A Feasible Nonconvex Relaxation Approach to Feature Selection

Cuixia Gao[*] **Naiyan Wang**[*] **Qi Yu** **Zhihua Zhang**
College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027 China
{cuixiagao0209, winsty, yuqi.rose, zhzhang}@gmail.com

Abstract

Variable selection problems are typically addressed under a penalized optimization framework. Nonconvex penalties such as the minimax concave plus (MCP) and smoothly clipped absolute deviation (SCAD), have been demonstrated to have the properties of sparsity practically and theoretically. In this paper we propose a new nonconvex penalty that we call *exponential-type penalty*. The exponential-type penalty is characterized by a positive parameter, which establishes a connection with the ℓ_0 and ℓ_1 penalties. We apply this new penalty to sparse supervised learning problems. To solve to resulting optimization problem, we resort to a reweighted ℓ_1 minimization method. Moreover, we devise an efficient method for the adaptive update of the tuning parameter. Our experimental results are encouraging. They show that the exponential-type penalty is competitive with MCP and SCAD.

Introduction

Feature selection plays a fundamental role in regression and classification models with applications in high-dimensional datasets. To enhance the performance of the model, we often seek a smaller subset of important features. Thus, sparsity is necessarily required in the resulting estimator. To pursue sparsity, Tibshirani (1996) proposed the novel lasso method to select features via the convex ℓ_1-norm penalty and soft shrinkage. However, Fan and Li (2001) showed that the lasso shrinkage produces biased estimates for the large coefficients, and Zou (2006) proved that the lasso might not be an oracle procedure (Fan and Li 2001).

In the same spirit of lasso, nonconvex penalties have been also studied. In particular, Fan and Li (2001) provided a deep insight into the properties that a good penalty function shares; that is, if the penalty function is singular at the origin and nonconvex, the resulting penalized estimate owes the properties of sparsity, continuity and unbiasedness. Moreover, the estimator with the nonconvex penalty performs as well as the oracle procedure when the tuning parameter is appropriately chosen.

A number of nonconvex penalty functions have been proposed in the literature. These functions, including the log-penalty (Mazumder, Friedman, and Hastie 2009), the minimax concave plus (MCP) (Zhang 2010a) and the smoothly clipped absolute deviation (SCAD) (Fan and Li 2001), have been demonstrated to have attractive theoretical properties and practical applications. However, they would yield computational challenges due to their non-differentiability and non-convexity.

In order to address this computational challenge, Fan and Li (2001) proposed a local quadratic approximation (LQA), while Zou and Li (2008) then devised a local linear approximation (LLA). In fact, the LLA method can be regraded as an iteratively reweighted ℓ_1 minimization method (Candès, Wakin, and Boyd 2008; Wipf and Nagarajan 2010). Moreover, all these methods can be unified under a majorization-minimization (MM) framework (Lange, Hunter, and Yang 2000). Recently, Mazumder, Friedman, and Hastie (2009) showed that a coordinate descent algorithm (Friedman et al. 2007) can be used for solving nonconvex penalized problems; also see Breheny and Huang (2010).

In this paper we further investigate nonconvex penalties for sparse supervised learning problems. In particular, we propose a new nonconvex penalty function that we refer as the *exponential-type penalty* (ETP). ETP bridges the ℓ_0 and ℓ_1 penalties via a positive parameter. More specifically, the limits of ETP are the ℓ_0 and ℓ_1 penalties when this parameter approaches ∞ and 0 respectively.

We apply ETP to sparse supervised learning problems. We explore a penalized linear regression with our ETP. We can also consider extensions involving other exponential family models; in particular we exemplify such an extension by discussing logistic regression for binary classification problems.

To obtain the resulting estimator, we resort to the iterative reweighted ℓ_1 minimization method. This method consists two steps. The first step transforms the original optimization as a weighted lasso problem, and the second step solves this new problem via some existing methods for the conventional lasso, such as the LARS (Efron et al. 2004) and the coordinate descent method. We note that applying the coordinate descent method to our case yields a so-called conditional MM algorithm.

