

Computation & optimization for Lasso - part 2

Luyang Han & Janosch Ott

ETH Zürich

22 October 2018

Overview

1. Coordinate Descent
2. Least Angle Regression
3. Comparison of Optimization Methods
4. Recall: Duality
5. ADMM
6. Screening Rules
7. Minor-Max Algorithms
8. Alternating Minimizations

Coordinate Descent Algorithm

What is Coordinate Descent (CD) Algorithm?

$$\beta_k^{t+1} = \underset{\beta_k}{\operatorname{argmin}} f(\beta_1^t, \beta_2^t, \dots, \beta_k, \beta_{k+1}^t, \dots, \beta_p^t) \quad (1)$$

and $\beta_j^{t+1} = \beta_j^t$ for $j \neq k$

- An iterative algorithm that updates from β^t to β^{t+1} by choosing a single coordinate, and minimizing over this coordinate.

Separability Condition

Motivation

Does CD procedure converge to the global minimum of a convex function?

Separability Condition

Motivation

Does CD procedure converge to the global minimum of a convex function?

- **Sufficient Condition:** the function is continuously differentiable and strictly convex in each coordinate.

⇒ restrictive

Separability Condition

Suppose the cost function f has the additive decomposition:

$$f(\beta_1, \dots, \beta_p) = g(\beta_1, \dots, \beta_p) + \sum_{j=1}^p h_j(\beta_j) \quad (2)$$

where $g : \mathbb{R}^p \rightarrow \mathbb{R}$ is differentiable and convex, and the univariate functions $h_j : \mathbb{R} \rightarrow \mathbb{R}$ is convex.

- Lasso: $g(\beta) = \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2$ and $h_j(\beta_j) = \lambda|\beta_j|$ satisfies the condition

Separability Condition: Example

An Example of failure of Coordinate Descent

$$\operatorname{argmin}_{\beta \in \mathbb{R}^p} \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=2}^p |\beta_j - \beta_{j-1}|$$

- $h(\beta)$ is not separable
- Fused Lasso: coordinate descent procedure is not guaranteed to find the global minimum

Separability Condition: Example

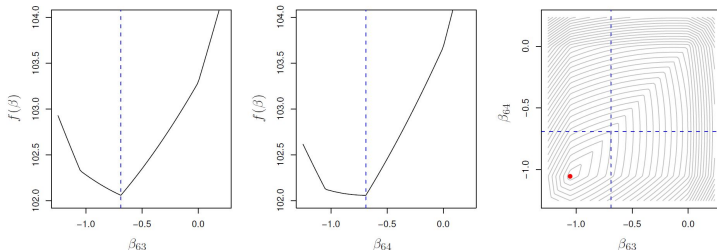


Figure: Fused Lasso: CD fail to reach the global minimum

Lasso & Coordinate Descent

Optimality Condition:

$$-\frac{1}{2N} \sum_{i=1}^N (y_i - \beta_0 - \sum_{k=1}^p x_{ik} \beta_k) x_{ij} + \lambda s_j = 0$$

where $s_j \in \text{sign}(\beta_j)$ for $j = 1, 2, \dots, p$

- Define the **partial residual**: $r_i^{(j)} = y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k$
- Then the solution for $\hat{\beta}_j$ satisfies:

$$\hat{\beta}_j = \frac{S_\lambda(\frac{1}{N} \sum_{i=1}^N r_i^{(j)} x_{ij})}{\frac{1}{N} \sum_{i=1}^N x_{ij}^2}$$

where $S_\lambda(\theta) = \text{sign}(\theta)(|\theta| - \lambda)_+$

Lasso & Coordinate Descent

- Illustration of Coordinate Descent in R

Strategies to make the operation efficient:

Naive Updating

$$r_i^{(j)} = y_i - \sum_{k \neq j} x_{ik} \hat{\beta}_k = r_i + x_{ij} \hat{\beta}_j$$

$$\frac{1}{N} \sum_{i=1}^N x_{ij} r_i^{(j)} = \frac{1}{N} \sum_{i=1}^N x_{ij} r_i + \hat{\beta}_j$$

Covariance Updating

$$\sum_{i=1}^N x_{ij} r_i = \langle x_j, y \rangle - \sum_{k || \beta_{\hat{k}} > 0} \langle x_j, x_k \rangle \beta_{\hat{k}}$$

Warm Starts: For a decreasing sequence of values $\{\lambda_0^L\}$, $\hat{\beta}(\lambda_l)$ is typically a very good warm start for the solution $\hat{\beta}(\lambda_{l+1})$.

