Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

A Concave Conjugate Approach for Nonconvex Penalized Regression with the MCP Penalty

Shubao Zhang and **Hui Qian*** and **Wei Chen** and **Zhihua Zhang**
College of Computer Science and Technology
Zhejiang University, Hangzhou 310027, China
{bravemind, qianhui, chenwei, zhzhang}@zju.edu.cn

Abstract

The minimax concave plus penalty (MCP) has been demonstrated to be effective in nonconvex penalization for feature selection. In this paper we propose a novel construction approach for MCP. In particular, we show that MCP can be derived from a concave conjugate of the Euclidean distance function. This construction approach in turn leads us to an augmented Lagrange multiplier method for solving the penalized regression problem with MCP. In our method each tuning parameter corresponds to a feature, and these tuning parameters can be automatically updated. We also develop a d.c. (difference of convex functions) programming approach for the penalized regression problem. We find that the augmented Lagrange multiplier method degenerates into the d.c. programming method under specific conditions. Experimental analysis is conducted on a set of simulated data. The result is encouraging.

1 Introduction

Learning or mining from high dimensional data is an important issue (Hastie, Tibshirani, and Friedman 2009). In this paper we are especially concerned with variable selection problems. Generally, the collinearity among the variables implies that the underlying model lies on an intrinsic low-dimensional subspace. Thus, it is interesting and challenging to find a sparse representation for high dimensional data.

To achieve sparsity, penalization methods have been widely used in the literature. A principled approach is the lasso of Tibshirani (1996), which employs the ℓ_1-norm penalty and does variable selection via the soft threshold operator. However, Fan and Li (2001) pointed out that the lasso shrinkage method produces biased estimates for the large coefficients. Zou (2006) argued that the lasso might not be an oracle procedure in certain scenarios.

According to the criteria proposed by Fan and Li (2001), for a good penalty function, the resulting estimator should enjoy sparsity, continuity and unbiasedness. Moreover, they showed that a penalty function satisfying certain conditions enjoys oracle properties. This leads to recent developments of nonconvex penalization in sparse learning. There exist many nonconvex penalties, including the ℓ_q penalty (with $0 < q < 1$), the Smoothly Clipped Absolute Deviation (SCAD) (Fan and Li 2001), the Log penalty (Mazumder, Friedman, and Hastie 2011), the minimax concave plus penalty (MCP) (Zhang 2010a), the nonconvex exponential-type penalty (EXP) (Gao et al. 2011), etc. These penalties are demonstrated to have attractive theoretical properties and wide practical applications. For example, MCP has been successfully applied to biological data analysis (Breheny and Huang 2011) and computer vision (Shi et al. 2011).

Compared with convex penalties such as the ℓ_1 norm penalty, nonconvex penalties generally suffer from computational challenges because of their nondifferentiability and nonconvexity. A variety of methods have been proposed to deal with this challenge. Fan and Li (2001) proposed a local quadratic approximation (LQA) for the SCAD penalty. In more general cases, Zou and Li (2008) studied a local linear approximation (LLA). These methods share a majorization-minimization (MM) idea (Hunter and Li 2005). In the same spirit of LQA and LLA, the iterative reweighted ℓ_2 and ℓ_1 methods have also been developed to find sparse solutions (Chartrand and Yin 2008; Candès, Wakin, and Boyd 2008; Daubechies et al. 2010; Wipf and Nagarajan 2010). Additionally, Gasso, Rakotomamonjy, and Canu (2009) developed an iterative algorithm for nonconvex penalties based on the d.c. programming method (An and Tao 2001). Recently, Mazumder, Friedman, and Hastie (2011) developed a so-called SparseNet algorithm based on the coordinate descent method (Friedman et al. 2007).

In this paper we focus on the MCP penalty due to its success theoretically and practically. Our work is motivated by Zhang (2010b) and Zhang and Tu (2012). Particularly, Zhang and Tu (2012) showed that the nonconvex LOG and EXP penalties can be derived from concave conjugate of the Kullback-Leibler (KL) distance function. We find that MCP can be defined as a concave conjugate of the Euclidean distance function. This finding encourages us to devise an augmented Lagrange multiplier (ALM) method for solving the corresponding penalized regression problem. The ALM method enjoys the idea of proximal minimization (Censor and Zenios 1997). Moreover, it has the same convergence properties as the expectation maximization (EM) algorithm (Dempster, Laird, and Rubin 1977).

*Corresponding author.

It is worth pointing out that there are multiple tuning (or regularization) hyperparameters in our approach; namely, each tuning parameter corresponds to a feature. This makes our approach have stronger ability in sparse modeling than the conventional setting (Mazumder, Friedman, and Hastie 2011). Moreover, the ALM algorithm can automatically update these tuning hyperparameters. For comparison, we also devise a d.c. (difference of convex functions) programming method for the MCP-based penalized regression problem. When we set the tuning hyperparameters as an identical one and prespecify it, our approach shares the same idea as the d.c. programming method. Compared with the SparseNet (Mazumder, Friedman, and Hastie 2011) which uses a two-dimensional grid search for selecting two hyperparameters, our approach only needs to select one hyperparameter via grid search. Thus, our approach is more efficient.

