Part 1: Regularization Methods for Variable Selection

Runze Li

The Pennsylvania State University

July 2017

2.1 Linear Regression Models and Penalized Least Squares

Data: $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$ iid from

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

where $E(\varepsilon|\mathbf{x}) = 0$, $\text{var}(\varepsilon|\mathbf{x}) = \sigma^2$, and $\dim(\mathbf{x}) = d$. To include an intercept, we can set $x_1 \equiv 1$.

Using Matrix notation:

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

Estimation of $\boldsymbol{\beta}$ and σ^2

$\boldsymbol{\beta}$ can be estimated by LSE or MLE:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

which is the best linear unbiased estimator (BLUE) with

$$\text{cov}\{\widehat{\boldsymbol{\beta}}|\mathbf{X}\} = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}.$$

σ^2 can be estimated by MSE:

$$\widehat{\sigma}^2 = \frac{1}{n-d}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$$

Compared with the MLE for σ^2:

$$\widehat{\sigma}^2_{mle} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$$

Test hypothesis on β

Linear hypothesis:

$$H_0 : C\beta = \mathbf{h} \quad \text{versus} \quad H_1 : C\beta \neq \mathbf{h},$$

where C is a $q \times d$ constant matrix with rank q and $\mathbf{h}$ is a q-dimensional vector. Under H_0, the least squares method is to minimize

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\beta)^2 = \|\mathbf{y} - \mathbf{X}\beta\|^2$$

subject to $C\beta = \mathbf{h}$. The resulting estimate is

$$\widehat{\beta}_0 = \widehat{\beta} - (\mathbf{X}^T\mathbf{X})^{-1}C^T\{C(\mathbf{X}^T\mathbf{X})^{-1}C^T\}^{-1}(C\widehat{\beta} - \mathbf{h}).$$

The residual sum of squares, denoted by RSS_1 and RSS_0 under the full model and the null hypothesis, are

$$\text{RSS}_1 = \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 \quad \text{and} \quad \text{RSS}_0 = \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_0\|^2$$

The F-test for H_0 is based on the decomposition

$$\text{RSS}_0 = (\text{RSS}_0 - \text{RSS}_1) + \text{RSS}_1$$

It can be shown that

$$\text{RSS}_1/\sigma^2 \sim \chi^2(n-d)$$

Assume that $\varepsilon \sim N(0, \sigma^2)$. Under H_0,

$$(\text{RSS}_0 - \text{RSS}_1)/\sigma^2 \sim \chi^2(q)$$

and is independent of RSS_1/σ^2.

$$F = \frac{(\text{RSS}_0 - \text{RSS}_1)/q}{\text{RSS}_1/(n-d)} \sim F_{q,n-d}$$

Under the normality assumption on the error ε, an alternative approach to F-test is the likelihood ratio test (LRT):

$$\lambda = 2\{\ell(H_1) - \ell(H_0)\} \to \chi^2(q)$$

It can be proved that

$$\lambda = 2\log(\mathrm{RSS}_0/\mathrm{RSS}_1)$$

which is a monotone increasing function of $\mathrm{RSS}_0/\mathrm{RSS}_1$, so is F-test. Thus, the LRT is equivalent to F-test.

Classical variable selection criteria

To reduce model bias, many variables are introduced

To enhance model predictability, one has to select significant variables

Tradeoff between bias and variance

There is a large amount of literature on the topic of variable selection

Miller, A. (2002). *Subset Selection in Regression*, 2nd Edition, Chapman and Hall, London.

Penalized least squares (PLS)

Penalized least squares:

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\sum_{j=1}^{d} p_\lambda(|\beta_j|)$$

where $p_\lambda(\cdot)$ is a penalty function with a tuning parameter λ.

Many classical variable (more precisely, **subset**) selection criteria are asymptotically equivalent to minimizing the following score

$$\text{RSS}_{\mathcal{P}} + c_{n,d}\text{df}_{\mathcal{P}}$$

over subset $\mathcal{P}$, where $c_{n,d}$ is a factor that may depend on the sample size n and the dimensional of covariate vector d, and $\text{df}_{\mathcal{P}}$ is the degrees of freedom of the subset $\mathcal{P}$.

L_0-Penalty (also called entropy penalty)

Consider L_0-penalty:

$$p_\lambda(|\beta_j|) = \frac{\lambda^2}{2} I(|\beta_j| \neq 0).$$

AIC and C_p is asymptotically equivalent to the following PLS:

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\frac{(\sigma\sqrt{2/n})^2}{2}\sum_{j=1}^{d} I(|\beta_j| \neq 0)$$

L_0-penalty and other ICs

BIC is asymptotically equivalent to the following PLS:

$$\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+n\frac{(\sigma\sqrt{\log(n)/n})^2}{2}\sum_{j=1}^{d}I(|\beta_j|\neq 0)$$

Risk Inflation Criterion (RIC, Foster and George, 1994) is defined as

$$\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+n\frac{(\sigma\sqrt{\log(d)/n})^2}{2}\sum_{j=1}^{d}I(|\beta_j|\neq 0)$$

Most information criteria, such as AICc and CIC, can be written

$$\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+\frac{\lambda^2}{2}I(|\beta_j|\neq 0)$$

with a specific value of λ.

Challenge

Since L_0 penalty is discontinuous, optimization of PLS with discontinuous penalty requires combinatory optimization over all possible subset.

This limits applications of these subset selection criteria for high dimensional data.

During the 25 years, people look for continuous penalty and corresponding problems can be solved by numerical optimization algorithms.

Before 2000, famous continuous penalties include
L_q penalty for bridge regression (Frank and Friedman, 1993), L_1 penalty for LASSO (Tibshirani, 1996) and L_2 penalty for ridge regression.

Constrained LS: Nonnegative garrote (Breiman, 1995) and LASSO.

How to choose penalty

Fan and Li (2001) provides deep insights how to choose penalty and advocates that a good penalized LSE should have three properties:

- continuity: to avoid instability in model prediction
  continuous in data

- sparsity: to reduce model complexity
  a thresholding rule: set small coeff. to 0

- Unbiasedness: to avoid unnecessary modeling bias
  unbiased when true coefficients are large

Orthonormal design cases

To get insights into the selection of penalty function, let us start with the simplest case, in which $\frac{1}{n}\mathbf{X}^T\mathbf{X} = I_d$.

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{LS}\|^2 + \frac{1}{2}\|\mathbf{X}\widehat{\boldsymbol{\beta}}_{LS} - \mathbf{X}\boldsymbol{\beta}\|^2$$

and

$$\frac{1}{2}\|\mathbf{X}\widehat{\boldsymbol{\beta}}_{LS} - \mathbf{X}\boldsymbol{\beta}\|^2 = \frac{n}{2}\|\widehat{\boldsymbol{\beta}}_{LS} - \boldsymbol{\beta}\|^2 = \frac{n}{2}\sum_{j=1}^{d}(\widehat{\beta}_{j,LS} - \beta_j)^2$$

Denote $z_j = \widehat{\beta}_{j,LS}$. Then for orthonormal design cases, the PLS becomes

$$\frac{n}{2}\sum_{j=1}^{d}(z_j - \beta_j)^2 + n\sum_{j=1}^{d} p_\lambda(|\beta_j|)$$

This allows us to consider the simplest PLS

Sufficient conditions for the desired properties

Consider the simplest PLS:

$$\frac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|)$$

where z corresponds to an element of LSE.
derivative:

$$\theta - z + p'_\lambda(|\theta|)\text{sgn}(\theta) = \text{sgn}(\theta)\{|\theta| + p'_\lambda(|\theta|)\} - z$$

- unbiasedness: iff $p'_\lambda(|\theta|) = 0$ for large $|\theta|$
- Sparsity: if $\min_\theta\{|\theta| + p'_\lambda(|\theta|)\} > 0$
- continuity: iff $\text{argmin}_\theta\{|\theta| + p'_\lambda(|\theta|)\} = 0$

Homework #1: Prove the above three statements.

Smoothly Clipped Absolute Deviation (SCAD) Penalty

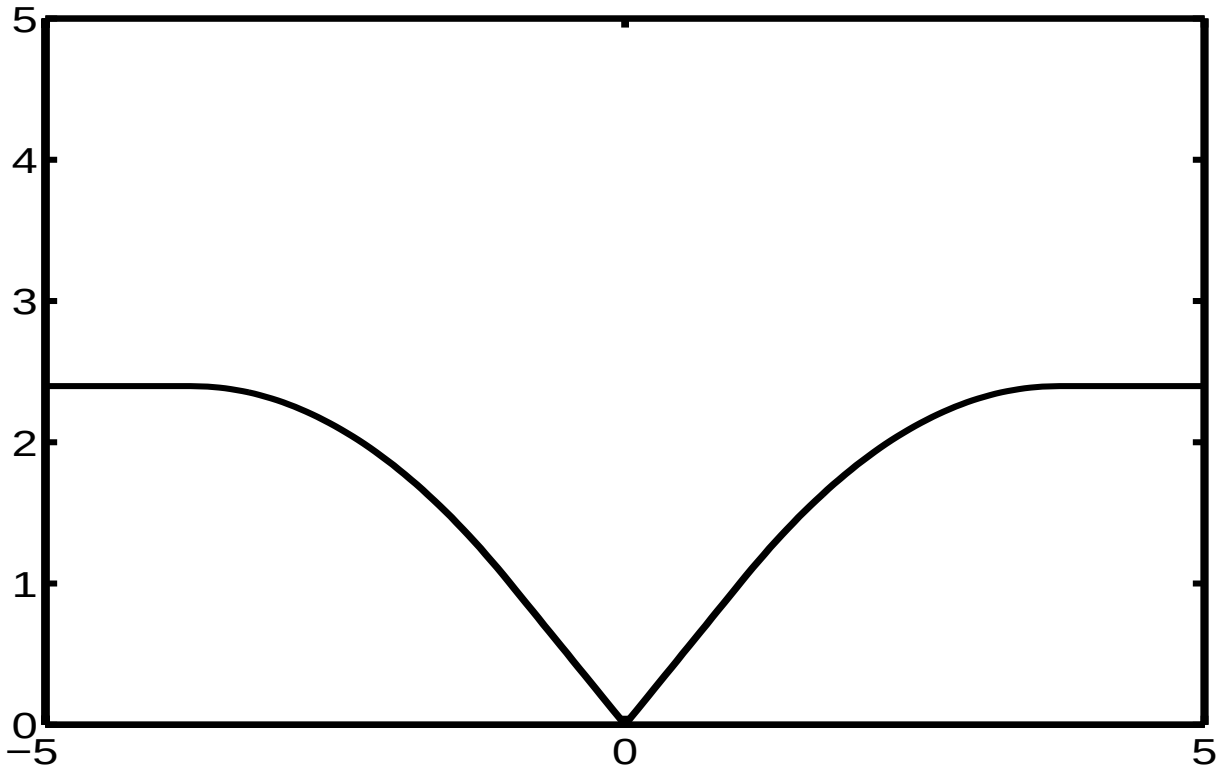
Fan and Li (2001, JASA) proposed using the SCAD penalty defined by

$$p'_\lambda(\theta) = \lambda\{I(\theta \leq \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda)\}$$

for $\theta > 0$ and $a > 2$ (Fan, 1997, Antoniadis and Fan, 2001 for wavelets)

In practice, set $a = 3.7$ from Bayesian point of view.

SCAD penalty



Other nonconvex penalties

Is the SCAD penalty only one satisfying the three desired properties?
Answer: NO.

Does there exist a better penalty than the SCAD?
Need to consider other criteria. There are some papers on this issue. Good for course projects.

Theoretic properties for PLS

Penalized least squares:

$$Q(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\sum_{j=1}^{d} p_\lambda(|\beta_j|)$$

where $p_\lambda(\cdot)$ is a penalty function with a tuning parameter λ.

Assumption on penalty:
I. Assume that $p_\lambda(\cdot)$ is a nonnegative, nondecreasing function with $p_\lambda(0) = 0$.

II. Assume that the penalty function $p_\lambda(\cdot)$ has a second order continuous derivative at nonzero components of $\boldsymbol{\beta}_0$, the true value of $\boldsymbol{\beta}$.

Let

$$\boldsymbol{\beta}_0 = (\beta_{10}, \cdots, \beta_{d0})^T = (\boldsymbol{\beta}_{10}^T, \boldsymbol{\beta}_{20}^T)^T,$$

where $\boldsymbol{\beta}_{10}$ consists of the first s components of $\boldsymbol{\beta}_0$.

Without loss of generality, assume that $\boldsymbol{\beta}_{20} = \mathbf{0}$ and all components of $\boldsymbol{\beta}_{10}$ are not equal to zero.

Denote

$$a_n = \max\{|p'_{\lambda_n}(|\beta_{j0}|)| : \beta_{j0} \neq 0\}$$

and

$$b_n = \max\{|p''_{\lambda_n}(|\beta_{j0}|)| : \beta_{j0} \neq 0\}.$$

Rate of Convergence

Theorem 1. Suppose that the observations $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$ are independent and identically distributed from the linear regression model, and assume that the matrix $E\mathbf{x}\mathbf{x}^T$, denoted by $\mathbf{V}$, is finite and positive definite, and the random error has mean zero and finite positive variance σ^2. If $a_n \to 0$ and $b_n \to 0$, then with probability tending to one, there exists a local minimizer $\widehat{\boldsymbol{\beta}}$ of $Q(\boldsymbol{\beta})$ such that $||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0|| = O_P(n^{-1/2} + a_n)$.

Remark: a_n depends on λ_n. So, the rate of convergence depends on λ_n through a_n.

To achieve root n consistency, we require that $a_n = O(1/\sqrt{n})$ and $b_n \to 0$ This requires that $\lambda_n \to 0$ for the hard-thresholding penalty and SCAD penalty, and that $\lambda_n = O_P(n^{-1/2})$ for the L_1 penalty.

Oracle Estimator

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}_1^T \boldsymbol{\beta}_1 + \mathbf{X}_2^T \boldsymbol{\beta}_2 + \boldsymbol{\varepsilon},$$

where $E(\boldsymbol{\varepsilon}|\mathbf{x}) = 0$ and $\text{cov}(\boldsymbol{\varepsilon}) = \sigma^2 I_n$.
Assume that $\boldsymbol{\beta}_2 = \mathbf{0}$, and all components of $\boldsymbol{\beta}_1$ are not equal to 0. This implies that the first part in the model is significant, while the second part is insignificant and should be deleted from the model.
An ideal estimator for $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_2 = 0,$$

and

$$\widehat{\boldsymbol{\beta}}_1 = (\mathbf{X}_1^T \mathbf{X})^{-1} \mathbf{X}_1^T \mathbf{y}.$$

This estimator correctly identifies the true model and efficiently estimates the nonzero components as if the true model were known.
Therefore, it is referred to as an **oracle estimator**,

Oracle Property

Theorem 2. Under the conditions of Theorem 1, assume that

$$\liminf_{n\to\infty} \liminf_{\beta\to 0^+} \sqrt{n} p'_{\lambda_n}(\beta) = +\infty,$$

then with probability tending to 1, the root n consistent local minimizer $\widehat{\boldsymbol{\beta}} = \begin{pmatrix} \widehat{\boldsymbol{\beta}}_1 \\ \widehat{\boldsymbol{\beta}}_2 \end{pmatrix}$ in Theorem 2.1 must satisfy:

(i) (**Sparsity**) $\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$;

(ii) (**Asymptotic normality**)

$$\sqrt{n}(\mathbf{V}_{11} + \Sigma)\left\{\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{10} + (\mathbf{V}_{11} + \Sigma)^{-1}\mathbf{b}\right\} \to N(\mathbf{0}, \sigma^2 \mathbf{V}_{11})$$

in distribution, where $\mathbf{V}_{11}$ consists of the first s rows and columns of $\mathbf{V}$ and

$$\Sigma = \text{diag}\left\{p''_{\lambda_n}(|\beta_{10}|), \cdots, p''_{\lambda_n}(|\beta_{s0}|)\right\},$$
$$\mathbf{b} = \left(p'_{\lambda_n}(|\beta_{10}|)\text{sgn}(\beta_{10}), \cdots, p'_{\lambda_n}(|\beta_{s0}|)\text{sgn}(\beta_{s0})\right)^T.$$

Oracle Property

For the SCAD penalty, if $\lambda_n \to 0$, and $\sqrt{n}\lambda_n \to \infty$, then $a_n = 0$. From Theorem 1, there exists a root n consistent local minimizer. Furthermore, from Theorem 2, this root n consistent local minimizer satisfies that $\widehat{\boldsymbol{\beta}}_2 = 0$ and $\sqrt{n}(\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{10})$ has an asymptotic normal distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{V}_{11}^{-1}$ as $b_n = 0$.

2.2 Generalized linear models and penalized likelihood

Reference book: McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models*, Chapman and Hall.

Is

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i$$

suitable for binary output: y_i =0 or 1?

For $\varepsilon \sim N(0, \sigma^2)$, LSE=MLE, so it is efficient.

For binary output, LSE=MLE? efficient?

The same questions for count data.

How to extend liner models to binary y, ..., ?

Three components of linear models

1. The *random component*: the components of $y|\mathbf{x}_i$ have independent $N(\mu_i, \sigma^2)$;
2. The *systematic component*: covariates $\mathbf{x}_i$ produce a *linear predictor* $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$;
3. The *link* between the random and systematic components:

$$\mu_i = \eta_i.$$

GLIM allows two extensions:

1. random components: allows y to be nonnormal
2. link components:

$$g(\mu_i) = \eta_i$$

$g(\cdot)$ is called the *link function*.

For the ordinal linear regression model, $g(u) = u$.

Binary response can be fitted by Bernoulli distribution,

Count response can be fitted by Poisson distribution,

Bernoulli, Poisson and normal belong to exponential family.

For the random component, GLIM considers $y|\mathbf{x}$ belongs to exponential family with density

$$f_y(y;\theta,\phi) = \exp\{(y\theta - b(\theta))/a(\phi) + c(y,\phi)\}$$

for some specific functions $a(\cdot) > 0$, $b(\cdot)$ and $c(\cdot)$.

