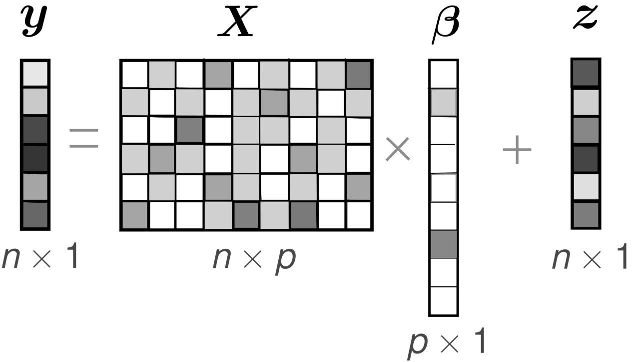
False Discovery Rate Control in High-dimensional Linear Regression

Weijie Su

University of Pennsylvania

Sparse high-dimensional linear regression



- Often $p > n$
- $\beta_j \neq 0$ means the jth variable is relevant and true
- Most of the coordinates of $\boldsymbol{\beta}$ are zero or close to zero

Model selection

$$\underset{n\times 1}{\boldsymbol{y}} = \underset{n\times p}{\boldsymbol{X}} \; \underset{p\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{z}}$$

- Interested in identifying which $\beta_j \neq 0$
- Context of multiple testing

$$H_j : \beta_j = 0 \quad j = 1, \ldots, p$$

- False discovery rate (FDR) control [Benjamini and Hochberg '95]

$$\text{FDR} \triangleq \mathbb{E}\left[\frac{\#\text{false discoveries}}{\#\text{discoveries}}\right] \leq q$$

  - #discoveries $= \#\{j : \widehat{\beta}_j \neq 0\}$, #false discoveries $= \#\{j : \widehat{\beta}_j \neq 0, \beta_j = 0\}$

- Reproducibility

An existing procedure: Lasso

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 \ + \ \underbrace{\lambda\|\boldsymbol{b}\|_1}_{\ell_1 \text{ norm}}$$

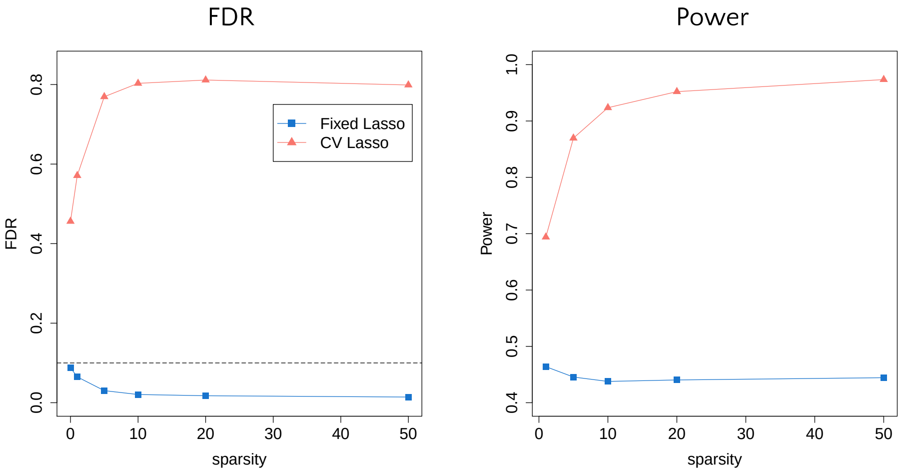
[Tibshirani '96]

An existing procedure: Lasso

$$\min_{\boldsymbol{b}} \; \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 \; + \; \underbrace{\lambda\|\boldsymbol{b}\|_1}_{\ell_1 \text{ norm}}$$

[Tibshirani '96]

A numerical example



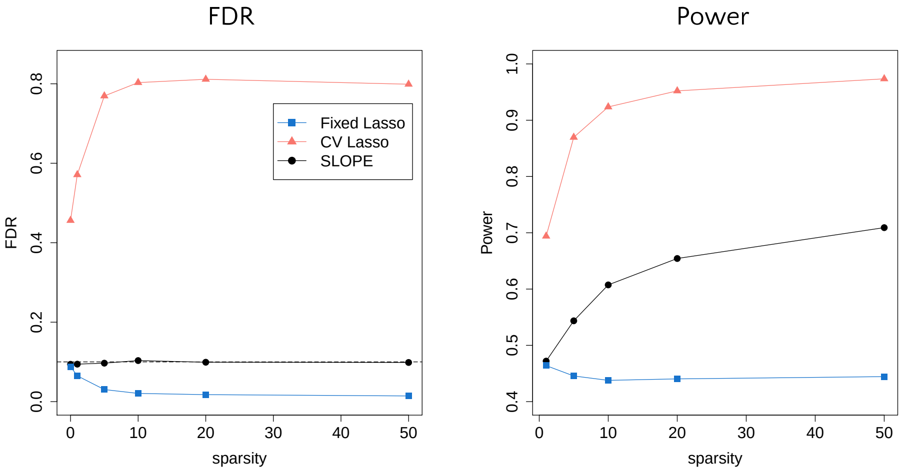
- Power $= \dfrac{\#\text{selected true variables}}{\#\text{all true variables}}$
- $q = 0.1, n = p = 5000$. Nonzero $\beta_j = \sqrt{2 \log p} \approx 4.13$ and $\sigma^2 = 1$.

A new procedure: SLOPE

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 \ + \ \underbrace{\lambda_1|b|_{(1)} + \cdots + \lambda_p|b|_{(p)}}_{\text{sorted } \ell_1 \text{ norm}}$$

[S. and Candès '16 (*Ann. Stat.*); Bogdan, Berg, Sabatti, S., and Candès '15 (*Ann. Appl. Stat.*)]

A numerical example



- $q = 0.1, n = p = 5000$. Nonzero $\beta_j = \sqrt{2 \log p} \approx 4.13$ and $\sigma^2 = 1$

Outline

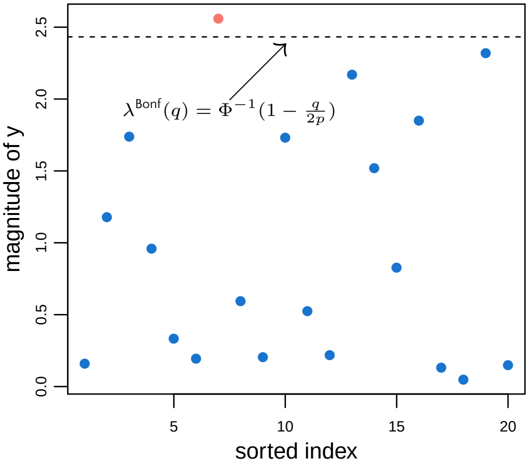
1. Deriving SLOPE with adaptive threshold
    - Inspiration from BH
    - Fast algorithm

2. FDR control
    - Exact in orthogonal design
    - Reasonable in general design

3. Estimation properties
    - Minimax over sparse signals
    - Cross-validation?

Bonferroni

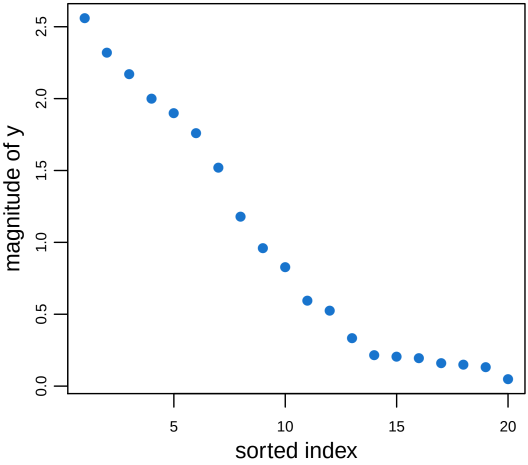
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p). \quad H_j : \beta_j = 0 \text{ for } j = 1, \dots, p$$



- Draw fixed threshold $\lambda^{\text{Bonf}} \triangleq \Phi^{-1}(1 - q/2p)$ (Φ is cdf of standard normal distribution)
- Reject H_j if $|y_j| \geq \lambda^{\text{Bonf}}$
- Control familywise error rate (probability of making one or more false discoveries)

Benjamini–Hochberg procedure

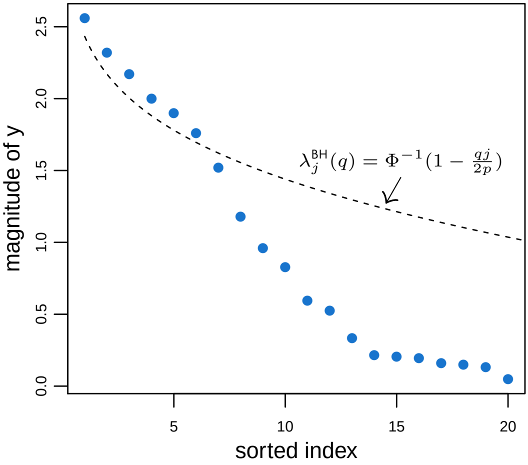
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p). \quad H_j : \beta_j = 0 \text{ for } j = 1, \ldots, p$$



- Sort $|y|_{(1)} \geq \cdots \geq |y|_{(p)}$

Benjamini–Hochberg procedure

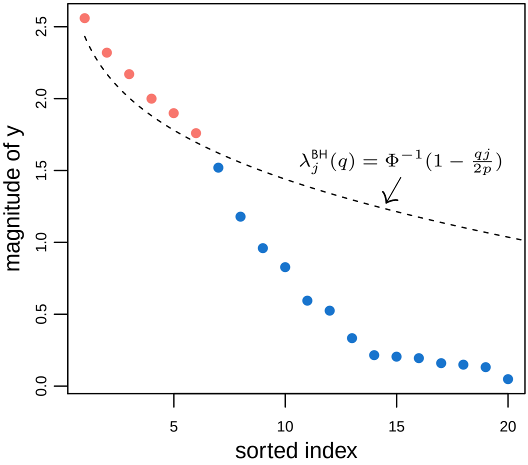
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p). \quad H_j : \beta_j = 0 \text{ for } j = 1, \ldots, p$$



- Sort $|y|_{(1)} \geq \cdots \geq |y|_{(p)}$
- Draw rank-dependent threshold $\lambda_j^{\mathrm{BH}} \triangleq \Phi^{-1}(1 - qj/2p)$

Benjamini–Hochberg procedure

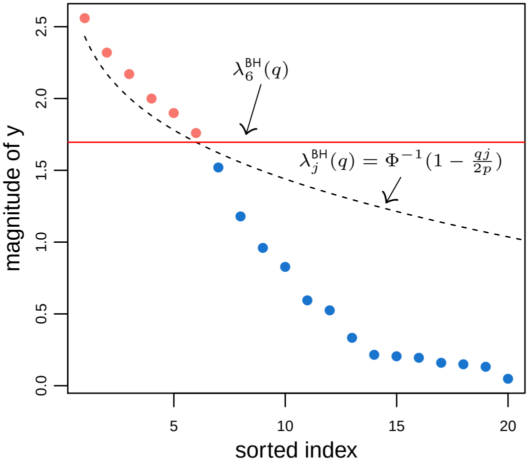
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p). \quad H_j : \beta_j = 0 \text{ for } j = 1, \dots, p$$



- Sort $|y|_{(1)} \geq \cdots \geq |y|_{(p)}$
- Draw rank-dependent threshold $\lambda_j^{\mathsf{BH}} \triangleq \Phi^{-1}(1 - qj/2p)$
- Reject all $H_{(j)}$ until last time $|y|_{(j)} \geq \lambda_j^{\mathsf{BH}}$

Benjamini–Hochberg procedure

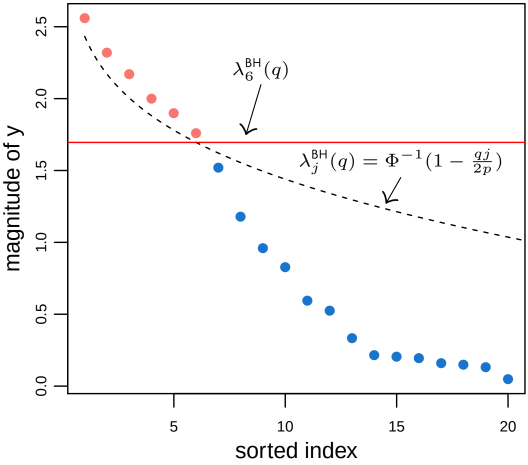
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p). \quad H_j : \beta_j = 0 \text{ for } j = 1, \ldots, p$$



- Sort $|y|_{(1)} \geq \cdots \geq |y|_{(p)}$
- Draw rank-dependent threshold $\lambda_j^{\mathrm{BH}} \triangleq \Phi^{-1}(1 - qj/2p)$
- Reject all $H_{(j)}$ until last time $|y|_{(j)} \geq \lambda_j^{\mathrm{BH}}$

