Research Article

A Two-Stage Regularization Method for Variable Selection and Forecasting in High-Order Interaction Model

Yao Dong ^{1,2} and **He Jiang** ^{1,2}

¹School of Statistics, Jiangxi University of Finance and Economics, Nanchang 330013, China

²Applied Statistics Research Center, Jiangxi University of Finance and Economics, Nanchang 330013, China

Correspondence should be addressed to He Jiang; jiangsky2005@aliyun.com

Received 19 August 2018; Revised 5 October 2018; Accepted 17 October 2018; Published 11 November 2018

Academic Editor: Mahdi Jalili

Copyright © 2018 Yao Dong and He Jiang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Forecasting models with high-order interaction has become popular in many applications since researchers gradually notice that an additive linear model is not adequate for accurate forecasting. However, the excessive number of variables with low sample size in the model poses critically challenges to predication accuracy. To enhance the forecasting accuracy and training speed simultaneously, an interpretable model is essential in knowledge recovery. To deal with ultra-high dimensionality, this paper investigates and studies a two-stage procedure to demand sparsity within high-order interaction model. In each stage, square root hard ridge (SRHR) method is applied to discover the relevant variables. The application of square root loss function facilitates the parameter tuning work. On the other hand, hard ridge penalty function is able to handle both the high multicollinearity and selection inconsistency. The real data experiments reveal the superior performances to other comparing approaches.

1. Introduction

Sparse representation has attracted a large deal of attention from researchers in different scientific fields including financial engineering and risk management computational biology and machine learning community [1]. For instance, in spectrum data analysis, scientists are exploring how to select few exact frequencies on which the beam energy is concentrated to avoid lower leakage problem [2]. The main goal of demanding sparsity is to discover quite a few numbers of relevant variables from a large pool of candidate variables which are essential when establishing an interpretable forecasting model. A model with high precision, low model complexity, and sensitivity to the model parameters can not only avoid the expensive cost of excessive supply but also reduce the time cost, service interruption, and resource waste [3]. Furthermore, researchers have noticed that a linear additive model using main effects only cannot provide accurate forecasting results. This motivates them to turn their attentions to high-order interaction model which considers interaction terms between variables. Interaction model has advantage over a linear model which does not

consider possible relationship between variables. In microarray data analysis, for example, biologists are particular interested in gene-gene interactions and gene-environment interactions than a single gene because interactions of single nucleotide polymorphisms (SNPs) are more important in cancer diagnosis [4]. However, new studies have also revealed the interactive effect of multiple environment factors. For example, a child with a poor quality environment would be more sensitive to a poor environment as an adult which ultimately led to higher psychological distress scores. This depicts a three-way interaction term between gene and environment. Furthermore, interaction with multiple genes is also considered to be essentially important in the molecular analysis [5]. This motivates us to consider a high-order interaction model which includes both two-way and three-way interaction terms in this paper:

$$\mathbf{y} = b_0 + \sum_j b_j \mathbf{x}_j + \sum_{j,k} \phi_{jk} \mathbf{x}_j \odot \mathbf{x}_k + \sum_{j,k,l} \psi_{jkl} \mathbf{x}_j \odot \mathbf{x}_k \odot \mathbf{x}_l + \boldsymbol{\varepsilon} \quad (1)$$

where b_0 is the intercept term which vanishes after centering, $\mathbf{x}_j \odot \mathbf{x}_k \odot \mathbf{x}_l$ represents the Hadamard product of $\mathbf{x}_j$, $\mathbf{x}_k$, and $\mathbf{x}_l$ which denotes the three-way interaction term. The noise term follows Gaussian distribution. b_j , ϕ_{jk} , and ψ_{jkl} are corresponding coefficients for main effects, two-way, and three-way interaction terms, respectively. Obviously, this high-order interaction model is more complicate than two-way interaction model.

Although high-order interactions play an important role in depicting the nonlinear relationship among variables, sparse representation of high-order interaction model is quite challenging since the number of interaction terms is huge (of order $O(p^2)$ or even $O(p^3)$) if there are p main effects in the model. For instance, even with 10^2 variables in the model, a high-order interaction model has to consider 10^4 or 10^6 variables which make variable selection procedure more difficult. Furthermore, computation cost is another issue because of the huge number of variables in the high-order interaction model. Even worse, severe multicollinearity exists among variables caused by the high dimensionality. There is a large body of literature on sparse representation methods. Subset selection methods based on L_0 type penalty have been discovered for decades. These methods including Akaike information criterion (AIC) [6], Bayesian information criterion (BIC) [7], and mallow's C_p [8] which are NP hard since all the possible combinations of variables are taken into account. Namely, if there are m variables, 2^m subset models have to be checked. To improve the computational efficiency, sparse representation approaches based on convex optimization are proposed. For instance, Tibshirani (1996) proposed the least absolute shrinkage and selection operator (LASSO) which applies L_1 type penalty to pursue sparsity. In wavelet study, it is famous as basis pursuit method [9]. It is more computational efficient than subset selection methods. Efron (2004) studied the least-angle regression (LAR) which solved the same problem as LASSO but in a forward selection manner. They have shown that the solution path of LAR is pretty similar as that of LASSO. Yuan and Lin (2006) investigated group LASSO method which selected group variables based on “all in and all out” principle. To overcome the shortcoming of LASSO and further improve the selection accuracy, Zou (2006) studied adaptive LASSO using weights for each variable to adjust the sparsity degree. Friedman (2012) summarized and provided a review of sparse regression and classification methods [10].

However, convex optimization has drawbacks since it is difficult to discover all the relevant variables if the model coherence is sufficiently high. To overcome the difficulties, nonconvex penalties including smoothly clipped absolute deviation (SCAD) [11], minimax concave penalty (MCP) [12], and hard ridge (HR) [13, 14] are proposed to boost the accuracy. Ye et al. (2017) studied sparse least squares support vector machine with L_p norm function applied to select the relevant support vectors [15]. To facilitate parameter tuning work and increase the forecasting accuracy, Jiang (2018) studied square root nonconvex optimization (SRNO) and showed its superior performance [2] via simulation and real data studies.

