Adaptive Validation, a possible alternative to Cross-Validation

Joseph Salmon
http://josephsalmon.eu
Télécom Paristech, Institut Mines-Télécom

Joint work with:
Didier Chételat (Cornell University)
Johannes Lederer (Cornell University)

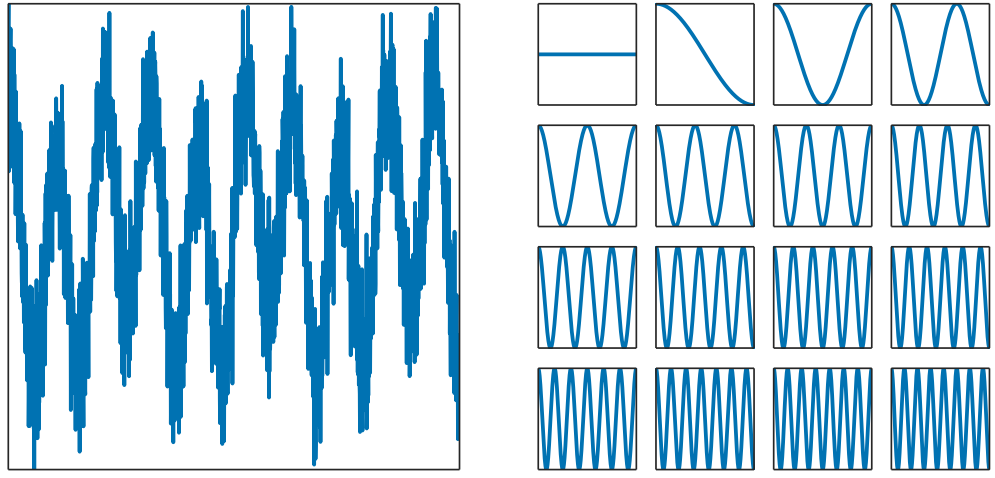
Didier

Johannes

Sparsity of signals is all around

Signals can often be represented through a combination of a **few** elements / **atoms** :

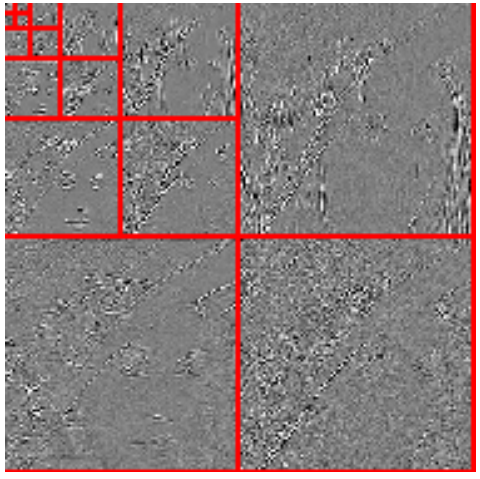
- Fourier decomposition for sounds



Sparsity of signals is all around

Signals can often be represented through a combination of a **few** elements / **atoms** :

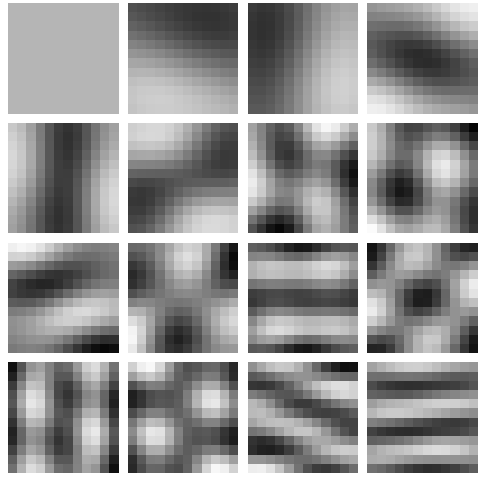
- Fourier decomposition for sounds
- Wavelet for images (1990's)



Sparsity of signals is all around

Signals can often be represented through a combination of a **few** elements / **atoms** :

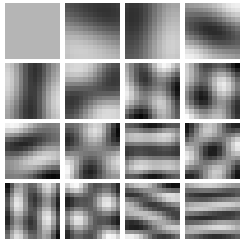
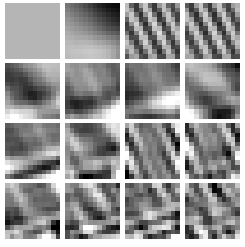
- Fourier decomposition for sounds
- Wavelet for images (1990's)
- Dictionary learning for images (late 2000's)



Sparse linear model

Let $y \in \mathbb{R}^n$ be a signal, *e.g.*, an image


Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$ be a collection of (normalized) atoms: corresponds to a **dictionary**

X well suited if one can approximate the signal $y \approx X\beta$ with a **sparse** vector $\beta \in \mathbb{R}^p$

$$\underbrace{\begin{pmatrix} \\ y \\ \\ \end{pmatrix}}_{y \in \mathbb{R}^n} \approx \underbrace{\left(\begin{array}{c|c|c} & & \\ \mathbf{x}_1 & \cdots & \mathbf{x}_p \\ & & \end{array} \right)}_{X \in \mathbb{R}^{n \times p}} \cdot \underbrace{\begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}}_{\beta \in \mathbb{R}^p}$$

The Lasso and its other names

Possible way to get a sparse vector when the dictionary is known:

$$\hat{\beta}^\lambda \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\underbrace{\frac{1}{2}\|y - X\beta\|_2^2}_{\textbf{data fitting term}} + \underbrace{\lambda \|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)$$

Rem: Convex optimization problem, can be solved efficiently
Rem: Did I mention you have to tune/choose λ?