The rest of the paper is organized as follows. First, we give an overview of the problem, and introduce the MCP penalty and the notion of concave conjugate. Then we expound our new idea over this problem and describe our work. In the following section, we conduct an empirical analysis. Finally, we conclude our work.

2 Problem Formulation

Consider the linear regression problem

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T \in \mathbb{R}^{n\times p}$ is an input matrix, $\boldsymbol{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ is the corresponding output vector, $\boldsymbol{\beta} \in \mathbb{R}^p$ is a vector of regression coefficients, and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a Gaussian error vector. We assume that $\boldsymbol{X}$ and $\boldsymbol{y}$ are standardized so that $\sum_{i=1}^n y_i = 0$, $\sum_{i=1}^n x_{ij} = 0$ and $\sum_{i=1}^n x_{ij}^2 = 1$ for all $j = 1, \ldots, p$. Our aim is to find a sparse estimate of the regression vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ under the regularization framework.

The classical regularization framework is based on a penalty function of $\boldsymbol{\beta}$. That is

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + P(\boldsymbol{\beta}; \lambda),$$

where $P(\boldsymbol{\beta}; \lambda) = \sum_{j=1}^p p(\beta_j; \lambda_j)$ is the penalty that achieves sparsity in the coefficients, and λ_j is the tuning (or regularization) parameter controlling the tradeoff between the loss function and the penalty.

A widely used setting for penalty is $P(\boldsymbol{\beta}; \lambda) = \lambda \sum_{j=1}^p p_j(\beta_j)$, which implies that the penalty function consists of p separable subpenalties and all subpenalties share a common tuning parameter λ. In order to find a sparse solution of $\boldsymbol{\beta}$, one imposes the ℓ_0-norm penalty $\|\boldsymbol{\beta}\|_0$ to $\boldsymbol{\beta}$ (i.e., the number of nonzero elements of $\boldsymbol{\beta}$). However, the resulting optimization problem is NP-hard. The ℓ_1-norm penalty $P(\boldsymbol{\beta}; \lambda) = \lambda\|\boldsymbol{\beta}\|_1 = \lambda \sum_{j=1}^p |\beta_j|$ is an effective convex alternative for the nonconvex ℓ_0-norm penalty.

Recently, some nonconvex alternatives, such as the ℓ_q penalty ($q \in (0, 1)$), log-penalty, SCAD and MCP, have been employed. Meanwhile, iteratively reweighted ℓ_q ($q = 1$ or 2) minimization or coordinate descent methods were developed for finding sparse solutions. Since MCP has good theoretical properties and successful applications, we are mainly concerned with the use of MCP in the penalized regression problem in this paper.

2.1 The Minimax Concave Plus Penalty

The MCP function is defined as

$$\begin{aligned} p_\gamma(t; \lambda) &= \lambda \int_0^{|t|} \left(1 - \frac{x}{\gamma\lambda}\right)_+ dx \\ &= (\lambda|t| - \frac{t^2}{2\gamma})\boldsymbol{I}(|t| < \lambda\gamma) + \frac{\lambda^2\gamma}{2}\boldsymbol{I}(|t| \geq \lambda\gamma) \end{aligned} \tag{1}$$

for $\lambda > 0$ and $\gamma > 0$, where $\boldsymbol{I}$ is the indicator function and $(z)_+ = \max(z, 0)$. With the MCP function in (1), the penalized regression problem is

$$\min_{\boldsymbol{\beta}} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + P_\gamma(\boldsymbol{\beta}; \lambda) \tag{2}$$

where $P_\gamma(\boldsymbol{\beta}; \lambda) = \sum_{j=1}^p p_\gamma(\beta_j; \lambda)$.

The MCP function and its first-order derivative are depicted in Figures 1 and 2 respectively. Recently, Mazumder, Friedman, and Hastie (2011) showed that for each value of $\lambda > 0$ there is a continuum of penalties and threshold operators as γ varying from ∞ to $1+$, which corresponds to the soft threshold operator and the hard threshold operator respectively. Therefore, the MCP at $\gamma \to \infty$ and $\gamma \to 1+$ performs like the ℓ_1 penalty and the ℓ_0 penalty respectively. Zhang (2010a) proved that the MCP function can result in a nearly unbiased estimate. In this paper we introduce the notion of concave conjugate to further explore MCP.

