Communications for Statistical Applications and Methods
2015, Vol. 22, No. 2, 147–157

DOI: http://dx.doi.org/10.5351/CSAM.2015.22.2.147
Print ISSN 2287-7843 / Online ISSN 2383-4757

An Additive Sparse Penalty for Variable Selection in High-Dimensional Linear Regression Model

Sangin Lee[1,a]

[a]University of Texas Southwestern Medical Center, USA

Abstract

We consider a sparse high-dimensional linear regression model. Penalized methods using LASSO or nonconvex penalties have been widely used for variable selection and estimation in high-dimensional regression models. In penalized regression, the selection and prediction performances depend on which penalty function is used. For example, it is known that LASSO has a good prediction performance but tends to select more variables than necessary. In this paper, we propose an additive sparse penalty for variable selection using a combination of LASSO and minimax concave penalties (MCP). The proposed penalty is designed for good properties of both LASSO and MCP. We develop an efficient algorithm to compute the proposed estimator by combining a concave convex procedure and coordinate descent algorithm. Numerical studies show that the proposed method has better selection and prediction performances compared to other penalized methods.

Keywords: Additive sparse penalty, concave convex procedure, coordinate descent algorithm, LASSO, minimax concave penalty, variable selection.

1. Introduction

Variable selection is a fundamental issue for high-dimensional statistical modeling; consequently, many penalized regressions have been proposed as effective methods for variable selection and estimation. Tibshirani (1996) introduced the least absolute shrinkage and selection operator (LASSO) that performs a subset selection in continuous fashion and satisfies good prediction performance. The LASSO has many attractive properties, however, it is also known that the LASSO tends to select many more variables than necessary.

There has been much work on various other penalized methods to overcome the variable selection deficiencies of the LASSO. These include nonconvex penalties such as smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001) and minimax concave penalty (MCP) (Zhang, 2010). Their authors showed that the SCAD and MCP methods have better selection performance than the LASSO in terms of the selection consistency in the asymptotic sense. Furthermore, many authors established the SCAD and MCP estimators satisfy the so-called *oracle property*, which means that they achieve the asymptotic equivalence to the ideal non-penalized estimator (*oracle estimator*) whose coefficients of irrelevant variables were zero in advance (Kim *et al.*, 2008; Zhang, 2010). Although the SCAD and MCP satisfy good asymptotic properties, it has been empirically observed that their prediction performances are not superior to the LASSO in many cases.

[1] Quantitative Biomedical Research Center, University of Texas Southwestern Medical Center, 5323 Harry Hines Blvd., Dallas, TX 75390, USA. E-mail: sanginlee44@gmail.com

Published 31 March 2015 / journal homepage: http://csam.or.kr

Table 1: Simulation results in the motivated examples.

Method	SNR = 1			SNR = 3		
	PE	SIG	NOI	PE	SIG	NOI
LASSO	4.7457	4.69	10.81	4.7771	5.00	11.44
MCP	4.9753	3.36	5.48	4.3935	5.00	3.82
SCAD	4.9592	3.97	7.97	4.3950	5.00	6.30

Motivated Example Here we briefly investigate selection and prediction performances of LASSO, SCAD and MCP through simple simulations. This is a motivation of the paper. We generate 100 simulated data sets that consist of $n = 100$ observations and $p = 100$ explanatory variables from the linear model,

$$y = x_1\beta_1^* + \cdots + x_p\beta_p^* + \varepsilon, \tag{1.1}$$

where $\varepsilon \sim N(0, 4)$. All explanatory variables marginally follow standard Gaussian distribution and the correlation between x_i and x_j is $0.5^{|i-j|}$. The first 5 true coefficients β_j^* are set to be c and the remaining 95 coefficients equal to zero. The values of the nonzero true coefficients c are chosen at two different signal levels so that the signal to noise ratios (SNR) are 1 and 3. For each data set, we record the prediction error (PE) based on 1,000 independent test data sets and the number of selected nonzero coefficients among the 5 true signal variables (SIG) and 95 noisy variables (NOI), respectively.

Table 1 presents the results of this simulation, averaged over the 100 replications. When the SNR is low, the LASSO outperforms both MCP and SCAD in terms of prediction and selection accuracy, whereas both MCP and SCAD show better prediction performance and selectivity than the LASSO under SNR = 3. Based on these results, we can show that the selection and prediction performances for each method depend on the true underlying model. The detailed descriptions and more simulations with the proposed method are presented in Section 4.

The goal of this paper is to develop a new additive penalty that works like an intermediate penalty between LASSO and MCP as well as LASSO or MCP according to a given data set. In Section 2, we propose the additive sparse penalty (ASP) by combining the LASSO and MCP functions. The ASP is designed to have good properties of both LASSO and MCP regardless of the true underlying model. In Section 3, we present the optimization algorithm for computing the ASP estimator, and in Section 4, we compare the performance of the proposed method with LASSO and MCP by simulation studies and real data analysis. Concluding remarks are provided in Section 5.

2. Additive Sparse Penalty

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.1}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the vector of n response variables, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ is the $n \times p$ design matrix with the j^{th} column $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$ is the vector of regression coefficients and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^T$ is the vector of random errors.