We set $\lambda_0 = \frac{1}{N} \max |\langle x_j, y \rangle|$ and $\lambda_L \approx 0$

Lasso & Coordinate Descent

Active-set Convergence: Define the active set A and iterate the algorithm using only variables in A .

Strong-set Convergence: Define the strong set S and iterate the algorithm using only variables in S .

Sparsity: Sparsity of the design matrix X makes the operation of inner product efficient.

Details in page 113 and page 114.

Elastic Net & Coordinate Descent

$$\text{minimize}_{\beta_0, \beta_p} \frac{1}{2} \sum_{i=1}^N (y_i - \beta_0 - x_i^T \beta)^2 + \lambda \left[\frac{1}{2} (1 - \alpha) \|\beta\|_2^2 + \alpha \|\beta\|_1 \right]$$

- Combination of L1 and L2 penalty
- Satisfy the separability condition
- The solution satisfies:

$$\hat{\beta}_j = \frac{S_{\alpha\lambda}(\frac{1}{N} \sum_{i=1}^N r_i^{(j)} x_{ij})}{\frac{1}{N} \sum_{i=1}^N x_{ij}^2 + (1 - \alpha)\lambda}$$

Logistic Regression & Coordinate Descent

Background

- **Class Label G :** Take values 1 and -1

Denote $p(x_i; \beta_0, \beta) = \Pr(G = 1|x_i)$

Define log odds: $\log \frac{\Pr(G=-1|x)}{\Pr(G=1|x)} = \beta_0 + x^T \beta$

Logistic Regression & Coordinate Descent

Background

- **Class Label G :** Take values 1 and -1

Denote $p(x_i; \beta_0, \beta) = \Pr(G = 1|x_i)$

Define log odds: $\log \frac{\Pr(G=-1|x)}{\Pr(G=1|x)} = \beta_0 + x^T \beta$

- **Maximize penalized log-likelihood:**

$$\frac{1}{N} \sum_{i=1}^N \{I(g_i = 1) \cdot \log p(x_i; \beta_0, \beta) + I(g_i = -1) \cdot \log(1 - p(x_i; \beta_0, \beta))\} - \lambda \|\beta\|_1$$

Denote $y_i = I(g_i = -1)$

Explicit form of log likelihood (without penalty):

$$l(\beta_0, \beta) = \frac{1}{N} \sum_{i=1}^N [y_i \cdot (\beta_0 + x_i^T \beta) - \log(1 + e^{\beta_0 + x_i^T \beta})]$$

Logistic Regression & Coordinate Descent

Background

- Form a **quadratic objective function** using Taylor expansion about current estimates $(\tilde{\beta}_0, \tilde{\beta})$: Idea of Newton method, Iterated Weighted Least Square problem

$$l_Q(\beta_0, \beta) = -\frac{1}{2N} \sum_{i=1}^N (w_i(z_i - \beta_0 - \mathbf{x}_i^T \beta))^2 + C(\tilde{\beta}_0, \tilde{\beta})$$

- Use Coordinate Descent to solve the problem

$$\underset{(\beta_0, \beta)}{\text{minimize}} \{ l_Q\{\beta_0, \beta\} + \lambda \|\beta\|_1 \}$$

Logistic Regression & Coordinate Descent

Algorithm

OUTER LOOP: Decrement λ

MIDDLE LOOP: Update the **quadratic approximation** l_Q
using the current parameters $(\tilde{\beta}_0, \tilde{\beta})$

INNER LOOP: Run the coordinate descent algorithm on the
penalized weighted least squares problem

Least Angle Regression

Introduction

- Relates to Forward Selection method
- Relates to Lasso method
- Able to deliver the entire solution path of the lasso problem with squared-error loss as a function of the regularization parameter λ

Least Angle Regression: Algorithm

- Start with all coefficients β_j equal to zero
- Find the predictor X_j **most correlated** with y
- **Increase** the coefficient β_j in the direction of the **sign** of its **correlation** with y
- Take **residuals** $r = y - \hat{y}$ along the way; Stop when some other predictor X_k has **as much correlation** with r as X_j has
- **Increase** β_j, β_k in their **joint least squares direction**, until some other predictor has as much correlation with the residual r