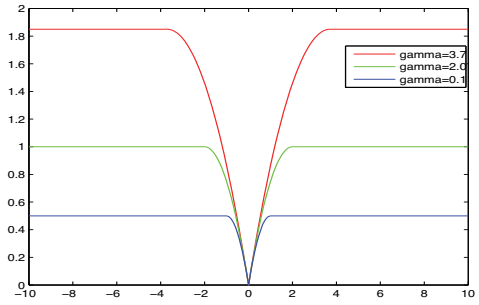


Figure 1: The MCP function for different values of γ.

2.2 Concave Conjugate

Given a function $f : S \subseteq \mathbb{R}^p \to (-\infty, \infty)$, its concave conjugate, denoted by g, is defined by

$$g(\mathbf{v}) = \inf_{\mathbf{u}\in S} \{\mathbf{u}^T\mathbf{v} - f(\mathbf{u})\}.$$

It is well known that g is concave whether or not f is concave. However, if f is proper, closed and concave, the concave conjugate of g is again f (Boyd and Vandenberghe 2004).

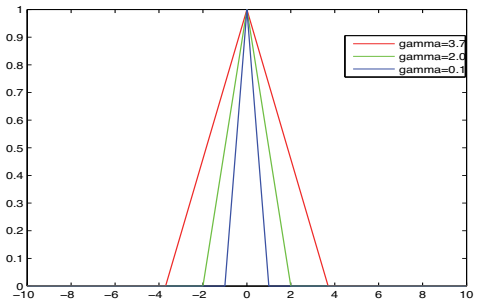


Figure 2: The first-order derivative of MCP for different values of γ.

We note that Wipf and Nagarajan (2010) used the idea of concave conjugate to express the automatic relevance determination (ARD) cost function, and Zhang (2010b) derived the bridge penalty by using the idea of concave conjugate. Recently, Zhang and Tu (2012) derived the LOG and EXP penalties from concave conjugate of the Kullback-Leibler (KL) divergence. We will see that the MCP function can also be derived out under the framework of concave conjugate.

3 Methodology

In this section we first give a new construction approach for the MCP function based on concave conjugate. Using this construction approach, we then develop an augmented Lagrange multiplier method for solving the corresponding penalized regression problem.

Given two non-negative vectors $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_p)^T$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_p)^T$, we consider the following concave conjugate problem:

$$\min_{\boldsymbol{\omega} \geq 0} \boldsymbol{\omega}^T |\boldsymbol{\beta}| + \frac{\gamma}{2} \|\boldsymbol{\omega} - \boldsymbol{\lambda}\|_2^2, \quad \gamma > 0 \tag{3}$$

w.r.t. $|\boldsymbol{\beta}| = (|\beta_1|, \ldots, |\beta_p|)^T$. We have the following theorem.

Theorem 1 *Let $P_\gamma(\boldsymbol{\beta}; \boldsymbol{\lambda})$ denote the minimum of the problem in (3). Then $P_\gamma(\boldsymbol{\beta}; \boldsymbol{\lambda}) = \sum_{j=1}^p \lambda_j \Psi_\gamma(|\beta_j|)$ where*

$$\Psi_\gamma(|\beta_j|) = \begin{cases} \frac{1}{2}\lambda_j \gamma & \text{if } |\beta_j| \geq \lambda_j \gamma, \\ |\beta_j| - \frac{\beta_j^2}{2\lambda_j \gamma} & \text{otherwise}. \end{cases}$$

Moreover, the minimum value is attained when

$$\omega_j = \begin{cases} 0 & \text{if } |\beta_j| > \lambda_j \gamma, \\ \lambda_j - \frac{1}{\gamma}|\beta_j| & \text{otherwise}. \end{cases}$$

The proof of this theorem is easily obtained. Since the Euclidean distance $\|\boldsymbol{\omega} - \boldsymbol{\lambda}\|_2^2$ is strictly convex in $\boldsymbol{\omega}$, $P_\gamma(\boldsymbol{\beta}; \boldsymbol{\lambda})$ can be viewed as the concave conjugate of $-\gamma/2\|\boldsymbol{\omega} - \boldsymbol{\lambda}\|_2^2$. Interestingly, if letting $\lambda_1 = \cdots = \lambda_p \triangleq \lambda > 0$, we see that $P_\gamma(\boldsymbol{\beta}; \boldsymbol{\lambda})$ becomes the conventional MCP. Thus, we derive out MCP under the framework of concave conjugate.

