

# An Iterative Algorithm for Fitting Nonconvex Penalized Generalized Linear Models with Grouped Predictors

Yiyuan She

Department of Statistics

Florida State University, FL 32306-4330, yshe@stat.fsu.edu

## Abstract

High-dimensional data pose challenges in statistical learning and modeling. Sometimes the predictors can be naturally grouped where pursuing the between-group sparsity is desired. Collinearity may occur in real-world high-dimensional applications where the popular  $l_1$  technique suffers from both selection inconsistency and prediction inaccuracy. Moreover, the problems of interest often go beyond Gaussian models. To meet these challenges, nonconvex penalized generalized linear models with grouped predictors are investigated and a simple-to-implement algorithm is proposed for computation. A rigorous theoretical result guarantees its convergence and provides tight preliminary scaling. This framework allows for grouped predictors and nonconvex penalties, including the discrete  $l_0$  and the ' $l_0 + l_2$ ' type penalties. Penalty design and parameter tuning for nonconvex penalties are examined. Applications of super-resolution spectrum estimation in signal processing and cancer classification with joint gene selection in bioinformatics show the performance improvement by nonconvex penalized estimation.

## 1 Introduction

Penalized log-likelihood estimation is a useful technique in high-dimensional statistical modeling. Two basic and popular penalties are the  $l_2$ -penalty or

ridge penalty, and the  $l_1$ -penalty or LASSO (Tibshirani, 1996). Both are convex and are computationally feasible. The ridge-penalty usually has the advantage of estimation and prediction accuracy. It is everywhere smooth and standard optimization methods such as Newton-Raphson can be applied. By contrast, the  $l_1$ -penalty is not differentiable at zero. This characteristic is however useful and necessary in high-dimensional model selection, because exact zero components can be obtained in the LASSO estimate so that a number of nuisance features can be discarded. For the  $l_1$  optimization algorithms in the Gaussian setup, refer to Efron et al. (2004), Daubechies et al. (2004), Friedman et al. (2007) among others.

On the other hand, the  $l_1$ -penalty cannot deal with *collinearity*. Small coherence in the design, in form of the irrerepresentable conditions (Zhao and Yu, 2006), RIP (Candes and Tao, 2005), sparse Riesz (Zhang and Huang, 2008) or others, is a *must* for the  $l_1$ -type regularization to have good performance. Many real-world applications in signal processing and bioinformatics cannot fulfill this stringent requirement. For example, the super-resolution spectral estimation must apply an overcomplete dictionary at fine enough frequency resolution and thus many sinusoidal atoms are highly correlated (see Section 6). When such collinearity occurs, (a) the prediction performance of the  $l_1$ -penalty is much worse than that of the  $l_2$ -penalty (Zou and Hastie, 2005); (b) the sparsity recovery with the  $l_1$  relaxation is inconsistent (Zhao and Yu, 2006).

To see the necessity of applying nonconvex penalties, we remind that there are two objectives involved in the task of statistical learning and modeling when one does not know the ground truth practically: **(O1)** accurate prediction, and **(O2)** parsimonious model representation. **O1**+ **O2** is consistent with Occam’s razor principle. A good approach must reflect both concerns to produce a stable parsimonious model with generalizability.

Seen from **O1**, a ridge penalty is desired to account for noise and collinearity in the data. But it never encourages sparsity. In the elastic net which uses a linear combination of the  $l_1$  penalty and the  $l_2$  penalty, the ridge part may counteract the parsimony (**O2**) in the estimate (Zou and Hastie, 2005). Yet the  $l_1$ -norm already provides the tightest convex relaxation of the  $l_0$ -norm. Therefore, to maintain accuracy *and* promote sparsity, one must take into account nonconvex penalties such as those of type ‘ $l_0 + l_2$ ’.

This paper studies some computational problems in statistical modeling in the following setup:

1. *The data are high-dimensional, and correlated.*

2. *The predictors can be naturally grouped, where pursuing the between-group sparsity is desired.*
3. *A large family of penalties should be allowed for regularization, such as the  $l_0$ -penalty,  $l_p$ -penalties, and SCAD, in addition to the convex penalty family.*
4. *The methodology and analysis should go much beyond Gaussian models to cover more applications such as classification.*

We briefly summarize some important (but absolutely not exhaustive) works in the literature as follows. In the Gaussian setup, Daubechies et al. (2004) showed an iterative soft-thresholded procedure solves the  $l_1$  penalized least-squares. Friedman et al. (2007) discovered a coordinate descent algorithm which can be viewed as a variant of the previous procedure. Recently Friedman et al. (2010b) extended the algorithm to penalized generalized linear models (GLMs), by approximating the optimization problem at each iteration via penalized weighted least-squares. However, this approximation has no guarantee of convergence and may not provide a solution to the original problem. These works focus on convex penalties.

Zou and Li (2008) recently proposed the local linear approximation (LLA) for GLMs. An adaptive LASSO optimization is carried out at *each* iteration step. The resulting algorithm has theoretical guarantee of convergence, but may not be efficient enough. Another popular approach is the DC programming (Gasso et al., 2009), which solves nonconvex penalized problems that can be represented as a difference of two convex functions (Fan and Li, 2001, Zhang, 2009, Zou and Li, 2008). Similarly, a weighted LASSO problem is solved at each iteration. Neither of the techniques directly applies to discrete penalties, such as  $l_0$  and  $l_0 + l_2$ , or group penalties.

To address the grouping concern, Yuan and Lin (2006) proposed the group LASSO. An algorithm was developed under the assumption that the predictors within each group are *orthogonal* to each other. Friedman et al. (2010a) provided an algorithm for solving convex group penalties in the Gaussian framework. How to address the nonconvex group penalties, e.g., the group  $l_0 + l_2$ , for GLMs remains unsolved.

This paper provides a general framework for penalized log-likelihood optimization for any GLMs, to address all points 1-4. Our proposed algorithm significantly generalizes She (2009) which was designed for Gaussian models only and could not attain discrete or group penalties. Using a  $q$ -function trick, this framework allows for essentially any penalties including the  $l_0$ ,  $l_p$ , and SCAD penalties. The predictors can be grouped to pursue the between-

group sparsity. Moreover, the convergence analysis in this paper is less restrictive than She (2009). No condition is imposed on the penalty function. The proof is self-contained and the conclusion applies to any thresholding rules (satisfying the mild conditions given in Definition 2.1).

The rest of the paper is organized as follows. Section 2 introduces the thresholding based algorithm with rigorous theoretical convergence analysis and presents concrete penalty examples. Section 3 discusses algorithm details and how to use numerical techniques and probabilistic screening for fast computation in high dimensions. Section 4 investigates different choices of the penalty function by simulation studies, from which a nonconvex hard-ridge penalty is advocated. Section 5 proposes a selective cross-validation (SCV) scheme for parameter tuning. In Section 6, super-resolution spectrum reconstruction is studied and a real microarray data example is analyzed to illustrate the proposed methodology. Technical details are left to Appendix.

## 2 Solving the Penalized Log-likelihood Estimation Problem

This paper assumes a **group GLM** setup that goes beyond Gaussianity. Assume the observations  $y_1, \dots, y_n$  are independent and  $y_i$  follows a distribution in the natural exponential family  $f(y_i; \theta_i) = \exp(y_i \theta_i - b(\theta_i) + c(y_i))$ , where  $\theta_i$  is the natural parameter. Let  $L_i = \log f(y_i, \theta_i)$ ,  $L = \sum L_i$ . Then  $\mu_i \triangleq E(y_i) = b'(\theta_i)$ . Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$  be the model matrix. The canonical link function, denoted by  $g$ , is applied. The Fisher information matrix at  $\boldsymbol{\beta}$  is given by  $\mathcal{I}(\boldsymbol{\beta}) = \mathbf{X}^T \mathbf{W} \mathbf{X}$  with  $\mathbf{W} \triangleq \text{diag} \{b''(\mathbf{x}_i^T \boldsymbol{\beta})\}$ . We assume the predictors are naturally *grouped*, i.e., the design matrix is grouped into  $K$  blocks:  $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_K] \in \mathbb{R}^{n \times p}$ , so that in model selection one wants to keep or kill a group of predictors as a whole. For a real example see the super-resolution spectral analysis in Section 6. The predictor groups do not overlap but the group sizes can be different. When there are  $p$  groups, each being a singleton, the model reduces to the common ‘ungrouped’ GLM. The criterion of the group  $P_k$ -penalized log-likelihood is defined by

$$F(\boldsymbol{\beta}) \triangleq -L(\boldsymbol{\beta}) + \sum_{k=1}^K P_k(\|\boldsymbol{\beta}_k\|_2; \lambda_k), \quad (1)$$

where  $L = \sum_{i=1}^n L_i$  is the log-likelihood,  $\beta_k$  are the coefficients associated with  $\mathbf{X}_k$ , and  $P_k$  are the penalty functions that can be discrete, nonconvex, and nondifferentiable at zero. The dimension  $p$  may be much greater than the sample size  $n$ . There may exist a large number of nuisance features.

Directly optimizing (1) can be tricky for a given penalty function. For example, the  $l_0$ -penalty  $\frac{\lambda^2}{2} \|\beta\|_0 = \frac{\lambda^2}{2} |\{i : \beta_i \neq 0\}|$  (where  $|\cdot|$  is the set cardinality) used for building a parsimonious model is discrete and nonconvex. We turn to another class of estimators defined via an arbitrarily given thresholding rule to solve (1) for essentially any  $P_k$ .

## 2.1 $\Theta$ -estimators

Somewhat interestingly, it is more convenient to tackle (1) from a thresholding viewpoint. The main tool of this paper is the so-called  $\Theta$ -estimators. First we define the thresholding rules rigorously as follows.

**Definition 2.1** (Threshold function). *A threshold function 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  $0 \leq t < \infty$ .

In words,  $\Theta(\cdot; \lambda)$  is an odd monotone unbounded shrinkage rule for  $t$ , at any  $\lambda$ . A vector version of  $\Theta$  (still denoted by  $\Theta$ ) is defined componentwise if either  $t$  or  $\lambda$  is replaced by a vector. Clearly,  $\Theta^{-1}(u; \lambda) \triangleq \sup\{t : \Theta(t; \lambda) \leq u\}$ ,  $\forall u > 0$  must be monotonically increasing and so its derivative is defined almost everywhere on  $(0, \infty)$ . For any  $\Theta$ , we introduce a finite positive constant  $\mathcal{L}_\Theta$  such that  $d\Theta^{-1}(u; \lambda)/du$  is bounded below almost everywhere by  $1 - \mathcal{L}_\Theta$ . For example, it is easy to show  $\mathcal{L}_\Theta$  can be 0 and 1 for soft-thresholding and hard-thresholding respectively.