In this paper we also devise an efficient approach for the

[*] Joint first authors; i.e., Gao and Wang contributed equally to this work.

automatical choice of the tuning parameter. It is well known that the performance of the existing nonconvex penalized supervised learning methods heavily relies on the value of the tuning parameter. The common methods for the tuning parameter selection use grid-search or gradient-based algorithms. However, these algorithms usually take large computational costs. Contrarily, the principal appeal of our approach is its simplicity and efficiency.

The rest of the paper is organized as follows. In the next section, we give a brief overview of existing nonconvex penalty terms and the reweighted ℓ_1 minimization method. A new nonconvex penalty function and a nonconvex penalized linear regression model are then presented, followed by an extension in the penalized logistic regression and by some experimental results on different data sets. The last section concludes this paper.

Problem Formulation

Suppose we are given a set of training data $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$, where the $\mathbf{x}_i \in \mathbb{R}^p$ are the input vectors and the y_i are the corresponding responses. Moreover, we assume that $\sum_{i=1}^n \mathbf{x}_i = \mathbf{0}$, $\sum_{i=1}^n y_i = 0$ and $\mathbf{x}_i^T \mathbf{x}_i = n$ for $i = 1, \ldots, p$. We now consider the following linear regression model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the $n{\times}1$ output vector, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ is the $n{\times}p$ input matrix, and $\boldsymbol{\varepsilon}$ is a Gaussian error vector. We aim to estimate the vector of regression coefficients $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ via a penalized likelihood framework; that is,

$$\max_{\boldsymbol{\beta}} \Big\{ Q(\boldsymbol{\beta}) \triangleq \sum_{i=1}^n L_i(\boldsymbol{\beta}) - n \sum_{j=1}^p P_\lambda(|\beta_j|) \Big\}, \qquad (1)$$

where L_i is the log-likelihood of y_i conditional on $\mathbf{x}_i$ and $P_\lambda(\cdot)$ is the penalty function characterized by a tuning parameter λ. In this paper we mainly consider a nonconvex penalty.

There are three popular nonconvex penalty terms: the log-penalty, MCP and SCAD, which and their first-order derivatives are listed in Table 1.

In order to solve the nonconvex penalized regression problem, Zou and Li (2008) proposed an important algorithm, which employ a local linear approximation (LLA) to the nonconvex penalty $P_{\lambda,\gamma}(|\beta_j|)$:

$$P_{\lambda,\gamma}(|\beta_j|) \approx P_{\lambda,\gamma}(|\beta_j^{(m)}|) + P'_{\lambda,\gamma}(|\beta_j^{(m)}|)(|\beta_j| - |\beta_j^{(m)}|)$$

where the $\beta_j^{(m)}$ are the mth estimates of the β_j. Their $(m{+}1)$th estimates are then calculated via

$$\boldsymbol{\beta}^{(m+1)} = \operatorname*{argmax}_{\boldsymbol{\beta}} \Big\{ \sum_{i=1}^n L_i(\boldsymbol{\beta}) - n \sum_{j=1}^p P'_{\lambda,\gamma}(|\beta_j^{(m)}|)|\beta_j| \Big\}.$$

Since the current estimator can be transformed into the conventional lasso by replacing $P'_{\lambda,\gamma}(|\beta_j^{(m)}|)|\beta_j|$ with $|\beta_j|$, we can resort to the existing methods for solving lasso such as the coordinate descent algorithm (Breheny and Huang 2010) to calculate $\beta_j^{(m+1)}$.

Recently, Zou and Li (2008) suggested using the unpenalized maximum likelihood estimate of $\boldsymbol{\beta}$ as its initial value $\boldsymbol{\beta}^{(0)}$ and then using a so-called one-step LARS estimator. Zhang (2010b) then proposed a multi-stage LLA algorithm. The LLA algorithm is in the same spirit of the iterative reweighed ℓ_1 method (Candès, Wakin, and Boyd 2008; Wipf and Nagarajan 2010). Moreover, it can be viewed as a majorization-minimization (MM) procedure (Hunter and Li 2005). With such a view, the coordinate method mentioned earlier can be then regarded as a conditional MM procedure.

