Published in final edited form as:

Stat Comput. 2014 September ; 24(5): 871–883. doi:10.1007/s11222-013-9407-3.

Majorization Minimization by Coordinate Descent for Concave Penalized Generalized Linear Models

Dingfeng Jiang and

Exploratory Statistics, Data and Statistical Science, AbbVie Inc. dingfengjiang@gmail.com

Jian Huang

Department of Statistics and Actuarial Science, and Department of Biostatistics, University of Iowa

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 the 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 to 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 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; minimax concave penalty; variable selection

1 Introduction

Variable selection is a fundamental problem in statistics. Penalized methods have been shown to have attractive theoretical properties for variable selection in $p \gg n$ models, with n being the sample size and p the number of variables. 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)

SUPPLEMENTAL MATERIALS

R-package for the MMCD Algorithm: R-package 'cyplogistic' 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.

penalty (Fan and Li (2001)) and the minimax 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 variable 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 the computational algorithms for penalized regression models. Efron *et al* (2004) showed that a modified version of their LARS algorithm 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 the SCAD penalty, Fan and Li (2001) proposed a local quadratic approximation (LQA) algorithm. A drawback of the LQA algorithm is that once a coefficient is set to zero at any iteration step, it permanently stays at zero and the corresponding variable is 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. The LLA algorithm approximates the concave penalized solutions by repeatedly using the algorithms for the Lasso penalty. Schifano, Strawderman and Wells (2010) generalized the idea of the LLA algorithm via the 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 least squares 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. 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 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 for various combinations of (n, p) and various designs of covariate matrices. Their results showed that the CDA converges much faster than the LLA-LARS algorithm in the 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 solution for the MCP penalized generalized linear models (GLM). However, the adaptive

rescaling approach can not be applied to the SCAD penalty. Furthermore, it is not clear whether its solution reach a local optimal point of the original objective function.

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 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. This 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 a GLM. Section 3 describes the proposed MMCD algorithm, explains the benefits of majorization and studies its convergence property. 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. Concluding remarks are given in section 5.

2 Concave Penalized Solutions for GLMs

Let $\{(y_i, x_i)_{i=1}^n\}$ be the observed data, where y_i is a response variable and x_i is a $(p + 1)$ -dimensional vector of predictors, with the first element being 1 corresponding to the intercept. We consider a GLM with the canonical link function, in which y_i relates to x_i through a linear combination $\eta_i = x_i^T \beta$, with $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T \in \mathbb{R}^{p+1}$. Here β_0 is the intercept. The conditional density function of y_i given x_i is $f_i(y_i) = \exp\{(y_i \theta_i - \psi(\theta_i))/\phi_i + c(y_i, \phi)\}$. 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. The (scaled) negative log-likelihood function is

$$\ell(\beta) \propto \frac{1}{n} \sum_{i=1}^n \{\psi(x_i^T \beta) - y_i x_i^T \beta\}. \quad (1)$$

Here $x_{i0} = 1, 1 \leq i \leq n$. For the other p variables, we assume they are standardized, that is,

$\|x^j\|_2^2/n=1$ with $x^j = (x_{1j}, \dots, x_{nj})^T, 1 \leq j \leq p$. Here $\|\mathbf{v}\|_2$ is the ℓ_2 norm of a n -dimensional vector $\mathbf{v}$. The standardization allows the penalization to be applied evenly to each variable.

We consider the concave penalized GLM criterion

$$Q(\beta; \lambda, \gamma) = \ell(\beta) + \sum_{j=1}^p \rho(|\beta_j|; \lambda, \gamma), \quad (2)$$

where ρ is a penalty function with a penalty parameter $\lambda \geq 0$ and a regularization parameter γ that controls the shape of ρ . Note that in (2) the intercept β_0 is not penalized. We focus on two concave penalties, SCAD and MCP. The SCAD penalty (Fan and Li (2001)) is defined as

$$\rho(t; \lambda, \gamma) = \lambda \int_0^{|t|} \mathbf{1}_{\{x \leq \lambda\}} + \frac{(\gamma\lambda - x)_+}{(\gamma - 1)\lambda} \mathbf{1}_{\{x > \lambda\}} dx, \quad (3)$$

with $\lambda \geq 0$ and $\gamma > 2$. Here $\mathbf{1}_{x \in A}$ is the indicator function and $x_+ = x \mathbf{1}_{\{x \geq 0\}}$ denotes the non-negative part of x . The MCP penalty (Zhang (2010)) is defined as

$$\rho(t; \lambda, \gamma) = \lambda \int_0^{|t|} \left(1 - \frac{x}{\gamma\lambda}\right)_+ dx \quad (4)$$

with $\lambda \geq 0$ and $\gamma > 1$.

For the SCAD and MCP penalties, the regularization parameter γ controls the degree of concavity, a smaller γ corresponds to a penalty that is more concave. The two penalties begin by applying the same degree of penalization as Lasso, and then gradually reduce the penalization to zero as $|t|$ increases. When $\gamma \rightarrow \infty$, the SCAD and MCP penalties converge to the ℓ_1 penalty.

