llagcdnet: (Adaptive) Elastic Net and Folded Concave Penalized Estimations using (LLA-)GCD Algorithm

– A Generalization of Package gcdnet

He Zhou

University of Minnesota

April 29, 2021

Table of Contents

Introduction to Package gcdnet

- Package `gcdnet` is the implementation of the generalized coordinate descent (GCD) algorithm proposed by Yang and Zou [2013] in their paper

  "An Efficient Algorithm for Computing the HHSVM and its Generalizations."

- It was designed originally to solve the elastic net penalized hybrid Huberized support vector machine (HHSVM);
- It was generalized to solve (adaptive) elastic net penalized least squares, logistic regression, HHSVM, squared hinge SVM, and expectile regression.

Table of Contents

Package `llagcdnet`: Contributions

- Use the GCD algorithm to solve the solution path of (adaptive) elastic net penalized probit regression, coded in `Fortran`;
- Use the LLA-GCD algorithm to solve the folded concave (SCAD) penalized estimations including least squares, logistic regression, probit regression, HHSVM, etc.

Table of Contents

Probit Regression as Large Margin Classifier

Given $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$ denote class labels.

The probit regression model assumes that

$$\mathbb{P}(Y_i = 1|\mathbf{x}_i) = \Phi(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}) \quad \text{and} \quad \mathbb{P}(Y_i = -1|\mathbf{x}_i) = \Phi(-(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})),$$

where $\Phi(\cdot)$ is the CDF of $N(0,1)$. Equivalently,

$$\mathbb{P}(Y_i = y_i|\mathbf{x}_i) = \Phi(\underbrace{y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})}_{\text{margin}}).$$

The negative log-likelihood function (scaled by n) becomes

$$\ell_n(\boldsymbol{\beta}) := \frac{1}{n}\sum_{i=1}^n \underbrace{-\log(\Phi(t_i))}_{L(t_i)},$$

where $t_i = y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})$ is the *margin* of the i-th pair of data.

Remark: Other large margin classifiers: SVM, HHSVM, logistic regression ($L(t) = \log(1 + e^{-1})$), etc.

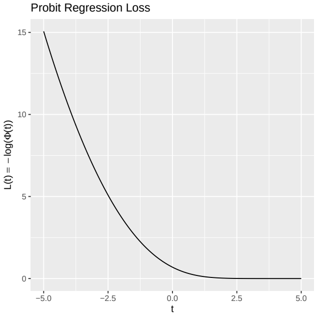
Probit Loss Function

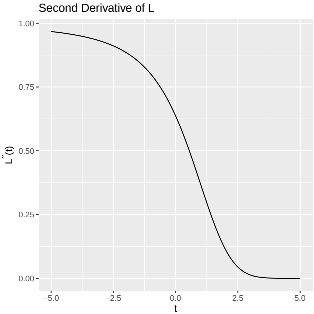
The probit regression loss function $L(t) = -\log(\Phi(t))$ has the following property:

Lemma (Bounded Second Derivative of Probit Loss)

The second derivative of the probit loss function satisfies

$$L''(t) = \frac{t\varphi(t)}{\Phi(t)} + \left(\frac{\varphi(t)}{\Phi(t)}\right)^2 \in [0,1]$$





Algorithm: Generalized Coordinate Descent

Solving the elastic net penalized probit regression problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} L(y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})) + \sum_i \lambda_1|\beta_j| + \frac{\lambda_2}{2}\beta_j^2, \qquad (1)$$

Generalized Coordinate Descent [Yang and Zou, 2013]:

Assume the input data are standardized: $\frac{1}{n}\sum_i x_{ij} = 0$, $\frac{1}{n}\sum_i x_{ij}^2 = 1$. Let $\tilde{\boldsymbol{\beta}}$ be the current estimate. Define the current margin $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_i^\top \tilde{\boldsymbol{\beta}})$. Coordinate descent algorithms {`glmnet`} [Friedman et al., 2010] updates the k-th coordinate by minimizing

$$F(\beta_k|\tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n} L(r_i + y_i x_{ik}(\beta_k - \tilde{\beta}_k)) + \lambda_1|\beta_k| + \frac{\lambda_2}{2}\beta_k^2. \qquad (2)$$

Instead, by Lemma, it can be majorized by a penalized quadratic function:

$$Q(\beta_k|\tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\left\{L(r_i) + L'(r_i)y_i x_{ik}(\beta_k - \tilde{\beta}_k) + \frac{1}{2}x_{ik}^2(\beta_k - \tilde{\beta}_k)^2\right\}}_{\text{quadratic majorization}} + \underbrace{\lambda_1|\beta_k| + \frac{\lambda_2}{2}\beta_k^2}_{\text{penalty}}. \qquad (3)$$

