Majorization Minimization by Coordinate Descent for Concave Penalized Generalized Linear Models

Dingfeng Jiang[1], Jian Huang[1,2]
1. Department of Biostatistics, University of Iowa
2. Department of Statistics and Actuarial Science, University of Iowa

May 2, 2012

Abstract

Recent studies have demonstrated theoretical attractiveness of a class of concave penalties in variable selection, including the smoothly clipped absolute deviation and minimax concave penalties. The computation of concave penalized solutions in high-dimensional models, however, is a difficult task. We propose a majorization minimization by coordinate descent (MMCD) algorithm for computing the concave penalized solutions in generalized linear models. In contrast to the existing algorithms that use local quadratic or local linear approximation for the penalty function, the MMCD seeks to majorize the negative log-likelihood by a quadratic loss, but does not use any approximation to the penalty. This strategy makes it possible to avoid the computation of a scaling factor in each update of the solutions, which improves the efficiency of coordinate descent. Under certain regularity conditions, we establish the theoretical convergence property of the MMCD. We implement this algorithm for a penalized logistic regression model using the SCAD and MCP penalties. Simulation studies and a data example demonstrate that the MMCD works sufficiently fast for the penalized logistic regression in high-dimensional settings where the number of covariates is much larger than the sample size.

Keywords: logistic regression, $p \gg n$ models, smoothly clipped absolute deviation penalty, minimum concave penalty, variable selection

1 Introduction

Variable selection is a fundamental problem in statistics. A subset of important variables is often pursued to reduce variability and increase interpretability when a model is built. Subset selection is generally adequate when p, the number of variables, is small. By imposing a proper penalty on the number of selected variables, one can perform subset selection using AIC (Akaike (1974)), BIC (Schwarz (1978)), or C_p (Mallows (1973)). However, when p, the number of variables is large, subset selection is computationally infeasible.

Penalized methods have been shown to have attractive theoretical properties for variable selection in $p \gg n$ models. Here n is the sample size. Several important penalties have been proposed. Examples include the ℓ_1 penalty or the least absolute shrinkage and selection operator (Lasso) (Tibshirani (1996)), the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li (2001)) and the minimum concave penalty (MCP) (Zhang (2010)). The SCAD and MCP are concave penalties that possess the oracle properties, meaning that they can correctly select important variables and estimate their coefficients with high probabilities as if the model were known in advance under certain sparsity conditions and other appropriate regularity conditions.

Considerable progress has been made on computational algorithms for penalized regression models. When Efron *et al* (2004) introduced the least angle regression (LARS) approach for variable selection, they showed that a modified version of the LARS can efficiently compute the entire Lasso solution path in a linear model. This modified LARS algorithm is the same as the homotopy algorithm proposed earlier by Osborne, Presnell and Turlach (2000). For concave penalties, Fan and Li (2001) proposed a local quadratic approximation (LQA) algorithm for computing the SCAD solutions. A drawback of LQA is that once a coefficient is set to zero at any iteration step, it permanently stays at zero and the corresponding variable is then removed from the final model. Hunter and Li (2005) used the majorization-minimization (MM) approach to optimize a perturbed version of LQA by bounding the denominator away from zero. How to choose the size of perturbation and how the perturbation affects the sparsity need to be determined in specific models. Zou and Li (2008) proposed a local linear approximation (LLA) algorithm for computing the solutions of SCAD penalized models. The LLA algorithm approximates the concave penalized solutions by repeated using the algorithms for the Lasso penalty. Schifano, Strawderman and Wells (2010) generalized the idea of LLA by MM approach to multiple penalties and proved the convergence properties of their minimization by iterated soft thresholding (MIST) algorithm. Zhang (2010) developed the PLUS algorithm for computing the concave penalized solutions, including the MCP solutions, in linear regression models.

In the last few years, it has been recognized that the coordinate descent algorithm (CDA) can efficiently compute the Lasso solutions in $p \gg n$ models (Friedman, Hastie, Höfling and Tibshirani (2007); Wu and Lange (2008); Friedman, Hastie and Tibshirani (2010)). This algorithm has a long history in applied mathematics and has its roots in the Gauss-Siedel method for solving linear systems (Warge (1963); Ortega and Rheinbold (1970); Tseng (2001)). The CDA optimizes an objective function by working on one coordinate (or a block of coordinates) at a time, iteratively cycling through all coordinates until convergence is reached. It is particularly suitable for the problems that have a simple closed form solution for each coordinate but lack one in higher dimensions. CDA for a Lasso penalized linear model has shown to be very competitive with LARS, especially in high-dimensional cases (Friedman, Hastie, Höfling and Tibshirani (2007); Wu and Lange (2008); Friedman, Hastie and Tibshirani (2010)).

Coordinate descent has also been used in computing the concave penalized solution paths. Breheny and Huang (2011) compared the CDA and LLA algorithms for various combinations of (n, p) and various designs of covariate matrices. They use LARS approach to compute the Lasso solutions when implementing the LLA algorithm. Their results showed that the CDA converges much faster than the LLA-LARS algorithm in the various settings they considered. Mazumder, Friedman and Hastie (2011) demonstrated that the CDA has better convergence properties than the LLA. Breheny and Huang (2011) also proposed an adaptive rescaling technique to overcome the difficulty due to the constantly changing scaling factors in computing the solutions for MCP penalized generalized linear models (GLM). However, the adaptive rescaling approach can not be applied to the SCAD penalty. Furthermore, it is not clear what is the effective concavity applied to the model beforehand using this approach.

We propose a majorization minimization by coordinate descent (MMCD) algorithm for computing the solutions of a concave penalized GLM model, with emphasis on the logistic regression. The MMCD algorithm seeks a closed form solution for each coordinate and avoids the computation of scaling factors by majorizing the loss function. Under reasonable regularity conditions, we establish the convergence property of the MMCD algorithm. The MMCD algorithm is particularly suitable for the logistic regression model due to the fact that a simple and effective majorization can be found.

This paper is organized as follows. Section 2 defines the concave penalized solutions in GLMs. Section 3 describes the proposed MMCD algorithm, explains the benefits of majorization and studies its convergence property. Comparison between the MMCD and several existing algorithms is made in this section. Section 4 implements the MMCD algorithm in a concave penalized logistic model. Simulation studies are performed to compare the MMCD algorithm and its competitors in terms of computational efficiency and selection performance. Section 5 extends the MMCD algorithm to a multinomial model. Concluding remarks are given in section 6.

2 Concave Penalized solutions for GLMs

Let $\{(y_i, \boldsymbol{x}_i)_{i=1}^n\}$ be the observed data, where y_i is a response variable and $\boldsymbol{x}_i$ is a $(p+1)$-dimensional vector of predictors. We consider a GLM with canonical link function, in which y_i relates to $\boldsymbol{x}_i$ through a linear combination $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$, with $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T \in \mathbb{R}^{p+1}$. Here β_0 is the intercept. The conditional density function of y_i given $\boldsymbol{x}_i$ is

$$f_i(y_i) = \exp\{\frac{y_i\theta_i - \psi(\theta_i)}{\phi_i} + c(y_i, \phi)\}. \tag{1}$$

Here $\phi_i > 0$ is a dispersion parameter. The form of $\psi(\theta)$ depends on the specified model. For example, $\psi(\theta) = \log(1 + \exp(\theta))$ in a logistic model. Consider the (scaled) negative log-likelihood function as a loss function $\ell(\boldsymbol{\beta})$,

$$\ell(\boldsymbol{\beta}) \propto \frac{1}{n} \sum_{i=1}^{n} \{\psi(\boldsymbol{x}_i^T \boldsymbol{\beta}) - y_i \boldsymbol{x}_i^T \boldsymbol{\beta}\}. \tag{2}$$

We assume $x_{i0} = 1, 1 \le i \le n$. For the other p variables, we assume they are standardized, that is, $\|\boldsymbol{x}^j\|_2^2/n = 1$ with $\boldsymbol{x}^j = (x_{1j}, ..., x_{nj})^T, 1 \le j \le p$. Here $\|\mathbf{v}\|_2$ is the ℓ_2 norm of a n-dimensional vector $\mathbf{v}$. The standardization allows the penalization to be evenly applied to each variable.

Define the concave penalized GLM criterion as

$$Q(\boldsymbol{\beta}; \lambda, \gamma) = \ell(\boldsymbol{\beta}) + \sum_{j=1}^{p} \rho(|\beta_j|; \lambda, \gamma), \tag{3}$$

where ρ is a penalty function. Note that the intercept β_0 is not penalized. We consider two concave penalties, SCAD and MCP. The SCAD penalty (Fan and Li (2001)) is defined as

$$\rho(t; \lambda, \gamma) = \begin{cases} \lambda|t|, & |t| \le \lambda; \\ \frac{\gamma\lambda|t| - 0.5(t^2+\lambda^2)}{\gamma-1}, & \lambda < |t| \le \gamma\lambda; \\ \frac{\lambda^2(\gamma^2-1)}{2(\gamma-1)}, & |t| > \gamma\lambda, \end{cases} \tag{4}$$

with $\lambda \ge 0$ and $\gamma > 2$. The MCP penalty (Zhang (2010)) is defined as

$$\rho(t; \lambda, \gamma) = \lambda \int_0^{|t|} (1 - \frac{x}{\gamma\lambda})_+ dx = \begin{cases} \lambda|t| - \frac{|t|^2}{2\gamma}, & |t| \le \lambda\gamma; \\ \frac{1}{2}\lambda^2\gamma, & |t| > \lambda\gamma, \end{cases} \tag{5}$$

with $\lambda \ge 0$ and $\gamma > 1$. Here $x_+ = x1\{x \ge 0\}$ denotes the non-negative part of x. For both SCAD and MCP, the regularization parameter γ controls the degree of concavity, with a smaller γ corresponding to a more concave shaped penalty. Both penalties begin by applying the same degree of penalization as Lasso, and then gradually reduce the penalization to zero as $|t|$ gets larger. When $\gamma \to \infty$, both SCAD and MCP converge to the ℓ_1 penalty. The SCAD and MCP penalties are illustrated in the middle and right panels of Figure 1.

To have a basic understanding of these penalties, consider a thresholding operator defined as the solution to a penalized univariate linear regression,

$$\hat{\theta}(\lambda,\gamma) = \underset{\theta}{\operatorname{argmin}} \big\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - x_i\theta)^2 + \rho(\theta;\lambda,\gamma) \big\}.$$

Let $\hat{\theta}_{LS} = \sum_{i=1}^{n} x_i y_i / \sum_{i=1}^{n} x_i^2$ be the least squares solution. Denote the soft-thresholding operator by $S(t,\lambda) = \text{sgn}(t)(|t| - \lambda)_+$ for $\lambda > 0$ (Donoho and Johnstone (1994)). Then for SCAD and MCP, $\hat{\theta}(\lambda,\gamma)$ has a closed form solution as follows,

$$\begin{aligned}
\text{For } \gamma > 2, \quad \hat{\theta}_{SCAD}(\lambda,\gamma) &= \begin{cases} S(\hat{\theta}_{LS},\lambda), & |\hat{\theta}_{LS}| \le 2\lambda, \\ \frac{\gamma-1}{\gamma-2} S(\hat{\theta}_{LS}, \lambda\gamma/(\gamma-1)), & 2\lambda < |\hat{\theta}_{LS}| \le \gamma\lambda, \\ \hat{\theta}_{LS}, & |\hat{\theta}_{LS}| > \lambda\gamma, \end{cases} \\
\text{For } \gamma > 1, \quad \hat{\theta}_{MCP}(\lambda,\gamma) &= \begin{cases} \frac{\gamma}{\gamma-1} S(\hat{\theta}_{LS},\lambda), & |\hat{\theta}_{LS}| \le \lambda\gamma, \\ \hat{\theta}_{LS}, & |\hat{\theta}_{LS}| > \lambda\gamma. \end{cases}
\end{aligned} \qquad (6)$$

Observe that both SCAD and MCP use the LS solution if $|\hat{\theta}_{LS}| > \lambda\gamma$; MCP only applies a scaled soft-thresholding operation for $|\hat{\theta}_{LS}| \le \lambda\gamma$ while SCAD apply a soft-thresholding operation to $|\hat{\theta}_{LS}| < 2\lambda$ and a scaled soft-thresholding operation to $2\lambda < |\hat{\theta}_{LS}| \le \lambda\gamma$. These thresholding operators will be the basic building blocks of the proposed MMCD algorithm described below.