Vocabulary:

- Statistics: **Lasso**
  Tibshirani (1996)
- Signal processing: **Basis Pursuit**
  Chen, Donoho and Saunders (1998)

Dictionary learning: last motivation word

One typically observes T training signals $y_1, \dots y_T$.
start with an initial dictionary X_0, then alternates over $t \in [T]$:

$$\begin{cases} \hat{\beta}_t^\lambda \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \dfrac{1}{2} \| y_t - X_{t-1}\beta \|_2^2 + \lambda \|\beta\|_1 & \text{(coefficients update)} \\ X_t \in \underset{X \in \mathcal{N}}{\arg\min} \dfrac{1}{2} \| y_{t-1} - X\hat{\beta}_{t-1}^\lambda \|_2^2 & \text{(dictionary update)} \end{cases}$$

where $\mathcal{N}$ is a set of **normalized dictionary** in $\mathbb{R}^{n \times p}$,

Rem: Applied to signals being small patches of images this reaches state-of-the-art on several image processing tasks *cf.* Mairal *et al.* (2010)

Model considered afterwards

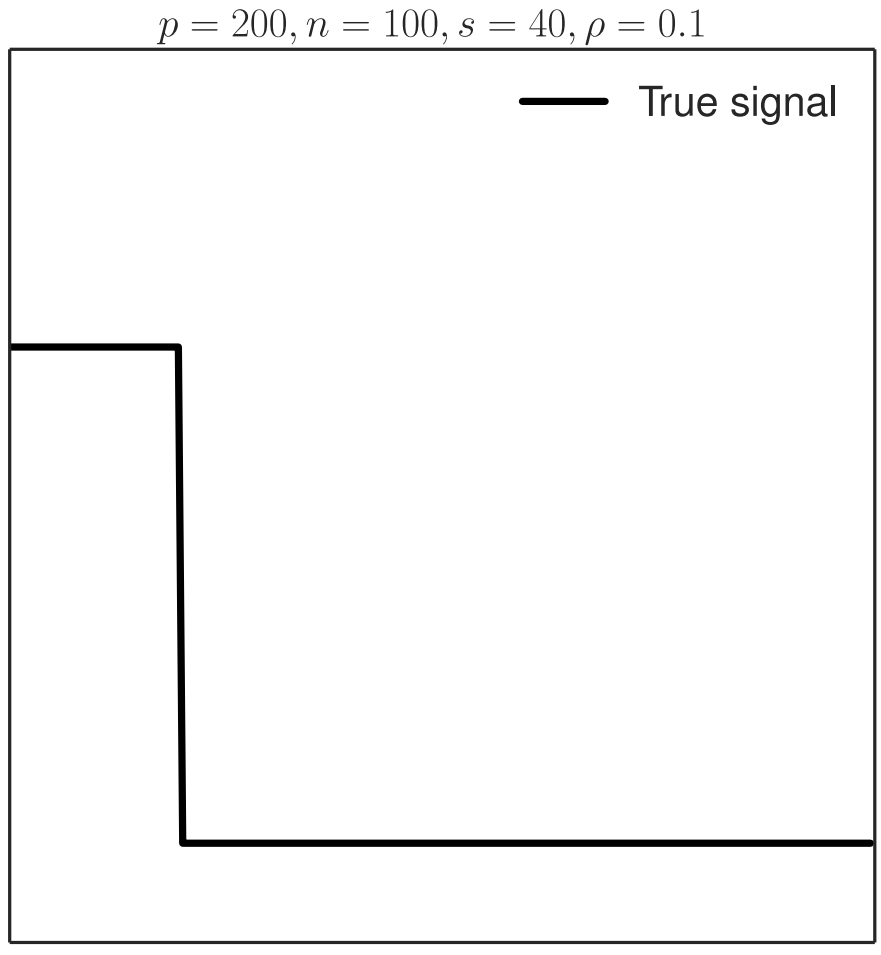
We consider a simple model for theory and simulation validation

$$\boxed{y = X\beta^* + \varepsilon}$$

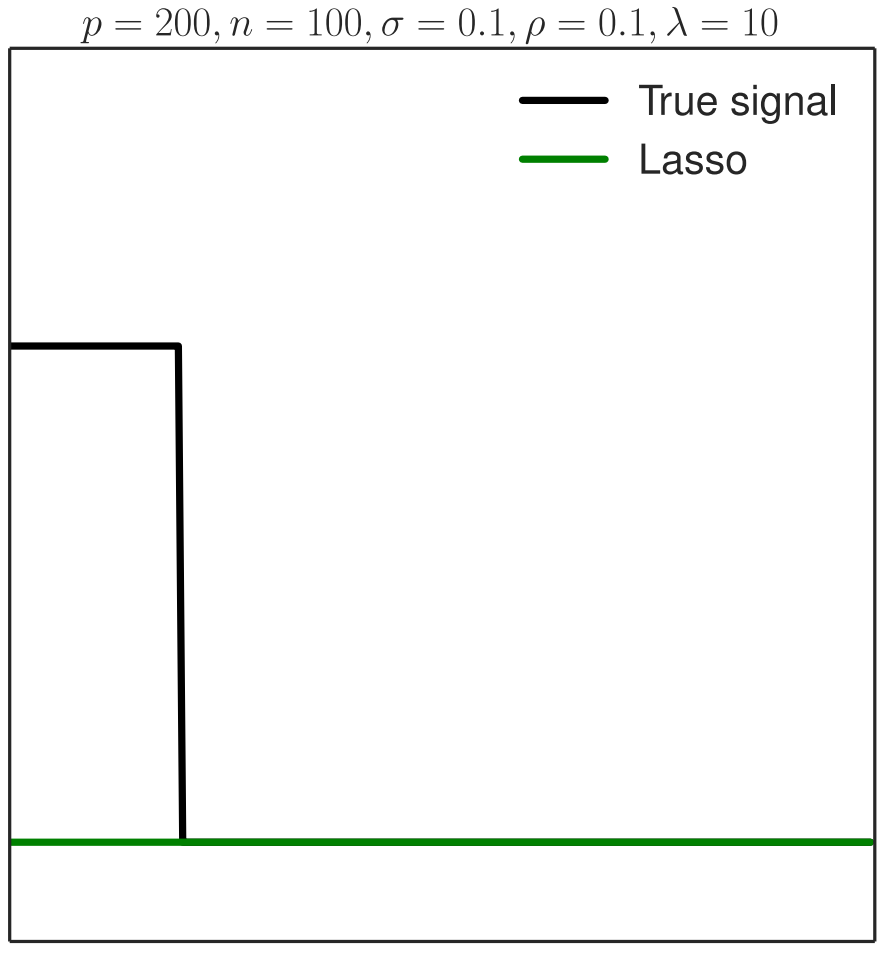
- additive white **Gaussian noise** $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_n)$
- $X \in \mathbb{R}^{n \times p}$ has **normalized** columns *e.g.*, $\|\mathbf{x}_j\|_2 = \sqrt{n}$ for all $j = 1, \ldots, p$
- the true signal β^* is **sparse**
- the **prediction error** $\mathbb{E}\|X\beta^* - X\hat{\beta}\|_2^2/n$ as a measure of performance for an estimator $\hat{\beta}$