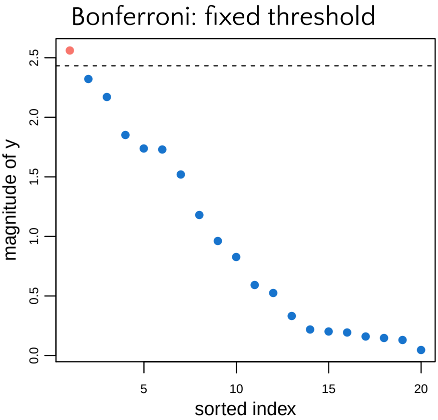
Benjamini–Hochberg procedure

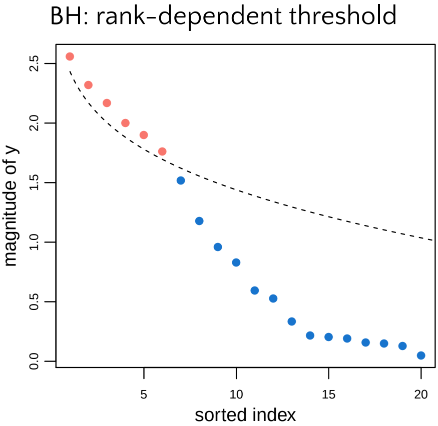
$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p). \quad H_j : \beta_j = 0 \text{ for } j = 1, \ldots, p$$



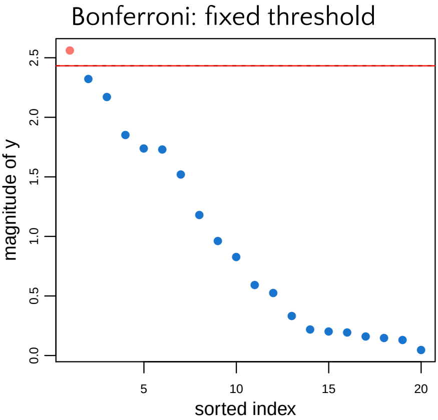
- Sort $|y|_{(1)} \geq \cdots \geq |y|_{(p)}$
- Draw rank-dependent threshold $\lambda_j^{\mathsf{BH}} \triangleq \Phi^{-1}(1 - qj/2p)$
- Reject all $H_{(j)}$ until last time $|y|_{(j)} \geq \lambda_j^{\mathsf{BH}}$
- Under independence $\mathrm{FDR} \leq q$

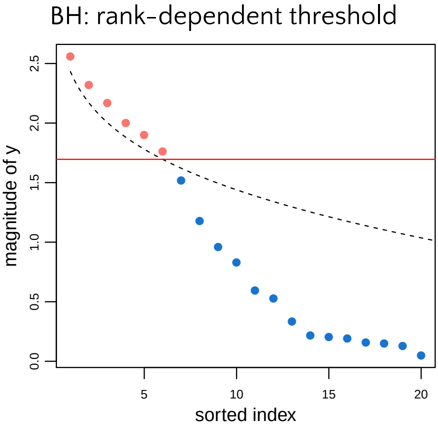
Bonferroni versus BH



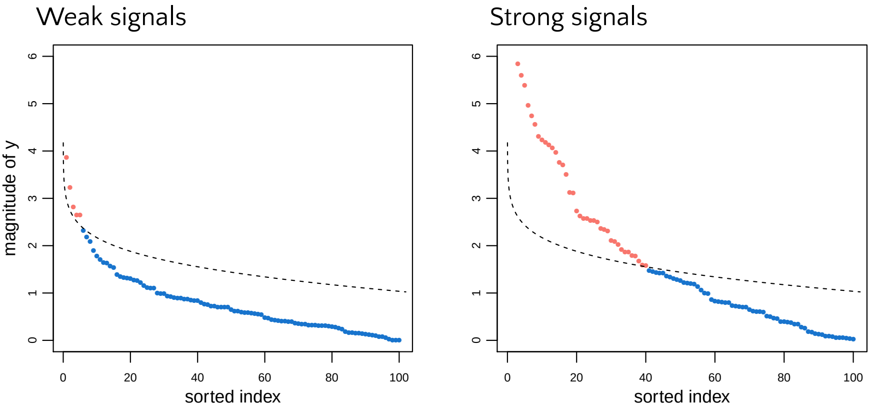


Bonferroni versus BH

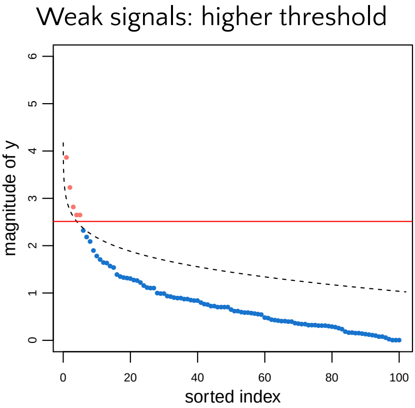


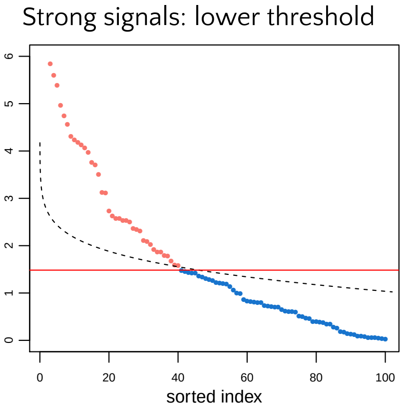


BH is adaptive



BH is adaptive





Effective cutoff is adaptive to strength of signals

How to Incorporate This

Adaptivity into Linear Regression?

Rank-dependent penalization

$\boldsymbol{y} = \boldsymbol{\beta} + \boldsymbol{z}. \quad x_+ = \max\{x, 0\}$

Bonferroni-style strategy: Lasso

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \lambda|b_1| + \cdots + \lambda|b_p|$$

- $\widehat{\beta}_j = \operatorname{sgn}(y_j) \cdot (|y_j| - \lambda)_+$

Rank-dependent penalization

$\boldsymbol{y} = \boldsymbol{\beta} + \boldsymbol{z}$. $\quad x_+ = \max\{x, 0\}$

Bonferroni-style strategy: Lasso

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \lambda|b_1| + \cdots + \lambda|b_p|$$

- $\widehat{\beta}_j = \operatorname{sgn}(y_j) \cdot (|y_j| - \lambda)_+$

BH-style strategy

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \underbrace{\lambda_1|b|_{(1)} + \lambda_2|b|_{(2)} + \cdots + \lambda_p|b|_{(p)}}_{\text{symmetric in } \boldsymbol{b}}$$

- $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$
- $|b|_{(1)} \geq \cdots \geq |b|_{(p)}$: order statistics of $|\boldsymbol{b}|$
- If $|y_1| \gg \cdots \gg |y_p|$, then $\widehat{\beta}_j = \operatorname{sgn}(y_j) \cdot (|y_j| - \lambda_j)_+$

Rank-dependent penalization

$\boldsymbol{y} = \boldsymbol{\beta} + \boldsymbol{z}. \quad x_+ = \max\{x, 0\}$

Bonferroni-style strategy: Lasso

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \lambda|b_1| + \cdots + \lambda|b_p|$$

- $\widehat{\beta}_j = \operatorname{sgn}(y_j) \cdot (|y_j| - \lambda)_+$

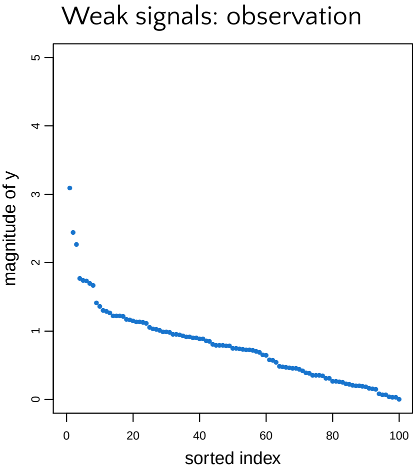
BH-style strategy

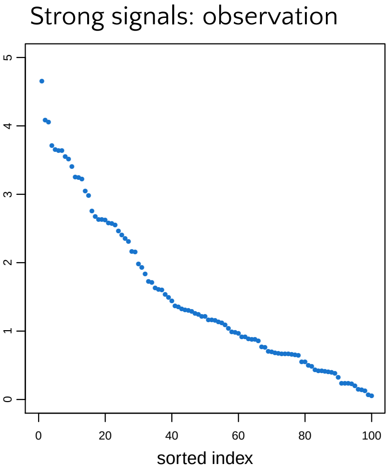
$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \underbrace{\lambda_1|b|_{(1)} + \lambda_2|b|_{(2)} + \cdots + \lambda_p|b|_{(p)}}_{\text{symmetric in } \boldsymbol{b}}$$

- $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$
- $|b|_{(1)} \geq \cdots \geq |b|_{(p)}$: order statistics of $|\boldsymbol{b}|$
- If $|y_1| \gg \cdots \gg |y_p|$, then $\widehat{\beta}_j = \operatorname{sgn}(y_j) \cdot (|y_j| - \lambda_j)_+$

Effective threshold is adaptive

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{b}\|_2^2 + \lambda_1|b|_{(1)} + \cdots + \lambda_p|b|_{(p)}, \quad \lambda_j = \lambda_j^{\mathsf{BH}} \equiv \Phi^{-1}(1-qj/2p)$$

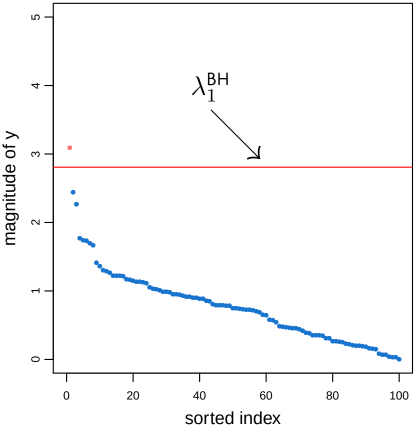




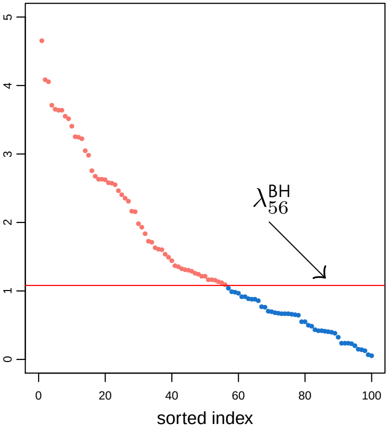
Effective threshold is adaptive

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{b}\|_2^2+\lambda_1|b|_{(1)}+\cdots+\lambda_p|b|_{(p)}, \quad \lambda_j=\lambda_j^{\mathsf{BH}}\equiv\Phi^{-1}(1-qj/2p)$$

Weak signals: higher threshold

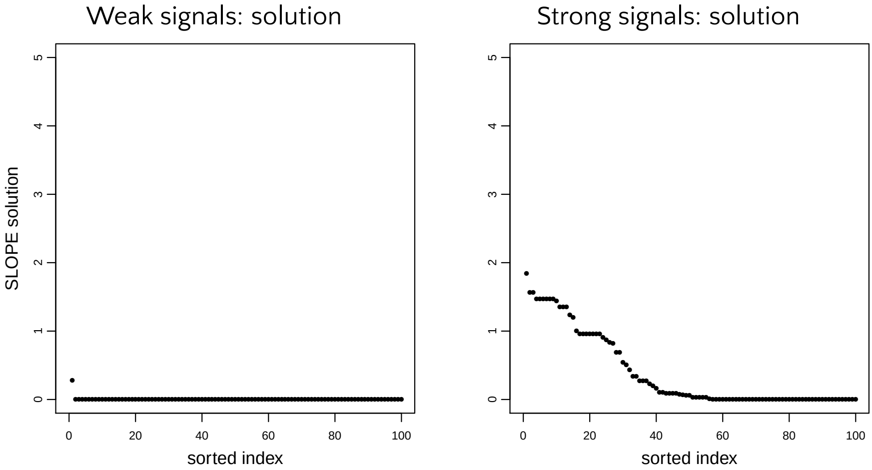


Strong signals: lower threshold



Effective threshold is adaptive

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{b}\|_2^2 + \lambda_1|b|_{(1)} + \cdots + \lambda_p|b|_{(p)}, \quad \lambda_j = \lambda_j^{\mathsf{BH}} \equiv \Phi^{-1}(1 - qj/2p)$$



BH-style strategy: SLOPE

$\boldsymbol{y} = \boldsymbol{\beta} + \boldsymbol{z}$

SLOPE: Sorted ℓ-One Penalized Estimation

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \lambda_1 |b|_{(1)} + \lambda_2 |b|_{(2)} + \cdots + \lambda_p |b|_{(p)}$$

BH-style strategy: SLOPE

$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z}$

SLOPE: Sorted ℓ-One Penalized Estimation

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 + \lambda_1|b|_{(1)} + \lambda_2|b|_{(2)} + \cdots + \lambda_p|b|_{(p)}$$

BH-style strategy: SLOPE

$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z}$

SLOPE: Sorted ℓ-One Penalized Estimation

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 + \lambda_1 |b|_{(1)} + \lambda_2 |b|_{(2)} + \cdots + \lambda_p |b|_{(p)}$$

- $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ given by $\lambda_j^{\mathsf{BH}} \equiv \Phi^{-1}(1 - qj/2p)$ or close
- $|b|_{(1)} \geq \cdots \geq |b|_{(p)}$: order statistics of $|\boldsymbol{b}|$
- Less penalization for smaller coefficients

BH-style strategy: SLOPE

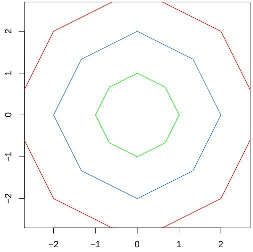
$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z}$

SLOPE: Sorted ℓ-One Penalized Estimation

$$\min_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 + J_{\boldsymbol{\lambda}}(\boldsymbol{b})$$

- $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$ given by $\lambda_j^{\text{BH}} \equiv \Phi^{-1}(1 - qj/2p)$ or close
- $|b|_{(1)} \geq \cdots \geq |b|_{(p)}$: order statistics of $|\boldsymbol{b}|$
- Less penalization for smaller coefficients
- $J_{\boldsymbol{\lambda}}(\boldsymbol{b}) \triangleq \lambda_1 |b|_{(1)} + \cdots + \lambda_p |b|_{(p)}$ is (sorted ℓ_1) norm, so is convex
    - $\lambda_1 = \cdots = \lambda_p$, then ℓ_1 norm
    - $\lambda_2 = \cdots = \lambda_p = 0$, then ℓ_∞ norm

Level curves $\lambda_1 = 2, \lambda_2 = 1$



How to Solve SLOPE?