A *multivariate* version of  $\Theta$ , denoted by  $\vec{\Theta}$ , is defined for any vector  $\alpha \in \mathbb{R}^p$ :

$$\vec{\Theta}(\alpha; \lambda) = \alpha^\circ \Theta(\|\alpha\|_2; \lambda), \quad (2)$$

where  $\boldsymbol{\alpha}^\circ = \begin{cases} \frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|_2}, & \text{if } \boldsymbol{\alpha} \neq \mathbf{0} \\ \mathbf{0}, & \text{if } \boldsymbol{\alpha} = \mathbf{0} \end{cases}$ . Obviously,  $\vec{\Theta}$  is still a shrinkage rule because  $\|\vec{\Theta}(\boldsymbol{\alpha}; \lambda)\|_2 = \Theta(\|\boldsymbol{\alpha}\|_2; \lambda) \leq \|\boldsymbol{\alpha}\|_2$ .

Now we define *group  $\Theta$ -estimators*. Given any threshold functions  $\Theta_1, \dots, \Theta_K$ , the induced group  $\Theta$ -estimator satisfies the following nonlinear equation

$$\boldsymbol{\beta}_k = \vec{\Theta}_k(\boldsymbol{\beta}_k + \mathbf{X}_k^T \mathbf{y} - \mathbf{X}_k^T \boldsymbol{\mu}(\boldsymbol{\beta}); \lambda_k), \quad 1 \leq k \leq K, \quad (3)$$

where  $\mu_i = g^{-1}(\mathbf{x}_i^T \boldsymbol{\beta})$  with  $g$  as the canonical link function. To avoid the influence of the ambiguity in defining some threshold functions (e.g., hard-thresholding), we always assume the quantity to be thresholded does not correspond to any discontinuity of  $\vec{\Theta}_k$ . This assumption is mild because a practical thresholding rule usually has at most finitely many discontinuity points and such discontinuities rarely occur in any real application.

As will be shown later, there is a universal connection between the penalized estimators and the group  $\Theta$ -estimators, but the latter are much easier to compute: at each iteration step  $j$ , the new  $\boldsymbol{\beta}^{(j+1)}$  can be updated through the multivariate thresholding

$$\text{Group-TISP: } \boldsymbol{\beta}_k^{(j+1)} = \vec{\Theta}_k(\boldsymbol{\beta}_k^{(j)} + \mathbf{X}_k^T \mathbf{y} - \mathbf{X}_k^T \boldsymbol{\mu}(\boldsymbol{\beta}^{(j)}); \lambda_k), \quad 1 \leq k \leq K, \quad (4)$$

provided that the norm of the global design  $\mathbf{X}$  is not large (as will be explained in Theorem 2.1). This suggests the need of scaling the data beforehand (which does not affect the sparsity of  $\boldsymbol{\beta}$ ). We refer to (4) as group thresholding-based iterative selection procedure (*Group TISP*). It generalizes the work by She (2009) in the Gaussian nongrouped setup. Next, we show (4) converges properly to a group  $\Theta$ -estimate under some appropriate conditions, which in turn solves the penalized log-likelihood problem (1) in a general sense.

**Theorem 2.1.** *Let  $\Theta_k$  ( $1 \leq k \leq K$ ) be arbitrarily given thresholding rules and  $\boldsymbol{\beta}^{(0)}$  be any  $p$ -dimensional vector. Denote by  $\boldsymbol{\beta}^{(j)}$ ,  $j = 1, 2, \dots$ , the group TISP iterates defined via (4). Define  $\rho = \sup_{\boldsymbol{\xi} \in A} \|\mathcal{I}(\boldsymbol{\xi})\|_2$  where  $A = \{\vartheta \boldsymbol{\beta}^{(j)} + (1 - \vartheta) \boldsymbol{\beta}^{(j+1)} : \vartheta \in (0, 1), j = 1, 2, \dots\}$ . If*

$$\rho \leq \max(1, 2 - \max_{1 \leq k \leq K} \mathcal{L}_{\Theta_k}), \quad (5)$$

then for any penalty functions  $P_k$  satisfying

$$P_k(\theta; \lambda_k) - P_k(0; \lambda_k) = \int_0^{|\theta|} (\sup\{s : \Theta_k(s; \lambda_k) \leq u\} - u) du + q_k(\theta; \lambda_k),$$

with  $q_k(\theta, \lambda_k)$  nonnegative and  $q_k(\Theta_k(t; \lambda_k); \lambda_k) = 0, \forall t \in \mathbb{R}$ , the value of the corresponding objective function  $F$  in (1) decreases at each iteration

$$F(\boldsymbol{\beta}^{(j)}) - F(\boldsymbol{\beta}^{(j+1)}) \geq C \|\boldsymbol{\beta}^{(j)} - \boldsymbol{\beta}^{(j+1)}\|_2^2, \quad j = 1, 2, \dots \quad (6)$$

where  $C = \max(1, 2 - \max_k \mathcal{L}_{\Theta_k}) - \rho$ . If, further,  $\rho < \max(1, 2 - \max_k \mathcal{L}_{\Theta_k})$ , any limit point of  $\boldsymbol{\beta}^{(j)}$  must be a fixed point of (3), or a group  $\Theta$ -estimate.

See A for its proof. The theorem allows for  $p > n$  and applies to *any* threshold functions, even if they are not nonexpansive. This covers essentially any penalties of practical interest, as will be shown below.

## 2.2 Concrete examples

The theorem indicates no matter how the predictors are grouped, for an arbitrarily given model matrix, performing a simple preliminary scaling  $\mathbf{X}/k_0$  always guarantees the convergence of the algorithm of (4), provided  $k_0$  is appropriately large. For a specific GLM, the choice of  $k_0$  can be made regardless of  $\Theta$ ,  $\lambda$ , and  $K$ .

**Example 2.1** (Gaussian GLM). If  $y_i$  are Gaussian,  $\boldsymbol{\mu}(\boldsymbol{\beta}) = \mathbf{X}\boldsymbol{\beta}$  in (4) and  $\boldsymbol{\mathcal{I}} = \boldsymbol{\Sigma} = \mathbf{X}^T \mathbf{X}$ . Therefore,  $k_0 \geq \|\mathbf{X}\|_2$  suffices regardless of the specific thresholding rules. This covers She (2009) where the predictors are ungrouped ( $K = p$ ) and all  $\Theta_k$ 's are identical.

**Example 2.2** (Binomial GLM). If  $y_i \sim \text{Bernoulli}(\pi_i)$  as in classification problems, we can write  $\boldsymbol{\mu}(\boldsymbol{\beta})$  as  $1/(1 + \exp(-\mathbf{X}\boldsymbol{\beta}))$  with the operations being elementwise except for the matrix-vector multiplication of  $\mathbf{X}\boldsymbol{\beta}$ . Now the proposed algorithm reduces to

$$\boldsymbol{\beta}_k^{(j+1)} = \vec{\Theta}_k \left( \boldsymbol{\beta}_k^{(j)} + \mathbf{X}_k^T \mathbf{y} - \mathbf{X}_k^T \left[ \frac{1}{1 + \exp(-\mathbf{X}\boldsymbol{\beta}^{(j)})} \right]_{n \times p} ; \lambda_k \right), 1 \leq k \leq K. \quad (7)$$

For ungrouped predictors ( $K = p$ ) and identical  $\Theta_k$ 's, the iteration can be simplified to

$$\boldsymbol{\beta}^{(j+1)} = \Theta \left( \boldsymbol{\beta}^{(j)} + \mathbf{X}^T \mathbf{y} - \mathbf{X}^T \left[ \frac{1}{1 + \exp(-\mathbf{X}\boldsymbol{\beta}^{(j)})} \right]_{n \times p} ; \lambda \right). \quad (8)$$

In either case, since  $w_i = b''(\mathbf{x}_i^T \boldsymbol{\beta}) = \pi_i(1 - \pi_i) \leq 1/4$ , a somewhat crude but general choice is  $k_0 \geq \|\mathbf{X}\|_2/2$ , regardless of  $\Theta$ . The procedure based on (8) is different than the algorithm in Friedman et al. (2010b) that approximates the original penalized logistic regression problem by penalized weighted least-squares at each iteration. Our algorithm has theoretical guarantee of convergence.

On the other hand, the experience indicates that if the algorithm converges, smaller values of  $k_0$  lead to faster convergence. It is a meaningful question in computation to find the *least* possible  $k_0$  in concrete applications. Theorem 2.1 provides useful guidance in this regard: the  $\rho$ -bound based on  $\mathcal{L}_\Theta$  seems to be tight enough in implementation for various  $\Theta$ . In the following, we give some examples of  $\Theta$  and  $P$  to show the power of the proposed algorithm for solving penalized likelihood estimation. See Figure 1 for an illustration. The function  $q$  in the theorem is often 0, but we use non-trivial  $q$ 's in Example 2.5 and Example 2.8 to attain the discrete  $l_0$  penalty and the  $l_0 + l_2$  penalty.

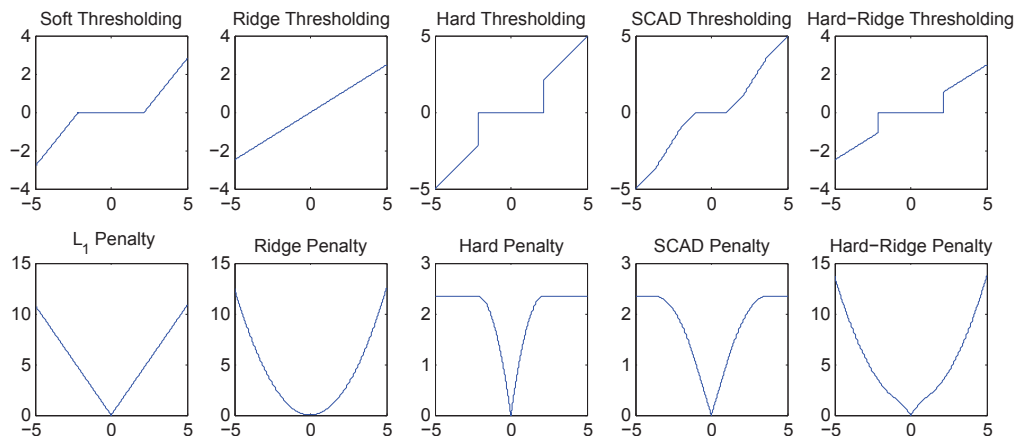


Figure 1: Some examples of the thresholding rules and their corresponding penalties. Left to right: Soft, Ridge, Hard, SCAD, and Hard-ridge.

**Example 2.3 ( $L_1$ ).** When  $\Theta$  is the soft-thresholding  $\Theta_S(t; \lambda) = \text{sgn}(t)(|t| - \lambda)1_{|t| \geq \lambda}$ , the associated penalty is  $P(\theta; \lambda) = \lambda|\theta|$ . Since we can set  $\mathcal{L}_\Theta = 0$ , the scaling constant can be relaxed to  $k_0 = \|\mathbf{X}\|_2/\sqrt{2}$  in regression and  $k_0 =$



$\|\mathbf{X}\|_2/(2\sqrt{2})$  in classification. For grouped predictors, the algorithm of (4) solves Problem (1) with the group  $l_1$ -penalty  $\sum_k \lambda_k \|\beta_k\|_2$  for any GLM, the scaling constant being the same. In comparison to Yuan and Lin (2006), we do not have to make the simplistic assumption that the predictors must be orthogonal to each other within each group.