Continue until: all predictors are in the model

Least Angle Regression: Algorithm

Luyang Han &
Janosch Ott

Coordinate
Descent

Least Angle
Regression

Comparison of
Optimization
Methods

Recall:
Duality

ADMM

Screening
Rules

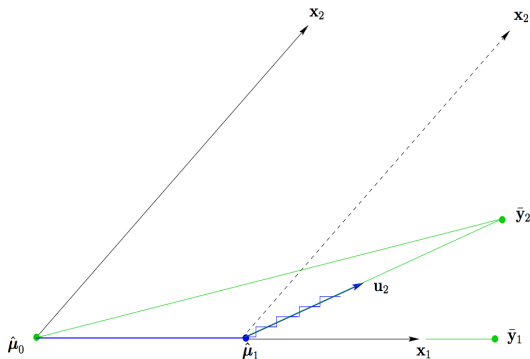
Minor-Max
Algorithms

Alternating
Minimizations

Algorithm 5.1 LEAST ANGLE REGRESSION.

1. Standardize the predictors to have mean zero and unit ℓ_2 norm. Start with the residual $\mathbf{r}_0 = \mathbf{y} - \bar{\mathbf{y}}$, $\beta^0 = (\beta_1, \beta_2, \dots, \beta_p) = \mathbf{0}$.
 2. Find the predictor \mathbf{x}_j most correlated with \mathbf{r}_0 ; i.e., with largest value for $|\langle \mathbf{x}_j, \mathbf{r}_0 \rangle|$. Call this value λ_0 , define the active set $\mathcal{A} = \{j\}$, and $\mathbf{X}_{\mathcal{A}}$, the matrix consisting of this single variable.
 3. For $k = 1, 2, \dots, K = \min(N - 1, p)$ do:
 - (a) Define the least-squares direction $\delta = \frac{1}{\lambda_{k-1}} (\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}})^{-1} \mathbf{X}_{\mathcal{A}}^T \mathbf{r}_{k-1}$, and define the p -vector Δ such that $\Delta_{\mathcal{A}} = \delta$, and the remaining elements are zero.
 - (b) Move the coefficients β from β^{k-1} in the direction Δ toward their least-squares solution on $\mathbf{X}_{\mathcal{A}}$: $\beta(\lambda) = \beta^{k-1} + (\lambda_{k-1} - \lambda)\Delta$ for $0 < \lambda \leq \lambda_{k-1}$, keeping track of the evolving residuals $\mathbf{r}(\lambda) = \mathbf{y} - \mathbf{X}\beta(\lambda) = \mathbf{r}_{k-1} - (\lambda_{k-1} - \lambda)\mathbf{X}\Delta$.
 - (c) Keeping track of $|\langle \mathbf{x}_{\ell}, \mathbf{r}(\lambda) \rangle|$ for $\ell \notin \mathcal{A}$, identify the largest value of λ at which a variable “catches up” with the active set; if the variable has index j , that means $|\langle \mathbf{x}_j, \mathbf{r}(\lambda) \rangle| = \lambda$. This defines the next “knot” λ_k .
 - (d) Set $\mathcal{A} = \mathcal{A} \cup \{j\}$, $\beta^k = \beta(\lambda_k) = \beta^{k-1} + (\lambda_{k-1} - \lambda_k)\Delta$, and $\mathbf{r}_k = \mathbf{y} - \mathbf{X}\beta^k$.
 4. Return the sequence $\{\lambda_k, \beta^k\}_0^K$.
-

Least Angle Regression: Geometric Representation



Computation
&
optimization

Luyang Han &
Janosch Ott

Coordinate
Descent

Least Angle
Regression

Comparison of
Optimization
Methods

Recall:
Duality

ADMM

Screening
Rules

Minor-Max
Algorithms

Alternating
Minimizations

Connection between LAR and Lasso

LAR

$\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\beta(\lambda)) = \lambda \cdot s_j, \forall j \in \mathbb{A}$ where s_j is the sign of inner product λ . [3(c)]

Connection between LAR and Lasso

LAR

$\mathbf{x}_j^T (\mathbf{y} - \mathbf{X}\beta(\lambda)) = \lambda \cdot s_j, \forall j \in \mathbb{A}$ where s_j is the sign of inner product λ . [3(c)]

LASSO

Let \mathbb{B} be the active set of variables in the solution for a given value of λ .

$$R(\beta) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1$$

For differentiable $R(\beta)$, the stationary conditions give:

$$\mathbf{x}_j^T (\mathbf{y} - \mathbf{X}\beta) = \lambda \cdot \text{sign}(\beta_j), \forall j \in \mathbb{B}$$

If $\text{sign}(\beta_j)$ matches s_j , the coefficient would be identical.