SLOPE as convex optimization

$$\min_{\boldsymbol{b}} \; f(\boldsymbol{b}) \triangleq \underbrace{\tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth but convex}}$$

- Proximal gradient descent: $O(1/k)$

SLOPE as convex optimization

$$\min_{\boldsymbol{b}} \; f(\boldsymbol{b}) \triangleq \underbrace{\tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth but convex}}$$

- Proximal gradient descent: $O(1/k)$
- Nesterov's accelerated schemes: $O(1/k^2)$

[Nesterov '83; Beck and Teboulle '09; Nesterov '13]

Empirical convergence

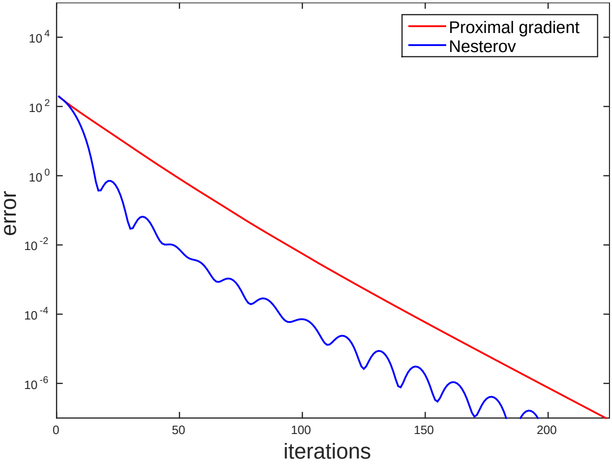


Figure: Solving SLOPE under 1000×10000 Gaussian design

Empirical convergence

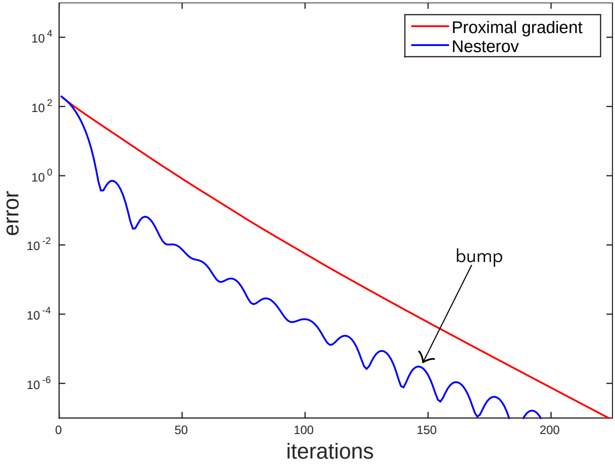


Figure: Solving SLOPE under 1000×10000 Gaussian design

Improve Nesterov's scheme by restarting

- Nesterov's scheme is *damping with decaying friction*
- Suggest a new restarting scheme

[S., Boyd, and Candès '16 (*JMLR*)]

Improve Nesterov's scheme by restarting

- Nesterov's scheme is *damping with decaying friction*
- Suggest a new restarting scheme

Theorem (S., Boyd, and Candès)

If objective function is strongly convex and smooth, then restarting Nesterov's scheme has exponential convergence

[S., Boyd, and Candès '16 (*JMLR*)]

Empirical convergence

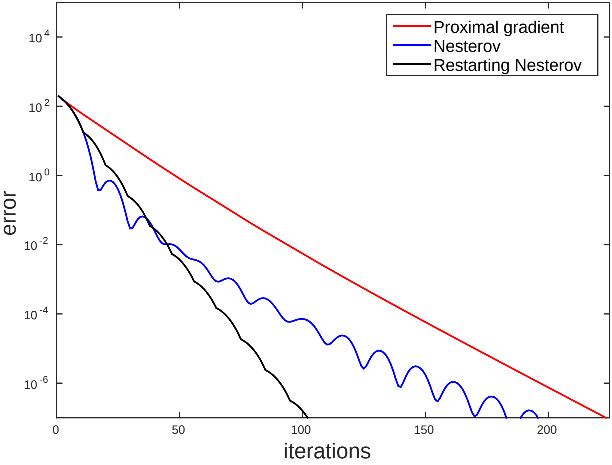


Figure: Solving SLOPE under 1000×10000 Gaussian design

Outline

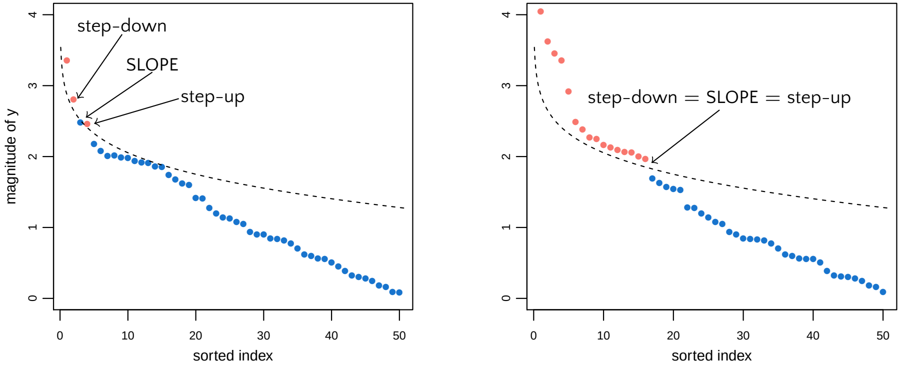
1. Deriving SLOPE with adaptive threshold
    - Inspiration from BH
    - Fast algorithm

2. FDR control
    - Exact in orthogonal design
    - Reasonable in general design

3. Estimation properties
    - Minimax over sparse signals
    - Cross-validation?

Connection with BH

SLOPE under $\boldsymbol{X} = \boldsymbol{I}_p$ with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathsf{BH}}(q)$

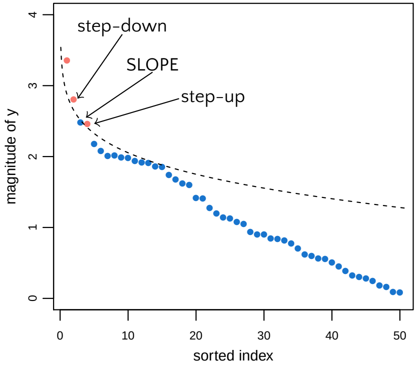


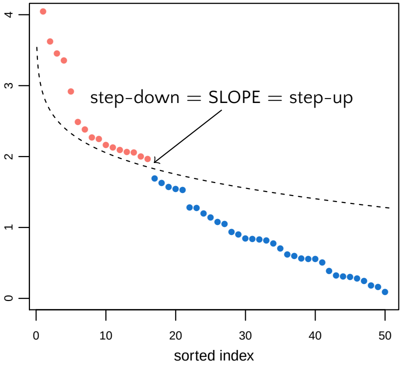
Sandwiching

$$R_{\text{step-down}} \le R_{\text{SLOPE}} \le R_{\text{step-up}}, \quad R_{\text{SLOPE}} = \#\{j : \widehat{\beta}_{\text{SLOPE}}(j) \neq 0\}$$

Connection with BH

SLOPE under $\boldsymbol{X} = \boldsymbol{I}_p$ with $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathsf{BH}}(q)$





Provable FDR control of SLOPE

$$\mathsf{FDR} \equiv \mathbb{E}\left[\frac{\#\text{false discoveries}}{\#\text{discoveries}}\right] \le \frac{qp_0}{p} \le q$$

$p_0 \triangleq \#\{1 \le j \le p : \beta_j = 0\}$ is number of true nulls

FDR control in general designs

Adjust weights $\boldsymbol{\lambda}$ based on BH cutoffs

- Variance inflation
- $\lambda_j = (1 + \omega_j)\lambda_j^{\mathsf{BH}}, \omega_j \geq 0$
- Active research area

Empirical FDR Control

FDR control in Gaussian design

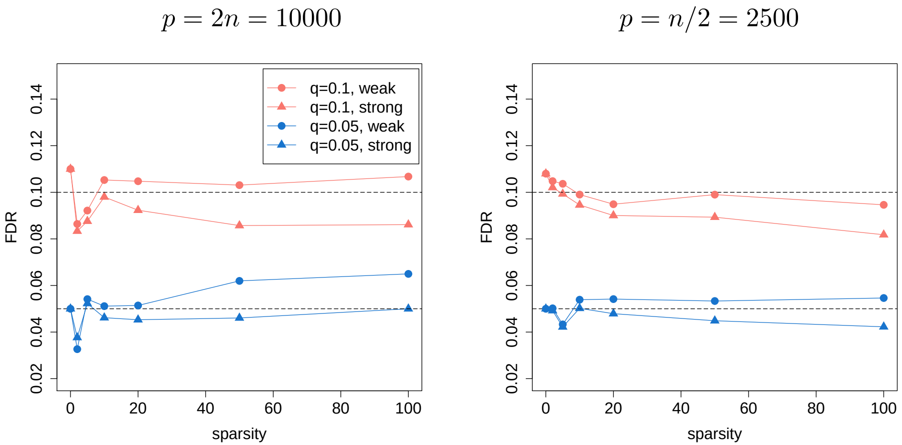


Figure: Strong signals have nonzero coefficients set to $5\sqrt{2\log p}$, $\sqrt{2\log p}$ for weak signals and variance $\sigma^2 = 1$. Average over 500 replicates

SLOPE with unknown variance

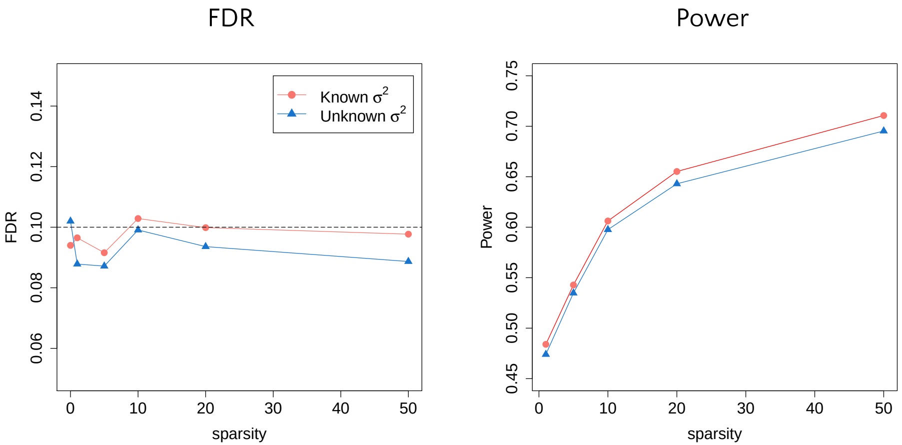


Figure: SLOPE w/ or w/o knowing $\sigma^2 = 1$. Nominal level $q = 0.1$ and $n = p = 5000$. Nonzero $\beta_j = \sqrt{2 \log p} \approx 4.13$. Over 500 replicates

Variability of false discovery proportion (FDP)

$$\mathrm{FDP} \triangleq \frac{\#\text{false discoveries}}{\#\text{discoveries}}$$

- FDP is realization of FDR
- Low variability of FDP is appreciated

Multiple testing in correlated noise

$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$. Suppose $\boldsymbol{\Sigma}$ is known

- BH: apply to $\boldsymbol{y}$
- SLOPE can incorporate $\boldsymbol{\Sigma}$:

$$\underbrace{\boldsymbol{\Sigma}^{-1/2}\boldsymbol{y}}_{\text{new } \boldsymbol{y}} = \underbrace{\boldsymbol{\Sigma}^{-1/2}}_{\boldsymbol{X}}\,\boldsymbol{\beta} + \boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$$

Multiple testing in correlated noise

$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{\Sigma})$. Suppose $\boldsymbol{\Sigma}$ is known