As we know, the existing methods such as SparseNet directly solve the problem (2) with setting of $\lambda_1 = \cdots = \lambda_p \triangleq \lambda > 0$. Moreover, λ as well as γ are usually selected via the cross validation or grid search. In this paper we consider the case that the λ_j's are not necessary identical and have the following regularization optimization problem

$$\min_{\boldsymbol{\beta}, \boldsymbol{\lambda}} \Big\{ J(\boldsymbol{\beta}, \boldsymbol{\lambda}) \triangleq \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^p \lambda_j \Psi_\gamma(|\beta_j|) \Big\}. \tag{4}$$

According to the construction of MCP based on concave conjugate, we consider an equivalent alternative as follows

$$\min_{\boldsymbol{\beta}, \boldsymbol{\lambda}} \Big\{ \min_{\boldsymbol{\omega} \geq 0} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \boldsymbol{\omega}^T |\boldsymbol{\beta}| + \frac{\gamma}{2}\|\boldsymbol{\omega} - \boldsymbol{\lambda}\|_2^2 \Big\}. \tag{5}$$

3.1 The Augmented Lagrange Multiplier Method

In this section, we deal with the problem (5) in which $\boldsymbol{\lambda}$ is also a vector that needs to be estimated. W resort to an augmented Lagrange multiplier (ALM) algorithm (Censor and Zenios 1997) to solve the problem.

In particular, we are given initial values $\boldsymbol{\omega}^{(0)}$, e.g., $\boldsymbol{\omega}^{(0)} = \varepsilon(1, \ldots, 1)^T$ where $\varepsilon > 0$ is prespecified. After the kth estimates $(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ of $(\boldsymbol{\beta}, \boldsymbol{\lambda})$ are obtained, the $(k+1)$st iteration of the ALM algorithm consists of two steps. The first step calculates $\boldsymbol{\omega}^{(k)}$ via

$$\boldsymbol{\omega}^{(k)} = \operatorname*{argmin}_{\boldsymbol{\omega} > 0} \; F(\boldsymbol{\omega} | \boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}),$$

where $F(\boldsymbol{\omega} | \boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ is given as

$$F(\boldsymbol{\omega} | \boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}) \triangleq \sum_{j=1}^p \omega_j |\beta_j^{(k)}| + \frac{\gamma}{2}\|\boldsymbol{\omega} - \boldsymbol{\lambda}^{(k)}\|_2^2.$$

The second step then calculates $\boldsymbol{\beta}^{(k+1)}$ and $\boldsymbol{\lambda}^{(k+1)}$ via

$$(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)})$$
$$= \operatorname*{argmin}_{\boldsymbol{\beta}, \boldsymbol{\lambda}} \Big\{ \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^p \omega_j^{(k)} |\beta_j| + \frac{\gamma}{2}\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\lambda}\|_2^2 \Big\}.$$

Note that given $\boldsymbol{\omega}^{(k)}$, $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$ are independent. Thus the second step can be partitioned into two parts; namely,

$$\boldsymbol{\lambda}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\lambda}} \; \frac{\gamma}{2}\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\lambda}\|_2^2$$

and

$$\boldsymbol{\beta}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{\beta}} \Big\{ \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^p \omega_j^{(k)} |\beta_j| \Big\}.$$

The former part shows that the ALM enjoys the idea behind the proximal minimization (Censor and Zenios 1997). The latter part is a reweighted ℓ_1 minimization problem which can be solved by the coordinate descent method. We summary the implementation of ALM in Algorithm 1.

Unlike the SparseNet algorithm which uses a two-dimensional grid search for selecting the hyperparameters γ and λ, the ALM method only needs to select γ. Thus

Algorithm 1 The ALM Algorithm

Input: $\{\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T, \mathbf{y}\}$, γ, $\boldsymbol{\omega}^{(0)}$
while $\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\|_2 < \tau$ **do**
Update

$$\boldsymbol{\omega}^{(k)} = \underset{\boldsymbol{\omega} \geq 0}{\operatorname{argmin}} \Big\{ \sum_{j=1}^{p} \omega_j |\beta_j^{(k)}| + \frac{\gamma}{2} \|\boldsymbol{\omega} - \boldsymbol{\lambda}^{(k)}\|_2^2 \Big\},$$

and the update rule is the following:

$$\omega_j^{(k)} = \begin{cases} 0 & \text{if } |\beta_j^{(k)}| > \lambda_j^{(k)} \gamma, \\ \lambda_j^{(k)} - \frac{1}{\gamma} |\beta_j^{(k)}| & \text{otherwise.} \end{cases}$$

Update

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\omega}^{(k)},$$

$$\boldsymbol{\beta}^{(k+1)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} \omega_j^{(k)} |\beta_j| \Big\}.$$

end while
Output: $\boldsymbol{\beta}$

our ALM method is more efficient than SparseNet. Also the ALM method bears an interesting resemblance to the EM algorithm, because we can treat $\boldsymbol{\omega}$ as missing data. With such a treatment, the first step of ALM is related to the E-step of EM, which calculates the expectations associated with missing data.

3.2 Convergence Analysis

We now investigate the convergence property of the ALM algorithm. Our analysis is based on the optimization problem in (4) with the objective function $J(\boldsymbol{\beta}, \boldsymbol{\lambda})$. Noting that $\boldsymbol{\omega}^{(k)}$ is a function of $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\lambda}^{(k)}$, we denote the objective function in the second step of ALM by

$$Q(\boldsymbol{\beta}, \boldsymbol{\lambda} | \boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} \omega_j^{(k)} |\beta_j| + \frac{\gamma}{2} \|\boldsymbol{\omega}^{(k)} - \boldsymbol{\lambda}\|_2^2.$$

We have the following lemma.