**Example 2.4** (Elastic net). Define  $\Theta(t; \lambda_1, \lambda_2) \triangleq \Theta_S(\frac{t}{1+\lambda_2}; \frac{\lambda_1}{1+\lambda_2})$ , where  $\Theta_S$  is the soft-thresholding. Then the elastic net (Zou and Hastie, 2005) problem is solved, where  $P(\theta; \lambda_1, \lambda_2) = \lambda_1|\theta| + \lambda_2\theta^2/2$ .

**Example 2.5** ( $L_0$ ). Let  $\Theta$  be the hard-thresholding  $t1_{|t|\geq\lambda}$ . Then  $\mathcal{L}_\Theta = 1$ . According to the theorem, letting  $q \equiv 0$ , our algorithm solves for the ‘hard penalty’

$$P_H(\theta; \lambda) = \begin{cases} -\theta^2/2 + \lambda|\theta|, & \text{if } |\theta| < \lambda \\ \lambda^2/2, & \text{if } |\theta| \geq \lambda. \end{cases} \quad (9)$$

Interestingly, setting

$$q(\theta; \lambda) = \begin{cases} \frac{(\lambda-|\theta|)^2}{2}, & \text{if } 0 < |\theta| < \lambda \\ 0, & \text{if } \theta = 0 \text{ or } |\theta| \geq \lambda, \end{cases}$$

we obtain the discrete  $l_0$ -penalty  $P(\theta; \lambda) = \frac{\lambda^2}{2}1_{\theta \neq 0}$ . Similarly, we can justify that the continuous penalty  $P(\theta; \lambda) = \alpha P_H(\theta; \lambda/\sqrt{\alpha})$  mimics the  $l_0$ -penalty and results in the same  $\Theta$ -estimate, for any  $\alpha \geq 1$ . For grouped predictors, our algorithm provides a solution to the group  $l_0$ -penalty  $\sum_{k=1}^K \frac{\lambda_k^2}{2} 1_{\|\beta_k\| \neq 0}$  which can attain more between-group sparsity than the group LASSO.

**Example 2.6** (Firm & SCAD). The firm shrinkage (Gao and Bruce, 1997) is defined by

$$\Theta(t; \lambda, \alpha) = \begin{cases} 0, & \text{if } |t| < \alpha\lambda \\ \frac{t - \alpha\lambda \operatorname{sgn}(t)}{1 - \alpha}, & \text{if } \alpha\lambda \leq |t| < \lambda \\ t, & \text{if } |t| \geq \lambda, \end{cases} \quad (10)$$

where  $0 \leq \alpha \leq 1$ . The penalty function is then  $\alpha P_H(t; \lambda)$ . An equivalent form of this penalty is used in MCP (Zhang, 2010). A related thresholding is the SCAD-thresholding (Fan and Li, 2001) and the SCAD-penalized GLMs with grouped predictors can be solved by (4).

**Example 2.7** ( $L_p$ ). We focus on  $0 < p < 1$ . Assuming  $\lambda \geq 0$ , define a function

$$g(\theta; \lambda) = \theta + \lambda p \theta^{p-1}$$

for any  $\theta \in [0, +\infty)$ . It is easy to verify that (i)  $g$  attains its minimum  $\tau(\lambda) = \lambda^{1/(2-p)}(2-p)[p/(1-p)^{1-p}]^{1/(2-p)}$  at  $\theta_o = \lambda^{1/(2-p)}[p(1-p)]^{1/(2-p)}$ ; (ii)  $g(\theta)$  is strictly increasing on  $[\theta_o, +\infty)$ ; (iii)  $g(\theta) \rightarrow +\infty$  as  $\theta \rightarrow +\infty$ . Therefore, given any  $t > \tau(\lambda)$ , the equation  $g(\theta) = t$  has one and only one root in  $[\theta_o, +\infty)$  (or  $[\theta_o, t)$ , as a matter of fact), which can be found numerically. Given  $p \in (0, 1)$ , introduce the following function

$$\Theta_{l_p}(t; \lambda) = \begin{cases} 0, & \text{if } |t| \leq \tau(\lambda) \\ \text{sgn}(t) \max\{\theta : g(\theta) = |t|\}, & \text{if } |t| > \tau(\lambda). \end{cases} \quad (11)$$

Based on the properties of  $g$ , it is not difficult to show that  $\Theta_{l_p}(\cdot; \lambda)$  is indeed a threshold function, i.e., an odd monotone unbounded shrinkage rule. From the theorem,  $\Theta_{l_p}$  can handle  $P_{\Theta_{l_p}}(\theta; \lambda) = \lambda|\theta|^p$ .

**Example 2.8** (Hard-ridge ( $L_0 + L_2$ )). The hybrid hard-ridge-thresholding is defined based on the hard-thresholding and the ridge-thresholding (She, 2009)

$$\Theta(t; \lambda, \eta) = \begin{cases} 0, & \text{if } |t| < \lambda \\ \frac{t}{1+\eta}, & \text{if } |t| \geq \lambda. \end{cases} \quad (12)$$

Letting  $q \equiv 0$ , we obtain a penalty function fusing the hard-penalty and the ridge-penalty

$$P_{HR}(\theta; \lambda, \eta) = \begin{cases} -\frac{1}{2}\theta^2 + \lambda|\theta|, & \text{if } |\theta| < \frac{\lambda}{1+\eta} \\ \frac{1}{2}\eta\theta^2 + \frac{1}{2}\frac{\lambda^2}{1+\eta}, & \text{if } |\theta| \geq \frac{\lambda}{1+\eta}. \end{cases} \quad (13)$$

Moreover, for  $q(\theta; \lambda, \eta) = \frac{1+\eta}{2}(|\theta| - \lambda)^2 1_{0 < |\theta| < \lambda}$ , we obtain the  $l_0 + l_2$  penalty

$$P(\theta) = \frac{1}{2}\eta\theta^2 + \frac{1}{2}\frac{\lambda^2}{1+\eta} 1_{\theta \neq 0}. \quad (14)$$

This hard-ridge penalty offers both selection and shrinkage into regularization, interplaying with each other during the iteration for nonorthogonal designs. In the group situation, the algorithm aims for a penalty of form  $\sum_{k=1}^K \frac{\lambda_k^2}{2(1+\eta_k)} 1_{\|\beta_k\| \neq 0} + \sum_{k=1}^K \frac{\eta_k}{2} \|\beta_k\|_2^2$  which is able to deal with collinearity in the design in the pursuit of between-group sparsity.

## 3 Algorithm Design and Fast Computation

### 3.1 Algorithm design details

Either the global scaling of  $\mathbf{X}$  based on  $\mathcal{L}_\Theta$  or the iteration (4) is simple to implement. We give more algorithm design details as follows.

First, the range of the threshold parameter is finite and can be determined from (3). Assuming  $\lambda$  is the threshold and  $\mathbf{X}$  has been column normalized, we can let  $\lambda$  vary over the interval 0 to  $\|\mathbf{X}^T \mathbf{y} - \mathbf{X}^T \boldsymbol{\mu}(\mathbf{0})\|_\infty$ .

The termination criterion can be based on  $\boldsymbol{\beta}^{(j)}$  or  $F(\boldsymbol{\beta}^{(j)})$ . Extensive simulation studies showed that the approximate solution  $\boldsymbol{\beta}^{(j)}$  often had good enough performance as  $j$  is reasonably large. Each iteration involves only low-cost operations like matrix-vector multiplications. Setting a maximum number of iterations can provide a tradeoff between performance and computational complexity. Moreover, it can be shown (proof omitted) that for hard-thresholding or hard-ridge thresholding, the limiting  $\boldsymbol{\beta}^{(\infty)}$  is an ML estimate or a ridge estimate, restricted to the selected dimensions. This fact can be used in implementation when the maximum number of iterations allowed has been reached.

It remains to specify the starting point for any given  $\lambda$ . Our theory guarantees local optimality given any initial point  $\boldsymbol{\beta}^{(0)}$ . One can try multiple random starts in computation, but a nice fact is that pursuing the globally optimal solution to (1) is not at all needed to achieve significant performance gains over the  $l_1$  technique. We have found simply using the zero start, i.e.,  $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ , makes a good choice empirically. It finds a  $\Theta$ -estimate close to zero in building a parsimonious model. (Of course, other initializations are available – see, e.g., Gasso et al. (2009).) Note that since the solution path associated with a nonconvex penalty is generally discontinuous in  $\lambda$  for *nonorthogonal* models, even though the penalty and threshold function are differentiable to any order on  $(0, +\infty)$ , such as the transformed  $l_1$  (Geman and Reynolds, 1992), a pathwise algorithm with warm starts is easy to trap into poor local optima. Warm starts for a grid of values of  $\lambda$  is not recommended over the zero start unless the problem is convex.

Finally, the  $\lambda_k$  in (4) are not necessarily equal to each other. The regularization vector  $\boldsymbol{\lambda}$  can be component-specific to offer relative weights in regularizing the coefficients. This weighted form can handle GLMs with *dispersion*:  $f(y_i; \theta_i, \phi) = \exp[(y_i \theta_i - b(\theta_i))/(A_i \phi) + c(y_i, \phi)]$ , where  $\phi$  is a dispersion parameter orthogonal to  $\theta_i$ , and  $A_i$  is a known prior weight (Agresti,

2002). The normal and binomial GLMs are concrete examples. Introducing weights is also useful when a shift vector  $\alpha$  appears in the model but is unpenalized. Two examples are mean-shift outlier detection (She and Owen, 2011) and the intercept estimation. (Note that although one can center both  $\mathbf{X}$  and  $\mathbf{y}$  in a Gaussian model to make the intercept vanish, centering the response may violate the distribution assumption for nonGaussian GLMs.)

### 3.2 Fast Computation

The iteration of the proposed algorithm involves no high-complexity operations like matrix inversion. We aim to improve its convergence speed especially for high-dimensional computation.

*Numerical techniques.* Although (4) is a nonlinear process, **relaxation** and **asynchronous updating** can be incorporated to accelerate the convergence. The asynchronous updating of (4) leads to in-place computation of  $\beta$ , and the mean vector  $\mu$  is always calculated using the recently updated  $\beta$ . Under the assumptions that  $y_i$  are Gaussian and the penalty is convex, this exactly corresponds to the *coordinate descent algorithm* in Friedman et al. (2007). Yet for nonGaussian GLMs, experience shows that the original synchronous form seems to be more efficient. The relaxation of (4) is introduced as

$$\begin{aligned}\xi^{(j+1)} &= (1 - \omega)\xi^{(j)} + \omega(\beta^{(j)} + \mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mu(\beta^{(j)})), \\ \beta_k^{(j+1)} &= \tilde{\Theta}_k(\xi_k^{(j+1)}; \lambda_k), \quad 1 \leq k \leq K.\end{aligned}\tag{15}$$

We used (15) with  $\omega = 2$  in experiments, where the number of iterations can be reduced by about 40% in comparison to the original form.