Sparse representation in the two-way interaction model has been studied for years. Bach (2008) explored hierarchical multiple learning and studied its selection property [16]. Zhao et al. (2009) proposed composite absolute penalty (CAP) which considers a mixture of L_p and L_q norm functions which are given by $\|\phi_{jk}\|_p + \|(b_j, b_k, \phi_{jk})\|_q$ [17]. Choi et al. (2010) studied SHIM method considering $\phi_{jk} = \rho b_j b_k$. They also designed a nonconvex optimization to perform sparsity representation. However, the number of features is only 10 which is pretty small. Thus, there is no guarantee that their method can work in the high dimensional setup [18]. Bickel et al. (2010) used a two-step algorithm to pursue sparsity. In the first step, LASSO was applied and backward procedure was used in the second step [19]. Wu et al. (2010) proposed a stagewise procedure which screened the main effects in the first step and the selected variable will be cleaned based on t-test statistic [20]. Radchenko and Jame (2010) investigated VANISH constructing the main effects and interactions from two small sets of orthogonal functions [21]. Bien et al. (2013) studied hierarchical LASSO (HL) which considered a L_1 optimization problem using a nonconvex constraint $\|\phi_j\|_1 \leq |b_j|$ [22]. She et al. (2014) studied GRESH method which enforced sparsity to main effects and two-way interactions simultaneously [23]. Hao et al. (2014) screened the main effects using forward regression in the first step and fitted the quadratic model with selected main effects and associated interactions [24]. Simon and Hastie (2015) proposed “Glinternet” to discover the two-way interaction terms between categorical variables and continuous variables [25]. Yan and Bien (2017) advocated VARXL method to implement sparse representation in a high dimensional time series data [26]. To the best of our knowledge, most of variable selection methods mentioned above only consider two-way interaction terms. Few studies have been done on the high-order interaction model with both two-way and three-way interaction terms. To tackle this issue, this paper explores sparse representation in a high-order interaction model. The main contributions of this paper are given as follows:

- (i) A two-stage square root hard ridge method (TSSRHR) for sparse representation in high-order interaction model is proposed.
- (ii) In computation, a fast and simple-to-implement algorithm is designed.
- (iii) In theory, we have shown the prediction error bound and estimation error bound of the TSSRHR estimate.
- (iv) Real data examples are shown to demonstrate the efficiency and superior performance of the proposed method over other existing competitors.

The rest of this paper is organized as follows. Section 2 represents the general framework of TSSRHR. Section 3 designs a fast and simple-to-implement algorithm with theoretical guarantee of its convergence and optimality. The theory part of TSSRHR is given in Section 4. In Section 5, real data analyses are shown to exhibit superior performance of the proposed method over other approaches. The conclusion of this paper is given in Section 6.

2. Two-Stage Square Root Hard Ridge Method

We investigate and study the square root hard ridge (SRHR) method which combines the benefits of square root loss function and nonconvex hard ridge penalty function for high-order interaction selection. We describe the two-stage SRHR algorithm at first. At stage 1, only main effects are selected by SRHR with all of high-order interactions including two-way and three-way interactions kept out. The selected variables are put in the set $\mathcal{A}$. At stage 2, the selected set in stage 1. $\mathcal{A}$ is expanded by adding both associated two-way and three-way interactions within $\mathcal{A}$. SRHR is applied again to implement variable selection task.

- (i) Stage 1. Define $\mathcal{G} = [p] = \{1, \dots, p\}$ which represents all the main effects. Select the important effects using SRHR method by considering the following optimization problem:

$$\min_{b, \phi, \psi} \left(\sum_{i=1}^n \left(y_i - b_0 - \sum_{j \in \mathcal{A}} b_j \mathbf{x}_j - \sum_{(j,k) \in \mathcal{A} \otimes \mathcal{A}} \phi_{jk} \mathbf{x}_j \mathbf{x}_k - \sum_{(j,k,l) \in \mathcal{A} \otimes \mathcal{A} \otimes \mathcal{A}} \psi_{jkl} \mathbf{x}_j \mathbf{x}_k \mathbf{x}_l \right)^2 \right)^{1/2} + \frac{\lambda_2^2}{2(1+\eta_2)} \left(\sum_{j \in \mathcal{A}} 1_{b_j \neq 0} + \sum_{(j,k) \in \mathcal{A} \otimes \mathcal{A}} 1_{\phi_{jk} \neq 0} + \sum_{(j,k,l) \in \mathcal{A} \otimes \mathcal{A} \otimes \mathcal{A}} 1_{\psi_{jkl} \neq 0} \right) + \frac{\eta_2}{2} \left(\sum_{j \in \mathcal{A}} b_j^2 + \sum_{(j,k) \in \mathcal{A} \otimes \mathcal{A}} \phi_{jk}^2 + \sum_{(j,k,l) \in \mathcal{A} \otimes \mathcal{A} \otimes \mathcal{A}} \psi_{jkl}^2 \right) \quad (3)$$

where $1(\cdot)$ denotes the indicator function, $\otimes$ represents the Cartesian product, and λ_2 and η_2 are two regularization parameters which need to be selected carefully. The solution path is provided by $S_{\lambda_2, \eta_2}^{(2)}$ and the selected variable are given in $\mathcal{B}$.

A penalized sparse representation method considers the sum of two components: the loss function and the penalty function. The loss function often applies the negative log-likelihood of the error and in Gaussian setup, it is a squared error loss function which involves the noise level σ which is difficult to estimate. Specifically, in a Gaussian linear regression setting $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, the negative likelihood function of $\mathbf{y}$ is $-\log(f(\mathbf{y}|\mathbf{X}\boldsymbol{\beta})) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 / \sigma^2 + h(\sigma) + c$, where $h(\sigma)$ is a function of σ and c is a constant. SRHR uses square root error loss function which can facilitate the parameter tuning work by avoiding estimating the noise level σ or using other forms of estimates [27]. Hard ridge penalty has two parts; it is obvious that the first part which is L_0 norm penalized the number of nonzero elements using the regularization parameter λ . The second part is L_2 type ridge penalty. It can compensate the shrinkage given by the L_0 norm and is able to handle the high collinearity of the high-order interaction model. Therefore, SRHR combines the advantages of square root loss function and hard ridge penalty.

The choices of regularization parameters (λ_1 , λ_2 , η_1 , and η_2) play an important role in balancing a model's in sample fit and out-of-sample forecasting ability. We grid λ_1 starting

$$\min_{b \in R^p} \left(\sum_{i=1}^n \left(y_i - b_0 - \sum_{j \in [p]} b_j \mathbf{x}_j \right)^2 \right)^{1/2} + \frac{\lambda_1^2}{2(1+\eta_1)} \sum_{j \in [p]} 1_{b_j \neq 0} + \frac{\eta_1}{2} \sum_{j \in [p]} b_j^2 \quad (2)$$

where λ_1 and η_1 are two regularization parameters which are supposed to be determined. The index for selected variables are given in $\mathcal{A}$ and the corresponding solution path is given by $S_{\lambda_1, \eta_1}^{(1)}$.