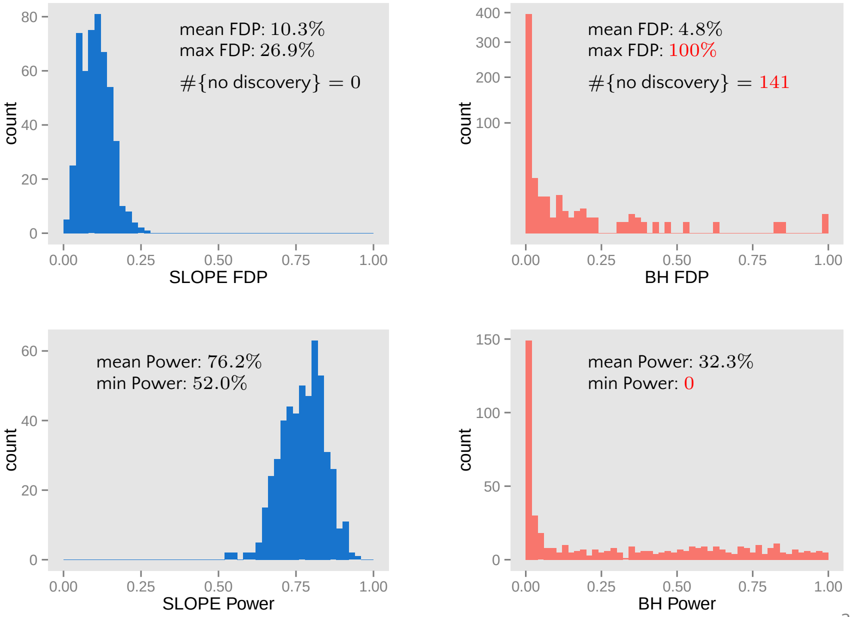
- BH: apply to $\boldsymbol{y}$
- SLOPE can incorporate $\boldsymbol{\Sigma}$:

$$\underbrace{\boldsymbol{\Sigma}^{-1/2}\boldsymbol{y}}_{\text{new } \boldsymbol{y}} = \underbrace{\boldsymbol{\Sigma}^{-1/2}}_{\boldsymbol{X}} \boldsymbol{\beta} + \boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$$

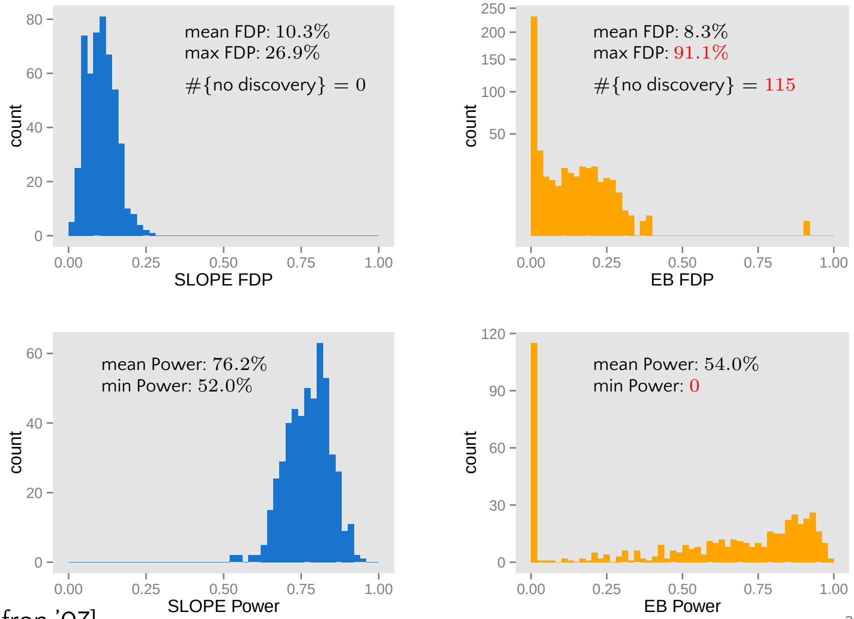
Setting

- $q = 0.1, p = 1000, s = 50$ and 500 replicates
- $\Sigma_{ij} = \begin{cases} 1 & i = j \\ 0.5 & i \neq j \end{cases}$
- $\beta_j = \begin{cases} \sqrt{2 \log p} = 3.717 & 1 \leq j \leq s \\ 0 & s < j \leq p \end{cases}$

Comparison with BH



Comparison with empirical Bayes approach



[Efron ’07]

A real-data example of $\boldsymbol{X}$

- 1000 individuals from admixture of the African-American and European populations
- 892 markers distributed over all chromosomes
- $X_{ij} \in \{0, 1, 2\}$ is the number of copies of ref. allele at marker j for individual i
- Simulate $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z}$ with nonzero $\beta_j = \sqrt{2 \log p} = 3.69$ and noise variance $\sigma^2 = 1$

Data source: The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. *Nature*, 449:851–862, 2007

Comparison with BHs

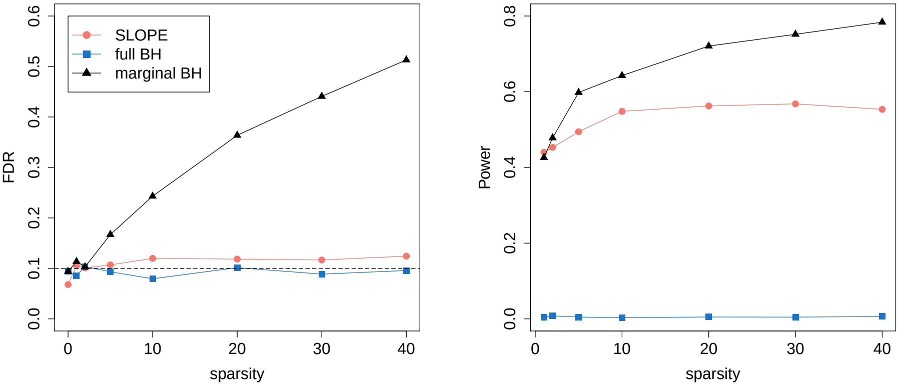


Figure: $q = 0.1$. Over 500 replicates

- **Marginal** (apply BH to univariate regression): *fail to control FDR*
- **Full** (apply BH to LS estimate): *low power*

Outline

1. Deriving SLOPE with adaptive threshold
    - Inspiration from BH
    - Fast algorithm

2. FDR control
    - Exact in orthogonal design
    - Reasonable in general design

3. Estimation properties
    - Minimax over sparse signals
    - Cross-validation?

Balance between bias and variance

Threshold	Control	Bias	Variance	MSE
High	FWER	High	Low	High
Low	No	Low	High	High
Adaptive	FDR	Low	Low	Low?

- MSE $=$ Bias^2 $+$ Variance
- FDR-thresholding [Abramovich, Benjamini, Donoho, and Johnstone '05]

SLOPE is minimax under Gaussian design

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

SLOPE is minimax under Gaussian design

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

- Sparsity class $\ell_0(s) \triangleq \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_0 \leq s\}$ ($\|\cdot\|_0$ denotes #nonzero entries)
- Weights $(1 + \epsilon/3)\boldsymbol{\lambda}^{\mathsf{BH}}(q)$ for any fixed $0 < \epsilon, q < 1$

SLOPE is minimax under Gaussian design

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

- Sparsity class $\ell_0(s) \triangleq \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_0 \le s\}$ ($\|\cdot\|_0$ denotes #nonzero entries)
- Weights $(1+\epsilon/3)\boldsymbol{\lambda}^{\mathsf{BH}}(q)$ for any fixed $0 < \epsilon, q < 1$

Theorem (S. and Candès)

If $s/p \to 0$ *and* $(s \log p)/n \to 0$, *then*

SLOPE is minimax under Gaussian design

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

- Sparsity class $\ell_0(s) \triangleq \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_0 \le s\}$ ($\|\cdot\|_0$ denotes #nonzero entries)
- Weights $(1+\epsilon/3)\boldsymbol{\lambda}^{\text{BH}}(q)$ for any fixed $0 < \epsilon, q < 1$

Theorem (S. and Candès)

If $s/p \to 0$ and $(s\log p)/n \to 0$, then

- $$\inf_{\widehat{\boldsymbol{\beta}}} \sup_{\boldsymbol{\beta}\in\ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 > (1-\epsilon)\cdot R(s)\right) \longrightarrow 1 \text{ (lower bound)}$$

Here $R(s) = 2s\log(p/s)$

- Minimax (probabilistic) risk is at least $(1 - o(1)) \cdot R(s)$

SLOPE is minimax under Gaussian design

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

- Sparsity class $\ell_0(s) \triangleq \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_0 \le s\}$ ($\|\cdot\|_0$ denotes #nonzero entries)
- Weights $(1+\epsilon/3)\boldsymbol{\lambda}^{\mathsf{BH}}(q)$ for any fixed $0 < \epsilon, q < 1$

Theorem (S. and Candès)

If $s/p \to 0$ *and* $(s \log p)/n \to 0$*, then*

- $\inf_{\widehat{\boldsymbol{\beta}}} \sup_{\boldsymbol{\beta} \in \ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 > (1-\epsilon) \cdot R(s) \right) \longrightarrow 1$ *(lower bound)*
- $\sup_{\boldsymbol{\beta} \in \ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}}_{\mathsf{SLOPE}} - \boldsymbol{\beta}\|_2^2 > (1+\epsilon) \cdot R(s) \right) \longrightarrow 0$ *(upper bound)*

Here $R(s) = 2s\log(p/s)$

- Minimax (probabilistic) risk is at least $(1 - o(1)) \cdot R(s)$
- SLOPE has worst case (probabilistic) risk at most $(1 + o(1)) \cdot R(s)$

Challenges in getting upper bound

$$\sup_{\boldsymbol{\beta} \in \ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}}_{\mathsf{SLOPE}} - \boldsymbol{\beta}\|_2^2 > (1+\epsilon) \cdot R(s) \right) \longrightarrow 0$$

Existing techniques in high-dimensional sparse regression are not applicable

Challenges in getting upper bound

$$\sup_{\boldsymbol{\beta} \in \ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}}_{\text{SLOPE}} - \boldsymbol{\beta}\|_2^2 > (1+\epsilon) \cdot R(s) \right) \longrightarrow 0$$

Existing techniques in high-dimensional sparse regression are not applicable

- Asymptotically exact minimax

  ⟶ many techniques are not sharp

Challenges in getting upper bound

$$\sup_{\boldsymbol{\beta} \in \ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}}_{\text{SLOPE}} - \boldsymbol{\beta}\|_2^2 > (1+\epsilon) \cdot R(s) \right) \longrightarrow 0$$

Existing techniques in high-dimensional sparse regression are not applicable

- Asymptotically exact minimax

  ⟶ many techniques are not sharp

- Sorted ℓ_1 norm is not decomposable

  ⟶ KKT conditions are complicated

Challenges in getting upper bound

$$\sup_{\boldsymbol{\beta} \in \ell_0(s)} \mathbb{P}\left(\|\widehat{\boldsymbol{\beta}}_{\text{SLOPE}} - \boldsymbol{\beta}\|_2^2 > (1+\epsilon) \cdot R(s) \right) \longrightarrow 0$$

Existing techniques in high-dimensional sparse regression are not applicable

- Asymptotically exact minimax

  ⟶ many techniques are not sharp

- Sorted ℓ_1 norm is not decomposable

  ⟶ KKT conditions are complicated

- False discoveries are allowed

  ⟶ solution support is unknown a priori

Minimax by adaptive threshold

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$$

Theorem (two-line version)

If $s/p \to 0$ and $(s \log p)/n \to 0$, SLOPE with weights $\boldsymbol{\lambda}^{\mathsf{BH}}$ achieves minimax risk $2s\log(p/s)$ over s-sparsity ball

Minimax by adaptive threshold

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

Theorem (two-line version)

If $s/p \to 0$ and $(s \log p)/n \to 0$, SLOPE with weights $\boldsymbol{\lambda}^{\mathsf{BH}}$ achieves minimax risk $2s\log(p/s)$ over s-sparsity ball

- Possibly $p \gg n$. e.g. $n = p^{0.75}, s = p^{0.5}$

Minimax by adaptive threshold

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

Theorem (two-line version)

If $s/p \to 0$ and $(s \log p)/n \to 0$, SLOPE with weights $\boldsymbol{\lambda}^{\text{BH}}$ achieves minimax risk $2s \log(p/s)$ over s-sparsity ball

- Possibly $p \gg n$. e.g. $n = p^{0.75}$, $s = p^{0.5}$
- Lasso with $\lambda = (1 + o(1))\lambda_1^{\text{BH}} \approx \sqrt{2 \log p}$ has worst case risk $2s \log p$

Minimax by adaptive threshold

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

Theorem (two-line version)

If $s/p \to 0$ and $(s \log p)/n \to 0$, SLOPE with weights $\boldsymbol{\lambda}^{\mathsf{BH}}$ achieves minimax risk $2s\log(p/s)$ over s-sparsity ball

- Possibly $p \gg n$. e.g. $n = p^{0.75}$, $s = p^{0.5}$
- Lasso with $\lambda = (1 + o(1))\lambda_1^{\mathsf{BH}} \approx \sqrt{2\log p}$ has worst case risk $2s\log p$
- Lasso is minimax with $\lambda = (1 + o(1))\lambda_s^{\mathsf{BH}} \approx \underbrace{\sqrt{2\log(p/s)}}_{\text{need to know sparsity!}}$

Minimax by adaptive threshold

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$$

Theorem (two-line version)

If $s/p \to 0$ and $(s \log p)/n \to 0$, SLOPE with weights $\boldsymbol{\lambda}^{\text{BH}}$ achieves minimax risk $2s\log(p/s)$ over s-sparsity ball

- Possibly $p \gg n$. e.g. $n = p^{0.75}$, $s = p^{0.5}$
- Lasso with $\lambda = (1 + o(1))\lambda_1^{\text{BH}} \approx \sqrt{2\log p}$ has worst case risk $2s\log p$
- Lasso is minimax with $\lambda = (1 + o(1))\lambda_s^{\text{BH}} \approx \underbrace{\sqrt{2\log(p/s)}}_{\text{need to know sparsity!}}$

Can Cross-Validation

Adapt to Unknown Sparsity?