Figure 1 shows the penalty functions and the thresholding functions of Lasso (left panel), SCAD (middle panel) and MCP (right panel), respectively. The first row shows the penalty functions and the second row shows the thresholding operator functions. Lasso penalizes all the variables without distinction. SCAD and MCP gradually reduce the degree of penalization for large coefficients.

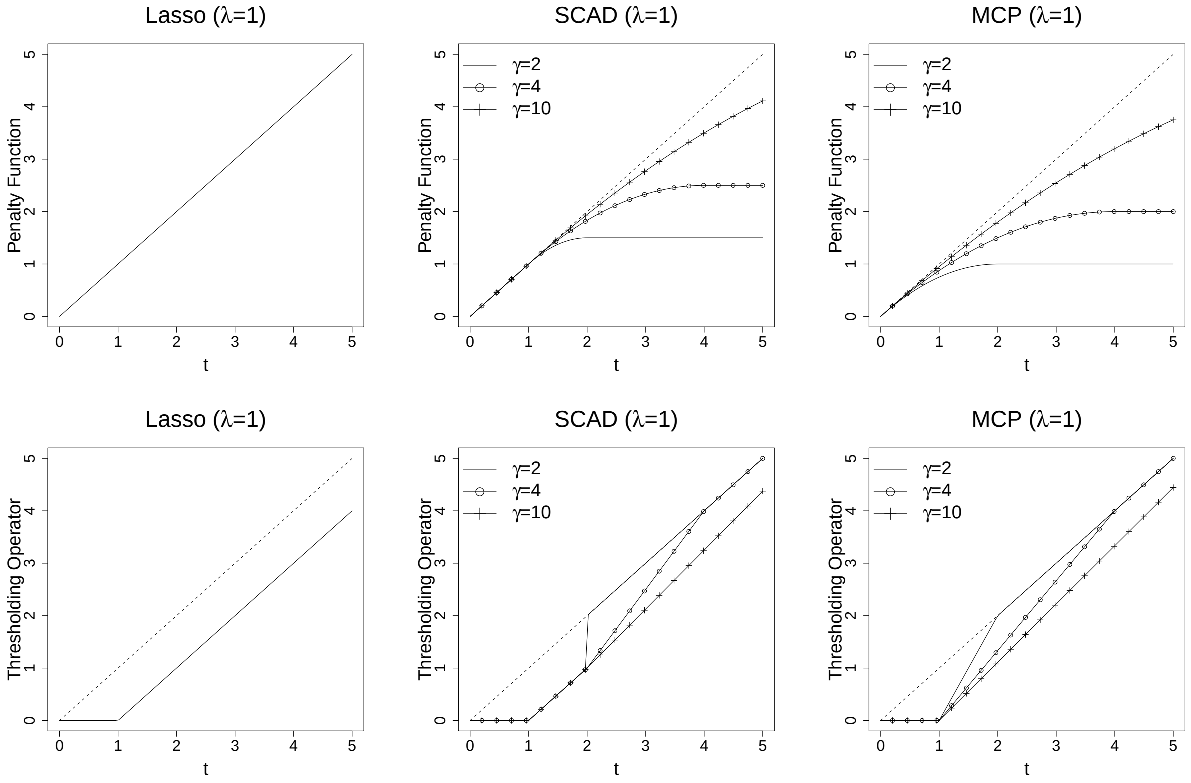

Figure 1: Penalty functions and threshold operator functions of Lasso(left), SCAD(middle) and MCP(right). The first row shows the penalty functions and the second row shows the operator functions. Lasso shrinks all coefficients without distinction. SCAD and MCP reduce the rate of penalization for large coefficients. Both MCP and SCAD converge to Lasso if $\gamma \to +\infty$.

3 Majorization Minimization by Coordinate Descent

3.1 The MMCD Algorithm

For a GLM, a quadratic approximation of the loss function $\ell(\boldsymbol{\beta})$ in a neighborhood of a given estimate $\tilde{\boldsymbol{\beta}}$ leads to an iteratively reweighed least squares (IRLS) form of the loss function as follows

$$\ell^s(\boldsymbol{\beta}|\tilde{\boldsymbol{\beta}}) = \frac{1}{2n}\sum_{i=1}^{n} w_i(z_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2, \tag{7}$$

with $w_i(\tilde{\boldsymbol{\beta}}) = \ddot{\psi}(\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}})$ and $z_i(\tilde{\boldsymbol{\beta}}) = \ddot{\psi}(\boldsymbol{x}_i\tilde{\boldsymbol{\beta}})^{-1}\{y_i - \dot{\psi}(\boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}})\} + \boldsymbol{x}_i^T\tilde{\boldsymbol{\beta}}$, where $\dot{\psi}(\theta)$ and $\ddot{\psi}(\theta)$ are the first and second derivatives of $\psi(\theta)$ with respect to (w.r.t.) θ. Using $\ell^s(\boldsymbol{\beta}|\tilde{\boldsymbol{\beta}})$ in the criterion function, the CDA updates the jth coordinate by fixing the remaining k $(k \neq j)$ coordinates. Let $\hat{\boldsymbol{\beta}}_j^m = (\hat{\beta}_0^{m+1}, ..., \hat{\beta}_j^{m+1}, \hat{\beta}_{j+1}^m, ..., \hat{\beta}_p^m)^T$, the CDA updates $\hat{\boldsymbol{\beta}}_{j-1}^m$ to $\hat{\boldsymbol{\beta}}_j^m$ by minimizing the criterion

$$\begin{aligned}\hat{\beta}_j^{m+1} &= \underset{\beta_j}{\operatorname{argmin}}\, Q^s(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m) \\ &= \underset{\beta_j}{\operatorname{argmin}}\, \frac{1}{2n}\sum_{i=1}^n w_i(z_i - \sum_{s<j} x_{ij}\hat{\beta}_s^{m+1} - x_{ij}\beta_j - \sum_{s>j} x_{ij}\hat{\beta}_s^m)^2 + \rho(|\beta_j|;\lambda,\gamma), \qquad (8)\end{aligned}$$

where w_i and z_i depend on $(\hat{\boldsymbol{\beta}}_{j-1}^m, \boldsymbol{x}_i, y_i)$. The jth coordinate-wise minimizer is then obtained by solving the equation,

$$\frac{1}{n}\sum_{i=1}^n w_i x_{ij}^2\beta_j + \rho'(|\beta_j|)\text{sgn}(\beta_j) - \frac{1}{n}\sum_{i=1}^n w_i x_{ij}(z_i - \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_{j-1}^m) - \frac{1}{n}\sum_{i=1}^n w_i x_{ij}^2\hat{\beta}_j^m = 0, \qquad (9)$$

with $\rho'(|t|)$ the first derivative of $\rho(|t|)$ w.r.t $|t|$ and $\text{sgn}(x) = 1, -1$ or $\in [-1, 1]$ for $x > 0, < 0$ or $x = 0$.

For the MCP penalty, directly solving (9) gives the jth coordinate-wise solution as follows,

$$\hat{\beta}_j^{m+1} = \begin{cases} \frac{S(\tau_j,\lambda)}{\delta_j - 1/\gamma}, & |\tau_j| \leq \delta_j\gamma\lambda, \\ \frac{\tau_j}{\delta_j}, & |\tau_j| > \delta_j\gamma\lambda, \end{cases} \qquad (10)$$

where the scaling factor $\delta_j \triangleq n^{-1}\sum_{i=1}^n w_i x_{ij}^2$ and $\tau_j = n^{-1}\sum_{i=1}^n w_i x_{ij}(z_i - \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_{j-1}^m) + \delta_j\hat{\beta}_j^m$. In a linear model, $w_i = 1, i = 1, ..., n$, thus the scaling factor $\delta_j = n^{-1}\sum_{i=1}^n w_i x_{ij}^2 = 1$ for standardized predictors. In a GLM, however, the dependence of w_i on $(\hat{\boldsymbol{\beta}}_{j-1}^m, \boldsymbol{x}_i, y_i)$ causes the scaling factor δ_j to change from iteration to iteration. This is problematic because $\delta_j - 1/\gamma$ can be very small and is not guaranteed to be positive. Thus direct application of CDA may not be numerically stable and can lead to unreasonable solutions.

To overcome this difficulty, Breheny and Huang (2011) proposed an adaptive rescaling approach, which uses

$$\hat{\beta}_j^{m+1} = \begin{cases} \frac{S(\tau_j,\lambda)}{\delta_j(1-1/\gamma)}, & |\tau_j| \leq \gamma\lambda, \\ \frac{\tau_j}{\delta_j}, & |\tau_j| > \gamma\lambda, \end{cases} \tag{11}$$

for the jth coordinate-wise solution. This is equivalent to apply a new regularization parameter $\gamma^* = \gamma/\delta_j$ at each coordinate-wise iteration. Hence, the effective regularization parameters are not the same for the penalized variables and are not known until the algorithm reaches convergence. Numerically, the scaling factor δ_j requires extra computation, which is not desirable for large p. In addition, small δ_j could also cause convergence issues. For SCAD, the adaptive rescaling approach cannot be adopted because the scaled soft-thresholding operation only applies to the middle clause of the SCAD thresholding operator as shown in (6).

The MMCD algorithm seeks a majorization of scaling factor δ_j. For standardized predictors, this is equivalent to finding a uniform upper bound of the weights $w_i = \ddot{\psi}(\boldsymbol{x}_i^T\boldsymbol{\beta}), 1 \leq i \leq n$. In principle, we can have a sequence of constants C_i such that $C_i \geq w_i$ for $i = 1, ..., n$ and use $M_j = \sum C_i x_{ij}^2/n$ to majorize δ_j. The standardization, however, allows a single M to majorize all the p scaling factors. Observe that in a GLM, the scaling factor δ_j is equal to the second partial derivative of the loss function, i.e. $\nabla_j^2\ell(\boldsymbol{\beta}) = \sum \ddot{\psi}(\boldsymbol{x}_i^T\boldsymbol{\beta})x_{ij}^2/n = \sum w_i x_{ij}^2/n$. Hence, a majorization of w_i results in a majorization of $\nabla_j^2\ell(\boldsymbol{\beta})$. For simplicity, we put the boundedness condition, $\delta_j \leq M$ on the term $\nabla_j^2\ell(\boldsymbol{\beta})$ rather than the individual w_i.

From the perspective of MM algorithm, the majorization of δ_j is equivalent to finding a surrogate function $\ell^{MM}(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m)$ that majorizes $\ell^s(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m)$ when optimizing the criterion function w.r.t the jth coordinate, where

$$\ell^{MM}(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m) = \ell(\hat{\boldsymbol{\beta}}_{j-1}^m) + \nabla_j\ell(\hat{\boldsymbol{\beta}}_{j-1}^m)(\beta_j - \hat{\beta}_j^m) + \frac{1}{2}M(\beta_j - \hat{\beta}_j^m)^2, \tag{12}$$

and

$$\ell^s(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m) = \ell(\hat{\boldsymbol{\beta}}_{j-1}^m) + \nabla_j\ell(\hat{\boldsymbol{\beta}}_{j-1}^m)(\beta_j - \hat{\beta}_j^m) + \frac{1}{2}\nabla_j^2\ell(\hat{\boldsymbol{\beta}}_{j-1}^m)(\beta_j - \hat{\beta}_j^m)^2, \tag{13}$$

with the second partial derivative $\nabla_j^2 \ell(\hat{\boldsymbol{\beta}}_{j-1}^m)$ in the Taylor expansion being replaced by its upper bound M. Note that the majorization is applied coordinate-wisely to better fit the CDA approach. The descent property of the MM approach ensures that iterative minimization of $\ell^{MM}(\beta_j | \hat{\boldsymbol{\beta}}_{j-1}^m)$ leads to a descent sequence of the original objective function. For more details about the MM algorithm, we refer to Lange, Hunter, and Yang (2000); Hunter and Lange (2004).