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

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

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 $S(t, \lambda) = \operatorname{sign}(t) (|t| - \lambda)_+$ for $\lambda \geq 0$ (Donoho and Johnstone (1994)), where $\operatorname{sign}(t) = 1, 0, -1$ if $t > 0, = 0, < 0$, respectively. Then the SCAD and MCP penalties have closed form solutions for $\hat{\theta}(\lambda, \gamma)$ as follows,

$$\hat{\theta}_{SCAD}(\lambda, \gamma) = S(\hat{\theta}_{LS}, \lambda) \mathbf{1}_{\{|\hat{\theta}_{LS}| \leq 2\lambda\}} + \frac{\gamma - 1}{\gamma} S(\hat{\theta}_{LS}, \lambda \gamma / (\gamma - 1)) \mathbf{1}_{\{2\lambda < |\hat{\theta}_{LS}| \leq \gamma\lambda\}} + \hat{\theta}_{LS} \mathbf{1}_{\{|\hat{\theta}_{LS}| > \gamma\lambda\}}, \quad (5)$$

$$\hat{\theta}_{MCP}(\lambda, \gamma) = \frac{\gamma}{\gamma - 1} S(\hat{\theta}_{LS}, \lambda) \mathbf{1}_{\{|\hat{\theta}_{LS}| \leq \lambda\gamma\}} + \hat{\theta}_{LS} \mathbf{1}_{\{|\hat{\theta}_{LS}| > \lambda\gamma\}}. \quad (6)$$

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

3 Majorization Minimization by Coordinate Descent

For a GLM, a quadratic approximation to $\ell(\beta)$ in a neighborhood of a given estimate $\tilde{\beta}$ leads to a weighed least squares loss,

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

with $w_i(\tilde{\beta}) = \ddot{\psi}(x_i^T \tilde{\beta})$ and $z_i(\tilde{\beta}) = \ddot{\psi}(x_i^T \tilde{\beta})^{-1} \{y_i - \dot{\psi}(x_i^T \tilde{\beta})\} + x_i^T \tilde{\beta}$, 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 $\ell^s(\beta|\tilde{\beta})$ in the criterion function, the CDA updates the j th coordinate by fixing the remaining k ($k \neq j$)

coordinates. Let $\hat{\beta}_j^m = (\hat{\beta}_0^{m+1}, \dots, \hat{\beta}_j^{m+1}, \hat{\beta}_{j+1}^m, \dots, \hat{\beta}_p^m)^T$, the CDA updates $\hat{\beta}_{j-1}^m$ to $\hat{\beta}_j^m$ by minimizing the criterion

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

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

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

with $\rho'(|t|)$ the first derivative of $\rho(|t|)$ w.r.t. $|t|$ for $t \neq 0$. Note that for $t = 0$, we have $\rho'(0+) = \lambda$. The notation $\operatorname{sgn}(t)$ is the subdifferential of $|t|$, that is,

$$\operatorname{sgn}(t) = \begin{cases} 1, & \text{if } t > 0; \\ -1, & \text{if } t < 0; \\ \in [-1, 1], & \text{if } t = 0. \end{cases}$$

We refer to Lange (2004) for a detailed description of sub-differentials.

Define the scaling factor by $\delta_j \triangleq n^{-1} \sum_{i=1}^n w_i x_{ij}^2$. Directly solving (9) with the MCP penalty gives the j th coordinate-wise solution as,

$$\hat{\beta}_j^{m+1} = \frac{S(\tau_j, \lambda)}{\delta_j - 1/\gamma} \mathbf{1}_{\{|\tau_j| \leq \delta_j \gamma \lambda\}} + \frac{\tau_j}{\delta_j} \mathbf{1}_{\{|\tau_j| > \delta_j \gamma \lambda\}}, \quad (10)$$

with $\tau_j = n^{-1} \sum_{i=1}^n w_i x_{ij} (z_i - x_i^T \hat{\beta}_{j-1}^m) + \delta_j \hat{\beta}_j^m$. Observe that when $|\tau_j| \leq \lambda$, we have

$\hat{\beta}_j^{m+1} = 0$. In a linear model, $w_i = 1$, $i = 1, \dots, n$. Thus the scaling factor $\delta_j = 1$ for

standardized predictors. In a GLM, however, the dependence of w_i on $(\hat{\beta}_{j-1}^m, x_i, y_i)$ 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

$$\hat{\beta}_j^{m+1} = \frac{S(\tau_j, \lambda)}{\delta_j(1 - 1/\gamma)} \mathbf{1}_{\{|\tau_j| \leq \gamma\lambda\}} + \frac{\tau_j}{\delta_j} \mathbf{1}_{\{|\tau_j| > \gamma\lambda\}}, \quad (11)$$

for the j th coordinate-wise solution. This is equivalent to applying 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, a small δ_j could cause convergence issues. The adaptive rescaling approach cannot be applied to the SCAD penalty because the scaled soft-thresholding operation only applies to the middle clause of the SCAD thresholding operator as shown in (5).

Observe that in a GLM, the scaling factor δ_j is equal to the second partial derivative of the loss function, that is, $\nabla_j^2 \ell(\beta) = \sum \ddot{\psi}(x_j^T \beta) x_{ij}^2 / n = \sum w_i x_{ij}^2 / n$. The MMCD algorithm seeks an upper bound of $\nabla_j^2 \ell(\beta)$. Thus we need to find an M such that $\delta_j \leq M$ for each coordinate. Such idea of finding a uniform bound on the second derivative was also proposed by Böhning and Lindsay (1988).

For the MM approach, the use of the upper bound of δ_j is equivalent to finding a surrogate function $\ell^{MM}(\beta_j | \hat{\beta}_{j-1}^m)$ that majorizes $\ell^s(\beta_j | \hat{\beta}_{j-1}^m)$, where

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

and

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

with the second partial derivative $\nabla_j^2 \ell(\hat{\beta}_{j-1}^m)$ in the Taylor expansion replaced by its upper bound M . Note that for a given $\hat{\beta}_{j-1}^m$, $\nabla_j^2 \ell(\hat{\beta}_{j-1}^m)$ is a constant, however, the value of $\nabla_j^2 \ell(\hat{\beta}_{j-1}^m)$ changes from iteration to iteration. The majorization uses the upper bound to avoid these changes. And such majorization is valid regardless the current estimate of $\hat{\beta}_{j-1}^m$. This is different from common implementation of the MM approach, where the majorization is constructed at the current estimation point. Also note that the majorization is applied coordinate-wisely to better fit the CDA. The descent property of the MM approach ensures that the iterative minimization of $\ell^{MM}(\beta_j | \hat{\beta}_{j-1}^m)$ 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 j th coordinate-wise solutions of the SCAD and MCP are

$$\text{SCAD: } \hat{\beta}_j^{m+1} = \frac{1}{M} S(\tau_j, \lambda) \mathbf{1}_{\{|\tau_j| \leq (1+M)\lambda\}} + \frac{S(\tau_j, \gamma\lambda/(\gamma-1))}{M-1/(\gamma-1)} \mathbf{1}_{\{(1+M)\lambda < |\tau_j| \leq M\gamma\lambda\}} + \frac{1}{M} \tau_j \mathbf{1}_{\{|\tau_j| > M\gamma\lambda\}}, \quad (14)$$

$$\text{MCP: } \hat{\beta}_j^{m+1} = \frac{S(\tau_j, \lambda)}{M-1/\gamma} \mathbf{1}_{\{|\tau_j| \leq M\gamma\lambda\}} + \frac{1}{M} \tau_j \mathbf{1}_{\{|\tau_j| > M\gamma\lambda\}}, \quad (15)$$