*Iterative quantile screening.* To reduce the computational cost even more dramatically in high dimensions without losing much performance, probabilistic means must be taken into account apart from the numerical techniques. A reasonable idea is to screen the predictors (features) preliminarily before running (4). But for correlated data applications much more caution is needed to (a) avoid too greedy preliminary screenings, and (b) keep the screening principle consistent with the final model fitting criterion. We perform iterative feature screening by running group TISP in a *quantile* fashion: at each iteration step of (4), we set a threshold value to have exactly  $\alpha n$  nonzero components arise in  $\beta^{(j+1)}$ . Similar to Section 2, we can show the procedure is associated with the *constrained* form of the optimization problem (1). After convergence,  $\alpha n$  candidate predictors are picked. As long as  $\alpha$

is reasonably large, all relevant predictors can be maintained with high probability. Under the sparsity assumption, one can set  $\alpha < 1$ ; we have found  $\alpha = 0.8$  to be safe empirically. Sparsity-pursuing algorithms converge much faster on the screened (relatively) large- $n$  data. If the model is Gaussian, the first step of the iterative quantile screening corresponds to independence screening (Fan and Lv, 2008) based on marginal correlation statistics.

## 4 Penalty Comparison

The design of the penalty  $P$  or the threshold function  $\Theta$  is an important topic in applying penalized log-likelihood estimation into real-world problems. We performed systematic simulation studies to compare difference penalty functions in sparse modeling. Five methods were studied: LASSO (with calibration), one-step SCAD, the nonconvex  $l_0$ -penalty, SCAD-penalty, and hard-ridge penalty. The first two are convex but multi-stage. Similar to the idea of the *LARS-OLS hybrid* (Efron et al., 2004), we calibrated the LASSO estimate by fitting an unpenalized likelihood model restricted to its selected predictors. One-step SCAD is an example of the one-step LLA (Zou and Li, 2008) which fits a weighted LASSO with the weights constructed from the ML estimate and the penalty function. We used the previous tuned LASSO-MLE as the initial estimate in weight construction which behaves better than the ML estimate and applies to  $p > n$ . The remaining three nonconvex methods were all be computed by the proposed algorithm. Neither SCAD nor  $l_0$  introduces estimation bias for large coefficients. Hard-ridge penalty does simultaneous selection and shrinkage with a thresholding parameter  $\lambda$  and a ridge parameter  $\eta$ . For efficiency, we did not run a full two-dimensional grid search when looking for the best parameters. Instead, for each  $\eta$  in the grid  $\{0.5\eta^*, 0.05\eta^*, 0.005\eta^*\}$  where  $\eta^*$  is the optimal ridge parameter, we find  $\lambda(\eta)$  to minimize the validation error; then for  $\lambda$  fixed at the optimal value, we find the best  $\eta$  to minimize the validation error.

We seek to evaluate and compare the performances of different penalties in this section. To understand the true potential of each method in an ideal situation and allow us to draw a stable performance comparison, we tuned all regularization parameters on a very large independent **validation** dataset. The simulation setup is as follows. Let  $\beta = (b, 0, b, b, 0, \dots, 0)^T$ ,  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$  and  $\mathbf{x}_i$  are i.i.d.  $\sim \text{MVN}(\mathbf{0}, \Sigma)$  where  $\Sigma_{jk} = \rho^{|j-k|}$ ,  $1 \leq j, k \leq p$ . Note that all group penalties in (1) use the same  $l_2$ -norm

for within-group penalization. The difference lies in between-group penalties. In the experiment, to make this difference more prominent, we let each predictor fall into an individual predictor group. The control parameters were varied by  $(n, p) = (100, 20), (100, 100), (100, 500)$ ,  $\rho = 0.1, 0.5, 0.9$ , and  $b = .75, 1, 2.5$ . We generated an additional large **test** dataset with 10,000 observations to evaluate the performance of any algorithm, as well as an **validation** dataset of the same size to tune the regularization parameters. All  $3^3 = 27$  combinations of the *problem size*, *design correlation*, and *signal strength* were covered in the simulations. We measured an algorithm’s performance by prediction accuracy and sparsity recovery, for each model simulated 50 times. We evaluated the scaled deviance error (SDE)  $100(\sum_{i=1}^N \log f(y_i; \hat{\beta}) / \sum_{i=1}^N \log f(y_i; \beta) - 1)$  on the test data. For stability, we reported the 40% trimmed-mean of the SDEs from the 50 runs. We also reported variable selection results via three benchmark measures: the mean masking (**M**) and swamping (**S**) probabilities, and the rate of successful joint detection (**JD**). The masking probability is the fraction of undetected relevant variables (*misses*), the swamping probability is the fraction of spuriously identified variables (*false alarms*), and the JD is the fraction of simulations with zero miss. In variable selection, masking is a much more serious problem than swamping, and an ideal method should have  $M \approx 0\%$ ,  $S \approx 0\%$ , and  $JD \approx 100\%$ . The simulation results for logistic regression are summarized in Figures 2, 3, and 4.

We briefly summarize the conclusions as follows. Seen from the results, the Lasso-MLE that chooses  $\lambda$  according to the bias corrected lasso alleviated the issue that even when the signal-to-noise ratio is pretty high, the lasso overselects (Leng et al., 2006), but still leaves much room for improvement. The nonconvex  $l_0$  yields a restricted ML estimate, too, but is single-stage, and often did better in variable selection. The weighting technique in *one-step* SCAD, though theoretically effective for  $p$  fixed and  $n \rightarrow \infty$ , requires a careful choice of the initial estimate in finite samples. The improvement brought by weighting was somewhat limited, especially when some predictors are correlated. Fully solving the nonconvex SCAD problem, though using a naïve zero start, showed good large- $p$  performance. In the 27 experiments, the nonconvex hard-ridge penalization (13) (or (14)) had striking advantage in prediction and sparsity recovery simultaneously, in various challenging situations of large  $p$ , low signal strength, and/or high collinearity. Its  $l_2$ -portion dealt with collinearity well and adapted to different noise levels; meanwhile, its  $l_0$ -portion, nondifferentiable at zero, enforced higher level of sparsity than

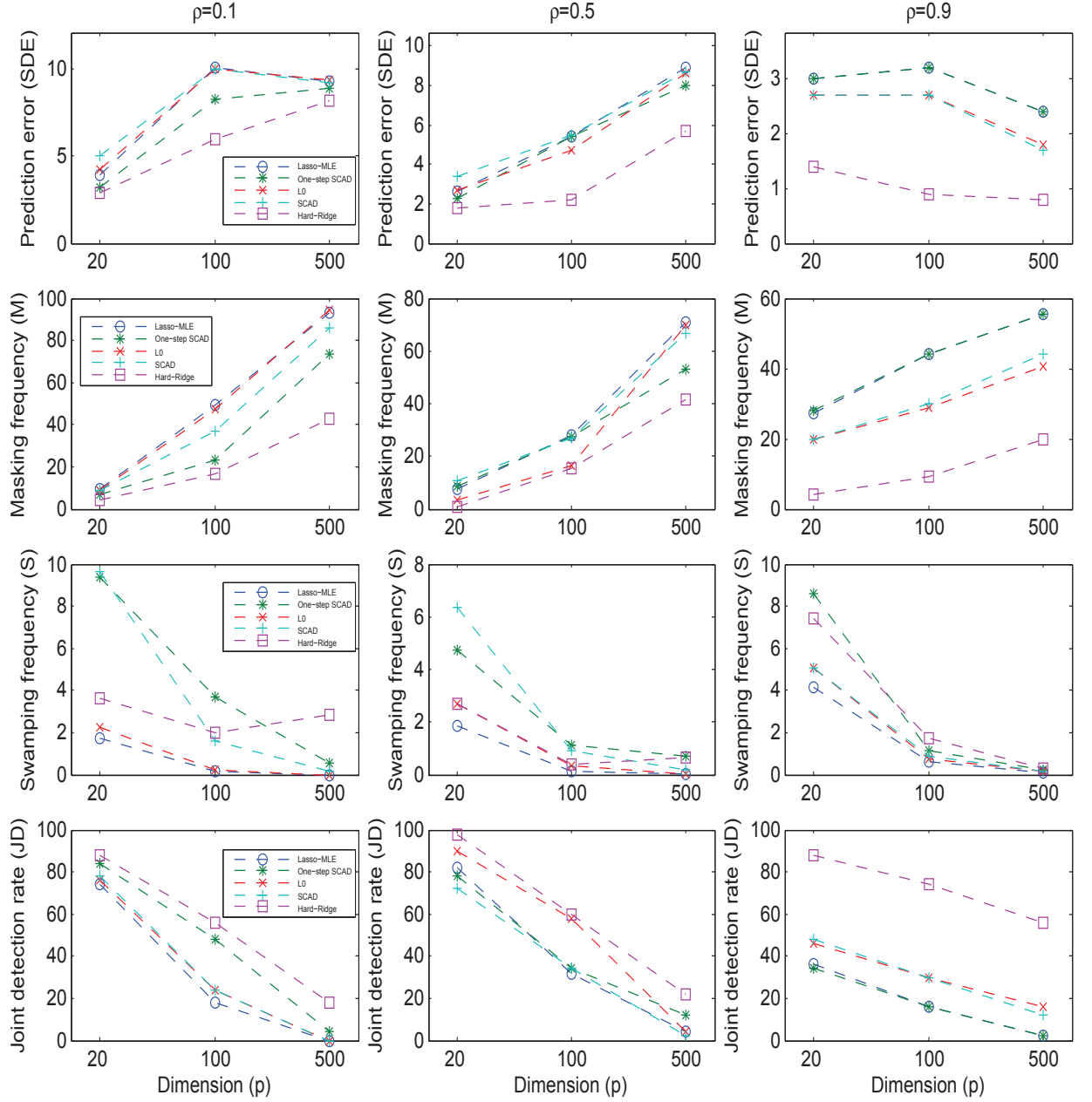


Figure 2: Performance comparison of different penalties in terms of test error, masking/swamping probabilities, and joint identification rate for logistic regression models with  $b = 0.75$ .