Rem: Note that p can be larger than n

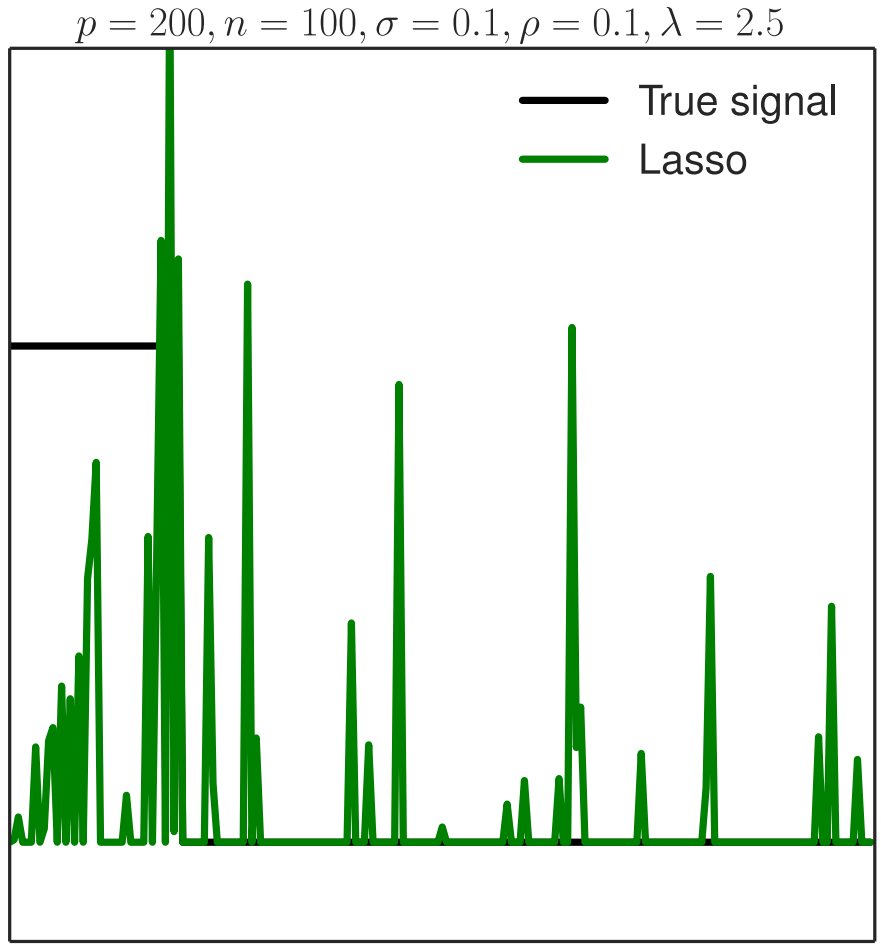
Least square refitting: the Lasso is biased



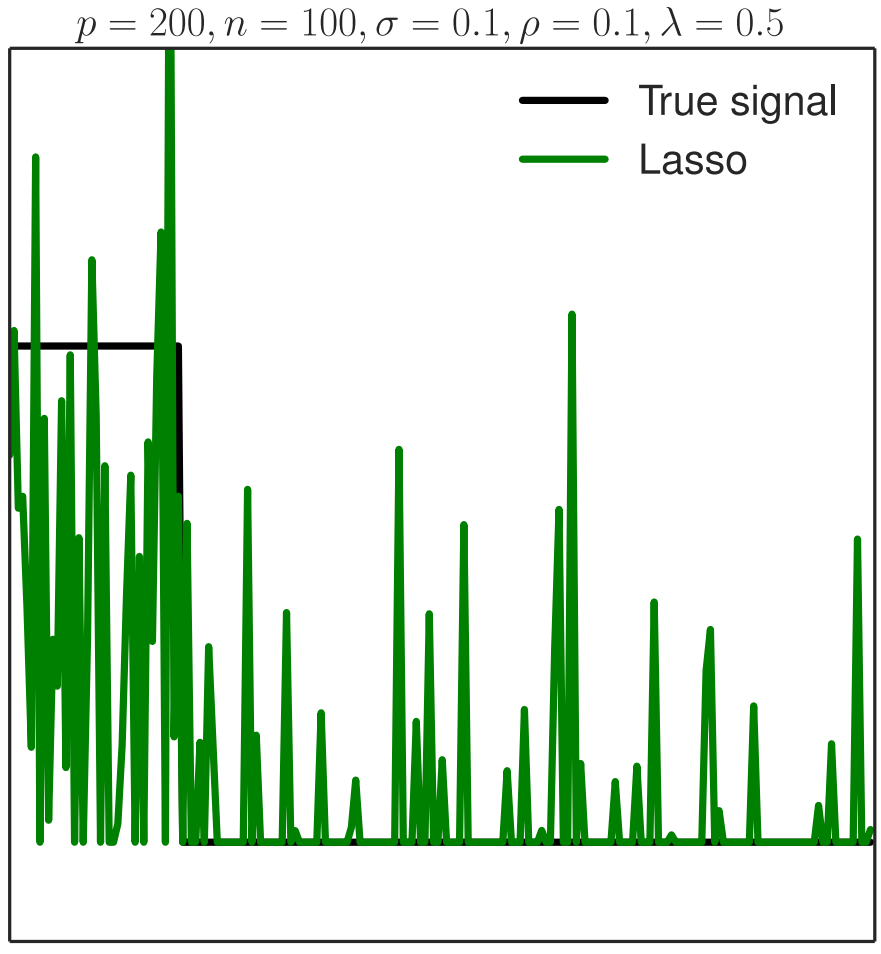
Least square refitting: the Lasso is biased