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

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

with $\tau_0 = M\hat{\beta}_0^m + n^{-1} \sum_{i=1}^n x_{i0}(y_i - \dot{\psi}(x_i^T \hat{\beta}^m))$, where $\hat{\beta}^m = (\hat{\beta}_0^m, \hat{\beta}_1^m, \dots, \hat{\beta}_p^m)^T$. In (14) and (15), we want to ensure that the denominators are positive, i.e. $M-1/(\gamma-1) > 0$ 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 the SCAD and MCP, the condition is satisfied by choosing a proper γ . Since $\inf_t \rho''(|t|; \lambda, \gamma) = -1/(\gamma-1)$ for the SCAD penalty, and $\inf_t \rho''(|t|; \lambda, \gamma) = -1/\gamma$ for the MCP, 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 $\eta = (\eta_1, \dots, \eta_p)^T$ and $X = (x_1^T, \dots, x_n^T)^T$, and $\hat{\eta}_j^m = X \hat{\beta}_j^m$ be the linear component corresponding to $\hat{\beta}_j^m$. Further efficiency can be achieved by using the equation

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

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

We can also majorize the penalty in each iteration using LQA (Fan and Li (2001)), perturbed LQA (Hunter and Li (2005)) or LLA (Zou and Li (2008)). However, this does not take full advantage of CDA. Indeed, the approximation of the penalty requires additional iterations for convergence and is not necessary, since an exact coordinate-wise solution exists. Thus the MMCD uses the exact form of the penalties and only majorizes the loss.

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

- a. The second partial derivative of $\ell(\beta)$ w.r.t. β_j is uniformly bounded for standardized X , i.e. there exists a real number $M > 0$ such that $\nabla_j^2 \ell(\beta) \leq M$ for all β and $j = 0, \dots, p$.
- 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 then proceeds as follows,

1. Given an initial value of $\beta^{\hat{0}}$, the MMCD computes the corresponding linear component $\eta^{\hat{0}}$.
2. For $m = 0, 1, \dots$, the MMCD updates the intercept by (16), and uses (14) or (15) to update $\hat{\beta}_j^m$ to $\hat{\beta}_{j+1}^m$ for the penalized variables. After each iteration, it also computes the corresponding linear component $\hat{\eta}_{j+1}^m$ using (17). Then the MMCD cycles through all the coordinates such that β^m is updated to β^{m+1} .
3. The MMCD checks the convergence criterion. If the algorithm converges then stops iteration, otherwise it repeats step 2 until the algorithm converges.

3.1 Convergence analysis of MMCD

Theorem 1 establishes that under certain conditions, the MMCD solution converges to a minimum of the objective function.

Theorem 1 Consider the objective function (2), where the given data (y, X) lies in a compact set and no two columns of X are identical. For a given (λ, γ) , suppose the penalty $\rho(|t|; \lambda, \gamma) \equiv \rho(t)$ satisfies $\rho(t) = \rho(-t)$ and $\rho'(|t|)$ is a non-negative, uniformly bounded function, 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 $\{\beta^m\}$ converges to a minimum of the function $Q(\beta; \lambda, \gamma)$.

The condition on (y, X) is mild. The standardization of the 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 convergence results of Mazumder, Friedman and Hastie (2011) for their SparseNet algorithm with a least squares loss. Theorem 1 covers more general loss functions.

4 The MMCD for Penalized Logistic Regression

As mentioned in the introduction, the MMCD algorithm is particularly suitable for a logistic regression, which is one of the most widely used models in biostatistical applications. For a logistic regression model, the response 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 $\nabla_j \ell(\hat{\beta}) = -(x^j)^T (y - \pi)/n$ and $\nabla_j^2 \ell(\hat{\beta}) = n^{-1} \sum w_i x_{ij}^2$, with $w_i = \pi_i(1 - \pi_i)$ and π_i being the estimated probability of the i th observation given the current estimate $\hat{\beta}$, i.e. $\hat{\pi}_i = 1/(1 + \exp(-x_i^T \hat{\beta}))$. Since for any $0 < \pi < 1$, $\pi(1 - \pi) \leq 1/4$ (Böhning and Lindsay (1988)), then the upper bound of $\nabla_j^2 \ell(\hat{\beta})$ is $M = 1/4$ for standardized x^j . Correspondingly $\tau_j = 4^{-1} \beta_j + n^{-1} (x^j)^T (y - \pi)$ for $j = 0, \dots, p$. By the condition (b), we require $\gamma > 5$ for the SCAD penalty and $\gamma > 4$ for the MCP penalty.

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 γ . For example, for linear regression models with standardized variables, it has been suggested one uses $\gamma \approx 3.7$ for the SCAD (Fan and Li (2001)) and $\gamma \approx 2.7$ (Zhang (2010)) for the 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.