2.1. Definition

We propose the additive sparse penalized estimator defined as

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} J_{\lambda_1}(|\beta_j|) + \lambda_2 \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.2}$$

where $J_{\lambda_1}(\cdot)$ is the MCP function of Zhang (2010) defined as

$$J_{\lambda_1}(\beta) = \begin{cases} -\dfrac{\beta^2}{2a} + \lambda_1 \beta, & \text{if } \beta \le a\lambda_1, \\ \dfrac{1}{2} a\lambda_1^2, & \text{if } \beta > a\lambda_1, \end{cases}$$

for $\beta \ge 0$, $\lambda_1, \lambda_2 \ge 0$ and $a > 1$. Here λ_1 and λ_2 are the regularization parameters for the MCP and LASSO, respectively. This penalty is designed by combining the MCP and LASSO, and hence it is expected to satisfy good properties of both MCP and LASSO. Note that the MCP function could be replaced by other nonconvex penalty such as the SCAD penalty of Fan and Li (2001) defined as

$$J_{\lambda_1}(\beta) = \begin{cases} \lambda_1 \beta, & \text{if } \beta \le \lambda_1, \\ \dfrac{a\lambda_1(\beta - \lambda_1) - (\beta^2 - \lambda_1^2)/2}{(a-1)}, & \text{if } \lambda < \beta \le a\lambda_1, \\ \dfrac{1}{2}(a+1)\lambda_1^2, & \text{if } \beta > a\lambda_1, \end{cases}$$

for $\beta \ge 0$, $\lambda_1 \ge 0$ and $a > 2$.

Consider the following parametrization: $\lambda = \lambda_1 + \lambda_2$, $\alpha = \lambda_1/\lambda$. Then the additive sparse penalty (ASP) can be expressed as

$$\sum_{j=1}^{p} \left\{ J_{\alpha\lambda}(|\beta_j|) + (1-\alpha)\lambda|\beta_j| \right\}. \tag{2.3}$$

Using this parametrization, we can show that the ASP is an intermediate penalty between the LASSO ($\alpha = 0$) and the MCP ($\alpha = 1$). The regularization parameter λ controls the overall sparsity of the estimator, and α represents the fraction of each shrinkage amount from two penalties. Figure 1 shows the ASP penalty with various values of α compared to the MCP and LASSO. When α is small, the ASP is close to the LASSO penalty. When α is large, the ASP becomes similar to the MCP. By controlling value of α, we can obtain a good estimator for a given data. For example, when the signal to noise ratio is low as in the motivated example, we can obtain an estimator with good prediction performance by selecting a small value of α. A true signal level for a given data is unknown in practice, however, an appropriate value of α can be chosen by a model selection criterion such as BIC or cross-validation methods (Wang *et al.*, 2009; Zou and Hastie, 2005).

2.2. Orthogonal design study

To gain more insight of the ASP, we consider the orthogonal case where the design matrix is orthogonal, *i.e.* $\mathbf{X}^T\mathbf{X}/n = \mathbf{I}_p$. In this case, the problem in (2.2) becomes the problem of estimating coefficients separately in each variable. Let $\hat{\mathbf{z}} = (\hat{z}_1, \ldots, \hat{z}_p)^T$ be the ordinary least square estimator, where $\hat{z}_j = \mathbf{x}_j^T \mathbf{y}/n$. Then the ASP estimator can be obtained by component-wise solutions of

$$\frac{1}{2}\left(\hat{z}_j - \beta_j\right)^2 + J_{\lambda_1}\left(\left|\beta_j\right|\right) + \lambda_2 \left|\beta_j\right|, \quad j = 1, \ldots, p. \tag{2.4}$$

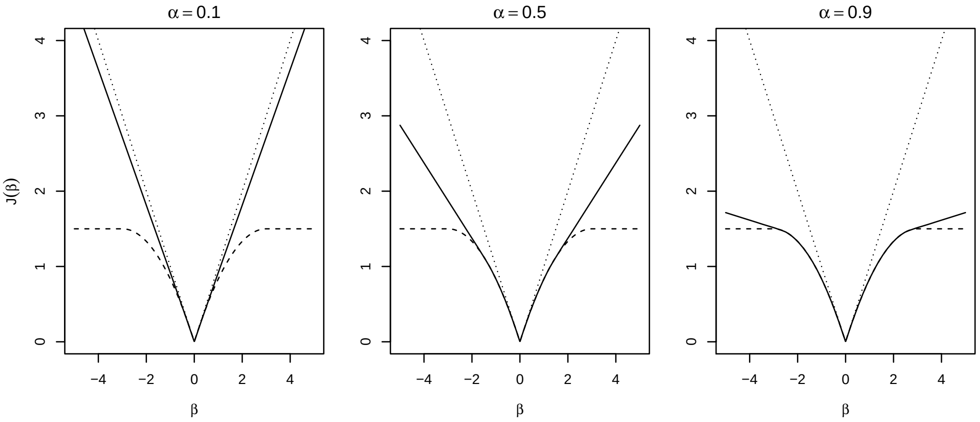

Figure 1: *Plot of the penalty functions with $\lambda = 1$. The dotted and dashed lines are the LASSO and MCP function, respectively. The solid line is the ASP functions with various values of α.*

