Vol. 38 (2018)
No. 6

数 学 杂 志
J. of Math. (PRC)

HIGH-DIMENSIONAL VARIABLE SELECTION WITH THE GENERALIZED SELO PENALTY

SHI Yue-yong[1,3], CAO Yong-xiu[2], YU Ji-chang[2], JIAO Yu-ling[2]

(*1.School of Economics and Management, China University of Geosciences, Wuhan 430074, China*)

(*2.School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan 430073, China*)

(*3.Center for Resources and Environmental Economic Research, China University of Geosciences, Wuhan 430074, China*)

Abstract: In this paper, we consider the variable selection and parameter estimation in high-dimensional linear models. We propose a generalized SELO (GSELO) method for solving the penalized least-squares (PLS) problem. A coordinate descent algorithm coupled with a continuation strategy and high-dimensional BIC on the tuning parameter are used to compute corresponding GSELO-PLS estimators. Simulation studies and a real data analysis show the good performance of the proposed method.

Keywords: continuation strategy; coordinate descent; high-dimensional BIC; local linear approximation; penalized least squares

2010 MR Subject Classification: 62J05; 62J07

Document code: A **Article ID:** 0255-7797(2018)06-0990-09

1 Introduction

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\mathbf{y} \in \mathbb{R}^n$ is a response vector, $\mathbf{X} \in \mathbb{R}^{n\times d}$ is a design matrix, $\boldsymbol{\beta} \in \mathbb{R}^d$ is an unknown sparse coefficient vector of interest and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a noise vector satisfying $\boldsymbol{\varepsilon} \sim (\mathbf{0}, \sigma^2\mathbf{I}_n)$. We focus on the high-dimensional case $d > n$. Without loss of generality, we assume that $\mathbf{y}$ is centered and the columns of $\mathbf{X}$ are centered and $\sqrt{n}$-normalized. To achieve sparsity, we consider the following SELO-penalized least squares (PLS) problem

$$\hat{\boldsymbol{\beta}} \triangleq \hat{\boldsymbol{\beta}}(\lambda,\gamma) = \mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^d} \left\{ Q(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{d} q_{\lambda,\gamma}(\beta_j) \right\}, \tag{1.2}$$

[*] **Received date:** 2017-06-02 **Accepted date:** 2017-10-18

Foundation item: Supported by National Natural Science Foundation of China (11501578; 11501579; 11701571; 41572315); Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (CUGW150809).

Biography: Shi Yueyong (1984–), male, born at Luzhou, Sichuan, lecturer, major in biostatistics.

Corresponding author: Jiao Yuling.

where

$$q_{\lambda,\gamma}(\beta_j)=\frac{\lambda}{\log(2)}\log\left(\frac{|\beta_j|}{|\beta_j|+\gamma}+1\right) \tag{1.3}$$

is the SELO penalty proposed by [1], λ and γ are two positive tuning (or regularization) parameters. In particular, λ is the sparsity tuning parameter obtaining sparse solutions and γ is the shape (or concavity) tuning parameter making SELO with small γ values mimic L_0, i.e., $q_{\lambda,\gamma}(\beta_j)\approx\lambda I(|\beta_j|\neq 0)$, when γ is small.

Intuitively, L_0 penalized methods directly penalize the number of variables in regression models, so they enjoy a nice interpretation of best subset selection [2]. The main challenge in implementing L_0 penalized methods is the discontinuity of the L_0 penalty function, which results in the lack of stability. As mentioned above, small γ values can make SELO mimic L_0, and the SELO penalty function is continuous, so SELO can largely retain the advantages of L_0 but yield a more stable model than L_0. The SELO penalized regression method was demonstrated theoretically and practically to be effective in nonconvex penalization for variable selection, including but not limited to linear models [1], generalized linear models [3], multivariate panel count data (proportional mean models) [4] and quantile regression [5].

In this paper, we first propose a generalized SELO (GSELO) penalty [6] closely resembling L_0 and retaining good features of SELO, and then we use the GSELO-PLS procedure to do variable selection and parameter estimation in high-dimensional linear models. Numerically, when coupled with a continuation strategy and a high-dimensional BIC , our proposed method is very accurate and efficient.