Write $\ell(\theta,\phi;y) = \log f_y(y;\theta,\phi)$

Bartlette's identities

$$E\left(\frac{\partial \ell}{\partial \theta}\right) = 0$$

$$E\left(\frac{\partial^2 \ell}{\partial \theta^2}\right) + E\left(\frac{\partial \ell}{\partial \theta}\right)^2 = 0$$

Using the first Bartlette identity,

$$0 = E\left(\frac{\partial \ell}{\partial \theta}\right) = \{\mu - b'(\theta)\}/a(\phi)$$

Thus,

$$E(y) = \mu = b'(\theta).$$

Using the second Bartlette identity,

$$0 = -\frac{b''(\theta)}{a(\phi)} + \frac{\text{var}(y)}{a^2(\phi)}$$

so that

$$\text{var}(y) = b''(\theta)a(\phi)$$

$b''(\theta)$ is called the *variance function*

$a(\phi)$ commonly has the form

$$a(\phi) = \phi/w$$

for some weight w. ϕ is called the *dispersion parameter*.

Example. For $y|\mathbf{x} \sim N(\mu, \sigma^2)$,

$$f_y(y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-(y-\mu)^2/2\sigma^2\}$$

which can be rewritten as

$$\exp\{(y\mu - \mu^2/2)/\sigma^2 - \frac{1}{2}(y^2/\sigma^2 + \log(2\pi\sigma^2))\}$$

so, $\theta = \mu$, $\phi = \sigma^2$, and

$$a(\phi) = \phi, b(\theta) = \theta^2/2, c(y, \phi) = -\frac{1}{2}\{y^2/\sigma^2 + \log(2\pi\sigma^2)\}.$$

How to select a link function g?

$$\mu = b'(\theta), \quad \text{and} \quad \eta = g(\mu).$$

So,

$$\eta = g \circ b'(\theta)$$

Thus, $g(\cdot) = b'^{-1}(\cdot)$ is called the canonical link. With the canonical link, we have

$$\theta = \eta = \mathbf{x}^T \boldsymbol{\beta}.$$

For $y|\mathbf{x} \sim N(\mu, \sigma^2)$, canonical link is $g(\mu) = \mu$.
For $y|\mathbf{x} \sim$ Beroulli with p, canonical link is

$$g(p) = \log\{p/(1-p)\}$$

For $y|\mathbf{x} \sim$ Poisson with λ, canonical link is $g(\lambda) = \log(\lambda)$.

Can we choose other links?

Disadvantage of non-canonical link.

Nonconcavity of lkhd function, difficult to maximize.

Goodness of fit measures

For linear model, $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i$,

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}})^2 = \sum_{i=1}^{n} (y_i - \widehat{\mu})^2$$

measures how well a model fits the data.
For GLIM, we can extend the RSS in two ways: Generalized Pearson X^2 statistic and deviance

$$X^2 = \sum_{i=1}^{n} (y - \widehat{\mu})^2 / V(\widehat{\mu})$$

where $V(\widehat{\mu})$ is the estimated variance function of the distribution concerned.
Define $\widehat{\theta}_i = \theta(\widehat{\mu}_i)$ and $\tilde{\theta} = \theta(y_i)$. Deviance of $\mu_i, 1, \cdots, n$ is defined as

$$D(\mathbf{y}; \widehat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} 2w_i \{y_i(\tilde{\theta}_i - \widehat{\theta}_i) - b(\tilde{\theta}_i) + b(\widehat{\theta}_i)\}$$

Comments:

Motivation of deviance: likelihood ratio statistics

◦ Both D/ϕ and X^2/ϕ have exact χ^2 distribution for normal theory linear models, and asymptotic results are available for the other distributions

◦ The deviance has a general advantage as a measure of discrepancy (goodness of fit) in that it is additive for nested sets of models if MLEs are used, whereas X^2 in general is not.

◦ X^2 may sometimes be preferred because of its more direct interpretation.

Residuals

For normal linear model with $\varepsilon \sim N(0, \sigma^2)$,

$$r_i = y_i - \widehat{y}_i$$

For GLIM, two commonly used residuals: Pearson r and deviance r

- Pearson residual:

$$r_{P,i} = \frac{y_i - \widehat{\mu}_i}{\sqrt{V(\widehat{\mu}_i)}}$$

Thus,

$$X^2 = \sum_{i=1}^{n} r_{P,i}^2$$

deviance residual

Write

$$D = \sum_{i=1}^{n} d_i$$

Define

$$r_{D,i} = \text{sign}(y_i - \widehat{\mu}_i)\sqrt{d_i}$$

Algorithms

In general, there is no closed form solution for GLIM.

Newton-Raphson algorithm or Fisher scoring algorithm

Penalized likelihood and variable selection

- Linear model with normal ε

Penalized LS $\equiv$ Penalized LKHD

- Log-likelihood: $\ell\{\mathbf{x}^T\boldsymbol{\beta}, y\}$

Penalized Likelihood:
Find $\boldsymbol{\beta}$ to maximize

$$\begin{aligned} & \sum_{i=1}^{n} \ell\{\mathbf{x}_i^T\boldsymbol{\beta}, y_i\} - n\sum_{j=1}^{d} p_\lambda(|\beta_j|) \\ \widehat{=} \quad & \ell(\boldsymbol{\beta}) - n\sum_{j=1}^{d} p_\lambda(|\beta_j|) \\ \equiv \quad & Q(\boldsymbol{\beta}) \end{aligned}$$

Oracle Properties

Oracle properties: e.g.

$$y = \mathbf{x}_1^T \boldsymbol{\beta}_1 + \mathbf{x}_2^T \boldsymbol{\beta}_2 + \varepsilon$$

with $\boldsymbol{\beta}_2 = \mathbf{0}$, and all components of $\boldsymbol{\beta}_1 \neq 0$. $\text{var}(\varepsilon) = \sigma^2$.

With an oracle, ideal estimator:

$$\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$$

and

$$\widehat{\boldsymbol{\beta}}_1 = (\mathbf{X}_1^T \mathbf{X}_1)^{-1} \mathbf{X}_1^T \mathbf{y}$$

Without an oracle, our approach works as well as if we had an oracle, under general likelihood settings.

Regularity conditions

Set $\mathbf{V}_i = (\mathbf{x}_i, y_i)$, $i = 1, \cdots, n$, denoted by Ω the parameter space for $\boldsymbol{\beta}$ and the true value of $\boldsymbol{\beta}$ by $\boldsymbol{\beta}_0$

(A) The observations $\mathbf{V}_i$ are i.i.d. with density $f(\mathbf{V}, \boldsymbol{\beta})$. $f(\mathbf{V}, \boldsymbol{\beta})$ has a common support and the model is identifiable. Furthermore, the first and second logarithmic derivatives of f satisfying the equations

$$E_{\boldsymbol{\beta}_0}\left[\frac{\partial \log f(\mathbf{V}, \boldsymbol{\beta})}{\partial \beta_j}\right] = 0 \quad \text{for} \quad j = 1, \cdots, d$$

and

$$I_{jk}(\boldsymbol{\beta}_0) = E_{\boldsymbol{\beta}_0}\left[\frac{\partial}{\partial \beta_j} \log f(\mathbf{V}, \boldsymbol{\beta}) \frac{\partial}{\partial \beta_k} \log f(\mathbf{V}, \boldsymbol{\beta})\right]$$

$$= E_{\boldsymbol{\beta}_0}\left[-\frac{\partial^2}{\partial \beta_j \partial \beta_k} \log f(\mathbf{V}, \boldsymbol{\beta})\right].$$

(B) The Fisher information matrix

$$I(\boldsymbol{\beta}) = E\left\{\left[\frac{\partial}{\partial \boldsymbol{\beta}} \log f(\mathbf{V}, \boldsymbol{\beta})\right]\left[\frac{\partial}{\partial \boldsymbol{\beta}} \log f(\mathbf{V}, \boldsymbol{\beta})\right]^T\right\}$$

is finite and positive definite at $\boldsymbol{\beta} = \boldsymbol{\beta}_0$.

(C) There exists an open subset ω of Ω containing the true parameter point $\boldsymbol{\beta}_0$ such that for almost all $\mathbf{V}$ the density $f(\mathbf{V}, \boldsymbol{\beta})$ admits all third derivatives $\frac{\partial f(\mathbf{V}, \boldsymbol{\beta})}{\partial \beta_j \partial \beta_k \partial \beta_l}$ for all $\boldsymbol{\beta} \in \omega$. Further there exist functions M_{jkl} such that

$$\left|\frac{\partial^3}{\partial \beta_j \partial \beta_k \partial \beta_l} \log f(\mathbf{V}, \boldsymbol{\beta})\right| \leq M_{jkl}(\mathbf{V}) \quad \text{for all} \quad \boldsymbol{\beta} \in \Omega,$$

where $m_{jkl} = E_{\boldsymbol{\beta}_0}[M_{jkl}(\mathbf{V})] < \infty$ for j, k, l.

These regularity conditions **guarantee asymptotic normality** of the ordinary MLE. See e.g. Lehmann (1983).

Let $\boldsymbol{\beta}_0$ the true value of $\boldsymbol{\beta}$. Denote

$$a_n = \max\{p'_{\lambda_n}(|\beta_{j0}|) : \beta_{j0} \neq 0\},$$
$$b_n = \max\{|p''_{\lambda_n}(|\beta_{j0}|)| : \beta_{j0} \neq 0\}$$

Theorem 3 Under the regularity conditions, if $b_n \to 0$, then there exists a local maximizer $\widehat{\boldsymbol{\beta}}$ of $Q(\boldsymbol{\beta})$ such that $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_P(n^{-1/2} + a_n)$.

- By choosing a proper λ_n, root-n consistency
- If $\lambda_n \to 0$, root-n consistency for Hard and SCAD.

$\boldsymbol{\beta}_0 = (\boldsymbol{\beta}_{10}^T, \boldsymbol{\beta}_{20}^T)^T$. WLOG, assume that $\boldsymbol{\beta}_{20} = \mathbf{0}$.

Theorem 4 (Oracle Property) Under the regularity conditions, assume that

$$\liminf_{n\to\infty} \liminf_{\theta\to 0^+} \lambda_n^{-1} p'_{\lambda_n}(\theta) > 0.$$

If $\lambda_n \to 0$ and $\sqrt{n}\lambda_n \to \infty$ as $n \to \infty$, then with probability tending to 1, the root n consistent local maximizer $\widehat{\boldsymbol{\beta}} = \begin{pmatrix} \widehat{\boldsymbol{\beta}}_1 \\ \widehat{\boldsymbol{\beta}}_2 \end{pmatrix}$ in Theorem 1 must satisfy: (i) (Sparsity) $\widehat{\boldsymbol{\beta}}_2 = \mathbf{0}$;

(ii) (Asymptotic Normality)

$$\begin{aligned} &\sqrt{n}(I_1(\boldsymbol{\beta}_{10}) + \Sigma)\left\{\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{10} + (I_1(\boldsymbol{\beta}_{10}) + \Sigma)^{-1}\mathbf{b}\right\} \\ &\to N\left\{\mathbf{0}, I_1(\boldsymbol{\beta}_{10})\right\}, \end{aligned}$$

where $I_1(\boldsymbol{\beta}_{10}) = I_1(\boldsymbol{\beta}_{10}, \mathbf{0})$, the Fisher information knowing $\boldsymbol{\beta}_2 = \mathbf{0}$, and

$$\begin{aligned} \Sigma &= \operatorname{diag}\left\{p''_{\lambda_n}(|\beta_{10}|), \cdots, p''_{\lambda_n}(|\beta_{s0}|)\right\}, \\ \mathbf{b} &= \left(p'_{\lambda_n}(|\beta_{10}|)\operatorname{sgn}(\beta_{10}), \cdots, p'_{\lambda_n}(|\beta_{s0}|)\operatorname{sgn}(\beta_{s0})\right)^T. \end{aligned}$$

- For the SCAD, if $\lambda_n \to 0$, then $a_n = 0$.

$$\Sigma \to 0 \quad \text{and} \quad \mathbf{b} \to 0$$

$$\sqrt{n}(\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{10}) \to N\{\mathbf{0}, I_1^{-1}(\boldsymbol{\beta}_{10})\}, \qquad \widehat{\boldsymbol{\beta}}_2 = \mathbf{0}.$$

Hard and SCAD have oracle property when $\sqrt{n}\lambda_n \to \infty$.
more efficient than MLE for estimating $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$.

- For L_1 penalty, $a_n = \lambda_n$.
  - Root-n consistency requires that $\lambda_n = O_P(n^{-1/2})$.
  - Oracle property requires that $\sqrt{n}\lambda_n \to \infty$.

Not be satisfied simultaneously.
Oracle property does not hold for LASSO.

2.3. Algorithms

Many authors have proposed various algorithms for PLS with nonconvex PLS or nonconcave Penalized likelihood:

Fan and Li (2001) proposed local quadratic approximation (LQA) the the penalty function

Hunter and Li (2005) analyzed the algorithm convergence behavior of LQA algorithm by using techniques of minorization-maximization (MM) algorithm, and further proposed an improved version of LQA.

With LQA to the penalty, the optimization can be carried out by Newton-Raphson type algorithm.

Coordinate descent algorithm was proposed for the PLS in Zhang and Li (2011).

L_1 penalty

Shooting algorithm (Fu, 1998, JCGS) is exactly the ICM

LARS algorithm (Efron, et al, 2004, Annals of Statistics)

Further comparison between LASSO and SCAD

Statistical properties:

$L_1 \Rightarrow$ a biased estimate for large coefficients

SCAD improves the LASSO by reducing the estimation bias

Optimization prospect:

$L_1 \Rightarrow$ the PLS function is convex

$\Rightarrow$ there exists unique global minimizer

solution by quadratic programming with linear constraints.

the whole solution path can obtained by LARS algorithm

SCAD $\Rightarrow$ the PLS function become nonconvex

$\Rightarrow$ there might exist multiple local minimizers

Improvement of LASSO and SCAD

Improvement of LASSO $\Rightarrow$ Adaptive LASSO

Improvement of SCAD/nonconvex penalties
$\Rightarrow$
one-step SCAD estimator
weighted L_1 penalties
by local linear approximation to the penalties.

Adaptive LASSO (Zou, 2006, JASA)

From Theorem 1, root n consistency of LASSO estimate requires that $\lambda_n = O(1/\sqrt{n})$

From Theorem 2, it is required that $\sqrt{n}\lambda_n \to \infty$ for the LASSO to achieve the oracle properties.

$\lambda_n = O(1/\sqrt{n})$ and $\sqrt{n}\lambda_n \to \infty$ cannot hold simultaneously.

Fan and Li (2001) conjecture that oracle property does not hold for LASSO.

Knight and Fu (2000, AOS) showed that for PLS with L_1 penalty,
(a) If $\lambda_n \to \lambda_0 > 0$, then the LASSO estimate $\widehat{\beta}$ is not consistent
(b) If If $\sqrt{n}\lambda_n \to \lambda_0 \geq 0$, then the LASSO estimate $\widehat{\beta}$ is root n consistent

Adaptive LASSO

Define
$\mathcal{A} = \{j : \beta_{j0} \neq\}$ and $\widehat{\mathcal{A}}_n = \{j : \widehat{\beta}_j \neq 0\}$

We say a variable selection procedure is consistent if and only
$\lim_n P\{\widehat{\mathcal{A}}_n = \mathcal{A}\} = 1$.

Based on results in Knight and Fu (2000), Zou (2006) showed that for the LASSO estimate
(a) if $\sqrt{n}\lambda_n \to \lambda_0 \geq 0$, then $\limsup_n P(\widehat{\mathcal{A}}_n = \mathcal{A}) \leq c < 1$, where c is a constant depending on the true model.
(b) If $\lambda_n \to 0$ and $\sqrt{n}\lambda_n \to \infty$, then

$$\lambda_n^{-1}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) \to_P \text{argmin}(V_1)$$

where

$$V_1(\mathbf{u}) = \mathbf{u}^T \mathbf{V}\mathbf{u} + \sum_{j=1}^{d}\{u_j \text{sgn}(\beta_{j0}) I(\beta_{j0} \neq 0) + |u_j| I(\beta_{j0} = 0)\}$$

Necessary Condition for consistency of LASSO estimator

Zou (2006) further showed that suppose that $\lim_n P\{\widehat{\mathcal{A}}_n = \mathcal{A}\} = 1$. Then there exists some sign vector $\mathbf{u} = (u_1, \cdots, u_s)^T$, $u_j = 1$ or -1, such that

$$|\mathbf{V}_{21}\mathbf{V}_{11}^{-1}\mathbf{u}| \leq 1$$

holds componentwisely, where $\mathbf{V} = E\mathbf{x}\mathbf{x}^T$ and

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{pmatrix}$$

Adaptive LASSO is the solution of the penalized LS with weighted L_1 penalty:

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\lambda \sum_{j=1}^{d} w_j|\beta_j|$$

where $\mathbf{w}$ is a known weights vector.

How to choose the weights?

Zou (2006) suggested setting $w_j = |\beta_{LS,j}|^{-\gamma}$ and showed that $\sqrt{n}\lambda_n \to 0$ and $\lambda_n n^{(\gamma+1)/2} \to \infty$. Then the adaptive LASSO has the oracle property.

Computation

(1) Make a transformation: $X_j^* = X_j / \widehat{w}_j$, for $j = 1, \cdots, d$

(2) Apply the LARS algorithm for

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}^*\boldsymbol{\beta}\|^2 + n\lambda \sum_{j=1}^{d} |\beta_j|$$

and denote by $\widehat{\boldsymbol{\beta}}^*$ the resulting estimate.

(3) Set $\widehat{\beta}_j = \widehat{\beta}_j^* / w_j$, for $j = 1, \cdots, d$.

