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

The University of Iowa
Department of Statistics and Actuarial Science
Technical Report No. 412

October 20, 2011

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, 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, minimum concave penalty, $p \gg n$ models, smoothly clipped absolute deviation 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 the number of variables is small. By imposing a proper penalty on the number of selected variables, one can perform subset selection based on AIC (Akaike (1974)), BIC (Schwarz (1978)), or C_p (Mallows (1973)). However, when the number of variables is large, subset selection is computationally infeasible.

For high-dimensional data, penalization has become an important approach for variable selection in regression models. With a suitable penalty, this approach sets some coefficients to be exactly zero, thus accomplishes the goal of variable selection. Several important penalization methods have been proposed. Examples include the l_1 penalized regression or the Least absolute shrinkage and selection operator (Lasso) (Donoho and Johnstone (1994); 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 know in advance under certain sparsity conditions and other appropriate regularity conditions.

Considerable progress has been made on computational algorithms for penalized regressions. Efron *et al* (2004) introduced the LARS algorithm that can efficiently compute an entire solution paths of the Lasso in a linear regression model. 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) suggested using the majorization-minimization (MM) algorithm 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 concave penalized solutions of SCAD. With the LLA, algorithms for the Lasso can be repeatedly used for approximating concave penalized solutions. Schifano, Strawderman and Wells (2010) also used MM algorithm to generalize

the idea of LLA 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 be used to 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 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 the 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 regression 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)). Two facts may explain the efficiency of coordinate descent. (1) It only takes $O(np)$ operations to cycle through all the coordinates; while the algorithms involving matrix inversion requires $O(np^2)$ operations, whose computational burden increases dramatically when p is large. Further efficiency is attained by using the closed form solution for each coordinate by avoiding iterative search. (2) When computing a continuous solution surface, if the initial values are properly chosen, then convergence can be obtained within a few iterations. Because in that situation, by the continuity, the solution should not be far away from the initials.

Coordinate descent has also been used in computing the concave penalized solution paths (Breheny and Huang (2010); Mazumder, Friedman and Hastie (2011). Breheny and Huang (2010) observed that the CDA converges much faster than the LLA algorithm for various combinations of the values of (n, p) and various designs of covariate matrices they considered. Mazumder, Friedman and Hastie (2011) demonstrated that the CDA has better convergence properties than the LLA. Breheny and Huang (2010) also proposed an adaptive rescaling technique to overcome the difficulty due to the constantly changing scaling factors in the computation of MCP penalty.

However, the adaptive rescaling approach can not be applied to the SCAD penalty and it is not clear what is the effective concavity applied to the model beforehand. In this article, we propose a majorization minimization by coordinate descent (MMCD) algorithm for computing the concave penalized solutions in GLMs. The MMCD algorithm seeks a closed form solution for each coordinate and avoid the computation of scaling factors by majorizing the loss function. Under reasonable regularity conditions, we establish the convergence property of the MMCD algorithm.

This paper is organized as follows. In Section (2) we define the concave penalized solutions in GLMs. In Section (3) we describe the proposed MMCD algorithm, explain the benefits of majorization and study its convergence property. We also compare the MMCD algorithm with several existing algorithms in this section. In Section (4) we implement the MMCD algorithm in concave penalized logistic regression models. In Section (5) we extend 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 model assuming that y_i depends on $\boldsymbol{x}_i$ through a linear combination $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T \in \mathbb{R}^{p+1}$, whose density function given $\boldsymbol{x}_i$ is

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

Here $\phi > 0$ is a dispersion parameter. The form of the function $\psi(\theta)$ depends on the specified model. For example, in a logistic regression model, $\psi(\theta) = \log(1 + \exp(\theta))$.

Consider the (scaled) negative log-likelihood as the loss function $\ell(\boldsymbol{\beta})$, under the canonical link function with $\theta_i = \eta_i$, we have

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

We assume $\{(x_{i0})_{i=1}^n\}$ equals to one and β_0 is the intercept and is not penalized. We also assume that all the penalized variables are standardized, that is, $\|\boldsymbol{x}^j\|_2/n = 1$ with $\boldsymbol{x}^j = (x_{1j}, ..., x_{nj})^T, 1 \leq$

$j \leq p$. The notation $\|\mathbf{v}\|_2$ is the L_2 norm of a n dimensional vector $\mathbf{v}$. The standardization allows the penalization to be evenly applied to each variable regardless of their scales.

Define the concave penalized criterion as

$$Q(\boldsymbol{\beta}; \lambda, \gamma) = \frac{1}{n} \sum_{i=1}^{n} \{\psi(\boldsymbol{x}_i^T \boldsymbol{\beta}) - y_i \boldsymbol{x}_i^T \boldsymbol{\beta}\} + \sum_{j=1}^{p} \rho(|\beta_j|; \lambda, \gamma), \tag{3}$$

where ρ is a penalty function. We consider two concave penalties, SCAD and MCP. The SCAD (Fan and Li (2001)) is defined as

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

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

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

for $\lambda \geq 0$ and $\gamma > 1$. Here $x_+ = x1\{x \geq 0\}$ 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 penalty that is more concave. Both penalties begin by applying the same rate of penalization as Lasso, and then gradually reduce the penalization rate to zero as $|t|$ gets bigger. When $\gamma \to \infty$, both SCAD and MCP converge to the ℓ_1 penalty. When $\gamma \to 1$, MCP converges to the hard thresholding penalty, and when $\gamma \to 2$, SCAD does not due to the transitional knot at $\gamma = 2$. The SCAD and MCP penalties are illustrated in the middle and right panel of Figure 1.

Consider the thresholding operator defined as the solution to a penalized univariate linear regression,

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

Denote the univariate least squares solution by $\hat{\theta}_{LS} = \sum_{i=1}^{n} x_i y_i / \sum_{i=1}^{n} x_i^2$. Denote the soft-thresholding operator by $S(t, \lambda) = \operatorname{sgn}(t)(|t| - \lambda)_+$ for $\lambda > 0$ (Donoho and Johnstone (1994)).

Then for the SCAD and MCP, $\hat{\theta}(\lambda,\gamma)$ have a close form expression as follows,

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

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 $2\lambda<|\hat{\theta}_{LS}|\le\lambda\gamma$.

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

3 Majorization Minimization by Coordinate Descent

3.1 The MMCD Algorithm

The MMCD algorithm applies a quadratic approximation to the loss function $\ell(\boldsymbol{\beta})$ given the current estimation $\tilde{\boldsymbol{\beta}}$. For a model in the GLM family, this results in an iteratively reweighted least squares (IRLS) form of the loss function. Hence, the loss function in (3) can be approximated by

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

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 θ.

