Computation & optimization for Lasso - part 2

Luyang Han & Janosch Ott

ETH Zürich

22 October 2018

Computation & optimization for Lasso - part 2

Luyang Han & Janosch Ott

ETH Zürich

22 October 2018

Overview

Overview

1. Coordinate Descent
2. A Simulation Study
3. Least Angle Regression
4. Digression: Duality
5. ADMM
6. Screening Rules
7. Minor-Max Algorithms
8. Alternating Minimizations

Recall: Duality in optimization

Recall: Duality in optimization

In various section, I came across terms like "dual" and "dual problem"

	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	$\max q(\lambda, \mu)$

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

	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	$\max q(\lambda, \mu)$

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

$x \in X$ for e.g. solutions in a cone or integer solutions
Terms: Primal problem, Lagrange function with dual variables/Lagrange-multipliers, dual function (λ and μ now in vector notation), dual problem (max q)
Dual problem is always convex! - I don't know much about optimization yet, but they really like convexity.
"(Convexity confers two advantages. The first is that, in a constrained problem, a convex feasible region makes it easier to ensure that you do not generate infeasible solutions while searching for an optimum.)
The second advantage is that all local optima are global optima. That allows local search algorithms to guarantee optimal solutions. And local search is often faster." [Rubin, 2016])

Alternating Direction Method of Multipliers (ADMM)

Problem

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

Lagrangian

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

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

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

Lagrangian

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

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

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

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

decomposable problem and constraints!

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 !
$\underset{\beta\in\mathbb{R}^m,\theta\in\mathbb{R}^n}{\text{minimize}}\, f(\beta)+g(\theta)$ 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
$L_\rho(\beta,\theta,\mu) := f(\beta)+g(\theta)+\langle\mu, \mathbf{A}\beta+\mathbf{B}\theta-c\rangle+\frac{\rho}{2}||\mathbf{A}\beta+\mathbf{B}\theta-c||_2^2$

Lagrangian problem can still be decomposed into β and μ terms
this has nice algorithm where we can execute some stuff in parallel, because we can decompose the Lagrangian

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 !

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

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 !

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

Augmented: scalar product with ρ gets added,
Method of Multipliers: is a way to make the algorithm more robust
advantage: better convergence
disadvantage: no longer parallel execution of subtasks due to l2-term, no longer decomposable in beta and theta terms, as l2 norm dquares every entry of the vector
alternating direction: semi-decomposable, i.e. keeping one variable fixed while updating the other
ρ is step length of iterative algorithm
All notes on this slide: see the slides by [Boyd]

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

Dual Variable Update

Alternating Direction Method of Multipliers

$$\beta^{t+1} = \underset{\beta \in \mathbb{R}^m}{\arg\min}\, L_\rho(\beta, \theta^t, \mu^t)$$
$$\theta^{t+1} = \underset{\theta \in \mathbb{R}^m}{\arg\min}\, L_\rho(\beta^{t+1}, \theta, \mu^t)$$
$$\mu^{t+1} = \mu^t + \rho(\mathbf{A}\beta^{t+1} + \mathbf{B}\theta^{t+1} - c)$$

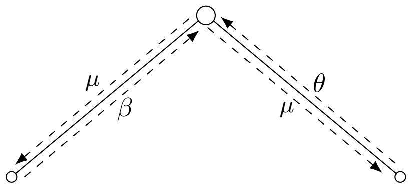
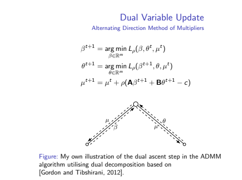


Figure: My own illustration of the dual ascent step in the ADMM algorithm utilising dual decomposition based on [Gordon and Tibshirani, 2012].



Method of Multipliers: is a way to make the algorithm more robust, (if in second line β^t statt β^{t+1})
alternating direction: semi-decomposable, i.e. keeping one variable fixed while updating the other
think of it as only the last line, sending μ to the updaters for β and θ
in this context ρ in last line can be thought of as "step length"

All notes on this slide: see the slides by [Boyd]

ADMM - Why?

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

ADMM - Why?

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

Details for blockwise computation in Exercise 5.12.