Comparison with a data-driven procedure

Lasso under identity design

C. Stein ('56)

$\mathrm{SURE}(\lambda)$ is an unbiased estimator of $\mathbb{E}\|\widehat{\boldsymbol{\beta}}_{\text{Lasso}}(\lambda) - \boldsymbol{\beta}\|_2^2$ for every λ

SureShrink [Donoho and Johnstone '95]

- Select λ by minimizing $\mathrm{SURE}(\lambda)$
- Cross-validation flavor

Comparison with a data-driven procedure

In situations of extreme sparsity, SureShrink is unstable

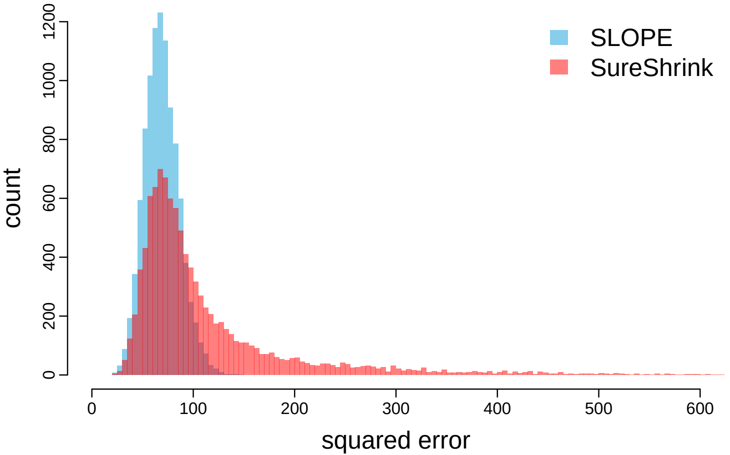


Figure: Sparsity is 1. SLOPE uses $\boldsymbol{\lambda}^{\mathrm{BH}}(0.5)$. Over 10000 replicates

Comparison with a data-driven procedure

In situations of extreme sparsity, SureShrink is unstable

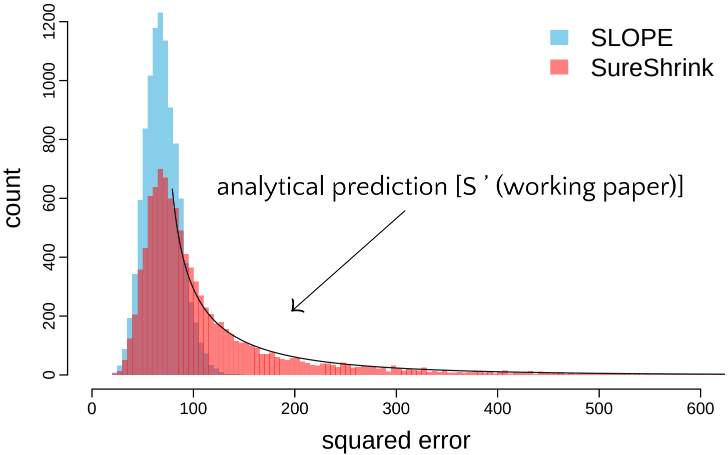


Figure: Sparsity is 1. SLOPE uses $\boldsymbol{\lambda}^{\mathrm{BH}}(0.5)$. Over 10000 replicates

Connection with FDR thresholding

$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p)$

FDR-thresholding: perform BH and let

$$\widehat{\beta}_{\mathsf{FDR}}(j) = \begin{cases} y_j & \text{if } H_j : \beta_j = 0 \text{ is rejected} \\ 0 & \text{if } H_j : \beta_j = 0 \text{ is accepted} \end{cases}$$

FDR thresholding

Hard-rule with adaptive cutoff

- $\log^5 p \le s \le p^{1-\delta}$, constant $\delta > 0$
- $0 < q \le 1/2$

Minimax over s-sparsity ball

[Abramovich, Benjamini, Donoho, Johnstone '05]

SLOPE under orthogonal design

Soft-rule with adaptive cutoff

- $s/p \to 0$
- Use $\boldsymbol{\lambda}^{\mathsf{BH}}(q)$ with $0 < q < 1$

Minimax over s-sparsity ball

Simple corollary of Gaussian design case

Connection with FDR thresholding

$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\beta}, \boldsymbol{I}_p)$

FDR-thresholding: perform BH and let

$$\widehat{\beta}_{\text{FDR}}(j) = \begin{cases} y_j & \text{if } H_j : \beta_j = 0 \text{ is rejected} \\ 0 & \text{if } H_j : \beta_j = 0 \text{ is accepted} \end{cases}$$

FDR thresholding

Hard-rule with adaptive cutoff

- $\log^5 p \le s \le p^{1-\delta}$, constant $\delta > 0$
- $0 < q \le 1/2$

Minimax over s-sparsity ball

[Abramovich, Benjamini, Donoho, Johnstone '05]

Only identity design matrix

SLOPE under orthogonal design

Soft-rule with adaptive cutoff

- $s/p \to 0$
- Use $\boldsymbol{\lambda}^{\text{BH}}(q)$ with $0 < q < 1$

Minimax over s-sparsity ball

Simple corollary of Gaussian design case

General design matrix

Connection with ℓ_0-penalized MLE

$$\min_{\boldsymbol{b}} \ \|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \mathrm{pen}(\|\boldsymbol{b}\|_0), \quad \mathrm{pen}(\|\boldsymbol{b}\|_0) = \sum_{j=1}^{\|\boldsymbol{b}\|_0} t_j^2$$

Constant

- Mallows' $\mathcal{C}_p$ and AIC: $t_j = \sqrt{2}$
- BIC: $t_j = \sqrt{\log n}$
- RIC: $t_j = \sqrt{2 \log p}$

Adaptive

- $t_j = \sqrt{2j \log \frac{p}{j} - 2(j-1) \log \frac{p}{j-1}}$
- $t_j = \sqrt{2 \log(p/j)}$
- $t_j = \lambda_j^{\mathrm{BH}}$ (FDR thresholding)

- All three adaptive t_j are equivalent for small j

[Foster and George '94; Abramovich and Benjamini '96; Tibshirani and Knight '99; Foster and Stine '99; Birgé and Massart '01; Abramovich et al '05; Wu and Zhou '13] (highly incomplete)

Connection with ℓ_0-penalized MLE

$$\min_{\boldsymbol{b}} \ \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2 + \mathrm{pen}(\|\boldsymbol{b}\|_0), \quad \mathrm{pen}(\|\boldsymbol{b}\|_0) = \sum_{j=1}^{\|\boldsymbol{b}\|_0} t_j^2$$

Constant

- Mallows' C_p and AIC: $t_j = \sqrt{2}$
- BIC: $t_j = \sqrt{\log n}$
- RIC: $t_j = \sqrt{2 \log p}$

Adaptive

- $t_j = \sqrt{2j \log \frac{p}{j} - 2(j-1) \log \frac{p}{j-1}}$
- $t_j = \sqrt{2 \log(p/j)}$
- $t_j = \lambda_j^{\mathrm{BH}}$ (FDR thresholding)

- All three adaptive t_j are equivalent for small j
- Computationally intractable for general design matrix $\boldsymbol{X}$

[Foster and George '94; Abramovich and Benjamini '96; Tibshirani and Knight '99; Foster and Stine '99; Birgé and Massart '01; Abramovich et al '05; Wu and Zhou '13] (highly incomplete)

Concluding Remarks

Extensions based on sorted ℓ_1 norm

Recent work

- Group SLOPE to account for strong correlations [Gossmann, Cao, and Wang '15; Brzyski, Gossman, S., and Bogdan '16]
- Square root SLOPE [Stucky and van de Geer '15]
- Sorted ℓ_1 Dantzig selector [Lee, Brzyski, and Bogdan '15]
- SLOPE with permissible region constraint to reconstruct fluorescence targets [He, Dong, Yu, Guo, and Hou '15]

Extensions based on sorted ℓ_1 norm

Recent work

- Group SLOPE to account for strong correlations [Gossmann, Cao, and Wang '15; Brzyski, Gossman, S., and Bogdan '16]
- Square root SLOPE [Stucky and van de Geer '15]
- Sorted ℓ_1 Dantzig selector [Lee, Brzyski, and Bogdan '15]
- SLOPE with permissible region constraint to reconstruct fluorescence targets [He, Dong, Yu, Guo, and Hou '15]

Future extensions

- Graphical SLOPE?
- Sorted ℓ_1 regularized logistic regression?
- Other sorted ℓ_1 regularized GLM?
- Sorted nuclear norm in matrix completion?

Summary

- **FDR control**
  - *Goal*
  - Controlled at reasonable level

- **Adaptive threshold**
  - *Sorted ℓ_1 norm*
  - High for weak signals and low for strong

- **Good estimation**
  - *Bonus*
  - Minimax under Gaussian designs

Summary

- **FDR control**: balance true and false discoveries
  - *Goal*
  - Controlled at reasonable level
- **Adaptive threshold**: balance signal and noise
  - *Sorted ℓ_1 norm*
  - High for weak signals and low for strong
- **Good estimation**: balance bias and variance
  - *Bonus*
  - Minimax under Gaussian designs

Summary

- **FDR control**: balance true and false discoveries
  - *Goal*
  - Controlled at reasonable level

- **Adaptive threshold**: balance signal and noise
  - *Sorted ℓ_1 norm*
  - High for weak signals and low for strong

- **Good estimation**: balance bias and variance
  - *Bonus*
  - Minimax under Gaussian designs

Thank You!

References

R package: `https://cran.r-project.org/web/packages/SLOPE`

- SLOPE is adaptive to unknown sparsity and asymptotically minimax. S. and Candès '16, Annals of Statistics
- SLOPE – adaptive variable selection via convex optimization. Bogdan, Berg, Sabatti, S., and Candès '15, Annals of Applied Statistics
- A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. S., Boyd, and Candès '16, Journal of Machine Learning Research
- Approximating Stein's unbiased risk estimate by drifted Brownian motion. S. '16, coming soon
- Group SLOPE – adaptive selection of groups of predictors. Brzyski, S., and Bogdan '15, arXiv paper

Backup Slides

Compressed sensing with sorted ℓ_1 norm

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} \qquad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n) \qquad \|\boldsymbol{\beta}\|_0 = s \ll p$$

Sorted ℓ_1 compressed sensing

$$\begin{array}{ll} \min & \lambda_1 |b|_{(1)} + \cdots + \lambda_p |b|_{(p)} \\ \text{s.t.} & \boldsymbol{X}\boldsymbol{b} = \boldsymbol{y} \end{array}$$

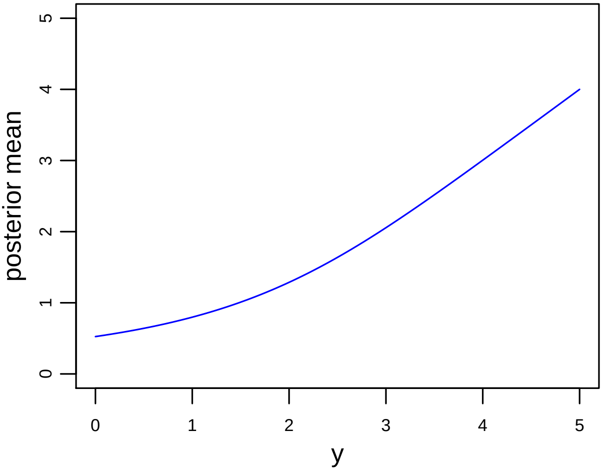
- If $\lambda_1 = \cdots = \lambda_p$, $2s\log(p/s)$ measurements are necessary and sufficient
- If $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathsf{BH}}$, $2s\log(p/s)$ measurements are necessary and sufficient

- Tool: statistical dimension [Amelunxen, Lotz, McCoy, and Tropp '14]
- Question: other optimal weights?