The MMCD algorithm updates the jth coordinate by treating the remaining coordinates as fixed values. Let $\hat{\boldsymbol{\beta}}_j^m=(\hat{\beta}_0^{m+1},...,\hat{\beta}_j^{m+1},\hat{\beta}_{j+1}^m,...,\hat{\beta}_p^m)^T$. For the loss function (7), the MMCD

updates $\hat{\boldsymbol{\beta}}_{j-1}^m$ to $\hat{\boldsymbol{\beta}}_j^m$ by minimizing the criterion

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

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 computed by taking derivative of $Q(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m)$ w.r.t β_j, which is

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

where $\rho'(|t|)$ is the first derivative of $\rho(|t|)$ with respect to $|t|$ and $\mathrm{sgn}(x) = 1, -1$ or $\in [-1,1]$ for $x > 0, < 0$ or $x = 0$.

For the MCP penalty, solving (9) for the jth coefficient we get

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

where $\delta_j = n^{-1}\sum_{i=1}^n w_i x_{ij}^2$ 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 regression model, $w_i = 1$ for $i = 1, ..., n$, thus the scaling factor $\delta_j \triangleq n^{-1}\sum_{i=1}^n w_i x_{ij}^2 = 1$ 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 coordinate descent may not be numerically stable and can lead to unreasonable solutions.

To overcome this difficulty, Breheny and Huang (2010) proposed an adaptive rescaling approach, which uses

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

for the jth coordinate-wise update. This is equivalent to apply a new regularization parameter $\gamma^* = \gamma/\delta_j$ to the MCP penalty at each coordinate-wise update in the iterations. Hence, the effective regularization parameters are not the same for the penalized variables and not known until the algorithm is converged. Numerically, the scaling factor δ_j requires extra computation.

This is not desirable when p is large. For SCAD, adaptive rescaling cannot be adopted because the scaled soft-thresholding operation only applies to the middle clauses of the three in the expression of the SCAD thresholding.

The MMCD algorithm seeks to majorize the scaling factor δ_j, $j = 1, ..., p$. 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 the scaling factor δ_j. Due to the standardization, we can use a single M to majorize all the p scaling factors. Note that in the 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 the 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.

For the MM algorithm, the majorization of the scaling factor δ_j is equivalent to finding a surrogate function $\ell(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m)$ with

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

when optimizing $\ell(\boldsymbol{\beta})$ with respect to the jth coordinate, where the second partial derivative $\nabla_j^2\ell(\boldsymbol{\beta})$ in the Taylor expansion is replaced by its upper bound M. Note that the majorization is applied coordinate-wisely to better fit the coordinate descent approach. The descent property of the MM approach ensures that iteratively minimizing $\ell(\beta_j|\hat{\boldsymbol{\beta}}_{j-1}^m)$ 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 the scaling factor, after some algebra, the jth ($j = 1, ..., p$) coordinate-wise solutions of the criterion function are

$$\text{SCAD: } \hat{\beta}_j^{m+1} = \begin{cases} \frac{1}{M}S(\tau_j, \lambda), & |\tau_j| \leq (1+M)\lambda, \\ \frac{S(\tau_j, \frac{\gamma\lambda}{\gamma-1})}{M - \frac{1}{\gamma-1}}, & (1+M)\lambda < |\tau_j| \leq M\gamma\lambda, \\ \frac{1}{M}\tau_j & |\tau_j| > M\gamma\lambda, \end{cases} \tag{13}$$

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

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 to the non-penalized intercept is

$$\beta_0 = \tau_0/M, \tag{15}$$

with $\tau_0 = M\hat{\beta}_0^m + n^{-1}\sum_{i=1}^n x_{i0}(y_i - \dot{\psi}(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_{j-1}^m))$.

In (13) and (14), we want to ensure the denominators in both expressions are positive, that is, $M - 1/(\gamma-1) > 0$ and $M - 1/\gamma > 0$. This naturally leads to the constraint on the the penalty, $\inf_t \rho''(|t|;\lambda,\gamma) > -M$, where $\rho''(|t|;\lambda,\gamma)$ is the second derivative of $\rho(|t|;\lambda,\gamma)$ with respect to $|t|$. 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 the SCAD and $\gamma > 1/M$ for the 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

$$\hat{\boldsymbol{\eta}}_{j+1}^m = \hat{\boldsymbol{\eta}}_j^m + \boldsymbol{x}^{j+1}(\hat{\beta}_{j+1}^{m+1} - \hat{\beta}_{j+1}^m) == \hat{\boldsymbol{\eta}}_j^m + (\hat{\boldsymbol{\beta}}_{j+1}^m - \hat{\boldsymbol{\beta}}_j^m)\boldsymbol{x}^{j+1}. \tag{16}$$

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 summarize the MMCD algorithm as follows. Assuming the conditions below hold:

- (i). The second partial derivative of $\ell(\boldsymbol{\beta})$ with respect to 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 $j = 0, ..., p$.
- (ii). $\inf_t \rho''(|t|;\lambda,\gamma) > -M$, with $\rho''(|t|;\lambda,\gamma)$ being the second derivative of $\rho(|t|;\lambda,\gamma)$ with respect to $|t|$.

The MMCD algorithm for a given (λ,γ) computes the concave penalized solution 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 $\hat{\boldsymbol{\beta}}_j^m$ to $\hat{\boldsymbol{\beta}}_{j+1}^m$ by using the solution in (13) or (14) for the penalized variables and (15) for the intercept. After each iteration, also compute the corresponding linear component $\hat{\boldsymbol{\eta}}_{j+1}^m$ using (16). Cycle through all the coordinates from $j = 0, ..., p$ such that $\hat{\boldsymbol{\beta}}^m$ is

updated to $\hat{\boldsymbol{\beta}}^{m+1}$.

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

We use the convergence criterion $\|\hat{\boldsymbol{\beta}}^{m+1} - \hat{\boldsymbol{\beta}}^{m}\|_2/(\|\hat{\boldsymbol{\beta}}^{m}\|_2 + 0.01) < \varepsilon$. We choose $\varepsilon = 0.001$ in our implementation.

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 algorithm always converges to a minimum of the objective function.