Connection between LAR and Lasso

- R Example
- LAR algorithm explains that the coefficient paths for the lasso are **piecewise linear**
- Coefficient paths differ if $\text{sign}(\beta_j)$ is different from s_j
- **Modification** of LAR for computing Lasso solution $[3(c)+]$:
If a **nonzero** coefficient **crosses zero** before the next variable enters, **drop** it from \mathbb{A} and recompute the current joint least squares direction.

Connection between LAR and Lasso

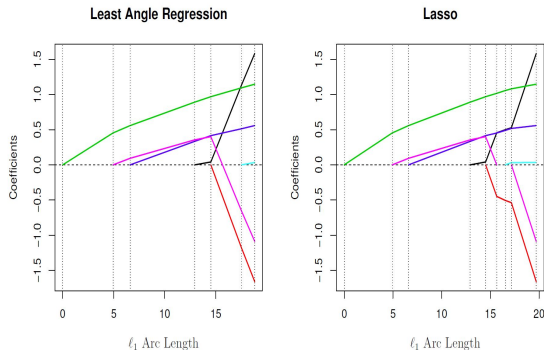


Figure: Cases where signs of λ and β disagree

Algorithm Performance

Simulation: Comparison of computation efficiency between CD and LAR

Set Up: ⁵

- Generate Gaussian data with N observations and p predictors, with each pair of predictors X_j, X_k having the same population correlation ρ .
- Try different combination of N and p ; Range ρ from 0 to 0.95.

$$Y = \sum_{j=1}^p X_j \beta_j + kZ \quad \text{where}$$

$$\beta_j = (-1)^j \exp\left(\frac{-2(j-1)}{20}\right), Z \sim N(0, 1) \text{ and } k \text{ is a constant.}$$

⁵Friedman, Hastie, Tibshirani (2010)

Algorithm Performance

Linear Regression — Dense Features						
	Correlation					
	0	0.1	0.2	0.5	0.9	0.95
<i>N</i> = 1000, <i>p</i> = 100						
glmnet-naive	0.05	0.06	0.06	0.09	0.08	0.07
glmnet-cov	0.02	0.02	0.02	0.02	0.02	0.02
lars	0.11	0.11	0.11	0.11	0.11	0.11
<i>N</i> = 5000, <i>p</i> = 100						
glmnet-naive	0.24	0.25	0.26	0.34	0.32	0.31
glmnet-cov	0.05	0.05	0.05	0.05	0.05	0.05
lars	0.29	0.29	0.29	0.30	0.29	0.29
<i>N</i> = 100, <i>p</i> = 1000						
glmnet-naive	0.04	0.05	0.04	0.05	0.04	0.03
glmnet-cov	0.07	0.08	0.07	0.08	0.04	0.03
lars	0.73	0.72	0.68	0.71	0.71	0.67
<i>N</i> = 100, <i>p</i> = 5000						
glmnet-naive	0.20	0.18	0.21	0.23	0.21	0.14
glmnet-cov	0.46	0.42	0.51	0.48	0.25	0.10
lars	3.73	3.53	3.59	3.47	3.90	3.52
<i>N</i> = 100, <i>p</i> = 20000						
glmnet-naive	1.00	0.99	1.06	1.29	1.17	0.97
glmnet-cov	1.86	2.26	2.34	2.59	1.24	0.79
lars	18.30	17.90	16.90	18.03	17.91	16.39
<i>N</i> = 100, <i>p</i> = 50000						

Figure: Comparison of computing time

Algorithm Performance

Simulation: Comparison of computation efficiency between Coordinate Descent, Proximal Gradient Descent and Nesterov Method

Set Up: ⁷

- Generated an $N \times p$ predictor matrix X with standard Gaussian entries and pairwise correlation 0 or 0.5 between the features.
- $|\beta_j| = \exp[-0.5(u(j-1))^2]$ and $u = \sqrt{\frac{\pi}{20}}$ and alternating signs -1,+1,-1...