Given the majorization of δ_j, some algebra shows that the jth ($j = 1, ..., p$) coordinate-wise solutions are

$$\text{SCAD: } \hat{\beta}_j^{m+1} = \begin{cases} \frac{1}{M} S(\tau_j, \lambda), & |\tau_j| \le (1+M)\lambda, \\ \frac{S(\tau_j, \gamma\lambda/(\gamma-1))}{M - 1/(\gamma-1)}, & (1+M)\lambda < |\tau_j| \le M\gamma\lambda, \\ \frac{1}{M}\tau_j & |\tau_j| > M\gamma\lambda, \end{cases} \tag{14}$$

$$\text{MCP: } \hat{\beta}_j^{m+1} = \begin{cases} \frac{S(\tau_j, \lambda)}{M - 1/\gamma} & |\tau_j| \le M\gamma\lambda, \\ \frac{1}{M}\tau_j & |\tau_j| > M\gamma\lambda, \end{cases} \tag{15}$$

with $\tau_j = M\hat{\beta}_j^m + n^{-1}\sum_{i=1}^n x_{ij}(y_i - \dot{\psi}(\boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_{j-1}^m))$. The solution of the intercept is

$$\hat{\beta}_0^{m+1} = \tau_0/M, \tag{16}$$

with $\tau_0 = M\hat{\beta}_0^m + n^{-1}\sum_{i=1}^n x_{i0}(y_i - \dot{\psi}(\boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}^m))$, where $\hat{\boldsymbol{\beta}}^m = (\hat{\beta}_0^m, \hat{\beta}_1^m, ..., \hat{\beta}_p^m)^T$. In the expressions (14) and (15), we want to ensure the denominators are positive, that is, $M - 1/(\gamma - 1) > 0$ and $M - 1/\gamma > 0$. This naturally leads to the constraint on the penalty, $\inf_t \rho''(|t|; \lambda, \gamma) > -M$, where $\rho''(|t|; \lambda, \gamma)$ is the second derivative of $\rho(|t|; \lambda, \gamma)$ w.r.t. $|t|$. For SCAD and MCP, this condition is satisfied by choosing a proper γ. For SCAD, $\inf_t \rho''(|t|; \lambda, \gamma) = -1/(\gamma - 1)$; for MCP, $\inf_t \rho''(|t|; \lambda, \gamma) = -1/\gamma$. Therefore, we require $\gamma > 1 + 1/M$ for SCAD and $\gamma > 1/M$ for MCP.

The MMCD algorithm can gain further efficiency by adopting the following tip. Let $\boldsymbol{\eta} = (\eta_1, ..., \eta_n)^T$ and $X = (\boldsymbol{x}_1^T, ..., \boldsymbol{x}_n^T)^T$, and $\hat{\boldsymbol{\eta}}_j^m = X\hat{\boldsymbol{\beta}}_j^m$ be the linear component corresponding to

$\hat{\boldsymbol{\beta}}_j^m$. Further efficiency can be achieved by using the equation

$$\hat{\boldsymbol{\eta}}_{j+1}^m = \hat{\boldsymbol{\eta}}_j^m + \boldsymbol{x}^{j+1}(\hat{\beta}_{j+1}^{m+1} - \hat{\beta}_{j+1}^m) = \hat{\boldsymbol{\eta}}_j^m + (\hat{\boldsymbol{\beta}}_{j+1}^m - \hat{\boldsymbol{\beta}}_j^m)\boldsymbol{x}^{j+1}. \tag{17}$$

This equation turns a $O(np)$ operation into a $O(n)$ one. Since this step is involved in each iteration for each coordinate, this simple step turns out to be significant in reducing the computational cost.

We now summarize the MMCD procedure for a given (λ, γ). Assume the conditions below hold:

(a) The second partial derivative of $\ell(\boldsymbol{\beta})$ w.r.t. β_j is uniformly bounded for standardized X, i.e. there exists a real number $M > 0$ such that $\nabla_j^2 \ell(\boldsymbol{\beta}) \leq M$ for $j = 0, ..., p$.

(b) $\inf_t \rho''(|t|; \lambda, \gamma) > -M$, with $\rho''(|t|; \lambda, \gamma)$ being the second derivative of $\rho(|t|; \lambda, \gamma)$ w.r.t. $|t|$.

The MMCD algorithm proceeds as follows,

1. Given an initial value $\hat{\boldsymbol{\beta}}^0$, compute the corresponding linear component $\hat{\boldsymbol{\eta}}^0$.

2. For $m = 0, 1, ...$, update the intercept by form (16), and use the solution form (14) or (15) to update $\hat{\boldsymbol{\beta}}_j^m$ to $\hat{\boldsymbol{\beta}}_{j+1}^m$ for the penalized variables. After each iteration, also compute the corresponding linear component $\hat{\boldsymbol{\eta}}_{j+1}^m$ using (17). Cycle through all the coordinates for $j = 0, ..., p$ such that $\hat{\boldsymbol{\beta}}^m$ is updated to $\hat{\boldsymbol{\beta}}^{m+1}$.

3. Check the convergence criterion. If converges then stop iteration, otherwise repeat step 2 until converges.

We use the convergence criterion $\|\hat{\boldsymbol{\beta}}^{m+1} - \hat{\boldsymbol{\beta}}^m\|_2 / (\|\hat{\boldsymbol{\beta}}^m\|_2 + \delta) < \varepsilon$. We choose $\delta = 0.01$ and $\varepsilon = 0.001$ unless mentioned otherwise.

3.2 Convergence Analysis

In this section, we present a convergence result for the MMCD algorithm. Theorem 1 establishes that under certain regularity conditions, the MMCD solution converges to a minimum of the objective function.

Theorem 1. *Consider the objective function (3), where the given data* $(\boldsymbol{y}, X)$ *lies on a compact set and no two columns of* X *are identical. Suppose the penalty* $\rho(|t|;\lambda,\gamma) \equiv \rho(t)$ *satisfies* $\rho(t) = \rho(-t)$, $\rho'(|t|)$ *is non-negative, uniformly bounded, with* $\rho'(|t|)$ *being the first derivative (assuming existence) of* $\rho(|t|)$ *w.r.t.* $|t|$. *Also assume that two conditions stated in the MMCD algorithm hold.*

Then the sequence generated by the MMCD algorithm $\{\boldsymbol{\beta}^m\}$ *converges to a local minimum of the function* $Q(\boldsymbol{\beta})$.

Note that the condition on $(\boldsymbol{y}, X)$ is a mild assumption. The standardization of columns of X can be performed as long as no columns are zero. The proof of theorem 1 is given in Appendix. It extends the work of Mazumder, Friedman and Hastie (2011) to cover more general loss functions other than the least squares.

3.3 Comparison with existing algorithms

The LQA (Fan and Li (2001)), perturbed LQA (Hunter and Li (2005)), LLA (Zou and Li (2008)) and MIST (Schifano, Strawderman and Wells (2010)) algorithms share the same spirit in the sense that they all optimize a surrogate function instead of the original penalty $\rho(|t|;\lambda,\gamma)$. Figure 2 illustrates the three majorizations of SCAD. The left panel of Figure 2 is majorized at $t = 3$, while the right one is majorized at $t = 1$. In both plots, $\gamma = 4$ and $\lambda = 2$ are chosen for better illustration purpose. To apply these methods, we need to approximate both the loss and penalty functions. This does not take full advantage of CDA. Indeed, the approximation of the penalty requires additional iterations for convergence and is not necessary, since exact coordinate-wise solution exists. Thus MMCD uses the exact form of the penalty and only majorizes the loss.

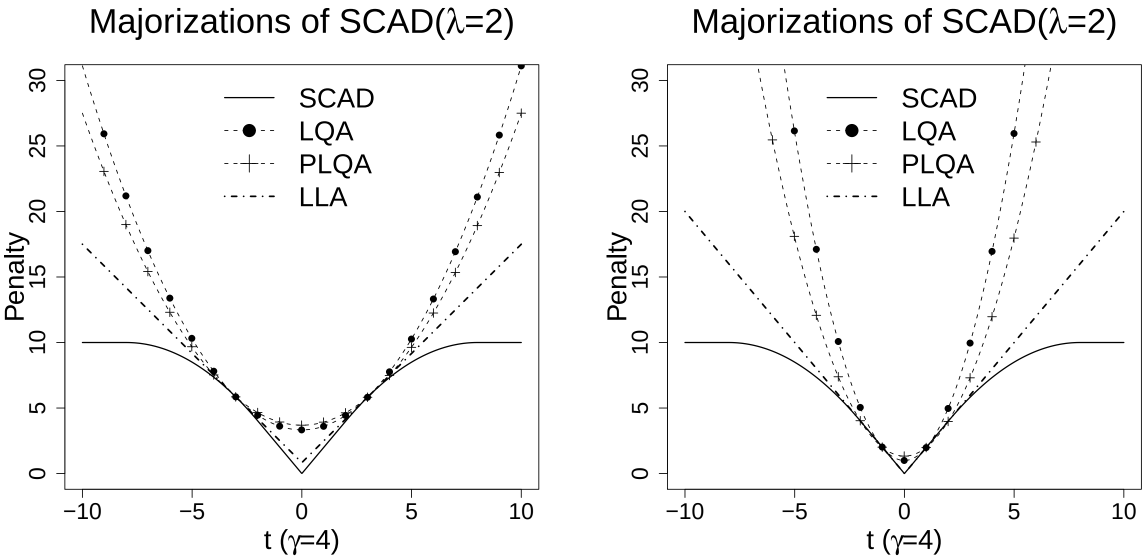

Figure 2: SCAD penalty and its majorizations, LQA, Perturbed LQA (PLQA) and LLA. The left plot is majorized at $t = 3$, the right one is majorized at $t = 1$. All the curves are plotted using $\gamma = 4$ and $\lambda = 2$ for better illustration effect.

4 The MMCD for Penalized Logistic Regression

As mentioned in the introduction, the MMCD algorithm is particularly suitable for logistic regression, which is one of the most widely used models in biostatistical applications. For a logistic regression model, the response $\boldsymbol{y}$ is a vector of 0 or 1 with 1 indicating the event of interest. The first and second derivatives of the loss function are $\nabla_j \ell(\hat{\boldsymbol{\beta}}) = -(\boldsymbol{x}^j)^T(\boldsymbol{y} - \hat{\boldsymbol{\pi}})/n$ and $\nabla_j^2 \ell(\hat{\boldsymbol{\beta}}) = n^{-1}\sum w_i x_{ij}^2$, with $w_i = \hat{\pi}_i(1-\hat{\pi}_i)$ and $\hat{\pi}_i$ being the estimated probability of ith observation given the current estimate $\hat{\boldsymbol{\beta}}$, i.e. $\hat{\pi}_i = 1/(1+\exp(-\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}))$. For any $0 \leq \pi \leq 1$, we have $\pi(1-\pi) \leq 1/4$. Hence the upper bound of $\nabla_j^2 \ell(\hat{\boldsymbol{\beta}})$ is $M = 1/4$ for standardized $\boldsymbol{x}^j$. Correspondingly $\tau_j = 4^{-1}\hat{\beta}_j + n^{-1}(\boldsymbol{x}^j)^T(\boldsymbol{y} - \hat{\boldsymbol{\pi}})$ for $j = 0, ..., p$. By condition (ii), we require $\gamma > 5$ for SCAD and $\gamma > 4$ for MCP.

4.1 Computation of Solution Surface

A common practice in applying the SCAD and MCP penalties is to calculate the solution path in λ for a fixed value of κ. For example, for linear regression models with standardized variables, it has been suggested one uses $\gamma \approx 3.7$ for SCAD (Fan and Li (2001)) and $\gamma \approx 2.7$ (Zhang (2010)) for MCP. However, in a GLM model including the logistic regression, these values may not be appropriate. Therefore, we use a data driven procedure to choose γ together with λ. This requires the computation of solution surface over a two-dimensional grid of (λ, γ). We re-parameterize $\kappa = 1/\gamma$ to facilitate the description of the approach for computing the solution surface. By condition (ii) of the MMCD algorithm, we require $\kappa \in [0, \kappa_{\max}]$, with $\kappa_{\max} = 1/5$ for SCAD and $\kappa_{\max} = 1/4$ for MCP. When $\kappa = 0$, both the SCAD and MCP become the Lasso.