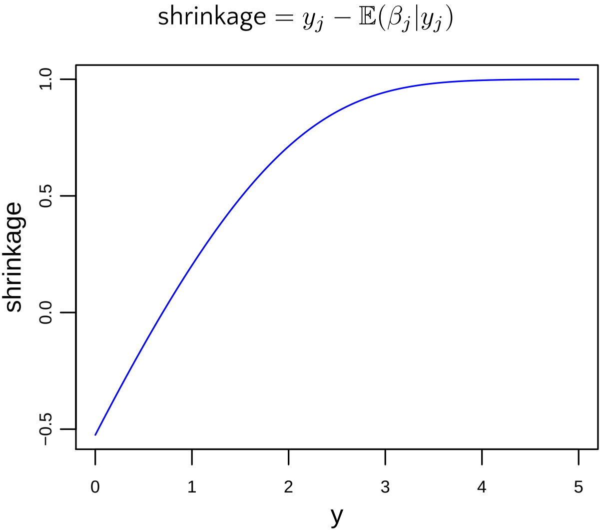
Posterior mean for exponential prior

$$\boldsymbol{y} = \boldsymbol{\beta} + \boldsymbol{z}, \quad \beta_j \overset{\text{iid}}{\sim} \text{Exp}(1)$$

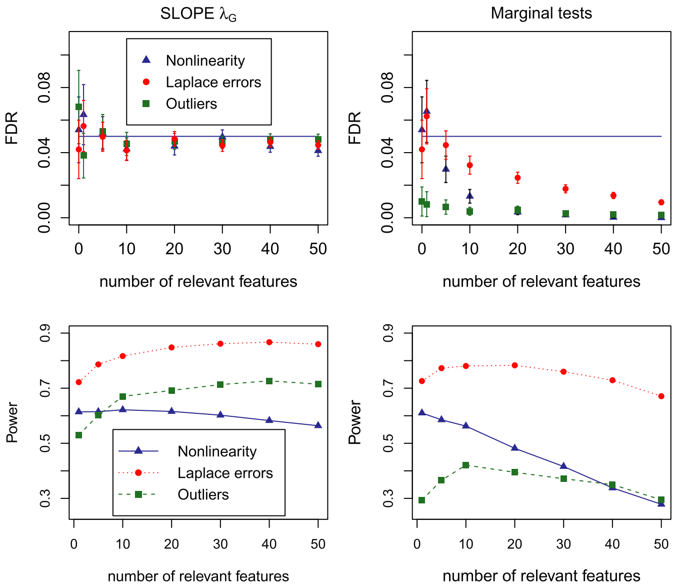
- MAP: $\operatorname{argmin} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \|\boldsymbol{b}\|_1$ subject to $\boldsymbol{b}_j \geq 0$
- Posterior mean: $\mathbb{E}(\beta_j | y_j)$



Posterior mean for exponential prior



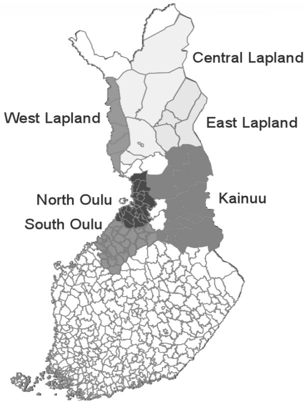
Model misspecification



Northern Finland Birth Cohort

Goal

- Identify variants that impact fasting blood high-density lipoprotein (HDL) levels



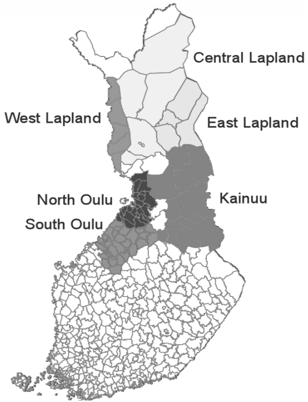
Northern Finland Birth Cohort

Goal

- Identify variants that impact fasting blood high-density lipoprotein (HDL) levels

Dataset

- 6121 individuals from northern Finland
- 1878 genetic variants in regions having documented association with lipid levels
- After perform filtering, dataset dimensions reduced to $(n, p) = (5375, 777)$



Northern Finland Birth Cohort

Goal

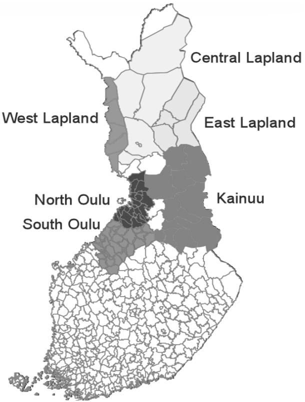
- Identify variants that impact fasting blood high-density lipoprotein (HDL) levels

Dataset

- 6121 individuals from northern Finland
- 1878 genetic variants in regions having documented association with lipid levels
- After perform filtering, dataset dimensions reduced to $(n, p) = (5375, 777)$

Reference

- Service et al reported 14 variants as having effect on HDL ["Re-sequencing expands our understanding of the phenotypic impact of variants at GWAS loci", Service et al '14]



Results

	SLOPE	FullBH	UnivBH	BIC	LassoBonf	LassoCV
	13	7	11	13	10	14
rs2303790	+	+	+	+	−	+
rs5883	+	+	−	+	+	+
rs794676	+	−	+	+	+	+
rs2575875	+	−	+	+	+	+
rs2066715	+	−	+	+	+	+
rs611229	+	−	+	+	+	+
rs12314392	+	−	−	+	−	+
rs11988	+	−	+	−	−	+
rs509360	−	−	−	+	−	+
	14	11	15	17	12	108
v_c9_107555091	+	−	+	+	+	+
rs5801	+	−	+	+	−	+
rs62136410	−	−	+	+	−	+
v_c16_57095439	+	−	−	−	−	+
rs149470424	−	−	−	+	−	+
v_c2_44223117	−	−	−	+	−	+
v_c1_109817524	−	−	−	+	−	+
rs5802	−	−	+	−	−	−
Total	27	18	26	30	22	122

Figure: 17 variants where there is disagreement between all methods, after eliminating 90 variants selected only by 10-fold cross-validation Lasso. We use $q = 0.05$

FDR control

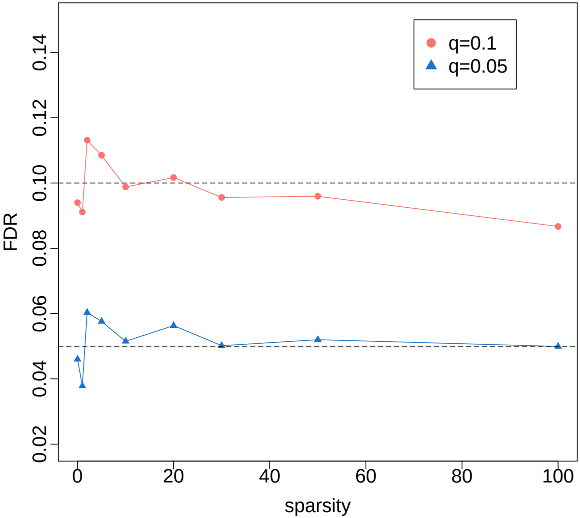
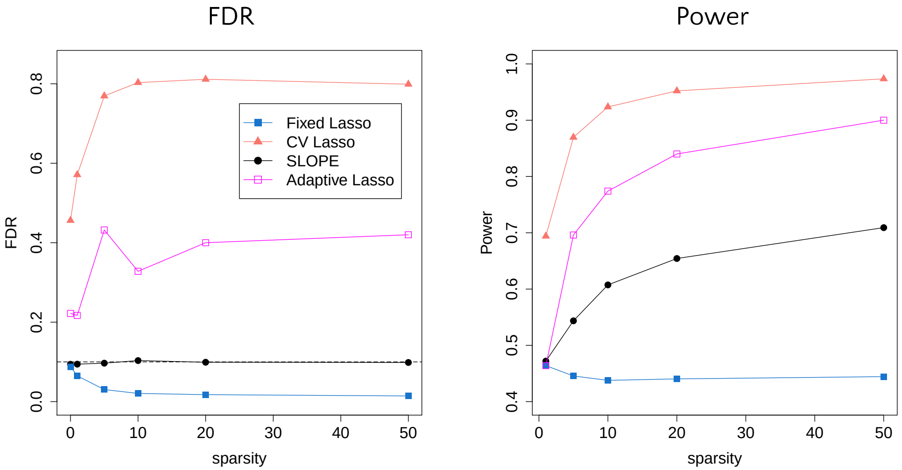


Figure: Simulate nonzero $\beta_j = \sqrt{2 \log p} = 3.65$ and $\sigma^2 = 1$. Over 500 replicates

Adaptive Lasso [Zou '06]



- $q = 0.1, n = p = 5000$. Nonzero $\beta_j = \sqrt{2 \log p} \approx 4.13$ and $\sigma^2 = 1$

Comparison with Knockoffs

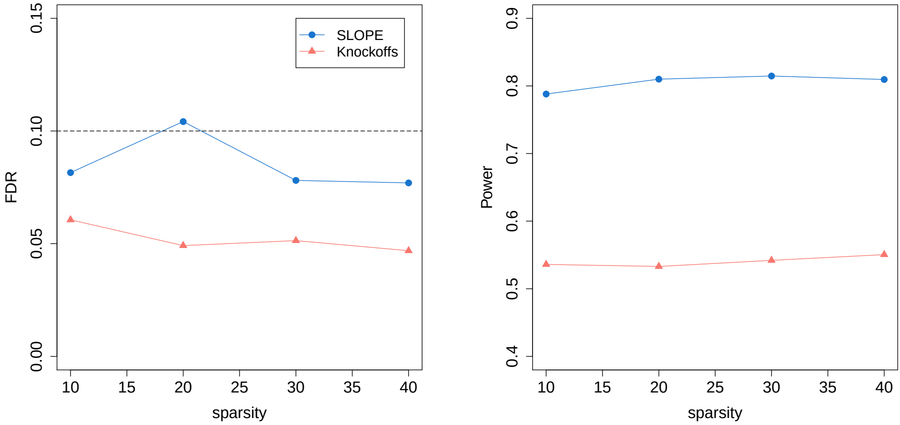


Figure: $q = 0.1, p = 500, n = 1000$. Nonzero $\beta_j = 1.2\sqrt{2 \log p} \approx 4.23$ and $\sigma^2 = 1$. Over 50 replicates

False discoveries occur (very) early on Lasso path

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{z}, \quad X_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1/n), \quad \|\boldsymbol{\beta}\|_0 = s, \quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_n)$$

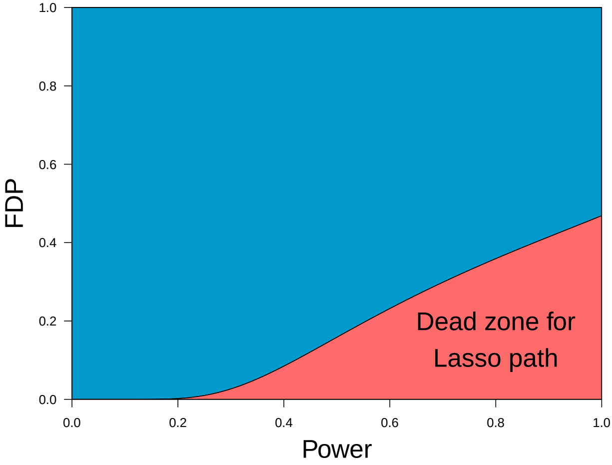


Figure: $n/p = 0.5, \quad s/p = 0.15$

[S., Bogdan, and Candès '15]

Shrinking on the edge

Why are BH cutoffs $\boldsymbol{\lambda}^{\mathrm{BH}}$ optimal also for estimation?

Shrinking on the edge

Why are BH cutoffs $\boldsymbol{\lambda}^{\mathsf{BH}}$ optimal also for estimation?

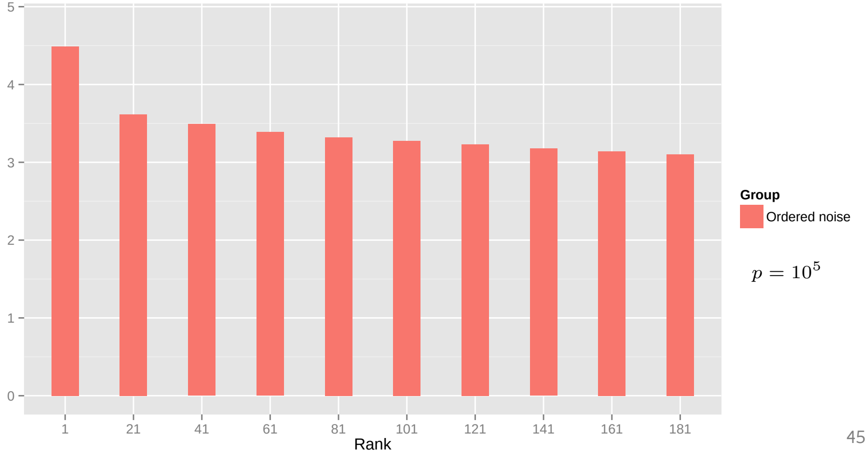
1. Dominate noise in the global null $\boldsymbol{\beta} = \mathbf{0}$;   *2. Introduce less bias*

Shrinking on the edge

Why are BH cutoffs $\boldsymbol{\lambda}^{\mathrm{BH}}$ optimal also for estimation?

