline{T} - I) \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}, \overline{T} \circ \boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\rangle.
\end{aligned}$$

Letting $a^2 = \omega'(1 - \omega')$, we obtain

$$\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}\|_2^2 + \omega'(1 - \omega')\|\boldsymbol{\xi}^{(j)} - \overline{T} \circ \boldsymbol{\xi}^{(j)}\|_2^2 \le \|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\|_2^2,$$

and so

$$\begin{aligned}
\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\|_2^2 &= \omega^2\|\boldsymbol{\xi}^{(j)} - \overline{T} \circ \boldsymbol{\xi}^{(j)}\|_2^2 \\
&\le \frac{\omega^2}{(\omega - 1)(2 - \omega)}\left(\|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}\|_2^2 - \|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}\|_2^2\right).
\end{aligned}$$

It follows that $\sum\|\boldsymbol{\xi}^{(j+1)} - \boldsymbol{\xi}^{(j)}\|_2^2$ converges. Note that we only used quasi-nonexpansiveness [12] in the above proof.

Hence $(1-\omega)I+\omega\overline{T}$ is asymptotically regular – in fact, it is a reasonable wanderer [7]. Furthermore, $\boldsymbol{\xi}^{(j)}$, or $\boldsymbol{\gamma}^{(j)}$, converges by Opial's classical work [21]. □

• **Proof of Proposition 4.4**

The SVD for $\boldsymbol{F}_1$ is well known (see, e.g., [2] for a detailed derivation).

Consider a d-by-d matrix $\boldsymbol{E}$ of all ones: $\boldsymbol{E} = \mathbf{1} \cdot \mathbf{1}^T$. It is easy to diagonalize $\boldsymbol{E}$. First,

$$\boldsymbol{E} \cdot \begin{bmatrix} \mathbf{1} & \boldsymbol{F}_1^T \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{F}_1^T \end{bmatrix} \cdot \mathrm{diag}\{d, 0, \cdots, 0\}.$$

So $\boldsymbol{F}_1^T \perp \mathbf{1}$, i.e., $\boldsymbol{F}_1 \mathbf{1} = \mathbf{0}$. It follows that $\boldsymbol{V}_1^T \mathbf{1} = \boldsymbol{D}_1^{-1} \boldsymbol{U}_1^T \boldsymbol{F}_1 \mathbf{1} = \mathbf{0}$, and $\tilde{\boldsymbol{V}}_1$ is orthonormal. Hence $\boldsymbol{E} = \tilde{\boldsymbol{V}}_1 \mathrm{diag}\{d, 0, \cdots, 0\} \boldsymbol{V}_1^T$.

For $\boldsymbol{T}_1 = \begin{bmatrix} \boldsymbol{I} \\ \lambda \boldsymbol{F}_1 \end{bmatrix}$, we have

$$\boldsymbol{T}_1^T \boldsymbol{T}_1 = \boldsymbol{I} + \lambda^2 \boldsymbol{V}_1 \boldsymbol{D}_1^2 \boldsymbol{V}_1^T = \tilde{\boldsymbol{V}}_1^T \tilde{\boldsymbol{V}}_1 + \lambda^2 \boldsymbol{V}_1 \boldsymbol{D}_1^2 \boldsymbol{V}_1^T = \tilde{\boldsymbol{V}}_1 \begin{bmatrix} 1 & \\ & \boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2 \end{bmatrix} \tilde{\boldsymbol{V}}_1^T.$$

On the other hand,

$$\begin{aligned} \boldsymbol{T}_1 \tilde{\boldsymbol{V}}_1 \begin{bmatrix} 1 & \\ & \boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2 \end{bmatrix}^{-\frac{1}{2}} &= \begin{bmatrix} \boldsymbol{I} \\ \lambda \boldsymbol{F}_1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{d}} \mathbf{1} & \boldsymbol{V}_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \\ & (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2)^{-\frac{1}{2}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\sqrt{d}} \mathbf{1} & \boldsymbol{V}_1 \\ \frac{\lambda}{\sqrt{d}} \cdot \mathbf{0} & \lambda \boldsymbol{U}_1 \boldsymbol{D}_1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \\ & (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_1^2)^{-\frac{1}{2}} \end{bmatrix} \\ &= \tilde{\boldsymbol{U}}_1. \end{aligned}$$