Methodology

In this section we first propose a novel nonconvex penalty that we call *exponential-type penalty*. We the study its applications in sparse modeling.

The Exponential-Type Penalty

The exponential-type penalty (ETP) is defined by

$$P_{\lambda,\gamma}(|\theta|) = \frac{\lambda}{1 - \exp(-\gamma)}(1 - \exp(-\gamma|\theta|)) \qquad (2)$$

for $\lambda \geq 0$ and $\gamma > 0$. It is clear that this penalty is concave in $|\theta|$. Moreover, we can establish its relationship with the ℓ_0 and ℓ_1 penalties. In particular, we have the following propositions.

Proposition 1 *Let $P_{\lambda,\gamma}(|\theta|)$ be given in (2). Then*

(1) $\lim_{\gamma \to 0^+} P_{\lambda,\gamma}(|\theta|) = |\theta|$.

(2) $\lim_{\gamma \to +\infty} P_{\lambda,\gamma}(|\theta|) = \begin{cases} 0 & \text{if } |\theta| = 0 \\ 1 & \text{if } |\theta| \neq 0. \end{cases}$

This proposition shows that the limits of ETP at $0+$ and $+\infty$ are the ℓ_1 penalty and the ℓ_0 penalty, respectively. The first-order derivative of $P_{\lambda,\gamma}(|\theta|)$ with respect to $|\theta|$ is

$$P'_{\lambda,\gamma}(|\theta|) = \frac{\lambda\gamma}{1 - \exp(-\gamma)} \exp(-\gamma|\theta|).$$

Figures 1 and 2 depict the ETP and its derivative together with other penalties.

Sparse Learning via ETP

For the sake of presentation, we first consider the linear regression problem. An extension to logistic regression for classification will be given in the next section.

Now the penalized regression model is

$$Q(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + P_{\lambda,\gamma}(|\boldsymbol{\beta}|).$$

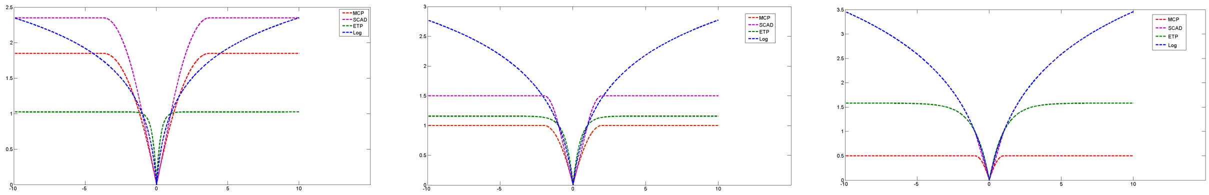
where $|\boldsymbol{\beta}| = (|\beta_1|, \ldots, |\beta_p|)^T$ and $P_{\lambda,\gamma}(|\boldsymbol{\beta}|) = \sum_{j=1}^p P_{\lambda,\gamma}(|\beta_j|)$. Here $P_{\lambda,\gamma}(|\beta_j|)$ is given in (2).

We now solve the current model by using the iteratively reweighted ℓ_1 method. Given the mth estimate $\boldsymbol{\beta}^{(m)}$ of $\boldsymbol{\beta}$, the reweighted ℓ_1 method finds its next estimate via

$$\boldsymbol{\beta}^{(m+1)} = \operatorname*{argmin}_{\boldsymbol{\beta}} \Big\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^p w_j^{(m+1)} |\beta_j| \Big\}, \qquad (3)$$