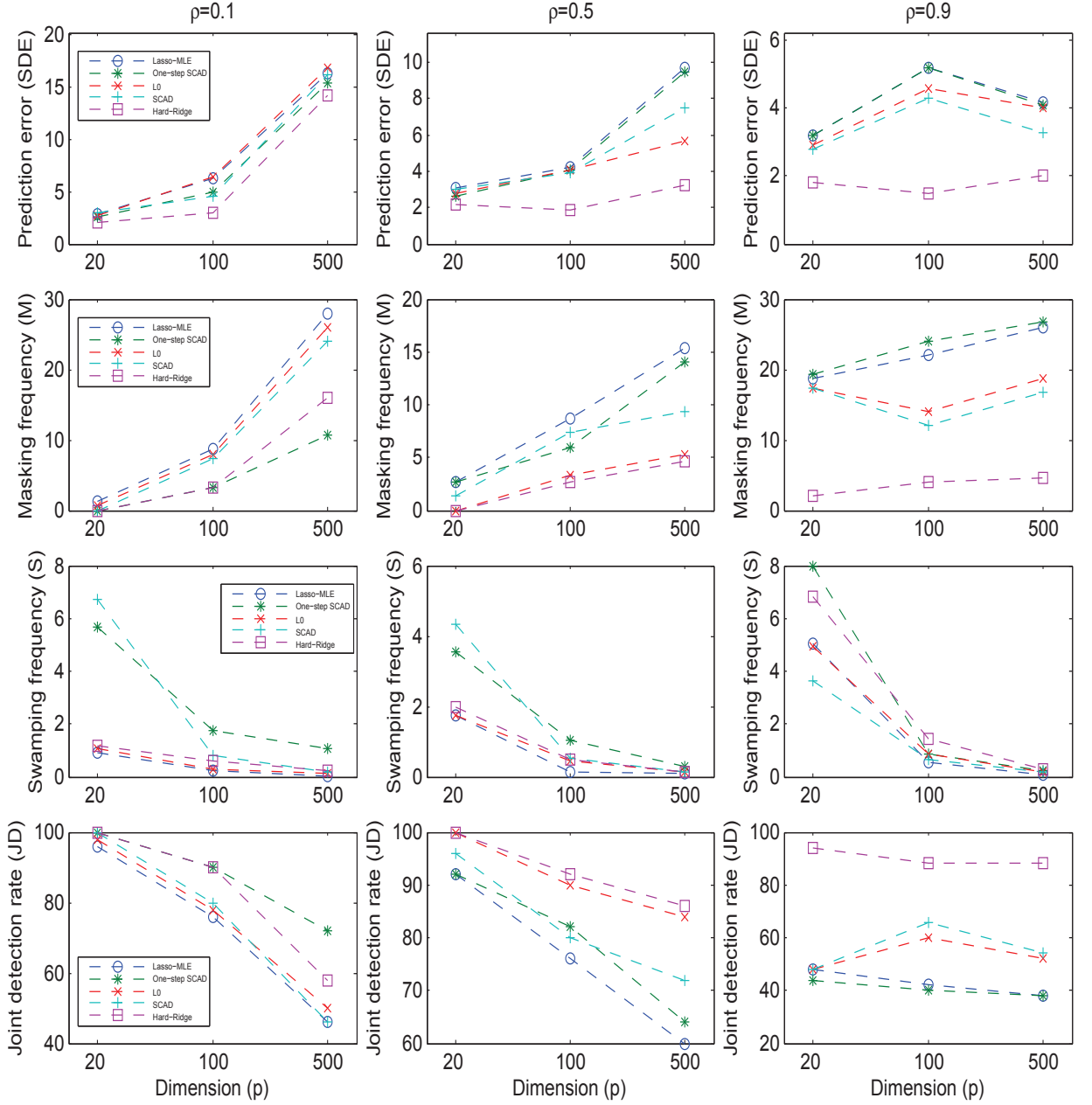


Figure 3: Performance comparison of different penalties in terms of test error, masking/swamping probabilities, and joint identification rate for logistic regression models with  $b = 1$ .



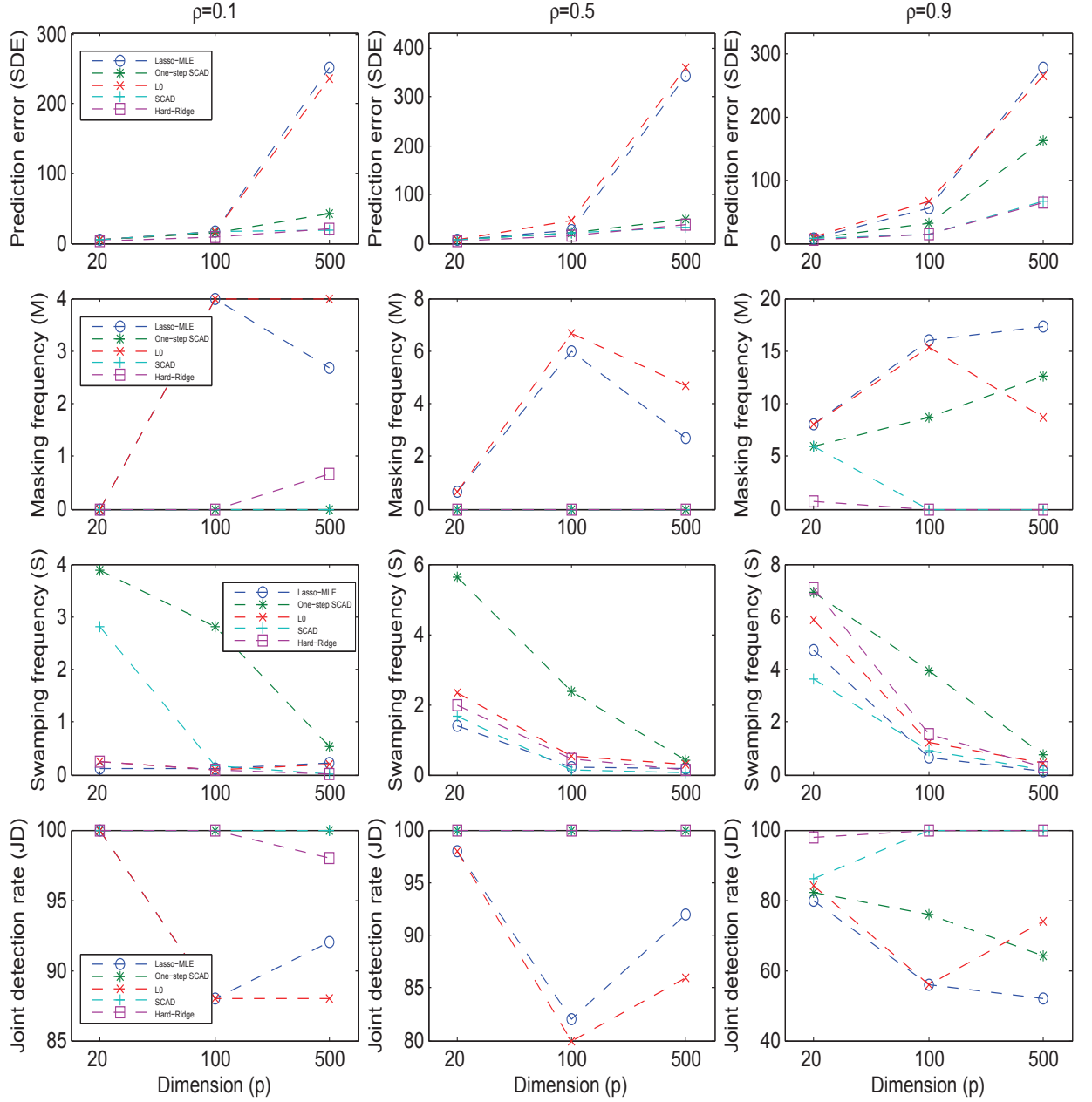


Figure 4: Performance comparison of different penalties in terms of test error, masking/swamping probabilities, and joint identification rate for logistic regression models with  $b = 2.5$ .

convex techniques. Not surprisingly, in computation, nonconvex penalties required more computational time than the  $l_1$ , but the cost is acceptable. For example, in the setup of  $(n, p) = (100, 500)$ ,  $\rho = 0.5$ ,  $b = 0.75$ , the total running time (in seconds) was 322.6, 1184.1, 1910.5, 1553.3 for  $l_1$ ,  $l_0$ , SCAD, and hard-ridge, respectively. The performance boost, with some sacrifice in computational time, is affordable.

## 5 Choice of the Regularization Parameter

Parameter tuning plays an important role in penalized log-likelihood estimation. If we assume  $\beta$  is sparse and the sample size  $n$  is large relative to the true dimensionality (denoted by, say,  $p_{nz}$ ), then BIC can be used but may still suffer from overselection (Chen and Chen, 2008). Directly cross-validating (CV) the regularization parameter  $\lambda$  is also popular in the literature. However, it may not be appropriate for nonconvex penalties. (i) The optimal value of  $\lambda$  in the penalized criterion (1) is a function of the true  $\beta$  and the data  $(\mathbf{X}, \mathbf{y})$ . As the training data change, the optimal value of the penalty parameter may not remain the same. But  $K$ -fold CV requires  $K$  different trainings. (ii) Even if a nonconvex penalty and its corresponding threshold function are smooth on  $(0, \infty)$ , the solution path  $\hat{\beta}(\lambda)$  is typically discontinuous in  $\lambda$  for nonorthogonal designs. As a consequence, for any given value of  $\lambda$ , the  $K$  fitted models in CV may not be directly comparable, and thus averaging the CV errors can be unstable and misleading. A crucial question is how to guarantee the  $K$  trainings (and validations) are associated with the same model.

To address the issue in sparsity problems, we propose  $K$ -fold *selective cross-validation* (SCV) outlined below. Let  $\mathcal{A}$  be a given sparsity algorithm.

1. Run  $\mathcal{A}$  on the *whole* dataset for  $\lambda$  in a grid of values, getting the solution path  $\hat{\beta}_l$ ,  $1 \leq l \leq L$ . The associated sparsity patterns are denoted by  $nz_l = nz(\hat{\beta}_l)$ ,  $1 \leq l \leq L$ .
  2. For each  $l$ , run cross-validation to fit  $K$  models with only the predictors picked by  $nz_l$ . We use *degree-of-freedom (df) matching* to find proper shrinkage parameter in each training.
  3. Summarize the CV deviance errors and determine the optimal estimate and sparsity pattern in the solution path.
- Step 1 determines candidate sparsity patterns to be used in the training step. Given  $k$  ( $1 \leq k \leq K$ ), on the data without the  $k$ th subset, Step 2 fits models

*restricted* to the selected dimensions only. Specifically, if  $\hat{\beta}_l$  is from the  $l_0$  penalization, the df is essentially the number of nonzero components in  $\hat{\beta}_l$ . Therefore, in each CV training, we simply fit a model with MLE, unpenalized and restricted to  $nz_l$ . For the hard-ridge penalty, the contribution of the ridge parameter  $\eta$  must be considered, for the df of an  $l_2$ -penalized GLM with estimate  $\hat{\beta}$  is approximately  $Tr\{(\mathcal{I}(\hat{\beta}) + \eta \mathbf{I})^{-1} \mathcal{I}(\hat{\beta})\}$  (Agresti, 2002). To guarantee the  $k$ th trained ridge model has the same df as  $\hat{\beta}_l$ , bisection search can be used to find the appropriate value  $\eta_k$ . Finally, in Step 3, the prediction errors on the left-out piece of data can be summarized by  $-2 \sum_{i=1}^n \log f(y_i; \hat{\beta}_l^{-k(i)}) =: \text{SCV}(l)$ , where  $\hat{\beta}_l^{-k(i)}$  denotes the above local estimate without the  $k(i)$ th subset and restricted to the selected dimensions. If the model is very sparse— $p_{nz} \ll n$  and  $p_{nz} \ll p$ , a BIC correction term can be added:  $\text{SCV-BIC}(l) = \text{SCV}(l) + \log n \cdot \text{df}(\hat{\beta}_l)$ . Empirically, this new criterion can overcome the overselection issue of BIC, through replacing the training error by the SCV error. A similar idea is used in Bunea and Barbu (2009).

In summary, SCV runs the given sparse algorithm only *once* and *globally*, instead of  $K$  times locally, to determine the common sparsity patterns. It can reduce the computational cost and resolve the model inconsistency issue of the plain CV.

## 6 Applications

We demonstrate the efficacy of our algorithm for computing nonconvex penalized models by super-resolution spectral analysis in signal processing, and cancer classification and gene selection in microarray data analysis.