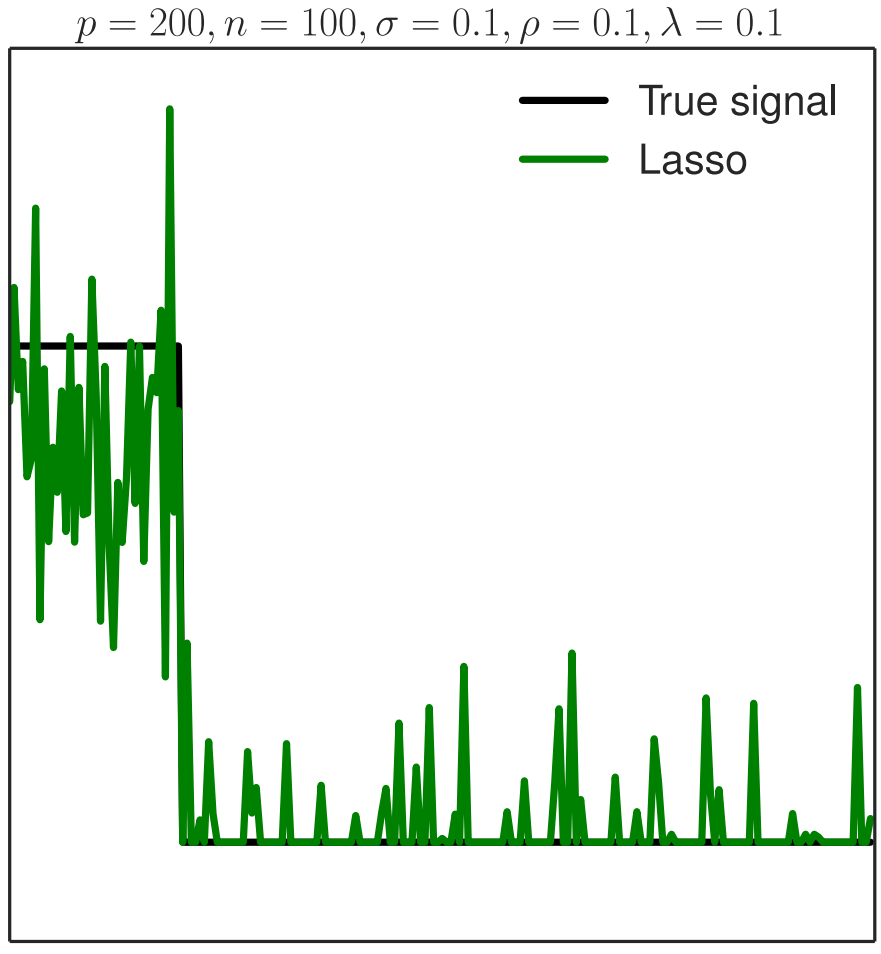
Least square refitting: the Lasso is biased



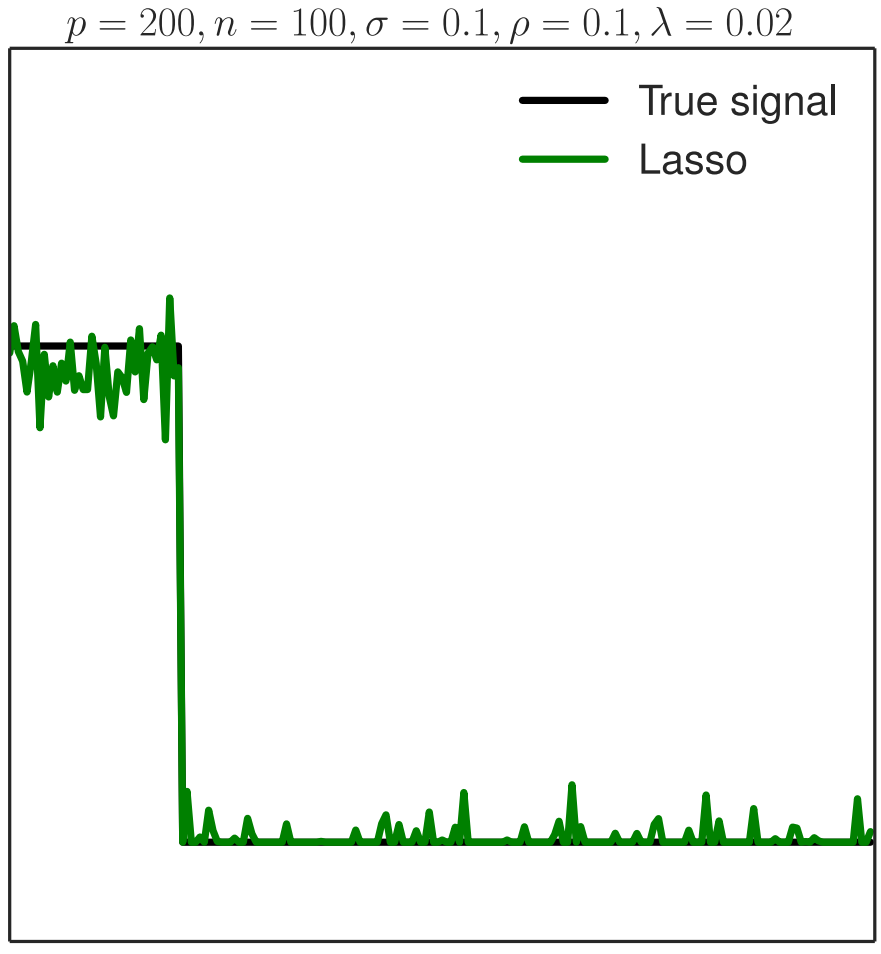
Least square refitting: the Lasso is biased