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
Problem
Problem in Lagrangian form
$\underset{\beta\in\mathbb{R}^p,\theta\in\mathbb{R}^p}{\text{minimize}}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|_2^2+\lambda\|\theta\|_1\right\}$ such that $\beta-\theta=0$
Augmented Lagrangian
$L_\rho(\beta,\theta,\mu):=\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{X}\beta\|_2^2+\lambda\|\theta\|_1\right\}+\langle\mu,\beta-\theta\rangle+\frac{\rho}{2}||\beta-\theta||_2^2$

In the problem, I can decompose into beta and theta terms, i.e.show $f(\beta)$ and $g(\theta)$
the problem itself and the constraints,
A and B are unit matrices here

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

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

$\mathcal{S}$ is a soft-thresholding parameter

Computational cost: Initially $\mathcal{O}(p^3)$, which is a lot, for the SVD(singular value decomposition of $\mathbf{X}$), after that comparable to coordinate descent or composite gradient from earlier

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 mgnitude

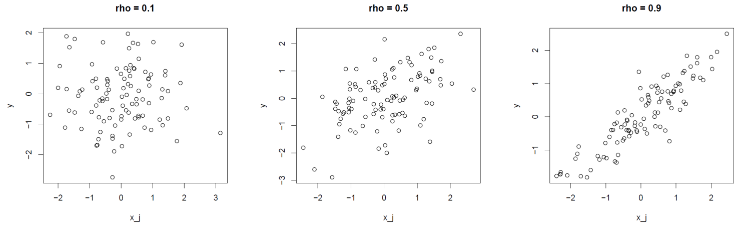
Screening Rules

- 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 mgnitude

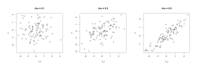
Imagine a big data set, a very big data set, with such a huge design matrix, that you cannot load it into memory (RAM).

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

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 - a different perspective I

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 $\left|x_j^T y\right|$ or better $\left|x_j^T(y - \hat{y}_\lambda)\right|$, 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 different perspective I

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

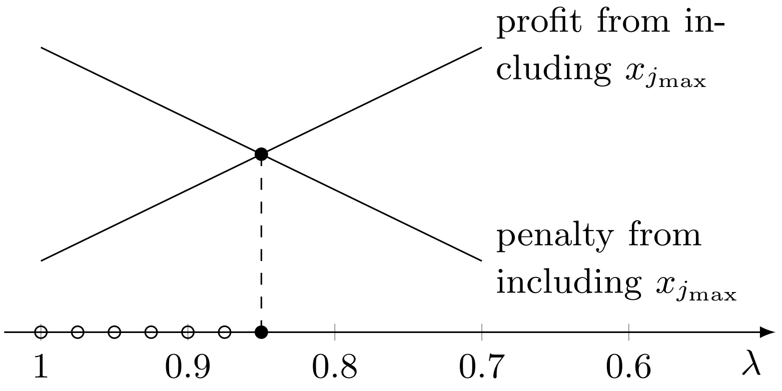
iterate

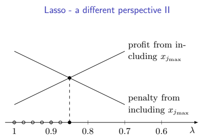
1. order predictors x_j not in $\mathcal{A}$ by their "effectiveness" using $\left|x_j^T y\right|$ or better $\left|x_j^T(y - \hat{y}_\lambda)\right|$, call the best predictor $x_{j_{\max}}$
2. move λ such that the positive effect from the best predictor $x_{j_{\max}}$ compensates the penalty by λ
3.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

this is just a formalisation of the previous slide

or better b/c we want to focus on the residuals once we have a preliminary solution

Lasso - a different perspective II





visualisation of step 2 of the previous slide
these lines are not linear, neither is usually the spacing on the lambda axis
we are walking along the lambda axis until we find a good point / the intersection between the penalty and the profit

Back to screening rules

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

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

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

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

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

Back to screening rules

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

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

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

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

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

for those wondering why in first equation not y-yhat? anybody?
yhat would come from the empty/intercept model, i.e. yhat=mean(y)
we assume standardised data (i.e. mean 0 and unit variance)
thus yhat = 0

Back to screening rules

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

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

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

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

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

Back to screening rules

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

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

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

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

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

this is the essential equation for screening rules, if, for a given lambda, a predictor does not fulfil this equation, we kick it out

SHOW R STUFF SCREENINGRULES 2

Global vs. Sequential

Global (one-time screening):
Suppose we want to calculate a lasso solution at $\lambda < \lambda_{\text{max}}$.

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

Global vs. Sequential
Global (one-time screening):
Suppose we want to calculate a lasso solution at $\lambda < \lambda_{\text{max}}$.
Sequential (iterative screening):
Suppose we have the lasso solution $\hat{\beta}(\lambda')$ at λ' and want to screen variables for solutions at $\lambda < \lambda'$.