An outline for this paper is as follows. In Section 2, we introduce the GSELO method and corresponding GSELO-PLS estimator. In Section 3, we present the algorithm for computing the GSELO-PLS estimator, the standard error formulae for estimated coefficients and the selection of the tuning parameter. The finite sample performance of GSELO-PLS through simulation studies and a real data analysis are also demonstrated in Section 3. We conclude the paper with Section 4.

2 Methodology

Let $\mathcal{P}$ denote all GSELO penalties, f is an arbitrary function that satisfies the following two hypotheses:

(H1) $f(x)$ is continuous with respect to x and has the first and second derivative in $[0,1]$;

(H2) $f'(x)\geq 0$ for all x in $[0,1]$ and $\lim\limits_{x\to 0}\frac{f(x)}{x}=1$.

Then a GSELO penalty $q_{\lambda,\gamma}(\cdot)\in\mathcal{P}$ is given by

$$q_{\lambda,\gamma}(\beta_j)=\frac{\lambda}{f(1)}f\left(\frac{|\beta_j|}{|\beta_j|+\gamma}\right), \tag{2.1}$$

where λ (sparsity) and γ (concavity) are two positive tunning parameters. It is noteworthy that $q_{\lambda,\gamma}(\beta_j)$ is the SELO penalty when we take $f(x)=\log(x+1)$, and $f(x)=x$ derives the transformed L_1 penalty [7]. Table 1 lists some representatives of $\mathcal{P}$.

Table 1: GSELO penalty functions (LIN, SELO, EXP, SIN and ATN)

Name	Types of functions	$f(x)$	$q_{\lambda,\gamma}(\beta_j)$
LIN	linear	x	$\lambda\frac{\lvert\beta_j\rvert}{\lvert\beta_j\rvert+\gamma}$
SELO	logarithmic	$\log(x+1)$	$\frac{\lambda}{\log(2)}\log\left(\frac{\lvert\beta_j\rvert}{\lvert\beta_j\rvert+\gamma}+1\right)$
EXP	exponential	$1-\exp(-x)$	$\frac{\lambda}{1-\exp(-1)}\left[1-\exp\left(-\frac{\lvert\beta_j\rvert}{\lvert\beta_j\rvert+\gamma}\right)\right]$
SIN	trigonometric	$\sin(x)$	$\frac{\lambda}{\sin(1)}\sin\left(\frac{\lvert\beta_j\rvert}{\lvert\beta_j\rvert+\gamma}\right)$
ATN	inverse trigonometric	$\arctan(x)$	$\frac{\lambda}{\arctan(1)}\arctan\left(\frac{\lvert\beta_j\rvert}{\lvert\beta_j\rvert+\gamma}\right)$

The GSELO-PLS estimator for (1.1) is obtained via solving

$$\hat{\boldsymbol{\beta}} \triangleq \hat{\boldsymbol{\beta}}(\lambda,\gamma)=\arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d}\left\{Q(\boldsymbol{\beta})=\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+\sum_{j=1}^{d}q_{\lambda,\gamma}(\beta_j)\right\}, \tag{2.2}$$

where $q_{\lambda,\gamma}(\cdot)\in\mathcal{P}$.

3 Computation

3.1 Algorithm

For solving (2.2), we first employ the local linear approximation (LLA) [8] to $q_{\lambda,\gamma}(\cdot)\in\mathcal{P}$:

$$q_{\lambda,\gamma}(\beta_j)\approx q_{\lambda,\gamma}(\beta_j^k)+q'_{\lambda,\gamma}(\beta_j^k)(|\beta_j|-|\beta_j^k|), \tag{3.1}$$

where β_j^k are the kth estimates of β_j, $j=1,2,\cdots,d$, and $q'_{\lambda,\gamma}(\beta_j)$ means the derivative of $q_{\lambda,\gamma}(\beta_j)$ with respect to $|\beta_j|$. Given $\boldsymbol{\beta}^k$ of $\boldsymbol{\beta}$, we find the next estimate via

$$\boldsymbol{\beta}^{k+1}=\arg\min_{\boldsymbol{\beta}}\left\{\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2+\sum_{j=1}^{d}\omega_j^{k+1}|\beta_j|\right\}, \tag{3.2}$$

where $\omega_j^{k+1}=q'_{\lambda,\gamma}(\beta_j^k)$. Then we use a Gauss-Seidel type coordinate descent (CD) algorithm in [9] for solving (3.2). We summarize the LLA-CD procedure in Algorithm 1. Table 2 shows the derivatives of $q_{\lambda,\gamma}(\beta_j)$ in Table 1.