Define the grid values in $[0, \kappa_{\max}) \times [\lambda_{\min}, \lambda_{\max}]$ to be $0 = \kappa_1 < \kappa_2 < \dots < \kappa_K < \kappa_{\max}$ and $\lambda_{\max} = \lambda_1 > \lambda_2 > \dots > \lambda_V = \lambda_{\min}$. The number of grid points K and V are pre-specified. In our implementation, the κ -grid points are uniform in the standard scale while those for λ are uniform in the log scale. The $\kappa_{\max}$ is the largest value of κ such that the condition (b) of the MMCD algorithm is valid. We have $\kappa_{\max} = 1/5$ for the SCAD and $\kappa_{\max} = 1/4$ for the MCP. Note that when $\kappa = 0$, the SCAD and MCP penalties both become the Lasso penalty. The $\lambda_{\max}$ is the smallest value of λ such that $\beta_j = 0, j = 1, \dots, p$. For the logistic regression model, $\lambda_{\max} = n^{-1} \max_j |(x^j)^T (y - \pi)|$ for every κ_k with $\pi = \bar{y}J$ and J being a vector whose elements are all equal to 1. We set $\lambda_{\min} = \varepsilon \lambda_{\max}$, with $\varepsilon = 0.0001$ if $n > p$ and $\varepsilon = 0.01$ 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 $\hat{\beta}_{\kappa_k, \lambda_v}$.

We follow the approach of Mazumder, Friedman and Hastie (2011) to compute the solution surface by initializing the MMCD 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. The approach first computes the Lasso solution along λ . When computing $\hat{\beta}_{\kappa_0, \lambda_{v+1}}$, it uses $\hat{\beta}_{\kappa_0, \lambda_v}$ as the initial value in the MMCD algorithm.
2. For a given λ_v , the approach computes the solution along κ . That is the $\hat{\beta}_{\kappa_k, \lambda_v}$ is used as the initial value in computing the solution of $\hat{\beta}_{\kappa_{k+1}, \lambda_v}$.
3. The approach then cycles through $v = 1, \dots, V$ 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 (1) presents the solution paths of a causal variable with $\beta = 2$ 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 result. This is the reason why we prefer a data-driven procedure to choose both κ and λ .

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 $A_0 \equiv \{1 - j - p : \beta_j \neq 0\}$ 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 $(\pm 0.6, \pm 1.2, 2.4, \mp 0.6, \mp 1.2, \pm 2.4)^T$ such that the signal-to-noise ratio (SNR), defined as $\text{SNR} = \sqrt{\beta^T X^T X \beta / n}$, is approximately in the range of (3,4). The covariates are generated from a multivariate normal distribution with zero means and variance Σ , with Σ being a positive-definite matrix with

dimension $p \times p$. The values of the response variable are generated from a Bernoulli distribution with $y_i \sim \text{Bernoulli}(1, p_i)$, and $p_i = \exp(\beta^T x_i) / (1 + \exp(\beta^T x_i))$ for $i = 1, \dots, n$.

We consider five types of correlation structures for Σ .

- (a) Independent structure (IN) for the p penalized variables. Here $\Sigma = I_p$, with I_p being the identity matrix of dimension $p \times p$.
- (a) Separate structure (SP). The causal and null variables are independent. Let Σ_0 and Σ_1 be the covariance matrix for the causal and the null variables, respectively, then $\Sigma = \text{block diagonal}(\Sigma_0, \Sigma_1)$. 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, $\Sigma = \text{block diagonal}(\Sigma_a, \Sigma_b, \Sigma_c)$, with Σ_a being the covariance matrix for the first 5 causal variables; Σ_b being the covariance matrix for the remaining 5 causal variables and 5 null variables; Σ_c being the covariance matrix for the remaining null variables. We also assume a compound symmetry structure within $\Sigma_a, \Sigma_b, \Sigma_c$.
- (d) First-order auto-regressive (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 $\hat{\beta}^m$. For the logistic regression, we also use a quadratic approximation (7) to the loss based on $\hat{\beta}^m$. To compute $\hat{\beta}^{m+1}$ using the LLA, we minimize a Lasso-type criterion

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

To compare the MMCD with the LLA, we implement 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 $\hat{\beta}^m$, with $\dot{\rho}(t)$

being the derivative of $\rho(\cdot)$. The computation of $\hat{\beta}^{m+1}$ 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 CDA 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 $\hat{\beta}$ 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 are programmed in FORTRAN with similar programming structures for fair comparison. We observed 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. We use the convergence criterion $\|\hat{\beta}^{m+1} - \hat{\beta}^m\|_2 / (\|\hat{\beta}^m\|_2 + \delta) < \varepsilon$. We choose $\delta = 0.01$ and $\varepsilon = 0.001$ if $n > p$ and $\varepsilon = 0.01$ if $n < p$. We set correlation coefficient $\rho = 0.5$, the number of grids $K = 10$, $V = 100$.

4.4 Comparison of computational efficiency

We first provide a brief description of the available algorithms, namely the LQA, perturbed LQA, LLA and MIST. These four algorithms share the same spirit in the sense that they all use a surrogate function to majorize the penalty $\rho(|t|; \lambda, \gamma)$. The LQA uses the following approximation,

$$\rho(|t|; \lambda, \gamma) \approx \rho(|t_0|; \lambda, \gamma) + \frac{\rho'(|t_0|; \lambda, \gamma)}{2|t_0|}(t^2 - t_0^2), t \approx t_0. \quad (19)$$

Then a Newton-Raphson type iteration is employed to minimize the penalized criterion with the surrogate penalty function. When t_0 is close to zero, the algorithm is unstable. Fan and Li (2001) suggested that if $\hat{\beta}_j$ is small enough, say $|\hat{\beta}_j| < \varepsilon$ (a pre-specified value), one sets $\hat{\beta}_j = 0$ and removes the j th variable from iterations. Hence, a drawback of LQA algorithm is that once a variable is removed in any iteration, it is necessarily excluded from the final model. Hunter and Li (2005) proposed a perturbed version of LQA to majorize the LQA.

$$\rho(|t|; \lambda, \gamma) \approx \rho(|t_0|; \lambda, \gamma) + \frac{\rho'(|t_0|; \lambda, \gamma)}{2|t_0 + \tau_0|}(t^2 - t_0^2), t \approx t_0. \quad (20)$$

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. The LLA algorithm by (Zou and Li (2008)) approximates the penalty by

$$\rho(|t|; \lambda, \gamma) \approx \rho(|t_0|; \lambda, \gamma) + \rho'(|t_0|; \lambda, \gamma)(|t| - |t_0|), t \approx t_0. \quad (21)$$

The MIST (Schifano, Strawderman and Wells (2010)) algorithm extends the LLA to several other penalties.