Local Liner Approximation (LLA) algorithm (Zou and Li, 2008)

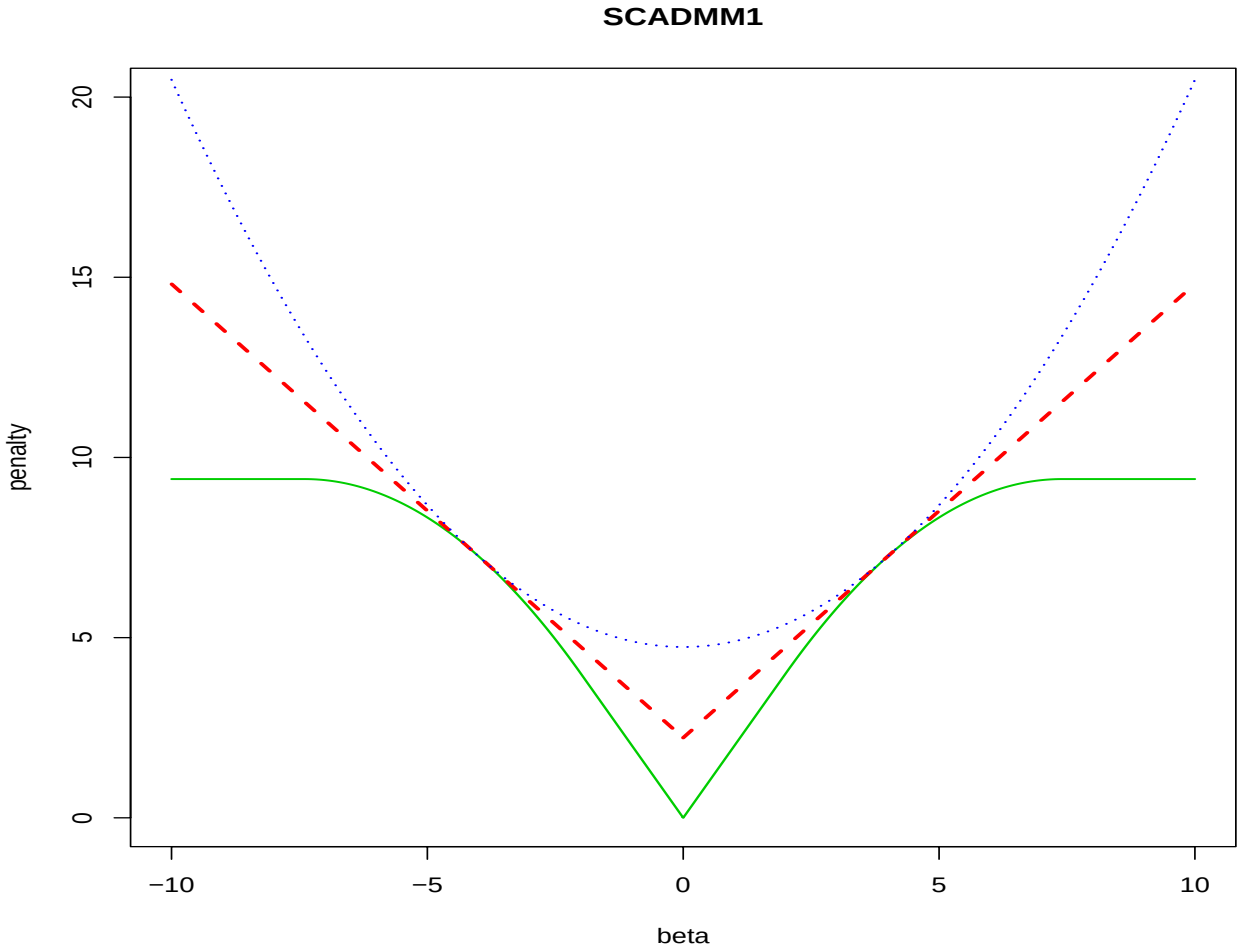
Step 1. **Initial Estimator**: unpenalized LSE

Step 2. **Local Linear approximation** (LLA):
Let $\boldsymbol{\beta}^0 = (\beta_1^0, \cdots, \beta_d^0)^T$ be the current value of each step
If β_j is close to β_j^0, approximate $p_\lambda(\beta)$ by

$$q_\lambda(\beta_j|\beta_j^0) = p_\lambda(|\beta_j^0|) + p'_\lambda(|\beta_j^0|+)(|\beta_j| - |\beta_j^0|)$$

Step 3. Apply the quadratic programming with the LLA for optimization.

Iterate between Steps 2 and 3.



LLA and LQA for SCAD penalty

LLA

Theorem 5 Under certain conditions, LLA is the best convex majorization for a nonconvex function.

See Theorem 2 of Zou and Li (2007) for more rigorous version

One-step sparse estimates

- Linear regression model

With the aid of LLA, we minimize

$$\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+n\sum_{j=1}^{d}p'_{\lambda}(|\beta_j^{(k)}|)\,|\beta_j|\,.$$

at the $k+1$-step.

One-step estimates:
(a) Set $\boldsymbol{\beta}^{(0)}$ to be the unpenalized LSE.
(b) Calculate

$$\widehat{\boldsymbol{\beta}}(\mathrm{ose})=\arg\min\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+n\sum_{j=1}^{d}p'_{\lambda}(|\beta_j^{(0)}|)\,|\beta_j|\}.$$

- Provided that $\boldsymbol{\beta}^{(0)}$ is root n consistent for $\boldsymbol{\beta}$, $\widehat{\boldsymbol{\beta}}(\mathrm{ose})$ enjoys the oracle properties (Theorem 4, Zou & Li, 2007))

- Minimize a convex function: minimizer is unique.

- Solved by the LARS algorithm as follows

Type 1

$p_\lambda(\theta) = \lambda p(\theta)$ and $p'(\theta) > 0$ for all t. Bridge penalties belong to this category which also covers many other penalties.

Algorithm 1

Step 1. Create working data by

$$X_{ij}^* = X_{ij}/p'(|\beta_j^{(0)}|), \quad \text{for} i = 1, 2, \ldots, n; \; j = 1, \ldots, p.$$

Step 2. Apply the LARS algorithm to solve

$$\widehat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^{*T} \boldsymbol{\beta})^2 + n\lambda \sum_{j=1}^{p} |\beta_j| . \right\}$$

Then it is not hard to show that

$$\beta_j^{(1)} = \widehat{\beta}_j^* / p'_\lambda(|\beta_j^{(0)}|) \quad j = 1, 2, \ldots, p.$$

Type 2

For some penalties, the derivative can be zero. In addition, the regularization parameter λ cannot be separated from the penalty function. The SCAD penalty is a typical example. Let us assume that

$$U = \{j : p'_\lambda(|\beta_j^{(0)}|) = 0\} \quad \text{and} \quad V = \{j : p'_\lambda(|\beta_j^{(0)}|) > 0\}.$$

We write

$$\mathbf{X} = [\mathbf{X}_U, \mathbf{X}_V] \quad \text{and} \quad \boldsymbol{\beta}^{(1)} = (\boldsymbol{\beta}^{(1)}{}_U^T, \boldsymbol{\beta}^{(1)}{}_V^T)^T.$$

We propose the following algorithm to compute $\boldsymbol{\beta}^{(1)}$.

Algorithm 2

Step 1a. Let $X_{ij}^* = X_{ij} \frac{\lambda}{p'_\lambda(\beta_j^{(0)})}$ for $j \in V$.

Step 1b. Let $\mathbf{H}_U$ be the projection matrix in the space of $\{\mathbf{x}_j^*, j \in U\}$. Compute $\mathbf{y}^* = \mathbf{y} - \mathbf{H}_U \mathbf{y}$ and $\mathbf{X}_V^{**} = \mathbf{X}_V^* - \mathbf{H}_U \mathbf{X}_V^*$.

Step 2. Apply the LARS algorithm to solve

$$\widehat{\beta}_V^* = \arg\min_\beta \left\{ \frac{1}{2} \|\mathbf{y}^* - \mathbf{X}_V^{**} \beta\|^2 + n\lambda \|\beta\|_1 \right\}.$$

Step 3. Compute $\widehat{\beta}_U^* = (\mathbf{X}_U^T \mathbf{X}_U)^{-1} \mathbf{X}_U^T (\mathbf{y}^* - \mathbf{X}_V^* \widehat{\beta}_V^*)$.

Then it is not hard to show that

$$\beta_U^{(1)} = \widehat{\beta}_U^* \quad \text{and} \quad \beta_j^{(1)} = \widehat{\beta}_j^* \frac{\lambda}{p'_\lambda(|\beta_j^{(0)}|)} \quad \text{for } j \in V.$$

In both algorithms the computational cost in the LARS step is the same order of computations of a single OLS fit. Thus it is very efficient to compute the one-step estimator.

It is also remarkable that if the penalty is of type 1, then the entire profile of the one-step estimator (as a function of λ) can be efficiently constructed.

For the SCAD type penalty, we still need to solve the one-step estimator for each fixed λ, for the sets U and V could change as λ varies.

K-step estimate

We may take the one-step estimate as an initial estimate, then calculate two-step estimate, ...

One-step/K-step estimator $\widehat{\boldsymbol{\beta}}$ enjoys the oracle properties (Theorem 5, Zou & Li, 2007)).

Minimize a convex function: minimizer is unique.

Solved by the LARS algorithm.

An example

Data generated from

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

$\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0)^T$,

$\epsilon \sim N(0, 1)$

$x_j \sim N(0, 1)$ and $\text{cov}(x_i, x_j) = 0.5^{|i-j|}$.

$n = 50,\ 100$

λ selected by 5-fold CV

Prediction Error (PE) and Model Error (ME):

$$\text{PE} = E^*\{Y^* - \widehat{m}(\mathbf{x}^*)\}^2 = E\text{Var}(Y^*|\mathbf{x}^*) + \text{ME}$$

where

$$\text{ME} = E^*\{\widehat{m}(\mathbf{x}^*) - m(\mathbf{x}^*)\}^2$$

Summarize in:

$$\text{MRME} = \text{median}\{\frac{\text{ME of a method}}{\text{ME of the full model}}\}$$

Purpose: Compare

- performance in terms of MRME
- sparsity in terms of average # of zero-coefficients
- accuracy of the sandwich formula for standard errors

Simulation Results for Linear Regression Models ($n = 50$)

Method	MRME	No. of Zeros		Proportion of		
		C	IC	Under-fit	Correct-fit	Over-fit
OS-SCAD	0.208	3.00	0.55	0.000	0.771	0.229
SCAD (LQA)	0.233	3.00	0.83	0.000	0.682	0.318
BS-AIC	0.660	3.00	1.84	0.000	0.195	0.805
BS-BIC	0.401	3.00	0.63	0.000	0.576	0.424

MRME: median relative of model error to the full model

C: avg # of nonzero coeff. correctly estimated to be nonzero

IC: avg # of zero coeff. incorrectly estimated to be nonzero

Simulation Results for Linear Regression Models ($n = 100$)

Method	MRME	No. of Zeros		Proportion of		
		C	IC	Under-fit	Correct-fit	Over-fit
OS-SCAD	0.234	3.00	0.55	0.000	0.784	0.216
SCAD(LQA)	0.252	3.00	0.75	0.000	0.732	0.268
BS-AIC	0.676	3.00	1.63	0.000	0.192	0.808
BS-BIC	0.337	3.00	0.32	0.000	0.728	0.272

MRME: median relative of model error to the full model

C: avg # of nonzero coeff. correctly estimated to be nonzero

IC: avg # of zero coeff. incorrectly estimated to be nonzero

LLA algorithm for penalized likelihood

Step 1. Initial Estimator: unpenalized MLE

Step 2. Local Linear approximation (LLA):
Let $\boldsymbol{\beta}^0 = (\beta_1^0, \cdots, \beta_d^0)^T$ be the current value of each step
If β_j is close to β_j^0, approximate $p_\lambda(\beta)$ by

$$q_\lambda(\beta_j|\beta_j^0) = p_\lambda(|\beta_j^0|) + p'_\lambda(|\beta_j^0|+)(|\beta_j| - |\beta_j^0|)$$

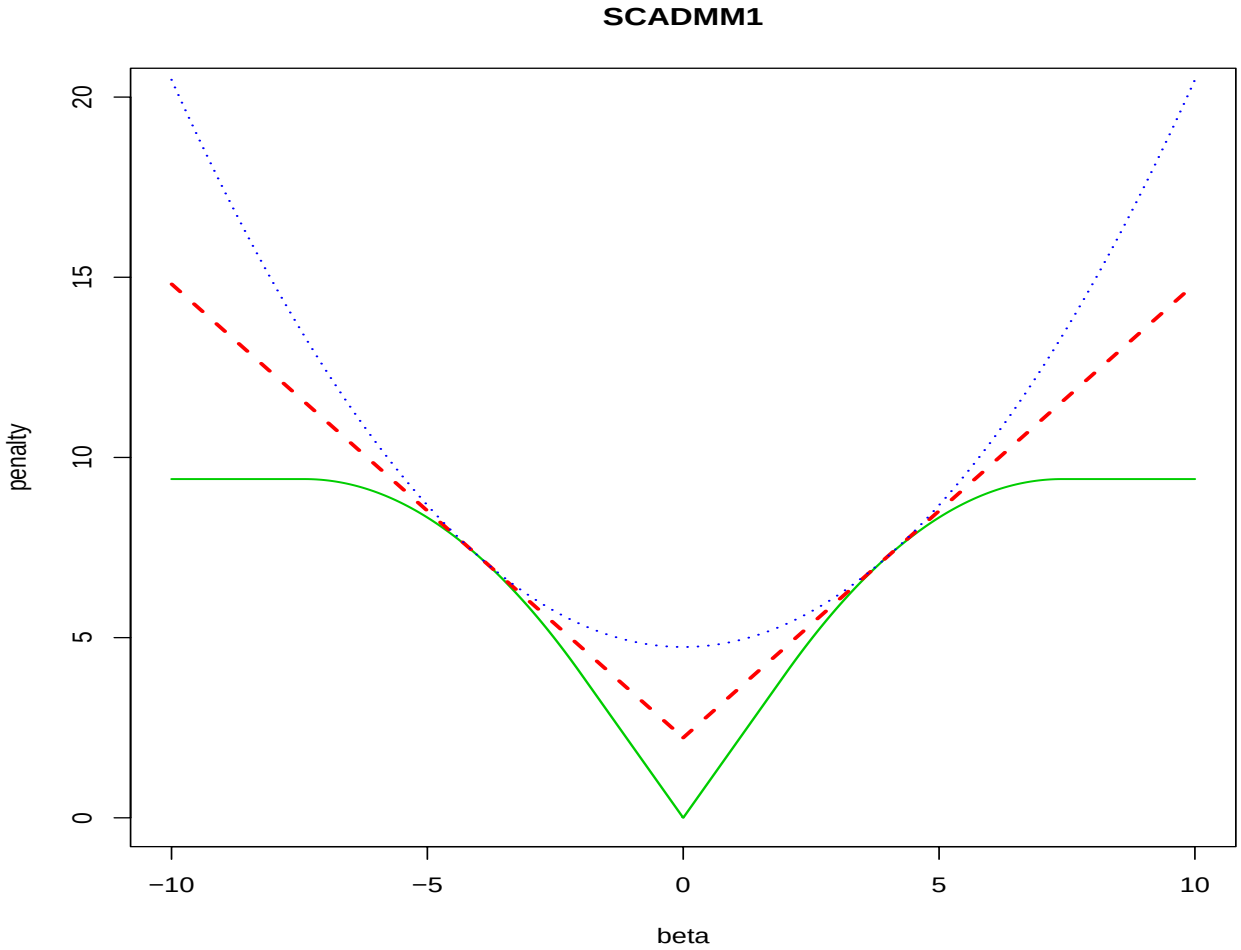
Step 3. Apply a modified linear programming with the LLA for optimization.

Iterate between Steps 2 and 3.

LLA as an MM algorithm

Theorem 6. Under certain conditions, the behavior of the LLA algorithm is the same as an EM algorithm.

Theorem 7. Under certain conditions, LLA is the best convex majorization for a nonconvex function.



LLA and LQA for SCAD penalty

One-step sparse estimates

With the aid of LLA, we minimize

$$-\ell(\boldsymbol{\beta}) + n\sum_{j=1}^{d} p'_{\lambda}(|\beta_j^{(k)}|)\,|\beta_j|\,.$$

at the $k+1$-step.

One-step estimates:
(a) Set $\boldsymbol{\beta}^{(0)}$ to be the unpenalized MLE. Approx. $\ell(\boldsymbol{\beta})$ by

$$\ell(\boldsymbol{\beta}^{(0)}) + \nabla\ell(\boldsymbol{\beta}^{(0)})^T(\boldsymbol{\beta}-\boldsymbol{\beta}^{(0)}) + \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}^{(0)})^T\nabla^2\ell(\boldsymbol{\beta}^{(0)})(\boldsymbol{\beta}-\boldsymbol{\beta}^{(0)}).$$

(b) Since $\nabla\ell(\boldsymbol{\beta}^{(0)}) = 0$ by the definition of MLE, minimize

$$\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}^{(0)})^T[-\nabla^2\ell(\boldsymbol{\beta}^{(0)})](\boldsymbol{\beta}-\boldsymbol{\beta}^{(0)}) + n\sum_{j=1}^{d} p'_{\lambda}(|\beta_j^{(0)}|)\,|\beta_j|\,.$$

to obtain $\widehat{\boldsymbol{\beta}}(\mathrm{ose})$

- $\widehat{\boldsymbol{\beta}}(\mathrm{ose})$ enjoys the oracle properties (Theorem 5, Zou & Li, 20078)).
- Minimize a convex function: minimizer is unique.
- Solved by the LARS algorithm:

Numerical Examples

Comparison criteria:

(a) Model error (ME):

$$ME\{\widehat{\mu}(\cdot)\} = E\{\widehat{\mu}(\mathbf{x}) - \mu(\mathbf{x})\}^2$$

(b) Model complexity: number of zero coefficients.

Tuning parameter selection: 5-fold CV

1000 replicates for each case.

To compare

1. One-step Logarithm penalty (OS-LOG): $p_\lambda(|\beta|) = \lambda \log(|\beta|)$
2. One-step $L_{0.01}$ (OS-$L_{0.01}$): $p_\lambda(|\beta|) = \lambda|\beta|^{0.01}$
3. SCAD:
   - One-step SCAD (OS-SCAD)
   - SCAD with LQA (Fan and Li, 2001)
   - P-SCAD: SCAD with perturbed-version of LQA (Hunter and Li, 2005)
4. Best subset selection with AIC and BIC

$$2\log(\text{likelihood}) - \lambda \sum_j I(\beta_j \neq 0),$$

AIC and BIC: $\lambda = 2$ and $\log(n)$, respectively.

Logistic regression

Data were generated from a logistic model:

$$Y|\mathbf{x} \sim \text{Bernoulli}\{p(\mathbf{x}^T\boldsymbol{\beta})\},$$

$p(u) = \exp(u)/(1+\exp(u))$.