Least square refitting: the Lasso is biased



Least square refitting: the Lasso is biased



Orthonormal dictionary ($n = p$)

Orthonormal case: $X^\top X = \mathrm{Id}_p$ or $\langle \mathbf{x}_i | \mathbf{x}_j \rangle = \delta_{i,j}$ for all i, j

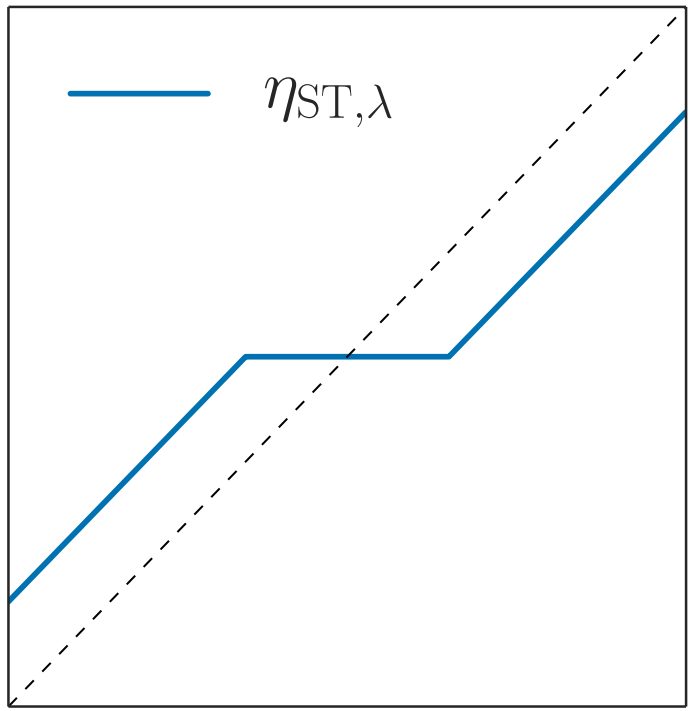
$$\hat{\beta}^\lambda = \arg\min_{\beta \in \mathbb{R}^p} \left(\frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right)$$

Solution = **Soft-Thresholding**:

$$\hat{\beta}^\lambda = \begin{pmatrix} \eta_{\mathrm{ST},\lambda}(\langle \mathbf{x}_1, y \rangle) \\ \vdots \\ \eta_{\mathrm{ST},\lambda}(\langle \mathbf{x}_p, y \rangle) \end{pmatrix}$$

where

$$\eta_{\mathrm{ST},\lambda}(x) = \mathrm{sign}(x) \cdot (|x| - \lambda)_+$$



Drawback: it shrinks large coefficients toward zero by a factor λ!

Debiasing the Lasso the easy way

$$\hat{\beta}^{\lambda}_{\text{Lasso}} \in \arg\min_{\beta \in \mathbb{R}^p} \left(\frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right)$$

Definition: active set

The **active set** or **support** of a vector β is the indexes of its non-zero coordinates

$$\text{supp}[\beta] = \{j \in [p] : \beta_j \neq 0\}$$

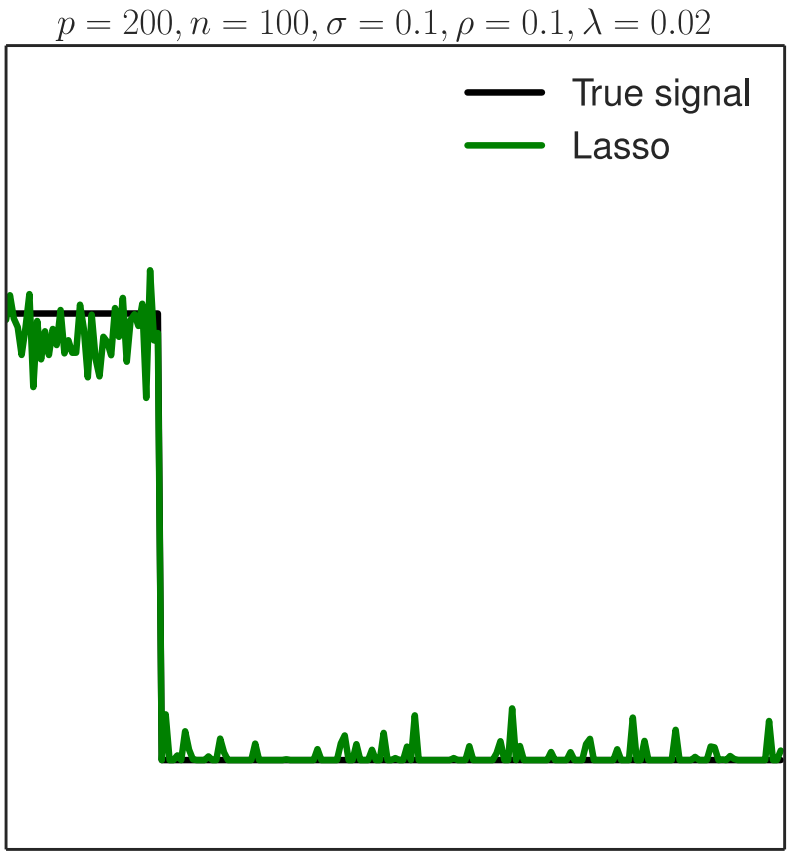
Rem: The Lasso is sparse means its active set $\text{supp}[\hat{\beta}^{\lambda}_{\text{Lasso}}]$ is small