We focus on the comparison of computational efficiency of the LLA, adaptive rescaling, LLA-CD and MMCD for the MCP penalized logistic regression models since the adaptive rescaling approach can only be applied to the MCP penalty. The computation is done on Inter Xeon (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 2 shows the average elapsed time 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. The speed of the LLA-CD is also impacted by the p/n ratio, although to a less

degree. The MMCD and LLA are fairly stable to the change of p/n ratio. Overall, the MMCD is the fastest one. It is worth noting that in the setting with $n = 1,000$, $p = 500$, the adaptive rescaling and LLA-CD did not converge within 1,000 iterations for a convergence criterion of $\varepsilon = 0.001$ in some replications.

For high dimensional data with $p \gg n$, we focus on the comparison between the adaptive rescaling, LLA-CD and MMCD. We used two sample sizes $n = 100$ and $n = 300$. The numbers of variables, p , is set to be 500, 1, 000, 2, 000, 5,000 and 10, 000. Figure 3 presents the average elapsed times in seconds based on 100 replications for $n = 300$. It shows that as p increases, the advantage of the MMCD becomes more apparent. For a fixed p , the MMCD gains more efficiency when the predictors are correlated and n is large. In addition, the MMCD has the smallest standard error of computation time, followed by the LLA-CD and adaptive rescaling. This suggests the MMCD is the most stable one among the three algorithms considered here in high-dimensional settings.

4.5 Comparison of selection performance

We further compare the selection performance of the LLA, adaptive rescaling, LLA-CD and MMCD 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 models 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 $\hat{\beta}_{\kappa_k, \lambda_v}$, we compute the predictive area under ROC curve (PAUC) $AUC_{(\kappa_k, \lambda_v)}$ for each $\hat{\beta}_{\kappa_k, \lambda_v}$ based on a validation set with $n^* = 3,000$. The model corresponding to the maximum predictive $AUC_{(\kappa_k, \lambda_v)}$ is selected as the final model for comparison. We compare the results 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 have similar performance, both having smaller model size and FDR than the adaptive rescaling and LLA. Table 2 presents the comparison among the adaptive rescaling, LLA-CD and MMCD 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 terms of model size and FDR, the MMCD and LLA-CD have 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 the 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, and the MCP penalty using the adaptive rescaling, LLA-CD and MMCD. The PAUCs from different approaches are close to each other. The model size of the MCP penalty by the LLA-CD algorithm happens to be the largest.

Finally, we 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. 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 using the MMCD, the MCP using the adaptive rescaling and the LLA-CD select the same model with 67 variables and CV-AUC = 0.7808. The MCP using the MMCD selects 16 variables with CV-AUC = 0.8024.

5 Discussion

In this article, we propose the MMCD algorithm for computing the concave penalized solutions. Our simulation studies and data example demonstrate that this algorithm is efficient in concave penalized logistic regression models with $p \gg n$. Unlike the existing algorithms, such as the LQA, LLA and MIST that approximate the penalty function, the MMCD seeks a closed form solution for each coordinate 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 is proved under reasonable conditions.

The comparison among the LLA, adaptive rescaling, LLA-CD and MMCD indicates that the MMCD is the more efficient approach, especially for large p and correlated covariates, although the LLA-CD is competitive in some cases. The LLA-CD implements the adjacent initiation idea to reduce the computational cost, that is, it uses $\hat{\beta}_{\kappa_k, \lambda_v}$ as the initial values to compute $\hat{\beta}_{\kappa_{k+1}, \lambda_v}$. 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 $\hat{\beta}^m$ to compute $\hat{\beta}^{m+1}$. 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 reason why the two implementations of the LLA perform so differently.

The application of the MMCD to logistic regression is facilitated by the fact that a simple and effective majorization function can be constructed for the logistic likelihood. Majorization functions can also be found for other GLM models, for example the baseline

category logit model. In such models, the MMCD can be implemented in a similar way. 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 authors thank the reviewers and the associate editor for their helpful comments that led to considerable improvements in the paper. The research of Huang is supported in part by NIH grants R01CA120988, R01CA142774 and NSF grant DMS 1208225.