Theorem 1. *Consider the objective function in (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* $\rho(|t|)$ *with respect to* $|t|$. *Also assume that conditions (i) and (ii) stated in the MMCD algorithm hold.*

Then the sequence generated by the MMCD $\{\boldsymbol{\beta}^m\}$ *converges to a 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 the columns are not identically zero. The proof of theorem (1) is provided 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 other 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 feature in that they all use a surrogate function to majorize the concave penalty term $\rho(|t|; \lambda, \gamma)$. The optimization procedure is carried out by minimizing the objective function with the surrogate penalty instead of the original penalty.

3.3.1 LQA and perturbed LQA algorithms

The LQA uses the following approximation to the penalty,

$$\rho(|t|;\lambda,\gamma) \approx \rho(|t_0|;\lambda,\gamma) + \frac{\rho^{'}(|t_0|;\lambda,\gamma)}{2|t_0|}(t^2 - t_0^2), \text{ for } t \approx t_0. \tag{17}$$

Then Newton-Raphson type iteration was employed to minimize the penalized criterion with the surrogate penalty function. When t_0 is close to zero, the algorithm is unstable. To avoid the numerical instability, Fan and Li (2001) suggested that if $\hat{\beta}_j$ is small enough, say $|\hat{\beta}_j| < \epsilon$ (a pre-specified value), set $\hat{\beta}_j = 0$ and remove the jth variable from the iteration. A drawback of LQA algorithm is that, if a variable is removed in an iteration, it will necessarily be excluded from the final model.

Hunter and Li (2005) studied the convergence property of LQA and showed that LQA is a type of MM algorithm with the SCAD penalty majorized by a quadratic function. Furthermore, to avoid numerical instability, they proposed a perturbed version of LQA to majorize the LQA.

$$\rho(|t|;\lambda,\gamma) \approx \rho(|t_0|;\lambda,\gamma) + \frac{\rho^{'}(|t_0|;\lambda,\gamma)}{2|t_0+\tau_0|}(t^2 - t_0^2), \text{ for } t \approx t_0. \tag{18}$$

Practically, how to determine the size of τ_0 is not easy since the size of τ_0 could impact the speed of convergence and the sparsity of the solution.

3.3.2 LLA and MIST algorithms

Zou and Li (2008) proposed a local linear approximation (LLA) for computing concave penalized estimates. This approximation takes the form

$$\rho(|t|;\lambda,\gamma) \approx \rho(|t_0|;\lambda,\gamma) + \rho^{'}(|t_0|;\lambda,\gamma)(|t| - |t_0|), \text{ for } t \approx t_0. \tag{19}$$

The LLA algorithm can be implemented by repeatedly using the algorithms for computing the Lasso solutions such as the LARS. Schifano, Strawderman and Wells (2010) generalized the idea of LLA to multiple penalties and linked the LLA to the soft-thresholding operation, and proposed the MIST algorithm. However, the authors indicated that the MIST tends to converge at a slower rate in the case of $p > n$.

The LQA, perturbed LQA, LLA and MIST can be viewed as the surrogate functions that majorize the SCAD penalty. Figure 2 illustrate the three majorizations of SCAD. The left panel

of Figure 2 is majorized at $t = 3$ while the right is majorized at $t = 1$. For perturbed LQA, we choose $\tau_0 = 0.5$. In both plots, $\gamma = 4$ and $\lambda = 2$ are chosen for better illustration effect.

To apply these methods to the GLM, we need to approximate both the likelihood and the penalty. This does not take full advantage of the coordinate descent algorithm. Indeed, the approximation of the penalty requires additional iterations for convergence and is not necessary, since exact solution exists when updating a single estimate in the coordinate descent. Thus in our proposed algorithm, we use the exact form of the penalty and only majorize the loss function to avoid the computation of scaling factor. Breheny and Huang (2010) reported that adaptive rescaling technique is at least 100 times faster than the LLA algorithm. Therefore, we focus on the comparison between the MMCD and the adaptive rescaling approach in the following section.

4 The MMCD for Penalized Logistic Regression

In this section, we implement the MMCD in the penalized logistic regression, which is one of the most widely used models in biostatistical applications. In this 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 a 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 for all the second partial derivatives $\nabla_j^2 \ell(\hat{\boldsymbol{\beta}})$ 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 penalty.

4.1 Computation of Solution Surface

A common practice in applying the SCAD and MCP 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 $\gamma \approx 3.7$ in the SCAD penalty (Fan and Li (2001)) and $\gamma \approx 2.7$ (Zhang (2010)) in the MCP. However, in generalized linear models including the logistic regression, these values are not 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

reparameterize $\kappa = 1/\gamma$ to facilitate the description of the approach for computing the solution surface. To meet the condition (ii) of MMCD algorithm, we require $\kappa \in [0, \kappa_{\max}]$, where $\kappa_{\max} = 1/5$ for SCAD and $\kappa_{\max} = 1/4$ for MCP. Note that when $\kappa = 0$, both SCAD and MCP simplify to 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 logistic regression, $\lambda_{\max} = n^{-1}\max_j |(\boldsymbol{x}^j)^T(\boldsymbol{y} - \hat{\boldsymbol{\pi}})|$ with $\hat{\boldsymbol{\pi}} = \bar{y}\boldsymbol{J}$ and $\boldsymbol{J}$ being a unit vector, for every κ_k. We 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 $[0, \kappa_{\max}) \times [\lambda_{\min}, \lambda_{\max}]$. We denote the MMCD solution for a given (κ_k, λ_v) to be $\hat{\boldsymbol{\beta}}_{\kappa_k, \lambda_v}$.

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 this 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. That is using $\hat{\boldsymbol{\beta}}_{\kappa_k, \lambda_v}$ as the initial value to compute the solution $\hat{\boldsymbol{\beta}}_{\kappa_{k+1}, \lambda_v}$.

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

Figure (3) presents the solution paths of a causal variable (plot a) and a null variable (plot b) along κ using the MCP penalty. Observe that although Lasso tends to over select in some cases, it could fail to select certain variables, which are selected by MCP (dash line in plot a). This could be a serious problem for Lasso if the missing predictor is a causal variable. Furthermore, we observe that the estimates could change substantially when κ cross certain threshold values. This justifies our treatment of κ as a tuning parameter since a pre-specified κ might not give the optimal results. This is the reason that we use a data-driven procedure to choose both κ and λ.