For $\boldsymbol{F}_2$, $\boldsymbol{F}_2^T \boldsymbol{F}_2 = d\boldsymbol{I} - \mathbf{1} \cdot \mathbf{1}^T = \tilde{\boldsymbol{V}}_1 \mathrm{diag}\{0, d, \cdots, d\} \tilde{\boldsymbol{V}}_1^T$. Therefore, $\boldsymbol{D}_2 = \mathrm{diag}\{0, \sqrt{d}, \cdots, \sqrt{d}\}$, and if we take $\boldsymbol{V}_2 = \tilde{\boldsymbol{V}}_1$, $\boldsymbol{U}_2 = \begin{bmatrix} \boldsymbol{u}_{21} & \cdots & \boldsymbol{u}_{2d} \end{bmatrix}$ satisfies $\boldsymbol{F}_2 \tilde{\boldsymbol{V}}_1 = \boldsymbol{U}_2 \boldsymbol{D}_2$. It implies $\begin{bmatrix} \boldsymbol{u}_{22} & \cdots & \boldsymbol{u}_{2d} \end{bmatrix} = \frac{1}{\sqrt{d}} \boldsymbol{F}_2 \boldsymbol{V}_1$. $\boldsymbol{u}_{21}$ is a normalized eigenvector of $\boldsymbol{F}_2 \boldsymbol{F}_2^T$ corresponding to eigenvalue 0 and can take

$$\begin{bmatrix} 0 & \cdots & 0 & 1 & -1 & 1 \end{bmatrix}^T / \sqrt{3},$$

which is easy to verify.

Finally, for $\boldsymbol{T}_2$, $\boldsymbol{T}_2^T \boldsymbol{T}_2 = \boldsymbol{I} + \lambda^2 \boldsymbol{F}_2^T \boldsymbol{F}_2 = \boldsymbol{I} + \lambda^2 \boldsymbol{V}_2 \boldsymbol{D}_2^2 \boldsymbol{V}_2^T = \boldsymbol{V}_2 (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_2^2) \boldsymbol{V}_2^T$. Moreover, $\boldsymbol{T}_2 \boldsymbol{V}_2 (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_2^2)^{-\frac{1}{2}} = \begin{bmatrix} \boldsymbol{I} \\ \lambda \boldsymbol{F}_2 \end{bmatrix} \boldsymbol{V}_2 (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_2^2)^{-\frac{1}{2}} = \begin{bmatrix} \boldsymbol{V}_2 \\ \lambda \boldsymbol{U}_2 \boldsymbol{D}_2 \end{bmatrix} (\boldsymbol{I} + \lambda^2 \boldsymbol{D}_2^2)^{-\frac{1}{2}} = \tilde{\boldsymbol{U}}_2$. □

References

[1] Amaldi, E. and Kann, V. (1998), "On the approximability of minimizing non zero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, 209, 237–260. MR1647530

[2] Anderson, T. (1971), *The Statistical Analysis of Time Series*, New York: Wiley. MR0283939

[3] Bazaraa, M. S. and Shetty, C. M. (1979), *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons. MR0533477