Define the grid values for a rectangle in $[0, \kappa_{\max}) \times [\lambda_{\min}, \lambda_{\max}]$ to be $0 = \kappa_1 \leq \kappa_2 \leq \cdots \leq \kappa_K < \kappa_{\max}$ and $\lambda_{\max} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_V = \lambda_{\min}$. The number of grid points K and V are pre-specified. In our implementation, the κ-grid points are uniform in normal scale while those for λ are uniform in log scale. The $\lambda_{\max}$ is the smallest value of λ such that $\hat{\beta}_j = 0$, $j = 1, ..., p$. For a logistic model, $\lambda_{\max} = n^{-1}\max_j|(\boldsymbol{x}^j)^T(\boldsymbol{y} - \hat{\boldsymbol{\pi}})|$ for every κ_k with $\hat{\boldsymbol{\pi}} = \bar{y}\boldsymbol{J}$ and $\boldsymbol{J}$ being a vector whose elements are all equal to 1. Let $\lambda_{\min} = \epsilon\lambda_{\max}$, with $\epsilon = 0.0001$ if $n > p$ and $\epsilon = 0.01$ otherwise. The solution surface is then calculated over the rectangle $[0, \kappa_{\max}) \times [\lambda_{\min}, \lambda_{\max}]$. Denote the MMCD solution for a given (κ_k, λ_v) by $\hat{\boldsymbol{\beta}}_{\kappa_k, \lambda_v}$.

We follow the approach of Mazumder, Friedman and Hastie (2011) to compute the solution surface by initializing the algorithm at the Lasso solutions on a grid of λ values. The Lasso solutions correspond to $\kappa = 0$. Then for each point in the grid of λ values, we compute the solutions on a grid of κ values starting from $\kappa = 0$, using the solution at the previous point as the initial value for the current point. The details of the approach are as follows.

(1) First compute the Lasso solution along λ. When computing $\hat{\boldsymbol{\beta}}_{\kappa_0, \lambda_{v+1}}$, using $\hat{\boldsymbol{\beta}}_{\kappa_0, \lambda_v}$ as the

initial value in the MMCD algorithm.

(2) For a given λ_v, compute the solution along κ. Here we use $\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$ as the initial value in computing the solution $\hat{\boldsymbol{\beta}}_{\kappa_{k+1},\lambda_v}$.

(3) Cycle through $v = 1, ..., V$ for step (2) to complete the solution surface.

Define a variable to be a causal one if its coefficient $\beta \neq 0$; otherwise call it to be a null variable. Figure (3) presents the solution paths of a causal variable with $\beta = 2$ (plot (a)) and a null variable with $\beta = 0$ (plot (b)) along κ using the MCP penalty. Observe that the estimates could change substantially when κ crosses certain values. This justifies our treating κ as a tuning parameter since a pre-specified κ might not give the optimal results. This is the reason why we prefer a data-driven procedure to choose both κ and λ.

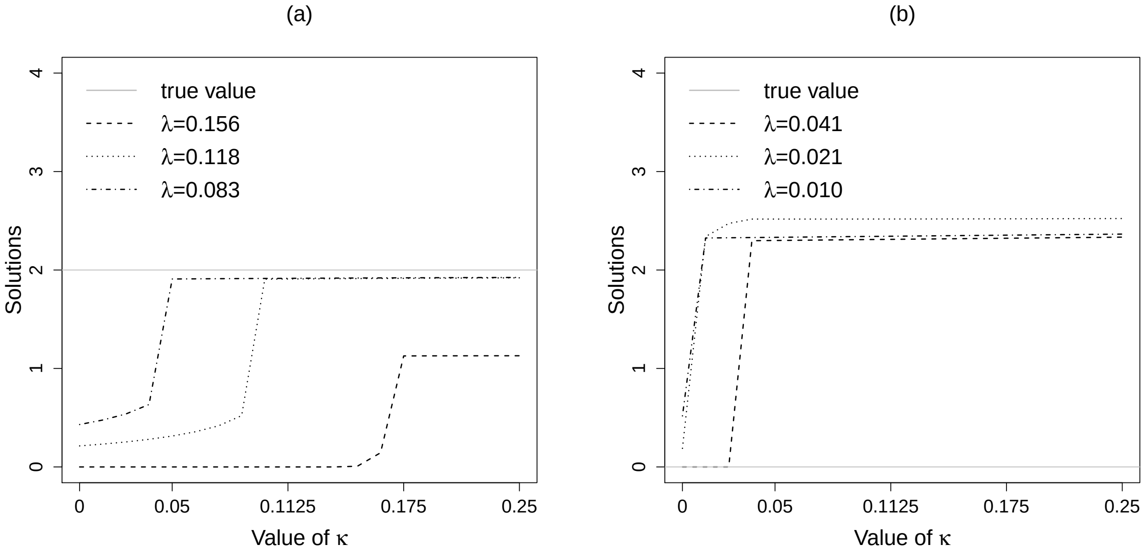

Figure 3: Plots of solution paths along κ. Plot (a) shows the paths for a causal variable with $\beta = 2$, while (b) shows the paths for a null variable with $\beta = 0$. Observe that the estimates could change substantially when κ crosses certain threshold values.

4.2 Design of simulation study

Let Z be the design matrix of the covariates, that is, it is a sub-matrix of X with its first column removed. Let $A_0 \equiv \{1 \leq j \leq p : \beta_j \neq 0\}$ be the set of causal variables with dimension p_0. We fix $p_0 = 10$, $\beta_0 = 0.0$ and the coefficients of A_0 to be $(0.6, -0.6, 1.2, -1.2, 2.4, -0.6, 0.6, -1.2, 1.2, -2.4)^T$ such that the signal-to-noise ratio (SNR), defined as $\text{SNR} = \sqrt{\boldsymbol{\beta}^T X^T X \boldsymbol{\beta} / n\sigma^2}$, is approximately in the range of $(3, 4)$. The covariates are generated from a multivariate normal distribution with zero means and variance $\sigma^2 \boldsymbol{\Sigma}$, with $\sigma^2 = 1$ and $\boldsymbol{\Sigma}$ being a positive-definite matrix with dimension $p \times p$. The outcomes $\boldsymbol{y}$ are generated from the Bernoulli distribution with $y_i \sim \text{Bernoulli}(1, p_i)$, with $p_i = \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i)/(1 + \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i))$ for $i = 1, ..., n$.

We consider five types of correlation structures for $\boldsymbol{\Sigma}$.

(a) Independent structure (IN) for the p penalized variables. Here $\boldsymbol{\Sigma} = I_p$, with I_p being the identity matrix of dimension $p \times p$.

(a) Separate structure (SP). The causal and null variables are independent. Let $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ being the covariance matrix for the causal variables and the null variables, respectively, then $\boldsymbol{\Sigma} = \text{block diagonal}(\boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1)$. Within each set of variables, we assume a compound symmetry structure, that is, $\rho(x_{ij}, x_{ik}) = \rho$ for $j \neq k$.

(c) Partial correlated structure (PC), i.e. part of the causal variables are correlated with part of the null variables. Specifically, $\boldsymbol{\Sigma} = \text{block diagonal}(\boldsymbol{\Sigma}_a, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_c)$, with $\boldsymbol{\Sigma}_a$ being the covariance matrix for the first 5 causal variables; $\boldsymbol{\Sigma}_b$ being the covariance matrix for the remaining 5 causal variables and 5 null variables; $\boldsymbol{\Sigma}_c$ being the covariance matrix for the remaining null variables. We also assume a compound symmetry structure within $\boldsymbol{\Sigma}_a$, $\boldsymbol{\Sigma}_b$, $\boldsymbol{\Sigma}_c$.

(d) First-order autoregressive (AR) structure, i.e. $\rho(x_{ij}, x_{ik}) = \rho^{(|j-k|)}$, for $j \neq k$.

(e) Compound symmetry (CS) structure for p variables.

4.3 Numerical implementation of the LLA algorithm

The basic idea of the LLA is to approximate a concave penalty $\rho(|\beta_j|;\gamma,\lambda)$ by $\dot\rho(|\hat\beta_j^m|;\gamma,\lambda)|\beta_j|$ based on the current estimate $\hat{\boldsymbol\beta}^m$. For the logistic regression, we also use a quadratic approximation (7) for the loss based on $\hat{\boldsymbol\beta}^m$. To compute $\hat{\boldsymbol\beta}^{m+1}$, we minimize a Lasso-like criterion

$$\ell^s(\boldsymbol\beta|\hat{\boldsymbol\beta}^m)+\sum_{j=1}^{p}\dot\rho(\hat\beta_j^m;\gamma,\lambda)|\beta_j|. \tag{18}$$

To compare the MMCD with the LLA, we implemented the LLA algorithm in two ways. The first implementation strictly follows the description in Zou and Li (2008). This uses working data based on the current estimate and separates the design matrix into two parts, $U=\{j:\dot\rho(|\hat\beta_j^m|;\gamma,\lambda)=0\}$ and $V=\{j:\dot\rho(|\hat\beta_j^m|;\gamma,\lambda)\neq 0\}$ for a current estimate $\hat{\boldsymbol\beta}^m$, with $\dot\rho(t)$ being the derivative of $\rho(\cdot)$. The computation of $\hat{\boldsymbol\beta}^{m+1}$ involves $(X_U^{*T}X_U^*)^{-1}$, with $X_U^*=(X_j:j\in U)$ being the design matrix of variables in U. Hence, the solution could be non-unique if $n<p_U$ with p_U being the number of variables in U. Therefore, this approach generally only works in the settings with $n>p$.

In the second implementation, we use the coordinate descent algorithm to minimize (18). This implementation can handle data sets with $p\gg n$. We call this implementation LLA-CD algorithm below.

Since both implementations require an initial estimate $\hat{\boldsymbol\beta}$ to approximate the penalty, we use the Lasso solutions to initiate the computation along κ for the LLA and LLA-CD algorithms. The LLA, adaptive rescaling, LLA-CD and MMCD algorithms were programmed in Fortran with similar programming structures for fair comparison. We observe that the adaptive rescaling algorithm does not converge within $1,000$ iterations if κ_{max} is large. Hence, we set $\kappa_{max}=0.25$ for the adaptive rescaling algorithm in our computation. In the simulation, we set correlation coefficient $\rho=0.5$, the number of grids $K=10$, $V=100$ and the convergence criterion $\varepsilon=0.001$ if $n>p$ and $\varepsilon=0.01$ if $n<p$.

4.4 Comparison of computational efficiency

Since the adaptive rescaling approach can only be applied to the MCP penalty, we compare the efficiency of the LLA, adaptive rescaling, LLA-CD and MMCD algorithms for MCP penalized logistic regression models. The computation is done on Inter Xeon CPU (W3540@2.93GHZ) machines with Ubuntu 10.04 operating system (Kernel version 2.6). We consider two settings with $n < p$ and $n < p$.

Figure 4 shows the average elapsed times measured in seconds based on 100 replications for $p = 100, 200$ and 500 with a fixed sample size $n = 1,000$. Observe that the time for the adaptive rescaling algorithm increases dramatically when $n = 1,000$ and $p = 500$. This suggests that the ratio of p/n has a greater impact on the efficiency of the adaptive rescaling algorithm. LLA-CD algorithm is also impacted by the p/n ratio to a certain extend. MMCD and LLA are fairly stable to the change of p/n ratio. Overall, the MMCD algorithm is the fastest one. It is worth noting that in the setting with ($n = 1,000$, $p = 500$), the adaptive rescaling and LLA-CD algorithms do not converge within $1,000$ iterations for a convergence criterion $\varepsilon = 0.001$ in some replications.