4.2 Design of simulation study

Denote the design matrix of the penalized variables as Z, which is the sub-matrix of X with the first column removed. Let $A_0 \equiv \{j \leq p : \beta_j \neq 0\}$ be the set of causal variables and let p_0 be the dimension of A_0. We set $p = 1,000$ with $p_0 = 10$. We fix $\beta_0 = 0.01$ and the coefficients for 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 SNR $= \sqrt{\boldsymbol{\beta}^T X^T X \boldsymbol{\beta}/n}$, is approximately in the range of $(3, 4)$. The covariates are generated from multivariate normal distributions. The outcomes $\boldsymbol{y}$ are generated from Bernoulli distributions with $y_i \sim$ Bernoulli$(1, p_i)$ for $i = 1, ..., n$. We set $K = 20$ and $V = 100$ in the simulation.

We consider five types of correlation structures of Z. They are (1) Independent structure (IN) among all the p penalized variables, i.e. $Var(Z) = I_p$, with I_p being the identity matrix of dimension $p \times p$. (2) Separate structure (SP), i.e. the causal variables and the null variables are independent, $Var(Z) =$ block diagonal$(\boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1)$, with $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ being the covariance matrix for the causal variables A_0 and the null variables A_1, respectively. Within each set of variables, we assume a compound symmetry structure for the variables. That is $\rho(x_{ij}, x_{ik}) = \rho$ for $j \neq k$, $j, k \in A_s, s = 0, 1$. (3) Partial Correlated structure (PC), i.e. part of the causal variables are correlated with the part of the null variables. Specifically, $Var(Z) =$ 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 $\boldsymbol{\Sigma}_a$, $\boldsymbol{\Sigma}_b$, $\boldsymbol{\Sigma}_c$. (4) First-order Autoregressive (AR1) structure, that is $\rho(x_{ij}, x_{ik}) = \rho^{(|j-k|)}$, for $j \neq k, j, k = 1, ..., p$; $i = 1, ..., n$. (5) Compound Symmetry (CS) structure for all the variables. In our simulation, $\rho = 0.5$ in all types of structures to present a median level of correlation.

4.3 Comparison of computational efficiency

The adaptive rescaling approach and the MMCD algorithm are applied to the same training datasets. The computation is done on Inter Xeon CPU (E5440@2.83GHZ) machines with Ubuntu system (Linux version 2.6). Table 1 reports the average time of computing the whole solution surface for the MCP penalty measured in seconds based on 100 replicates for datasets with $n = 100$

and $p = 1,000$. In all the models explored, the MMCD is twice faster than the adaptive rescaling approach. We expect that the MMCD will gain more efficiency when p gets larger.

4.4 Comparison of Lasso, SCAD and MCP by Simulation

In this section, we compare the empirical performance of Lasso, SCAD and MCP penalties in penalized logistic regression models using the MMCD algorithm. Since we are not addressing the issue of tuning parameter selection in this article, three penalties are compared based on the model with the best predictive performance rather than the model chosen by any tuning parameter selection approach. To fulfill this purpose, a large validation dataset with 2,000 observations is generated from the same model simulating the training dataset. The solution surface over the rectangle $[0, \kappa_{\max}) \times [\lambda_{\min}, \lambda_{\max}]$ is computed by the MMCD algorithm entirely based on the training dataset. Given the solution surface $\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$, we compute the predictive Area Under ROC Curve (P-AUC) for the validation set, $AUC_{(\kappa_k,\lambda_v)}$ for each $\hat{\boldsymbol{\beta}}_{\kappa_k,\lambda_v}$. The well-known connection between between AUC and the Mann-Whitney U statistics, Bamber (1975) is used for computing the AUC:

$$AUC = max\left\{1 - \frac{U_1}{n_1 n_2}, \frac{U_1}{n_1 n_2}\right\},$$

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.

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 (null variables) among the total selected variables; the maximum predictive area under ROC curve (AUC) of the validation dataset. Two sample sizes $n = 100, 300$ are explored and the results are similar. For the sake of space, we only report the results of $n = 100$. The results reported below are based on 1,000 replicates.

Table 2 presents the average and standard error of model size, FDR and the predictive AUC of validation dataset. It seems that as a selection tool, Lasso is inferior to the concave penalty

such as SCAD and MCP in the sense that it favors a larger model size, with a lower predictive AUC and a higher FDR. This results is consistent with the theoretical results from Zhang and Huang (2008). Our results also suggest that SCAD has a similar predictive performance but with a slightly larger model size and a higher false discovery rate, compared to MCP.

4.5 Application to a Cancer Gene Expression Dataset

We further apply the penalized logistic regression model to a cancer study. 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.

We first compare the predictive performance of the LASSO, SCAD and MCP. For the same reason as in the simulation study, we do not resort to any tuning parameter selection procedure to choose the model for comparison. Instead, we randomly partition the whole dataset $n = 295$ into a training (approximately 1/3 of the observations) and validation data (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 further comparison. This split process is repeated for 900 times.

The results presented in table 3 are consistent with those from simulation. Lasso tends to select a larger model. The SCAD and MCP perform very similarly. Their predictive AUCs in the validation datasets are close to each other. Also, the sizes of the models selected by the SCAD and MCP are almost the same.

4.6 Analysis results of the cancer study

We now present the results for the breast cancer study. We use the cross-validated area under the ROC (CV-AUC) method for tuning parameter selection. 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. Models with the highest CV-AUC are 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, Lasso penalty selects 101 variables with CV-AUC=0.7797, SCAD penalty selects 26 variables with CV-AUC=0.7859 and MCP selects 24 variables with CV-AUC=0.7886. All the 24 variables selected by MCP are also selected by SCAD. Among the 26 variables selected by SCAD, only 2 are not selected by Lasso. The results are consistent with those of simulation. In particular, the MCP selects a model with the highest CV-AUC with the smallest model.

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

$$\log(\frac{\pi_k(\boldsymbol{x})}{\pi_K(\boldsymbol{x})}) = \boldsymbol{x}^T\boldsymbol{\beta}_k, \tag{20}$$

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 (20), we have $\pi_k(\boldsymbol{x}) = \frac{\exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)}{1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}^T\boldsymbol{\beta}_k)}$. Hence the loss function for the multinomial case is

$$\ell(\boldsymbol{\beta}) \;=\; \frac{1}{n}\{\sum_{i=1}^{n}\log\{1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_k)\} - \sum_{i=1}^{n}\sum_{k=1}^{K-1} y_{ik}\boldsymbol{x}_i^T\boldsymbol{\beta}_k\}. \tag{21}$$

Correspondingly, the penalized regression model for the multinomial outcome is

$$Q(\boldsymbol{\beta}) = \frac{1}{n}\{\sum_{i=1}^{n}\log\{1+\sum_{k=1}^{K-1}\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_k)\} - \sum_{i=1}^{n}\sum_{k=1}^{K-1} y_{ik}\boldsymbol{x}_i^T\boldsymbol{\beta}_k\} + \sum_{k=1}^{K-1}\sum_{j=1}^{p}\rho(|\beta_{kj}|;\lambda,\gamma). \tag{22}$$