- (ii) Stage 2. Update $\mathcal{G} = \mathcal{A} \cup \{(j,k); j \in \mathcal{A}, k \in \mathcal{A}\} \cup \{(j,k,l); j \in \mathcal{A}, l \in \mathcal{A}, k \in \mathcal{A}\}$. Select the important main effects, two-way and three-way interaction using SRHR method again by considering the following problem (for notation simplicity, $\mathbf{x}_j \odot \mathbf{x}_k$ and $\mathbf{x}_j \odot \mathbf{x}_k \odot \mathbf{x}_l$ are denoted by $\mathbf{x}_j \mathbf{x}_k$ and $\mathbf{x}_j \mathbf{x}_k \mathbf{x}_l$, respectively):

from the largest value $\lambda_{1,\max} = \max(\mathbf{X}^T \mathbf{y})$ such that all the coefficients are shrunk to zeros. Then let the smallest value $\lambda_{1,\min}$ be a small proportion of $\lambda_{1,\max}$, that is, $\lambda_{1,\min} = 0.05\lambda_{1,\max}$ and the grid values are generated between $\lambda_{1,\max}$ and $\lambda_{1,\min}$ exponentially. For λ_2 , the same strategy is also applied. The optimal values for η_1 and η_2 are obtained from a grid of values $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1e+1\}$.

To choose the optimal regularization parameters, extended Bayesian information criterion (EBIC) [28] and high dimensional Bayesian information criteria (HDBIC) [29] are applied in this paper. Particularly, EBIC is defined as

$$\text{EBIC}(\mathcal{A}) = \frac{1}{n - |\mathcal{A}|} \sum_{i=1}^n \left(y_i - \hat{b}_0 - \sum_{j \in \mathcal{A}} \hat{b}_j \mathbf{x}_j \right)^2 + n^{-1} |\mathcal{A}| (\log n + \log p), \quad (4)$$

and HDBIC is defined as

$$\text{EBIC}(\mathcal{B}) = \frac{1}{n - |\mathcal{B}|} \sum_{i=1}^n \left(y_i - \hat{b}_0 - \sum_{j \in \mathcal{A}} \hat{b}_j \mathbf{x}_j - \sum_{(j,k) \in \mathcal{A} \otimes \mathcal{A}} \hat{\phi}_{jk} \mathbf{x}_j \mathbf{x}_k - \sum_{(j,k,l) \in \mathcal{A} \otimes \mathcal{A} \otimes \mathcal{A}} \hat{\psi}_{jkl} \mathbf{x}_j \mathbf{x}_k \mathbf{x}_l \right)^2 + n^{-1} |\mathcal{B}| (\log n + \log(|\mathcal{A}| + |\mathcal{A}|^2 + |\mathcal{A}|^3)) \quad (5)$$

where $|\mathcal{A}|$ and $|\mathcal{B}|$ are represent the cardinalities of $\mathcal{A}$ and $\mathcal{B}$. In the first stage, λ_1 and η_1 will be selected by the EBIC since large number of main effects are considered. It worth mentioning that different from [24], we did not apply BIC in the second stage. Although the number of variables is reduced dramatically, we still use HDBIC which is a high dimensional version of BIC because both two-way and three-way interaction terms are considered which still cause the dimensions of the model to be high.

3. Algorithm

In this section, for notation simplicity, two design matrices are designed as follows:

$$\bar{\mathbf{X}} = [\mathbf{x}_1^2, \dots, \mathbf{x}_1 \mathbf{x}_p, \mathbf{x}_p \mathbf{x}_1, \dots, \mathbf{x}_p^2] \in R^{n \times p^2}, \quad (6)$$

and

$$\tilde{\mathbf{X}} = [\mathbf{x}_1^3, \mathbf{x}_1^2 \mathbf{x}_2, \dots, \mathbf{x}_1^2 \mathbf{x}_p, \dots, \mathbf{x}_p^2 \mathbf{x}_1, \dots, \mathbf{x}_p^3] \in R^{n \times p^3} \quad (7)$$

Furthermore, let $\tilde{\mathbf{X}} = [\mathbf{X}, \bar{\mathbf{X}}, \tilde{\mathbf{X}}]$. Clearly, $\bar{\mathbf{X}}$ represents the design matrix for two-way interaction terms, $\tilde{\mathbf{X}}$ denotes the design matrix for three-way interaction terms, and $\tilde{\mathbf{X}}$ is the overall design matrix for main effects, two-way interaction terms, and three-way interaction terms. For the coefficients to be estimated, let Φ and Ψ be the coefficient matrix for two-way and three-way interaction terms, respectively. Then (1) can be written as

$$\mathbf{y} = b_0 + \mathbf{X}\mathbf{b} + \bar{\mathbf{X}}\text{vec}(\Phi) + \tilde{\mathbf{X}}\text{vec}(\Psi) + \varepsilon \quad (8)$$

where $\text{vec}(\mathbf{A}) = [a_{11}, a_{12}, \dots, a_{nm}]$ for any $\mathbf{A} = [a_{ij}]_{n \times m}$. Define $\Omega = [\mathbf{b}, \text{vec}(\Phi)^T, \text{vec}(\Psi)^T]^T$. To design an algorithm for TSSRHR, we consider the following optimization problem in the first stage:

$$\min_{\mathbf{b}} \quad \|\mathbf{y} - \mathbf{X}\mathbf{b}\|_2 + \frac{\lambda_1^2}{2(1+\eta_1)} \|\mathbf{b}\|_0 + \frac{\eta_1}{2} \|\mathbf{b}\|_2^2 \quad (9)$$

and in the second stage, we consider

$$\min_{\Omega} \quad \|\mathbf{y} - \tilde{\mathbf{X}}\Omega\|_2 + \frac{\lambda_2^2}{2(1+\eta_2)} \|\Omega\|_0 + \frac{\eta_2}{2} \|\Omega\|_2^2 \quad (10)$$

To solve (9) and (10) accurately and efficiently, threshold rule is defined at first.

Definition (threshold rule). A threshold rule is a real-valued function $\Theta(t; \lambda)$ defined for $-\infty < t < \infty$ and $0 \leq \lambda < \infty$ such that

- (1) $\Theta(-t; \lambda) = -\Theta(t; \lambda)$;
- (2) $\Theta(t; \lambda) \leq -\Theta(t'; \lambda)$ for $t \leq t'$;
- (3) $\lim_{t \rightarrow +\infty} \Theta(t; \lambda) = \infty$; and
- (4) $0 \leq \Theta(t; \lambda) \leq t$ for $0 \leq t \leq \infty$.