Lemma 1 *Let $\{(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}) : k = 1, 2, \ldots\}$ be a sequence defined by the ALM algorithm. Then,*

$$J(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}) \leq J(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}),$$

with equality if and only if $\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)}$.

Since $J(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}) \geq 0$, this lemma shows that $J(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ converges monotonically to some $J^* \geq 0$. In fact, the ALM algorithm enjoys the same convergence as the standard EM algorithm (Dempster, Laird, and Rubin 1977).

Let $\mathcal{A}(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ be the set of values of $(\boldsymbol{\beta}, \boldsymbol{\lambda})$ that minimize $Q(\boldsymbol{\beta}, \boldsymbol{\lambda} | \boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ over $\Omega \subset \mathbb{R}^p \times \mathbb{R}_{++}^p$ and $\mathcal{S}$ be the set of stationary points of Q in the interior of Ω. We can immediately derive the following theorem from Zangwill's *global convergence theorem* or the literature (Wu 1983; Sriperumbudur and Lanckriet 2009).

Theorem 2 *Let $\{\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}\}$ be a sequence of the ALM algorithm generated by $(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}) \in \mathcal{A}(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$. Suppose that* (i) *$\mathcal{A}(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ is closed over the complement of $\mathcal{S}$ and* (ii)

$$J(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}) < J(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}) \text{ for all } (\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}) \notin \mathcal{S}.$$

Then all the limit points of $\{\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)}\}$ are stationary points of $J(\boldsymbol{\beta}, \boldsymbol{\lambda})$ and $J(\boldsymbol{\beta}^{(k)}, \boldsymbol{\lambda}^{(k)})$ converges monotonically to $J(\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*)$ for some stationary point $(\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*)$.

4 The DC Programming Method

For comparison, in this section, we devise a d.c. (DC) programming method for the penalized regression problem with the MCP penalty, which follows the method of Gasso, Rakotomamonjy, and Canu (2009). The key idea is to consider a proper decomposition of the objective function, converting the nonconvex penalized regression problem into a convex reweighted ℓ_1 minimization problem. Particularly, we decompose the penalty $P_\gamma(\boldsymbol{\beta}; \lambda)$ as

$$P_\gamma(\boldsymbol{\beta}; \lambda) = \lambda \|\boldsymbol{\beta}\|_1 - H(\boldsymbol{\beta})$$

for some $H(\boldsymbol{\beta})$. Accordingly, we decompose the original optimization problem in (2) as

$$\min_{\boldsymbol{\beta}} \{G(\boldsymbol{\beta}) - H(\boldsymbol{\beta})\}$$

where $G(\boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1$. This is the primal problem. The corresponding dual problem is

$$\min_{\boldsymbol{\alpha}} \{H^*(\boldsymbol{\alpha}) - G^*(\boldsymbol{\alpha})\}$$

where H^* and G^* represent the corresponding Fenchel conjugates (Borwein and Lewis 2000).

With a local linear approximation, the above primal-dual pair problem evolves into the following equivalent problem:

$$\partial G^*(\boldsymbol{\alpha}') = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \{G(\boldsymbol{\beta}) - \langle \boldsymbol{\beta}, \boldsymbol{\alpha}' \rangle\}, \tag{6}$$

$$\partial H(\boldsymbol{\beta}') = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \{H^*(\boldsymbol{\alpha}) - \langle \boldsymbol{\alpha}, \boldsymbol{\beta}' \rangle\}, \tag{7}$$

where $\boldsymbol{\alpha}' \in \partial H(\boldsymbol{\beta}^{(k)})$ and $\boldsymbol{\beta}' \in \partial G^*(\boldsymbol{\alpha}^{(k)})$. If $H(\boldsymbol{\beta})$ is differentiable, the subdifferential in (7) is a singleton so that $\partial H(\boldsymbol{\beta}') = \{\nabla H(\boldsymbol{\beta})\}$. For MCP, it is just the first-order derivative. Then we obtain a reweighted ℓ_1 minimization problem

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \{\frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 - \langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle\}$$

where $\boldsymbol{\alpha} = \nabla H(\boldsymbol{\beta})$. And this problem can be efficiently solved via the coordinate descent method.