There are two main classes of Screening rules

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

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

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 $\lambda < \lambda_{\text{max}}$. The global strong rule discards the j^{th} variable if

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

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

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

Global Strong Rule

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

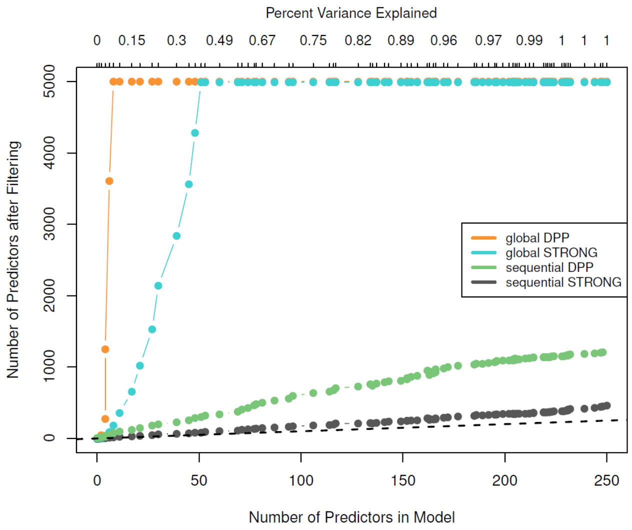
$$\left|\mathbf{x}_j^T \mathbf{y}\right| < \lambda - (\lambda_{\text{max}} - \lambda) = 2\lambda - \lambda_{\text{max}}$$

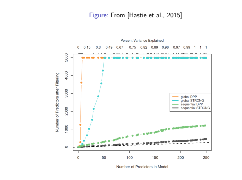
Sequential Strong Rule

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

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

Figure: From [Hastie et al., 2015]





Lasso regression: Results of different rules applied to a simulated dataset. There are N = 200 observations and p = 5000 uncorrelated Gaussian predictors; one-quarter of the true coefficients are nonzero. Shown are the number of predictors left after screening at each stage, plotted against the number of predictors in the model for a given value of λ. The value of λ is decreasing as we move from left to right. In the plots, we are fitting along a path of 100 decreasing λ values equally spaced on the log-scale, A broken line with unit slope is added for reference. The proportion of variance explained by the model is shown along the top of the plot. There were no violations for either of the strong rules.

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 analoguosly.

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 analoguosly.

MMA visually

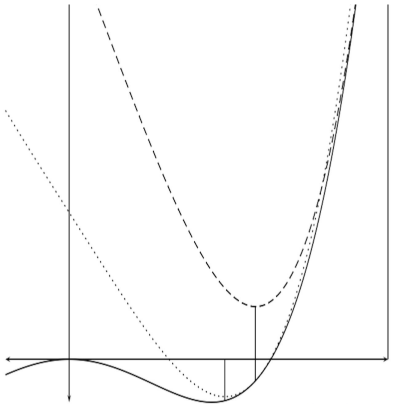


Figure: Figure from [de Leeuw, 2015]

MMA visually

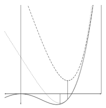


Figure: Figure from [de Leeuw, 2015]

MMA analytically I

Def. $\Psi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ 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 I

Def. $\Psi : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ 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

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

where

(i) & (iii) Definiton of majorize

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

MMA analytically II

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

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

where

(i) & (iii) Definiton of majorize

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

for inequalities: show previous slide

Biconvexity

Let's consider an example ...

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

Biconvexity
Let's consider an example ...
$f(\alpha, \beta) = (1 - \alpha\beta)^2$

Mathematica: `3D plot (1-xy)^2, x in [-2,2], y in [-2,2]`
The formula is a link.

Biconvexity

Let's consider an example ...

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

Def. A function $f(\alpha, \beta) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ 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. Analoguosly, a set $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{B}$, for $\mathcal{A}, \mathcal{B}$ convex sets, is called biconvex, if it is convex

Biconvexity

Let's consider an example ...

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

Def. A function $f(\alpha, \beta) : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ 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. Analoguosly, a set $\mathcal{C} \subseteq \mathcal{A} \times \mathcal{B}$, for $\mathcal{A}, \mathcal{B}$ convex sets, is called biconvex, if it is convex

Mathematica: `3D plot (1-xy)^2, x in [-2,2], y in [-2,2]`
The formula is a link.

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 $\alpha^{t+1} \in \underset{\alpha \in \mathcal{C}_{\beta^t}}{\arg\min} f(\alpha, \beta^t)$
   (ii) Fix $\alpha = \alpha^{t+1}$ and update $\beta^{t+1} \in \underset{\alpha \in \mathcal{C}_{\alpha^{t+1}}}{\arg\min} f(\alpha^{t+1}, \beta)$

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).

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 $\alpha^{t+1} \in \underset{\alpha \in \mathcal{C}_{\beta^t}}{\arg\min} f(\alpha, \beta^t)$
   (ii) Fix $\alpha = \alpha^{t+1}$ and update $\beta^{t+1} \in \underset{\alpha \in \mathcal{C}_{\alpha^{t+1}}}{\arg\min} f(\alpha^{t+1}, \beta)$

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 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)

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 . . .

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

That's it.
Thanks for listening.

Fill out your feedback sheets!

That's it.
Thanks for listening.
Fill out your feedback sheets!