$n = 200$

$\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0)^T$,

$x_j \sim N(0,1)$ and $\text{cov}(x_i, x_j) = 0.5^{|i-j|}$.

Simulation Results for Logistic Regression Model

Method	MRME	No. of Zeros		Proportion of		
		C	IC	Under-fit	Correct-fit	Over-fit
OS-SCAD	0.238	2.95	0.82	0.051	0.565	0.384
OS-LOG	0.229	2.97	0.61	0.029	0.518	0.453
OS-$L_{0.01}$	0.230	2.97	0.61	0.028	0.516	0.456
SCAD	0.238	2.92	0.51	0.076	0.706	0.218
P-SCAD	0.237	2.92	0.50	0.079	0.707	0.214
AIC	0.596	2.98	1.56	0.021	0.216	0.763
BIC	0.208	2.95	0.22	0.053	0.800	0.147

MRME: median relative of model error to the full model

C: avg # of nonzero coeff. correctly estimated to be nonzero

IC: avg # of zero coeff. incorrectly estimated to be nonzero

Poisson regression

Data were generated from a Poisson regression model

$$Y|\mathbf{x} \sim \text{Poisson}\{\lambda(\mathbf{x}^T\boldsymbol{\beta})\},$$

$\lambda(u) = \exp(u)$,

$\boldsymbol{\beta} = (1.2, 0.6, 0, 0, 0.8, 0, 0, 0, 0, 0, 0, 0)^T$

$x_j \sim N(0, 1)$ and $\text{cov}(x_i, x_j) = 0.5^{|i-j|}$.

$n = 60,\ 120$

Simulation Results for Poisson Regression Models ($n = 60$)

Method	MRME	No. of Zeros		Proportion of		
		C	IC	Under-fit	Correct-fit	Over-fit
OS-SCAD	0.284	2.99	1.35	0.011	0.386	0.603
OS-LOG	0.260	2.99	1.10	0.006	0.460	0.534
OS-$L_{0.01}$	0.260	2.99	1.10	0.006	0.460	0.534
SCAD	0.292	3.00	2.75	0.003	0.095	0.902
P-SCAD	0.327	2.91	1.72	0.055	0.270	0.675
AIC	0.496	3.00	1.40	0.001	0.265	0.734
BIC	0.228	3.00	0.34	0.002	0.735	0.263

MRME: median relative of model error to the full model

C: avg # of nonzero coeff. correctly estimated to be nonzero

IC: avg # of zero coeff. incorrectly estimated to be nonzero

Simulation Results for Poisson Regression Models ($n = 120$)

Method	MRME	No. of Zeros		Proportion of		
		C	IC	Under-fit	Correct-fit	Over-fit
OS-SCAD	0.271	3.00	1.00	0.001	0.552	0.447
OS-LOG	0.266	3.00	0.76	0.000	0.603	0.397
OS-$L_{0.01}$	0.266	3.00	0.77	0.000	0.601	0.399
SCAD	0.342	3.00	2.36	0.000	0.174	0.826
P-SCAD	0.356	2.95	1.60	0.037	0.322	0.641
AIC	0.594	3.00	1.45	0.000	0.235	0.765
BIC	0.277	3.00	0.25	0.000	0.790	0.210

MRME: median relative of model error to the full model

C: avg # of nonzero coeff. correctly estimated to be nonzero

IC: avg # of zero coeff. incorrectly estimated to be nonzero

2.4 Regularization parameter selection

2.4.1 Tuning parameter selectors for PLS

2.4.2 Tuning parameter selectors for penalized likelihood

Tuning parameter selectors for PLS

Naive definition of degrees of freedom $\mathrm{df}_N(\lambda)$: # of nonzero coefficients in the model

$$\widehat{\mathbf{y}} = \mathbf{X}^T\widehat{\boldsymbol{\beta}} = \mathbf{X}\{\mathbf{X}^T\mathbf{X} + n\Sigma_\lambda(\widehat{\boldsymbol{\beta}})\}^{-1}\mathbf{X}^T\mathbf{y}$$

We may define **degrees of freedom** (effective number of parameter)

$$\mathrm{df}_L(\lambda) = \mathrm{trace}\{\mathbf{X}\{\mathbf{X}^T\mathbf{X} + n\Sigma_\lambda(\widehat{\boldsymbol{\beta}})\}^{-1}\mathbf{X}^T\}$$

GCV tuning parameter selector: λ is usually selected by minimizing the GCV criterion

$$\mathrm{GCV}_\lambda = \frac{||\mathbf{y} - \mathbf{X}\widehat{\beta}_\lambda||^2}{n(1 - \mathrm{df}_\lambda/n)^2},$$

where df_λ = either $\mathrm{df}_L(\lambda)$ or $\mathrm{df}_N(\lambda)$.

Question 1: With λ selected by minimizing GCV criterion, does the SCAD possess the oracle property?

Denote $\widehat{\sigma}^2_\lambda = n^{-1}||\mathbf{y} - \mathbf{X}\widehat{\beta}_\lambda||^2$. Then

$$\mathrm{GCV}_\lambda = \frac{\widehat{\sigma}^2_\lambda}{(1 - \mathrm{df}_\lambda/n)^2}.$$

Thus,

$$\log \mathrm{GCV}_\lambda = \log \widehat{\sigma}^2_\lambda - 2\log(1 - \mathrm{df}_\lambda/n) \approx \log \widehat{\sigma}^2_\lambda + 2\mathrm{df}_\lambda/n \widehat{=} \mathrm{AIC}_\lambda.$$

Hence, $\log \mathrm{GCV}_\lambda$ is very similar to the traditional AIC.

Theorem 8 Under certain conditions, with probability tending one, the model selected by SCAD-GCV contains all significant variables, but with nonzero probability includes some non-significant variables.

BIC tuning parameter selector

Question 2: How to select λ such that the resulting SCAD estimate possess the oracle property?

Define

$$\text{BIC}_\lambda = \log \widehat{\sigma}_\lambda^2 + \text{df}_\lambda \log(n)/n.$$

Select λ by minimizing BIC_λ, and denote $\widehat{\lambda}_{\text{BIC}}$.

Theorem 9 Under certain conditions, with probability tending to one, SCAD-BIC estimate possesses the oracle property.

Numerical examples

Data are generated from

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

where $\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$,

$\epsilon \sim N(0, \sigma^2)$

$\mathbf{x} \sim N_8(0, \Sigma_x)$, and $(\Sigma_x)_{ij} = 0.5^{|i-j|}$

$n = 50, 100$ and 200.

# of simulations: 1000.

Simulation results (SCAD)

n	Method	Under-fitted(%)	Correctly fitted(%)	Overfitted(%) 1	2	≥ 3	# of zeros	MRME (%)
50	GCV	0	17.1	24.5	33.2	25.2	3.27	55.64
	BIC	0	45.6	23.9	21.1	9.4	4.04	40.97
100	GCV	0	19.0	24.5	31.8	24.7	3.32	55.91
	BIC	0	54.9	23.6	16.9	4.6	4.27	40.53
200	GCV	0	48.1	37.5	11.7	2.7	3.30	55.00
	BIC	0	81.8	16.4	1.3	0.5	4.41	38.36

ME for linear model:

$$\text{ME}(\widehat{\boldsymbol{\beta}}) = (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T \Sigma_x (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})$$

MRME, median of relative model error (ME)

Tuning parameter selector penalized likelihood

GCV: Fan and Li (2001)
Wang, Li and Tsai (2007): BIC for penalized LS with SCAD penalty.

GIC-type Tuning parameter selector

Some notation:
α: a subset of $\bar{\alpha} = \{1, ..., d\}$, as a candidate model, i.e, the corresponding predictors labelled by α are included in the model.

Accordingly, $\bar{\alpha}$ is the full model.

d_α: the size of model α

$\boldsymbol{\beta}_\alpha$: the coefficients associated with the predictors in model α.

$\mathcal{A}$: the collection of all candidate models.

Linear regression model:

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta}_\alpha + e_i \qquad e_i \sim N(0, \sigma^2)$$

for $i = 1, \cdots, n$.
The generalized information criterion (GIC, Nishii, 1984) for classical variable selections: It is

$$\mathrm{GIC}_{\kappa_n}(\alpha) = \log \widehat{\sigma}_\alpha^2 + (\kappa_n / n) d_\alpha.$$

Here
$\widehat{\sigma}_\alpha^2$: the MLE of σ^2,
κ_n: a positive number that controls the properties of variable selection.
E.g. GIC $\Rightarrow$ AIC with $\kappa_n = 2$; GIC $\Rightarrow$ BIC with $\kappa_n = \log n$.

GIC contains a broad range of selection criteria, $\Rightarrow$ develop a regularization parameter selector.

GIC tuning parameter selector:

$$\text{GIC}_{\kappa_n}(\lambda) = \frac{1}{n} G(\mathbf{y}, \widehat{\boldsymbol{\beta}}_\lambda) + (\kappa_n / n) df_\lambda,$$

where
$G(\mathbf{y}, \widehat{\boldsymbol{\beta}}_\lambda)$ measures the fitting of model α_λ

$\widehat{\boldsymbol{\beta}}_\lambda$: penalized likelihood estimator with λ

df_λ: is the df of model α_λ.

For any given κ_n, we select λ that minimizes $\text{GIC}_{\kappa_n}(\lambda)$.

Condition on $G(\mathbf{y}, \widehat{\boldsymbol{\beta}}_\lambda)$

1. For any candidate model $\alpha \in \mathcal{A}$, there exists $c_\alpha > 0$ such that $\frac{1}{n} G(\mathbf{y}, \widehat{\boldsymbol{\beta}}_\alpha) \to c_\alpha$ in probability.

2. For any underfitted model $\alpha \in \mathcal{A}$, $c_\alpha > c_{\alpha_0}$, where c_{α_0} is the convergence of $\frac{1}{n} G(\mathbf{y}, \boldsymbol{\beta}_{\alpha_0})$ and $\boldsymbol{\beta}_{\alpha_0}$ is the parameter vector of the true model α_0.

Degrees of Freedom df_λ:

1. Naive definition $df_N(\lambda)$: equal # of nonzero coefficients.
2. Trace of the approximate linear projection matrix, $df_L(\lambda)$ (Fan and Li, 2001)

Proposition: Under certain conditions, we have

$$P\left\{df_L(\lambda) = d_{\alpha_\lambda}\right\} \to 1,$$

where d_{α_λ} is the size of model α_λ.

Thus, the difference between $df_L(\lambda)$ and the size of model, $d_{\alpha_\lambda} = df_N(\lambda)$, is small.

Selection Consistency

Deviance for model α_λ: $D(\mathbf{y}; \widehat{\boldsymbol{\mu}}_\lambda) = 2\{\ell(\mathbf{y}; \mathbf{y}) - \ell(\widehat{\boldsymbol{\mu}}_\lambda; \mathbf{y})\}$,
GIC tuning parameter selector for GLIM:

$$\text{GIC}_{\kappa_n}(\lambda) = \frac{1}{n} D(\mathbf{y}; \widehat{\boldsymbol{\mu}}_\lambda) + \frac{1}{n} \kappa_n df_\lambda.$$

Theorem 10 Suppose that certain conditions hold.

(A) If there exists a positive constant M such that $\kappa_n < M$, then the model selected by penalized likelihood with λ selected by GIC-criterion contains all significant variables, but with nonzero probability includes some non-significant variables.

(B) If $\kappa_n \to \infty$ and $\kappa_n / \sqrt{n} \to 0$, then the tuning parameter λ selected by minimizing $\text{GIC}_{\kappa_n}(\lambda)$ satisfies $P\{\alpha_\lambda = \alpha_0\} \to 1$.

Theorem 10 provides guidance on the choice of the regularization parameter.

◦ Call GIC with $\kappa_n \to 2$ to be the AIC-type selector. Theorem 1(A) implies that the AIC-type selector tends to overfit without regard to which penalty function being used.

◦ Call GIC with $\kappa_n \to \infty$ and $\kappa_n/\sqrt{n} \to 0$ the BIC-type selector. Theorem 1(B) indicates that BIC-type selector enables us to identify the true model consistently. Thus, the nonconcave penalized likelihood of the generalized linear model with the BIC-type selector possesses the oracle property.

Numerical comparisons

Example 1. (Logistic Regression) Data are generated from logistic regression model, $y|\mathbf{x} \sim \text{Bernoulli}(p(\mathbf{x}^T\beta))$ with

$$p(\mathbf{x}^T\beta) = \mu(\mathbf{x}^T\beta) = \frac{\exp(\mathbf{x}^T\beta)}{1+\exp(\mathbf{x}^T\beta)},$$

where $\beta = (3, 1.5, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0,)^T$,

$\mathbf{x}_1 = (X_1, \cdots, X_9)^T \sim N_9(0, \Sigma)$, and $(\Sigma)_{ij} = 0.5^{|i-j|}$,
$\mathbf{x}_2 = (X_{10}, X_{11}, X_{12})^T$ with elements being iid Bernoulli with $p = 0.5$.

$n = 200$ and 400.
# of simulations: 1000.

Simulation results for logistic regression model

Method	MRME (%)	Under (%)	Exact (%)	Overfitted (%) 1	2	3	4	≥ 5
				n=200				
SCAD-AIC	58.02	0.2	30.2	23.2	19.6	13.6	7.5	5.9
SCAD-BIC	16.90	0.7	83.7	13.2	2.4	0.4	0.0	0.0
AIC	57.05	0.1	19.9	31.5	26.7	14.8	4.3	2.8
BIC	18.27	0.8	80.4	17.0	2.3	0.1	0.0	0.0
Oracle	12.49	0.0	100.0	0.0	0.0	0.0	0.0	0.0
				n=400				
SCAD-AIC	64.45	0.0	29.4	21.9	21.0	15.1	8.5	4.1
SCAD-BIC	19.03	0.0	89.9	7.9	1.8	0.2	0.2	0.0
AIC	63.17	0.0	21.7	29.2	27.3	14.6	5.5	1.7
BIC	20.46	0.0	86.3	12.1	1.4	0.2	0.0	0.0
Oracle	15.88	0.0	100.0	0.0	0.0	0.0	0.0	0.0

Take home message for consistency

If we assume that there exists a true model,

- AIC-type selectors are not consistent in variable selection.
- BIC-type selectors are consistent and leads to **oracle property**.

Do we always prefer BIC over AIC?

Asymptotic Loss Efficiency

- Variable selection for linear regression

Consider the following linear regression model

$$y_i = \mu_i + \epsilon_i, \text{ for } i = 1, ..., n,$$

where $\boldsymbol{\mu} = (\mu_1, ..., \mu_n)^T$ is an unknown mean vector, and ϵ_i's are i.i.d random error with mean 0 and variance σ^2. Model $\boldsymbol{\mu}$ with $\mathbf{X}\boldsymbol{\beta}$.

Note: we do not assume there exists a correct model ($\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$).

- Missing covariates/interactions, etc.;
- Approximation of the truth.

GIC tuning parameter selectors for linear regression

Penalized LS estimator:

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \sum_{j=1}^{d} p_\lambda(|\beta_j|) \right\},$$

and λ is selected by minimizing

$$\text{GIC}^*_{\kappa_n}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_\lambda)^2 + \frac{\kappa_n}{n} \sigma^2 df_\lambda.$$

L_2-Loss and Risk

Average squared loss:

$$L(\widehat{\boldsymbol{\beta}}) = \frac{1}{n}||\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}||^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\mu_i - \mathbf{x}_i^T\widehat{\boldsymbol{\beta}}\right)^2,$$

Risk under squared loss $R(\widehat{\boldsymbol{\beta}}) = E\left[L(\widehat{\boldsymbol{\beta}})\right]$.

Here $\widehat{\boldsymbol{\beta}}$ can be either original least squares estimate $\widehat{\boldsymbol{\beta}}_\alpha^*$ or the penalized estimate $\widehat{\boldsymbol{\beta}}_\lambda$.

Our goal is to find a model with a loss as small as possible.

Asymptotic loss efficiency

It is said that the penalized estimator $\widehat{\boldsymbol{\beta}}_{\widehat{\lambda}}$ corresponding to the tuning parameter $\widehat{\lambda}$ is **asymptotic loss efficient** if

$$\frac{L(\widehat{\boldsymbol{\beta}}_{\widehat{\lambda}})}{\inf_{\lambda \in [0, \lambda_{\max}]} L(\widehat{\boldsymbol{\beta}}_{\lambda})} \to 1,$$

in probability.

Theorem 11. Under certain regularity conditions, we have

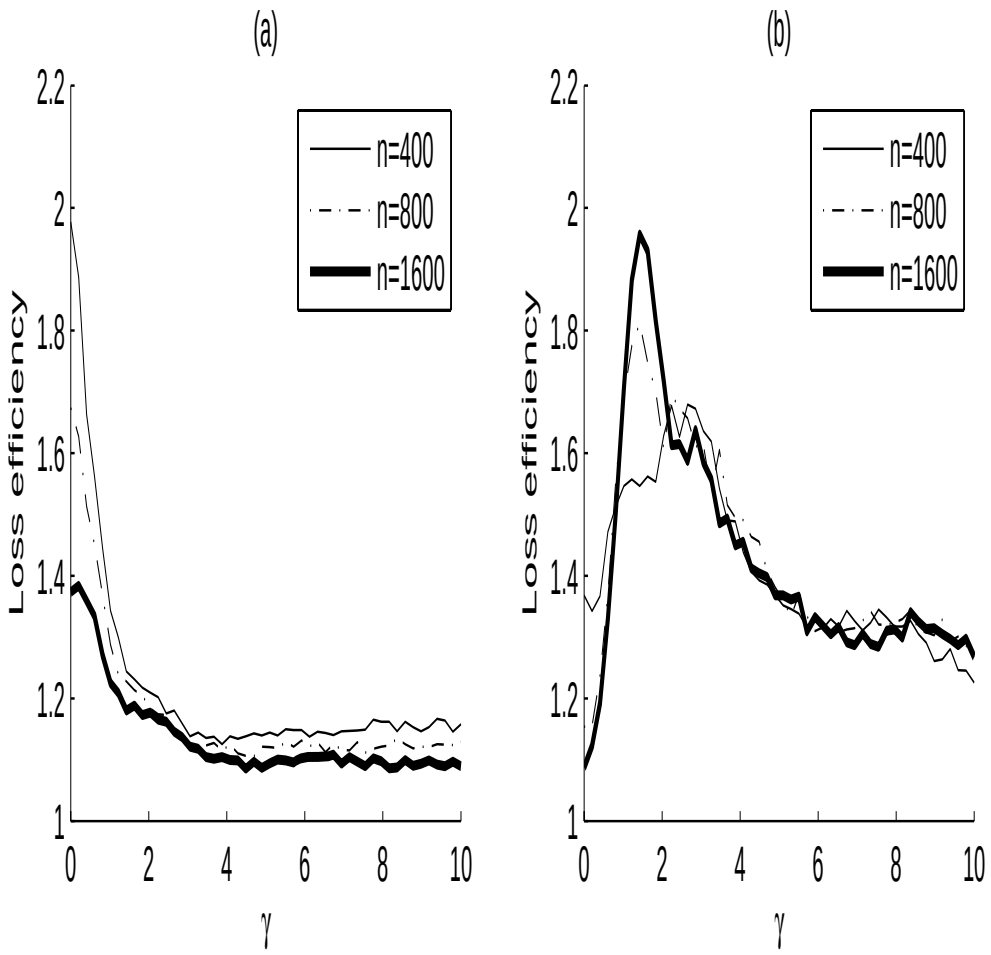
- AIC-type selector is asymptotically loss efficient.
- In general, BIC-type selector is not loss efficient.