Definition: LSLasso

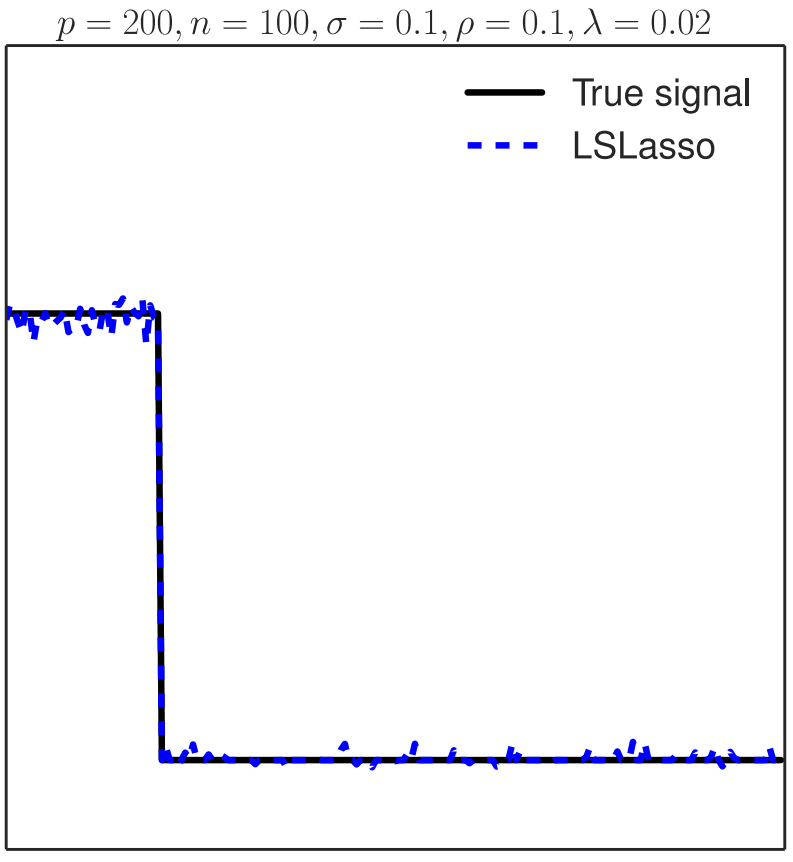
The Least Square Lasso (**LSLasso**): performs a least square fitting for the variables $\mathbf{x}_j$ that are activated

$$\hat{\beta}^{\lambda}_{\text{LSLasso}} \in \underset{\text{supp}[\beta] = \text{supp}[\hat{\beta}^{\lambda}_{\text{Lasso}}]}{\arg\min} \|Y - X\beta\|_2^2$$