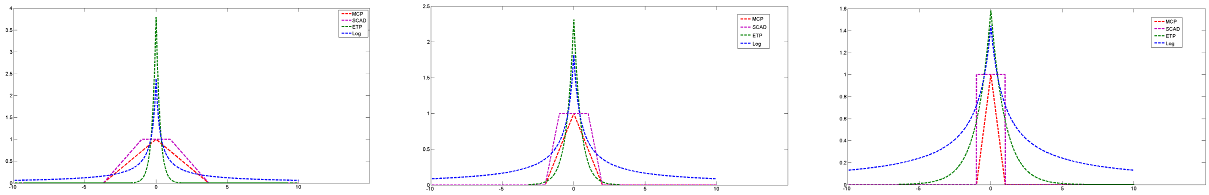
Table 1: The log-penalty, MCP and SCAD ($P_{\lambda,\gamma}(|\theta|)$) as well as their first-order derivatives ($P'_{\lambda,\gamma}(|\theta|)$).

	LOG-PENALTY ($\gamma > 0$)	SCAD ($\gamma > 2$)	MCP ($\gamma > 1$)
FUNCTIONS	$\frac{\lambda}{\log(\gamma+1)} \log(\gamma\|\theta\|+1)$	$\begin{cases} \lambda\|\theta\| & \text{IF } \|\theta\| \leq \lambda \\ \frac{\gamma\lambda\|\theta\|-0.5(\|\theta\|^2+\lambda^2)}{\gamma-1} & \text{IF } \lambda < \|\theta\| \leq \gamma\lambda \\ \frac{\lambda^2(\gamma^2-1)}{2(\gamma-1)} & \text{IF } \|\theta\| > \gamma\lambda \end{cases}$	$\begin{cases} \lambda\|\theta\| - \frac{\|\theta\|^2}{2\gamma} & \text{IF } \|\theta\| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2 & \text{IF } \|\theta\| > \gamma\lambda \end{cases}$
DERIVATIVES	$\frac{\lambda}{\log(\gamma+1)} \frac{\gamma}{\gamma\|\theta\|+1}$	$\begin{cases} \lambda & \text{IF } \|\theta\| \leq \lambda \\ \frac{\gamma\lambda-\|\theta\|}{\gamma-1} & \text{IF } \lambda < \|\theta\| \leq \gamma\lambda \\ 0 & \text{IF } \|\theta\| > \gamma\lambda \end{cases}$	$\begin{cases} \lambda - \frac{\|\theta\|}{\gamma} & \text{IF } \|\theta\| \leq \gamma\lambda \\ 0 & \text{IF } \|\theta\| > \gamma\lambda \end{cases}$



($\gamma = 3.7$) ($\gamma = 2.0$) ($\gamma = 0.1$)

Figure 1: Penalty functions: MCP, SCAD and ETP.



($\gamma = 3.7$) ($\gamma = 2.0$) ($\gamma = 0.1$)

Figure 2: The first-order derivatives of Log, MCP, SCAD and ETP

where $w_j^{(m+1)}$ is calculated via the $P'_{\lambda,\gamma}(|\beta_j^{(m)}|)$. We thus have for $j = 1, \dots, p$,

$$w_j^{(m+1)} = P'_{\lambda^{(m)},\gamma}(|\beta_j^{(m)}|) = \frac{\lambda^{(m)}\gamma\exp(-\gamma|\beta_j^{(m)}|)}{1-\exp(-\gamma)}. \quad (4)$$

where $\lambda^{(m)}$ is the mth estimate of λ. Unlike from the conventional reweighted ℓ_1 method in which the tuning parameter λ is specified by users, however, we also consider the adaptive update of λ at each iteration.

Since $w_j \geq 0$ for all j, we consider the maximization of $\sum_{j=1}^p \{w_j \log(w_j/\lambda) - w_j + \lambda\}$, which is Kullback-Leibler distance between nonnegative vectors $(w_1, \dots, w_p)$ and $(\lambda, \dots, \lambda)$. Given $\mathbf{w} = \mathbf{w}^{(m)}$, the minimizer is then

$$\lambda^{(m)} = \frac{1}{p}\sum_{j=1}^p w_j^{(m)}. \quad (5)$$

We can apply the LARS method to solve the weighted lasso problem in (3). In this case, we also employ the suggestion of Zou and Li (2008); that is, we use the one-step LARS estimation. It is worth pointing out that when applying the one-step scheme, it is not necessary to update λ via (5). However, if we use a k-step or multi-stage scheme (Zhang 2010b), such a update for λ will be very efficient.