[4] Bertsekas, D. (1999), *Nonlinear Programming*, Athena Scientific.
[5] Bondell, H. D. and Reich, B. J. (2008), "Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR," *Biometrics*, 64, 115–123. MR2422825
[6] Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge, MA: Cambridge University Press. MR2061575
[7] Browder, F. E. and Petryshyn, W. V. (1967), "Construction of fixed points of nonlinear mappings in Hilbert space," *Journal of Mathematical Analysis and Applications*, 20, 197–228. MR0217658
[8] Bunea, F., Tsybakov, A. B., and Wegkamp, M. (2007), "Sparsity oracle inequalities for the lasso," *Electronic Journal of Statistics*, 1, 169–194. MR2312149
[9] Candès, E. and Tao, T. (2005), "The Dantzig selector: statistical estimation when p is much smaller than n," *Annals of Statistics*, 35, 2392–2404. MR2382651
[10] Daubechies, I., Defrise, M., and De Mol, C. (2004), "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, 57, 1413–1457. MR2077704
[11] Dettling, M. and Bühlmann, P. (2004), "Finding Predictive Gene Groups from Microarray Data," *Journal of Multivariate Analysis*, 90, 106–131. MR2064938
[12] Dotson Jr., W. (1970), "On the Mann iterative process," *Trans. Amer. Math. Soc.*, 149, 65–73. MR0257828
[13] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *Annals of Statistics*, 32, 407–499. MR2060166
[14] Friedman, J., Hastie, T., Hofling, H., and Tibshirani, R. (2007), "Pathwise coordinate optimization," *Annals of Applied Statistics*, 1, 302–332. MR2415737
[15] Fu, W. (1998), "Penalized regressions: the bridge vs the lasso," *JCGS*, 7, 397–416. MR1646710
[16] Geyer, C. J. (1996), "On the Asymptotics of Convex Stochastic Optimization".
[17] Jörnsten, R. and Yu, B. (2003), "Simultaneous Gene Clustering and Subset Selection," *Bioinformatics*, 19, 1100–1109.
[18] Knight, K. and Fu, W. (2000), "Asymptotics for lasso-type estimators," *Annals of Statistics*, 28, 1356–1378. MR1805787
[19] Mann, W. R. (1953), "Mean value methods in iteration," *Proc. Amer. Math. Soc.*, 4, 506–510. MR0054846
[20] Nolte, A. and Schrader, R. (2000), "A note on the finite time behavior of simulated annealing," *Math. Operat. Res.*, 25, 476–484. MR1855178
[21] Opial, Z. (1967), "Weak convergence of the sequence of successive approximations for nonexpansive mappings," *Bull. Amer. Math. Soc.*, 73, 591–597. MR0211301
[22] Osborne, M., Presnell, B., and Turlach, B. (2000), "On the LASSO and its Dual," *J. Comput. Graph. Statist.*, 9, 319–337. MR1822089

[23] She, Y. (2008), "Sparse Regression with Exact Clustering," Ph.D. thesis, Stanford University. MR2712372

[24] She, Y. (2009), "Thresholding-based Iterative Selection Procedures for Model Selection and Shrinkage," *Electronic Journal of Statistics*, 3, 384–415. MR2501318

[25] Shimizu, K., Ishizuka, Y., and Bard, J. (1997), *Nondifferentiable and Two-Level Mathematical Programming*, Kluwer Academic Publishers. MR1429865

[26] Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *JRSSB*, 58, 267–288. MR1379242

[27] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and smoothness via the fused lasso," *JRSSB*, 67, 91–108. MR2136641

[28] Wrinkler, G. (1990), "An Ergodic L^2-Theorem for Simulated Annealing in Bayesian Image Reconstruction," *Journal of Applied Probability*, 28, 779–791.

[29] Wu, T. and Lange, K. (2008), "Coordinate Descent Algorithm for Lasso Penalized Regression," *Ann. Appl. Stat.*, 2, 224–244. MR2415601

[30] Yuan, M. and Lin, Y. (2006), "Model selection and estimation in regression with grouped variables," *JRSSB*, 68, 49–67. MR2212574

[31] Zhang, C.-H. and Huang, J. (2008), "The sparsity and bias of the Lasso selection in high-dimensional linear regression," *Ann. Statist*, 36, 1567–1594. MR2435448

[32] Zhao, P. and Yu, B. (2006), "Grouped and Hierarchical Model Selection through Composite Absolute Penalties," Tech. rep., Dept. of Statistics, University of California Berkeley.

[33] Zhao, P. and Yu, B. (2006), "On Model Selection Consistency of Lasso," *Journal of Machine Learning Research*, 7, 2541–2563. MR2274449

[34] Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *JASA*, 101, 1418–1429. MR2279469

[35] Zou, H. and Hastie, T. (2005), "Regularization and Variable Selection via the Elastic Net," *JRSSB*, 67, 301–320. MR2137327

[36] Zou, H. and Li, R. (2008), "One-step Sparse Estimates in Nonconcave Penalized Likelihood Models," *Annals of Statistics*, 1509–1533. MR2435443