Numerical Results

Example: Linear regression without a true model

- $\mu(\mathbf{x}) = \mathbf{x}_{full}\beta_{full} + x_{exc}\beta_{exc}$. $\mathbf{x}_{full}$ is 12 dimensional, x_{exc} is scalar.
- Only 12 predictors in $\mathbf{x}_{full}$ are used for the regression.
- $\beta_{exc} = 0.2$ and $\beta_{full} = \beta_0 + \gamma\delta/\sqrt{n}$,
  where $\beta_0 = (3, 1.5, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0)^T$,
  $\delta = (0, 0, 1.5, 1.5, 1, 1, 0, 0, 0, 0, 0.5, 0.5)^T$.
  γ: Controls β, and the sparsity of the model.
  $\gamma = 0$: many zeros in β_{full}.

The plot of Loss Efficiency



(a) AIC-selector: $\kappa_n = 2$, (b) BIC-selector: $\kappa_n = \log n$

Mammographic mass data

Follow Elter, Schulz-Wendtland and Wittenberg (2007)

$y = 0$ (benign) or $y = 1$ (malignant)

x_1-*birads*: assessment assigned in a double-review process by doctors
x_2-*age*: patient's age in years;
x_3-*mass density*
x_4 to x_6: the dummy variables to represent the four *mass shapes*
x_7 to x_{10}: the dummy variables to represent the five *mass margins*

$\lambda_{AIC} = 0.0291$ and $\lambda_{BIC} = 0.1487$.

Estimates for mammographic mass data

	MLE	SCAD-AIC	SCAD-BIC
β_0	-11.04(1.48)	-11.17(1.13)	-11.16(1.07)
BIRADS (x_1)	2.18(0.23)	2.19(0.23)	2.25(0.23)
age (x_2)	0.05(0.01)	0.04(0.01)	0.04(0.01)
density (x_3)	-0.04(0.29)	0(-)	0(-)
sRound (x_4)	-0.98(0.37)	-0.99(0.37)	-0.80(0.34)
sOval (x_5)	-1.21(0.32)	-1.22(0.32)	-1.07(0.30)
sLobular (x_6)	-0.53(0.35)	-0.54(0.34)	0(-)
mCircum (x_7)	-1.05(0.42)	-0.98(0.32)	-1.01(0.30)
mMicro (x_8)	-0.03(0.65)	0(-)	0(-)
mObscured (x_9)	-0.48(0.39)	-0.42(0.30)	0(-)
mIlldef (x_{10})	-0.09(0.33)	0(-)	0(-)

2.5 Regularization method for high-dimensional regression models

High-dimension means $d > n$ or $d >> n$.

Consider linear regression model:

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

When $d > n$, $\mathbf{X}^T\mathbf{X}$ is not invertible. Thus, LSE is not well-defined.

Sparsity assumption becomes indispensable.

Algorithms such as LQA and LLA for low-dimensional cases may not be directly applicable for HD cases directly.

LSE is the solution of the normal equation:

$$\mathbf{X}^T\mathbf{X}\beta = \mathbf{X}\mathbf{y}$$

Assume β is sparse. That is, many elements of β equal 0. For $d > n$, there are different sparse representation of β for the normal equation.

Intuitively, we want to obtain the sparest solution:

$$\min_{\beta} \|\beta\|_0 \quad \text{subject to } \mathbf{X}^T\mathbf{X}\beta = \mathbf{X}\mathbf{y}$$

Challenge: hard to get the solution
Method of frame:

$$\min_{\beta} \|\beta\|_2 \quad \text{subject to } \mathbf{X}^T\mathbf{X}\beta = \mathbf{X}\mathbf{y}$$

which leads to the ridge regression.

Dantzig Selector, Candes and Tao (2007, AOS)

Model: $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim N_n(0, \sigma^2 I_n)$.

Candes and Tao (2007) introduced the Dantzig selector, a solution of the following minimization problem:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \text{ subject to } \|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\|_\infty \leq \lambda_d \sigma \qquad \text{(DS)}$$

for some $\lambda_d > 0$. Indeed λ_d is a tuning parameter.

The minimization problem in (DS) can be solved by a linear programming. Dantzig: the father of linear programming, who passed away while the authors were finalizing this paper.

The minimization problem (DS) can be recast as a linear program (LP)

$$\min \sum_{j=1}^{d} u_j$$

subject to

$$-u_j \leq \beta_j \leq u_j \quad \text{for} \quad j = 1, \cdots, d$$

and

$$-\lambda_d \sigma \mathbf{1} \leq \mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) \leq \lambda_d \sigma \mathbf{1}$$

where the optimization variables are u, and $\mathbf{1}$ is a d-dimensional vector of ones.

Thus, computation for Dantzig selector is tractable.

Candes and Tao derived asymptotical accuracy of the Dantzig selector under uniform uncertainty principle (UUP), a strong condition, which implies that the design matrix is nearly orthogonal.

Elastic net (Zou and Hastie, 2005)

The Elastic net was proposed to improve LASSO when $n >> d$.

Zou and Hastie (2005) pointed out some limitations of LASSO when $d >> n$ and further proposed an improvement.
Naive elastic net

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \tag{A}$$

subject to $(1-\alpha)||\boldsymbol{\beta}\|_1 + \alpha\|\boldsymbol{\beta}\|^2 \leq t$ for some t, where $\|\cdot\|_1$ stands for L_1 norm.

The function $(1-\alpha)\|\boldsymbol{\beta}\|_1 + \alpha\|\boldsymbol{\beta}\|^2$ is called the elastic net penalty., which is a convex combination of the LASSO and ridge penalty. It is like a stretchable fishing net that retains 'all the big fish'.
If $\alpha = 1$, it retains all fish.
The elastic net penalty is strictly convex, while LASSO penalty is convex, but not strictly convex

Define

$$L(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|^2 \qquad \text{(B)}$$

Minimization problem (A) is equivalent to the minimization problem (B) with $\alpha = \lambda_2/(\lambda_1 + \lambda_2)$.

This allows us to
(a) Bayesian interpretation
(b) use the LAR algorithm to obtain the NEN solution.

Solution of NEN

Given data set $\mathbf{y}$ and $\mathbf{X}$, and (λ_1, λ_2), define an artificial data set $\mathbf{y}^*$ and $\mathbf{X}^*$ by

$$\mathbf{X}^*_{(n+d)\times d} = (1+\lambda_2^{-1/2}(\mathbf{X}^T, \sqrt{\lambda_2} I_d)^T$$

and

$$\mathbf{y}^*_{(n+d)\times 1} = (\mathbf{y}^T, \mathbf{0})^T$$

Let $\gamma = \lambda_1(1+\lambda_2)^{-1/2}$ and $\boldsymbol{\beta}^* = (1+\lambda_2)^{1/2}\boldsymbol{\beta}$. Then the naive elastic net can be written as

$$L(\gamma, \boldsymbol{\beta}) = L(\gamma, \boldsymbol{\beta}^*) = \|\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}^*\|^2 + \gamma\|\boldsymbol{\beta}^*\|_1$$

This is equivalent to having sample size $n+d$.

Let

$$\widehat{\boldsymbol{\beta}}^*(\gamma) = \text{argmin}_{\boldsymbol{\beta}^*} L(\gamma, \boldsymbol{\beta}^*)$$

then

$$\widehat{\boldsymbol{\beta}}(\gamma) = (1 + \lambda_2)^{-1/2} \widehat{\boldsymbol{\beta}}^*(\gamma).$$

When design matrix is orthonormal, there is a closed form the elastic net.

Elastic net

From the plot, we can see the bias of naive elastic net due to double shrinkage.

The shrinkage due to L_1 penalty can't avoid in the elastic net because we want to the sparsity

The shrinkage due to L_2 penalty can be undone.

When design matrix is orthonormal, the ridge regression estimator

$$\widehat{\beta}_R = (1 + \lambda_2)^{-1} \widehat{\beta}_{LS}$$

i.e.

$$(1 + \lambda_2) \widehat{\beta}_R = \widehat{\beta}_{LS}$$

Define the elastic net

$$\widehat{\boldsymbol{\beta}}(EN) = (1 + \lambda_2) \widehat{\boldsymbol{\beta}}(NEN)$$

A Theorem

Given data $\mathbf{y}$, $\mathbf{X}$ and (λ_1, λ_2), then the elastic net estimates $\widehat{\boldsymbol{\beta}}$ are given by

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}}\{\boldsymbol{\beta}\left(\frac{\mathbf{X}^T\mathbf{X} + \lambda_2 I}{1+\lambda_2}\right)\boldsymbol{\beta} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \lambda_1\|\boldsymbol{\beta}\|_1\}.$$

Compared with LASSO ($\lambda_2 = 0$), the elastic net is a stabilized version of the lasso.

$$\frac{\mathbf{X}^T\mathbf{X} + \lambda_2 I}{1+\lambda_2} = (1-\gamma)\widehat{\Sigma} + \gamma I$$

with $\gamma = \lambda/(1+\lambda_2)$ shrinks $\widehat{\Sigma}$ towards the identity matrix.

Two major references on asymptotical properties of HD SCAD

Fan, J. and Peng, H. (2004). On non-concave penalized likelihood with diverging number of parameters. *The Annals of Statistics*, **32**, 928-961.

Fan and Peng (2004) deals with $d_n = o(n^{\alpha})$ with $\alpha = 1/4$ for convergence rate and $\alpha = 1/5$ for oracle property.

Fan, J. and Lv, J. (2011) Non-concave penalized likelihood with NP-Dimensionality. *IEEE – Information Theory*, **57**, 5467-5484.

Fan and Lv (2011) deals with $d_n = O(\exp(n^{1-2\alpha}))$, where $0 < \alpha \leq 1/2$.

Weak oracle property

True value: $\boldsymbol{\beta}_0 = (\boldsymbol{\beta}_{10}^T, \boldsymbol{\beta}_{20}^T)^T$. $\boldsymbol{\beta}_{20} = \mathbf{0}$, $s_n = \dim(\boldsymbol{\beta}_{10})$

Estimator: $\widehat{\boldsymbol{\beta}} = (\widehat{\boldsymbol{\beta}}_1^T, \widehat{\boldsymbol{\beta}}_2^T)^T$.

Weak oracle property: with probability tending to one, it follows
(1) (Sparsity). $\widehat{\boldsymbol{\beta}}_2 = 0$
(2) (L_∞ loss). $\|\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{10}\|_\infty = O_P(n^{-\gamma}\log(n))$ if $\min_{1\le j\le s} |\beta_{j0}| > 2n^{-\gamma}\log(n)$.

Oracle property

Oracle property: with probability tending to one, it follows
(1) (Sparsity). $\widehat{\boldsymbol{\beta}}_2 = 0$
(2) (Asymptotic normality) $\sqrt{n}\mathbf{a}^T \mathrm{Cov}_a(\widehat{\boldsymbol{\beta}}_1)^{-1/2}(\widehat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{10})$.

Strong oracle property

Let $\widehat{\boldsymbol{\beta}}_{Oracle}$ be the oracle estimator.

$$P(\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{Oracle}) \to 1.$$

Two-step SCAD estimator

LLA algorithm requires a good initial value to do local approximation.

Good initial value is easy to obtain for low dimensional cases but not for HD cases.

For HD case, we start with initial value $\boldsymbol{\beta}^{(0)} = \mathbf{0}$ and apply LLA for the penalty. This leads to a LASSO estimate.
Set $\boldsymbol{\beta}^{(1)}$ to be the LASSO estimate, and apply LLA to penalty one more time, we can get two-step estimator.

Two-step sparse estimator for HD linear regression model

Let $Q(\boldsymbol{\beta}|\boldsymbol{\gamma})$ be the PLS function with LLA of the penalty at $\boldsymbol{\gamma}$.

Set the initial value $\widehat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$ and a tuning parameter $\lambda > 0$.

Two-step sparse estimator:

Step 1. Let $\widehat{\boldsymbol{\beta}}^{(1)} = \arg\min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}|\widehat{\boldsymbol{\beta}}^{(0)}, \tau\lambda)$, where $\tau \to 0$.

Step 2. Let $\widehat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} Q(\boldsymbol{\beta}|\widehat{\boldsymbol{\beta}}^{(1)}, \lambda)$.

In Step 1, a smaller tuning parameter $\tau\lambda$ is adopted to increase the estimation accuracy.
This step may be viewed as a feature screening using LASSO. We set $\tau = \lambda$ or $\tau = 1/\log(n)$ in our simulation.

Path Consistency

When we consider a sequence of tuning parameter values, we obtain a solution path $\{\widehat{\boldsymbol{\beta}}(\lambda) : \lambda > 0\}$.

We call a solution path "*path consistent*" if it contains the oracle estimator. We are able to prove that the two-step algorithm produces a consistent solution path under rather weak conditions.

Given such a solution path, a critical question is how to tune the regularization parameter λ in order to identify the oracle estimator.

The performance of a penalized regression estimator is known to heavily depend on the choice of the tuning parameter.

High-dimensional BIC

To further calibrate non-convex penalized regression, define high-dimensional BIC (HBIC):

$$\mathrm{HBIC}(\lambda) = \log(\widehat{\sigma}_\lambda^2) + |M_\lambda| \frac{\log(n)\log(p)}{n}, \tag{1}$$

where $M_\lambda = \{j : \widehat{\beta}_j(\lambda) \neq 0\}$ is the model identified by $\widehat{\beta}(\lambda)$, $|M_\lambda|$ denotes the cardinality of M_λ, and $\widehat{\sigma}_\lambda^2 = n^{-1}||\mathbf{Y} - \mathbf{X}\widehat{\beta}(\lambda)||^2$.

We select the tuning parameter

$$\widehat{\lambda} = \arg\min_{\lambda \in \Lambda_n} \mathrm{HBIC}(\lambda). \tag{2}$$

The above criterion extends the recent works of Chen and Chen (2008, Biometrika) and Kim et al. (2012, JLMR) on the high-dimensional BIC for the least squares regression to tuning parameter selection for nonconvex penalized regression.

Notation

$\boldsymbol{\beta}^*$: the true parameter, and assumed to be sparse: the majority of $\beta_j^* = 0$

$A_0 = \{j : \beta_j^* \neq 0\}$: be the index set of significant covariates

$|A_0| = q =$ the cardinality of A_0.

WLOG, assume that $\boldsymbol{\beta}^* = (\boldsymbol{\beta}_1^{*T}, \mathbf{0}^T)^T$ with $\dim(\boldsymbol{\beta}_1) = q$

The oracle estimator $\widehat{\boldsymbol{\beta}}^{(o)} = (\widehat{\boldsymbol{\beta}}_1^{(o)T}, \mathbf{0}^T)^T$, where $\widehat{\boldsymbol{\beta}}_1^{(o)}$ is the least squares estimator fitted using only the covariates whose indices are in A_0.

$d_* = \min\{|\beta_j^*| : \beta_j^* \neq 0\}$

Technical Conditions

(A1) $\exists\ C_1 > 0$ s.t. $\lambda_{\min}\left(n^{-1}\mathbf{X}_{A_0}^T\mathbf{X}_{A_0}\right) \geq C_1$.

(A2) Mean zero sub-Gaussian error: for all t and some finite positive σ,

$$E[\exp(t\epsilon_i)] \leq e^{\sigma^2 t^2/2}$$

(A3) The penalty function $p_\lambda(t)$ is assumed to be increasing and concave for $t \in [0, +\infty)$ with a continuous derivative $\dot{p}_\lambda(t)$ on $(0, +\infty)$. It admits a difference convex decomposition with $J_\lambda(\cdot)$ satisfies:
$\nabla J_\lambda(|t|) = -\lambda \text{sign}(t)$ for $|t| > a\lambda$, where $a > 1$ is a constant; and
$\nabla J_\lambda(|t|) \leq |t|$ for $|t| \leq b\lambda$, where $b \leq a$ is a positive constant.

(A4) Assume that $\lambda = o(d_*)$ and $\tau = o(1)$

Path consistency

Under Conditions (A1)—A(4) and some other conditions,
(a)

$$\begin{aligned} P(\widehat{\boldsymbol{\beta}}(\lambda) = \widehat{\boldsymbol{\beta}}^{(o)}) \quad \geq \quad & 1 - 2q \exp[-C_1 n(d_* - (a+1)\lambda)^2/(2\sigma^2)] \\ & -2(p-q)\exp[-n\lambda^2/(8\sigma^2)] \\ & -4p \exp\big[-n\tau^2\lambda^2/(8\sigma^2)\big]. \end{aligned}$$

(B) If $\log q = o(nd_*^2)$ and $\log p = o(n\tau^2\lambda^2)$, then

$$P(\widehat{\boldsymbol{\beta}}(\lambda) = \widehat{\boldsymbol{\beta}}^{(o)}) \to 1 \tag{3}$$

as $n \to \infty$.