We now devise a conditional MM algorithm for solving (3). The key idea of the algorithm is based on

$$F(\boldsymbol{\beta}) \triangleq \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^p w_j^{(m+1)}|\beta_j|$$
$$= \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{l\neq j} w_l^{(m+1)}|\beta_l| + w_j^{(m+1)}|\beta_j|.$$

We then minimize F with respect to each β_j with the remaining elements of $\boldsymbol{\beta}$ fixed. We summary the details in Algorithm 1. Here S is the soft-thresholding operator, which is defined by

$$S(z, u) = \begin{cases} z - u & \text{if } z > u \\ 0 & \text{if } |z| \leq u \\ z + \lambda & \text{if } z < -u, \end{cases}$$

for $u \geq 0$. The notation "$-j$" is referred to the portion that remains after the jth column or element is removed from a matrix or a vector in question.

It is worth noting that $w_j = 1/|\beta_j^{(m)}|^\gamma$ was set in the original iterative reweighed ℓ_1 method (Candès, Wakin, and Boyd 2008; Wipf and Nagarajan 2010). Such a setting suffers from numerical instability. If $\beta_j^{(m)} = 0$, the jth element of $\mathbf{x}$ should be removed from the iteration. Thus, it

Algorithm 1 The Conditional MM Algorithm for Sparse Learning Regression with ETP

```
Input: $\{\mathbf{X} = [\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot p}], \mathbf{y}\}$, $\gamma$, $\lambda^{(0)}$ and $\boldsymbol{\beta}^{(0)}$
for $m = 0, 1, \ldots$ do
  while not convergent do
    for $j = 1$ to $p$ do
      Calculate
          $z_j = n^{-1}\mathbf{x}_{\cdot j}^T \hat{\mathbf{r}} = n^{-1}\mathbf{x}_{\cdot j}^T \mathbf{r} + \beta_j^{(m)},$
      where $\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}^{(m)}$ and $\hat{\mathbf{r}} = \mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j}^{(m)}$.
      Update
          $\beta_j^{(m+1)} \longleftarrow S\big(z_j, w_j^{(m+1)}\big).$
      Update $\mathbf{r} \longleftarrow \mathbf{r} - (\beta_j^{(m+1)} - \beta_j^{(m)})\mathbf{x}_{\cdot j}$.
    end for
  end while
  Compute $w_j^{(m+1)}$ via (4)
  Compute $\lambda^{(m+1)}$ via (5)
end for
Output: $\boldsymbol{\beta}$
```

Algorithm 2 The Conditional MM Algorithm for ETP-based Logistic Regression

```
Input: $\{\mathbf{X} = [\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot p}], \mathbf{y}\}$, $\gamma$, $\lambda^{(0)}$, $\boldsymbol{\beta}^{(0)}$, $\varepsilon$ (tolerance).
repeat
  Calculate
      $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}^{(m)}$
      $\pi_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}, i = 1, \ldots, n$
      $\mathbf{W} = \mathsf{diag}(\pi_1(1 - \pi_1), \ldots, \pi_n(1 - \pi_n))$
      $\mathbf{r} = \mathbf{W}^{-1}(\mathbf{y} - \boldsymbol{\pi})$
      $\tilde{\mathbf{y}} = \boldsymbol{\eta} + \mathbf{r}$
  while not convergent do
    for $j = 1$ to $p$ do
      Calculate $v_j = n^{-1}\mathbf{x}_{\cdot j}^T \mathbf{W}\mathbf{x}_{\cdot j}$
      Calculate
          $z_j = \frac{1}{n}\mathbf{x}_{\cdot j}^T \mathbf{W}(\tilde{\mathbf{y}} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j})$
              $= \frac{1}{n}\mathbf{x}_{\cdot j}^T \mathbf{W}\mathbf{r} + v_j \beta_j^{(m)}$
      Update $\beta_j^{(m+1)} \longleftarrow \frac{S(z_j, w_j^{(m+1)})}{v_j}$
    end for
  end while
  Calculate $w_j^{(m+1)}$ via (4)
  Calculate $\lambda^{(m+1)}$ via (5)
until $||\boldsymbol{\beta}^{(m+1)} - \boldsymbol{\beta}^{(m)}||_2^2 \le \varepsilon$
Output: $\boldsymbol{\beta}$
```