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

$$\nabla_k^2 \ell(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{k=1}^{K-1} \exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_k)}{[1 + \sum_{k=1}^{K-1} \exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_k)]^2} \boldsymbol{x}_i^T \boldsymbol{x}_i \tag{23}$$

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* $\hat{\boldsymbol{\beta}}_{k(j+1)}^m = (\hat{\beta}_{k0}^{m+1}, ..., \hat{\beta}_{kj}^{m+1}, \hat{\beta}_{k(j+1)}^{m+1}, ..., \hat{\beta}_{kp}^m)$ *by using the solution in (13 or 14) for the penalized variables and (15) for the intercept, with* $\tau_{kj} = \hat{\beta}_{kj}^m/4 + \frac{1}{n} \sum_{i=1}^{n} \{\sum_{k=1}^{K-1} y_{ik} - \frac{\sum_{k=1}^{K-1} \exp(\boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k)}{[1+\sum_{k=1}^{K-1} \exp(\boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}_k)]^2}\} x_{ij}$*, 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* $\hat{\boldsymbol{\beta}}_k^m$ *is updated to* $\hat{\boldsymbol{\beta}}_k^{m+1}$*.*

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 efficient and stable MMCD algorithm for computing the concave penalized solutions in the GLMs. Unlike the existing MM algorithms for computing concave penalized solutions, such as the LQA, LLA and MIST that majorize the penalty term, the MMCD seeks a closed form solution for each coordinate by using the exact penalty term. The majorization is, however, applied to the loss function to avoid the computation of the scaling factor. This approach increases the efficiency of coordinate descent in high-dimensional settings. The convergence of the MMCD algorithm is proved under certain regularity conditions.

A penalized logistic regression model is used to illustrate the MMCD algorithm. The comparison with adaptive rescaling approach indicates that the MMCD is more efficient in high-dimensional settings. The results from the simulation and data analysis reveal the adequacy of the MMCD algorithm in high-dimensional settings. Based on the MMCD solution of penalized logistic regression, we compare Lasso penalty and the concave penalties including SCAD and MCP for their empirical performance. The MCP has the best performance in terms of predictive AUC and FDR in the simulated models we considered.

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.

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

SUPPLEMENTAL MATERIALS

R-package for MMCD Algorithm: R-package containing code to compute the concave penalized logistic regression, 'cvplogistic', is available at `www.r-project.org` (R Development Core Team (2011).

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. (2010) Coordinate descent algorithms for nonconvex penalized regression, with application to biological feature selection. *Ann Appl Stat*, **5**(1), 232–253.

Donoho, DL., and Johnstone, JM. (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, DR., and Lange, K. (2004) A tutorial on MM algorithms. *J Am Stat Assoc* **58**(1), 30–37.

Hunter, DR., 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.

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, LJ., Dai, H., van de Vijver, MJ., *et al* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**(31), 530–536.

van de Vijver, MJ., He, YD., van't Veer, LJ., *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, TT., and Lange K. (2008) Coordinate descent algorithms for Lasso penalized regression. *Ann Appl Stat* **2**(1), 224–244.

Zhang, CH., and Huang, J. (2008) The sparsity and bias of the Lasso selection in high-dimensional linear regression. *Ann Stat* **36**(4), 1567–1594.

Zhang, CH. (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

To prove the following theorem, we first present a lemma.

Lemma 1. *Suppose the data* $(\boldsymbol{y}, \boldsymbol{X})$ *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.*

$$\liminf_{\alpha \downarrow 0+} \{\frac{Q(\boldsymbol{\beta}^\infty + \alpha\boldsymbol{\delta}) - Q(\boldsymbol{\beta}^\infty)}{\alpha}\} \geq 0, \tag{24}$$

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

$$\begin{aligned} \liminf_{\alpha \downarrow 0+} \{\frac{Q(\boldsymbol{\beta} + \alpha\boldsymbol{\delta}_j) - Q(\boldsymbol{\beta})}{\alpha}\} &= \nabla_j \ell(\boldsymbol{\beta})\delta_j + \liminf_{\alpha \downarrow 0+} \{\frac{\rho(|\beta_j + \alpha\delta_j|) - \rho(|\beta_j|)}{\alpha}\} \\ &= \nabla_j \ell(\boldsymbol{\beta})\delta_j + \partial\rho(\beta_j; \delta_j), \end{aligned} \tag{25}$$

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

$$\partial\rho(\beta_j; \delta_j) = \begin{cases} \rho^{'}(|\beta_j|)\text{sgn}(\beta_j)\delta_j, & |\beta_j| > 0; \\ \rho^{'}(0)|\delta_j|, & |\beta_j| = 0, \end{cases} \tag{26}$$

where

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

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

$$\boldsymbol{\beta}_j^{n_k-1} = (\beta_0^{n_k}, ..., \beta_{j-1}^{n_k}, \beta_j^{n_k}, \beta_{j+1}^{n_k-1}, ..., \beta_p^{n_k-1}) \to (\beta_0^\infty, ..., \beta_{j-1}^\infty, \beta_j^\infty, \beta_{j+1}^\infty, ..., \beta_p^\infty) \tag{27}$$

By (26) and (27), we have the results below for $j \in \{1, ..., p\}$.

$$\partial\rho(\beta_j^{n_k}; \delta_j) \to \partial\rho(\beta_j^\infty; \delta_j), \text{ if } \beta_j^\infty \neq 0; \;\; \partial\rho(\beta_j^\infty; \delta_j) \geq \liminf_k \partial\rho(\beta_j^{n_k}; \delta_j), \text{ if } \beta_j^\infty = 0. \tag{28}$$

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

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

Thus (28, 29) implies that for all $j \in \{1, ..., p\}$,

$$\nabla_j \ell(\boldsymbol{\beta}^{\infty})\delta_j + \partial\rho(\beta_j^{\infty}; \delta_j) \geq \liminf_k \{\nabla_j \ell(\boldsymbol{\beta}_j^{n_k - 1})\delta_j + \partial\rho(\beta_j^{n_k}; \delta_j)\} \geq 0. \tag{30}$$

By (25,30), for $j \in \{1, ..., p\}$, we have

$$\liminf_{\alpha \downarrow 0+} \{\frac{Q(\boldsymbol{\beta}^{\infty} + \alpha\boldsymbol{\delta}_j) - Q(\boldsymbol{\beta}^{\infty})}{\alpha}\} \geq 0. \tag{31}$$