It is easy to show that the minimizers of component-wise objective function (2.4) are

$$\hat{\beta}_j = \begin{cases} 0, & \text{if } \left|\hat{z}_j\right| \le \lambda_1 + \lambda_2, \\ \dfrac{a}{a-1}\text{sign}\left(\hat{z}_j\right)\left(\left|\hat{z}_j\right| - (\lambda_1 + \lambda_2)\right), & \text{if } \lambda_1 + \lambda_2 < \left|\hat{z}_j\right| \le a\lambda_1 + \lambda_2, \\ \text{sign}\left(\hat{z}_j\right)\left(\left|\hat{z}_j\right| - \lambda_2\right), & \text{if } \left|\hat{z}_j\right| > a\lambda_1 + \lambda_2 \end{cases}$$

for $a > 1$. This expression illustrates the feature of the ASP estimator. The $\lambda = \lambda_1 + \lambda_2$ controls the overall sparsity of the ASP estimator by thresholding small coefficients less than λ, while λ_2 represents the amount of shrinkages over nonzero large coefficients as in the LASSO. Figure 2 shows the corresponding solutions according to $\hat{z}$ for the ASP with various values of α. When α is small, the ASP solutions are very similar to ones of LASSO. When α is large, the ASP solutions become similar to ones of MCP. The thresholding rules for small values of $\hat{z}$ are the same as λ, but the amount of shrinkages for large values of $\hat{z}$ depends on α. Note that the amount of shrinkages for large values of $\hat{z}$ in the LASSO and MCP are λ and zero, respectively, $\lambda(1 - \alpha)$ in the ASP.

3. Computation

In this section, we develop an optimization algorithm to compute the ASP estimate. We adopt the idea of CCCP-SCAD algorithm proposed by Kim *et al.* (2008). The main idea of the proposed algorithm is to convert the ASP problem to the standard LASSO problem via concave convex procedure (CCCP) of Yuille and Rangarajan (2003), and then apply the coordinate descent (CD) algorithm of Friedman *et al.* (2010) to solve the LASSO problem.

3.1. Concave convex procedure

The ASP function can be decomposed by the sum of concave and convex functions,

$$J_{\lambda_1}(|\beta|) + \lambda_2|\beta| = \tilde{J}_{\lambda_1}(|\beta|) + (\lambda_1 + \lambda_2)|\beta|,$$

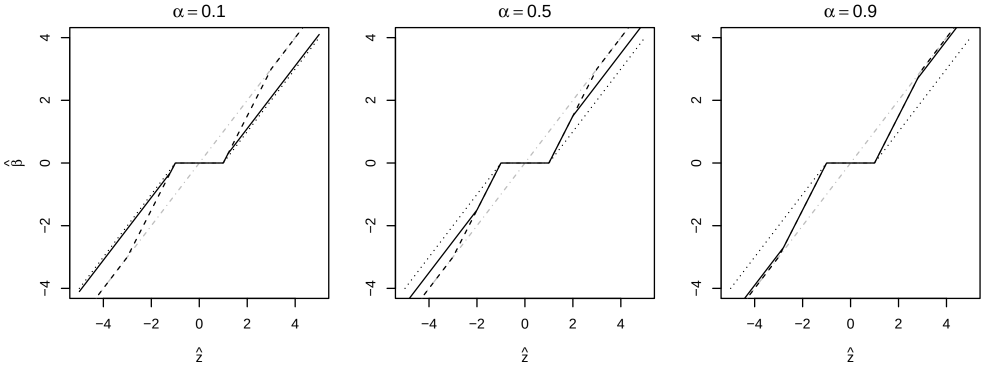

Figure 2: *Plot of the solution paths according to the least square estimator $\hat{z}$ for each method with $\lambda = 1$ in orthogonal design study. The dotted and dashed lines are the solution paths of the LASSO and MCP estimators, respectively. The solid line represents the ASP estimator with various values of α, and the dot-dashed line represents the least square estimator $\hat{z}$.*

where $\tilde{J}_{\lambda_1}(|\beta|) = J_{\lambda_1}(|\beta|) - \lambda_1|\beta|$ is a continuously differentiable concave function, and $|\beta|$ is a convex function. Hence, the objective function with the ASP in (2.2) can be rewritten as

$$Q(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1, \tag{3.1}$$

where $\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta}) = \sum_{j=1}^{p}\{J_{\lambda_1}(|\beta_j|) - \lambda_1|\beta_j|\}$, $\lambda = \lambda_1 + \lambda_2$ and $\|\cdot\|_1$ is the ℓ_1-norm operator. Note that $\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta})$ is a differentiable concave function with respect to $\boldsymbol{\beta}$, and hence the objective function in (3.1) consists of the sum of convex and concave functions. Thus, we can apply the CCCP algorithm. Since $\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta})$ is a concave function, for a given solution $\hat{\boldsymbol{\beta}}^c$ we have $\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta}) \leq \tilde{\mathbf{J}}_{\lambda_1}(\hat{\boldsymbol{\beta}}^c) + \nabla\tilde{\mathbf{J}}_{\lambda_1}(\hat{\boldsymbol{\beta}}^c)^T(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^c)$, where $\nabla\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta}) = \partial\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta})/\partial\boldsymbol{\beta}$. Given a current solution $\hat{\boldsymbol{\beta}}^c$, the tight convex upper bound of $Q(\boldsymbol{\beta})$ in (3.1) becomes