For high dimensional data with $p \gg n$, we focus on the comparison between the adaptive rescaling, LLA-CD and MMCD algorithms. Figure 5 presents the average elapsed times in seconds based on 100 replications for $n = 100$ and figure 6 presents those for $n = 300$. The numbers of variables, p, is chosen to be 500, $1,000$, $2,000$, $5,000$ and $10,000$. Both plots show that as p increases, the advantage of MMCD algorithm becomes more apparent. For a fixed p, the MMCD algorithm gains more efficiency when the predictors are correlated and n is large. In addition, the MMCD algorithm has the smallest standard error of computation times, followed by the LLA-CD and adaptive rescaling algorithms. This suggests the MMCD is the most stable one among the three algorithms in high-dimensional settings.

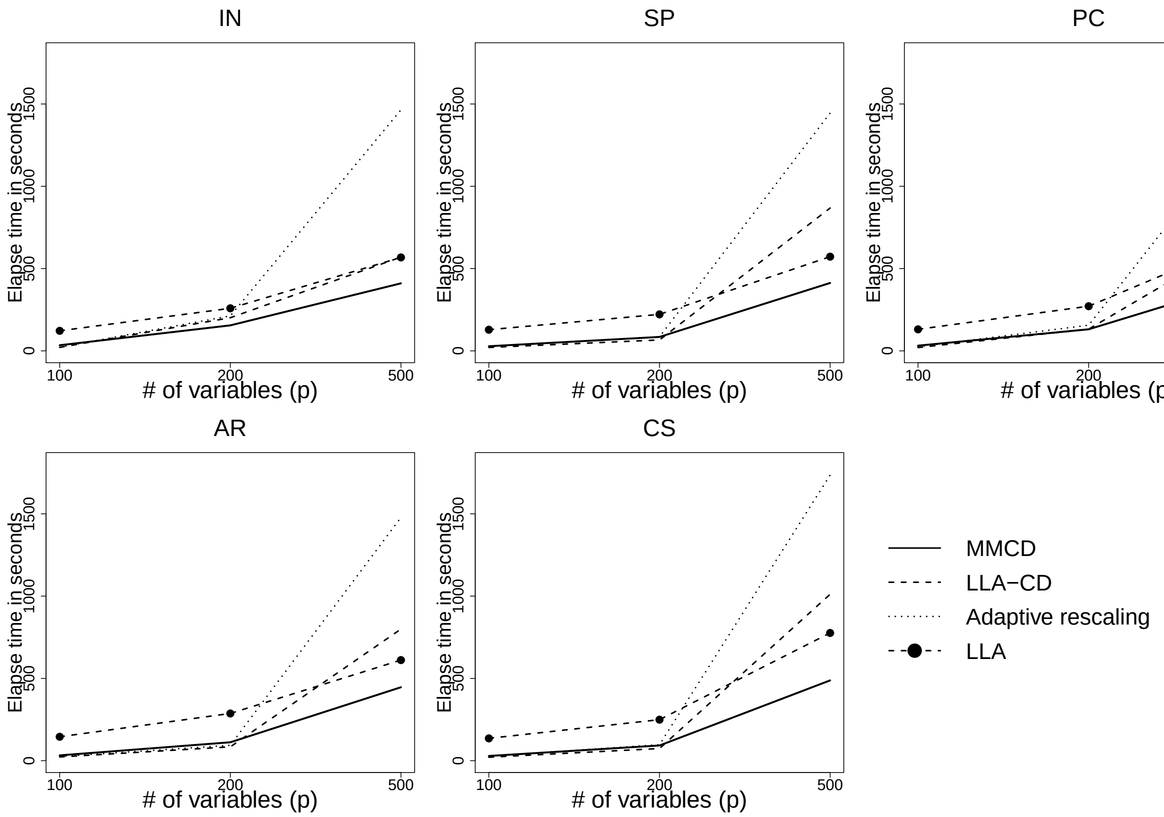

Figure 4: Computational efficiency of the LLA, adaptive rescaling, LLA-CD and MMCD algorithms with fixed sample size ($n = 1,000$). The solid line is the average elapse time of the MMCD algorithm, the dash line is that of the LLA-CD algorithm, the dotted line is that of the adaptive rescaling algorithm and the dashed line with dark circles is that of the LLA algorithm. Here, IN, SP, PC, AR and CS refers to the five design matrix described in Subsection 4.2.

4.5 Comparison of selection performance

We further compare the selection performance of the LLA, adaptive rescaling, LLA-CD and MMCD algorithms for the MCP penalized logistic models. Since we are not addressing the issue of tuning parameter selection in this article, the algorithms are compared based on the model with the best predictive performance rather than the models chosen by a particular tuning parameter selection approach. This is done as follows. We first compute the solution surface over $[0, \kappa_{\max}) \times [\lambda_{\min}, \lambda_{\max}]$ by each algorithm based on training datasets. Given the solution surface

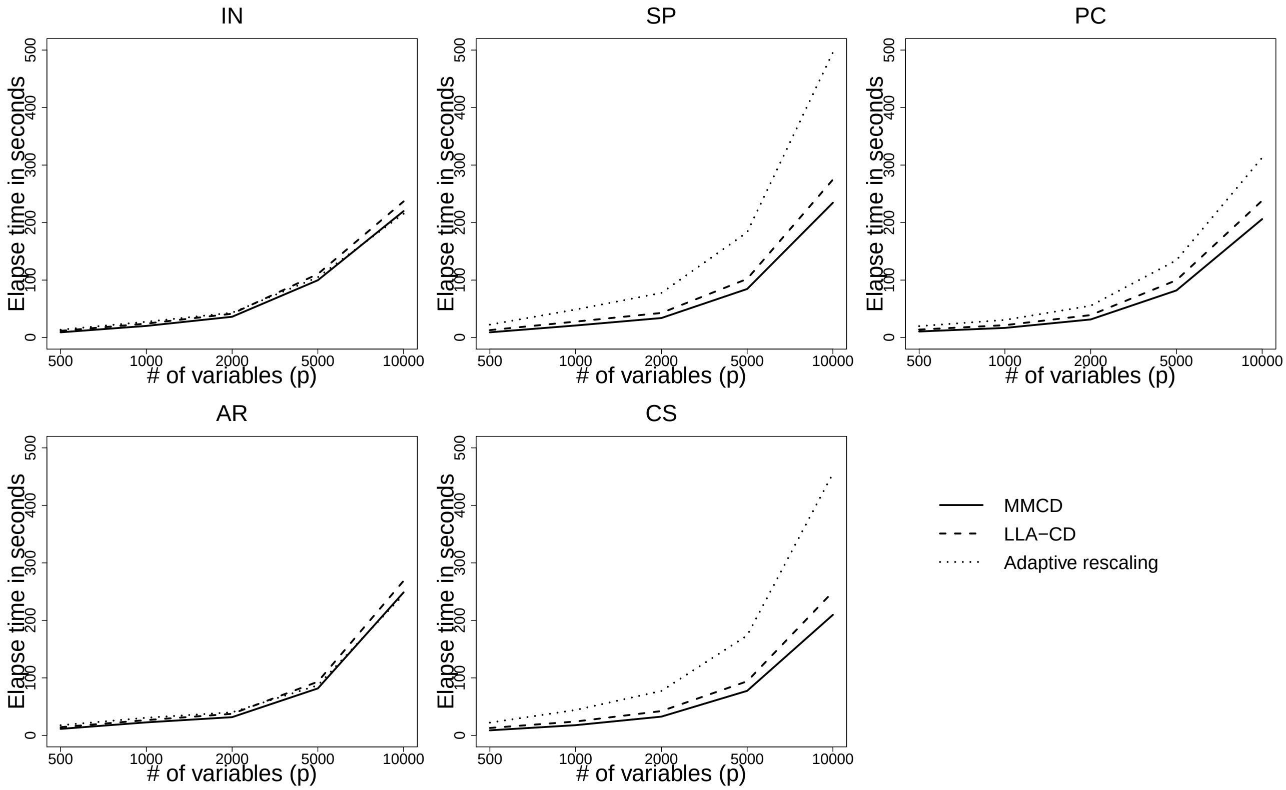

Figure 5: Computational efficiency of the adaptive rescaling, LLA-CD and MMCD algorithms for $p \gg n$ models. The sample size is fixed at $n = 100$. The solid line is the average elapse time of the MMCD algorithm, the dash line is that of the LLA-CD algorithm and the dotted line is that of the adaptive rescaling algorithm. Here, IN, SP, PC, AR and CS refers to the five design matrix described in Subsection 4.2.

$\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$, we compute the predictive area under ROC curve (PAUC) $AUC_{(\kappa_k,\lambda_v)}$ for each $\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$ based on a validation set with $n^* = 3,000$. The well-known connection between AUC and the Mann-Whitney U statistics (Bamber (1975)) is used, that is, $AUC = max\,\{1 - U_1/n_1n_2, U_1/n_1n_2\}$, with $U_1 = R_1 - (n_1(n_1+1)/2)$, where n_1 is the number of observations with outcome $y_i^* = 1$ in the validation set, R_1 is the sum of ranks for the observations with $y_i^* = 1$ in the validation set. The rank is based on the predictive probability of validation samples with $\hat{\boldsymbol{\pi}}_(\kappa_k, \lambda_v)$ computed from $\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$. The model corresponding to the maximum predictive $AUC_{(\kappa_k,\lambda_v)}$ is selected as the final model for comparison.

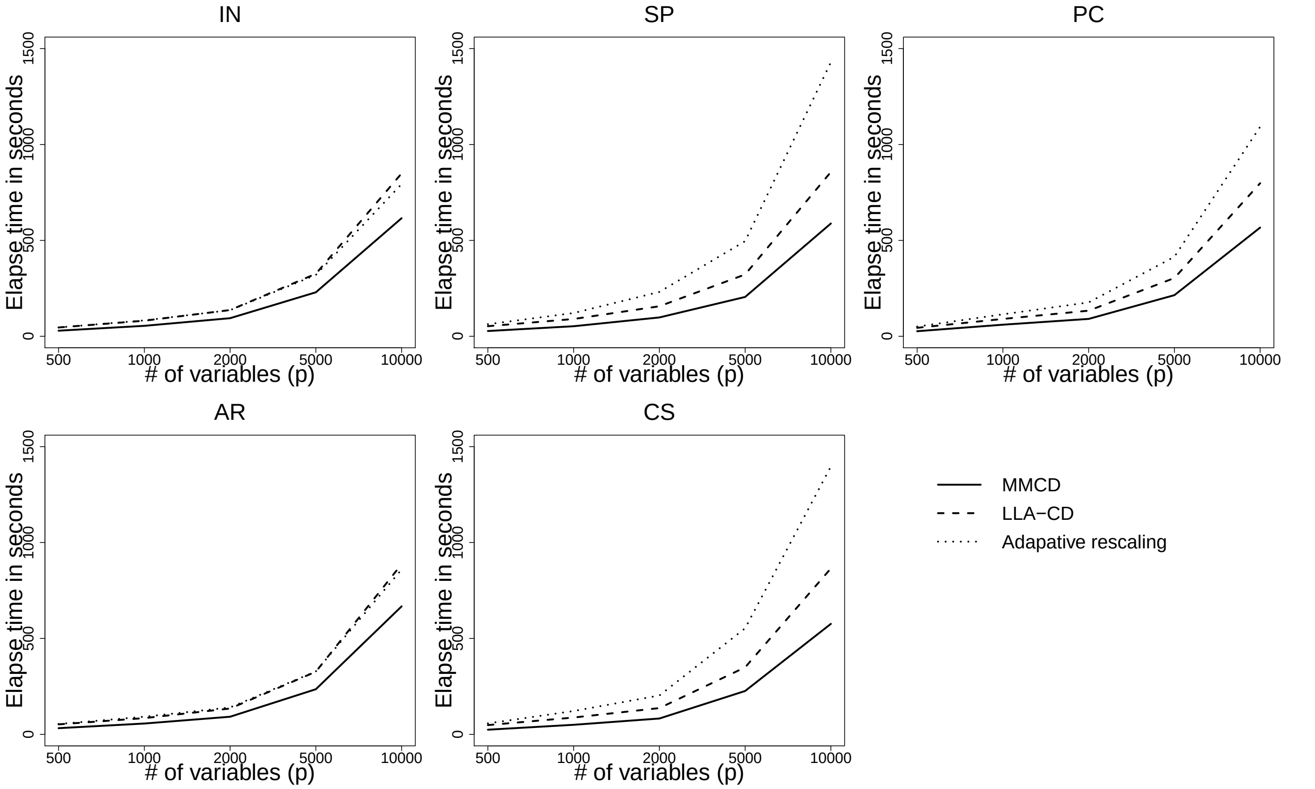

Figure 6: Computational efficiency of the adaptive rescaling, LLA-CD and MMCD algorithms for $p \gg n$ models. The sample size is fixed at $n = 300$. The solid line is the average elapse time of the MMCD algorithm, the dash line is that of the LLA-CD algorithm and the dotted line is that of the adaptive rescaling algorithm. Here, IN, SP, PC, AR and CS refers to the five design matrix described in Subsection 4.2.