3.2 Covariance Estimation

Following [1], we estimate the covariance matrix for $\hat{\boldsymbol{\beta}}$ by using a sandwich formula

$$\widehat{\operatorname{cov}}(\hat{\boldsymbol{\beta}}_{\hat{\mathcal{A}}})=\hat{\sigma}^2\left\{\mathbf{X}_{\hat{\mathcal{A}}}^T\mathbf{X}_{\hat{\mathcal{A}}}+n\Delta_{\hat{\mathcal{A}},\hat{\mathcal{A}}}(\hat{\boldsymbol{\beta}})\right\}^{-1}\mathbf{X}_{\hat{\mathcal{A}}}^T\mathbf{X}_{\hat{\mathcal{A}}}\left\{\mathbf{X}_{\hat{\mathcal{A}}}^T\mathbf{X}_{\hat{\mathcal{A}}}+n\Delta_{\hat{\mathcal{A}},\hat{\mathcal{A}}}(\hat{\boldsymbol{\beta}})\right\}^{-1}, \tag{3.3}$$

where

$$\hat{\sigma}^2=(n-\hat{s})^{-1}\|\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\|^2, \hat{s}=|\widehat{\mathcal{A}}|, \widehat{\mathcal{A}}=\left\{j;\hat{\beta}_j\neq 0\right\}$$

Algorithm 1 LLA-CD

```
Input: $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{y} \in \mathbb{R}^n$, $\boldsymbol{\beta}^0 \in \mathbb{R}^d$, $\gamma$, $\lambda$, $\delta$ (tolerance) and $k_{\max}$ (the maximum number of iterations).
Output: $\hat{\boldsymbol{\beta}}$, the estimate of $\boldsymbol{\beta}$ in equation (3.2).
for $k = 0, 1, 2, \cdots$ do
  while $k < k_{\max}$ do
    for $j = 1, 2, \cdots, d$ do
      Calculate $z_j = n^{-1}\mathbf{x}_j^T\mathbf{r}_{-j} = n^{-1}\mathbf{x}_j^T\mathbf{r} + \beta_j^k$, where $\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}^k$, $\mathbf{r}_{-j} = \mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}^k_{-j}$, "$-j$" is introduced to refer to the portion that remains after the $j$th column or element is removed, and $\mathbf{r}_{-j}$ is the partial residuals of $\mathbf{x}_j$.
      Update $\beta_j^{k+1} \leftarrow S(z_j, \omega_j^{k+1})$, where $\omega_j^{k+1} = q'_{\lambda,\gamma}(\beta_j^k)$ and $S(t,\lambda) = \mathrm{sgn}(t)(|t| - \lambda)_+$ is the soft-thresholding operator.
      Update $\mathbf{r} \leftarrow \mathbf{r} - (\beta_j^{k+1} - \beta_j^k)\mathbf{x}_j$.
    end for
    if $\|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\|_\infty < \delta$ then
      break, $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{k+1}$.
    else
      Update $k \leftarrow k + 1$.
    end if
  end while
end for
```

and

$$\Delta(\boldsymbol{\beta}) = \mathrm{diag}\{q'_{\lambda,\gamma}(|\beta_1|)/|\beta_1|, \cdots, q'_{\lambda,\gamma}(|\beta_d|)/|\beta_d|\}.$$

For variables with $\hat{\beta}_j = 0$, the estimated standard errors are 0.

3.3 Tuning Parameter Selection

Following [1], we fix $\gamma = 0.01$ and concentrate on tuning λ via a high-dimensional BIC (HBIC) proposed by [10] to select the optimal tuning parameter $\hat{\lambda}$, which is defined as

$$\hat{\lambda} = \underset{\lambda\in\Lambda}{\arg\min}\left\{\mathrm{HBIC}(\lambda) = \log(\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)\|^2/n) + \frac{C_n\log(d)}{n}|M(\lambda)|\right\}, \tag{3.4}$$

Table 2: Derivatives of GSELO penalty functions (LIN, SELO, EXP, SIN and ATN)