$$U(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \nabla\tilde{\mathbf{J}}_{\lambda_1}\left(\hat{\boldsymbol{\beta}}^c\right)^T\boldsymbol{\beta} + \lambda\|\boldsymbol{\beta}\|_1, \tag{3.2}$$

which is a standard LASSO problem with the quadratic loss function. We then update the current solution with the minimizer of $U(\boldsymbol{\beta})$ by applying an efficient LASSO algorithm. The ASP estimator can be obtained by iterating these two steps until convergence. By the descent property of CCCP algorithm, the objective function $Q(\boldsymbol{\beta})$ always decreases after each iteration and hence the sequence of solution vectors converges to a local minimum (Yuille and Rangarajan, 2003).

3.2. Coordinate descent algorithm

We now investigate details of the CD algorithm to optimize the tight convex upper bound $U(\boldsymbol{\beta})$ in (3.2). To explain the CD algorithm, we consider the j^{th} coordinate descent step. For a given fixed values of parameters $(\tilde{\beta}_k, k \neq j)$ at their current estimates $\tilde{\boldsymbol{\beta}}$, we wish to partially minimize the convex upper bound $U(\boldsymbol{\beta})$ with respect to β_j. Let q_j is the j^{th} diagonal entry of $\mathbf{X}^T\mathbf{X}/n$, a_j and b_j are the j^{th} element of $\mathbf{X}^T\mathbf{y}/n$ and $\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta})$, respectively, and $\boldsymbol{\zeta}_j$ and $\boldsymbol{\eta}_j$ are $(p-1)$-dimensional vectors obtained by deleting the j^{th} element from $\tilde{\boldsymbol{\beta}}$ and the j^{th} row vector of $\mathbf{X}^T\mathbf{X}/n$, respectively. Then using some algebra, it can be easily shown that this problem is equivalent to minimizing $R(\beta_j|\tilde{\boldsymbol{\beta}})$ defined as

$$R\left(\beta_j|\tilde{\boldsymbol{\beta}}\right) = \frac{1}{2}q_j\beta_j^2 + l_j\beta_j + \lambda|\beta_j| + C\left(\tilde{\boldsymbol{\beta}}\right), \tag{3.3}$$

where q_j and $l_j = \boldsymbol{\zeta}_j^T\boldsymbol{\eta}_j - a_j + b_j$ are the coefficients for the quadratic and linear terms, respectively. Here $C(\tilde{\boldsymbol{\beta}})$ is the constant terms free of β_j. It can be shown that the minimizer of $R(\beta_j|\tilde{\boldsymbol{\beta}})$ in (3.3) is

$$\hat{\beta}_j = -\frac{1}{q_j}\operatorname{sign}\left(l_j\right)\left(\left|l_j\right| - \lambda\right)_+,$$

where the subscript '+' stands for the positive part. This explicit form of solutions facilitates the implementation of the proposed algorithm which is summarized in Algorithm 1.

Algorithm 1 The proposed algorithm for minimizing $Q(\boldsymbol{\beta})$

```
Set an initial estimator $\hat{\boldsymbol{\beta}}^c \in \mathbb{R}^p$
Compute $\mathbf{X}^T\mathbf{X}/n$ and $\mathbf{X}^T\mathbf{y}/n$
repeat
    Calculate $\nabla\tilde{\mathbf{J}}_{\lambda_1}(\boldsymbol{\beta})$ at $\hat{\boldsymbol{\beta}}^c$ and set $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}^c$
    repeat
        for $j = 1, 2, \ldots, p$ do
            Calculate $q_j$ and $l_j$ in (3.3)
            Update $\tilde{\beta}_j$ with $\hat{\beta}_j = -\mathrm{sign}(l_j)(|l_j| - \lambda)_+/q_j$
        end for
    until convergence
    Update $\hat{\boldsymbol{\beta}}^c$ by $\hat{\boldsymbol{\beta}}$
until convergence
```

4. Numerical Illustrations

In this section, we investigate the finite sample performance of the ASP estimator through simulation experiments and real data analysis. We compare the ASP estimator with the LASSO, MCP and SCAD estimators in terms of prediction accuracy and variable selectivity.

4.1. Simulation studies

We consider the linear regression model

$$y = \mathbf{x}^T\boldsymbol{\beta}^* + \varepsilon,$$

where $\mathbf{x} \sim N_p(\mathbf{0}, \boldsymbol{\Sigma})$ with covariance structure $\Sigma_{ij} = \mathrm{Cov}(x_i, x_j) = 0.5^{|i-j|}$ and $\varepsilon \sim N(0, \sigma^2)$ with $\sigma = 2$ that of independent $\mathbf{x}$. For each independently generated data set, we set $n = 100, 200$ and $p = 100, 1000$, with the first 5 nonzero coefficients are set to be c and the remaining coefficients equal to zero. The values of the nonzero true coefficients c are chosen to two different signal levels so that the signal to noise ratios (SNR) are 1 and 3. The SNR is defined as

$$\mathrm{SNR} \equiv \sqrt{\frac{\mathrm{Var}\left(\mathbf{x}^T\boldsymbol{\beta}^*\right)}{\mathrm{Var}(\varepsilon)}} = \sqrt{\frac{\boldsymbol{\beta}^{*T}\boldsymbol{\Sigma}\boldsymbol{\beta}^*}{\sigma^2}}.$$

We use $c = 0.599$ for weak signal scenarios while we set $c = 1.798$ for strong signal scenarios. In summary, we have 8 simulation scenarios, where each scenario is replicated 100 times.