Algorithm 2 The DC Algorithm

Input: $\{\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T, \boldsymbol{y}\}$, γ, λ
Initialization: $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ and $\boldsymbol{\beta}^{(0)} = \mathbf{0}$.
while $||\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}||_2 < \tau$ **do**
Update

$$\alpha_j = \begin{cases} \beta_j/\gamma & \text{if } |\beta_j| < \lambda\gamma, \\ \lambda * \text{sign}(\beta_j) & \text{otherwise,} \end{cases}$$

where α is the derivative of the MCP penalty.
Update

$$\omega_j = \lambda - \alpha_j * \text{sign}(\beta_j).$$

Update

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}}{\text{argmin}}\{\frac{1}{2}||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \sum_{j=1}^{p}\omega_j|\beta_j|\}.$$

end while
Output: $\boldsymbol{\beta}$

Based on the above analysis, we develop the DC algorithm for the penalized regression. The algorithm constructs two sequences $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\boldsymbol{\beta}^{(k)}\}$ and stops until convergence. Essentially, this DC method is the same to the LLA method, as both of them involve a local linear approximation. The procedure is presented in Algorithm 2.

Interestingly, Eqns. (6) and (7) suggest that the DC method is in fact derived from the concave conjugate theory. Indeed, the ALM algorithm degenerates to the DC algorithm when we set $\boldsymbol{\lambda} = \lambda(1, \ldots, 1)^T$ and specify λ. However, the advantage of the ALM method over the DC method is in that the ALM method allows each tuning parameter corresponding to a feature. Moreover, these tuning parameters can be automatically updated during the ALM iteration.

5 Experimental Analysis

In this section we conduct experimental analysis of the ALM and DC algorithms. For comparison, we use two SparseNet algorithms: SparseNet_R and SparseNet_M. The former is from the standard R package of SparseNet. And the latter is implemented using Matlab without the *df* calibration technique that suggested in (Mazumder, Friedman, and Hastie 2011). We use cross validation to select the hyperparameters λ and γ in the DC algorithm. For the ALM algorithm, we set $\boldsymbol{\omega}^{(0)} = \varepsilon\mathbf{1}^T$, in which ε is chosen via cross validation too. For τ, the threshold of termination criterion, we set it as 10^{-4} in our numerical experiments.

We take experiments on a set of simulated datasets. To generate the datasets, we employ the following data model. That is,

$$y = \boldsymbol{x}^T\boldsymbol{\beta} + \sigma\epsilon,$$

where $\epsilon \sim N(0, 1)$. The coefficient vector $\boldsymbol{\beta}$ is a 200-dimensional vector with 10 non-zero elements which are determined by $\beta_i = \beta_{i+100}$, $i = 1, \ldots, 5$. We sample the observation data $\boldsymbol{x}$ from a multivariate Gaussian distribution which has a zero mean and a covariance matrix $\Sigma = \{0.7^{|i-j|}\}_{1\le i,j\le 200}$. We choose a σ satisfying

$$SNR = \frac{\sqrt{\boldsymbol{\beta}^T\Sigma\boldsymbol{\beta}}}{\sigma},$$

in which SNR is the signal-to-noise ratio. Numerical experiments are carried out with different n and SNR values. In particular, we also take an experiment in the case with a high signal-to-noise ratio. For each test, $m = 1000$ testing instances are generated. To assess the performance of the algorithms, we use the Standardized Prediction Error (SPE) and Feature Selection Error (FSE) as the assessment criterion. The SPE is defined as

$$SPE = \frac{\sum_{i=1}^{m}(y_i - \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}})^2}{m\sigma^2}.$$

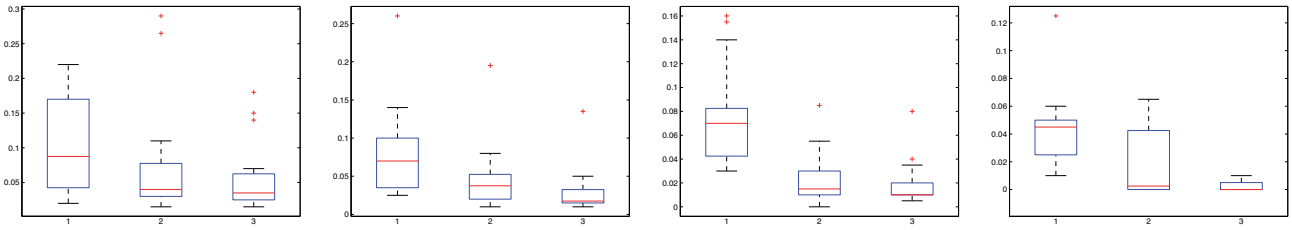
And the FSE is the proportion of coefficients in $\hat{\boldsymbol{\beta}}$ which is not correctly set to zero or non-zero based on the true $\boldsymbol{\beta}$.

We set up our experimental program as follows. Four sets of data are generated with $SNR = \{3, 10\}$ and $n = \{50, 100\}$, respectively. We repeat all the algorithms with 20 times for each dataset. The reported results are based on these 20 repeats.

Table 1: Simulated results of SparseNet, DC and ALM.