From the definition, it is easy to see that $\Theta(t; \lambda)$ is an odd monotone unbounded shrinkage rule for t , at any λ . A vector

version of Θ is defined component-wise if either t or λ is replaced by a vector. Now define the HR thresholding rule, which is closely associated with the HR penalty, as

$$\Theta_{HR}(t; \lambda, \eta) = \frac{t}{1+\eta} 1_{|t| \geq \lambda} \quad (11)$$

where $\eta > 0$, $\lambda > 0$ are two regularization parameters. Using HR thresholding function, (9) and (10) can be solved based on the following iterations:

$$\begin{aligned} \text{The 1}^{\text{st}} \text{ stage: } \mathbf{b}^{(j+1)} &= \vec{\Theta}_{HR}(\mathbf{b}^{(j)} \\ &+ \mathbf{X}^T(\mathbf{X}\mathbf{b}^{(j)} - \mathbf{y}); \lambda_1 \|\mathbf{y} - \mathbf{X}\mathbf{b}^{(j)}\|_2, \eta_1 \|\mathbf{y} - \mathbf{X}\mathbf{b}^{(j)}\|_2) \end{aligned} \quad (12)$$

and

$$\begin{aligned} \text{The 2}^{\text{nd}} \text{ stage: } \Omega^{(k+1)} &= \vec{\Theta}_{HR}(\Omega^{(k)} \\ &+ \tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\Omega^{(k)} - \mathbf{y}); \lambda_2 \|\mathbf{y} - \tilde{\mathbf{X}}\Omega^{(k)}\|_2, \\ &\eta_2 \|\mathbf{y} - \tilde{\mathbf{X}}\Omega^{(k)}\|_2) \end{aligned} \quad (13)$$

where $\vec{\Theta}_{HR}$ is a vector version of HR thresholding function with t replaced by a vector.

Algorithm 1 exhibits the details of TSSRHR algorithm with $\omega^{(-1)} = 0$, $\omega^{(0)} = 1$, $b^{(0)} = b^{(-1)}$, and $\Omega^{(0)} = \Omega^{(-1)}$ defined. Consider the nonconvex optimization is not easy to solve using ADMM algorithm [30]. HR thresholding function is applied in the algorithm. The basic structure of this algorithm follows coordinate decent algorithm. The error tolerance of tol_1 , tol_2 and maximum number of iterations m_1 and m_2 are determined based on trial and error. Usually, the default values of tol_1 , tol_2 and m_1 and m_2 are $1e-4$, $1e-4$, 500, and 500, respectively. TSSRHR algorithm has some advantages. Firstly, it does not involve any matrix inversion. Secondly, $\Theta_{HR}(t; \lambda, \eta)$ is applied to enforce elementwise sparsity. The step sizes τ_1 and τ_2 are able to guarantee the convergence of the algorithm. The theoretical results show that $\tau_1 \geq \|\mathbf{X}\|_2$ and $\tau_2 \geq \|\tilde{\mathbf{X}}\|_2$ provide satisfactory results. The step sizes do not depend on line search which takes a lot of computation time. The TSSRHR algorithm is simple to implement and computational efficient. To further speed up the convergence, a fast iterative shrinkage thresholding algorithm (FISTA) [31] is used to obtain the TSSRHR estimate. The following theorem provides the theoretical guarantee of convergence of Algorithm 1.

4. Prediction and Estimation

Notice that $P(\Omega) = \lambda^2 \|\Omega\|_0 / 2(1+\eta) + \eta \|\Omega\|_2^2 / 2$ does not follow the triangle inequality $P(\mathbf{a} + \mathbf{b}) \leq P(\mathbf{a}) + P(\mathbf{b})$ but follows the following inequality $P(\mathbf{a} + \mathbf{b}) \leq 2(P(\mathbf{a}) + P(\mathbf{b}))$ for any vectors $\mathbf{a}$, $\mathbf{b}$. This poses critically challenges to the derivation the forecasting and estimation error bounds. Namely, the ways of performing error rate analysis on square root lasso [27] are not applicable to SRHR because of its nonconvexity. Therefore, a novel approach is urgently needed to derive the forecasting and estimation error bounds.