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

$$\nabla_0 \ell(\boldsymbol{\beta}^{\infty})\delta_0 \geq 0. \tag{32}$$

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

$$\begin{aligned}
\liminf_{\alpha \downarrow 0+} \{\frac{Q(\boldsymbol{\beta}^{\infty} + \alpha\boldsymbol{\delta}) - Q(\boldsymbol{\beta}^{\infty})}{\alpha}\} &= \nabla_0 \ell(\boldsymbol{\beta}^{\infty})\delta_0 \\
&+ \sum_{j=1}^{p} [\nabla_j \ell(\boldsymbol{\beta}^{\infty})\delta_j + \liminf_{\alpha \downarrow 0+} \{\frac{\rho(|\beta_j^{\infty} + \alpha\delta_j|) - \rho(|\beta_j^{\infty}|)}{\alpha}\}] \\
&= \nabla_0 \ell(\boldsymbol{\beta}^{\infty})\delta_1 + \sum_{j=1}^{p} \liminf_{\alpha \downarrow 0+} \{\frac{Q(\boldsymbol{\beta}^{\infty} + \alpha\boldsymbol{\delta}_j) - Q(\boldsymbol{\beta}^{\infty})}{\alpha}\} \\
&\geq 0,
\end{aligned} \tag{33}$$

by (31, 32).

This completes the proof. □

Proof of Theorem 1

Proof. For the sake of notational convenience, we 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

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

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 (35) as follows:

$$\begin{aligned}
RHS(\text{of } 35) &= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p)\delta + (\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p) - M)\delta^2 \\
&+ \rho'(|u|)\text{sgn}(u)\delta \\
&+ \rho'(|u|)(|u+\delta| - |u|) - \rho'(|u|)\text{sgn}(u)\delta + \frac{1}{2}\rho''(|u^*|)(|u+\delta| - |u|)^2 \\
&+ (M - \frac{1}{2}\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u, \beta_{j+1}, ..., \beta_p))\delta^2 + o(\delta^2). \qquad (36)
\end{aligned}$$

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

$$Q_j(u|\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \nabla_j \ell(\boldsymbol{\beta})(u - \beta_j) + \frac{1}{2}\nabla_j^2 \ell(\boldsymbol{\beta})(u - \beta_j)^2 + \rho(|u|), \qquad (37)$$

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

$$\tilde{Q}_j(u|\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \nabla_j \ell(\boldsymbol{\beta})(u - \beta_j) + \frac{1}{2}M(u - \beta_j)^2 + \rho(|u|). \qquad (38)$$

Since u is to minimize (38), we have, for the jth ($j \in \{1, ..., p\}$) coordinate ,

$$\nabla_j \ell(\boldsymbol{\beta}) + M(u - \beta_j) + \rho'(|u|)\text{sgn}(u) = 0, \qquad (39)$$

Because $\chi(u)$ is minimized at u_0, by (39), we have

$$\begin{aligned} 0 &= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u_0 + \delta, \beta_{j+1}, ..., \beta_p) + M(u_0 - u_0 - \delta) + \rho'(|u_0|)\text{sgn}(u_0) \\ &= \nabla_j \ell(\beta_0, ..., \beta_{j-1}, u_0, \beta_{j+1}, ..., \beta_p) + \nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u_0, \beta_{j+1}, ..., \beta_p)\delta + o(\delta) \\ &- M\delta + \rho'(|u_0|)\text{sgn}(u_0), \end{aligned} \tag{40}$$

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

$$\rho'(|u|)(|u + \delta| - |u|) - \rho'(|u|)\text{sgn}(u)\delta = \rho'(|u|)[(|u + \delta| - |u|) - \text{sgn}(u)\delta] \geq 0 \tag{41}$$

Therefore using (40, 41) in (36) at u_0, we have, for $j \in \{1, ..., p\}$,

$$\begin{aligned} \chi(u_0 + \delta) - \chi(u_0) &\geq \frac{1}{2}\rho''(|u^*|)(|u + \delta| - |u|)^2 \\ &+ \delta^2(M - \frac{1}{2}\nabla_j^2 \ell(\beta_0, ..., \beta_{j-1}, u_0, \beta_{j+1}, ..., \beta_p)) + o(\delta^2) \\ &\geq \frac{1}{2}M\delta^2 + \frac{1}{2}\rho''(|u^*|)(|u + \delta| - |u|)^2 + o(\delta^2). \end{aligned} \tag{42}$$

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

$$\chi(u_0 + \delta) - \chi(u_0) \geq \theta_2 \delta^2. \tag{43}$$

Now consider β_0, observe that

$$\begin{aligned} \chi(u + \delta) - \chi(u) &= \ell(u + \delta, \beta_1, ..., \beta_p) - \ell(u, \beta_1, ..., \beta_p) \\ &= \nabla_1 \ell(u, \beta_1, ..., \beta_p)\delta + \frac{1}{2}\nabla_1^2 \ell(u, \beta_1, ..., \beta_p)\delta^2 + o(\delta^2) \\ &= \nabla_1 \ell(u, \beta_1, ..., \beta_p)\delta + (\nabla_1^2(\ell(u, \beta_1, ..., \beta_p) - M)\delta^2 \\ &+ (M - \frac{1}{2}\nabla_1^2 \ell(u, \beta_1, ..., \beta_p))\delta^2 + o(\delta^2), \end{aligned} \tag{44}$$

By similar arguments to (40), we have

$$\begin{aligned} 0 &= \nabla_1 \ell(u_0 + \delta, \beta_1, ..., \beta_p) + M(u_0 + \delta - u_0) \\ &= \nabla_1 \ell(u_0, \beta_1, ..., \beta_p) + \nabla_1^2 \ell(u_0, \beta_1, ..., \beta_p)\delta + o(\delta) - M\delta. \end{aligned} \tag{45}$$

Therefore, by (44, 45), for the first coordinate of $\boldsymbol{\beta}$

$$\begin{aligned}\chi(u_0+\delta)-\chi(u_0) &= (M-\frac{1}{2}\nabla_1^2\ell(u_0,\beta_1,...,\beta_p))\delta^2+o(\delta^2)\\ &= \frac{1}{2}M\delta^2+\frac{1}{2}(M-\nabla_1^2\ell(u_0,\beta_1,...,\beta_p))\delta^2+o(\delta^2)\\ &\geq \frac{1}{2}\delta^2(M+o(1)). \end{aligned} \tag{46}$$