The results are compared in terms of model size (MS) defined as the total number of selected variables; false discover rate (FDR), defined as the proportion of false positive variables among the total selected variables; the maximum predictive area under ROC curve (PAUC) of the validation dataset. The results reported below are based on $1,000$ replicates.

Table 1 presents the comparison among four algorithms in $n > p$ settings with $n = 1000$ and $p = 100$. The results show that the models selected by four approaches have similar PAUC. In terms of models size and FDR, the MMCD and LLA-CD algorithms performs similarly, both having smaller model size and lowest FDR than the adaptive rescaling and LLA approaches. Table

2 presents the comparison among the adaptive rescaling, LLA-CD and MMCD algorithms in high dimensional settings with $n = 100$ and $p = 2,000$. Similar to the low dimensional case, the PAUC of three methods are almost identical in $p \gg n$ models. In terms of model size and FDR, the MMCD and LLA-CD algorithms have very similar results.

4.6 Application to a Cancer Gene Expression Dataset

The purpose of this study is to discover the biomarkers associated with the prognosis of breast cancer (van't Veer *et al* (2002); Van de Vijver *et al* (2002)). Approximately $25,000$ genes were scanned using microarrays for $n = 295$ patients. Metastasis within five years is modeled as the outcome. A subset of 1,000 genes with highest Spearman correlations to the outcomes are used in the penalized models to stabilize the computation. For the same reason as in the simulation study, we do not resort to any tuning parameter selection procedure to choose models for comparison. Instead, we randomly partition the whole dataset $n = 295$ into a training (approximately 1/3 of the observations) and a validation dataset (approximately 2/3 of the observations). The model fitting is solely based on the training dataset; the solution corresponding to the maximum predictive AUC of the validation dataset is chosen as the final model for comparison. We repeat this random partition process for 900 times.

Table 3 shows the results for the SCAD penalty using the MMCD algorithm, and the MCP penalty using the adaptive rescaling, LLA-CD and MMCD algorithms. Three PAUCs are close to each other. The model size of LLA-CD algorithm for MCP penalty happens to be the largest.

4.7 Analysis results of the cancer study using tuning parameter selection method

We now present the results for the breast cancer study using the cross-validated area under ROC curve (CV-AUC) as a tuning parameter selection method. This method uses a combination of cross validation and ROC methodology. The logistic regression model is fitted based on a training sample and the (predictive) AUC of the fitted model is calculated for the test sample. Both the training and test samples are created by the cross validation. Repeat the process for multiple times to compute the average predictive AUC, which is defined as the CV-AUC. A models with the highest CV-AUC is chosen as the final model. For details of using the CV-AUC for tuning parameter selection in penalized logistic regression, we refer to Jiang, Huang, and Zhang (2011). We use 5-fold cross validation to compute the CV-AUC.

For this dataset, the SCAD penalty with the MMCD algorithm, the MCP penalty with the adaptive rescaling and LLA-CD algorithms select the same model with 67 variables and CV-AUC 0.7808. The MCP penalty with the MMCD algorithm selects 16 variables with CV-AUC 0.8024.

5 Further example of the MMCD algorithm

When the outcome variable has $K > 2$ levels, the logistic model can be extended to a baseline-category logit model. Let y_{ik} be the indicator of the outcome of the ith observation in the kth level, $k = 1, ..., K$ and $\boldsymbol{x}_i$ be the corresponding covariates. The baseline-category logit model assumes that

$$\log(\frac{\pi_k(\boldsymbol{x})}{\pi_K(\boldsymbol{x})}) = \boldsymbol{x}^T \boldsymbol{\beta}_k, \tag{19}$$

with $\pi_k(\boldsymbol{x})$ being the probability of the outcome in the kth level, and $\boldsymbol{\beta}_k$ being the corresponding coefficients. As in the case of Logistic regression, we assume $\boldsymbol{\beta}_k \in \mathbb{R}^{p+1}$ and β_{k0} being the intercept and not penalized.

Denote $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, ..., \boldsymbol{\beta}_{K-1}^T)$ as the vector of regression coefficients. Given the structure of (19), we have $\pi_k(\boldsymbol{x}) = \exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)/\{1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)\}$. Hence the loss function for the multinomial case is

$$\ell(\boldsymbol{\beta}) \;=\; \frac{1}{n}\{\sum_{i=1}^{n}\log\{1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_k)\} - \sum_{i=1}^{n}\sum_{k=1}^{K-1} y_{ik}\boldsymbol{x}_i^T\boldsymbol{\beta}_k\}. \tag{20}$$

Correspondingly, the penalized regression model for the multinomial outcome is

$$Q(\boldsymbol{\beta}) = \frac{1}{n}\{\sum_{i=1}^{n}\log\{1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_k)\} - \sum_{i=1}^{n}\sum_{k=1}^{K-1} y_{ik}\boldsymbol{x}_i^T\boldsymbol{\beta}_k\} + \sum_{k=1}^{K-1}\sum_{j=1}^{p}\rho(|\beta_{kj}|;\lambda,\gamma). \tag{21}$$

Take second derivative of $\ell(\boldsymbol{\beta})$ w.r.t. $\boldsymbol{\beta}_k$, we have

$$\nabla_k^2\ell(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_k)}{[1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_k)]^2}\boldsymbol{x}_i^T\boldsymbol{x}_i \tag{22}$$

Therefore, for the jth component in $\boldsymbol{\beta}_k$, the upper bound can be easily identified as

$$\nabla_{kj}^2\ell(\boldsymbol{\beta}) \leq \sum_{i=1}^{n} 1/4\boldsymbol{x}_{ij}^2 = 1/4.$$

Thus, we could still use $M = 1/4$ to meet the condition (ii) of the MMCD algorithm for the model. However, because of the multinomial outcome, we need two levels of cycling in the implementation of MMCD algorithm, first cycling through all the jth coordinates within $\boldsymbol{\beta}_k$, then cycling through the $k = 1, ...K-1$ to update $\boldsymbol{\beta}$.

We below outline the MMCD approach for the concave penalized baseline-category logit model.

the MMCD Algorithm for the penalized baseline-category logit model

1. *Given any initial value of $\hat{\boldsymbol{\beta}}^0$, computing the corresponding linear component $\hat{\boldsymbol{\eta}}^1$.*

Outer cycling:

2. *At step $m = 0, 1, ...$, update $\hat{\boldsymbol{\beta}}_k^m$ to $\hat{\boldsymbol{\beta}}_k^{m+1}$ by the inner cycling.*

Inner cycling:

a. *Given the current estimate of $\hat{\boldsymbol{\beta}}_{kj}^m = (\hat{\beta}_{k0}^{m+1}, ..., \hat{\beta}_{kj}^{m+1}, \hat{\beta}_{k(j+1)}^m, ..., \hat{\beta}_{kp}^m)$, update the estimate* *to* $\hat{\boldsymbol{\beta}}^m_{k(j+1)} = (\hat{\beta}^{m+1}_{k0}, ..., \hat{\beta}^{m+1}_{kj}, \hat{\beta}^{m+1}_{k(j+1)}, ..., \hat{\beta}^m_{kp})$ *by using the solution in (14 or 15) for the penalized variables and (16) for the intercept, with*

$$\tau_{kj} = \frac{\hat{\beta}^m_{kj}}{4} + \frac{1}{n}\sum_{i=1}^{n}\{\sum_{k=1}^{K-1} y_{ik} - \frac{\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_k)}{[1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_k)]^2}\}x_{ij},$$

with $\hat{\boldsymbol{\beta}}_k$ *being the latest estimate of* $\boldsymbol{\beta}_k$*. After each iteration, also update the corresponding linear component.*

b. Cycle through all the coordinate $j = 0, ..., p$ *such that* $\hat{\boldsymbol{\beta}}^m_k$ *is updated to* $\hat{\boldsymbol{\beta}}^{m+1}_k$*.*

3. Repeat the inner cycling and cycle through the $k = 1, ..., K-1$ *blocks of* $\boldsymbol{\beta}$*, update* $\hat{\boldsymbol{\beta}}^m$ *to* $\hat{\boldsymbol{\beta}}^{m+1}$*.*

4. Check the convergence criterion. If converges then stop the iteration, otherwise repeat step 2 and 3 until converge.

6 Concluding Remarks

In this article, we propose an MMCD algorithm for computing the concave penalized solutions in the GLMs. Our simulation studies and data example demonstrate that this algorithm is efficient in calculating the concave penalized solution in logistic regression models with $p \gg n$. Unlike the existing algorithms for computing concave penalized solutions, such as the LQA, LLA and MIST that approximates the penalty term, the MMCD seeks a closed form solution for each coordinate by using the exact penalty term. The majorization is only applied to the loss function. This approach increases the efficiency of CDA in high-dimensional settings. The convergence of the MMCD algorithm is proved under certain regularity conditions.

The comparison among the LLA, adaptive rescaling, LLA-CD and MMCD algorithms indicates that the MMCD is more efficient than the other approaches especially for large p and correlated covariates. Our results suggest that the LLA-CD algorithm is very competitive to the adaptive rescaling approach, in some cases, even better. The LLA-CD algorithm implements the adjacent initiation idea to reduce the computational cost, i.e. uses $\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$ as the initial values to compute $\hat{\boldsymbol{\beta}}_{\kappa_{k+1},\lambda_v}$. Within the CDA component, the solutions are updated in a sequential manner, i.e. using $\hat{\beta}_j^{m+1}$ to compute $\hat{\beta}_{j+1}^{m+1}$, rather than in a vector form, which uses $\hat{\boldsymbol{\beta}}^m$ to compute $\hat{\boldsymbol{\beta}}^{m+1}$. This is different from the the LLA-LARS implementation by Breheny and Huang (2011). The adjacent initiation and the sequential updating scheme may be the main reasons why the two implementation of LLA performs so differently.

The application of the MMCD algorithm to the logistic regression is facilitated by the fact that a simple and effective majorization function can be constructed for the logistic likelihood. However, in some other important models in the GLM family such as the log-linear model, it appears that no simple majorization function exists. One possible approach is to design a sequence of majorization functions according to the solutions at each iteration. This is an interesting problem that requires further investigation.

Acknowledgments The research of Huang is supported in part by NIH grants R01CA120988, R01CA142774 and NSF grant DMS 0805670.

SUPPLEMENTAL MATERIALS

R-package for MMCD Algorithm: R-package 'cvplogistic' is available at www.r-project.org (R Development Core Team (2011). It implements the adaptive rescaling, LLA-CD and MMCD algorithms for the logistic regressions with concave penalties.

References

Akaike, H. (1974) A new look at the statistical model identification. *IEEE T Automat Contr*, **19**(6): 716–723.

Bamber, D. (1975). The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. *J Math Psychol* **12**(4), 387–415.

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

Donoho, D. L., and Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**(3), 425–455.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. (2004) Least angle regression. *Ann Stat*, **32**2: 407–451.

Fan, J., and Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc* **96**(456), 1348–13608.