Inputs: $\mathbf{Z} = (\mathbf{X}, \mathbf{y}) = \{(\tilde{\mathbf{x}}_1, \mathbf{y}_1), \dots, (\tilde{\mathbf{x}}_N, \mathbf{y}_N)\} : \tilde{\mathbf{x}}_k \in R^p, \mathbf{X} \in R^{n \times p}, \mathbf{y} \in R^n$ the dataset
 m_1 : maximum number of iterations used in the SRHR algorithm of the first stage;
 m_2 : maximum number of iterations used in the SRHR algorithm of the second stage;
 tol_1 : the error tolerance used in the SRHR algorithm of the first stage;
 tol_2 : the error tolerance used in the SRHR algorithm of the second stage;
 τ_1 : scale parameter used in the SRHR algorithm of the first stage;
 τ_2 : scale parameter used in the SRHR algorithm of the second stage;
Output: The forecasting test error and selected pattern;
(1) Randomly divide the original data $\mathbf{Z}$ into the training dataset $(\mathbf{X}_{train}, \mathbf{y}_{train})$ and test dataset $(\mathbf{X}_{tst}, \mathbf{y}_{tst})$.
The first stage using SRHR algorithm:
(2) Generate grid values of $\lambda_1^G = \{\lambda_{1,u}\}_{u=1}^{s_1}$ and $\eta_1^G = \{\eta_{1,v}\}_{v=1}^{t_1}$.
(3) **for** $u = 1$ **to** s_1
(4) **for** $v = 1$ **to** t_1
(5) Initialization: $j \leftarrow 0, \mathbf{b}^{(j)} \leftarrow \mathbf{0}$
(6) Scaling: $\mathbf{X}_{train} \leftarrow \mathbf{X}_{train}/\tau_1, \mathbf{y}_{train} \leftarrow \mathbf{y}_{train}/\tau_1$
(7) **while** $\|\mathbf{b}^{(j+1)} - \mathbf{b}^{(j)}\|_2 < tol_1$ or $j > m_1$ **do**
(8) Step 1. $\xi_1^{(j)} \leftarrow \mathbf{b}^{(j)} + (\omega^{(j-1)}/\omega^{(j)})(\mathbf{b}^{(j)} - \mathbf{b}^{(j-1)})$
(9) Step 2. $\zeta_1^{(j)} \leftarrow \xi_2^{(j)} + \mathbf{X}_{train}^T(\mathbf{y}_{train} - \mathbf{X}_{train}\xi_1^{(j)})$
(10) Step 3. $\mathbf{b}^{(j+1)} \leftarrow \vec{\Theta}_{HR}(\zeta_1^{(j)}; \lambda_{1,u}\|\mathbf{y} - \tilde{\mathbf{X}}\mathbf{b}^{(j)}\|_2/\tau_1^2, \eta_{1,v}\|\mathbf{y} - \tilde{\mathbf{X}}\mathbf{b}^{(j)}\|_2/\tau_1^2)$
(11) Step 4: $\omega^{(j+1)} = (1 + \sqrt{1 + 4\omega^{(j)^2}})/2$
(12) **end while**
(13) **end for**
(14) **end for**
(15) Obtain the solution path $\mathcal{A}$ and the corresponding sparsity patterns based on EBIC criterion.
The second stage using SRHR algorithm:
(16) Generate the high-order interaction model $(\tilde{\mathbf{X}}_{train}, \mathbf{y}_{train})$ and $(\tilde{\mathbf{X}}_{tst}, \mathbf{y}_{tst})$ based on the sparsity pattern $\mathcal{A}$.
(17) Generate grid values of $\lambda_2^G = \{\lambda_{2,u}\}_{u=1}^{s_2}$ and $\eta_2^G = \{\eta_{2,v}\}_{v=1}^{t_2}$.
(18) **for** $u = 1$ **to** s_2
(19) **for** $v = 1$ **to** t_2
(20) Initialization: $k \leftarrow 0, \Omega^{(k)} \leftarrow \mathbf{0}$
(21) Scaling: $\tilde{\mathbf{X}}_{train} \leftarrow \tilde{\mathbf{X}}_{train}/\tau_2, \mathbf{y}_{train} \leftarrow \mathbf{y}_{train}/\tau_2$
(22) **while** $\|\Omega^{(k+1)} - \Omega^{(k)}\|_2 < tol_2$ or $k > m_2$ **do**
(23) Step 1. $\xi_2^{(k)} \leftarrow \Omega^{(k)} + (\omega^{(k-1)}/\omega^{(k)})(\Omega^{(k)} - \Omega^{(k-1)})$
(24) Step 2. $\zeta_2^{(k)} \leftarrow \xi_2^{(k)} + \tilde{\mathbf{X}}_{train}^T(\mathbf{y}_{train} - \tilde{\mathbf{X}}_{train}\xi_2^{(k)})$
(25) Step 3. $\Omega^{(k+1)} \leftarrow \vec{\Theta}_{HR}(\zeta_2^{(k)}; \lambda_{2,u}\|\mathbf{y} - \tilde{\mathbf{X}}\Omega^{(k)}\|_2/\tau_2^2, \eta_{2,v}\|\mathbf{y} - \tilde{\mathbf{X}}\Omega^{(k)}\|_2/\tau_2^2)$
(26) Step 4: $\omega^{(k+1)} = (1 + \sqrt{1 + 4\omega^{(k)^2}})/2$
(27) **end while**
(28) **end for**
(29) **end for**
(30) Obtain the solution path $\mathcal{B}$ and update the sparsity patterns $\mathcal{G}$ using HDBIC criterion.
(31) Calculate the test error using test dataset $(\tilde{\mathbf{X}}_{tst}, \mathbf{y}_{tst})$

ALGORITHM 1: The TSRRHR algorithm.

Assumption 1 (overfitting assumption). Suppose that $\|\mathbf{y} - \tilde{\mathbf{X}}\hat{\Omega}\|_2 = 0$, then $\hat{\Omega}$ does the tasks as the Ordinary Least Square (OLS) estimator $\hat{\Omega}_{OLS} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y}$.

Assumption 2 (weak decomposability assumption). A norm M in R^p is called weakly decomposable for an index set $\mathcal{J} \subset [p]$, if there exists a norm $M^{\mathcal{J}^c}$ on R^{p-J} such that

$$M(\Omega) \geq M(\Omega_{\mathcal{J}}) + M^{\mathcal{J}^c}(\Omega_{\mathcal{J}^c}) \quad \forall \Omega \in R^p \quad (14)$$

Furthermore, a set $\mathcal{J}$ is called allowed set if M is a weakly decomposable norm for this set.

We can show that hard ridge penalty is weakly decomposable as follows: define $M(\Omega) := \lambda^2 \|\Omega\|_0/2(1 + \eta) + \eta \|\Omega\|_2^2$ and $M^{\mathcal{J}^c}(\Omega) := \lambda^2 \|\Omega\|_0/2(1 + \eta) + \eta \|\Omega\|_2^2$. Then we can obtain weak decomposability for any set $\mathcal{J}$

$$M(\Omega_{\mathcal{J}}) + M^{\mathcal{J}^c}(\Omega_{\mathcal{J}^c}) \leq M(\Omega) \quad (15)$$

This is due to the weak decomposability property of the penalty and the penalty.

Definition 3 (M -effective sparsity). For an allowed set of a norm M , and $L > 0$ a constant, the M eigenvalue is defined as

$$\delta_M(L, \mathcal{F}) := \min \{ \|\tilde{\mathbf{X}}\boldsymbol{\Omega}_{\mathcal{F}} - \tilde{\mathbf{X}}\boldsymbol{\Omega}_{\mathcal{F}^c}\|_2 : M^{\mathcal{F}^c}(\boldsymbol{\Omega}_{\mathcal{F}^c}) \leq LM(\boldsymbol{\Omega}_{\mathcal{F}}) \} \quad (16)$$

Then the M -effective sparsity is defined as

$$\Gamma_M^2(L, \mathcal{F}) := \frac{1}{\delta_M^2(L, \mathcal{F})} \quad (17)$$

M -effective sparsity is a variant of regularity condition used to establish the nonasymptotic oracle inequality, such as Restricted Eigenvalue condition [32], Mutual Coherence condition [33], Compatible Condition [34], Cone invertibility factor [35], and Restricted invertibility factor [36]. With Assumptions 1 and 2 satisfied, based on the Theorem 1 in [37], we can obtain the forecasting and estimation error bounds of the TSSRHR estimator in the following remark.

Remark. Choose

$$\lambda = C_1 \sqrt{\frac{2}{n}} \left(t\delta + 2 + \sqrt{\log(p - J)} \right) \quad (18)$$

and

$$\eta = C_2 \sqrt{\frac{2}{n}} \left(t\delta + 2 + \sqrt{\log(p - J)} \right) \quad (19)$$

where $t = \sqrt{\log(2/\alpha)}$ with $\alpha > 2/e^{n/4}$ and $\delta = \sqrt{1 - 2t\sqrt{2/n}}$. Under the overfitting assumption and weak decomposability for $M(\cdot) := \lambda^2 \|\cdot\|_0/2(1+\eta) + \eta \|\cdot\|_2$, the following inequalities hold with probability $1 - \alpha$

$$\|\tilde{\mathbf{X}}\hat{\boldsymbol{\Omega}} - \tilde{\mathbf{X}}\boldsymbol{\Omega}\|_2 \leq C_3 \lambda^2 \Gamma_M^2(L, \mathcal{F}^*) \quad (20)$$

and

$$\|\hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}\|_2^2 \leq C_4 \lambda^2 \Gamma_M^2(L, \mathcal{F}^*) \quad (21)$$

where C_1, C_2, C_3 , and C_4 are some positive constants.