References

- Böhning D, Lindsay B. Monotonicity of quadratic-approximation algorithms. *Ann. Inst. Stat. Math.* 1988; 40(4):641–663.
- Breheny P, Huang J. Coordinate descent algorithms for non-convex penalized regression, with application to biological feature selection. *Ann. Appl. Stat.* 2011; 5(1):232–253. [PubMed: 22081779]
- Donoho DL, Johnstone JM. Ideal spatial adaptation by wavelet shrinkage. *Biometrika.* 1994; 81(3): 425–455.
- Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. *Ann. Stat.* 2004; 32(2):407–451.
- Fan J, Li R. Variable selection via non-concave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* 2001; 96(456):1348–13608.
- Friedman J, Hastie T, Höfling H, Tibshirani R. Pathwise coordinate optimization. *Ann. Appl. Stat.* 2007; 1(2):302–332.
- Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* 2010; 33(1):1–22. [PubMed: 20808728]
- Hunter DR, Lange K. A tutorial on MM algorithms. *Am. Stat.* 2004; 58(1):30–37.
- Hunter DR, Li R. Variable selection using MM algorithms. *Ann. Stat.* 2005; 33(4):1617–1642. [PubMed: 19458786]
- Jiang D, Huang J, Zhang Y. The cross-validated AUC for MCP-Logistic regression with high-dimensional data. *Stat. Methods. Med. Res.* 2011 Accepted.
- Lange K. *Optimization*. New York: Springer; 2004.
- Lange K, Hunter D, Yang I. Optimization transfer using surrogate objective functions (with discussion). *J. Comput. Graph. Stat.* 2000; 9(1):1–59.
- Mazumder R, Friedman J, Hastie T. *SparseNet*: Coordinate descent with non-convex penalties. *J. Am. Stat. Assoc.* 2011; 106(495):1125–1138.
- Ortega, JM.; Rheinbold, WC. *Iterative solution of nonlinear equations in several variables*. New York, NY: Academic Press; 1970.
- Osborne MR, Presnell B, Turlach BA. A new approach to variable selection in least square problems. *IMA. J. Numer. Anal.* 2000; 20(3):389–403.
- Schifano ED, Strawderman RL, Wells MT. Majorization-minimization algorithms for non-smoothly penalized objective functions. *Electronic Journal of Statistics.* 2010; 4:1258–1299.
- Tibshirani R. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Ser. B.* 1996; 58(1): 267–288.
- Tseng P. Convergence of a Block Coordinate Descent Method for Non-differentiable Minimization. *J. Optimiz. Theory. App.* 2001; 109(3):475–494.
- van't Veer LJ, Dai H, van de Vijver MJ, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature.* 2002; 415(31):530–536. [PubMed: 11823860]
- van de Vijver MJ, He YD, van't Veer LJ, et al. A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.* 2002; 347(25):1999–2009. [PubMed: 12490681]

- Warge J. Minimizing certain convex functions. SIAM J. Appl. Math. 1963; 11(3):588–593.
- Wu TT, Lange K. Coordinate descent algorithms for Lasso penalized regression. Ann. Appl. Stat. 2008; 2(1):224–244.
- Zhang CH. Nearly unbiased variable selection under minimax concave penalty. Ann. Stat. 2010; 38(2): 894–942.
- Zou H, Li R. One-step sparse estimates in non-concave penalized likelihood models. Ann. Stat. 2008; 36(4):1509–1533. [PubMed: 19823597]
- R Development Core Team. R Foundation for Statistical Computing. Vienna, Austria: ISBN; R: a Language and environment for statistical computing. 3-900051-07-0, <http://www.R-project.org>

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 (y, X) lies on a compact set and the following conditions hold:

1. The loss function $\ell(\beta)$ is (total) differentiable w.r.t. β for any $\beta \in \mathbb{R}^{p+1}$.
2. The penalty function $\rho(t)$ is symmetric around 0 and is differentiable on $t \neq 0$; $\rho'(|t|)$ is non-negative, continuous and uniformly bounded, where $\rho'(|t|)$ is the derivative of $\rho(|t|)$ w.r.t. $|t|$.
3. The sequence $\{\beta^k\}$ is bounded.
4. For every convergent subsequence $\{\beta^{n_k}\} \subset \{\beta^n\}$, the successive differences converge to zero: $\beta^{n_k} - \beta^{n_k-1} \rightarrow 0$.

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

$$\lim_{\alpha \downarrow 0+} \inf \left\{ \frac{Q(\beta^\infty + \alpha \delta) - Q(\beta^\infty)}{\alpha} \right\} \geq 0, \quad (22)$$

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

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

$$\lim_{\alpha \downarrow 0+} \inf \left\{ \frac{Q(\beta + \alpha \delta_j) - Q(\beta)}{\alpha} \right\} = \nabla_j \ell(\beta) \delta_j + \lim_{\alpha \downarrow 0+} \inf \left\{ \frac{\rho(|\beta_j + \alpha \delta_j|) - \rho(|\beta_j|)}{\alpha} \right\} = \nabla_j \ell(\beta) \delta_j + \partial \rho(\beta_j; \delta_j), \quad (23)$$

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

$$\partial \rho(\beta_j; \delta_j) = \begin{cases} \rho'(|\beta_j|) \text{sgn}(\beta_j) \delta_j, & |\beta_j| > 0; \\ \rho'(0) |\delta_j|, & |\beta_j| = 0, \end{cases} \quad (24)$$

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

$$\beta_j^{n_k-1} = (\beta_0^{n_k}, \dots, \beta_{j-1}^{n_k}, \beta_j^{n_k}, \beta_{j+1}^{n_k-1}, \dots, \beta_p^{n_k-1}) \rightarrow (\beta_0^\infty, \dots, \beta_{j-1}^\infty, \beta_j^\infty, \beta_{j+1}^\infty, \dots, \beta_p^\infty) \quad (25)$$

By (24) and (25), we have the results below for $j \in \{1, \dots, p\}$.

$$\partial \rho(\beta_j^{n_k}; \delta_j) \rightarrow \partial \rho(\beta_j^\infty; \delta_j) \text{ if } \beta_j^\infty \neq 0; \partial \rho(\beta_j^\infty; \delta_j) \geq \lim_k \inf \partial \rho(\beta_j^{n_k}; \delta_j) \text{ if } \beta_j^\infty = 0. \quad (26)$$

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

$$\nabla_j \ell(\beta_j^{n_k-1}) \delta_j + \partial \rho(\beta_j^{n_k}; \delta_j) \geq 0, \text{ for all } k. \quad (27)$$

Thus (26, 27) implies that for all $j \in \{1, \dots, p\}$,

$$\nabla_j \ell(\beta^\infty) \delta_j + \partial \rho(\beta_j^\infty; \delta_j) \geq \lim_k \inf \{ \nabla_j \ell(\beta_j^{n_k-1}) \delta_j + \partial \rho(\beta_j^{n_k}; \delta_j) \} \geq 0. \quad (28)$$

By (23, 28), for $j \in \{1, \dots, p\}$, we have

$$\lim_{\alpha \downarrow 0+} \inf \left\{ \frac{Q(\beta^\infty + \alpha \delta_j) - Q(\beta^\infty)}{\alpha} \right\} \geq 0. \quad (29)$$