Algorithm Performance

Table 5.1 *Lasso for linear regression: Average (standard error) of CPU times over ten realizations, for coordinate descent, generalized gradient, and Nesterov's momentum methods. In each case, time shown is the total time over a path of 20 λ values.*

Correlation	$N = 10000, p = 100$		$N = 200, p = 10000$	
	0	0.5	0	0.5
Coordinate descent	0.110 (0.001)	0.127 (0.002)	0.298 (0.003)	0.513 (0.014)
Proximal gradient	0.218 (0.008)	0.671 (0.007)	1.207 (0.026)	2.912 (0.167)
Nesterov	0.251 (0.007)	0.604 (0.011)	1.555 (0.049)	2.914 (0.119)

Figure: Comparison of computing efficiency between 3 methods

Recall: Duality in optimization

Primal	
Optimize	$\min f(x)$
Constraints	$g_i(x) \leq 0, h_j(x) = 0, x \in X$
Function	$L(x, \lambda, \mu) := f(x) + \sum_i \lambda_i g_i(x) + \sum_j \mu_j h_j(x)$
Dual	
Function	$q(\lambda, \mu) = \inf_{x \in X} L(x, \lambda, \mu)$
Constraints	$\lambda \geq 0$
Optimize	$\max_{\lambda \geq 0, \mu} q(\lambda, \mu)$

Why though? - **Dual problem is always convex!**

Alternating Direction Method of Multipliers (ADMM)

Problem

$$\underset{\beta \in \mathbb{R}^m, \theta \in \mathbb{R}^n}{\text{minimize}} \quad f(\beta) + g(\theta) \quad \text{subject to} \quad \mathbf{A}\beta + \mathbf{B}\theta - c = 0$$

Lagrangian

$$f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle$$

Augmented Lagrangian

$$L_\rho(\beta, \theta, \mu) := f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle + \frac{\rho}{2} \|\mathbf{A}\beta + \mathbf{B}\theta - c\|_2^2$$

Alternating Direction Method of Multipliers (ADMM)

Problem - decomposable !

$$\underset{\beta \in \mathbb{R}^m, \theta \in \mathbb{R}^n}{\text{minimize}} \quad f(\beta) + g(\theta) \quad \text{subject to} \quad \mathbf{A}\beta + \mathbf{B}\theta - c = 0$$

Lagrangian

$$f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle$$

Augmented Lagrangian

$$L_\rho(\beta, \theta, \mu) := f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle + \frac{\rho}{2} \|\mathbf{A}\beta + \mathbf{B}\theta - c\|_2^2$$

Alternating Direction Method of Multipliers (ADMM)

Problem - decomposable !

$$\underset{\beta \in \mathbb{R}^m, \theta \in \mathbb{R}^n}{\text{minimize}} f(\beta) + g(\theta) \quad \text{subject to } \mathbf{A}\beta + \mathbf{B}\theta - c = 0$$

Lagrangian - decomposable !

$$f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle$$

Augmented Lagrangian

$$L_\rho(\beta, \theta, \mu) := f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle + \frac{\rho}{2} \|\mathbf{A}\beta + \mathbf{B}\theta - c\|_2^2$$

Alternating Direction Method of Multipliers (ADMM)

Problem - decomposable !

$$\underset{\beta \in \mathbb{R}^m, \theta \in \mathbb{R}^n}{\text{minimize}} f(\beta) + g(\theta) \quad \text{subject to } \mathbf{A}\beta + \mathbf{B}\theta - c = 0$$

Lagrangian - decomposable !

$$f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle$$

Augmented Lagrangian - NOT decomposable !

$$L_\rho(\beta, \theta, \mu) := f(\beta) + g(\theta) + \langle \mu, \mathbf{A}\beta + \mathbf{B}\theta - c \rangle + \frac{\rho}{2} \|\mathbf{A}\beta + \mathbf{B}\theta - c\|_2^2$$

Dual Variable Update

Alternating Direction Method of Multipliers

$$\beta^{t+1} = \arg \min_{\beta \in \mathbb{R}^m} L_{\rho}(\beta, \theta^t, \mu^t)$$

$$\theta^{t+1} = \arg \min_{\theta \in \mathbb{R}^m} L_{\rho}(\beta^{t+1}, \theta, \mu^t)$$

$$\mu^{t+1} = \mu^t + \rho(\mathbf{A}\beta^{t+1} + \mathbf{B}\theta^{t+1} - c)$$

Dual Variable Update

Alternating Direction Method of Multipliers

$$\beta^{t+1} = \arg \min_{\beta \in \mathbb{R}^m} L_{\rho}(\beta, \theta^t, \mu^t)$$

$$\theta^{t+1} = \arg \min_{\theta \in \mathbb{R}^m} L_{\rho}(\beta^{t+1}, \theta, \mu^t)$$

$$\mu^{t+1} = \mu^t + \rho(\mathbf{A}\beta^{t+1} + \mathbf{B}\theta^{t+1} - c)$$

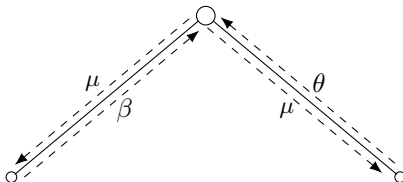


Figure: My own illustration of the dual ascent step in the ADMM algorithm utilising dual decomposition based on [Gordon and Tibshirani, 2012].