Name	$q_{\lambda,\gamma}(\beta_j)$	$q'_{\lambda,\gamma}(\beta_j)$
LIN	$\lambda\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}$	$\lambda\frac{\gamma}{(\|\beta_j\|+\gamma)^2}$
SELO	$\frac{\lambda}{\log(2)}\log\left(\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}+1\right)$	$\frac{\lambda}{\log(2)}\frac{\gamma}{(\|\beta_j\|+\gamma)(2\|\beta_j\|+\gamma)}$
EXP	$\frac{\lambda}{1-\exp(-1)}\left[1-\exp\left(-\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}\right)\right]$	$\frac{\lambda}{1-\exp(-1)}\exp\left(-\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}\right)\frac{\gamma}{(\|\beta_j\|+\gamma)^2}$
SIN	$\frac{\lambda}{\sin(1)}\sin\left(\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}\right)$	$\frac{\lambda}{\sin(1)}\cos\left(\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}\right)\frac{\gamma}{(\|\beta_j\|+\gamma)^2}$
ATN	$\frac{\lambda}{\arctan(1)}\arctan\left(\frac{\|\beta_j\|}{\|\beta_j\|+\gamma}\right)$	$\frac{\lambda}{\arctan(1)}\frac{\gamma}{\|\beta_j\|^2+(\|\beta_j\|+\gamma)^2}$

where Λ is a subset of $(0, +\infty)$, $M(\lambda) = \{j : \hat{\beta}_j(\lambda) \neq 0\}$ and $|M(\lambda)|$ denotes the cardinality of $M(\lambda)$, and $C_n = \log(\log n)$.

For SELO, it is shown in [1] that $\hat{\boldsymbol{\beta}}(\lambda, \gamma) = 0$ for (1.2) whenever $\lambda > \lambda_{\max}$, where

$$\lambda_{\max} := \frac{\|\mathbf{y}\|^2}{2n} \log(2) \left\{ \log\left(\frac{\|\mathbf{y}\|^2}{\|\mathbf{y}\|^2 + 2n\gamma\|\mathbf{X}^T\mathbf{y}\|_\infty} + 1 \right) \right\}^{-1}.$$

Taking $q_{\lambda,\gamma}(\beta_j) = \lambda I(|\beta_j| \neq 0)$ (i.e., the L_0 penalty) in (2.2), we have $\hat{\boldsymbol{\beta}}(\lambda) = 0$ whenever $\lambda > \frac{1}{2}\|\mathbf{X}^T\mathbf{y}/n\|_\infty^2$ (e.g., [11]). Since GSELO approaches L_0 when γ is small, we set $\lambda_{\max} = \frac{1}{2}\|\mathbf{X}^T\mathbf{y}/n\|_\infty^2$ for GSELO for simplicity and convenience. Then we set $\lambda_{\min} = 1e - 10\lambda_{\max}$ and divide the interval $[\lambda_{\min}, \lambda_{\max}]$ into G (the number of grid points) equally distributed subintervals in the logarithmic scale. For a given γ, we consider a range of values for $\lambda : \lambda_{\max} = \lambda_0 > \lambda_1 > \cdots > \lambda_G = \lambda_{\min}$, and apply the continuation strategy [11] on the set $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_G\}$, i.e., solving the λ_{s+1}-problem initialized with the solution of λ_s-problem, then select the optimal λ from Λ using (3.4). For sufficient resolution of the solution path, G usually takes $G \geq 50$ (e.g., $G = 100$ or 200). Due to the continuation strategy, one can set $k_{\max} \leq 5$ in Algorithm 1 to get an approximate solution with high accuracy. Interested readers can refer to [11] for more details.

3.4 Simulation

In this subsection, we illustrate the finite sample properties of GSELO-PLS-HBIC with simulation studies. All simulations are conducted using MATLAB codes.

We simulated 100 data sets from (1.1), where $\boldsymbol{\beta} \in \mathbb{R}^d$, with $\beta_1 = 3$, $\beta_2 = 1.5$, $\beta_3 = -2$, and $\beta_j = 0$, if $j \neq 1, 2, 3$. The d covariates $\mathbf{z} = (z_1, \cdots, z_d)^T$ are marginally standard normal with pairwise correlations $\text{corr}(z_j, z_k) = \rho^{|j-k|}$. We assume moderate correlation between the covariates by taking $\rho = 0.5$. The noise vector $\boldsymbol{\varepsilon}$ is generated independently from $N(\mathbf{0}, \sigma^2\mathbf{I}_n)$, and two noise levels $\sigma = 0.1$ and 1 were considered. The sample size and the number of regression coefficients are $n = 100$ and $d = 400$, respectively. The number of simulations is $N = 100$.