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

$$\chi(u_0+\delta)-\chi(u_0)\geq\theta_1\delta^2. \tag{47}$$

Let $\theta=\min(\theta_1,\theta_2)$, using (43,47), we have for all the coordinates of $\boldsymbol{\beta}$,

$$\chi(u_0+\delta)-\chi(u_0)\geq\theta\delta^2, \tag{48}$$

By (48) we have

$$\begin{aligned}Q(\boldsymbol{\beta}_j^{m-1})-Q(\boldsymbol{\beta}_{j+1}^{m-1}) &\geq \theta(\beta_{j+1}^m-\beta_{j+1}^{m-1})^2\\ &= \theta\parallel\boldsymbol{\beta}_j^{m-1}-\boldsymbol{\beta}_{j+1}^{m-1}\parallel_2^2, \end{aligned} \tag{49}$$

where $\boldsymbol{\beta}_j^{m-1}=(\beta_1^m,...,\beta_j^m,\beta_{j+1}^{m-1},...,\beta_p^{m-1})$. The (49) 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 (49) over all the coordinates, we have for all m

$$Q(\boldsymbol{\beta}^m)-Q(\boldsymbol{\beta}^{m+1})\geq\theta\parallel\boldsymbol{\beta}^{m+1}-\boldsymbol{\beta}^m\parallel_2^2. \tag{50}$$

Since the (decreasing) sequence $Q(\boldsymbol{\beta}^m)$ converges, (50) 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) of Lemma 1 holds by (50). 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 computational efficiency for adaptive rescaling approach and the MMCD algorithm in MCP penalized logistic regression models. Average and standard error (SE) are computed based on 100 replicates for datasets with $n = 100$ and $p = 1,000$. IN, SP, PC, AR and CS are the five correlation structures among the penalized variables considered in the article. The time is measured in seconds.

Algorithm	IN(SE)	SP(SE)	PC(SE)	AR(SE)	CS(SE)
SNR	4.34(0.03)	3.10(0.02)	3.90(0.02)	3.19(0.02)	3.05(0.02)
MMCD	188.64(0.89)	105.98(0.56)	107.78(0.49)	119.51(0.55)	107.08(0.42)
Adap Rescaling	374.18(1.38)	201.93(1.10)	206.11(1.14)	223.13(1.13)	206.74(1.40)

Table 2: Comparison of Lasso, SCAD and MCP in terms of model size (MS), false discover rate (FDR) and predictive AUC (P-AUC). Average and standard error are computed based on 1,000 replicates. The sample size in the training data is $n = 100$. Predictive AUC is the maximum predictive AUC of the validation dataset, which contains 2,000 observations from the same model generating the training data.

Structure	Penalty	SNR	MS(SE)	FDR(SE)	P-AUC(SE)
IN	Lasso	4.332	13.63 (0.34)	0.5908 (0.0067)	0.8315 (0.0012)
	SCAD		9.92 (0.25)	0.4451 (0.0078)	0.8558 (0.0010)
	MCP		7.95 (0.23)	0.3143 (0.0088)	0.8562 (0.0010)
SP	Lasso	3.069	18.40 (0.47)	0.6841 (0.0063)	0.7712 (0.0017)
	SCAD		7.14 (0.19)	0.3942 (0.0073)	0.8177 (0.0012)
	MCP		6.10 (0.17)	0.2983 (0.0076)	0.8185 (0.0012)
PC	Lasso	3.879	8.60 (0.22)	0.4330 (0.0067)	0.8726 (0.0006)
	SCAD		6.30 (0.14)	0.3311 (0.0068)	0.8806 (0.0005)
	MCP		5.78 (0.13)	0.2743 (0.0069)	0.8807 (0.0005)
AR	Lasso	3.204	6.01 (0.15)	0.4774 (0.0079)	0.8182 (0.0012)
	SCAD		4.83 (0.13)	0.3497 (0.0087)	0.8391 (0.0009)
	MCP		3.78 (0.11)	0.2214 (0.0081)	0.8394 (0.0009)
CS	Lasso	3.049	17.72 (0.49)	0.6792 (0.0063)	0.7723 (0.0016)
	SCAD		8.70 (0.28)	0.4468 (0.0083)	0.8086 (0.0015)
	MCP		7.32 (0.25)	0.3481 (0.0089)	0.8098 (0.0014)

Table 3: Application of Lasso, 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	p-AUC(SE)	MS(SE)
Lasso	0.7523 (0.0010)	46.97 (0.53)
SCAD	0.7563 (0.0010)	34.48 (0.51)
MCP	0.7565 (0.0010)	34.85 (0.50)