Friedman, J., Hastie, T., Höfling, H., and Tibshirani, R. (2007) Pathwise coordinate optimization. *Ann Appl Stat* **1**(2), 302–332.

Friedman, J., Hastie, T., and Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* **33**(1), 1–22.

Hunter, D. R., and Lange, K. (2004) A tutorial on MM algorithms. *J Am Stat Assoc* **58**(1), 30–37.

Hunter, D. R., and Li, R. (2005) Variable selection using MM algorithms. *Ann Stat* **33**(4), 1617–1642.

Jiang, D., Huang, J., and Zhang, Y. (2011) The cross-validated AUC for MCP-Logistic regression with high-dimensional data. *Stat Methods Med Res*, Accepted.

Lange, K., Hunter, D., and Yang, I. (2000) Optimization transfer using surrogate objective functions (with discussion). *J Comput Graph Stat*, **9**: 1–59.

Mallows, C. L. (1973) Some comments on Cp. *Technometrics*, **12**: 661–675.

Mazumder, R., Friedman, J., and Hastie, T. (2011) *SparseNet*: Coordinate descent with non-convex penalties. *J Am Stat Assoc* **106**(495): 1125–1138.

Ortega, J. M., and Rheinbold, W. C. (1970). *Iterative solution of nonlinear equations in several variables.* Academic Press, New York, NY.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2000). *A new approach to variable selection in least square problems. IMA Journal of Numerical Analysis*, **20**(3): 389-403.

Schifano, E. D., Strawderman, R. L., and Wells, M. T. (2010) Majorization-minimization algorithms for nonsmoothly penalized objective functions. *Electronic Journal of Statistics*, **4**: 1258–1299.

Schwarz, G. (1978) Estimation the dimension of a model. *Ann Stat*, **6**(2): 461–464.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *J R Stat Soc Ser B* **58**(1), 267–288.

Tseng, P. (2001) Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization. *J Optimiz Theory App*, **109**(3), 475–494.

van't Veer, L. J., Dai, H., van de Vijver, M. J., *et al* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**(31), 530–536.

van de Vijver, M. J., He, Y. D., van't Veer, L. J., *et al* (2002) A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, **347**(25), 1999–2009.

Warge, J. (1963). Minimizing certain convex functions. *SIAM Journal on Applied Mathematics*, 11, 588-593.

Wu, T. T., and Lange K. (2008) Coordinate descent algorithms for Lasso penalized regression. *Ann Appl Stat* **2**(1), 224–244.

Zhang, C. H. (2010) Nearly unbiased variable selection under minimax concave penalty. *Ann Stat* **38**(2), 894–942.

Zou, H., and Li, R. (2008) One-step sparse estimates in nonconcave penalized likelihood models. *Ann Stat*, **36**4: 1509–1533.

R Development Core Team R: a Language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, http://www.R-project.org

7 Appendix

In the Appendix, we prove Theorem 1. The proof follows the basic idea of Mazumder, Friedman and Hastie (2011). However, there are also some important differences. In particular, we need to take care of the intercept in Lemma 1 and Theorem 1, the quadratic approximation to the loss function and the coordinate-wise majorization in Theorem 1.

Lemma 1. *Suppose the data* $(\boldsymbol{y}, X)$ *lies on a compact set and the following conditions hold:*

1. The loss function $\ell(\boldsymbol{\beta})$ *is (total) differentiable w.r.t.* $\boldsymbol{\beta}$ *for any* $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$.

2. The penalty function $\rho(t)$ *is symmetric around 0 and is differentiable on* $t \geq 0$*;* $\rho'(|t|)$ *is non-negative, continuous and uniformly bounded, where* $\rho'(|t|)$ *is the derivative of* $\rho(|t|)$ *w.r.t.* $|t|$.

3. The sequence $\{\boldsymbol{\beta}^k\}$ *is bounded.*

4. For every convergent subsequence $\{\boldsymbol{\beta}^{n_k}\} \subset \{\boldsymbol{\beta}^n\}$*, the successive differences converge to zero:* $\boldsymbol{\beta}^{n_k} - \boldsymbol{\beta}^{n_k-1} \to 0$.

Then if $\boldsymbol{\beta}^{\infty}$ is any limit point of the sequence $\{\boldsymbol{\beta}^{k}\}$, then $\boldsymbol{\beta}^{\infty}$ is a minimum for the function $Q(\boldsymbol{\beta})$; i.e.

$$\liminf_{\alpha\downarrow 0+}\{\frac{Q(\boldsymbol{\beta}^{\infty}+\alpha\boldsymbol{\delta})-Q(\boldsymbol{\beta}^{\infty})}{\alpha}\}\geq 0, \tag{23}$$

for any $\boldsymbol{\delta}=(\delta_0,...,\delta_p)\in\mathbb{R}^{p+1}$.

Proof. For any $\boldsymbol{\beta}=(\beta_0,...,\beta_p)^T$ and $\boldsymbol{\delta}_j=(0,...,\delta_j,...,0)\in\mathbb{R}^{p+1}$, we have

$$\begin{aligned}\liminf_{\alpha\downarrow 0+}\{\frac{Q(\boldsymbol{\beta}+\alpha\boldsymbol{\delta}_j)-Q(\boldsymbol{\beta})}{\alpha}\} &= \nabla_j\ell(\boldsymbol{\beta})\delta_j+\liminf_{\alpha\downarrow 0+}\{\frac{\rho(|\beta_j+\alpha\delta_j|)-\rho(|\beta_j|)}{\alpha}\}\\ &= \nabla_j\ell(\boldsymbol{\beta})\delta_j+\partial\rho(\beta_j;\delta_j), \qquad (24)\end{aligned}$$

for $j\in\{1,...,p\}$, with

$$\partial\rho(\beta_j;\delta_j)=\begin{cases}\rho'(|\beta_j|)\mathrm{sgn}(\beta_j)\delta_j, & |\beta_j|>0;\\ \rho'(0)|\delta_j|, & |\beta_j|=0,\end{cases} \tag{25}$$

where

$$\mathrm{sgn}(x)=\begin{cases}1, & \text{if } x>0;\\ -1, & \text{if } x<0;\\ \text{any } u\in(-1,1), & \text{if } x=0.\end{cases}$$

Assume $\boldsymbol{\beta}^{n_k}\to\boldsymbol{\beta}^{\infty}=(\beta_0^{\infty},...,\beta_p^{\infty})$, and by assumption 4, as $k\to\infty$

$$\boldsymbol{\beta}_j^{n_k-1}=(\beta_0^{n_k},...,\beta_{j-1}^{n_k},\beta_j^{n_k},\beta_{j+1}^{n_k-1},...,\beta_p^{n_k-1})\to(\beta_0^{\infty},...,\beta_{j-1}^{\infty},\beta_j^{\infty},\beta_{j+1}^{\infty},...,\beta_p^{\infty}) \tag{26}$$

By (25) and (26), we have the results below for $j\in\{1,...,p\}$.

$$\partial\rho(\beta_j^{n_k};\delta_j)\to\partial\rho(\beta_j^{\infty};\delta_j),\ \text{if } \beta_j^{\infty}\neq 0;\ \ \partial\rho(\beta_j^{\infty};\delta_j)\geq\liminf_{k}\partial\rho(\beta_j^{n_k};\delta_j),\ \text{if } \beta_j^{\infty}=0. \tag{27}$$

By the coordinate-wise minimum of jth coordinate $j\in\{1,...,p\}$, we have

$$\nabla_j\ell(\boldsymbol{\beta}_j^{n_k-1})\delta_j+\partial\rho(\beta_j^{n_k};\delta_j)\geq 0, \text{ for all } k. \tag{28}$$

Thus (27, 28) implies that for all $j\in\{1,...,p\}$,

$$\nabla_j\ell(\boldsymbol{\beta}^{\infty})\delta_j+\partial\rho(\beta_j^{\infty};\delta_j)\geq\liminf_{k}\{\nabla_j\ell(\boldsymbol{\beta}_j^{n_k-1})\delta_j+\partial\rho(\beta_j^{n_k};\delta_j)\}\geq 0. \tag{29}$$

By (24,29), for $j \in \{1, ..., p\}$, we have

$$\liminf_{\alpha \downarrow 0+} \{ \frac{Q(\boldsymbol{\beta}^\infty + \alpha \boldsymbol{\delta}_j) - Q(\boldsymbol{\beta}^\infty)}{\alpha} \} \geq 0. \tag{30}$$

Following the above arguments, it is easy to see that for $j = 0$

$$\nabla_0 \ell(\boldsymbol{\beta}^\infty) \delta_0 \geq 0. \tag{31}$$

Hence for $\boldsymbol{\delta} = (\delta_0, ..., \delta_p) \in \mathbb{R}^{p+1}$, by the differentiability of $\ell(\boldsymbol{\beta})$, we have

$$\begin{aligned}
\liminf_{\alpha \downarrow 0+} \{ \frac{Q(\boldsymbol{\beta}^\infty + \alpha \boldsymbol{\delta}) - Q(\boldsymbol{\beta}^\infty)}{\alpha} \} &= \nabla_0 \ell(\boldsymbol{\beta}^\infty) \delta_0 \\
&+ \sum_{j=1}^{p} [\nabla_j \ell(\boldsymbol{\beta}^\infty) \delta_j + \liminf_{\alpha \downarrow 0+} \{ \frac{\rho(|\beta_j^\infty + \alpha \delta_j|) - \rho(|\beta_j^\infty|)}{\alpha} \}] \\
&= \nabla_0 \ell(\boldsymbol{\beta}^\infty) \delta_1 + \sum_{j=1}^{p} \liminf_{\alpha \downarrow 0+} \{ \frac{Q(\boldsymbol{\beta}^\infty + \alpha \boldsymbol{\delta}_j) - Q(\boldsymbol{\beta}^\infty)}{\alpha} \} \\
&\geq 0, \qquad (32)
\end{aligned}$$

by (30, 31). This completes the proof. □

Proof of Theorem 1

Proof. To ease notation, write $\chi^j_{\beta_0, ..., \beta_{j-1}, \beta_{j+1}, ..., \beta_p} \equiv \chi(u)$ for $Q(\boldsymbol{\beta})$ as a function of the jth coordinate with $(\beta_0, ..., \beta_{j-1}, \beta_{j+1}, ..., \beta_p)$ being fixed. We first deal with the $j \in \{1, ..., p\}$ coordinates, then the intercept (0th coordinate) in the following arguments.

For $j \in \{1, ..., p\}$th coordinate, observe that

$$\begin{aligned}
\chi(u + \delta) - \chi(u) &= \ell(\beta_0, ..., \beta_{j-1}, u + \delta, \beta_{j+1}, ..., \beta_p) - \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p) \\
&+ \rho(|u + \delta|) - \rho(|u|) \qquad (33) \\
&= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p) \delta + \frac{1}{2} \nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p) \delta^2 \\
&+ o(\delta^2) + \rho'(|u|)(|u + \delta| - |u|) + \frac{1}{2} \rho^{''}(|u^*|)(|u + \delta| - |u|)^2, \qquad (34)
\end{aligned}$$

with $|u^*|$ being some number between $|u+\delta|$ and $|u|$. Notation $\nabla_j \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p)$ and $\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p)$ denote the first and second derivative of the function ℓ w.r.t. the jth coordinate (assuming to be existed by condition (1)).