We consider the LASSO, MCP with $a = 3$ and SCAD with $a = 3.7$, roughly in line with recommendations suggested in Zhang (2010) and Fan and Li (2001). For ASP, we consider the ASP with various values of α in the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ (ASP$_\alpha$) and the ASP with optimal α (ASP$_{\alpha^*}$). For all methods, regularization parameters are selected by external validation on an independent data set with size of $n/2$. Note that all methods except ASP$_{\alpha^*}$ have only one regularization parameter λ, while ASP$_{\alpha^*}$ has two regularization parameters λ and α. For prediction accuracy, we compute the prediction error (PE) based on the independent test data set with size $N = 1{,}000$, and the model error (ME) which are defined by

$$\mathrm{PE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\,,$$

$$\mathrm{ME} = \left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right)^T \boldsymbol{\Sigma}\left(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\right),$$

respectively. For variable selectivity, we compute the number of selected nonzero coefficients among the true nonzero coefficients (SIG) and the true zero coefficients (NOI), respectively, as well as the number of selected variables (NUM). The average values of each measure based on 100 replications are summarized in Table 2 and Table 3.

Table 2 presents the weak signal scenario where SNR = 1. The LASSO outperforms the MCP and SCAD in terms of prediction accuracy and variable selectivity. When n is small or p is large, the MCP overall fails to detect the true signal variables compared to other methods. The LASSO successfully selects the true signal variables but it also selects more noisy variables than other methods. The SCAD shows intermediate selection performances between the LASSO and MCP, but its prediction performances are similar to the MCP. The ASP with various values of α shows intermediate performances between the LASSO and MCP. Furthermore, the ASP with a small value of α has better performances than others, which means the LASSO might be a more appropriate method than MCP and SCAD in the weak signal scenario. The ASP$_{\alpha^*}$ has better prediction performances than the LASSO and also deletes noisy variables without much missing of true signal variables.

Table 3 shows the strong signal scenario where SNR = 3. The MCP and SCAD has better PE and ME as well as variable selectivity regardless of n and p. While the LASSO shows poor prediction performances and also selects more noisy variables. Contrast to the weak signal scenario, the MCP and SCAD might be more appropriate method than the LASSO in the strong signal scenario. However, we can obtain the ASP estimator with good selection and prediction performances by choosing a large value of α. It is interesting to notice that the LASSO has similar selection and prediction performances in both strong and weak signal scenarios, whereas we can show that performances of the MCP and SCAD depend much on the signal levels and sample sizes. In this sense, the LASSO might be more robust than nonconvex penalized methods regardless of a feature of data.

Table 4 displays the frequency of each value of α being selected in the ASP$_{\alpha^*}$ among 100 random partitions. In weak signal scenarios, small values of α are selected so that the ASP performs similar to the LASSO, whereas large values of α are selected in strong signal scenarios in which the MCP performs well. Based on these simulation studies, we conclude that the ASP method has better performances than the LASSO and MCP by selecting an appropriate value of α depending on a given data.

Table 2: Simulation results for LASSO, MCP and ASP with various values of α, where the signal to noise ratio was set to 1. The corresponding standard errors are in parentheses.