LSLasso in practice

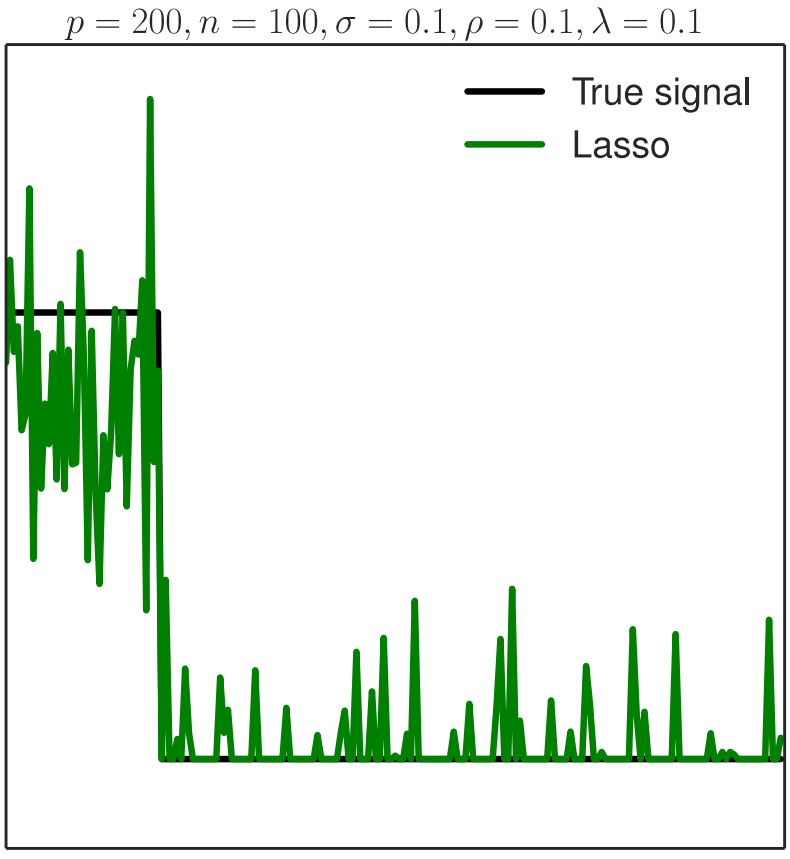


Lasso

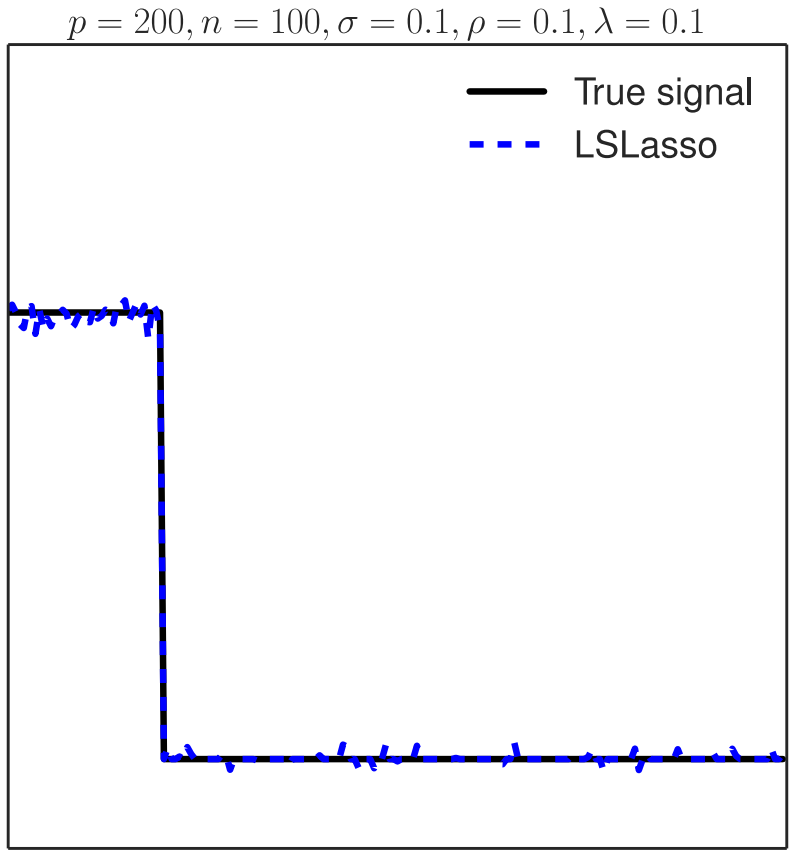


LSLasso

LSLasso in practice

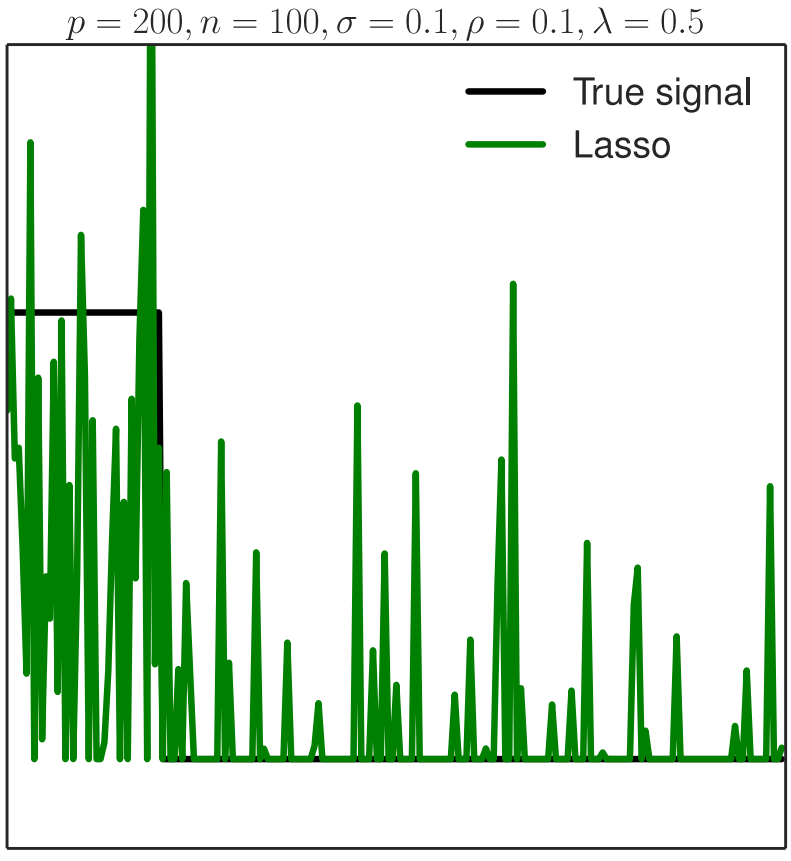


Lasso

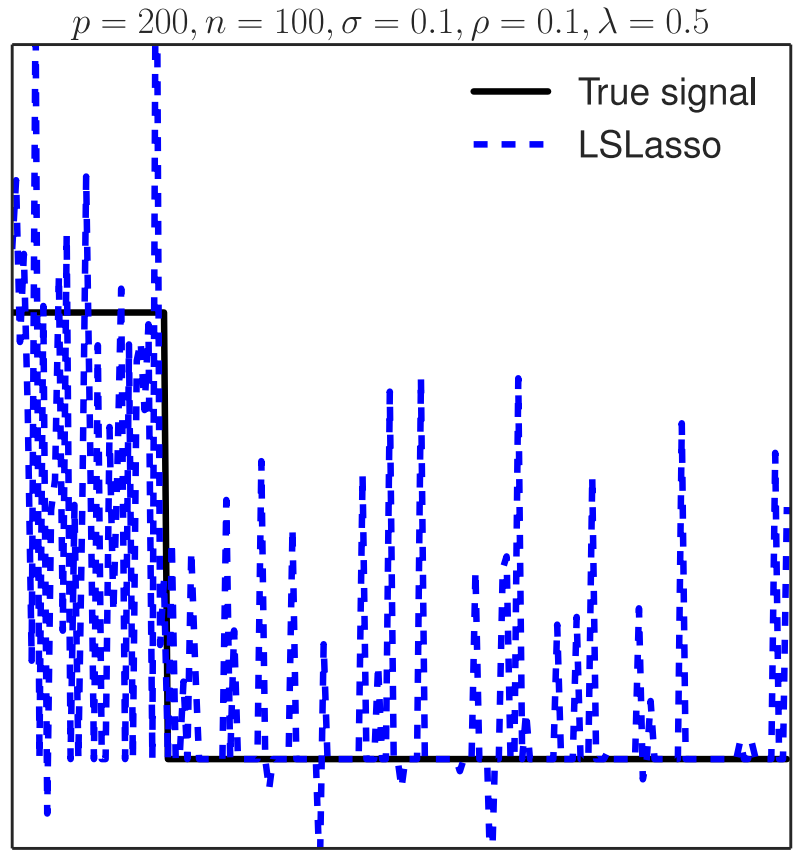