### 6.1 Super-resolution spectral analysis

The problem of spectral estimation studies how the signal power is distributed over frequencies, and has rich applications in speech coding and radar sonar signal processing. It becomes very challenging when the required frequency resolution is high, because the number of the frequency levels at a desired resolution can be (much) greater than the sample size, referred to as *super-resolution* spectral estimation. Super-resolution spectral analysis goes beyond the traditional Fourier analysis and is one of the first areas where the  $l_1$ -relaxation technique, i.e., the *Basis Pursuit* by Chen et al. (1998),

was proposed. Here we revisit the problem and demonstrate the advantage brought by group nonconvex penalized likelihood estimation. We focus on the classical **TwinSine** signal arising from target detection:

$$y(t) = a_1 \cos(2\pi f_1 t + \phi_1) + a_2 \cos(2\pi f_2 t + \phi_2) + n(t)$$

where  $a_1 = 2$ ,  $a_2 = 3$ ,  $\phi_1 = \pi/3$ ,  $\phi_2 = \pi/5$ ,  $f_1 = 0.25\text{Hz}$ ,  $f_2 = 0.252\text{Hz}$  and  $n(t)$  is white Gaussian noise with variance  $\sigma^2$ . Obviously, the frequency resolution needs to be as fine as 0.002 Hz to perceive and distinguish the two sinusoidal components. For convenience, assume the data sequence is evenly sampled at  $n = 100$  time points  $t_i = i$ ,  $1 \leq i \leq n$ . (Our approach does not require uniform sampling.) An *overcomplete* dictionary to attain the desired frequency resolution can be constructed by setting the maximum frequency  $f_{\max} = 1/2 = 0.5$  Hz, and the number of frequency bins  $D = 250$ . Concretely, let  $f_k = f_{\max} \cdot k/D$  for  $k = 0, 1, \dots, D$  and define the frequency atoms  $\mathbf{X}_{\cos} = [\cos(2\pi t_i f_k)]_{1 \leq i \leq n, 1 \leq k \leq D}$  and  $\mathbf{X}_{\sin} = [\sin(2\pi t_i f_k)]_{1 \leq i \leq n, 1 \leq k \leq D-1}$ , where the last sine atom vanishes because  $\sin(2\pi t_i f_D) = 0$  for integer-valued  $t_i$ . Then  $\mathbf{X} = [\mathbf{X}_{\cos} \ \mathbf{X}_{\sin}]$  is of dimension 100-by-499 without the intercept, resulting in a challenging high-dimensional learning problem. In this situation, the classical Fourier transform based periodogram or least-squares periodogram (LSP) suffers from severe power leakage, while the basis pursuit (BP) is able to super-resolve under the spectral sparsity assumption. On the other hand, the *pairing* structure of cosine and sine atoms is often ignored in spectrum recovery. More seriously, when the desired frequency resolution is sufficiently high, the dictionary contains many similar sinusoidal components and the high pairwise correlations may make the  $l_1$  relaxation of the  $l_0$ -norm corrupted in selecting all frequencies consistently.

We simulated the signal model at given noise levels  $\sigma^2 = 8, 1, 0.1$ , each with 20 times to evaluate the performance of an algorithm. At each run, we generated additional test data at  $N = 2000$  time points different than those of the training data to calculate the effective prediction error  $\text{MSE}^* = \sum_{i=1}^N (y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}} - \hat{\alpha})^2 / (N - \sigma^2)$ . The median of  $\text{MSE}^*$  was reported, denoted by **Err**, as the goodness of fit of the obtained model. The frequency detection is measured by joint detection rates – **JD**, misses – **M**, and false alarms – **S** defined in Section 4. Table 1 compares the performance of BP, grouped lasso, hard-ridge and grouped hard-ridge penalized regressions on the TwinSine signal, all of which were computed via the proposed algorithm.

To see the true potential of each penalty in an ideal situation, in the first 4 experiments we used independent large validation data (of 2000 observations)

Table 1: Performance comparison of basis pursuit, grouped lasso (G-Lasso), hard-ridge and grouped hard-ridge (G-Hard-Ridge) penalized regressions for spectral estimation.

|                     |                | $\sigma^2 = 8$ |    |      |     | $\sigma^2 = 1$ |     |    |     | $\sigma^2 = 0.1$ |     |    |     |
|---------------------|----------------|----------------|----|------|-----|----------------|-----|----|-----|------------------|-----|----|-----|
|                     |                | SNR = 18.13    |    |      |     | SNR = 8.13     |     |    |     | SNR = -0.90      |     |    |     |
|                     | Tuning         | Err            | JD | M    | S   | Err            | JD  | M  | S   | Err              | JD  | M  | S   |
| Basis pursuit       | Large-Val      | 4.15           | 0  | 55   | 0.5 | 3.00           | 0.0 | 50 | 0.3 | 2.87             | 0.0 | 50 | 0.3 |
| Hard-Ridge          | Large-Val      | 1.70           | 45 | 16.3 | 0.4 | 0.36           | 80  | 5  | 0.2 | 0.03             | 100 | 0  | 0.1 |
| G-LASSO             | Large-Val      | 1.58           | 90 | 5.0  | 1.8 | 0.25           | 100 | 0  | 1.6 | 0.04             | 100 | 0  | 2.8 |
| <b>G-Hard-Ridge</b> | Large-Val      | 0.66           | 95 | 2.5  | 0.1 | 0.16           | 100 | 0  | 0.0 | 0.02             | 100 | 0  | 0.0 |
| <b>G-Hard-Ridge</b> | <b>SCV-BIC</b> | 1.11           | 85 | 7.5  | 0.0 | 0.27           | 100 | 0  | 0.0 | 0.12             | 100 | 0  | 0   |

to tune the parameters. The penalty comparison showed the improvement of the nonconvex hard-ridge penalty in both time-domain prediction and frequency-domain spectrum reconstruction. In the last experiment, the hard-ridge was run with no additional validation data. We used SCV with BIC correction on the 100 training observations.

Again, our results showed that pursuing the global minimum of the non-convex criterion (1) is not necessary; the zero start in (4) offered good accuracy and regularization. In the experiments we predefined a maximum iteration number  $M_{\max} = 5000$  (see Section 3.1). To solve the  $l_0 + l_2$  type problems, our algorithm required 4 to 6 times as much time as the  $l_1$  in computing one solution path. The higher computational complexity is expected but is an acceptable tradeoff between performance and computational complexity in super-resolution spectral analysis.

## 6.2 Classification and gene selection

We then illustrate our algorithms with an example of cancer classification with joint gene selection. We analyzed real acute lymphoblastic leukemia (ALL) data conducted with HG-U95Av2 Affymetrix arrays (Chiaretti et al., 2004). Following Scholtens and von Heydebreck (2005), we focus on the B-cell samples and would like to contrast the patients with the BCR/ABL fusion gene resulting from a translocation of the chromosomes 9 and 22, with those who are cytogenetically normal (NEG). The preprocessed data can be loaded from the Bioconductor data package ALL. This leads to 2,391 probe sets and 79 samples, 42 labeled with “NEG” and 37 labeled with “BCR/ABL”.

We first ran iterative quantile screening introduced in Section 3.2 for dimension reduction. Specifically, we ran quantile TISP with the hard-ridge thresholding function for  $\eta$  in a small grid of values, and then chose the optimal one by 5-fold SCV. We used  $\alpha = 0.8$ . Then we ran the original form of the algorithm (4) to solve hard-ridge penalized logistic regression. The parameters were tuned by 5-fold SCV with no/AIC/BIC correction. For comparison, we tested another two up-to-date classifiers with *joint* gene selection: the nearest shrunken centroids (Tibshirani et al., 2002) (denoted by NSC) and the Ebay algorithm (Efron, 2009). (The results of the  $l_1$  penalized logistic regression are not reported, because in comparison, it gave similar error rates but selected too many ( $> 30$ ) genes.) For an implementation of NSC, refer to the package `pamr` in R. The R-code for Ebay is also available online (Efron, 2009). Their regularization parameters were tuned by cross-validation. To prevent from getting over-optimistic error rate estimates, we used a *hierarchical* cross-validation procedure where an outer 10-fold CV was used for performance evaluation while the inner CVs were used for parameter tuning. Table 2 summarizes the prediction and selection performances of the three classifiers. The proposed algorithms had excellent performance. Hard-ridge-penalty with SCV-BIC tuning behaved the best for the given data: it gave the smallest error rate and produced the most parsimonious model with only about 8 genes involved.

Table 2: Prediction error and the number of selected genes.

|                         | Misclassification error rate<br>(mean, median) | # of selected genes<br>(mean, median) |
|-------------------------|--|---------------------------------------|
| NSC                     | 16.4%, 12.5%                                   | 19.3, 14                              |
| Ebay                    | 12.7%, 12.5%                                   | 16.2, 16                              |
| Hard-Ridge with SCV     | 11.3%, 12.5%                                   | 21.4, 22.5                            |
| Hard-Ridge with SCV-AIC | 10.2%, 6.3%                                    | 11.2, 9.5                             |
| Hard-Ridge with SCV-BIC | 8.9%, 6.3%                                     | 8.1, 8                                |

Next we identify the relevant genes. We bootstrapped the data 100 times. For each bootstrap dataset, after standardizing the predictors, we fit a hard-ridge penalized logistic regression with the parameters tuned by 5-fold SCV-BIC. Figure 5 plots the frequencies of the coefficient estimates being nonzero and the estimate histograms over the 100 replications. The bootstrap results

give us a confidence measure of selecting each gene. The top three probesets had nonzero coefficients more frequently ( $> 50\%$  of the time) and they jointly appeared 63 times in the selected models, the most frequently visited triple in bootstrapping. Annotation shows that all three probe sets – 1636\_g\_at, 39730\_at, and 1635\_at – are associated with the same gene – ABL1.

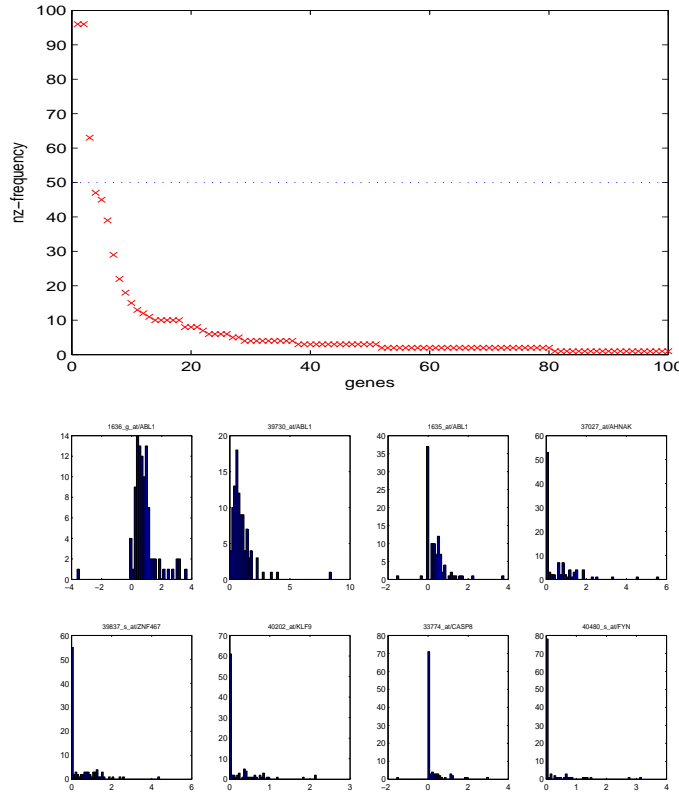


Figure 5: Upper panel: Proportions of the coefficient estimates being nonzero over the 100 bootstrap replications (only the top 100 genes are plotted). Lower panel: Histograms of the bootstrap coefficient estimates of the top 8 genes.