Property of HBIC

Assume that the conditions of (B) and

$$\lim_{n\to\infty} \min_{A\not\supseteq A_0, |A|\le K_n} \{n^{-1}||(\mathbf{I}_n - \mathbf{P}_A)\mathbf{X}_{A_0}\boldsymbol{\beta}^*_{A_0}||^2\} \ge \kappa, \tag{4}$$

where $\mathbf{I}_n$ denotes the $n \times n$ identity matrix and $\mathbf{P}_A$ denotes the projection matrix onto the linear space spanned by the columns of $\mathbf{X}_A$. Let $\widehat{\lambda}$ be the regularizing parameter selected by HBIC. If $q\log(n)\log(p) = o(n)$ and $K_n^2\log(p)\log(n) = o(n)$, then

$$P(M_{\widehat{\lambda}} = A_0) \to 1, \tag{5}$$

as $n, p \to \infty$. Recall $M_\lambda = \{j : \widehat{\boldsymbol{\beta}}_j(\lambda) \neq 0\}$

Numerical comparison

Compare the following estimators:

Oracle estimator: assumes the availability of the knowledge of the true underlying model;

Lasso estimator with λ chosen by 5-fold CV tuning parameter selector

SCAD-CCCP: estimator from the original CCCP algorithm without calibration with λ chosen by 5-fold CV.

MCP estimator with $a = 1.5$ and 3. Computed using the R package PLUS with the theoretical optimal tuning parameter value $\lambda = \sigma\sqrt{(2/n)\log p}$, where the standard deviation σ **is taken** to be known.

New estimator with $C_n = \log\log n$ and $\tau = \lambda$ or $1/\log n$

Example 1

$\{y_i, \mathbf{x}_i\}$, $i = 1, \ldots, n = 100$, from

$$y_i = \mathbf{x}_i^T \beta^* + \epsilon_i, \tag{6}$$

where $\beta^* = (3, 1.5, 0, 0, 2, \mathbf{0}_{p-5}^T)^T$. $p = 3000$, and 100 simulations.

$\epsilon_i \sim N(0, \sigma^2)$ with $\sigma = 2$ and is independent of $\mathbf{x}_i \sim N(\mathbf{0}_p, \Sigma)$ with the following choices of Σ:
Case 1a: the (i, j)th entry of Σ is equal to $0.5^{|i-j|}$, $1 \leq i, j \leq p$;
Case 1b: the (i, j)th entry of Σ is equal to $0.8^{|i-j|}$, $1 \leq i, j \leq p$;
Case 1c: the (i, j)th entry of Σ equal to 1 if $i = j$ and 0.5 if $1 \leq i \neq j \leq p$

Table: Results of Example 1 (Case 1a)

method	CN	FN	TM	MSE($\widehat{\beta}$)
Oracle	3.00	0.00	1.00	0.146
Lasso	3.00	28.99	0.00	1.101
ALasso	3.00	11.47	0.01	1.327
SCAD	3.00	10.12	0.08	1.496
MCP($a = 1.5$)	2.89	0.28	0.76	0.561
MCP($a = 3$)	2.91	0.42	0.68	1.292
New($\tau = \lambda$)	2.99	0.16	0.90	0.234
New($\tau = 1/\log(n)$)	2.99	0.09	0.91	0.222

CN: average # of non-zero coefficients correctly estimated to be nonzero,
FN: average # of zero coefficients incorrectly estimated to be nonzero,
TM: the proportion of the true model bing exactly identified
New: SCAD with calibrated CCCP

Table: Results of Example 1 (Case 1b)

method	CN	FN	TM	MSE($\widehat{\beta}$)
Oracle	3.00	0.00	1.00	0.314
Lasso	3.00	20.64	0.00	1.248
ALasso	3.00	8.84	0.02	1.527
SCAD	2.99	7.42	0.17	1.598
MCP($a = 1.5$)	2.02	0.51	0.06	5.118
MCP($a = 3$)	1.99	0.60	0.02	5.437
New($\tau = \lambda$)	2.75	0.21	0.67	1.236
New($\tau = 1/\log(n)$)	2.77	0.21	0.66	1.150

Table: Results of Example 1 (Case 1c)

method	CN	FN	TM	MSE($\widehat{\beta}$)
Oracle	3.00	0.00	1.00	0.195
Lasso	2.99	28.22	0.00	2.987
ALasso	2.96	10.09	0.02	2.433
SCAD	2.96	18.09	0.01	3.428
MCP($a = 1.5$)	2.67	0.17	0.72	1.636
MCP ($a = 3$)	2.77	0.22	0.68	1.677
New($\tau = \lambda$)	2.79	0.43	0.58	1.251
New($\tau = 1/\log(n)$)	2.79	0.46	0.58	1.244

Example 2

We consider a more challenging case by modifying Example 1.

We divide the p components of $\boldsymbol{\beta}^*$ into continuous blocks of size 20.

We randomly select 10 blocks and assign each block the value $(3, 1.5, 0, 0, 2, \mathbf{0}_{15}^T)/1.5$.

The entries in other blocks are set to be zero. We consider $\sigma = 1$.
Two different cases are investigated:
Case 2a: $n = 200$ and $p = 3000$,
Case 2b: $n = 300$ and $p = 4000$.

Others are the same as those in Example 1

Table: Results of Example 2.

Case	method	CN	FN	TM	MSE
2a	Oracle	30.00	0.00	1.00	0.223
	Lasso	30.00	143.14	0.00	3.365
	ALasso	29.98	7.50	0.00	0.393
	SCAD	29.98	46.15	0.00	2.495
	MCP ($a = 3$)	29.83	0.50	0.92	0.807
	New ($\tau = \lambda$)	29.97	0.17	0.87	0.291
	New ($\tau = 1/\log(n)$)	29.99	0.20	0.89	0.247
2b	Oracle	30.00	0.00	1.00	0.137
	Lasso	30.00	133.65	0.00	1.089
	ALasso	30.00	1.32	0.29	0.165
	SCAD	30.00	21.83	0.00	0.599
	MCP ($a = 3$)	30.00	0.08	0.92	0.137
	New($\tau = \lambda$)	30.00	0.00	1.00	0.134
	New($\tau = 1/\log(n)$)	30.00	0.00	0.99	0.135

Extension to logistic regression

Logistic regression assumes that

$$\log\left\{\frac{P(y_i=1|\mathbf{x}_i)}{P(y_i=0|\mathbf{x}_i)}\right\}=\mathbf{x}_i^T\boldsymbol{\beta}. \tag{7}$$

The corresponding penalized likelihood is

$$n^{-1}\sum_{i=1}^{n}\left[-(\mathbf{x}_i^T\boldsymbol{\beta})y_i+\log\{1+\exp(\mathbf{x}_i^T\boldsymbol{\beta})\}\right]+\sum_{j=1}^{p}p_\lambda(|\beta_j|). \tag{8}$$

For binary response, the sub-Gaussian tail condition automatically holds. Can show the path consistency and sparse recovery

Example 3

Generate $\mathbf{x}_i$ as in Example 1, and the response variable y_i is generated according to (7) with $\boldsymbol{\beta}^* = (3, 1.5, 0, 0, 2, \mathbf{0}_{p-50}^T)^T$.

We consider sample size $n = 300$ and feature dimension $p = 2000$.

Furthermore, an independent test set of size 1000 is used to evaluate the misclassificaiton error.

Table: Simulation results for logistic regression ($n = 300$, $p = 2000$).

method	CN	FN	TM	Misclassification Rate
Oracle	3.00	0.00	1.00	0.116
Lasso	3.00	46.48	0.00	0.134
SCAD	2.08	4.02	0.04	0.161
ALASSO	2.02	4.58	0.00	0.188
MCP ($a = 3$)	2.96	0.56	0.54	0.128
New ($\tau = \lambda$)	2.99	0.02	0.97	0.116
New ($\tau = 1/\log(n)$)	2.99	0.00	0.99	0.116

A Real Data Example

Data were collected in a study conducted by Scheetz, et al. (2006).

Goal: To study how expression of gene TRIM32 depends on expression of other genes.

$n = 120$, $p = 1000$ or 2000 probes that largest absolute value of marginal correlation with the response.

In our analysis, we randomly partition the 120 rats into the training data set (80 rates) and testing data set (40 rats).

We use the training data set to fit the model and select the tuning parameter; and use the testing data set to evaluate the prediction performance.

We perform 1000 random partitions and report the average model sizes and the average prediction error on the testing data set for $p = 1000$ and 2000.

Table: Gene expression data analysis. The results are based on 1000 random partitions of the original data set.

p	method	ave model size	Prediction Error
1000	Lasso	31.17	0.586
	SCAD	4.81	0.827
	MCP($a = 1.5$)	11.79	0.668
	MCP($a = 3$)	7.02	0.768
	New($\tau = \lambda$)	11.29	0.709
	New($\tau = 1/\log(n)$)	8.50	0.689
2000	Lasso	32.01	0.604
	SCAD	4.57	0.850
	MCP($a = 1.5$)	11.33	0.700
	MCP($a = 3$)	6.78	0.788
	New($\tau = \lambda$)	9.87	0.736
	New($\tau = 1/\log(n)$)	7.91	0.736

LLA algorithms

Fan, Xue and Zou (2016) shows that

1. Under some technical conditions and with good initial value, LLA algorithm fins the oracle estimate after one iteration.

2. If the oracle estimator is obtained, the LLA algorithm will find the oracle estimator in next step. That is, it converges to the oracle estimator in next iteration and is a fixed point.

Key: How to find a good initial estimator that satisfies the conditions.

Algorithms how to get a local solution with desired property

1. ZHANG, T. (2010). Analysis of multi-stage convex relaxation for sparse regularization. *J. Mach. Learn. Res.*, **11**, 10811107.

2. ZHANG, T. (2011). Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Trans. Inform. Theory*, **57**, 46894708.

3. ZHANG, C.-H. and ZHANG, T. (2012). A general theory of concave regularization for highdimensional sparse estimation problems. *Statist. Sci.* **27**, 576593.

Theoretical analysis of local solutions

1. WANG, Z., LIU, H. and ZHANG, T. (2014). Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *Ann. Statist.*, **42**, 21642201.

2. LOH, P.-L. and WAINWRIGHT, M. J. (2015). Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *J. Mach. Learn. Res.*, **16**, 559616.

3. LIU, H., YAO, T., LI, R. AND YE, Y. (2017). Folded concave penalized sparse linear regression: sparsity, statistical performance, and algorithmic theory for local solutions. *Mathematical Programming, Series A*. In press.

ADMM algorithm

ADMM: Alternating Direction Method of Multipliers

$$\min\{f(\mathbf{x}) + g(\mathbf{z})\}, \text{ subject to } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}$$

Lagrangian Multiplier Method will minimize

$$L^*(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c})$$

Augmented Lagrangian Multiplier Method minimize

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2$$

ADMM consists of the iterations:

$$\mathbf{x}^{(k+1)} = \text{argmin}_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^{(k)}, \mathbf{y}^{(k)}) \quad (9)$$

$$\mathbf{z}^{(k+1)} = \text{argmin}_{\mathbf{z}} L_\rho(\mathbf{x}^{(k+1)}, \mathbf{z}, \mathbf{y}^{(k)}) \quad (10)$$

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}) \quad (11)$$

ADMM for Regularization Methods

$$\ell(\boldsymbol{\beta}) + \sum_{j=1}^{d} p_\lambda(|\beta_j|)$$

can be written as an equivalent form

$$\ell(\boldsymbol{\beta}) + \sum_{j=1}^{d} p_\lambda(|\gamma_j|) \quad \text{subject to} \quad \boldsymbol{\beta} = \boldsymbol{\gamma}$$

Thus, we will consider minimization problem:

$$\ell(\boldsymbol{\beta}) + \boldsymbol{\lambda}^T(\boldsymbol{\beta} - \boldsymbol{\gamma}) + \frac{\rho}{2}\|\boldsymbol{\beta} - \boldsymbol{\gamma}\|^2 + \sum_{j=1}^{d} p_\lambda(|\gamma_j|)$$

For PLS, $\boldsymbol{\beta}^{(k+1)}$, $\boldsymbol{\gamma}^{(k+1)}$ and $\boldsymbol{\lambda}^{(k+1)}$ have closed form.
For penalized likelihood, $\boldsymbol{\gamma}^{(k+1)}$ and $\boldsymbol{\lambda}^{(k+1)}$ have closed form, but $\boldsymbol{\beta}^{(k+1)}$ does not.

Remark

L_1 penalty and weighted L_1 penalty such as the penalty in adaptive lasso or LLA algorithm are convex penalty, regularization methods have a unique solution.
The ADMM converges to the unique minimizer.
Use the following matrix inversion lemma to save computational cost.

$$(P + \rho A^T A)^{-1} = P^{-1} - \rho P^{-1} A^T (I + \rho A P^{-1} A^T)^{-1} A P^{-1}$$

Global solutions

To derive a reasonably efficient solution scheme with a provable guarantee on global optimality.

For a symmetric matrix $\mathbf{Q}$, define a quadratic function $\mathbb{L}(\,\cdot\,):\ \mathbb{R}^d \to \mathbb{R}$,

$$\mathbb{L}(\boldsymbol{\beta}) := \frac{1}{2}\boldsymbol{\beta}^T\mathbf{Q}\boldsymbol{\beta} + \mathbf{q}^T\boldsymbol{\beta},$$

which is an abstract representation of a proper (quadratic) statistical loss function with $\mathbf{Q} \in \mathbb{R}^{d\times d}$ and $\mathbf{q} \in \mathbb{R}^d$ denoting matrices from data samples. Notice that $\mathbb{L}(\boldsymbol{\beta})$ does not have to be convex.

Define the folded concave penalized learning problem as follows:

$$\min_{\boldsymbol{\beta}\in\Lambda} \mathcal{L}(\boldsymbol{\beta}) := \mathbb{L}(\boldsymbol{\beta}) + n\sum_{j=1}^{d} p_\lambda(|\beta_j|), \tag{12}$$

Denote by $\Lambda := \{\boldsymbol{\beta} \in \mathbb{R}^d : \mathcal{A}^T\boldsymbol{\beta} \leq \mathbf{b}\}$ the feasible region defined by a set of linear constraints with $\mathcal{A} \in \mathbb{R}^{d\times m}$ and $\mathbf{b} \in \mathbb{R}^m$ for some proper m : $0 \leq m < d$.

Assume that $\mathcal{A}$ is full rank, and Λ is non-empty. Also assume that problem (12) is well defined, i.e., there exists a finite global solution to (12).

To ensure the well-defined, it suffices to assume that the statistical loss function $\mathbb{L}(\boldsymbol{\beta})$ is bounded from below on Λ.

Some Examples to illustrate the generality of (12)

(a) Penalized least squares with L_2-loss: $\mathbb{L}_2(\beta) := \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$.

(b) The ℓ_1 loss, formulated as $\mathbb{L}_1(\boldsymbol{\beta}) := \sum_{i=1}^n |y_i - \mathbf{x}_i^T\boldsymbol{\beta}| = |\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|$. In this case, we can instantiate the abstract form (12) into

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d,\ \psi\in\mathbb{R}^n} \left\{ \mathbf{1}^T\psi + n\sum_{j=1}^d p_\lambda(|\beta_j|) : -\psi \le \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \le \psi \right\},$$

where $\mathbf{1}$ denotes the all-ones vector with a proper dimension.

Penalized quantile regression

(c) The quantile loss function in a quantile regression problem, defined as

$$\mathbb{L}_\tau(\beta) := \sum_{i=1}^{n} \rho_\tau(y_i - \mathbf{x}_i^T\beta) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\beta)\{\tau - \mathbb{I}(y_i < \mathbf{x}_i^T\beta)\},$$

where, for any given $\tau \in (0,1)$, we have $\rho_\tau(u) := u\{\tau - \mathbb{I}(u < 0)\}$.

This problem with a penalty term can be written in the form of (12) as

$$\min_{\beta\in\mathbb{R}^d,\ \psi\in\mathbb{R}^n} \mathbf{1}^\top[(\mathbf{y} - \mathbf{X}\beta)\tau + \psi] + n\sum_{j=1}^{d} p_\lambda(|\beta_j|),$$

$$s.t.\ \ \psi \geq \mathbf{X}\beta - \mathbf{y};\ \ \psi \geq 0.$$

Penalized SVM

(d) The hinge loss function of a linear support vector machine classifier, which is formulated as $\mathbb{L}_{SVM} = \sum_{i=1}^{n}[1 - y_i \mathbf{x}_i^T \boldsymbol{\beta}]_+$. Here, assume that $y_i \in \{-1, \ +1\}$, the class label, for all $i = 1, \ldots, n$.

The corresponding instantiation of the abstract form (12) in this case can be written as

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^d, \ \psi \in \mathbb{R}^n} \mathbf{1}^T \psi + n \sum_{j=1}^{d} p_\lambda(|\beta_j|),$$
$$s.t. \ \ \psi_t \geq 1 - y_i \mathbf{x}_i^T \boldsymbol{\beta}; \ \ \psi_t \geq 0 \qquad \forall i = 1, \cdots, n.$$

(e) Penalized linear discriminant analysis (LDA)

Assume that samples come from two populations: $N_d(\boldsymbol{\mu}_1, \Sigma)$ and $N_d(\boldsymbol{\mu}_2, \Sigma)$. The Fisher LDA is

$$\{\mathbf{x} - 0.5(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)\}\Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Note that the direction $\mathbf{a} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ is the minimizer of

$$\frac{1}{2}\mathbf{w}^T\Sigma\mathbf{w}, \quad s.t. \quad \mathbf{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \leq 1$$

For high-dimensional LDA, one may consider penalized LDA:

$$\frac{1}{2}\mathbf{w}^T\mathbf{Q}\mathbf{w} + n\sum_{j=1}^{d} p_\lambda(|w_j|), \quad s.t. \quad \mathbf{w}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \leq 1$$

where $\mathbf{Q} = n\mathbf{S}_n$ with $\mathbf{S}_n$ being the sample covariance matrix. The corresponding minimization problem is a special case of (12)

(f) Markowitz's MVPO

Mean-Variance portfolio optimization (MVPO): To minimize risk for a given level of expected return. This implies that MVPO seeks the minimizer of

$$\mathbf{c}^T \Sigma \mathbf{c}, \quad s.t. \quad \mathbf{c}^T \boldsymbol{\mu} \geq w_0, \text{ and } \mathbf{c}^T \mathbf{1} \leq 1.$$

where $\mathbf{c} = (c_1, \cdots, c_d)^T$ is the portfolio allocation plan. For high-dimensional MVPO, we consider sparse MVPO:

$$\frac{1}{2}\mathbf{c}^T \mathbf{Q} \mathbf{c} + n \sum_{j=1}^{d} p_\lambda(|c_j|), \quad s.t. \quad \mathbf{w}^T \bar{\mathbf{x}} \geq w_0 \text{ and } \mathbf{c}^T \mathbf{1} \leq 1.$$

where $\mathbf{Q} = (n/2)\mathbf{S}_n$ with $\mathbf{S}_n$ being the sample covariance matrix. The corresponding minimization problem is a special case of (12)

(g) Penalized likelihood ratio test

For a **sparse** linear model,

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

It is of interest to test a linear hypothesis $H_0 : B\boldsymbol{\beta} = \mathbf{h}$. Penalized least squares under H_0 leads to minimizing

$$\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + n\sum_{j=1}^{d} p_\lambda(|\beta_j|), \quad s.t. \quad B\boldsymbol{\beta} = \mathbf{h}.$$

The corresponding minimization problem is a special case of (12).