Algorithm: Generalized Coordinate Descent (Cont'd)

$$Q(\beta_k|\tilde{\boldsymbol{\beta}}) = \frac{1}{n}\sum_{i=1}^{n}\left\{L(r_i) + L'(r_i)y_i x_{ik}(\beta_k - \tilde{\beta}_k) + \frac{1}{2}x_{ik}^2(\beta_k - \tilde{\beta}_k)^2\right\} + \lambda_1|\beta_k| + \frac{\lambda_2}{2}\beta_k^2.$$

Function $Q(\beta_k|\tilde{\boldsymbol{\beta}})$ satisfys

- **Touching condition:** $Q(\tilde{\beta}_k|\tilde{\boldsymbol{\beta}}) = F(\tilde{\beta}_k|\tilde{\boldsymbol{\beta}})$;
- **Majorization condition:** $Q(\beta_k|\tilde{\boldsymbol{\beta}}) \geq F(\beta_k|\tilde{\boldsymbol{\beta}})$, $\forall \beta_k \in \mathbb{R}$;

which guarantee the descent property of minimization-majorization update:

$$\tilde{\beta}_k \leftarrow \min_{\beta_k} Q(\beta_k|\tilde{\boldsymbol{\beta}}). \tag{4}$$

We can easily solve it by a simple soft-thresholding rule [Zou and Hastie, 2005]:

$$\tilde{\beta}_k \leftarrow \frac{\mathcal{S}\left(\tilde{\beta}_k - \frac{1}{n}\sum_{i=1}^{n} L'(r_i)y_i x_{ik}, \lambda_1\right)}{1+\lambda_2}, \tag{5}$$

where $\mathcal{S}(z, t) = (|z| - t)_+ \mathrm{sgn}(z)$ and similarly,

$$\tilde{\beta}_0 \leftarrow \tilde{\beta}_0 - \frac{1}{n}\sum_{i=1}^{n} L'(r_i)y_i x_{ik}. \tag{6}$$

Implementation Details

Adaptive elastic net: different weights for different coefficients:
$\lambda_1 \sum_j w_j^{(1)} |\beta_j| + \frac{\lambda_2}{2} \sum_j w_j^{(2)} \beta_j^2$.

Solution path in λ_1:

- Given a fixed λ_2, we fit the solution path for a decreasing sequence of λ_1's.
- $\lambda_{1,\max}$ is the smallest λ_1 s.t. all β_j, $1 \le j \le p$ are zero.
- Set $\lambda_{1,\min} = \tau \lambda_{1,\max}$. Default value of τ is $\tau = 10^{-2}$ for $n < p$; $\tau = 10^{-4}$ for $n \ge p$.

Implementation tricks:

- Warm-start trick: the solution at $\lambda_1[k]$ is used as the initial value for $\lambda_1[k+1]$.
- Active-set trick:

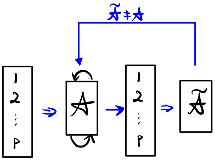


Table of Contents

Folded Concave Penalty

The SCAD penalty [Fan and Li, 2001]: for $t > 0$,

$$p'_\lambda(t) = \lambda I_{\{t \le \lambda\}} + \frac{(a\lambda - t)_+}{a-1} I_{\{t>\lambda\}}, \quad \text{for some } a > 2,$$

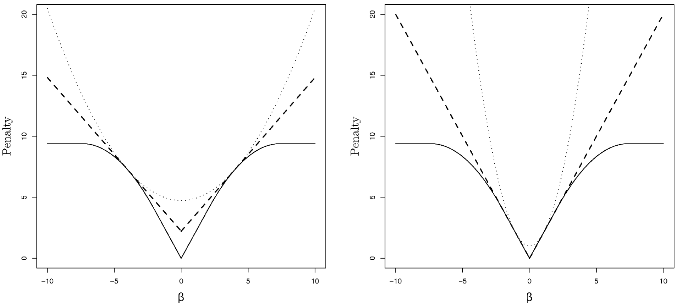


Figure: Solid Curve: SCAD penalty with $\lambda = 2$, $a = 3.7$.

Folded Concave Penalty (Cont'd)

Properties of a folded concave penalty [Fan and Li, 2001]: Unbiasedness, Sparsity, Continuity.