This remark provides prediction error bound and estimation error bound for the TSSRHR estimate. The proof of remark follows Theorem 1 and Corollaries 1 and 2 in [37] directly. Thus, the details are omitted here.

5. Real World Data

In this section, twinsine signal data including 8 scenarios and microarray gene expression data about inbred mouse are treated as real data examples to test the performances of sparse representation methods. It worth mentioning both

of these two real data examples are ultra-high dimensional data when three-way interaction terms are considered. Comparisons will be made between the proposed two-stage hard ridge (TSSRHR) and other two-stage sparse representation methods including two-stage LASSO (TSLASSO), two-stage SCAD (TSSCAD), and two-stage square root lasso (TSSRL) in terms of forecasting accuracy, sparsity, and computational efficiency. The unknown parameters in TSLASSO and TSSCAD are determined using 10-fold cross validation. TSSRL selects the unknown parameter based on a theoretical choice: $\lambda = 1.1G^{-1}(1 - 0.05/2K)$ where K is the number of variables and G is the cumulative distribution function of Gaussian distribution. Here we remind that TSSRHR selects the regularization parameters by EBIC and HDBIC (cf. Section 2). The forecasting model will be trained with 75% of the samples and the remaining 25% will be used to calculate the test errors. The entire procedure will be repeated for 150 times and the median of evaluation criteria including number of nonzero elements (NZ), mean square error (MSE), root mean square error (RMSE), and computational cost (CC) will be reported.

5.1. Twinsine Signal Data. Spectral estimation studying the distribution over frequencies is widely applied in speech coding and radar solar signal processing. Nevertheless, it is critically difficult and challenging when there are a large number of frequencies which causes high resolution problem. This type of problem is called super resolution spectral estimation. To avoid the power leakage and noisy observation caused by the super resolution, a novel method is urgently needed to choose the important frequencies which plays a dominant role in super-resolution estimation. The twinsine signal is generated from the target function:

$$y(t) = A_1 \cos(2\pi f_1 t + \phi_1) + A_2 \cos(2\pi f_2 t + \phi_2) + \varepsilon(t) \quad (22)$$

where $A_1 = 2$, $A_2 = 3$, $\phi_1 = \pi/3$, $f_1 = 0.25\text{HZ}$, and $\varepsilon(t)$ follows white Gaussian noise with variance given by σ^2 . The frequency resolution is chosen as 0.002HZ to detect and distinguish the two sinusoidal components. $f_k = f_{\max} \cdot k/D$ for $k = 0, 1, \dots, D$ and the frequency atoms are given by $\mathbf{X}_{\cos} = [\cos(2\pi t_i f_k)]_{1 \leq i \leq n, 1 \leq k \leq D}$ and $\mathbf{X}_{\sin} = [\sin(2\pi t_i f_k)]_{1 \leq i \leq n, 1 \leq k \leq D}$. In our experiments, we will consider the following 8 different scenarios:

- (i) Scenario 1: $(n, D, \sigma^2) = (100, 300, 1)$
- (ii) Scenario 2: $(n, D, \sigma^2) = (100, 500, 1)$
- (iii) Scenario 3: $(n, D, \sigma^2) = (100, 1000, 1)$
- (iv) Scenario 4: $(n, D, \sigma^2) = (100, 5000, 1)$
- (v) Scenario 5: $(n, D, \sigma^2) = (100, 300, 5)$
- (vi) Scenario 6: $(n, D, \sigma^2) = (100, 500, 5)$
- (vii) Scenario 7: $(n, D, \sigma^2) = (100, 1000, 5)$
- (viii) Scenario 8: $(n, D, \sigma^2) = (100, 5000, 5)$

FIGURE 1: Boxplots of compared forecasting methods in Scenarios 1-10.

(ix) Scenario 9: $(n, D, \sigma^2) = (500, 300, 1)$

(x) Scenario 10: $(n, D, \sigma^2) = (1000, 300, 1)$

The results of forecasting, variable selection, and computation are summarized in Table 1 and Figures 1, 2, 3, and 4. It is not difficult to observe that TSSRHR obtains the most accurate outcome in all the scenarios including the most challenging scenario when $D = 5000$ and $\sigma^2 = 5$. TSSRL outperforms TSLASSO and TSSCAD method by delivering lower MSE and its forecasting results are comparable with TSSRHR in scenario 6. From the prospective of model sparsity, TSSRL achieves the most interpretable model followed by TSSRHR. TSLASSO and TSSCAD tend to select a large number of variables which can be found in all the scenarios except scenario 5 and scenario 6. For instance, TSLASSO selects 31458 variables and TSSCAD chooses 29113 variables in scenario 8. A large number of nuisance variables are included in the models given by TSLASSO and TSSCAD which will affect the forecasting results negatively. In terms of computational efficiency, TSSCAD spends more

computational time in scenarios 1-2 and scenarios 5-6 when the dimensions are moderate. As the dimensions increase, all of the forecasting approaches take more computation cost. For instance, TSSRL took 1215.93 seconds and 1529.38 seconds in scenarios 4 and 8 when $D = 5000$. TSLASSO runs fastest by using the convex penalty. However, notice that when D is large, although TSSRL also uses the convex penalty, it is not computational efficient since the design of the TSSRL is based on COORD algorithm which runs much slower than coordinate descent algorithm. In scenario 9 and 10 where the sample sizes are large, TSSRHR still performs better than other methods. Same as other scenarios, TSSRL gives the most sparse model with fewer variables. Also notice that the train speed of TSSRHR is acceptable. Therefore, TSSRHR outperforms other forecasting methods in terms of forecasting accuracy, model sparsity, and computation speed.

5.2. Microarray Gene Expression Data. The inbred mouse microarray gene expression dataset with 31 female mice and

TABLE 1: The statistical performance of TSSRHR, TSLASSO, TSSCAD, and TSSRL. The performances are measured by number of nonzero elements of main effects, two-way interactions, three-way interactions, MSE ($\times 100$), RMSE ($\times 100$), and computational costs (CC, in seconds, unless otherwise specified).