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

$$\nabla_0 \ell(\beta^\infty) \delta_0 \geq 0. \quad (30)$$

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

$$\begin{aligned} & \lim_{\alpha \downarrow 0+} \inf \left\{ \frac{Q(\beta^\infty + \alpha \delta) - Q(\beta^\infty)}{\alpha} \right\} \\ &= \nabla_0 \ell(\beta^\infty) \delta_0 \\ &+ \sum_{j=1}^p \left[\nabla_j \ell(\beta^\infty) \delta_j + \lim_{\alpha \downarrow 0+} \inf \left\{ \frac{\rho(|\beta_j^\infty + \alpha \delta_j|) - \rho(|\beta_j^\infty|)}{\alpha} \right\} \right] \\ &= \nabla_0 \ell(\beta^\infty) \delta_1 \\ &+ \sum_{j=1}^p \lim_{\alpha \downarrow 0+} \inf \left\{ \frac{Q(\beta^\infty + \alpha \delta_j) - Q(\beta^\infty)}{\alpha} \right\} \geq 0, \end{aligned} \quad (31)$$

by (29, 30). This completes the proof.

Proof of Theorem 1

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

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

$$\chi(u+\delta)-\chi(u)=\ell(\beta_0,\dots,\beta_{j-1},u+\delta,\beta_{j+1},\dots,\beta_p)-\ell(\beta_0,\dots,\beta_{j-1},u,\beta_{j+1},\dots,\beta_p)+\rho(|u+\delta|)-\rho(|u|) \quad (32)$$

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

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

We re-write the RHS of (33) as follows:

$$\begin{aligned} RHS(\text{of } 33) &= \nabla_j \ell(\beta_0, \dots, \beta_{j-1}, u, \beta_{j+1}, \dots, \beta_p) \delta \\ &\quad + (\nabla_j^2 \ell(\beta_0, \dots, \beta_{j-1}, u, \beta_{j+1}, \dots, \beta_p) \\ &\quad - M) \delta^2 \\ &\quad + \rho'(|u|) \text{sgn}(u) \delta \\ &\quad + \rho'(|u|)(|u \\ &\quad + \delta| - |u|) \\ &\quad - \rho'(|u|) \text{sgn}(u) \delta \\ &\quad + \frac{1}{2} \rho''(|u^*|)(|u+\delta| - |u|)^2 \\ &\quad + (M - \frac{1}{2} \nabla_j^2 \ell(\beta_0, \dots, \beta_{j-1}, u, \beta_{j+1}, \dots, \beta_p)) \delta^2 + o(\delta^2). \end{aligned} \quad (34)$$

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

$$Q_j(u|\beta) = \ell(\beta) + \nabla_j \ell(\beta)(u - \beta_j) + \frac{1}{2} \nabla_j^2 \ell(\beta)(u - \beta_j)^2 + \rho(|u|), \quad (35)$$

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

$$\tilde{Q}_j(u|\beta) = \ell(\beta) + \nabla_j \ell(\beta)(u - \beta_j) + \frac{1}{2} M(u - \beta_j)^2 + \rho(|u|). \quad (36)$$

Since u is to minimize (36), we have, for the j th ($j \in \{1, \dots, p\}$) coordinate,

$$\nabla_j \ell(\beta) + M(u - \beta_j) + \rho'(|u|) \text{sgn}(u) = 0, \quad (37)$$

Because $\chi(u)$ is minimized at u_0 , by (37), we have

$$\begin{aligned}
 0 &= \nabla_j \ell(\beta_0, \dots, \beta_{j-1}, u_0 \\
 &\quad + \delta, \beta_{j+1}, \dots, \beta_p) \\
 &\quad + M(u_0 \\
 &\quad - u_0 - \delta) + \rho'(|u_0|) \operatorname{sgn}(u_0) \\
 &= \nabla_j \ell(\beta_0, \dots, \beta_{j-1}, u_0, \beta_{j+1}, \dots, \beta_p) \\
 &\quad + \nabla_j^2 \ell(\beta_0, \dots, \beta_{j-1}, u_0, \beta_{j+1}, \dots, \beta_p) \delta \\
 &\quad + o(\delta) \\
 &\quad - M\delta + \rho'(|u_0|) \operatorname{sgn}(u_0),
 \end{aligned} \tag{38}$$

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

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

$$\rho'(|u|)(|u+\delta| - |u|) - \rho'(|u|) \operatorname{sgn}(u) \delta = \rho'(|u|)[(|u+\delta| - |u|) - \operatorname{sgn}(u) \delta] \geq 0 \tag{39}$$

Therefore using (38, 39) in (34) at u_0 , we have, for $j \in \{1, \dots, p\}$,

$$\begin{aligned}
 \chi(u_0 + \delta) - \chi(u_0) &\geq \frac{1}{2} \rho''(|u^*|)(|u+\delta| - |u|)^2 \\
 &\quad + \delta^2 (M \\
 &\quad - \frac{1}{2} \nabla_j^2 \ell(\beta_0, \dots, \beta_{j-1}, u_0, \beta_{j+1}, \dots, \beta_p)) \\
 &\quad + o(\delta^2) \geq \frac{1}{2} M \delta^2 \\
 &\quad + \frac{1}{2} \rho''(|u^*|)(|u+\delta| - |u|)^2 + o(\delta^2).
 \end{aligned} \tag{40}$$

By the condition (b) of the MMCD algorithm $\inf_t \rho''(|t|; \lambda, \gamma) > -M$ and $(|u+\delta| - |u|)^2 \geq \delta^2$.