1. Dominate noise in the global null $\boldsymbol{\beta} = \mathbf{0}$;   2. Introduce less bias

- Ordered $|\boldsymbol{z}|$ are rank-dependent, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$

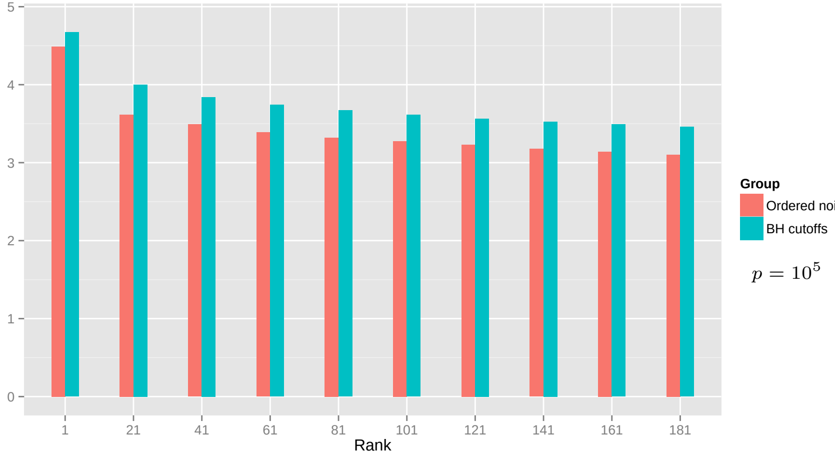


Shrinking on the edge

Why are BH cutoffs $\boldsymbol{\lambda}^{\mathrm{BH}}$ optimal also for estimation?

1. Dominate noise in the global null $\boldsymbol{\beta} = \mathbf{0}$; *2. Introduce less bias*

- Ordered $|\boldsymbol{z}|$ are rank-dependent, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$
- BH cutoffs are rank-dependent and just a bit larger



Proximal gradient algorithm for SLOPE

$$\min_{\boldsymbol{b}} \underbrace{\tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth}}$$

Algorithm 1: Proximal gradient descent

Require: $\boldsymbol{b}^0 \in \mathbb{R}^p$, step sizes t_k

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \boldsymbol{b}^{k+1} = \mathrm{prox}_{t_k \boldsymbol{\lambda}}(\boldsymbol{b}^k - t_k \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{b}^k - \boldsymbol{y}))$

3: **end for**

Proximal gradient algorithm for SLOPE

$$\min_{\boldsymbol{b}} \underbrace{\tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth}}$$

Algorithm 1: Proximal gradient descent

Require: $\boldsymbol{b}^0 \in \mathbb{R}^p$, step sizes t_k

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \boldsymbol{b}^{k+1} = \mathrm{prox}_{t_k \boldsymbol{\lambda}}(\boldsymbol{b}^k - t_k \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{b}^k - \boldsymbol{y}))$

3: **end for**

- $O(1/k)$ convergence rate

Accelerated proximal gradient algorithm for SLOPE

$$\min_{\boldsymbol{b}} \ \underbrace{\tfrac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth}}$$

Algorithm 2: Accelerated proximal gradient descent

Require: $\boldsymbol{b}^0 \in \mathbb{R}^p$, set $\boldsymbol{a}^0 = \boldsymbol{b}^0, \theta_0 = 1$, and step sizes t_k

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \boldsymbol{b}^{k+1} = \operatorname{prox}_{t_k \boldsymbol{\lambda}}(\boldsymbol{a}^k - t_k \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{a}^k - \boldsymbol{y}))$

3: $\quad \theta_{k+1}^{-1} = \frac{1}{2}(1 + \sqrt{1 + 4/\theta_k^2})$

4: $\quad \boldsymbol{a}^{k+1} = \boldsymbol{b}^{k+1} + \underbrace{\theta_{k+1}(\theta_k^{-1} - 1)(\boldsymbol{b}^{k+1} - \boldsymbol{b}^k)}_{\text{momentum}}$

5: **end for**

Accelerated proximal gradient algorithm for SLOPE

$$\min_{\boldsymbol{b}} \underbrace{\tfrac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth}}$$

Algorithm 2: Accelerated proximal gradient descent

Require: $\boldsymbol{b}^0 \in \mathbb{R}^p$, set $\boldsymbol{a}^0 = \boldsymbol{b}^0, \theta_0 = 1$, and step sizes t_k

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \boldsymbol{b}^{k+1} = \mathrm{prox}_{t_k\boldsymbol{\lambda}}(\boldsymbol{a}^k - t_k\boldsymbol{X}'(\boldsymbol{X}\boldsymbol{a}^k - \boldsymbol{y}))$

3: $\quad \theta_{k+1}^{-1} = \frac{1}{2}(1 + \sqrt{1 + 4/\theta_k^2})$

4: $\quad \boldsymbol{a}^{k+1} = \boldsymbol{b}^{k+1} + \underbrace{\theta_{k+1}(\theta_k^{-1} - 1)(\boldsymbol{b}^{k+1} - \boldsymbol{b}^k)}_{\text{momentum}}$

5: **end for**

- $O(1/k^2)$ convergence rate

Accelerated proximal gradient algorithm for SLOPE

$$\min_{\boldsymbol{b}} \underbrace{\tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}\|_2^2}_{\text{smooth}} + \underbrace{J_{\boldsymbol{\lambda}}(\boldsymbol{b})}_{\text{nonsmooth}}$$

Algorithm 2: Accelerated proximal gradient descent

Require: $\boldsymbol{b}^0 \in \mathbb{R}^p$, set $\boldsymbol{a}^0 = \boldsymbol{b}^0, \theta_0 = 1$, and step sizes t_k

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \boldsymbol{b}^{k+1} = \operatorname{prox}_{t_k \boldsymbol{\lambda}}(\boldsymbol{a}^k - t_k \boldsymbol{X}'(\boldsymbol{X}\boldsymbol{a}^k - \boldsymbol{y}))$

3: $\quad \theta_{k+1}^{-1} = \frac{1}{2}(1 + \sqrt{1 + 4/\theta_k^2})$

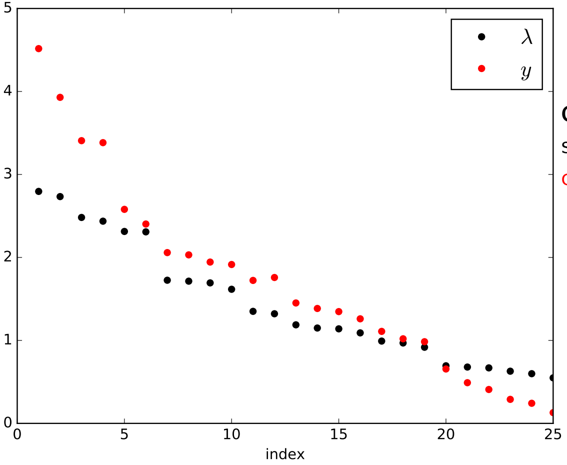
4: $\quad \boldsymbol{a}^{k+1} = \boldsymbol{b}^{k+1} + \underbrace{\theta_{k+1}(\theta_k^{-1} - 1)(\boldsymbol{b}^{k+1} - \boldsymbol{b}^k)}_{\text{momentum}}$

5: **end for**

- $O(1/k^2)$ convergence rate
- Prox operator: $\operatorname{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\operatorname{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + J_{\boldsymbol{\lambda}}(\boldsymbol{b})$

Compute the prox

$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\mathrm{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



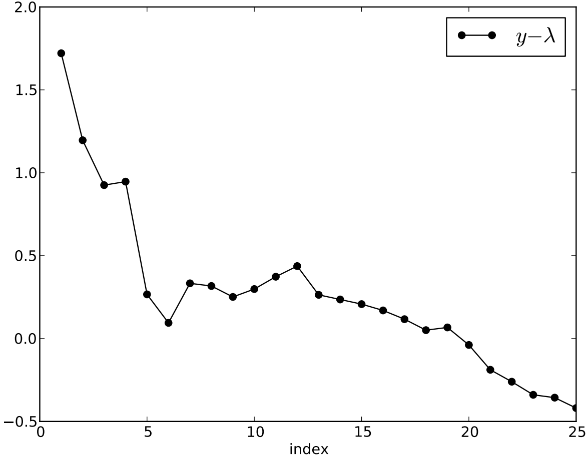
Observation
solution has same signs and ordering with $\boldsymbol{y}$

$$y_1 \geq \cdots \geq y_p \geq 0$$
$$\lambda_1 \geq \cdots \geq \lambda_p$$

Adapted from pool adjacent violators algorithm (PAVA), Kruskal ('64), Barlow, Bartholomew, Bremner, and Brunk ('72)

Compute the prox

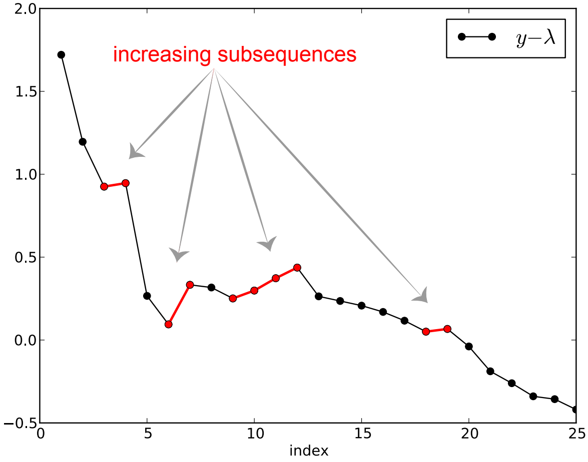
$$\operatorname{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\operatorname{argmin}}\ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Plot $\Delta \boldsymbol{y} \triangleq \boldsymbol{y} - \boldsymbol{\lambda}$

Compute the prox

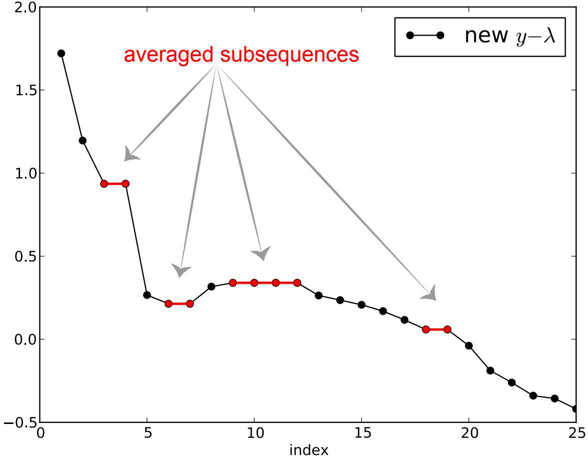
$$\text{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\text{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Detect increasing subsequences

Compute the prox

$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$

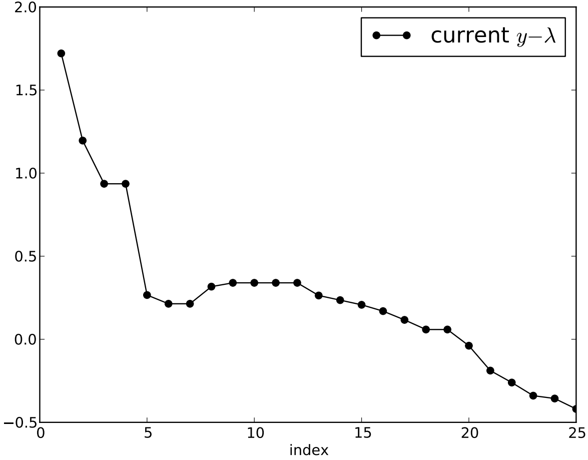


Average over increasing subsequences

$$\Delta \boldsymbol{y} \leftarrow \overline{\Delta \boldsymbol{y}}$$

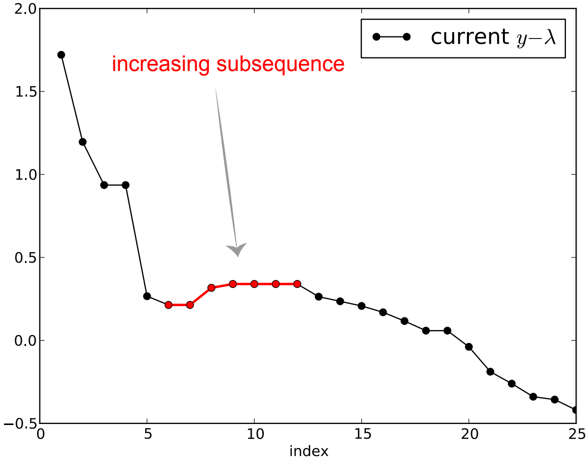
Compute the prox

$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{b}} \; \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Compute the prox

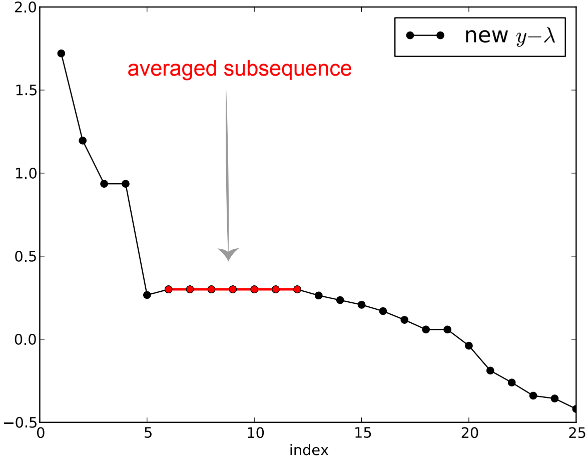
$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\mathrm{argmin}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Detect increasing subsequences