To evaluate the model selection performance of GSELO-PLS-HBIC, we record the average estimated model size (MS) $N^{-1}\sum\limits_{s=1}^{N}|\widehat{\mathcal{A}}^{(s)}|$, the proportion of correct models (CM) $N^{-1}\sum\limits_{s=1}^{N} I\{\widehat{\mathcal{A}}^{(s)} = \mathcal{A}\}$, the average ℓ_∞ absolute error (AE) $N^{-1}\sum\limits_{s=1}^{N}\|\hat{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}\|_\infty$, the average ℓ_2 relative error (RE) $N^{-1}\sum\limits_{s=1}^{N}(\|\hat{\boldsymbol{\beta}}^{(s)} - \boldsymbol{\beta}\|_2/\|\boldsymbol{\beta}\|_2)$ and the median of the prediction mean squared error (MPMSE) over N simulated datasets, where the prediction mean squared error (PMSE) for each dataset is $n^{-1}\sum\limits_{i=1}^{n}(\hat{y}_i^{(s)} - y_i)^2$, $s = 1, 2, \cdots, N$. Table 3 summarizes simulation results for variable selection. With respect to parameter estimation, Table 4 presents the average of estimated nonzero coefficients (Mean), the average of estimated standard error (ESE) and the sample standard deviations (SSD).

Table 3: Simulation results for model selection. The numbers in parentheses are the corresponding standard deviations of PMSE

σ	Method	MS	CM	AE	RE	MPMSE
0.1	LIN	3.05	96%	0.0164	0.0051	0.0098(0.0015)
	SELO	3.08	94%	0.0165	0.0052	0.0096(0.0015)
	EXP	3.04	97%	0.0165	0.0051	0.0098(0.0015)
	SIN	3.05	96%	0.0163	0.0051	0.0097(0.0015)
	ATN	3.05	96%	0.0164	0.0051	0.0098(0.0015)
1	LIN	3.27	86%	0.2406	0.0799	0.9452(0.1300)
	SELO	3.25	83%	0.2390	0.0787	0.9501(0.1337)
	EXP	3.29	84%	0.2423	0.0799	0.9431(0.1313)
	SIN	3.24	87%	0.2379	0.0782	0.9494(0.1302)
	ATN	3.24	87%	0.2384	0.0791	0.9499(0.1320)

Overall, from Table 3 and Table 4, we see that the performance of LIN, SELO, EXP, SIN and ATN are quite similar, and these five GSELO penalties all can work efficiently in all considered criteria. ESEs agree well with SSDs. In addition, all procedures have worse performance in all metrics when the noise level σ increases from 0.1 to 1.

Table 4: Simulation results for parameter estimation

		$\beta_1 = 3$			$\beta_2 = 1.5$			$\beta_3 = -2$		
σ	Method	Mean	ESE	SSD	Mean	ESE	SSD	Mean	ESE	SSD
0.1	LIN	2.9997	0.0117	0.0118	1.5006	0.0131	0.0134	-1.9997	0.0118	0.0124
	SELO	2.9997	0.0117	0.0118	1.5006	0.0131	0.0134	-1.9997	0.0117	0.0125
	EXP	2.9997	0.0117	0.0118	1.5006	0.0131	0.0134	-1.9997	0.0118	0.0124
	SIN	2.9997	0.0117	0.0118	1.5006	0.0131	0.0134	-1.9997	0.0118	0.0124
	ATN	2.9997	0.0117	0.0118	1.5006	0.0131	0.0134	-1.9997	0.0118	0.0124
1	LIN	3.1243	0.1164	0.1640	1.3703	0.1306	0.1897	-1.9596	0.1159	0.1338
	SELO	3.1225	0.1167	0.1641	1.3667	0.1310	0.1893	-1.9598	0.1160	0.1296
	EXP	3.1225	0.1163	0.1613	1.3716	0.1305	0.1866	-1.9601	0.1158	0.1333
	SIN	3.1241	0.1168	0.1662	1.3667	0.1311	0.1889	-1.9598	0.1161	0.1301
	ATN	3.1216	0.1165	0.1626	1.3710	0.1307	0.1900	-1.9591	0.1159	0.1344