Hence there exist $\theta_2 = \frac{1}{2}(M + \inf_x \rho''(|x|) + o(1)) > 0$, such that for the j th coordinate, $j \in \{1, \dots, p\}$,

$$\chi(u_0 + \delta) - \chi(u_0) \geq \theta_2 \delta^2. \tag{41}$$

Now consider β_0 , observe that

$$\begin{aligned}
& \chi(u+\delta) - \chi(u) \\
&= \ell(u + \delta, \beta_1, \dots, \beta_p) \\
&\quad - \ell(u, \beta_1, \dots, \beta_p) \\
&= \nabla_1 \ell(u, \beta_1, \dots, \beta_p) \delta \\
&\quad + \frac{1}{2} \nabla_1^2 \ell(u, \beta_1, \dots, \beta_p) \delta^2 \\
&\quad + o(\delta^2) \\
&= \nabla_1 \ell(u, \beta_1, \dots, \beta_p) \delta \\
&\quad + (\nabla_1^2 \ell(u, \beta_1, \dots, \beta_p) \\
&\quad \quad - M) \delta^2 \\
&\quad + (M - \frac{1}{2} \nabla_1^2 \ell(u, \beta_1, \dots, \beta_p)) \delta^2 + o(\delta^2),
\end{aligned} \tag{42}$$

By similar arguments to (38), we have

$$0 = \nabla_1 \ell(u_0 + \delta, \beta_1, \dots, \beta_p) + M(u_0 + \delta - u_0) = \nabla_1 \ell(u_0, \beta_1, \dots, \beta_p) + \nabla_1^2 \ell(u_0, \beta_1, \dots, \beta_p) \delta + o(\delta) - M\delta. \tag{43}$$

Therefore, by (42, 43), for the first coordinate of β

$$\begin{aligned}
& \chi(u_0 + \delta) - \chi(u_0) \\
&= (M - \frac{1}{2} \nabla_1^2 \ell(u_0, \beta_1, \dots, \beta_p)) \delta^2 \\
&\quad + o(\delta^2) = \frac{1}{2} M \delta^2 + \frac{1}{2} (M \\
&\quad \quad - \nabla_1^2 \ell(u_0, \beta_1, \dots, \beta_p)) \delta^2 \\
&\quad + o(\delta^2) \geq \frac{1}{2} \delta^2 (M + o(1)).
\end{aligned} \tag{44}$$

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

$$\chi(u_0 + \delta) - \chi(u_0) \geq \theta_1 \delta^2. \tag{45}$$

Let $\theta = \min(\theta_1, \theta_2)$, using (41, 45), we have for all the coordinates of β ,

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

By (46) we have

$$Q(\beta_j^{m-1}) - Q(\beta_{j+1}^{m-1}) \geq \theta (\beta_{j+1}^m - \beta_j^{m-1})^2 = \theta \|\beta_j^{m-1} - \beta_{j+1}^{m-1}\|_2^2, \tag{47}$$

where $\beta_j^{m-1} = (\beta_1^m, \dots, \beta_j^m, \beta_{j+1}^{m-1}, \dots, \beta_p^{m-1})$. The (47) establishes the boundedness of the sequence $\{\beta^m\}$ for every $m > 1$ since the starting point of $\{\beta^1\} \in \mathbb{R}^{p+1}$.

Applying (47) over all the coordinates, we have for all m

$$Q(\beta^m) - Q(\beta^{m+1}) \geq \theta \|\beta^{m+1} - \beta^m\|_2^2. \quad (48)$$

Since the (decreasing) sequence $Q(\beta^m)$ converges, (48) shows that the sequence $\{\beta^k\}$ have a unique limit point. This completes the proof of the convergence of $\{\beta^k\}$.

The assumption (3) and (4) in Lemma 1 holds by (48). Hence, the limit point of $\{\beta^k\}$ is a minimum of $Q(\beta)$ by Lemma 1. This completes the proof of the theorem.

Fig. 1.

Plots of solution paths along κ . It shows the paths for a causal variable with $\beta = 2$. Observe that the estimates could change substantially when κ crosses certain threshold values.

Fig. 2.

Computational efficiency of the LLA, adaptive rescaling, LLA-CD and MMCD algorithms with fixed sample size ($n = 1,000$). The solid, dashed, dotted lines and dashed line with dark circles represent the average elapsed times of the MMCD, LLA-CD, adaptive rescaling and LLA, respectively. Here, IN, SP, PC, AR and CS refers to the five correlation structures of covariates described in Subsection 4.2.

Fig. 3.

Computational efficiency of the adaptive rescaling, LLA-CD and MMCD algorithms for $p \gg n$ models. The sample size is $n = 300$. The solid, dashed and dotted lines represent the average elapse times of the MMCD, LLA-CD and the adaptive rescaling, respectively. Here, IN, SP, PC, AR and CS refers to the five correlation structures of covariates described in Subsection 4.2.

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 refer to the five correlation structures of covariates described in Subsection 4.2.

Σ (SNR)	Algorithm	PAUC (SE* 10^5)	MS (SE* 10^4)	FDR (SE* 10^3)
IN (4.32)	LLA	0.947(4.58)	10.96(0.55)	0.07(3.32)
	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 (3.05)	LLA	0.915(7.74)	11.39(0.77)	0.10 (3.96)
	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 (3.89)	LLA	0.945(5.95)	14.25(1.50)	0.24(5.72)
	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 (3.20)	LLA	0.921(8.83)	13.83(1.28)	0.24(5.34)
	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 (3.06)	LLA	0.919(8.13)	12.42(1.07)	0.16(4.70)
	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 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 refer to the five correlation structures of covariates described in Subsection 4.2.

Σ (SNR)	Algorithm	PAUC (SE* 10^5)	MS (SE* 10^4)	FDR (SE* 10^3)
IN (4.33)	Adp res	0.828(1.30)	12.25(3.01)	0.60(6.95)
	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 (3.05)	Adp res	0.778(1.96)	12.06(3.77)	0.62(6.05)
	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 (3.87)	Adp res	0.872(0.64)	7.12(1.37)	0.42(6.43)
	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 (3.19)	Adp res	0.812(1.21)	6.21(1.69)	0.49(8.53)
	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 (3.04)	Adp res	0.770(1.79)	11.89(3.58)	0.64(6.32)
	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 the 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)