ADMM - Why?

- convex problems with nondifferentiable constraints
- blockwise computation
 - sample blocks
 - feature blocks

ADMM for the Lasso

Problem

Problem in Lagrangian form

$$\underset{\beta \in \mathbb{R}^p, \theta \in \mathbb{R}^p}{\text{minimize}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\theta\|_1 \right\} \quad \text{such that } \beta - \theta = 0$$

Augmented Lagrangian

$$L_\rho(\beta, \theta, \mu) := \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\theta\|_1 \right\} + \langle \mu, \beta - \theta \rangle + \frac{\rho}{2} \|\beta - \theta\|_2^2$$

ADMM for the Lasso

Update

Update

$$\beta^{t+1} = (\mathbf{X}^T \mathbf{X} + \rho \mathbf{I})^{-1} (\mathbf{X}^T \mathbf{y} + \rho \theta^t - \mu^t)$$

$$\theta^{t+1} = \mathcal{S}_{\lambda/\rho}(\beta^{t+1} + \mu^t/\rho)$$

$$\mu^{t+1} = \mu^t + \rho(\beta^{t+1} - \theta^{t+1})$$

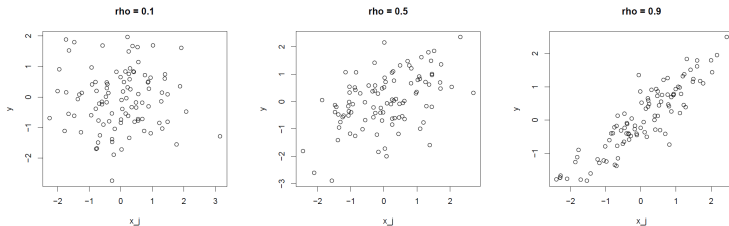
where $\mathcal{S}_{\lambda/\rho}(z) = \text{sign}(z)(|z| - \frac{\lambda}{\rho})_+$.

Screening Rules

- Pre-processing to eliminate features
- very big data set, esp. huge number of predictors
- maybe too big to load into memory
- Screening rules eliminate predictors with minor calculation
- and very high / safe certainty (i.e. eliminated predictors would not show up in lasso model based on full data)

They achieve a reduction in the number of variables, typically by an order of magnitude

What is a good predictor?



correlation is an inner product

high absolute correlation (=large absolute inner product)

— $>$ high predictive power (compare plots)

— $>$ x_j with largest inner product has highest predictive power,
thus for that j we are most willing to accept some penalty from
 λ

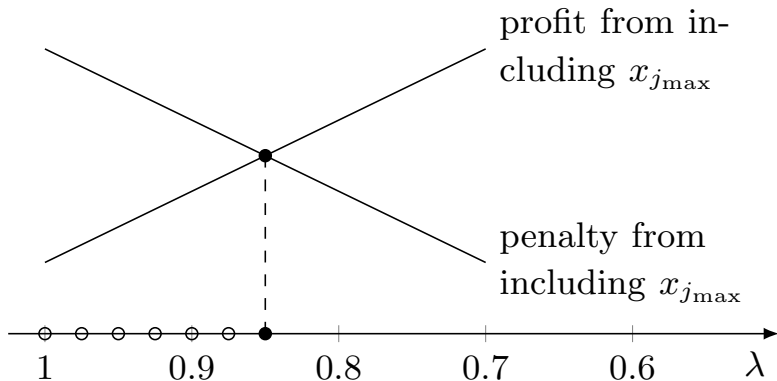
Lasso - an iterative algorithm

Let \mathcal{A} be the active set of predictors. Let λ take values on a decreasing sequence.

iterate

1. order predictors x_j not in \mathcal{A} by their "effectiveness" using $|x_j^T y|$ or better $|x_j^T (y - \hat{y}_\lambda)|$, call the best predictor $x_{j_{\max}}$
2. move λ such that the positive effect from the best predictor $x_{j_{\max}}$ compensates the penalty by λ
3. calculate solution for chosen λ

Lasso - a visual interpretation



Back to screening rules

Let λ take values on a decreasing sequence. Let λ_{\max} be the λ where the first predictor has a non-zero coefficient.

$$\lambda_{\max} = \max_j |x_j^T y|$$

Let \mathcal{A} be the active set of predictors.