results in a drawback of backward stepwise variable selection; that is, if a covariant is deleted at any step, it will necessarily be excluded from the final selected model. To deal with this drawback, Wipf and Nagarajan (2010) suggested using $w_j = 1/(|\beta_j^{(m)}|^\gamma + \epsilon^2)$. Although this can alleviate the aforementioned drawback to some extent, it is difficult to choose the perturbation ϵ^2. Fortunately, our method does not meet this numerical instability due to the use of ETP.

In addition, our conditional MM algorithm enjoys the simple computational procedure same to the coordinate descent algorithms for the penalized linear regression with MCP or SCAD (Mazumder, Friedman, and Hastie 2009; Breheny and Huang 2010). However, an attractive advantage of our method over the coordinate descent algorithms is in that it also incorporates the adaptive update of the tuning parameter λ.

Extensions to Logistic Regression

In this section we consider a logistic regression model for a binary classification problem in which $y \in \{0, 1\}$. Let $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^T = \mathbf{X}\boldsymbol{\beta}$. The model is

$$P(y_i = 1|\mathbf{x}_i) = \pi_i = \frac{e^{\eta_i}}{1 + e^{\eta_i}}.$$

Estimation of $\boldsymbol{\beta}$ is now accomplished via minimization of the objective function

$$Q(\boldsymbol{\beta}) \\ = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log \pi_i + (1-y_i)\log(1-\pi_i)] + \sum_{j=1}^{p} P_{\lambda,\gamma}(|\beta_j|).$$

As pointed out by Breheny and Huang (2010), minimization can be approached by first obtaining a quadratic approximation to the loss function based on a Taylor series expansion about the value of the regression coefficients. That is,

$$Q(\boldsymbol{\beta}) \approx \frac{1}{2n}(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{W}(\tilde{\mathbf{y}} - \mathbf{X}\boldsymbol{\beta}) + \sum_{j=1}^{p} P_{\lambda,\gamma}(|\beta_j|),$$

where $\tilde{\mathbf{y}}$, the working response, is defined by $\tilde{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}^{(m)} + \mathbf{W}^{-1}(\mathbf{y} - \boldsymbol{\pi})$ and $\mathbf{W}$ is a diagonal matrix with diagonal elements $w_i = \pi_i(1 - \pi_i)$, and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)^T$ is evaluated at $\boldsymbol{\beta}^{(m)}$. With this preparation, we give a conditional MM algorithm in Algorithm 2.

Numerical Experiments

In this section we conduct experimental analysis about our sparse learning methods with ETP and also compare them with other closely related nonconvex methods.

Linear Regression

In this simulation example, we use a toy data model given by Tibshirani (1996). The data model is given as

$$y = \mathbf{x}^T\boldsymbol{\beta} + \sigma\epsilon \tag{6}$$

where $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0)^T$, $\epsilon \sim N(0, 1)$, and the input $\mathbf{x}$ is a 12-dimensional vector from multivariate normal distribution with covariance between x_i and x_j

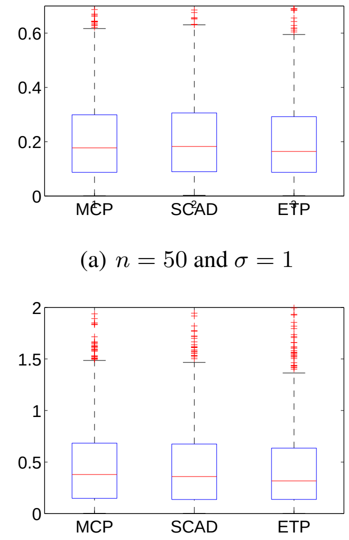
Table 2: Simulation results from the linear regression methods with MCP, SCAD and ETP, respectively. Here “C” is for “Correct” and “IC” for “Incorrect”.