We re-write the RHS of (34) as follows:

$$\begin{aligned} RHS(\text{of } 34) &= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p)\delta + (\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p) - M)\delta^2 \\ &+ \rho'(|u|)\text{sgn}(u)\delta \\ &+ \rho'(|u|)(|u+\delta| - |u|) - \rho'(|u|)\text{sgn}(u)\delta + \frac{1}{2}\rho^{''}(|u^*|)(|u+\delta| - |u|)^2 \\ &+ (M - \frac{1}{2}\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p))\delta^2 + o(\delta^2). \end{aligned} \tag{35}$$

On the other hand, the solution of the jth coordinate ($j \in \{1, ..., p\}$) is to minimize the following function,

$$Q_j(u|\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \nabla_j \ell(\boldsymbol{\beta})(u - \beta_j) + \frac{1}{2}\nabla_j^2 \ell(\boldsymbol{\beta})(u - \beta_j)^2 + \rho(|u|), \tag{36}$$

By majorization, we bound $\nabla_j^2 \ell(\boldsymbol{\beta})$ by a constant M for standardized variables. So the actual function being minimized is

$$\tilde{Q}_j(u|\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \nabla_j \ell(\boldsymbol{\beta})(u - \beta_j) + \frac{1}{2}M(u - \beta_j)^2 + \rho(|u|). \tag{37}$$

Since u is to minimize (37), we have, for the jth ($j \in \{1, ..., p\}$) coordinate ,

$$\nabla_j \ell(\boldsymbol{\beta}) + M(u - \beta_j) + \rho'(|u|)\text{sgn}(u) = 0, \tag{38}$$

Because $\chi(u)$ is minimized at u_0, by (38), we have

$$\begin{aligned} 0 &= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u_0 + \delta, \beta_{j+1}, ..., \beta_p) + M(u_0 - u_0 - \delta) + \rho'(|u_0|)\text{sgn}(u_0) \\ &= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u_0, \beta_{j+1}, ..., \beta_p) + \nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u_0, \beta_{j+1}, ..., \beta_p)\delta + o(\delta) \\ &- M\delta + \rho'(|u_0|)\text{sgn}(u_0), \end{aligned} \tag{39}$$

if $u_0 = 0$ then the above holds true for some value of $\text{sgn}(u_0) \in (-1, 1)$.

Observe that $\rho'(|x|) \geq 0$, then

$$\rho'(|u|)(|u+\delta|-|u|) - \rho'(|u|)\text{sgn}(u)\delta = \rho'(|u|)[(|u+\delta|-|u|) - \text{sgn}(u)\delta] \geq 0 \tag{40}$$

Therefore using (39, 40) in (35) at u_0, we have, for $j \in \{1, ..., p\}$,

$$\begin{aligned}
\chi(u_0+\delta) - \chi(u_0) &\geq \frac{1}{2}\rho^{''}(|u^*|)(|u+\delta|-|u|)^2 \\
&+ \delta^2(M - \frac{1}{2}\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u_0, \beta_{j+1}, ..., \beta_p)) + o(\delta^2) \\
&\geq \frac{1}{2}M\delta^2 + \frac{1}{2}\rho^{''}(|u^*|)(|u+\delta|-|u|)^2 + o(\delta^2). \qquad (41)
\end{aligned}$$

By condition (ii) of the MMCD algorithm $\inf_t \rho''(|t|; \lambda, \gamma) > -M$ and $(|u+\delta|-|u|)^2 \leq \delta^2$. Hence there exist $\theta_2 = \frac{1}{2}(M + \inf_x \rho^{''}(|x|) + o(1)) > 0$, such that for the jth coordinate, $j \in \{1, ..., p\}$,

$$\chi(u_0+\delta) - \chi(u_0) \geq \theta_2\delta^2. \tag{42}$$

Now consider β_0, observe that

$$\begin{aligned}
\chi(u+\delta) - \chi(u) &= \ell(u+\delta, \beta_1, ..., \beta_p) - \ell(u, \beta_1, ..., \beta_p) \\
&= \nabla_1 \ell(u, \beta_1, ..., \beta_p)\delta + \frac{1}{2}\nabla_1^2 \ell(u, \beta_1, ..., \beta_p)\delta^2 + o(\delta^2) \\
&= \nabla_1 \ell(u, \beta_1, ..., \beta_p)\delta + (\nabla_1^2(\ell(u, \beta_1, ..., \beta_p) - M)\delta^2 \\
&+ (M - \frac{1}{2}\nabla_1^2 \ell(u, \beta_1, ..., \beta_p))\delta^2 + o(\delta^2), \qquad (43)
\end{aligned}$$

By similar arguments to (39), we have

$$\begin{aligned}
0 &= \nabla_1 \ell(u_0+\delta, \beta_1, ..., \beta_p) + M(u_0+\delta-u_0) \\
&= \nabla_1 \ell(u_0, \beta_1, ..., \beta_p) + \nabla_1^2 \ell(u_0, \beta_1, ..., \beta_p)\delta + o(\delta) - M\delta. \qquad (44)
\end{aligned}$$

Therefore, by (43, 44), for the first coordinate of $\boldsymbol{\beta}$

$$\begin{aligned}
\chi(u_0+\delta) - \chi(u_0) &= (M - \frac{1}{2}\nabla_1^2 \ell(u_0, \beta_1, ..., \beta_p))\delta^2 + o(\delta^2) \\
&= \frac{1}{2}M\delta^2 + \frac{1}{2}(M - \nabla_1^2 \ell(u_0, \beta_1, ..., \beta_p))\delta^2 + o(\delta^2) \\
&\geq \frac{1}{2}\delta^2(M + o(1)). \qquad (45)
\end{aligned}$$

Hence there exists a $\theta_1 = \frac{1}{2}(M + o(1)) > 0$, such that for the first coordinate of $\boldsymbol{\beta}$

$$\chi(u_0 + \delta) - \chi(u_0) \geq \theta_1 \delta^2. \tag{46}$$

Let $\theta = \min(\theta_1, \theta_2)$, using (42,46), we have for all the coordinates of $\boldsymbol{\beta}$,

$$\chi(u_0 + \delta) - \chi(u_0) \geq \theta \delta^2, \tag{47}$$

By (47) we have

$$\begin{aligned} Q(\boldsymbol{\beta}_j^{m-1}) - Q(\boldsymbol{\beta}_{j+1}^{m-1}) &\geq \theta(\beta_{j+1}^m - \beta_{j+1}^{m-1})^2 \\ &= \theta \parallel \boldsymbol{\beta}_j^{m-1} - \boldsymbol{\beta}_{j+1}^{m-1} \parallel_2^2, \end{aligned} \tag{48}$$

where $\boldsymbol{\beta}_j^{m-1} = (\beta_1^m, ..., \beta_j^m, \beta_{j+1}^{m-1}, ..., \beta_p^{m-1})$. The (48) establishes the boundedness of the sequence $\{\boldsymbol{\beta}^m\}$ for every $m > 1$ since the starting point of $\{\boldsymbol{\beta}^1\} \in \mathbb{R}^{p+1}$.

Apply (48) over all the coordinates, we have for all m

$$Q(\boldsymbol{\beta}^m) - Q(\boldsymbol{\beta}^{m+1}) \geq \theta \parallel \boldsymbol{\beta}^{m+1} - \boldsymbol{\beta}^m \parallel_2^2 . \tag{49}$$

Since the (decreasing) sequence $Q(\boldsymbol{\beta}^m)$ converges, (49) shows that the sequence $\{\boldsymbol{\beta}^k\}$ have a unique limit point. This completes the proof of the convergence of $\{\boldsymbol{\beta}^k\}$.

The assumption (3) and (4) in lemma 1 holds by (49). Hence, the limit point of $\{\boldsymbol{\beta}^k\}$ is a minimum of $Q(\boldsymbol{\beta})$ by lemma 1. This completes the proof of the theorem. □

Table 1: Comparison of selection performance among four algorithms with $n = 1,000$ and $p = 100$. PAUC refers to the maximum predictive area under ROC curve (PAUC) of the validation dataset. MS is model size. FDR is false discovery rate. SE is the standard errors based on $1,000$ replications. Here, IN, SP, PC, AR and CS refers to the five design matrix described in Subsection 4.2.

Structure (SNR)	Algorithm	PAUC(SE*10^5)	MS(SE*10^1)	FDR(SE*10^3)
IN	LLA	0.947 (4.58)	10.96 (0.55)	0.07 (3.32)
(4.32)	Adp res	0.948 (4.35)	16.28 (1.21)	0.36 (4.23)
	LLA-CD	0.948 (3.45)	10.79 (0.57)	0.06 (3.09)
	MMCD	0.948 (3.37)	10.90 (0.56)	0.07 (3.31)
SP	LLA	0.915 (7.74)	11.39 (0.77)	0.10 (3.96)
(3.05)	Adp res	0.916 (6.93)	14.14 (0.87)	0.27 (3.90)
	LLA-CD	0.917 (7.24)	11.35 (0.84)	0.10 (4.10)
	MMCD	0.917 (6.67)	11.27 (0.64)	0.10 (3.57)
PC	LLA	0.945 (5.95)	14.25 (1.50)	0.24 (5.72)
(3.89)	Adp res	0.947 (5.25)	15.55 (1.09)	0.33 (3.97)
	LLA-CD	0.947 (5.61)	11.61 (1.07)	0.11 (4.46)
	MMCD	0.947 (5.07)	11.41 (0.79)	0.10 (3.93)
AR	LLA	0.921 (8.83)	13.83 (1.28)	0.24 (5.34)
(3.20)	Adp res	0.924 (6.73)	18.76 (1.34)	0.44 (3.94)
	LLA-CD	0.924 (7.88)	11.29 (0.76)	0.10 (3.49)
	MMCD	0.925 (5.98)	12.11 (0.82)	0.15 (4.45)
CS	LLA	0.919 (8.13)	12.42 (1.07)	0.16 (4.70)
(3.06)	Adp res	0.921 (7.02)	14.15 (0.90)	0.27 (3.98)
	LLA-CD	0.922 (6.61)	10.64 (0.54)	0.05 (2.80)
	MMCD	0.922 (6.60)	10.94 (0.58)	0.07 (3.32)

Table 2: Comparison of selection performance among the adaptive rescaling, LLA-CD and MMCD algorithms with $n = 100$ and $p = 2,000$. PAUC refers to the maximum predictive area under ROC curve (PAUC) of the validation dataset. MS is model size. FDR is false discovery rate. SE is the standard errors based on $1,000$ replications. Here, IN, SP, PC, AR and CS refers to the five design matrix described in Subsection 4.2.

Structure (SNR)	Algorithm	PAUC(SE*10^3)	MS(SE*10^3)	FDR(SE*10^4)
IN	Adp res	0.828 (1.30)	12.25 (3.01)	0.60 (6.95)
(4.33)	LLA-CD	0.842 (1.28)	5.56 (2.02)	0.25 (8.42)
	MMCD	0.844 (1.27)	6.41 (2.09)	0.28 (9.06)
SP	Adp res	0.778 (1.96)	12.06 (3.77)	0.62 (6.05)
(3.05)	LLA-CD	0.795 (1.74)	5.25 (2.15)	0.26 (8.16)
	MMCD	0.797 (1.76)	5.75 (2.25)	0.28 (8.34)
PC	Adp res	0.872 (0.64)	7.12 (1.37)	0.42 (6.43)
(3.87)	LLA-CD	0.877 (0.54)	5.19 (1.29)	0.24 (7.48)
	MMCD	0.877 (0.54)	5.37 (1.27)	0.26 (7.46)
AR	Adp res	0.812 (1.21)	6.21 (1.69)	0.49 (8.53)
(3.19)	LLA-CD	0.830 (1.10)	3.02 (0.71)	0.17 (7.73)
	MMCD	0.831 (1.07)	3.21 (0.94)	0.18 (8.10)
CS	Adp res	0.770 (1.79)	11.89 (3.58)	0.64 (6.32)
(3.04)	LLA-CD	0.776 (1.80)	6.99 (3.09)	0.37 (9.27)
	MMCD	0.781 (1.80)	7.39 (2.99)	0.39 (9.49)

Table 3: Application of SCAD and MCP in a microarray dataset. The average and standard error are computed based on the 900 split processes. The predictive AUC is calculated as the maximum predictive AUC of the validation dataset created by the random splitting process. In each split process, approximately $n = 100$ samples are assigned to the training dataset and $n = 200$ samples into the validation dataset.

Solution surface	PAUC(SE*10^3)	MS(SE)
SCAD (MMCD)	0.7567 (0.99)	35.50 (0.47)
MCP(Adap res)	0.7565 (1.15)	39.06 (0.68)
MCP(LLA-CD)	0.7537 (0.99)	43.07 (0.63)
MCP(MMCD)	0.7570 (0.99)	35.66 (0.49)