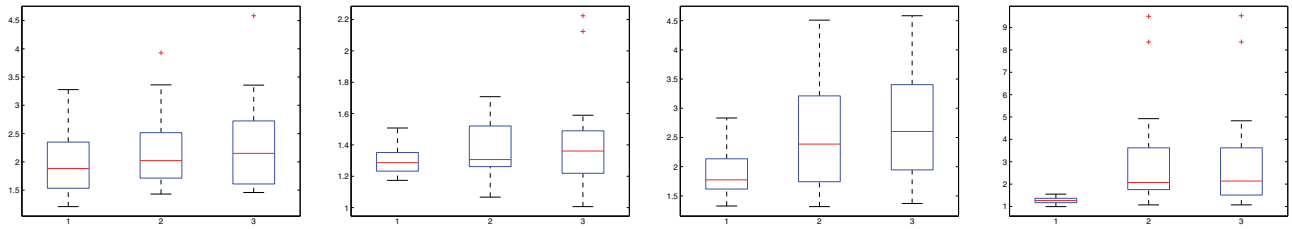
ALGORITHM	AVERAGE SPE	AVERAGE FSE
$n = 50, SNR = 3$		
SPARSENET_M	4.4289(±1.1320)	0.1946(±0.0144)
SPARSENET_R	**1.9518(±0.5569)**	0.1038(±0.0685)
DC	2.2003 (± 0.6880)	0.0703(± 0.0762)
ALM	2.2780(± 0.8000)	**0.0533(± 0.0480)**
$n = 50, SNR = 10$		
SPARSENET_M	7.7827(± 3.788)	0.1468(± 0.0192)
SPARSENET_R	**1.8839(±0.4152)**	0.0755(± 0.0435)
DC	2.5743(± 0.9746)	0.0238(± 0.0204)
ALM	2.7288(± 0.9006)	**0.0178(± 0.0174)**
$n = 100, SNR = 3$		
SPARSENET_M	4.6127(± 0.8042)	0.4007(± 0.0281)
SPARSENET_R	**1.2996(±0.0899)**	0.0755(±0.0552)
DC	1.3883(± 0.1849)	0.0455 (± 0.0400)
ALM	1.4140(± 0.3010)	**0.0270(± 0.0277)**
$n = 100, SNR = 10$		
SPARSENET_M	6.1749(± 2.6990)	0.3172(± 0.0205)
SPARSENET_R	**1.2720 (±0.1432)**	0.0448(± 0.0313)
DC	3.0134(± 2.2871)	0.0175(± 0.0238)
ALM	2.9553(± 2.3128)	**0.0022(± 0.0035)**

Table 1 reports the prediction mean and the corresponding standard deviation over 20 runs. And Figures 3 and 4 exhibit the average result over 20 runs with the box-and-whisker plot. The results show that ALM works better than the other three methods in feature selection accuracy. This strongly demonstrates the merit of ALM. Since in ALM each tuning parameter corresponds to a feature and the tuning parameter is automatically updated, ALM is more capable of selecting the correct features. As for prediction accuracy, SparseNet_R performs the best because of its adoption of calibration technique. When no calibration technique is used for further performance improvement, ALM and DC perform better than



$n = 50$ and $SNR = 3$ $n = 100$ and $SNR = 3$ $n = 50$ and $SNR = 10$ $n = 100$ and $SNR = 10$

Figure 3: Feature Selection Error. Here "1," "2," and "3" are for SparseNet_R, DC and ALM respectively.



$n = 50$ and $SNR = 3$ $n = 100$ and $SNR = 3$ $n = 50$ and $SNR = 10$ $n = 100$ and $SNR = 10$

Figure 4: Prediction Errors. Here "1," "2," and "3" are for SparseNet_R, DC and ALM respectively.

SparseNet_M in both prediction accuracy and feature selection accuracy. In addition, we can see that DC and ALM perform almost equally with respect to the prediction accuracy. However, ALM achieves a lower feature selection error. It just demonstrates ALM's strong ability in feature selection. We also find that ALM has a better convergence property than DC in the practical experiments. Usually, ALM takes less iterations than DC to converge.

6 Conclusion

In this paper we have studied the penalized regression problem with the MCP penalty. In particular, we have proposed a new construction of MCP based on concave conjugate of the Euclidean distance function. Accordingly, we have devised an ALM method for solving the corresponding penalized regression problem. Additionally, we have revealed the relationship between the ALM method and the DC method for MCP. The ALM method has an advantage in the automatic choice of the tuning (or regularization) parameters in comparison with the DC method and the SparseNet. Numerical experiments have demonstrated computational efficiency of our ALM as well as its ability in sparse modeling.