	Nz(b)	Nz(Φ)	Nz(Ψ)	MSE	RMSE	CC	Nz(b)	Nz(Φ)	Nz(Ψ)	MSE	RMSE	CC
Scenario 1												
TSSRHR	3.00	1.00	3.00	93.47	96.68	10.62	TSSRHR	3.00	1.00	5.00	82.71	91.13
TSLASSO	5.00	59.50	3505.50	145.72	119.13	6.94	TSLASSO	7.00	46.00	3596.00	156.48	124.12
TSSCAD	2.00	1.00	8.00	127.40	112.92	18.77	TSSCAD	14.00	21.50	3678.00	185.49	132.81
TSSRL	3.00	0.00	1.00	102.71	101.36	5.26	TSSRL	2.00	0.00	4.00	86.78	91.98
Scenario 3												
TSSRHR	2.00	1.00	6.00	80.21	89.38	11.72	TSSRHR	3.00	0.00	0.00	87.92	93.13
TSLASSO	12.00	20.00	4796.00	137.54	117.13	4.32	TSLASSO	35.00	0.00	31458.00	264.13	161.81
TSSCAD	4.00	0.00	15.00	101.92	100.83	19.72	TSSCAD	19.00	0.00	6661.00	237.35	152.51
TSSRL	2.00	0.00	1.00	88.86	93.53	25.43	TSSRL	2.00	0.00	1.00	96.92	97.43
Scenario 5												
TSSRHR	3.00	0.00	4.00	413.85	200.94	4.21	TSSRHR	3.00	0.00	4.00	418.72	205.19
TSLASSO	6.00	7.00	94.00	533.48	228.12	3.98	TSLASSO	5.00	2.00	50.50	438.59	207.14
TSSCAD	3.00	2.50	6.00	470.83	215.94	20.15	TSSCAD	6.00	1.00	94.00	473.21	216.83
TSSRL	3.00	0.00	1.00	468.10	215.18	2.13	TSSRL	2.00	0.00	3.00	428.13	206.89
Scenario 7												
TSSRHR	2.00	0.00	3.00	405.13	201.98	10.98	TSSRHR	3.00	0.00	2.00	407.95	201.34
TSLASSO	5.00	0.00	77.50	436.82	207.92	4.35	TSLASSO	16.00	0.00	1195.50	870.94	294.48
TSSCAD	9.00	28.00	451.50	434.12	206.15	30.95	TSSCAD	28.00	0.00	29113	454.79	211.83
TSSRL	2.00	0.00	1.00	430.93	205.73	3.43	TSSRL	2.00	0.00	1.00	445.23	209.89
Scenario 8												
TSSRHR	2.00	0.00	3.00	405.13	201.98	10.98	TSSRHR	3.00	0.00	2.00	407.95	201.34
TSLASSO	5.00	0.00	77.50	436.82	207.92	4.35	TSLASSO	16.00	0.00	1195.50	870.94	294.48
TSSCAD	9.00	28.00	451.50	434.12	206.15	30.95	TSSCAD	28.00	0.00	29113	454.79	211.83
TSSRL	2.00	0.00	1.00	430.93	205.73	3.43	TSSRL	2.00	0.00	1.00	445.23	209.89
Scenario 10												
TSSRHR	3.00	1.00	4.00	147.33	121.38	14.03	TSSRHR	3.00	1.00	3.00	142.62	119.42
TSLASSO	3.00	1.00	12.00	194.07	139.31	4.09	TSLASSO	4.00	0.00	19.00	183.93	135.62
TSSCAD	3.00	4.00	18.00	195.77	139.92	8.73	TSSCAD	3.00	1.00	15.00	165.67	128.71
TSSRL	3.00	0.00	3.00	183.71	135.54	2.84	TSSRL	3.00	0.00	3.00	174.79	132.21

FIGURE 2: Sparsity pattern of compared forecasting methods in Scenarios 1-4.

FIGURE 3: Sparsity pattern of compared forecasting methods in Scenarios 5-8.

TABLE 2: The statistical performances of TSSRHR, TSLASSO, TSSCAD, and TSSRL. The performances are measured by number of nonzero elements of main effects, two-way interaction, three-way interactions, three-way interactions, MSE ($\times 100$), RMSE ($\times 100$), and computational costs (CC, in seconds, unless otherwise specified).

	Nz($\mathbf{b}$)	Nz(Φ)	Nz(Ψ)	MSE	RMSE	CC
TSSRHR	1.00	1.00	7.00	8.61	29.33	0.48
TSLASSO	1.00	63.00	6311.00	136.66	116.89	0.22
TSSCAD	1.00	1.00	1.00	142.75	119.48	0.24
TSSRL	0.00	0.00	14.00	10.98	33.13	1.50

FIGURE 4: Sparsity pattern of compared forecasting methods in Scenarios 9-10.

29 male mice is considered to demonstrate the effectiveness of the proposed TSSRHR method. The gene expression values of 22,690 genes are measured by each array and the response gene is a continuous phenotypic variable measured by stearoyl-coenzyme desaturase (*scd1*) with probe set ID 1415965 at. The remaining 22,689 probe set IDs are regarded as candidate variables. We are seeking a good model which is able to describe the relationship between the response variable and other candidate genes. The performances of forecast models on inbred mouse microarray gene expression data are shown in Table 2. It worth mentioning that a gene filtering preprocessing procedure is applied to the inbred mouse data to reduce the expensive computational cost. TSSRHR delivers the best forecasting performances by giving the lowest MSE and RMSE followed by TSSRL, TSLASSO, and TSSCAD. TSLASSO selects a model with 6311 three-way interaction terms while TSSCAD only selects 3 variables which is fewer than those of TSSRL and TSSRHR. This can be verified in Figure 5 which shows the selection pattern of the compared methods. The computational cost of TSSRL is almost 4 times of TSSRHR. Comparing with TSSRHR and TSSCAD, TSLASSO and TSCAD spend less computational cost. Overall, TSSRHR provides the lowest forecasting error with sufficient model sparsity and acceptable computational time.

6. Conclusion

Two-way interaction model has been widely used in numbers of scientific fields. However, researchers find that three-way interaction terms play more important role than two-way interaction terms when establishing forecasting model. To

this end, this paper studies high-order interaction model which has both two-way and three-way interaction terms. To establish an interpretable model and reduce the computation time, sparse representation method using a two-stage square root hard ridge (TSSRHR) is proposed and studied. Based on the property of two-stage method, the number of interactions will decrease dramatically so that the computational issue is solved. To implement SRHR in each stage, a simple and efficient algorithm is designed. Furthermore, the prediction and estimation error bounds of TSSRHR are also exhibited using two regularity assumptions. For real data analysis, a twinsine signal data and microarray gene expression data are provided to show the forecasting and selection performances of TSSRHR comparing with other stagewise sparse representation methods. A hierarchical variable selection problem on the high-order interaction model will be conducted for the future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grants no. 71861012 and no. 71761016),

FIGURE 5: Sparsity pattern of compared forecasting methods in the inbred mouse gene expression data.

the Natural Science Foundation of Jiangxi, China (Grants no. 20181BAB211020 and no. 20171BAA218001), and the China Postdoctoral Science Foundation (Grants no. 2017M620277 and no. 2018T110654).