Today's goal

(a) To establish a connection between folded concave penalized nonconvex learning and quadratic programming. This connection enables us to analyze the complexity of solving the problem globally.

(b) To provide a mixed integer (linear) program (MIP) reformulations, and further provide MIP-based global optimization (MIPGO) scheme to SCAD and MCP penalized nonconvex learning, and further prove that MIPGO ensures global optimality.

I will focus on the SCAD penalty, the results are applicable for the MCP penalty, but may not be applicable for other penalties.

Connection between folded concave nonconvex learning and a general quadratic program

Given $a > 1$ and $\lambda > 0$, the SCAD penalty (Fan & Li, 2001) is defined as:

$$p_\lambda(\theta) := \int_0^\theta \lambda \left\{ \mathbb{I}(t \leq \lambda) + \frac{(a\lambda - t)_+}{(a-1)\lambda} \mathbb{I}(t > \lambda) \right\} dt, \tag{13}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $(b)_+$ denotes the positive part of b.

Let $\mathcal{F}(\,\cdot\,,\,\cdot\,):\,\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ be defined as

$$\mathcal{F}(\boldsymbol{\beta},\mathbf{g})=\mathbb{L}(\boldsymbol{\beta})+n\sum_{i=1}^{d}\left\{(|\beta_i|-a\lambda)\cdot g_i+\frac{1}{2}(a-1)\cdot g_i^2\right\},$$

where $\mathbf{g}=(g_1,\cdots,g_d)$.

Proposition 1. *Let $p_\lambda(\cdot)$ be the SCAD penalty. (a) The minimization problem* (12) *is equivalent to the following program*

$$\min_{\beta\in\Lambda,\ g\in[0,\ \lambda]^d}\mathcal{F}(\boldsymbol{\beta},\mathbf{g}). \tag{14}$$

(b) The first derivative of the SCAD penalty can be rewritten as $p'_\lambda(\theta)=\text{argmin}_{g\in[0,\,\lambda]}\left\{(\theta-a\lambda)g+\frac{1}{2}(a-1)g^2\right\}$ for any $\theta\geq 0$.

Equivalence between (14) and a general quadratic program

This proposition enables us to show that program (14) is equivalent to a general quadratic program.

Corollary 1. *Program* (14) *is equivalent to*

$$\begin{aligned} \min_{\boldsymbol{\beta},\mathbf{g},\mathbf{h}\in\mathbb{R}^d} \quad & \frac{1}{2}(\boldsymbol{\beta}^T\mathbf{Q}\boldsymbol{\beta} + n(a-1)\mathbf{g}^T\mathbf{g} + 2n\mathbf{g}^T\mathbf{h}) + \mathbf{q}^T\boldsymbol{\beta} - na\lambda\mathbf{1}^T\mathbf{g} \\ s.t. \quad & \boldsymbol{\beta}\in\Lambda; \quad h_j \geq \beta_j; \quad h_j \geq -\beta_j; \quad 0 \leq g_j \leq \lambda. \end{aligned} \tag{15}$$

Similar results in Proposition 1 and Corollary 1 hold for the MCP penalty.

Complexity of Globally Solving (12)

We first introduce the concept of ϵ-approximate of global optimum that will be used Theorem 1(b). Assume that (12) has finite global optimal solutions. Denote by $\boldsymbol{\beta}^* \in \Lambda$ a finite, globally optimal solution to (12). Following Vavasis (1992), we call $\boldsymbol{\beta}^*_\epsilon$ to be an ϵ-approximate solution if there exists another feasible solution $\bar{\beta} \in \Lambda$ such that

$$\mathcal{L}(\boldsymbol{\beta}^*_\epsilon) - \mathcal{L}(\boldsymbol{\beta}^*) \leq \epsilon[\mathcal{L}(\bar{\boldsymbol{\beta}}) - \mathcal{L}(\boldsymbol{\beta}^*)] \tag{16}$$

Denote by $I_{d\times d} \in \mathbb{R}^{d\times d}$ an identity matrix, and by

$$\mathcal{H} := \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & n(a-1)I_{d\times d} & nI_{d\times d} \\ \mathbf{0} & nI_{d\times d} & \mathbf{0} \end{bmatrix} \tag{17}$$

the Hessian matrix of (15).

Complexity of Globally Solving (12)

Theorem 1.

(a) Let $1 < a < \infty$, then $\mathcal{H}$ has at least one negative eigenvalue (i.e., $\mathcal{H}$ is not positive semidefinite).

(b) Assume that (12) has finite global optimal solutions. Problem (12) admits an algorithm with complexity of $O(\lceil 3d(3d+1)/\sqrt{\epsilon}\rceil^r I)$ to attain an ϵ-approximate of global optimum, where I denotes the worst-case complexity of solving a convex quadratic program with $3d$ variables, and r is the number of negative eigenvalues of $\mathcal{H}$ for the SCAD penalty.

MIP-based Global Optimization Technique

Now we are ready to provide the proposed MIPGO, which essentially is a reformulation of nonconvex learning with the SCAD into an MIP problem. Our reformulation is inspired by Vanderbussche & Nemhauser (2005), who provided MIP to a quadratic program with box constraints.

It is well known that an MIP can be solved with provable global optimality by solution schemes such as the branch-and-bound algorithm (B&B, Marti & Reinelt, 2011). Essentially, the B&B algorithm keeps track of both a global lower bound and a global upper bound on the objective value of the global minimum.

Theoretically, the global optimal solution is achieved, once the gap between the two bounds is zero. In practice, the B&B is terminated until the two bounds are close enough. The state-of-the-art MIP solvers incorporate B&B with additional features such as local optimization and heuristics to facilitate computation.

MIPGO for Nonconvex Learning with the SCAD Penalty

Let us introduce a notation. For two d-dimensional vectors $\Phi = (\phi_i) \in \mathbb{R}^d$ and $\Delta = (\delta_i) \in \mathbb{R}^d$, a complementarity constraint $0 \leq \Phi \perp \Delta \geq 0$ means that $\phi_i \geq 0$, $\delta_i \geq 0$, and $\phi_i \delta_i = 0$ for all $i: \ 1 \leq i \leq d$.

A natural representation of this complementarity constraint is a set of logical constraints involving binary variables $\mathbf{z} = (z_i) \in \{0, \ 1\}^d$:

$$\Phi \geq 0; \ \Delta \geq 0; \ \Phi \leq \mathcal{M}\mathbf{z}; \ \Delta \leq \mathcal{M}(1 - \mathbf{z}); \ \mathbf{z} \in \{0, \ 1\}^d. \qquad (18)$$

Key Reformulation

Theorem 2. Program (15) is equivalent to a linear program with (linear) complementarity constraints (LPCC) of the following form:

$$\min \ \frac{1}{2}\mathbf{q}^T\beta - \frac{1}{2}\mathbf{b}^T\rho - \frac{1}{2}na\lambda\mathbf{1}^T\mathbf{g} - \frac{1}{2}\lambda\gamma_4^T\mathbf{1}; \quad s.t. \tag{19}$$

$$\begin{cases} \mathbf{Q}\beta + \mathbf{q} + \gamma_1 - \gamma_2 + \mathcal{A}\rho = 0; \quad n\mathbf{g} - \gamma_1 - \gamma_2 = 0 \\ n(a-1)\mathbf{g} + n\mathbf{h} - na\lambda\mathbf{1} - \gamma_3 + \gamma_4 = 0 \\ 0 \le \gamma_1 \perp \mathbf{h} - \beta \ge 0; \quad 0 \le \gamma_2 \perp \mathbf{h} + \beta \ge 0 \\ 0 \le \gamma_3 \perp \mathbf{g} \ge 0; \quad 0 \le \gamma_4 \perp \lambda - \mathbf{g} \ge 0 \\ 0 \le \rho \perp \mathbf{b} - \mathcal{A}^T\beta \ge 0 \\ \beta, \mathbf{g}, \mathbf{h}, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{R}^d; \quad \rho \in \mathbb{R}^m. \end{cases} \tag{20}$$

This theorem gives the key reformulation that will lead to the MIP reformulation.

Rewriting constraints in (20) into the system of logical constraints following (18), Problem (15) now becomes

$$\min \frac{1}{2}\mathbf{q}^T\boldsymbol{\beta} - \frac{1}{2}\mathbf{b}^T\rho - \frac{1}{2}na\lambda\mathbf{1}^T\mathbf{g} - \frac{1}{2}\lambda\gamma_4^T\mathbf{1}; \quad s.t. \tag{21}$$

$$\begin{cases} \mathbf{Q}\boldsymbol{\beta} + \mathbf{q} + \gamma_1 - \gamma_2 + \mathcal{A}\rho = 0; \quad n\mathbf{g} - \gamma_1 - \gamma_2 = 0 \\ n(a-1)\mathbf{g} + n\mathbf{h} - na\lambda\mathbf{1} - \gamma_3 + \gamma_4 = 0 \\ \gamma_1 \leq \mathcal{M}\mathbf{z}_1; \quad h_i - \beta_i \leq \mathcal{M}(1-\mathbf{z}_1); \gamma_2 \leq \mathcal{M}\mathbf{z}_2; \\ \quad \mathbf{h} + \boldsymbol{\beta} \leq \mathcal{M}(1-\mathbf{z}_2); \gamma_3 \leq \mathcal{M}\mathbf{z}_3; \\ \quad g_i \leq \mathcal{M}(1-\mathbf{z}_3), \gamma_4 \leq \mathcal{M}\mathbf{z}_4; \quad \lambda - g \leq \mathcal{M}(1-\mathbf{z}_4) \\ \rho \leq \mathcal{M}\mathbf{z}_5; \quad \mathbf{b} - \mathcal{A}^T\boldsymbol{\beta} \leq \mathcal{M}(1-\mathbf{z}_5) \\ -\boldsymbol{\beta} \leq \mathbf{h}; \quad \gamma_2 \geq 0; \quad \boldsymbol{\beta} \leq \mathbf{h}; \quad \gamma_1 \geq 0 \\ \mathbf{g} \geq 0; \quad \gamma_3 \geq 0; \quad \mathbf{g} \leq \lambda; \quad \gamma_4 \geq 0, \rho \geq 0; \quad \mathbf{b} - \mathcal{A}^T\boldsymbol{\beta} \geq 0 \\ \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4 \in \{0,1\}^d; \quad \mathbf{z}_5 \in \{0,1\}^m \\ \boldsymbol{\beta}, \mathbf{g}, \mathbf{h}, \gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{R}^d; \quad \rho \in \mathbb{R}^m, \end{cases} \tag{22}$$

where we recall that $\mathcal{M}$ is a properly large constant.

The above program is in the form of an MIP, which admits finite algorithms that ascertain global optimality.

Theorem 3. Program (21)-(22) admits algorithms that attain a global optimum in finite iterations.

Numerical Stability of MIPGO

The representations of SCAD nonconvex learning problems as MIPs introduce dummy variables to the original problem. These dummy variables are in fact Lagrangian multipliers in the KKT conditions of (15). In cases when no finite Lagrangian multipliers exist, the proposed MIPGO can result in numerical instability. To address this issue, we also study an abstract form of SCAD penalized the nonconvex learning problems.

Key message: The Lagrangian multipliers corresponding to a global optimal solution cannot be arbitrarily large under proper assumptions. Hence, we may conclude that the proposed method can be numerically stable.

Comparison with Gradient Methods

Wang, Liu and Zhang (2014) and Loh and Wainwright (2015) independently developed two gradient methods (Nesterov, 2007) for PLS:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathcal{L}(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + n\sum_{j=1}^{d} p_\lambda(|\beta_j|). \tag{23}$$

We will refer to this problem as SCAD or MCP penalized linear regression (LR-SCAD or LR-MCP) when $p_\lambda(\cdot)$ is the SCAD penalty.

To ensure high computational and statistical performance, both Wang, Liu and Zhang (2014) and Loh and Wainwright (2015) considered conditions called "**restricted strong convexity**" (RSC).
To illustrate that RSC can be a fairly strong condition in LR-SCAD or -MCP problems and that MIPGO may potentially outperform the gradient methods regardless of whether the RSC is satisfied.

Necessary Condition for RSC

Lemma 1. If $k \geq 1$, $\frac{64k\tau \log d}{n} + \mu \leq \alpha$, and $\mu\|\beta\|_2^2 + \sum_{i=1}^p P_\lambda(|\beta_i|)$ is convex, the following condition (24) is a necessary condition of RSC imposed in either Wang, Liu and Zhang (2014) or Loh and Wainwright (2015):

$$\frac{1}{n}\mathcal{L}(\boldsymbol{\beta}') - \frac{1}{n}\mathcal{L}(\boldsymbol{\beta}'') \geq \left\langle \frac{1}{n}\nabla\mathcal{L}(\boldsymbol{\beta}''),\ \boldsymbol{\beta}' - \boldsymbol{\beta}'' \right\rangle + \alpha_3\|\boldsymbol{\beta}' - \boldsymbol{\beta}''\|_2^2,\ \forall\|\boldsymbol{\beta}' - \boldsymbol{\beta}''\|_0 \leq s, \quad (24)$$

for some $\alpha_3 > 0$, where $s = 64k - 1$, $\nabla\mathcal{L}(\boldsymbol{\beta}'') \in \left[n\nabla\tilde{L}(\boldsymbol{\beta}) + n\lambda\partial|\boldsymbol{\beta}|\right]_{\boldsymbol{\beta}=\boldsymbol{\beta}''}$, and $\partial|\boldsymbol{\beta}|$ denotes the subdifferential of $|\boldsymbol{\beta}|$.

Numerical illustration how restrictive the RSC

We simulated a sequence of samples $\{(\mathbf{x}_i, y_i) : 1 \leq i \leq n\}$ randomly from the following sparse linear regression model:

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i,$$

where $d = 100$, $\boldsymbol{\beta} = [1;\ 1;\ \mathbf{0}_{d-2}]$, $\epsilon_i \sim N(0, 0.09)$ and $\mathbf{x}_i \sim \mathcal{N}_d(0, \Sigma)$. with covariance matrix $\Sigma = (\rho^{|i-j|}) \in \mathbb{R}^{d \times d}$.

This numerical test considers only SCAD for an example. We set the parameters for the SCAD penalty as $a = 3.7$ and $\lambda = 0.2$. The necessary condition (24) in this example can be further simplified: When $\boldsymbol{\beta}^1 \neq \boldsymbol{\beta}^2$,

$$\frac{\mathcal{L}(\boldsymbol{\beta}^1) + \mathcal{L}(\boldsymbol{\beta}^2)}{2} > \mathcal{L}\left(\frac{\boldsymbol{\beta}^1 + \boldsymbol{\beta}^2}{2}\right). \tag{25}$$

Table 1: Percentage for successfully passing the random RSC test out of 100 randomly generated instances.

ρ	$n = 35$	$n = 30$	$n = 25$	$n = 20$
0.1	93%	81%	53%	4%
0.3	94%	76%	39%	9%
0.5	55%	50%	21%	1%

We observe from Table 166 that in some cases the percentage for passing the random RSC test is noticeably low.

We compare MIPGO with both gradient methods in two sets of the sample instances from the table: (i) the one that seems to provide the most advantageous problem properties ($\rho = 0.1$, and $n = 35$) to the gradient methods; and (ii) the one with probably the most adversarial parameters ($\rho = 0.5$, and $n = 20$) to the gradient methods.

More specifically, we use the following criteria for our comparison:

- Absolute deviation (AD), defined as the ℓ_1-distance between the computed solution and the true parameter vector.
- False positive (FP), defined as the number of entries in the computed solution that are wrongly selected as nonzero dimensions.
- False negative (FN), defined as the number of entries in the computed solution that are wrongly selected as zero dimensions.
- objective gap ("Gap"), defined as the difference between the objective value of the computed solution and the objective value of the MIPGO solution.
- Computational time ("Time"), which measures the total computational time to generate the solution.

AD, FP, and FN are commonly used statistical criteria, and "Gap" is a natural measure of optimization performance. A positive Gap value indicates a worse relative performance compared to MIPGO.

Table 2 presents our comparison results in terms of computational, statistical and optimization measures.

In Table 2, we report the average values for all the above criteria out of 100 randomly generated instances aforementioned.

Table 2: Comparison between MIPGO and the gradient methods.

Method	AD	FP	FN	Gap	Time
	$\rho = 0.5,\ n = 20$				
MIPGO	0.188	0.230	0	0	29.046
	(0.016)	(0.042)	(0)	(0)	(5.216)
Loh and Wainwright	2.000	0	2	25.828	0.002
(2015)	(0.000)	(0)	(0)	(0.989)	(0.001)
Wang, Liu and Zhang	0.847	5.970	0	1.542	0.504
(2014)	(0.055)	(0.436)	(0)	(0.119)	(0.042)
	$\rho = 0.1,\ n = 35$				
MIPGO	0.085	0.020	0	0	27.029
	(0.005)	(0.141)	(0)	(0)	(4.673)
Loh and Wainwright	2.000	0	2	31.288	0.002
(2015)	(0.000)	(0)	(0)	(1.011)	(0.000)
Wang, Liu and Zhang	0.936	6.000	0	4.179	0.524
(2014)	(0.044)	(0.348)	(0)	(0.170)	(0.020)

From this table, we observe an outperformance of MIPGO over the other solution schemes on solution quality for both statistical and optimization criteria. However, MIPGO generates a higher computational overhead than the gradient methods.