n, p	Method	PE	ME	SIG	NOI	NUM
$n = 100$ $p = 100$	LASSO	4.746 (0.030)	0.712 (0.027)	4.7 (0.046)	11.1 (0.838)	15.8 (0.842)
	ASP$_{0.1}$	4.763 (0.033)	0.729 (0.030)	4.5 (0.059)	9.1 (0.794)	13.6 (0.810)
	ASP$_{0.3}$	4.742 (0.031)	0.705 (0.029)	4.2 (0.074)	7.0 (0.570)	11.2 (0.598)
	ASP$_{0.5}$	4.808 (0.033)	0.768 (0.031)	3.9 (0.082)	6.4 (0.544)	10.3 (0.588)
	ASP$_{0.7}$	4.901 (0.046)	0.862 (0.045)	3.7 (0.082)	6.4 (0.605)	10.0 (0.648)
	ASP$_{0.9}$	4.939 (0.038)	0.903 (0.038)	3.3 (0.086)	5.8 (0.523)	9.2 (0.560)
	MCP	4.972 (0.040)	0.936 (0.040)	3.3 (0.084)	5.5 (0.505)	8.7 (0.540)
	SCAD	4.965 (0.040)	0.930 (0.040)	3.9 (0.081)	8.4 (0.755)	12.3 (0.756)
	ASP$_{\alpha^*}$	4.739 (0.033)	0.703 (0.031)	4.2 (0.089)	7.2 (0.605)	11.4 (0.638)
$n = 100$ $p = 1{,}000$	LASSO	5.072 (0.039)	1.025 (0.038)	4.4 (0.065)	20.2 (1.476)	24.6 (1.490)
	ASP$_{0.1}$	5.062 (0.040)	1.015 (0.039)	4.2 (0.071)	13.3 (1.110)	17.5 (1.117)
	ASP$_{0.3}$	5.075 (0.045)	1.027 (0.044)	3.6 (0.079)	9.4 (0.811)	13.0 (0.830)
	ASP$_{0.5}$	5.123 (0.046)	1.067 (0.043)	3.2 (0.076)	7.9 (0.653)	11.0 (0.667)
	ASP$_{0.7}$	5.188 (0.041)	1.133 (0.039)	2.9 (0.074)	7.0 (0.646)	9.9 (0.659)
	ASP$_{0.9}$	5.241 (0.042)	1.186 (0.040)	2.8 (0.072)	6.3 (0.584)	9.1 (0.594)
	MCP	5.265 (0.044)	1.210 (0.042)	2.7 (0.072)	5.8 (0.563)	8.6 (0.567)
	SCAD	5.232 (0.042)	1.178 (0.040)	3.5 (0.097)	14.2 (1.245)	17.6 (1.203)
	ASP$_{\alpha^*}$	5.067 (0.042)	1.019 (0.041)	3.9 (0.089)	13.1 (1.160)	17.0 (1.182)
$n = 200$ $p = 100$	LASSO	4.347 (0.018)	0.319 (0.015)	5.0 (0.000)	12.3 (0.881)	17.3 (0.881)
	ASP$_{0.1}$	4.346 (0.017)	0.318 (0.015)	5.0 (0.010)	10.1 (0.836)	15.1 (0.837)
	ASP$_{0.3}$	4.304 (0.019)	0.277 (0.015)	4.9 (0.028)	6.8 (0.593)	11.8 (0.594)
	ASP$_{0.5}$	4.310 (0.018)	0.284 (0.015)	4.8 (0.041)	6.5 (0.463)	11.3 (0.471)
	ASP$_{0.7}$	4.361 (0.019)	0.334 (0.016)	4.7 (0.048)	6.9 (0.502)	11.7 (0.519)
	ASP$_{0.9}$	4.432 (0.020)	0.404 (0.016)	4.5 (0.064)	7.6 (0.562)	12.1 (0.592)
	MCP	4.474 (0.021)	0.444 (0.018)	4.5 (0.067)	7.9 (0.568)	12.3 (0.600)
	SCAD	4.491 (0.023)	0.462 (0.019)	4.7 (0.058)	12.1 (0.846)	16.8 (0.862)
	ASP$_{\alpha^*}$	4.306 (0.018)	0.281 (0.015)	4.9 (0.036)	7.3 (0.664)	12.2 (0.670)
$n = 200$ $p = 1{,}000$	LASSO	4.567 (0.018)	0.529 (0.017)	4.9 (0.026)	19.5 (1.576)	24.5 (1.581)
	ASP$_{0.1}$	4.562 (0.019)	0.523 (0.019)	4.9 (0.036)	13.1 (1.296)	18.0 (1.305)
	ASP$_{0.3}$	4.512 (0.021)	0.473 (0.020)	4.6 (0.059)	9.7 (0.913)	14.3 (0.934)
	ASP$_{0.5}$	4.546 (0.020)	0.507 (0.019)	4.2 (0.075)	9.1 (0.776)	13.3 (0.810)
	ASP$_{0.7}$	4.624 (0.021)	0.581 (0.018)	3.8 (0.082)	9.5 (0.833)	13.3 (0.875)
	ASP$_{0.9}$	4.682 (0.021)	0.641 (0.018)	3.6 (0.076)	9.4 (0.844)	13.1 (0.870)
	MCP	4.703 (0.022)	0.664 (0.020)	3.5 (0.080)	8.5 (0.844)	12.1 (0.865)
	SCAD	4.754 (0.022)	0.717 (0.020)	4.1 (0.076)	20.8 (2.032)	24.9 (2.026)
	ASP$_{\alpha^*}$	4.511 (0.020)	0.472 (0.019)	4.5 (0.063)	11.8 (1.289)	16.3 (1.305)

4.2. Real data analysis

We analyze the data set used by Scheetz *et al.* (2006) to illustrate the application of the proposed method as well as the LASSO and MCP. This data set consists of the gene expression levels of 18,976 genes obtained from 120 twelve-week-old male rats. This data set is available from the **R** package `picasso`. The main goal of the analysis is to identify genes whose expressions are most correlated with that of gene `TRIM32`, which has been found to cause Bardet-Biedl syndrome (Chiang *et al.*, 2006). We first select 3,000 genes with the largest variances in expression and then select top p genes that have the largest absolute correlations with gene `TRIM32`. We apply penalized linear regressions using the LASSO, MCP and the proposed method, with `TRIM32` expression as the response variable and the selected top $p = 100, 1000$ genes as the covariates.