References

- [1] J. Fan and R. Li, "Statistical challenges with high dimensionality: Feature selection in knowledge discovery," *Proceedings of the International Congress of Mathematicians*, pp. 595–622, 2006.
- [2] H. Jiang, "Sparse estimation based on square root nonconvex optimization in high-dimensional data," *Neurocomputing*, vol. 282, pp. 122–135, 2018.
- [3] H. Jiang, "Model forecasting based on two-stage feature selection procedure using orthogonal greedy algorithm," *Applied Soft Computing*, vol. 63, pp. 110–123, 2017.
- [4] H. Schwender and K. Ickstadt, "Identification of SNP interactions using logic regression," *Biostatistics*, vol. 9, no. 1, pp. 187–198, 2008.
- [5] E. Assary, J. P. Vincent, R. Keers, and M. Pluess, "Gene-environment interaction and psychiatric disorders: Review and future directions," *Seminars in Cell & Developmental Biology*, vol. 77, pp. 133–143, 2017.
- [6] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, 1974.
- [7] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [8] C. L. Mallows, "Some comments on C_p ," *Technometrics*, vol. 42, no. 1, pp. 87–94, 2000.
- [9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
- [10] J. H. Friedman, "Fast sparse regression and classification," *International Journal of Forecasting*, vol. 28, no. 3, pp. 722–738, 2012.
- [11] A. Antoniadis, "Wavelets in statistics: a review," *Statistical Methods & Applications*, vol. 6, no. 2, pp. 97–130, 1997.
- [12] C. H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.
- [13] Y. She, J. Wang, H. Li, and D. Wu, "Group iterative spectrum thresholding for super-resolution sparse spectral selection," *IEEE Transactions on Signal Processing*, vol. 61, no. 24, pp. 6371–6386, 2013.
- [14] H. Jiang and Y. Dong, "Dimension reduction based on a penalized kernel support vector machine model," *Knowledge-Based Systems*, vol. 138, pp. 79–90, 2017.
- [15] Y.-F. Ye, Y.-H. Shao, N.-Y. Deng, C.-N. Li, and X.-Y. Hua, "Robust l_p -norm least squares support vector regression with feature selection," *Applied Mathematics and Computation*, vol. 305, pp. 32–52, 2017.
- [16] F. Bach, "Exploring large feature spaces with hierarchical multiple kernel learning," in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems, NIPS 2008*, pp. 105–112, Canada, December 2008.
- [17] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6, pp. 3468–3497, 2009.
- [18] N. H. Choi, W. Li, and J. Zhu, "Variable selection with the strong heredity constraint and its oracle property," *Journal of the American Statistical Association*, vol. 105, no. 489, pp. 354–364, 2010.
- [19] P. J. Bickel, Y. Ritov, A. B. Tsybakov et al., "Hierarchical selection of variables in sparse high-dimensional regression," in *Borrowing Strength: Theory Powering Applications—a Festschrift for Lawrence D. Brown*, pp. 56–59, Institute of Mathematical Statistics, 2010.

- [20] J. Wu, B. Devlin, S. Ringquist, M. Trucco, and K. Roeder, "Screen and clean: a tool for identifying interactions in genome-wide association studies," *Genetic Epidemiology*, vol. 34, no. 3, pp. 275–285, 2010.
- [21] P. Radchenko and G. M. James, "Variable selection using adaptive nonlinear interaction structures in high dimensions," *Journal of the American Statistical Association*, vol. 105, no. 492, pp. 1541–1553, 2010.
- [22] J. Bien, J. Taylor, and R. Tibshirani, "A lasso for hierarchical interactions," *The Annals of Statistics*, vol. 41, no. 3, pp. 1111–1141, 2013.
- [23] Y. She, Z. Wang, and H. Jiang, "Group regularized estimation under structural hierarchy," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 445–454, 2018.
- [24] N. Hao and H. H. Zhang, "Interaction screening for ultrahigh-dimensional data," *Journal of the American Statistical Association*, vol. 109, no. 507, pp. 1285–1301, 2014.
- [25] M. Lim and T. Hastie, "Learning interactions via hierarchical group-lasso regularization," *Journal of Computational and Graphical Statistics*, vol. 24, no. 3, pp. 627–654, 2015.
- [26] X. Yan and J. Bien, "Hierarchical sparse modeling: a choice of two group lasso formulations," *Statistical Science. A Review Journal of the Institute of Mathematical Statistics*, vol. 32, no. 4, pp. 531–560, 2017.
- [27] A. Belloni, V. Chernozhukov, and L. Wang, "Square-root lasso: pivotal recovery of sparse signals via conic programming," *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.
- [28] J. Chen and Z. Chen, "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, vol. 95, no. 3, pp. 759–771, 2008.
- [29] C.-K. Ing and T. L. Lai, "A stepwise regression method and consistent model selection for high-dimensional sparse linear models," *Statistica Sinica*, vol. 21, no. 4, pp. 1473–1513, 2011.
- [30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [31] Y. Nesterov, "Gradient methods for minimizing composite objective function," Tech. Rep., Université catholique de Louvain, Center for Operations Research and Econometrics, 2007.
- [32] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, "Simultaneous analysis of lasso and Dantzig selector," *The Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.
- [33] K. Lounici, "Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators," *Electronic Journal of Statistics*, vol. 2, pp. 90–102, 2008.
- [34] S. A. van de Geer and P. Bühlmann, "On the conditions used to prove oracle results for the Lasso," *Electronic Journal of Statistics*, vol. 3, pp. 1360–1392, 2009.
- [35] F. Ye and C.-H. Zhang, "Rate minimaxity of the lasso and dantzig selector for the l_q loss in l_r balls," *The Journal of Machine Learning Research*, vol. 11, pp. 3519–3540, 2010.
- [36] C.-H. Zhang and T. Zhang, "A general theory of concave regularization for high-dimensional sparse estimation problems," *Statistical Science*, vol. 27, no. 4, pp. 576–593, 2012.
- [37] B. Stucky and S. van de Geer, "Sharp oracle inequalities for square root regularization," *Journal of Machine Learning Research*, vol. 18, no. 67, pp. 1–29, 2017.