## 7 Conclusion

The paper proposed a simple-to-implement algorithm for solving penalized log-likelihoods. The predictors can be arbitrarily grouped to pursue the

between-group sparsity and we do not require the within-group predictors to be orthogonal. Our treatment is rigorous and applies to any GLM. We proved a convergence condition in theory and it leads to a tight preliminary scaling which helps reduce the number of iterations in implementation. Our algorithm and theoretical analysis allow for essentially any nonconvex penalty, and a  $q$ -function trick was used to attain the exact discrete  $l_0$  and  $l_0 + l_2$  penalties.

## A Proof of Theorem 2.1

**Lemma 1.** *Given an arbitrary thresholding rule  $\Theta$ , let  $P$  be any function satisfying  $P(\theta; \lambda) - P(0; \lambda) = P_\Theta(\theta; \lambda) + q(\theta; \lambda)$  where  $P_\Theta(\theta; \lambda) \triangleq \int_0^{|\theta|} (\sup\{s : \Theta(s; \lambda) \leq u\} - u) du$ ,  $q(\theta; \lambda)$  is nonnegative and  $q(\Theta(t; \lambda)) = 0$  for all  $t$ . Then, the minimization problem*

$$\min_{\beta \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \beta\|_2^2 + P(\|\beta\|_2; \lambda) \triangleq Q(\beta; \lambda)$$

*has a unique optimal solution given by  $\hat{\beta} = \vec{\Theta}(\mathbf{y}; \lambda)$  for every  $\mathbf{y}$  provided that  $\Theta(\cdot; \lambda)$  is continuous at  $\|\mathbf{y}\|_2$ .*

Note that  $P$  (and  $P_\Theta$ ) may not be differentiable at 0 and may be nonconvex. For notational simplicity, we simply write  $Q(\beta)$  for  $Q(\beta; \lambda)$  when there is no ambiguity. This lemma may be considered as a generalization of Proposition 3.2 in Antoniadis (2007).

*Proof of Lemma 1.* First, it suffices to consider  $\beta$  satisfying  $y_i \beta_i \geq 0$  because for any  $\beta$ ,  $Q(\beta) \geq Q(\beta')$  with  $\beta'_i = \text{sgn}(y_i) |\beta_i|$ . By definition, we have

$$\begin{aligned} Q(\beta) - Q(\hat{\beta}) &= -\mathbf{y}^T(\beta - \hat{\beta}) + \frac{1}{2}(\|\beta\|_2^2 - \|\hat{\beta}\|_2^2) + P_\Theta(\|\beta\|_2; \lambda) - P_\Theta(\|\hat{\beta}\|_2; \lambda) \\ &\quad + q(\|\beta\|_2; \lambda) - q(\Theta(\|\mathbf{y}\|_2; \lambda); \lambda) \\ &= -\mathbf{y}^T(\beta - \hat{\beta}) + \int_{\Theta(\|\mathbf{y}\|_2; \lambda)}^{\|\beta\|_2} (u + \Theta^{-1}(u; \lambda) - u) du + q(\|\beta\|_2; \lambda). \end{aligned}$$

On the other hand,

$$\begin{aligned} -\mathbf{y}^T(\beta - \hat{\beta}) &= -\mathbf{y}^T \beta + \|\mathbf{y}\|_2 \Theta(\|\mathbf{y}\|_2; \lambda) \\ &\geq -\|\mathbf{y}\|_2 \|\beta\|_2 + \|\mathbf{y}\|_2 \Theta(\|\mathbf{y}\|_2; \lambda) \\ &= -\|\mathbf{y}\|_2 (\|\beta\|_2 - \Theta(\|\mathbf{y}\|_2; \lambda)) \\ &= -\|\mathbf{y}\|_2 (\|\beta\|_2 - \|\hat{\beta}\|_2). \end{aligned}$$



Hence  $Q(\boldsymbol{\beta}) - Q(\hat{\boldsymbol{\beta}}) \geq \int_{\Theta(\|\mathbf{y}\|_2; \lambda)}^{\|\boldsymbol{\beta}\|_2} (\Theta^{-1}(u; \lambda) - \|\mathbf{y}\|_2) du + q(\|\boldsymbol{\beta}\|_2; \lambda)$ .

Suppose  $\|\boldsymbol{\beta}\|_2 > \Theta(\|\mathbf{y}\|_2; \lambda)$ . By definition  $\Theta^{-1}(\|\boldsymbol{\beta}\|_2; \lambda) \geq \|\mathbf{y}\|_2$ , and thus  $Q(\boldsymbol{\beta}) \geq Q(\hat{\boldsymbol{\beta}})$ . Furthermore, there must exist some  $u \in [\Theta(\|\mathbf{y}\|_2; \lambda), \|\boldsymbol{\beta}\|_2)$  s.t.  $\Theta^{-1}(u; \lambda) > \|\mathbf{y}\|_2$ , and hence  $Q(\boldsymbol{\beta}) > Q(\hat{\boldsymbol{\beta}})$  due to the monotonicity of  $\Theta^{-1}$ . In fact, if this were not true, we would have  $\Theta(t; \lambda) > \|\boldsymbol{\beta}\|_2 \geq \Theta(\|\mathbf{y}\|_2; \lambda)$  for any  $t > \|\mathbf{y}\|_2$ , and  $\Theta(\cdot; \lambda)$  would be discontinuous at  $t$ . A similar reasoning applies to the case when  $\|\boldsymbol{\beta}\|_2 < \Theta(\|\mathbf{y}\|_2; \lambda)$ . The proof is now complete.  $\square$

Hereinafter, we always assume  $\Theta(t; \lambda)$  is continuous at any  $t$  to be thresholded, since a practical thresholding rule usually has at most finitely many discontinuity points and such discontinuities rarely occur in any real application.

**Lemma 2.** Let  $Q_0(\boldsymbol{\beta}) = \|\mathbf{y} - \boldsymbol{\beta}\|_2^2/2 + P_\Theta(\|\boldsymbol{\beta}\|_2; \lambda)$ . Denote by  $\hat{\boldsymbol{\beta}}$  the unique minimizer of  $Q_0(\boldsymbol{\beta})$ . Then for any  $\boldsymbol{\delta}$ ,  $Q_0(\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}) - Q_0(\hat{\boldsymbol{\beta}}) \geq C_1 \|\boldsymbol{\delta}\|_2^2/2$ , where  $C_1 = \max(0, 1 - \mathcal{L}_\Theta)$ .

*Proof of Lemma 2.* Let  $s(u; \lambda) = \Theta^{-1}(u; \lambda) - u = \sup\{t : \Theta(t; \lambda) \leq u\} - u$ . We have

$$\begin{aligned} Q_0(\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}) - Q_0(\hat{\boldsymbol{\beta}}) &= \frac{1}{2} \|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta} - \mathbf{y}\|_2^2 - \frac{1}{2} \|\hat{\boldsymbol{\beta}} - \mathbf{y}\|_2^2 + P_\Theta(\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2) - P_\Theta(\|\hat{\boldsymbol{\beta}}\|_2) \\ &= \frac{1}{2} \|\boldsymbol{\delta}\|_2^2 + (\hat{\boldsymbol{\beta}} - \mathbf{y})^T \boldsymbol{\delta} + \int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} s(u; \lambda) du \end{aligned} \quad (16)$$

(i) If  $\hat{\boldsymbol{\beta}} = \mathbf{0}$ ,  $\vec{\Theta}(\mathbf{y}; \lambda) = \mathbf{0}$  and so  $\Theta(\|\mathbf{y}\|_2; \lambda) = 0$ , from which it follows that  $\|\mathbf{y}\|_2 \leq \Theta^{-1}(0; \lambda)$ . Therefore,

$$(\hat{\boldsymbol{\beta}} - \mathbf{y})^T \boldsymbol{\delta} \geq -\|\mathbf{y}\|_2 \cdot \|\boldsymbol{\delta}\|_2 \geq -\Theta^{-1}(0; \lambda) \|\boldsymbol{\delta}\|_2 = - \int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda) du.$$

(ii) If  $\hat{\boldsymbol{\beta}} \neq \mathbf{0}$ , it is easy to verify by Lemma 1 that  $\hat{\boldsymbol{\beta}}$  satisfies  $\mathbf{y} - \hat{\boldsymbol{\beta}} = s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda) \mathbf{y}^\circ = s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda) \hat{\boldsymbol{\beta}}^\circ$ , and thus

$$(\hat{\boldsymbol{\beta}} - \mathbf{y})^T \boldsymbol{\delta} = -s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda) \boldsymbol{\delta}^T \hat{\boldsymbol{\beta}}^\circ.$$

For any  $\mathbf{a} \neq \mathbf{0}$ , it follows from Cauchy's inequality that

$$\mathbf{b}^T \mathbf{a}^\circ + \|\mathbf{a}\|_2 = \mathbf{a}^T (\mathbf{a} + \mathbf{b}) / \|\mathbf{a}\|_2 \leq \|\mathbf{a} + \mathbf{b}\|_2,$$

or  $\|\mathbf{a} + \mathbf{b}\|_2 - \|\mathbf{a}\|_2 \geq \mathbf{b}^T \mathbf{a}^\circ$ . Making use of this fact, we obtain

$$(\hat{\boldsymbol{\beta}} - \mathbf{y})^T \boldsymbol{\delta} \geq -(\|\boldsymbol{\delta} + \hat{\boldsymbol{\beta}}\|_2 - \|\hat{\boldsymbol{\beta}}\|_2) s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda) = - \int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda) \, du.$$

In either case, (16) can be bounded in the following way:

$$\begin{aligned} Q_0(\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}) - Q_0(\hat{\boldsymbol{\beta}}) &\geq \frac{1}{2} \|\boldsymbol{\delta}\|_2^2 + \int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} (s(u; \lambda) - s(\|\hat{\boldsymbol{\beta}}\|_2; \lambda)) \, du \\ &= \frac{1}{2} \|\boldsymbol{\delta}\|_2^2 + \int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} \left( (\Theta^{-1}(u; \lambda) - \Theta^{-1}(\|\hat{\boldsymbol{\beta}}\|_2; \lambda)) - (u - \|\hat{\boldsymbol{\beta}}\|_2) \right) \, du. \end{aligned}$$