Comparison with Local Linear Approximation (LLA) algorithm

Zou & Li (2008), Huang & Zhang (2012, JMLR), Wang, Kim & Li (2013, AOS) and Fan, Xue & Zou (2014, AOS) proposed LLA for SCAD-PLS.

To make a comparison, we randomly generate $\boldsymbol{\beta} \in \mathbb{R}^2$ with a uniformly distributed random vector on $[-1, 5]^2$ and then generate 2 observations $\mathbf{x}_i \sim N_2(0, \Sigma)$, with covariance matrix $\Sigma = (0.5^{|i-j|})$. Finally, we compute y_i as $y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i$ with $\varepsilon_i \sim N(0, 1)$.

Despite their small dimensionality, these problems are nonconvex with multiple local solutions. Their nonconvexity can be visualized via the 2-D and 3-D contour plots provided in Figure 1(a)-(b) (LR-SCAD).

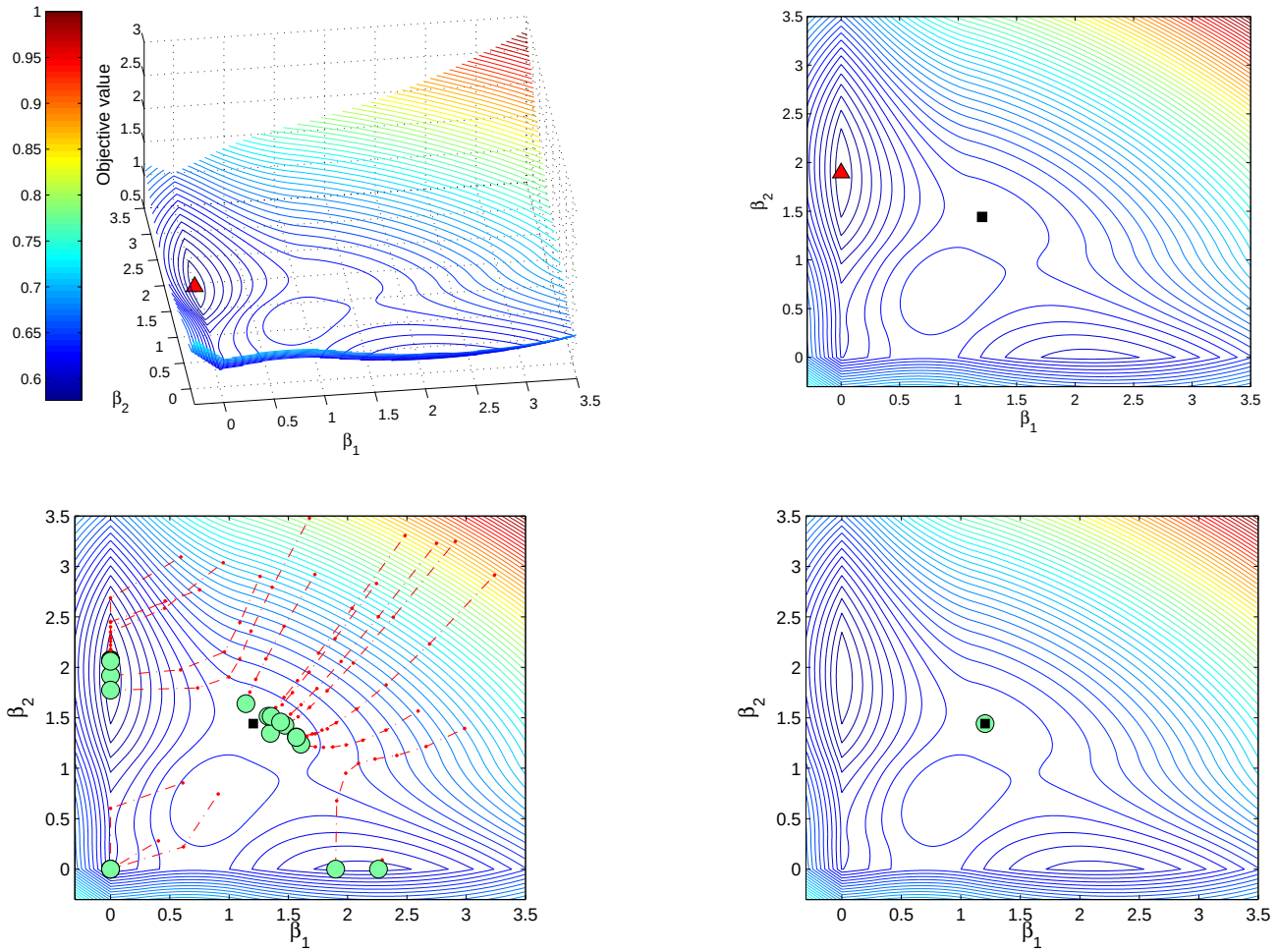


Figure 1: (a) 3-D contour plots of the 2-dimension LR-SCAD problem and the solution generated by MIPGO in 20 runs with random initial value. The triangle is the MIPGO solution in both subplots. (b) 2-D representation of subplot (a). (c) Trajectories of 20 runs of LLA with random initial value. (d) Trajectories of 20 runs of LLA with the LSE as the initial value.

Numerical Comparison

We now examine MIPGO on the statistical performance in comparison with several existing local algorithms, including coordinate descent, LLA, and gradient methods.

We simulate the random samples $\{(x_i, y_i),\ i = 1, \ldots, n\}$ from the following linear model: $y_i = x_i^T \beta + \varepsilon_i$.
Set $d = 1000$, $n = 100$. β is constructed by randomly choosing 5 elements among dimensions $\{1, \ldots, d\}$ to be 1.5, and setting the other $d - 5$ elements as zeros.
$\varepsilon_i \sim N(0, 1.2^2)$ and $x_i \sim N_d(0, \Sigma)$ with $\Sigma = (0.5^{|i-j|})$
For both LR-SCAD and LR-MCP, we set the parameter $a = 2$, and tune λ the same way as presented by Fan, Xue & Zou (2014).

We generate 100 instances using the above procedures, and solve each of these instances using MIPGO and other solutions schemes, including: (i) coordinate descent; (ii) gradient methods; (iii) SCAD-based and MCP-based LLA; and (iv) the LASSO method.

Details about methods to be compared:

LASSO: The LASSO penalized linear regression, coded in MatLab that invokes Gurobi 6.0 using CVX as the interface.

GM_1-SCAD/MCP: The SCAD/MCP penalized linear regression computed by the local solution method by Loh & Wainwright (2015) on MatLab.

GM_2-SCAD/MCP: The SCAD/MCP penalized linear regression computed by the approximate path following algorithm by Wang, Liu & Zhang (2014) on MatLab.

SparseNet: The R-package *sparsenet* for SCAD/MCP penalized linear regression computed by coordinate descent (Mazumder, et al., 2011).

Ncvreg-SCAD/-MCP: The R-package *ncvreg* for MCP penalized linear regression computed by coordinate descent (Breheny & Huang 2011).

SCAD-LLA_f/MCP-LLA_f: The SCAD/MCP penalized linear regression computed by (fully convergent) LLA with the tuned LASSO estimator as its initial solution, following Fan, Xue & Zou (2014).

Table 3a: Comparison of statistical performance.

Method	$n = 100,\ d = 1000$			
	AD	FP	FN	Time
LASSO	2.558	5.700	0	2.332
	(0.047)	(0.255)	(0)	(0.108)
GM_1-SCAD	0.526	0.600	0	4.167
	(0.017)	(0.084)	(0)	(0.254)
GM_2-SCAD	3.816	18.360	0	3.968
	(0.104)	(0.655)	(0)	(0.049)
SparseNet	1.012	5.850	0	2.154
	(0.086)	(1.187)	(0)	(0.017)
Ncvreg-SCAD	1.068	9.220	0	0.733
	(0.061)	(0.979)	(0)	(0.007)
SCAD-LLA_f	0.526	0.600	0	31.801
	(0.017)	(0.084)	(0)	(1.533)
MIPGO-SCAD	0.509	0	0	472.673
	(0.017)	(0)	(0)	(97.982)
Oracle	0.509			
	(0.017)			

Table 3b: Comparison of statistical performance.

Method	$n = 100,\ d = 1000$			
	AD	FP	FN	Time
LASSO	2.558	5.700	0	2.332
	(0.047)	(0.255)	(0)	(0.108)
GM_1-MCP	0.543	0.540	0	4.42
	(0.018)	(0.073)	(0)	(0.874)
GM_2-MCP	0.548	0.610	0	3.916
	(0.019)	(0.083)	(0)	(0.143)
SparseNet	1.012	5.850	0	2.154
	(0.086)	(1.187)	(0)	(0.017)
Ncvreg-MCP	0.830	3.200	0	0.877
	(0.045)	(0.375)	(0)	(0.009)
MCP-LLA_f	0.543	0.540	0	28.695
	(0.018)	(0.073)	(0)	(1.473)
MIPGO-MCP	0.509	0	0	361.460
	(0.017)	(0)	(0)	(70.683)
Oracle	0.509			
	(0.017)			

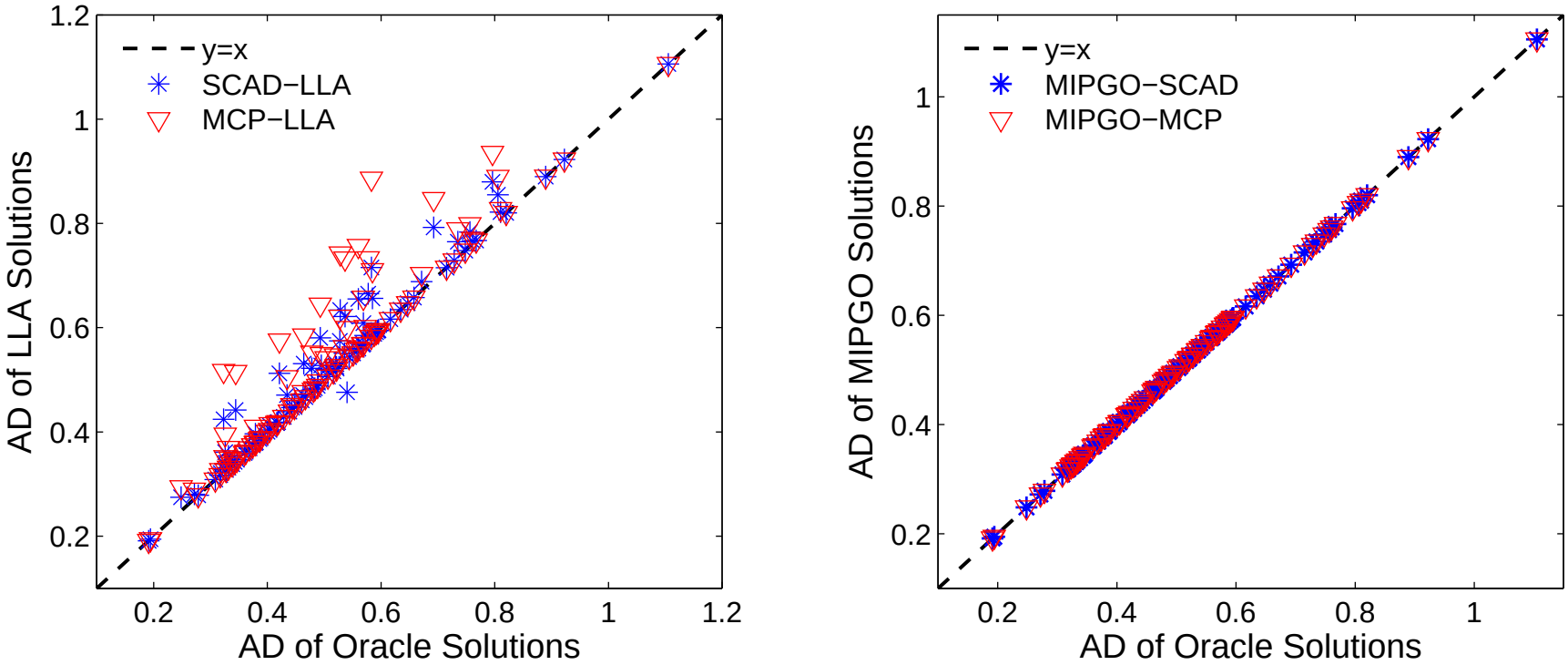


Figure 2: Comparison between generated solutions and the oracle solutions in AD when (a) solutions are generated by LLA_f, and (b) solutions are generated by MIPGO.

An application

A real data set collected in a marketing study (Wang, 2009), which has a total of $n = 463$ daily records. For each record, the response variable is the number of customers and the originally 6397 predictors are sales volumes of products.

To facilitate computation, we employ the feature screening scheme in Li, Zhong and Zhu (2012) to reduce the dimension to 1500.

Table 4: Results of the Real Data Example. "NZ" and $\#_\alpha$ stand for the numbers of parameters that are nonzero, that has a p-value greater or equal to α. "AIC", "BIC" and "Obj." stand for AIC, BIC and objective function value, respectively.

Method	SCAD: $\lambda = 0.02$; $a = 3.7$						
	NZ	$\#_{0.05}$	$\#_{0.10}$	R^2	AIC	BIC	Obj.
GM_1	1500	—	—	0.997	357.101	6563.691	212.279
GM_2	401	401	401	0.698	246.886	1906.114	103.626
LLA_0	185	119	115	0.864	-554.093	211.387	76.031
LLA_1	181	83	80	0.912	-763.718	14.789	71.673
MIPGO	129	35	34	0.898	-796.581	-262.814	68.474

Remarks

We provide a reformulation of the nonconvex learning problem into a general quadratic program. This reformulation leads to findings:

(a) To formally state the complexity of finding the global optimal solution to the nonconvex learning with the SCAD/MCP penalties.

(b) To derive a MIP-based global optimization approach, MIPGO, to solve the SCAD/MCP penalized nonconvex learning problems with theoretical guarantee. Numerical results indicate that the proposed MIPGO outperforms the gradient methods and LLA approach with different initialization schemes in solution quality and statistical performance.

The complexity bound of solving the nonconvex learning with the MCP and SCAD penalties globally has not been reported in literature and MIPGO is the first optimization scheme with provable guarantee on global optimality for solving the SCAD/MCP-penalized learning problem.

Algorithms for Big Data

1. Algorithm for huge-size data: n is huge
By huge size, it means that one computer cannot restore all data.
Need to split sample

2. Algorithm for huge-dimensional data
By huge dimension, it means that computer memory cannot read all variables in simultaneously.
Need to split variables/features

Sample splitting

Divide and conquer:

Split data $\mathcal{D}$ into K disjointed subsets: $\mathcal{D}_1, \cdots, \mathcal{D}_K$. (Typically equal sizes)

Apply statistical procedures to each subset, and obtain estimate $\tilde{\boldsymbol{\theta}}_1, \cdots, \tilde{\boldsymbol{\theta}}_K$.

Integrate all estimates from each subset together, for example, $\widehat{\boldsymbol{\theta}} = \frac{1}{K} \sum_{j=1}^{K} \tilde{\boldsymbol{\theta}}_j$.

ADMM for sample splitting

Divide and conquer:

$$\tilde{\boldsymbol{\theta}}_k = \min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}|\mathcal{D}_k) = \min_{\boldsymbol{\theta}_k} \ell(\boldsymbol{\theta}_k|\mathcal{D}_k)$$

where $\ell(\cdot)$ is a criterion function, such as LS function, negative log-likelihood function.

Essentially, we want to minimize $\sum_{k=1}^{K} \ell(\boldsymbol{\theta}|\mathcal{D}_k)$.

ADMM for sample splitting formulates this problem as

$$\min_{\boldsymbol{\theta}_k's} \sum_{k=1}^{K} \ell(\boldsymbol{\theta}_k|\mathcal{D}_k) \quad \text{subject to} \quad \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \cdots = \boldsymbol{\theta}_K$$

Application to regularization methods

We want to minimize

$$\frac{1}{n}\sum_{k=1}^{K}\ell(\boldsymbol{\beta}|\mathcal{D}_k)+r(\boldsymbol{\beta}),$$

where $r(\boldsymbol{\beta})=\sum_{j=1}^{d}p_\lambda(|\beta_j|)$.

Sample splitting strategy is to minimize

$$\frac{1}{n}\sum_{k=1}^{K}\ell(\boldsymbol{\beta}_k|\mathcal{D}_k)+r(\boldsymbol{\gamma}) \quad \text{subject to} \quad \boldsymbol{\beta}_k=\boldsymbol{\gamma}, k=1,\cdots,K$$

with respect to $\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_K$ and $\boldsymbol{\gamma}$.

This can be solved by a (K+1)-block ADMM algorithm. Direct application of (K+1)-block ADMM algorithm may not converge, and some modification will need.

Feature splitting

Objective function: $\frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{x}_i^T\boldsymbol{\beta})+\sum_{j=1}^{d}p_\lambda(|\beta_j|)$.
Splitting feature $\mathbf{x}_i$ and regression coefficient vector $\boldsymbol{\beta}$ as $\mathbf{x}_i=(\mathbf{x}_{i1}^T,\cdots,\mathbf{x}_{iM}^T)^T$ and $\boldsymbol{\beta}=(\boldsymbol{\beta}_1^T,\cdots,\boldsymbol{\beta}_M^T)^T$.

Write $\sum_{j=1}^{d}p_\lambda(|\beta_j|)=\sum_{m=1}^{M}r(\boldsymbol{\beta}_m)$. ADMM algorithm for feature splitting is to minimize

$$\sum_{m=1}^{M}r(\boldsymbol{\beta}_m)+\frac{1}{n}\sum_{i=1}^{n}\ell(\sum_{m=1}^{M}z_{im})$$

subject to

$$\mathbf{x}_{im}^T\boldsymbol{\beta}_m=z_{im},i=1,\cdots,n,m=1,\cdots,M,$$

which can be written as $\mathbf{X}_m^T\boldsymbol{\beta}_m=\mathbf{z}_m,m=1,\cdots,M$, where $\mathbf{X}=[\mathbf{X}_1,\cdots,\mathbf{X}_M]^T$, and $\mathbf{z}_m=(z_{1m},\cdots,z_{nm})^T$.