We compare the prediction accuracy and selectivity of ASP with various values of α and the optimal value of α, LASSO and MCP as in the simulation studies. The results are obtained by 100 random partitions of data set divided into two parts, training (70%) and test (30%) data sets. For each

Table 3: Simulation results for LASSO, MCP and ASP with various values of α, where the signal to noise ratio was set to 3. The corresponding standard errors are in parentheses.

n, p	Method	PE	ME	SIG	NOI	NUM
n = 100 p = 100	LASSO	4.779 (0.032)	0.745 (0.029)	5.0 (0.000)	11.7 (0.843)	16.7 (0.843)
	ASP$_{0.1}$	4.782 (0.034)	0.750 (0.032)	5.0 (0.000)	9.4 (0.798)	14.4 (0.798)
	ASP$_{0.3}$	4.658 (0.032)	0.624 (0.030)	5.0 (0.000)	5.9 (0.532)	10.9 (0.532)
	ASP$_{0.5}$	4.504 (0.029)	0.472 (0.027)	5.0 (0.000)	4.0 (0.422)	9.0 (0.422)
	ASP$_{0.7}$	4.379 (0.027)	0.346 (0.024)	5.0 (0.000)	3.1 (0.370)	8.1 (0.370)
	ASP$_{0.9}$	4.357 (0.029)	0.325 (0.026)	5.0 (0.000)	3.1 (0.365)	8.1 (0.365)
	MCP	4.366 (0.028)	0.336 (0.026)	5.0 (0.000)	3.3 (0.352)	8.3 (0.352)
	SCAD	4.371 (0.034)	0.340 (0.032)	5.0 (0.000)	5.8 (0.553)	10.8 (0.553)
	ASP$_{\alpha^*}$	4.360 (0.027)	0.328 (0.024)	5.0 (0.000)	3.3 (0.387)	8.3 (0.387)
n = 100 p = 1,000	LASSO	5.183 (0.046)	1.134 (0.045)	5.0 (0.000)	21.4 (1.432)	26.4 (1.432)
	ASP$_{0.1}$	5.140 (0.044)	1.092 (0.043)	5.0 (0.000)	12.3 (1.023)	17.3 (1.023)
	ASP$_{0.3}$	4.870 (0.040)	0.824 (0.039)	5.0 (0.000)	7.3 (0.736)	12.3 (0.736)
	ASP$_{0.5}$	4.607 (0.033)	0.564 (0.032)	5.0 (0.000)	4.7 (0.534)	9.7 (0.534)
	ASP$_{0.7}$	4.445 (0.029)	0.407 (0.028)	5.0 (0.000)	4.2 (0.403)	9.2 (0.403)
	ASP$_{0.9}$	4.519 (0.046)	0.481 (0.045)	4.9 (0.010)	6.4 (0.532)	11.4 (0.532)
	MCP	4.600 (0.050)	0.561 (0.049)	4.9 (0.017)	8.4 (0.559)	13.4 (0.558)
	SCAD	4.651 (0.055)	0.606 (0.053)	5.0 (0.010)	18.7 (1.111)	23.7 (1.111)
	ASP$_{\alpha^*}$	4.470 (0.033)	0.434 (0.032)	5.0 (0.000)	6.0 (0.652)	11.0 (0.652)
n = 200 p = 100	LASSO	4.347 (0.018)	0.319 (0.015)	5.0 (0.000)	12.3 (0.881)	17.3 (0.881)
	ASP$_{0.1}$	4.346 (0.018)	0.319 (0.015)	5.0 (0.000)	10.1 (0.839)	15.1 (0.839)
	ASP$_{0.3}$	4.294 (0.017)	0.268 (0.014)	5.0 (0.000)	6.5 (0.606)	11.5 (0.606)
	ASP$_{0.5}$	4.225 (0.015)	0.199 (0.012)	5.0 (0.000)	4.3 (0.458)	9.3 (0.458)
	ASP$_{0.7}$	4.189 (0.015)	0.162 (0.011)	5.0 (0.000)	3.7 (0.490)	8.7 (0.490)
	ASP$_{0.9}$	4.169 (0.015)	0.141 (0.011)	5.0 (0.000)	3.0 (0.488)	8.0 (0.488)
	MCP	4.173 (0.015)	0.145 (0.011)	5.0 (0.000)	2.9 (0.483)	7.9 (0.483)
	SCAD	4.164 (0.017)	0.136 (0.013)	5.0 (0.000)	4.6 (0.719)	9.6 (0.719)
	ASP$_{\alpha^*}$	4.167 (0.014)	0.139 (0.010)	5.0 (0.000)	3.1 (0.506)	8.1 (0.506)
n = 200 p = 1,000	LASSO	4.570 (0.019)	0.532 (0.018)	5.0 (0.000)	19.9 (1.572)	24.9 (1.572)
	ASP$_{0.1}$	4.560 (0.020)	0.521 (0.019)	5.0 (0.000)	13.1 (1.275)	18.1 (1.275)
	ASP$_{0.3}$	4.435 (0.022)	0.397 (0.020)	5.0 (0.000)	8.9 (1.022)	13.9 (1.022)
	ASP$_{0.5}$	4.304 (0.015)	0.267 (0.013)	5.0 (0.000)	5.5 (0.692)	10.5 (0.692)
	ASP$_{0.7}$	4.224 (0.014)	0.186 (0.011)	5.0 (0.000)	4.1 (0.668)	9.1 (0.668)
	ASP$_{0.9}$	4.192 (0.014)	0.154 (0.011)	5.0 (0.000)	3.8 (0.695)	8.8 (0.695)
	MCP	4.184 (0.013)	0.145 (0.010)	5.0 (0.000)	3.3 (0.560)	8.3 (0.560)
	SCAD	4.197 (0.015)	0.158 (0.012)	5.0 (0.000)	9.5 (1.440)	14.5 (1.440)
	ASP$_{\alpha^*}$	4.197 (0.015)	0.158 (0.012)	5.0 (0.000)	4.2 (1.023)	9.2 (1.023)