3.5 Application

We analyze the NCI60 microarray data which is publicly available in R package ISLR [12] to illustrate the application of GSELO-PLS-HBIC in high-dimensional settings. The data contains expression levels on 6830 genes from 64 cancer cell lines. More information can be obtained at http://genome-www.stanford.edu/nci60/. Suppose that our goal is to assess the relationship between the firt gene and the rest under model (1.1). Then, the response variable $\mathbf{y}$ is a numeric vector of length 64 giving expression level of the first gene, and the design matrix $\mathbf{X}$ is a 64×6829 matrix which represents the remaining expression values of 6829 genes. Since the exact solution for the NCI60 data is unknown, we consider an adaptive LASSO (ALASSO) [13] procedure using the glmnet package as the gold standard in comparison with the proposed GSELO-PLS-HBIC method. The following commands complete the main part of the ALASSO computation:

```
library(ISLR); X = NCI60$data[,-1]; y = NCI60$data[,1]
library(glmnet); set.seed(0); fit_ridge = cv.glmnet(X, y, alpha=0)
co_ridge = coef(fit_ridge, s = fit_ridge$lambda.min)[-1]
gamma=1;w= 1/abs(co_ridge)^gamma; w = pmin(w,1e10)
set.seed(0);fit_alasso= cv.glmnet(X, y, alpha=1, penalty.factor=w)
co_alasso = coef(fit_alasso, s = "lambda.min")
yhat=predict(fit_alasso, s = "lambda.min", newx=X,type="response")
```

Table 5: Gene selection results of the NCI60 data

Method	MS	PMSE	Gene
ALASSO	63	0.0003	12, 114, 376, 461, 462, 532, 570, 571, 837, 977, 1016, 1088, 1131, 1138, 1187, 1207, 1225, 1262, 1539, 1571, 1622, 1643, 1663, 1852, 1921, 2232, 2238, 2278, 2279, 2353, 2358, 2396, 2484, 2497, 2568, 2950, 3087, 3233, 3350, 3461, 3715, 3751, 3832, 4408, 4708, 4817, 4972, 5035, 5036, 5037, 5038, 5089, 5119, 5162, 5230, 5258, 5289, 5426, 5653, 6575, 6608, 6620, 6738
LIN	29	0.0004	1, 115, 323, 363, 570, 601, 843, 1571, 1643, 1879, 2233, 2358, 2450, 2712, 2969, 3195, 3231, 3393, 3517, 3574, 3675, 3776, 4392, 4568, 5509, 6144, 6537, 6608, 6758
SELO	31	0.0002	2, 254, 523, 535, 560, 812, 1050, 1087, 1213, 1571, 1626, 1879, 2233, 2358, 2419, 2896, 3517, 3692, 3896, 4159, 4392, 4422, 4499, 4855, 4912, 5509, 6537, 6585, 6608, 6670, 6758
EXP	28	0.0004	523, 606, 640, 1571, 1879, 2101, 2233, 2358, 3077, 3437, 3491, 3517, 3796, 3935, 4392, 4499, 4644, 4653, 4661, 4733, 4822, 5223, 5456, 5509, 6409, 6608, 6695, 6758
SIN	30	0.0003	374, 510, 744, 1028, 1099, 1229, 1283, 1571, 1879, 2233, 2358, 2543, 2966, 3022, 3101, 3105, 3331, 3423, 3439, 3728, 4408, 4499, 4719, 5119, 5199, 5509, 6507, 6608, 6719, 6758
ATN	32	0.0001	115, 316, 702, 772, 1457, 1520, 1571, 1879, 2233, 2326, 2575, 2579, 2728, 2966, 3039, 3195, 3314, 3332, 3517, 3557, 3776, 3857, 3927, 3935, 4199, 4392, 4443, 4499, 4916, 5252, 5509, 6608

Table 6: Estimated coefficients for common genes

Gene	ALASSO	LIN	SELO	EXP	SIN	ATN
1571	-0.1039	-0.5946	-0.4941	-0.4003	-0.3779	-0.2796
6608	0.0245	0.1421	0.2464	0.1692	0.2182	0.2049

Table 5 lists the results of ALASSO and GSELO (LIN, SELO, EXP, SIN and ATN), including the model size (MS), the prediction mean square errors (PMSE) and selected genes (i.e., the column indices of the design matrix $\mathbf{X}$). From Table 5, six sets identify 63, 29, 31, 28, 30 and 32 genes respectively and give similar PMSEs. The results indicate that GSELO-PLS-HBIC is well suited to the considered sparse regression problem and can generate a more parsimonious model while keeping almost the same prediction power. In particular, for 2 common genes shown in Table 6, although the magnitudes of estimates are not equal, they have the same signs, which suggests similar biological conclusions.