Penalty	MRME(%)	”C”	”IC”	Total Time (s)
$n = 50, \sigma = 3$				
MCP	25.66	8.88	0.31	8.23
SCAD	25.39	8.68	0.13	21.07
ETP	**25.34**	8.76	0.18	2.34
$n = 50, \sigma = 1$				
MCP	17.00	9.00	0.03	7.86
SCAD	18.24	8.99	0.01	22.43
ETP	**16.41**	9.00	.0	1.61
$n = 100, \sigma = 1$				
MCP	20.21	9.00	.0	8.47
SCAD	**18.79**	9.00	.0	17.70
ETP	18.89	9.00	.0	1.97
$n = 100, \sigma = 5$				
MCP	37.87	8.80	0.34	9.49
SCAD	35.96	8.82	0.36	24.62
ETP	**31.70**	8.78	0.31	2.22

as $0.5^{|i-j|} (1 \leq i, j \leq 12)$. In our experiments, we use the different n (the data size) and σ. For each pair (n, σ), we randomly generate 1000 datasets. In other words, we randomly repeat 1000 times for each pair setting. Our reported results are based on the average of 1000 runs.

In the experiment we use the conditional MM method in Algorithm 1 to train the linear regression model based on ETP. For comparison, we also implement the coordinate descent methods (Breheny and Huang 2010) for the two linear regression methods based on MCP and SCAD, respectively. As we know, these three methods include the parameter γ. Here we use the same setting of γ for them. We use the median of relative model errors (MRME) as an evaluation criterion. The relative model error is defined as $\frac{d^2_{etp}}{d^2_{ols}}$, where d is the Mahalanobis distance between $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}$. The variable selection accuracy is measured by the average number of coefficients correctly setting to 0 in $\hat{\boldsymbol{\beta}}$, and vice visa. If a method is accurate, then the number of “correct” zeros in $\hat{\boldsymbol{\beta}}$ is 9 and “incorrect” is 0.

Table 2 reports the average results over the 1000 runs. From Table 2, we can see that these three nonconvex approaches are competitive in regression accuracy and sparse ability at low noise level. However, ETP has some advantages when the noise is large. Besides, it is worth pointing out that the performance of the methods based on MCP and SCAD is sensitive to the value of the tuning parameter λ. Here the reported results for these two methods are based on the optimum value of λ, which is selected via the grid search. However, this search usually takes large computational costs. Fortunately, our method can avoid this problem. In Table 2, we also present the computational times of the three methods. It is seen that our method is more efficient than the other two methods.



(a) $n = 50$ and $\sigma = 1$

(b) $n = 1000$ and $\sigma = 5$

Figure 3: Box-and-whisker plots of regression errors, which was conducted on the data from 1000 independent runs using MCP, SCAD and ETP.

Table 3: Classification dataset description (%).

Data set	# of Feature	# of Instance
Breast cancer	30	569
Diabetes	8	768
Gene	5000	72
Statlog(heart)	14	270
Musk(version1)	166	578
Musk(version2)	166	6700
Australian	14	690
WebKB	300	2053

Logistic Regression for Classification

In this experiment, we conduct the performance analysis of Algorithm 2 in classification problems. We also compare our method with the two nonconvex methods based on MCP and SCAD respectively. For the fair of comparison, these two methods are implemented via the coordinate descent methods (Breheny and Huang 2010).