By the Lebesgue Differentiation Theorem,  $(\Theta^{-1})'$  exists almost everywhere and

$$\int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} (\Theta^{-1}(u; \lambda) - \Theta^{-1}(\|\hat{\boldsymbol{\beta}}\|_2; \lambda)) \, du \geq \int_{\|\hat{\boldsymbol{\beta}}\|_2}^{\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2} \int_{\|\hat{\boldsymbol{\beta}}\|_2}^u (\Theta^{-1})'(v; \lambda) \, dv \, du.$$

By the definition of  $\mathcal{L}_\Theta$ ,  $Q_0(\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}) - Q_0(\hat{\boldsymbol{\beta}}) \geq \frac{1}{2} \|\boldsymbol{\delta}\|_2^2 - \frac{\mathcal{L}_\Theta}{2} (\|\hat{\boldsymbol{\beta}} + \boldsymbol{\delta}\|_2 - \|\hat{\boldsymbol{\beta}}\|_2)^2$ . Lemma 2 is now proved.  $\square$

Now we prove the theorem. Recall that the model matrix is  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T = [\mathbf{X}_1, \dots, \mathbf{X}_K] \in \mathbb{R}^{n \times p}$ . Define

$$\begin{aligned} G(\boldsymbol{\beta}, \boldsymbol{\gamma}) &= - \sum_{i=1}^n L_i(\boldsymbol{\gamma}) + \sum_{k=1}^K P_k(\|\boldsymbol{\gamma}_k\|_2; \lambda_k) + \frac{1}{2} \|\boldsymbol{\gamma} - \boldsymbol{\beta}\|_2^2 \\ &\quad - \sum_{i=1}^n (b(\mathbf{x}_i^T \boldsymbol{\gamma}) - b(\mathbf{x}_i^T \boldsymbol{\beta})) + \sum_{i=1}^n \mu_i(\boldsymbol{\beta})(\mathbf{x}_i^T \boldsymbol{\gamma} - \mathbf{x}_i^T \boldsymbol{\beta}). \end{aligned} \quad (17)$$

Given  $\boldsymbol{\beta}$ , algebraic manipulations (details omitted) show that minimizing  $G$  over  $\boldsymbol{\gamma}$  is equivalent to

$$\min_{\boldsymbol{\gamma}} \frac{1}{2} \|\boldsymbol{\gamma} - [\boldsymbol{\beta} + \mathbf{X}^T \mathbf{y} - \mathbf{X}^T \boldsymbol{\mu}(\boldsymbol{\beta})]\|_2^2 + \sum_{k=1}^K P_k(\|\boldsymbol{\gamma}_k\|_2; \lambda_k). \quad (18)$$

By Lemma 1, the unique optimal solution can be obtained through multivariate thresholding

$$\boldsymbol{\gamma}_k = \vec{\Theta}_k(\boldsymbol{\beta}_k + \mathbf{X}_k^T \mathbf{y} - \mathbf{X}_k^T \boldsymbol{\mu}(\boldsymbol{\beta}); \lambda_k), \quad 1 \leq k \leq K$$

even though  $P_k$  may be nonconvex. This indicates the iterates defined by (4) can be characterized by  $\boldsymbol{\beta}^{(j+1)} = \arg \min_{\boldsymbol{\gamma}} G(\boldsymbol{\beta}^{(j)}, \boldsymbol{\gamma})$ . Furthermore, for any  $\boldsymbol{\delta} \in \mathbb{R}^p$  we obtain

$$G(\boldsymbol{\beta}^{(j)}, \boldsymbol{\beta}^{(j+1)} + \boldsymbol{\delta}) - G(\boldsymbol{\beta}^{(j)}, \boldsymbol{\beta}^{(j+1)}) \geq \frac{C'_1}{2} \|\boldsymbol{\delta}\|_2^2 + \sum_k q_k(\|\boldsymbol{\beta}_k^{(j+1)} + \boldsymbol{\delta}_k\|_2; \lambda_k), \quad (19)$$

where  $C'_1 = \max(0, 1 - \max_k \mathcal{L}_{\Theta_k})$ , by applying Lemma 2, and noting that  $q_k(\|\boldsymbol{\beta}_k^{(j+1)}\|_2; \lambda_k) = 0$  by definition. Taylor series expansion gives

$$\begin{aligned} \sum_{i=1}^n (b(\mathbf{x}_i^T \boldsymbol{\beta}^{(j+1)}) - b(\mathbf{x}_i^T \boldsymbol{\beta}^{(j)})) - \sum_{i=1}^n \mu_i(\boldsymbol{\beta}^{(j)}) (\mathbf{x}_i^T \boldsymbol{\beta}^{(j+1)} - \mathbf{x}_i^T \boldsymbol{\beta}^{(j)}) \\ = \frac{1}{2} (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)})^T \mathbf{I}(\boldsymbol{\xi}^{(j)}) (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}) \end{aligned}$$

for some  $\boldsymbol{\xi}^{(j)} = \vartheta \boldsymbol{\beta}^{(j)} + (1 - \vartheta) \boldsymbol{\beta}^{(j+1)}$  with  $\vartheta \in (0, 1)$ . Therefore,

$$\begin{aligned} F(\boldsymbol{\beta}^{(j+1)}) + \frac{1}{2} (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)})^T (\mathbf{I} - \mathbf{I}(\boldsymbol{\xi}^{(j)})) (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}) \\ = G(\boldsymbol{\beta}^{(j)}, \boldsymbol{\beta}^{(j+1)}) \leq G(\boldsymbol{\beta}^{(j)}, \boldsymbol{\beta}^{(j)}) - \frac{C'_1}{2} (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)})^T (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}) \\ = F(\boldsymbol{\beta}^{(j)}) - \frac{C'_1}{2} (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)})^T (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}). \end{aligned}$$

(6) follows from the following inequality

$$F(\boldsymbol{\beta}^{(j)}) - F(\boldsymbol{\beta}^{(j+1)}) \geq \frac{1}{2} (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)})^T \left( C'_1 \mathbf{I} + \mathbf{I} - \mathbf{I}(\boldsymbol{\xi}^{(j)}) \right) (\boldsymbol{\beta}^{(j+1)} - \boldsymbol{\beta}^{(j)}).$$

Now assume a subsequence  $\boldsymbol{\beta}^{(j_l)} \rightarrow \boldsymbol{\beta}^*$  as  $l \rightarrow \infty$ . Under the condition  $\rho < \max(1, 2 - \max_k \mathcal{L}_{\Theta_k})$ ,  $C > 0$  and

$$\|\boldsymbol{\beta}^{(j_l+1)} - \boldsymbol{\beta}^{(j_l)}\|_2^2 \leq (F(\boldsymbol{\beta}^{(j_l)}) - F(\boldsymbol{\beta}^{(j_l+1)}))/C \leq (F(\boldsymbol{\beta}^{(j_l)}) - F(\boldsymbol{\beta}^{(j_{l+1})}))/C \rightarrow 0.$$

That is,  $\vec{\Theta}_k(\boldsymbol{\beta}_k^{(j_l)} + \mathbf{X}_k^T \mathbf{y} - \mathbf{X}_k^T \boldsymbol{\mu}(\boldsymbol{\beta}^{(j_l)}); \lambda_k) - \boldsymbol{\beta}_k^{(j_l)} \rightarrow 0$ . From the continuity assumption,  $\boldsymbol{\beta}^*$  is a group  $\Theta$ -estimate satisfying (3).  $\square$

## References

- Agresti, A., 2002. Categorical Data Analysis, 2nd Edition. Wiley Series in Probability and Statistics. Wiley-Interscience.
- Antoniadis, A., 2007. Wavelet methods in statistics: Some recent developments and their applications. *Statistics Surveys* 1, 16–55.
- Bunea, F., Barbu, A., 2009. Dimension reduction and variable selection in case control studies via regularized likelihood optimization. *Electron. J. Stat.* 3, 1257–1287.
- Candes, E. J., Tao, T., 2005. Decoding by linear programming. *IEEE Transactions on Information Theory* 51 (12), 4203–4215.
- Chen, J., Chen, Z., 2008. Extended Bayesian information criterion for model selection with large model space. *Biometrika* 95, 759–771.
- Chen, S., Donoho, D., Saunders, M., 1998. Atomic decomposition by basis pursuit. *SIAM Journal of Scientific Computing* 20 (1), 33–61.
- Chiaretti, S., Li, X., Gentleman, R., Vitale, A., Vignetti, M., Mandelli, F., Ritz, J., Foa, R., 2004. Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood* 103 (7), 2771–2778.
- Daubechies, I., Defrise, M., 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.
- Efron, B., 2009. Empirical bayes estimates for large-scale prediction problems. *JASA* 104, 1015–1028.
- Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. *Annals of Statistics* 32, 407–499.
- Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* 96, 1348–1360.
- Fan, J., Lv, J., 2008. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B* 70 (5), 849–911.

- Friedman, J., Hastie, T., Hofling, H., Tibshirani, R., 2007. Pathwise coordinate optimization. *Annals of Applied Statistics* 1, 302–332.
- Friedman, J., Hastie, T., Tibshirani, R., 2010a. A note on the group lasso and a sparse group lasso. *arXiv:1001.0736v1*.
- Friedman, J., Hastie, T., Tibshirani, R., 2010b. Regularized paths for generalized linear models via coordinate descent. *Journal of Statistical Software* 33 (1).
- Gao, H.-Y., Bruce, A. G., 1997. Waveshrink with firm shrinkage. *Stat. Sin.* 7 (4), 855–874.
- Gasso, G., Rakotomamonjy, A., Canu, S., 2009. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Transactions on Signal Processing* 57 (12), 4686–4698.
- Geman, D., Reynolds, G., 1992. Constrained restoration and the recovery of discontinuities. *IEEE PAMI* 14 (3), 367–383.
- Leng, C., Lin, Y., Wahba, G., 2006. A note on the lasso and related procedures in model selection. *Statist. Sinica* 16 (4), 1273–1284.
- Scholtens, D., von Heydebreck, A., 2005. Analysis of differential gene expression studies. In: Gentleman, R., Carey, V., Huber, W., Irizarry, R., Dudoit, S. (Eds.), *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. Springer, pp. 229–248.
- She, Y., 2009. Thresholding-based iterative selection procedures for model selection and shrinkage. *Electronic Journal of Statistics* 3, 384–415.
- She, Y., Owen, A. B., 2011. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association* 106 (494), 626–639.
- Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *JRSSB* 58, 267–288.
- Tibshirani, R., Hastie, T., Narashiman, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunk centroids of gene expression. *Proc. Nat’l Academy of Sciences USA* 99, 6567–6572.

- Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. *JRSSB* 68, 49–67.
- Zhang, C.-H., 2010. Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.* 38 (2), 894–942.
- Zhang, C.-H., Huang, J., 2008. The sparsity and bias of the Lasso selection in high-dimensional linear regression. *Ann. Statist.* 36, 1567–1594.
- Zhang, T., 2009. Some sharp performance bounds for least squares regression with l1 regularization. *Ann. Statist.* 37, 2109–2144.
- Zhao, P., Yu, B., 2006. On model selection consistency of lasso. *Journal of Machine Learning Research* 7, 2541–2563.
- Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. *JRSSB* 67 (2), 301–320.
- Zou, H., Li, R., 2008. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics* 36 (4), 1509–1533.