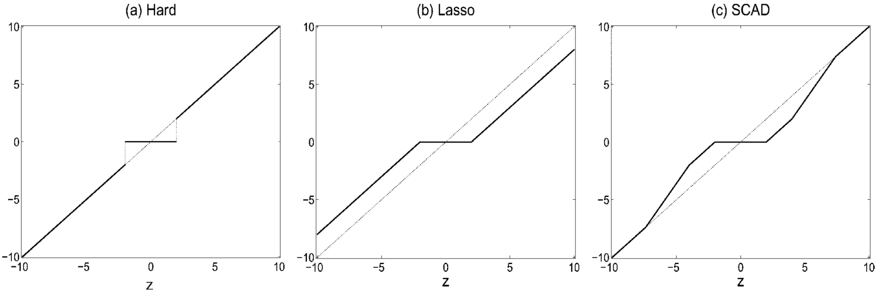


Figure: Plot [Fan and Li, 2001] of Thresholding Functions for (a) the Hard [Not continuous], (b) the Soft [Bias], and (c) the SCAD Thresholding Functions With $\lambda = 2$ and $a = 3.7$ for SCAD.

Local Linear Approximation [Zou and Li, 2008]

Solving the problem

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n} L(y_i, \beta_0 + \mathbf{x}_i^{\mathsf{T}}\boldsymbol{\beta}) + \sum_j p_\lambda(|\beta_j|), \tag{7}$$

where $L(\cdot,\cdot)$ is the convex loss function, and $p_\lambda(\cdot)$ is the SCAD penalty.

Let $\tilde{\boldsymbol{\beta}}$ be the current estimate. The folded concave penalty could be majorized by a local linear approximation function:

$$\sum_j p_\lambda(|\beta_j|) \leq \sum_j p_\lambda(|\tilde{\beta}_j|) + p'_\lambda(|\tilde{\beta}_j|)(|\beta_j| - |\tilde{\beta}_j|), \tag{8}$$

The objective function could be majorized by a weighted ℓ_1 penalized problem:

$$\min_{\boldsymbol{\beta}} \ell_n(\boldsymbol{\beta}) + \sum_j p_\lambda(|\tilde{\beta}_j|) + p'_\lambda(|\tilde{\beta}_j|)(|\beta_j| - |\tilde{\beta}_j|). \tag{9}$$

Outter Loop: Local Linear Approximation (Cont'd)

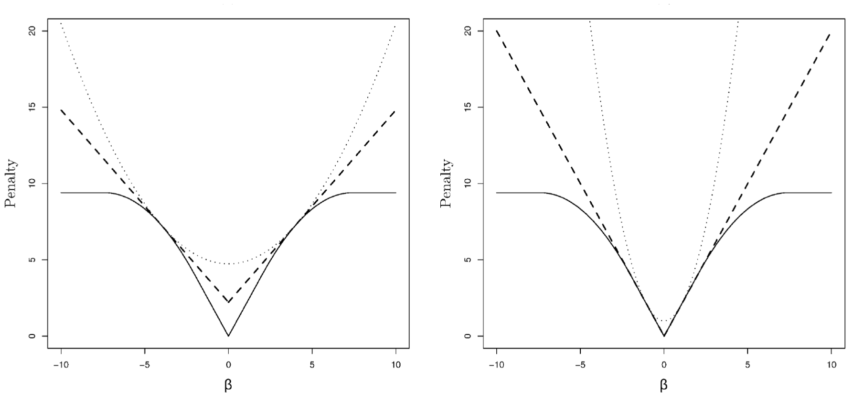


Figure: *Solid lines: SCAD penalty with $\lambda = 2$. Dashed lines: Local linear majorization of SCAD penalty [Zou and Li, 2008]. Dotted lines: Local quadratic majorization of SCAD penalty [Fan and Li, 2001].*

Outter Loop: Local Linear Approximation (Cont'd)

Algorithm 1: The Local Linear Approximation (LLA) Algorithm

1: Initialize $\hat{\boldsymbol{\beta}}^{(0)} = \hat{\boldsymbol{\beta}}^{\text{initial}}$ and compute the adaptive weight

$$\hat{w}^{(0)} = \left(\hat{w}_1^{(0)}, \cdots, \hat{w}_p^{(0)}\right)^\intercal = \left(p'_\lambda(|\hat{\beta}_1^{(0)}|), \cdots, p'_\lambda(|\hat{\beta}_p^{(0)}|)\right)^\intercal$$

2: For $m = 1, 2, \cdots$, repeat the LLA iteration till convergence

(2.a) Obtain $\hat{\beta}^{(m)}$ by solving the following optimization problem

$$\hat{\beta}^{(m)} = \arg\min_{\beta} \ell_n(\boldsymbol{\beta}) + \sum_i \hat{w}_j^{(m-1)} \cdot |\beta_j|,$$

(2.b) Update the adaptive weight vector $\hat{w}^{(m)}$ with $\hat{w}_j^{(m)} = p'_\lambda(|\hat{\beta}_j^{(m)}|)$.