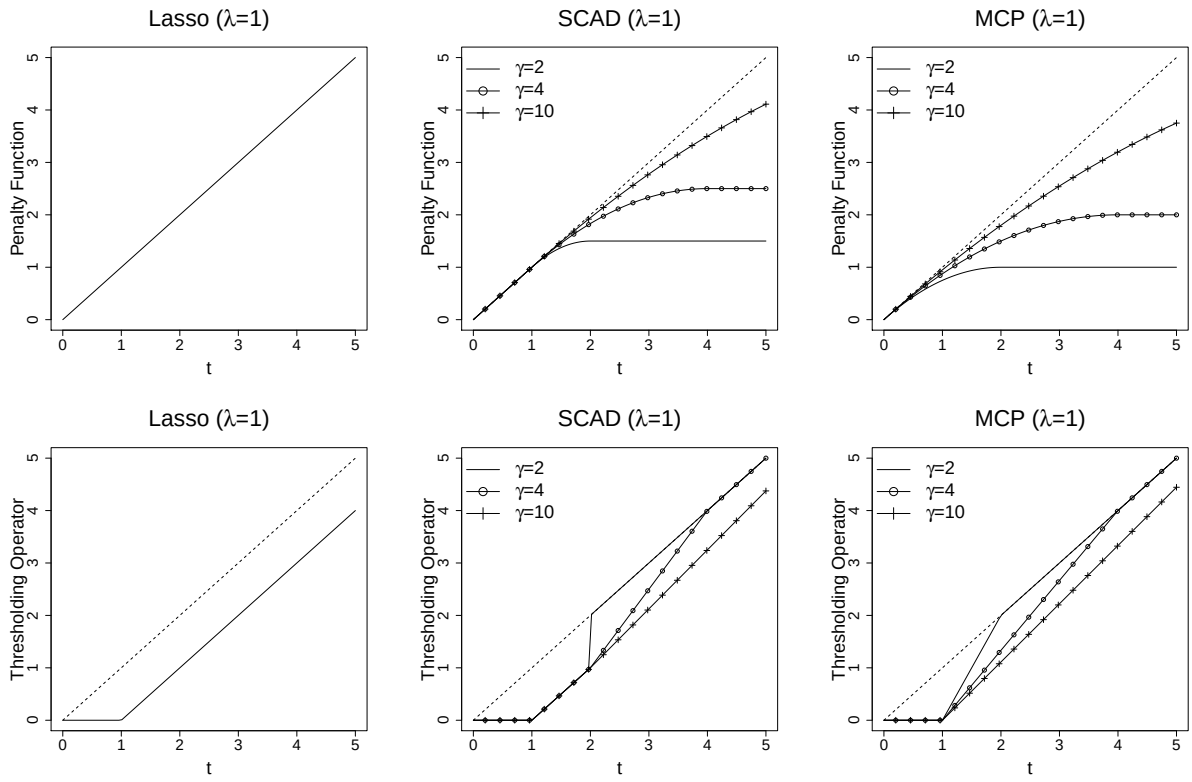

Figure 1: Penalty functions and threshold operator functions of Lasso(left), SCAD(middle) and MCP(right). The first row shows the penalty functions and the second row shows the operator functions. Lasso shrinks all coefficients without distinction. SCAD and MCP release the rate of penalization for larger coefficients. MCP reduces to Lasso if $\gamma \to +\infty$ and converges to hard-threshold penalty if $\gamma \to 1$. SCAD converges to Lasso when $\gamma \to +\infty$.

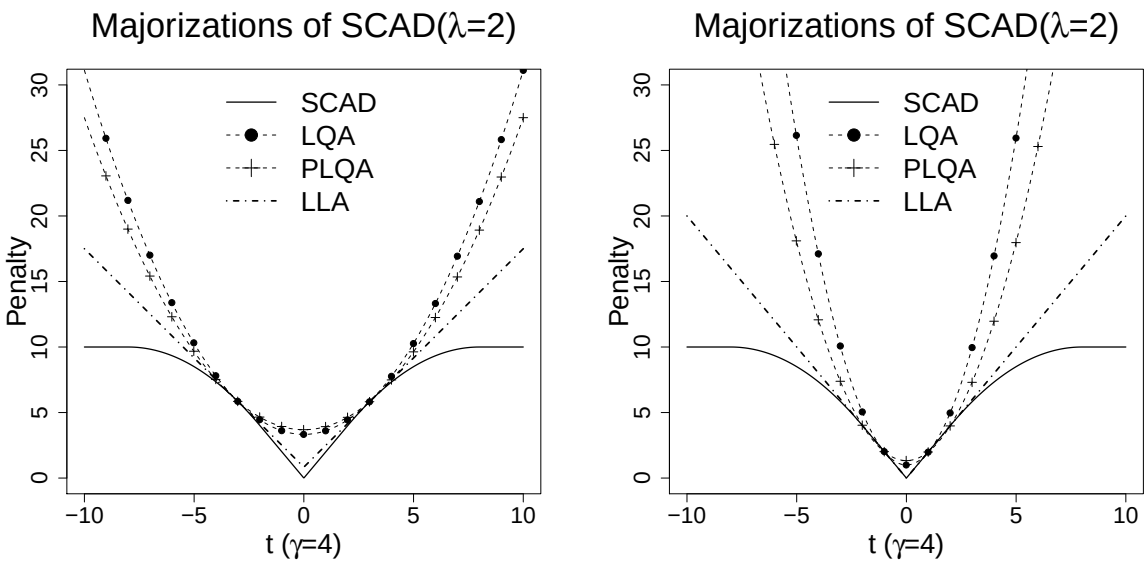

Figure 2: SCAD penalty and its majorizations, LQA, Perturbed LQA (PLQA) and LLA. The left plot is majorized at $t = 3$, the right one is majorized at $t = 1$. In the PLQA, τ_0 is chosen to be 0.5. All the curves are plotted using $\gamma = 4$ and $\lambda = 2$ for better illustration effect.

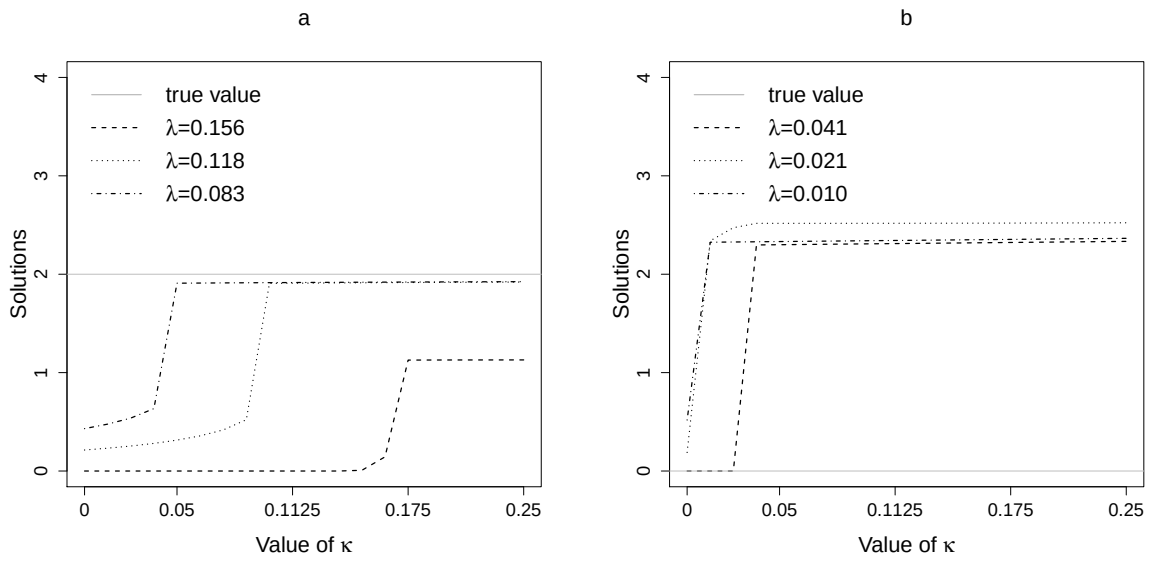

Figure 3: Plots of solution paths along κ . Plot a are paths for a causal variable, while b are paths for a null variable. Observe that although Lasso tends to over select in some cases, it could fail to select certain variables, which are selected by MCP (dash line in plot a).