A The Proof of Lemma 1

Consider that

$$
\begin{aligned}
&J(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\lambda}^{(k+1)}) \\
&= \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^{(k+1)}\|_2^2 + P_\gamma(\boldsymbol{\beta}^{(k+1)}; \boldsymbol{\lambda}^{(k+1)}) \\
&= \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^{(k+1)}\|_2^2 + \min_{\boldsymbol{\omega}>0}\left\{\boldsymbol{\omega}^T\boldsymbol{\beta}^{(k+1)} + \frac{\gamma}{2}\|\boldsymbol{\omega}-\boldsymbol{\lambda}^{(k+1)}\|_2^2\right\} \\
&\le \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^{(k+1)}\|_2^2 + (\boldsymbol{\omega}^{(k)})^T\boldsymbol{\beta}^{(k+1)} + \frac{\gamma}{2}\|\boldsymbol{\omega}^{(k)}-\boldsymbol{\omega}^{(k)}\|_2^2 \\
&= \min_{\boldsymbol{\beta}}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + (\boldsymbol{\omega}^{(k)})^T\boldsymbol{\beta}\right\} \\
&\le \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^{(k)}\|_2^2 + (\boldsymbol{\omega}^{(k)})^T\boldsymbol{\beta}^{(k)} + \frac{\gamma}{2}\|\boldsymbol{\omega}^{(k)} - \boldsymbol{\omega}^{(k-1)}\|_2^2 \\
&= \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^{(k)}\|_2^2 + \min_{\boldsymbol{\omega}>0}\left\{\boldsymbol{\omega}^T\boldsymbol{\beta}^{(k)} + \frac{\gamma}{2}\|\boldsymbol{\omega}-\boldsymbol{\lambda}^{(k)}\|_2^2\right\} \\
&= \frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}^{(k)}\|_2^2 + P_\gamma(\boldsymbol{\beta}^k; \boldsymbol{\lambda}^{(k)}) \\
&= J(\boldsymbol{\beta}^k, \boldsymbol{\lambda}^{(k)}).
\end{aligned}
$$

Acknowledgments

Zhihua Zhang acknowledges support from the Natural Science Foundations of China (No. 61070239). Shubao Zhang, Wei Chen and Hui Qian acknowledge support from the Natural Science Foundations of China (No. 90820306).

References

An, L. T. H., and Tao, P. D. 2001. Dc programming approach and solution algorithm to multidimensional scaling problem. *Nonconvex optimization and its application* 231–276.

Borwein, J. M., and Lewis, A. S. 2000. *Convex analysis and nonlinear optimization*. New York: Springer.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge, UK: Cambridge University Press.

Breheny, P., and Huang, J. 2011. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics* 2:232–253.

Candès, E. J.; Wakin, M. B.; and Boyd, S. P. 2008. Enhancing sparsity by reweighted ℓ_1 minimization. *The Journal of Fourier Analysis and Applications* 14(5):877–905.

Censor, Y., and Zenios, S. A. 1997. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford, UK: Oxford University Press.

Chartrand, R., and Yin, W. 2008. Iteratively reweighted algorithms for compressive sensing. In *The 33rd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Daubechies, I.; Devore, R.; Fornasier, M.; and Güntürk, C. S. 2010. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics* 63(1):1–38.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B* 39(1):1–38.

Fan, J., and Li, R. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96(456):1348–1360.

Friedman, J. H.; Hastie, T.; Hoefling, H.; and Tibshirani, R. 2007. Pathwise coordinate optimization. *The Annals of Applied Statistics* 2(1):302–332.

Gao, C.; Wang, N.; Yu, Q.; and Zhang, Z. 2011. A feasible nonconvex relaxation approach to feature selection. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Gasso, G.; Rakotomamonjy, A.; and Canu, S. 2009. Recovering sparse signals with a certain family of non-convex penalties and dc programming. In *IEEE Transactions on Signal Proessing*, volume 57, 4686–4698.

Hastie, T. J.; Tibshirani, R. J.; and Friedman, J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, second edition.

Hunter, D., and Li, R. 2005. Variable selection using MM algorithms. *The Annals of Statistics* 33(4):1617–1642.

Mazumder, R.; Friedman, J.; and Hastie, T. 2011. SparseNet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association* 106(495):1125–1138.

Shi, J.; Ren, X.; Dai, G.; Wang, J.; and Zhang, Z. 2011. A non-convex relaxation approach to sparse dictionary learning. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sriperumbudur, B. K., and Lanckriet, G. R. G. 2009. On the convergence of the concave-convex procedure. In *Advances in Neural Information Processing Systems 22*.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* 58:267–288.

Wipf, D., and Nagarajan, S. 2010. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *IEEE Journal of Selected Topics in Signal Processing* 4(2):317–329.

Wu, C. F. J. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics* 11:95–103.

Zhang, Z., and Tu, B. 2012. Nonconvex penalization using laplace exponents and concave conjugates. In *Advances in Neural Information Processing Systems*, 611–619.

Zhang, C.-H. 2010a. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics* 38:894–942.

Zhang, T. 2010b. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research* 11:1081–1107.

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

Zou, H. 2006. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* 101(476):1418–1429.