Computing Algorithm: LLA-GCD

Functions and Methods:

- `lla.gcdnet(...)`: apply LLA-GCD algorithm to solve the solution path of folded concave (SCAD) penalized estimations:

```
method = c("hhsvm", "logit", "sqsvm", "ls", "er", "probit")
```

- `cv.lla.gcdnet(...)`: do K-fold cross-validation to choose the penalization parameter λ.
- S3 method: `plot`, `print`, `coef`, `predict`.

Discussion:

- For folded concave penalized estimations, how to put the whole LLA-GCD algorithm in `Fortran` to speed up?

Table of Contents

Example: Simulation data

Simulation dataset:

- Generate data from probit model;
- $n = 100$ and $p = 500$;
- $X \sim N(\mathbf{0}, \Sigma)$, Σ is AR(1) with $\rho = 0.5$;
- $\beta^* = (3, 1.5, 0, 0, 2, \mathbf{0}_{p-5})$, support set $\mathcal{A} = \{1, 2, 5\}$.

Oracle estimator: The MLE given true support set $\mathcal{A}$:

$$(\hat{\beta}_1^{\text{oracle}}, \hat{\beta}_2^{\text{oracle}}, \hat{\beta}_5^{\text{oracle}}) = (2.436,\ 1.584,\ 1.616)$$

Example: Elastic Net Penalized Probit Regression

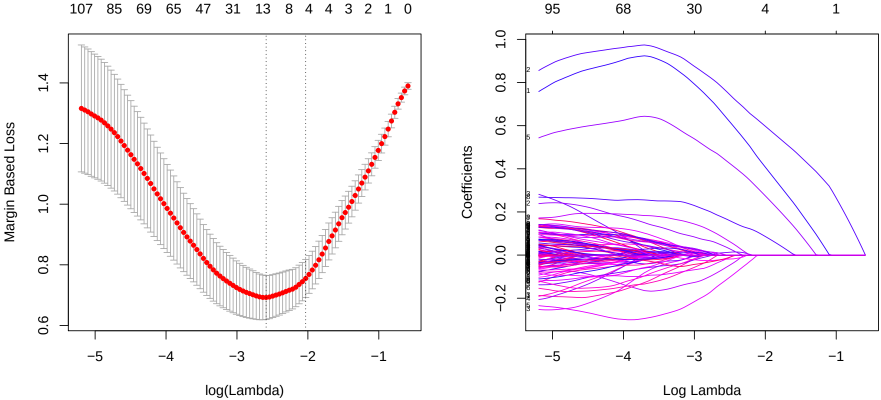


Figure: *Set* $\lambda_2 = 0.01$*. At* `lambda.1se`*, variables 1, 2, 5, 58 are selected.*

Example: Folded Concave (SCAD) Penalized Probit Regression

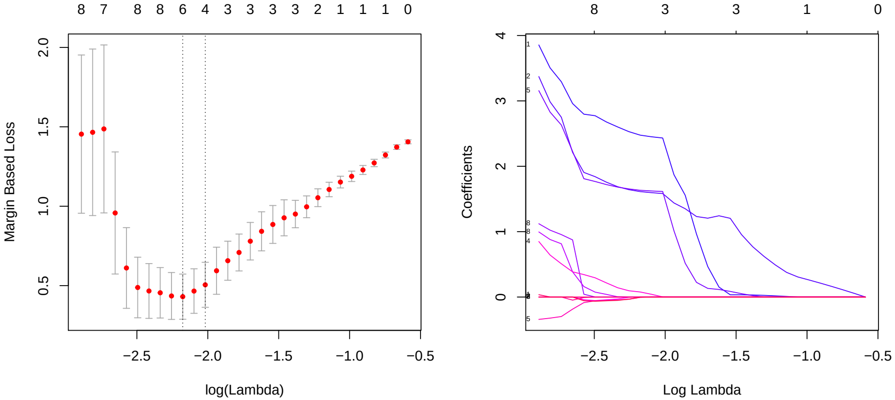


Figure: *At* `lambda.min`*, variables 1, 2, 5, 394 are selected.*
$\hat{\beta}_{(1,2,5)} = (2.433, 1.583, 1.614)$, $\hat{\beta}^{oracle}_{(1,2,5)} = (2.436, 1.584, 1.616)$. *[Fan et al., 2014].*

References I

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

J. Fan, L. Xue, and H. Zou. Strong oracle optimality of folded concave penalized estimation. *Annals of statistics*, 42(3):819, 2014.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Y. Yang and H. Zou. An efficient algorithm for computing the hhsvm and its generalizations. *Journal of Computational and Graphical Statistics*, 22(2): 396–415, 2013.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2): 301–320, 2005.

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

Thank You