4 Concluding Remarks

We have focused on the GSELO method in the context of linear regression models. This method can be applied to other models, such as the Cox models, by using arguments as those used in [14, 15, 16, 17], which are left for future research.

References

[1] Dicker L, Huang B, Lin X. Variable selection and estimation with the seamless-L_0 penalty[J]. Stat. Sinica, 2013, 23: 929–962.

[2] Fan J, Lv J. A selective overview of variable selection in high dimensional feature space[J]. Stat. Sinica, 2010, 20: 101–148.

[3] Li Z, Wang S, Lin X. Variable selection and estimation in generalized linear models with the seamless L_0 penalty[J]. Canad. J. Stat., 2012, 40(4): 745–769.

[4] Zhang H, Sun J, Wang D. Variable selection and estimation for multivariate panel count data via the seamless-L_0 penalty[J]. Canad. J. Stat., 2013, 41(2): 368–385.

[5] Ciuperca G. Model selection in high-dimensional quantile regression with seamless L_0 penalty[J]. Stat. Prob. Lett., 2015, 107: 313–323.

[6] Shi Y, Cao Y, Yu J, Jiao Y. Variable selection via generalized SELO-penalized linear regression models[J]. Appl. Math. J. Chinese Univ., 2018, 33(2): 145–162.

[7] Nikolova M. Local strong homogeneity of a regularized estimator[J]. SIAM J. Appl. Math., 2000, 61(2): 633–658.

[8] Zou H, Li R. One-step sparse estimates in nonconcave penalized likelihood models[J]. Ann. Stat., 2008, 36(4): 1509–1533.

[9] Breheny P, Huang J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection[J]. Ann. Appl. Stat., 2011, 5(1): 232–253.

[10] Wang L, Kim Y, Li R. Calibrating nonconvex penalized regression in ultra-high dimension[J]. Ann. Statist., 2013, 41(5): 2505–2536.

[11] Jiao Y, Jin B, Lu X. A primal dual active set with continuation algorithm for the ℓ^0-regularized optimization problem[J]. Appl. Comput. Harmon. Anal., 2015, 39(3): 400–426.

[12] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning: with applications in R[M]. New York: Springer, 2013.

[13] Zou H. The adaptive lasso and its oracle properties[J]. J. Amer. Statist. Assoc., 2006, 101(476): 1418–1429.

[14] Shi Y, Cao Y, Jiao Y, Liu, Y. SICA for Cox's proportional hazards model with a diverging number of parameters[J]. Acta Math. Appl. Sinica, English Ser., 2014, 30(4): 887–902.

[15] Shi Y, Jiao Y, Yan L, Cao Y. A modified BIC tuning parameter selector for SICA-penalized Cox regression models with diverging dimensionality[J]. J. Math., 2017, 37(4): 723–730.

[16] Antoniadis A, Fryzlewicz P, Letué F. The Dantzig selector in Cox's proportional hazards model[J]. Scand. J. Stat., 2010, 37(4): 531–552.

[17] Cao Y, Huang J, Liu Y, Zhao X. Sieve estimation of Cox models with latent structures[J]. Biometrics, 2016, 72(4): 1086–1097.

基于广义SELO惩罚的高维变量选择

石跃勇^{1,3}, 曹永秀^{2}, 余吉昌^{2}, 焦雨领^{2}

(1.中国地质大学(武汉)经济管理学院, 湖北 武汉 430074)

(2.中南财经政法大学统计与数学学院, 湖北 武汉 430073)

(3.中国地质大学(武汉)资源环境经济研究中心, 湖北 武汉 430074)

摘要: 本文考虑高维线性模型中的变量选择和参数估计. 提出了一种广义的SELO方法求解惩罚最小二乘问题. 一种坐标下降算法结合调节参数的一种连续化策略和高维BIC被用来计算相应的GSELO-PLS估计. 模拟研究和实际数据分析显示了提出方法的良好表现.

关键词: 连续化策略; 坐标下降; 高维BIC; 局部线性逼近; 惩罚最小二乘

MR(2010)**主题分类号:** 62J05; 62J07 **中图分类号:** O212.1