$$\forall j \in \mathcal{A} \quad \lambda = |x_j^T (y - \hat{y})|$$

$$\forall j \notin \mathcal{A} \quad \lambda \geq |x_j^T (y - \hat{y})|$$

Back to screening rules

Let λ take values on a decreasing sequence. Let λ_{\max} be the λ where the first predictor has a non-zero coefficient.

$$\lambda_{\max} = \max_j |x_j^T y|$$

Let \mathcal{A} be the active set of predictors.

$$\forall j \in \mathcal{A} \quad \lambda = |x_j^T (y - \hat{y})|$$

$$\forall j \notin \mathcal{A} \quad \lambda \geq |x_j^T (y - \hat{y})|$$

R example

Global vs. Sequential

Global (one-time screening):

Suppose we want to calculate a lasso solution at $\lambda < \lambda_{\max}$.

Sequential (iterative screening):

Suppose we have the lasso solution $\hat{\beta}(\lambda')$ at λ' and want to screen variables for solutions at $\lambda < \lambda'$.

Dual Polytope Projection (DPP)

Suppose we want to calculate a lasso solution at $\lambda < \lambda_{\max}$.
The DPP rule discards the j^{th} variable if

$$\left| \mathbf{x}_j^T \mathbf{y} \right| < \lambda_{\max} - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda}$$

Sequential DPP rule

Suppose we have the lasso solution $\hat{\beta}(\lambda')$ at λ' and want to
screen variables for solutions at $\lambda < \lambda'$. We discard the j^{th}
variable if

$$\left| \mathbf{x}_j^T (\mathbf{y} - \mathbf{X} \hat{\beta}(\lambda')) \right| < \lambda' - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda}$$

Global Strong Rule

Suppose we want to calculate a lasso solution at $\lambda < \lambda_{\max}$.
The global strong rule discards the j^{th} variable if

$$\left| \mathbf{x}_j^T \mathbf{y} \right| < \lambda - (\lambda_{\max} - \lambda) = 2\lambda - \lambda_{\max}$$

Sequential Strong Rule

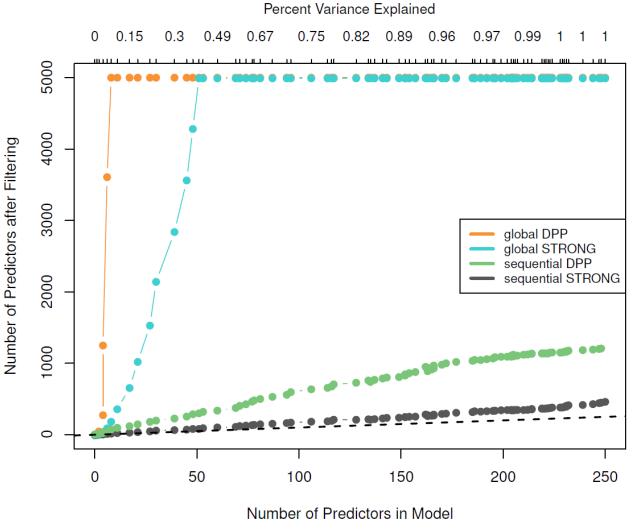
Suppose we have the lasso solution $\hat{\beta}(\lambda')$ at λ' and want to screen variables for solutions at $\lambda < \lambda'$. We discard the j^{th} variable if

$$\left| \mathbf{x}_j^T (\mathbf{y} - \mathbf{X} \hat{\beta}(\lambda')) \right| < 2\lambda - \lambda'$$

Screening Rules - Example Setup

- simulated dataset
- $N = 200, p = 5000$ uncorrelated Gaussian predictors,
- $1/4$ true non-zero coefficients
- 100 decreasing lambda values equally spaced on the log-scale
- Compare Global DPP, Global Strong, Sequential DDP, Sequential Strong
- no violations for either of the strong rules

Figure: From [Hastie et al., 2015]



Summary I

Coordinate Descent

- An efficient algorithm implemented in glmnet but requires separability condition
- Application: Ridge, Lasso, Elastic Net, Logistic Regression, etc.

Least Angle Regression

- Similar to the idea of Forward Selection
- Computationally efficient but does not scale well to large problems

Connection between LASSO and LAR

- LAR could be modified to obtain Lasso solution
- Explains the fact that Lasso coefficient solution path is piece-wise linear

Summary II

ADMM

- Use duality to your advantage
- Limitations in speed for Lasso, but useful in more complex settings

Screening Rules

- Promising for very large p 's
- Difficult to find best rule, field in development

Minorization-Maximization Algorithms (MMA)

- Problem: minimize $f(\beta)$ over $\beta \in \mathbb{R}^p$
for f possibly non-convex
- Introduce additional variable θ
- Use θ to majorize (bound from above) the objective
function to be minimized

Majorization-Minimization Algorithms work analogously.