We perform the analysis on eight binary classification datasets. The sizes of the datasets are described in Table 3. We split each dataset into 80% for training and 20% for test. We repeat 10 splits for our analysis and the reported results are based on the average of these 10 repeats. Table 4 gives the classification accuracy on the test datasets and Table 5 gives the coefficient sparsity (zero entries proportion) of the regression vector $\boldsymbol{\beta}$ estimated from the three methods, respectively. The running time of each algorithm on each dataset is given in Table 6.

The results are encouraging, because in most cases our method performs over the other two methods in both accu-

Table 4: Classification accuracies on the eight datasets (%)

DATA SET	ETP	MCP	SCAD
BREAST CANCER	**97.3**	**97.3**	96.4
DIABETES	**72.7**	70.6	70.6
GENE	**80.0**	71.4	71.4
STATLOG(HEART)	**90.7**	**90.7**	88.9
MUSK(VERSION1)	**82.3**	80.0	80.0
MUSK(VERSION2)	**72.4**	70.4	71.5
AUSTRALIAN	89.1	89.1	**92.0**
WEBKB	96.0	**96.5**	95.3

Table 5: Sparsity on the eight datasets (%)

DATA SET	ETP	MCP	SCAD
BREAST CANCER	76.7	**93.3**	83.3
DIABETES	**87.5**	**87.5**	75.0
GENE	**99.9**	68.1	**99.9**
STATLOG(HEART)	**76.2**	61.5	61.5
MUSK(VERSION1)	**90.4**	87.4	89.2
MUSK(VERSION2)	**92.8**	88.6	69.3
AUSTRALIAN	**85.7**	78.6	78.6
WEBKB	**92.3**	90.3	90.7

racy and sparsity. Moreover, the computational times given in Table 6 again show that our method is computationally feasible in comparison with the other two methods.

Conclusion

In this paper we have proposed the exponential-type penalty, which is nonconvex and singular at the origin. Thus, the resulting penalized estimator enjoys the the properties of sparsity, continuity and unbiasedness. In particular, we have applied the exponential-type penalty to sparse linear regression and sparse logistic regression problems. We have also devised iterative reweighted ℓ_1 minimization methods for the solutions of the problems. Moreover, we have presented a simple scheme for the adaptive update of the tuning parameter under our nonconvex approach. This simple scheme makes our approach very efficient and effective in comparison with other popular nonconvex approaches. Our experiment results have demonstrated the efficiency and effectiveness of our approach.

Acknowledgments

This work is supported by the Natural Science Foundations of China (No. 61070239), the 973 Program of China (No. 2010CB327903), the Doctoral Program of Specialized Research Fund of Chinese Universities (No. 20090101120066), and the Fundamental Research Funds for the Central Universities.

References

Breheny, P., and Huang, J. 2010. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *To appear in the Annals of Applied Statistics.*

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.

Efron, B.; Hastie, T.; Johnstone, I.; and Tibshirani, R. 2004. Least angle regression (with discussions). *The Annals of Statistics* 32(2):407–499.

Fan, J., and Li, R. 2001. Variable selection via nonconcave penalized likelihood and its Oracle properties. *Journal of the American Statistical Association* 96:1348–1361.

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

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

Lange, K.; Hunter, D.; and Yang, I. 2000. Optimization transfer using surrogate objective functions(with discussion). *Journal of Computational and Graphical Statistics* 9:1–59.

Mazumder, R.; Friedman, J.; and Hastie, T. 2009. Sparsenet:coordinate descent with non-convex penalties. Technical report, Department of Statistics, Stanford University.

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.

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.

Table 6: Computational times on the eight datasets (s).

DATA SET	ETP	MCP	SCAD
BREAST CANCER	0.132	2.29	2.44
DIABETES	0.079	0.53	0.60
GENE	0.590	3.66	5.97
STATLOG(HEART)	0.028	0.37	0.43
MUSK(VERSION1)	0.212	1.95	2.69
MUSK(VERSION2)	113.6	1034	737.7
AUSTRALIAN	0.132	1.12	1.04
WEBKB	3.84	48.2	48.9