Compute the prox

$$\operatorname{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\operatorname{argmin}}\ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$

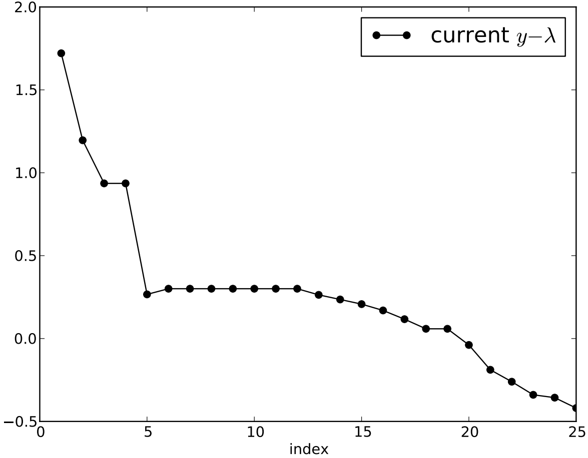


Average over increasing subsequences

$$\Delta \boldsymbol{y} \leftarrow \overline{\Delta \boldsymbol{y}}$$

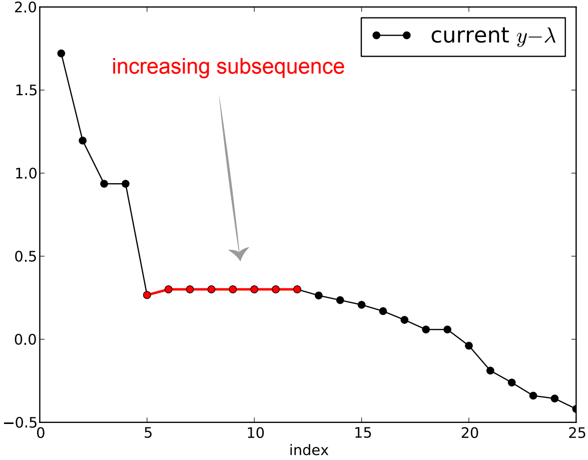
Compute the prox

$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \operatorname*{argmin}_{\boldsymbol{b}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Compute the prox

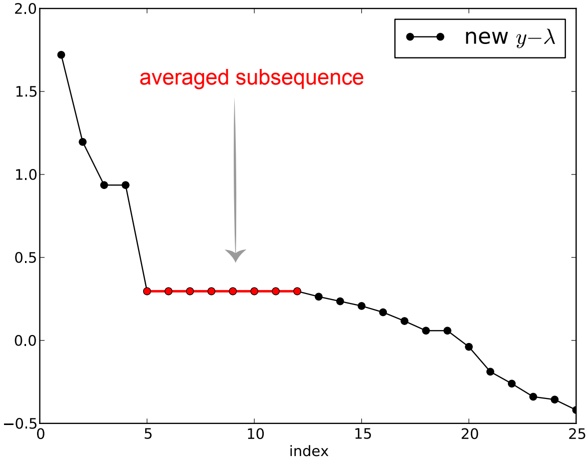
$$\operatorname{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\operatorname{argmin}}\ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Detect increasing subsequences

Compute the prox

$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$

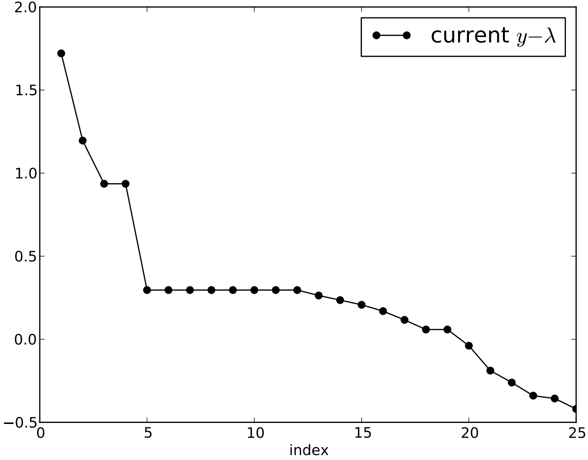


Average over increasing subsequences

$$\Delta \boldsymbol{y} \leftarrow \overline{\Delta \boldsymbol{y}}$$

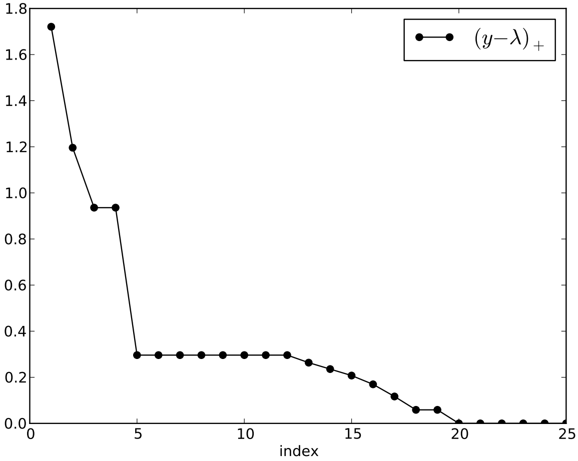
Compute the prox

$$\mathrm{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\mathrm{argmin}}\ \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



Compute the prox

$$\operatorname{prox}_{\boldsymbol{\lambda}}(\boldsymbol{y}) = \underset{\boldsymbol{b}}{\operatorname{argmin}}\ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{b}\|_2^2 + \sum_{j=1}^{p} \lambda_j |b|_{(j)}$$



- Take positive part ($x_+ = \max\{x, 0\}$)
- Restore signs and ordering
- Done!

Comparison with Lasso under Gaussian design

- Lasso with $\lambda = (1 + o(1))\sqrt{2 \log p}$ has worst case risk $2s \log p$
- SLOPE with $\boldsymbol{\lambda} = (1 + o(1))\boldsymbol{\lambda}^{\mathsf{BH}}$ has worst case risk $2s \log(p/s)$

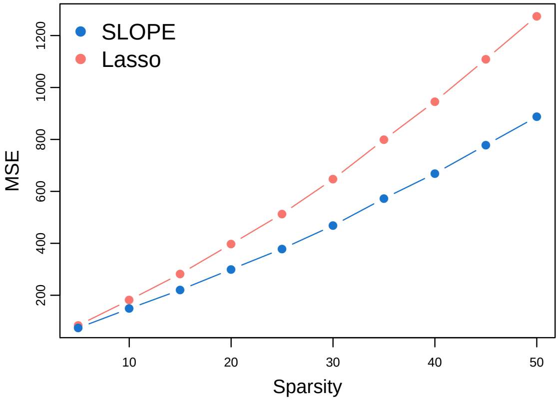


Figure: $n = 500, p = 1000$, nonzero $\beta_j = 10\lambda_1^{\mathsf{BH}}, \sigma = 1, q = 0.05$

Related methods (some are very new)

- Lasso
- Selective inference on Lasso path (G'Sell et al, '15)
- Knockoffs (Barber and Candès, '15)

Accelerated gradient descent

f convex and ∇f Lipschitz

$$\min \quad f(\boldsymbol{x})$$

Gradient descent: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s\nabla f(\boldsymbol{x}_k)$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k)$

Nesterov's accelerated gradient descent ('83)

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$
- Generalize to $f(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_1$, $f(\boldsymbol{x}) + J_{\boldsymbol{\lambda}}(\boldsymbol{x})$, etc

An ordinary differential equation

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \mathbf{0}, \quad \boldsymbol{X}(0) = \boldsymbol{x}_0, \dot{\boldsymbol{X}}(0) = \mathbf{0}$$

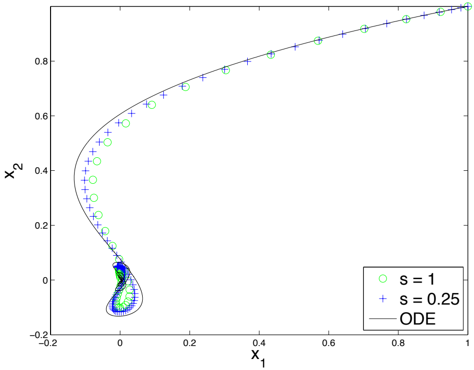


Figure: $f(\boldsymbol{x}) = 4x_1^2 + x_2^2$ starting from $\boldsymbol{x}_0 = (1, 1)$

[S., Boyd, Candès ’15]

Comparison and generalization

$$
\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1}) \\
\boldsymbol{y}_k &= \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})
\end{aligned}
$$

$$
\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \mathbf{0}
$$

Comparison and generalization

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \mathbf{0}$$

- $f(\boldsymbol{X}(t)) - f^\star \le O(1/t^2)$

Comparison and generalization

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$
- sophisticated proof

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \boldsymbol{0}$$

- $f(\boldsymbol{X}(t)) - f^\star \le O(1/t^2)$
- Simple Lyapunov function

Comparison and generalization

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$
- sophisticated proof
- $k+2-(k-1) = 3$

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \boldsymbol{0}$$

- $f(\boldsymbol{X}(t)) - f^\star \le O(1/t^2)$
- Simple Lyapunov function
- $3/t$

Comparison and generalization

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$
- sophisticated proof
- $k + 2 - (k-1) = 3$

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \mathbf{0}$$

- $f(\boldsymbol{X}(t)) - f^\star \le O(1/t^2)$
- Simple Lyapunov function
- $3/t$

Theorem (S., Boyd, and Candès)

For any $r > 0$, replace $(k-1)/(k+2)$ with $(k-1)/(k+r-1)$, and $3/t$ with r/t

Comparison and generalization

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$
- sophisticated proof
- $k + 2 - (k-1) = 3$

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \mathbf{0}$$

- $f(\boldsymbol{X}(t)) - f^\star \le O(1/t^2)$
- Simple Lyapunov function
- $3/t$

Theorem (S., Boyd, and Candès)

For any $r > 0$, replace $(k-1)/(k+2)$ with $(k-1)/(k+r-1)$, and $3/t$ with r/t

- *If $r \ge 3$, quadratic convergence holds for both*

Comparison and generalization

$$\boldsymbol{x}_k = \boldsymbol{y}_{k-1} - s\nabla f(\boldsymbol{y}_{k-1})$$
$$\boldsymbol{y}_k = \boldsymbol{x}_k + \frac{k-1}{k+2}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$$

- $f(\boldsymbol{x}_k) - f^\star \le O(1/k^2)$
- sophisticated proof
- $k + 2 - (k-1) = 3$

$$\ddot{\boldsymbol{X}}(t) + \frac{3}{t}\dot{\boldsymbol{X}}(t) + \nabla f(\boldsymbol{X}(t)) = \mathbf{0}$$

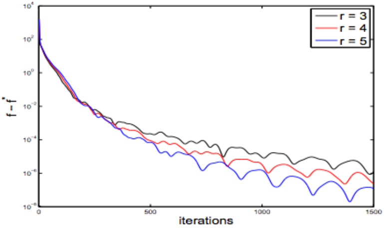
- $f(\boldsymbol{X}(t)) - f^\star \le O(1/t^2)$
- Simple Lyapunov function
- $3/t$

Theorem (S., Boyd, and Candès)

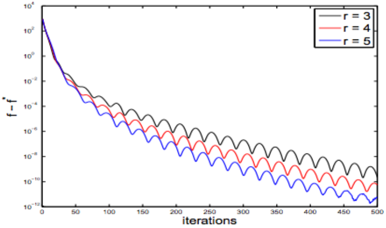
For any $r > 0$, replace $(k-1)/(k+2)$ with $(k-1)/(k+r-1)$, and $3/t$ with r/t

- *If $r \ge 3$, quadratic convergence holds for both*
- *If $r < 3$, counterexamples*

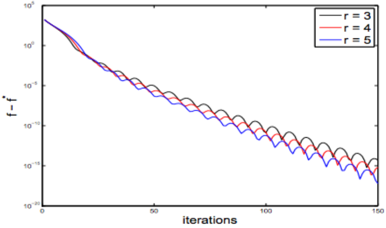
Numerical examples



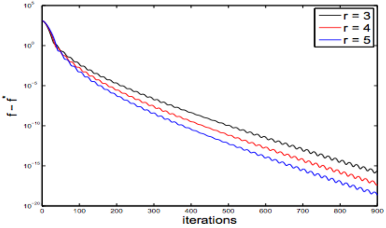
(a) Lasso with fat design.



(b) Lasso with square design.



(c) NLS with fat design.



(d) NLS with square design.

Choice of λ

- Variance inflation caused by shrinkage
-

$$\lambda_1 = \sigma \lambda_1^{\mathsf{BH}}$$
$$\lambda_j = \sigma \lambda_j^{\mathsf{BH}} \sqrt{1 + \frac{w_j}{\sigma^2} \sum_{i<j} \lambda_i^2}, \ j \geq 2$$

- $w_j = \frac{1}{j} \mathbb{E} \| (\boldsymbol{X}_T' \boldsymbol{X}_T)^{-1} \boldsymbol{X}_T' \boldsymbol{X}_i \|^2$. Monte Carlo simulations over all $T \subset \{1, \ldots, p\}$ with $|T| = k$ and $i \notin S$
- $\lambda_j > \sigma \lambda_j^{\mathsf{BH}}$: more conservative procedure