MMA visually

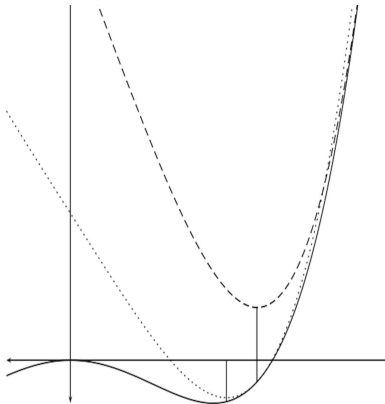


Figure: Figure from [de Leeuw, 2015]

MMA analytically I

Def. $\Psi : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ **majorizes** f at $\beta \in \mathbb{R}^p$ if

$$\forall \theta \in \mathbb{R}^p \quad \Psi(\beta, \theta) \geq f(\beta)$$

with equality for $\theta = \beta$.

Minor-Maxxalgorithm

- initialize β^0
- update with $\beta^{t+1} = \arg \min_{\beta \in \mathbb{R}^p} \Psi(\beta, \beta^t)$

MMA analytically II

This scheme generates a sequence of β 's for which the cost $f(\beta^t)$ is nonincreasing, because

$$f(\beta^t) \stackrel{(i)}{=} \Psi(\beta^t, \beta^t) \stackrel{(ii)}{\geq} \Psi(\beta^{t+1}, \beta^t) \stackrel{(iii)}{\geq} f(\beta^{t+1})$$

where

(i) & (iii) Definiton of majorize

(ii) β^{t+1} is a minimizer of $\beta \mapsto \Psi(\beta, \beta^t)$

Biconvexity

Let's consider an example . . .

$$f(\alpha, \beta) = (1 - \alpha\beta)^2$$

Biconvexity

Let's consider an example . . .

$$f(\alpha, \beta) = (1 - \alpha\beta)^2$$

Def. A function $f(\alpha, \beta) : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ is **biconvex**, if for each $\alpha \in \mathbb{R}^m$ the function $\alpha \mapsto f(\alpha, \beta)$ is convex and for each $\beta \in \mathbb{R}^n$ the function $\beta \mapsto f(\alpha, \beta)$ is convex. Analogously, a set $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{B}$, for \mathcal{A}, \mathcal{B} convex sets, is called biconvex, if it is convex

Alternate Convex Search

Block coordinate descent applied to α and β blocks

1. Initialize (α^0, β^0) at some point in the biconvex set to minimize over
2. For $t = 0, 1, 2, \dots$
 - (i) Fix $\beta = \beta^t$ and update $\alpha^{t+1} \in \arg \min_{\alpha \in \mathcal{C}_{\beta^t}} f(\alpha, \beta^t)$
 - (ii) Fix $\alpha = \alpha^{t+1}$ and update $\beta^{t+1} \in \arg \min_{\beta \in \mathcal{C}_{\alpha^{t+1}}} f(\alpha^{t+1}, \beta)$

For a function bounded from below, the algorithm converges to a partial optimum (i.e. as biconvexity, only optimal in one coordinate if the other coordinate is fixed).

References I



Trevor Hastie, Robert Tibshirani, and Martin Wainwright (2015)
Statistical learning with sparsity: the Lasso and generalizations
CRC Press; Boca Raton, FL



Jan De Leeuw (2015)
Block Relaxation Methods in Statistics
doi.org/10.13140/RG.2.1.3101.9607 (last accessed: 02.10.18)



S. Boyd
Alternating Direction Method of Multipliers
https://web.stanford.edu/~boyd/papers/pdf/admm_slides.pdf
(last accessed: 14.10.18)



Geoff Gordon and Ryan Tibshirani (2012)
Uses of Duality
<https://www.cs.cmu.edu/~ggordon/10725-F12/slides/18-dual-uses.pdf> (last accessed: 14.10.18)

References II



Paul Rubin (2016)

What are the advantages of convex optimization compared to more general optimization problems?

<https://www.quora.com/>

What-are-the-advantages-of-convex-optimization-compared-to-m
(last accessed: 14.10.18)

Comments . . .
Questions . . .
Suggestions . . .

That's it.
Thanks for listening.

Fill out your feedback sheets!