Table 4: The frequency of each value of α being selected in the ASP$_{\alpha^*}$ methods based on 100 random partitions when p = 1,000.

n	SNR	α = 0	α = 0.1	α = 0.3	α = 0.5	α = 0.7	α = 0.9	α = 1
100	1	16	32	25	13	5	7	2
	3	0	3	4	18	33	15	27
200	1	6	18	40	18	8	4	6
	3	2	0	3	3	15	24	53

training data, the optimal values of regularization parameters are chosen by ten-fold cross-validation method, and then we compute the prediction error and the number of the selected nonzero variables based on each test data set. Table 5 summarizes the results based on 100 random partitions.

Table 5 shows the prediction error and the number of selected variables averaged over 100 random partitions. The LASSO performs better in terms of the prediction accuracy, but selects more variables than other methods. The MCP always performs worst in terms of the prediction accuracy, but it

Table 5: Average prediction errors and the number of selected variables based on 100 random partitions. The corresponding standard errors are in parentheses.

p	Measure	LASSO	ASP$_{0.1}$	ASP$_{0.3}$	ASP$_{0.5}$	ASP$_{0.7}$	ASP$_{0.9}$	MCP	ASP$_{\alpha^*}$
100	Prediction error	0.468	0.476	0.489	0.489	0.506	0.536	0.553	0.475
		(0.030)	(0.032)	(0.033)	(0.031)	(0.032)	(0.044)	(0.046)	(0.030)
	No. of variables	19.5	16.1	12.2	10.3	8.7	8.1	7.4	13.8
		(0.948)	(0.976)	(0.786)	(0.607)	(0.494)	(0.503)	(0.472)	(0.712)
1,000	Prediction error	0.495	0.494	0.479	0.543	0.614	0.648	0.667	0.478
		(0.055)	(0.053)	(0.038)	(0.046)	(0.063)	(0.068)	(0.071)	(0.039)
	No. of variables	43.5	36.7	29.0	21.9	20.1	14.1	12.3	32.9
		(1.595)	(1.380)	(1.319)	(1.209)	(1.185)	(0.962)	(0.898)	(1.524)

produces the most sparse model. Even though the ASP has slightly larger prediction accuracy than the LASSO when $p = 100$, it produces more sparse model. When $p = 1{,}000$, the ASP outperforms the LASSO and it also selects less variables than the LASSO. These results illustrate that the proposed method has better selection and prediction performances by choosing an appropriate value of α for a given data set.

5. Concluding Remarks

In this paper, we have proposed an additive sparse penalty for variable selection by combining the LASSO and MCP functions. We also have developed an optimization algorithm of a hybrid of the concave convex procedure and coordinate descent algorithm. The numerical results given in Section 4 show that the ASP estimator has both advantages of the LASSO and MCP by selecting an appropriate value of α. Furthermore, we provide the implementation of the proposed algorithm at `https://sites.google.com/site/sanginlee0404/`.

We only have focused on the linear regression model, however, the ASP method could be extended in a straightforward manner to various regression problems such as generalized linear models and Cox's regressions. Such extensions can be conducted by replacing the sum of squared residuals with the corresponding loss functions for various models. For examples, the corresponding loss functions are taken to the negative log-likelihood for a generalized linear model and negative partial likelihood for Cox's regression. We leave these problems as future works.

References

Chiang, A. P., Beck, J. S., Yen, H. J., Tayeh, M. K., Scheetz, T. E., Swiderski, R. E., Nishimura, D. Y., Braun, T. A., Kim, K. Y., Huang, J., Elbedour, K., Carmi, R., Slusarski, D. C., Casavant, T. L., Stone, E. M. and Sheffield, V. C. (2006). Homozygosity mapping with SNP arrays identifies TRIM32, an E3 ubiquitin ligase, as a Bardet–Biedl syndrome gene (BBS11), *Proceedings of the National Academy of Sciences*, **103**, 6287–6292.

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

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

Kim, Y., Choi, H. and Oh, H.-S. (2008). Smoothly clipped absolute deviation on high dimensions, *Journal of the American Statistical Association*, **103**, 1665–1673.

Scheetz, T. E., Kim, K.-Y. A., Swiderski, R. E., Philp, A. R., Braun, T. A., Knudtson, K. L., Dorrance, A. M., DiBona, G. F., Huang, J., Casavant, T. L., Sheffield, V. C. and Stone, E. M. (2006). Regulations of gene expression in the mammalian eye and its relevance to eye disease, *Proceedings of the National Academy of Sciences*, **103**, 14429–14434.

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

Wang, H., Li, B. and Leng, C. (2009). Shrinkage tuning parameter selection with a diverging number of parameters, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **71**, 671–683.

Yuille, A. L. and Rangarajan, A. (2003). The concave-convex procedure, *Neural Computation*, **15**, 915–936.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty, *Annals of Statistics*, **58**, 894–942.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**, 301–320.

Received December 30, 2014; Revised January 28, 2015; Accepted January 29, 2015