LSLasso

LSLasso in practice

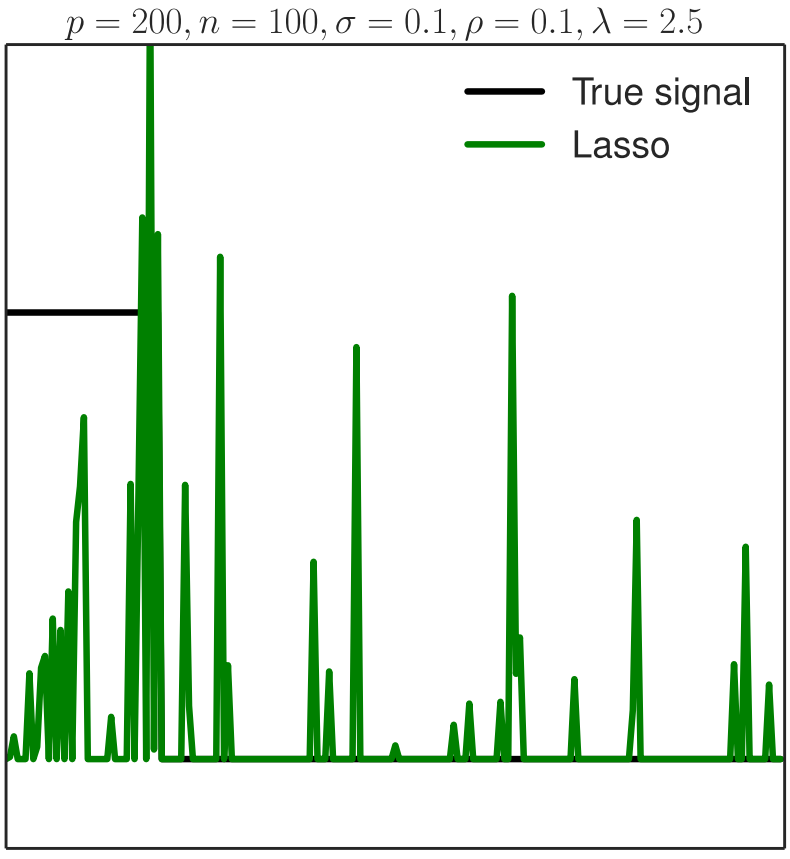


Lasso

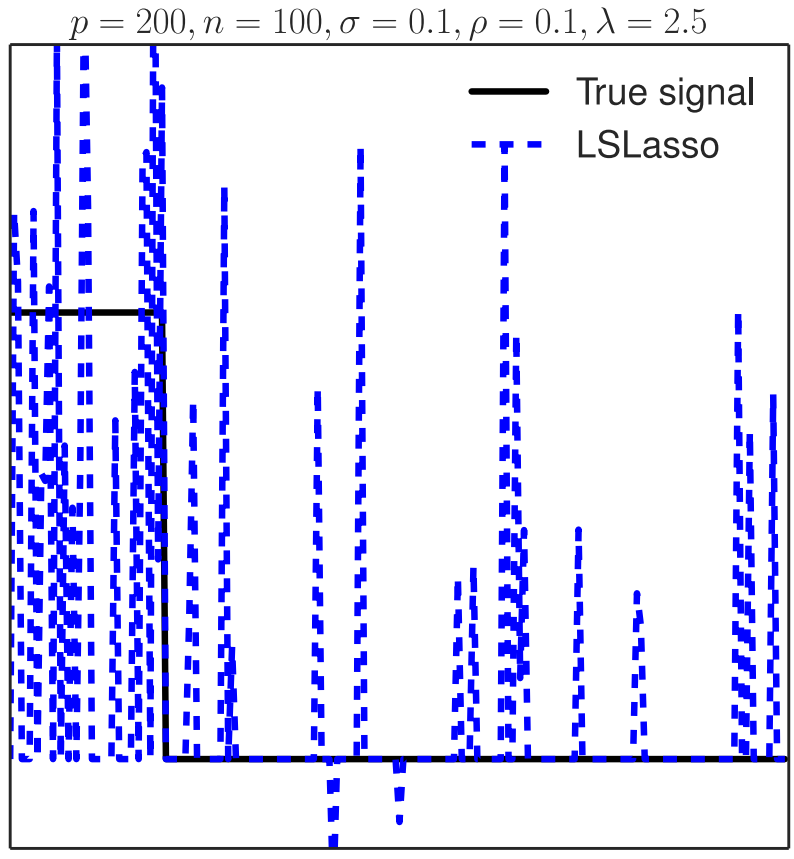


LSLasso

LSLasso in practice



Lasso

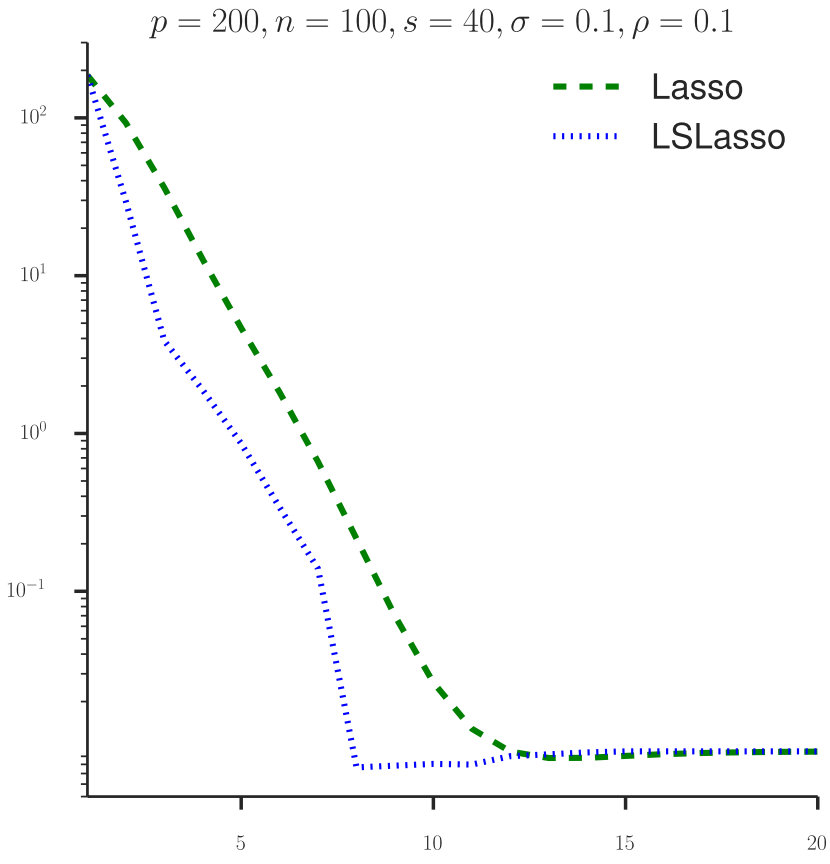


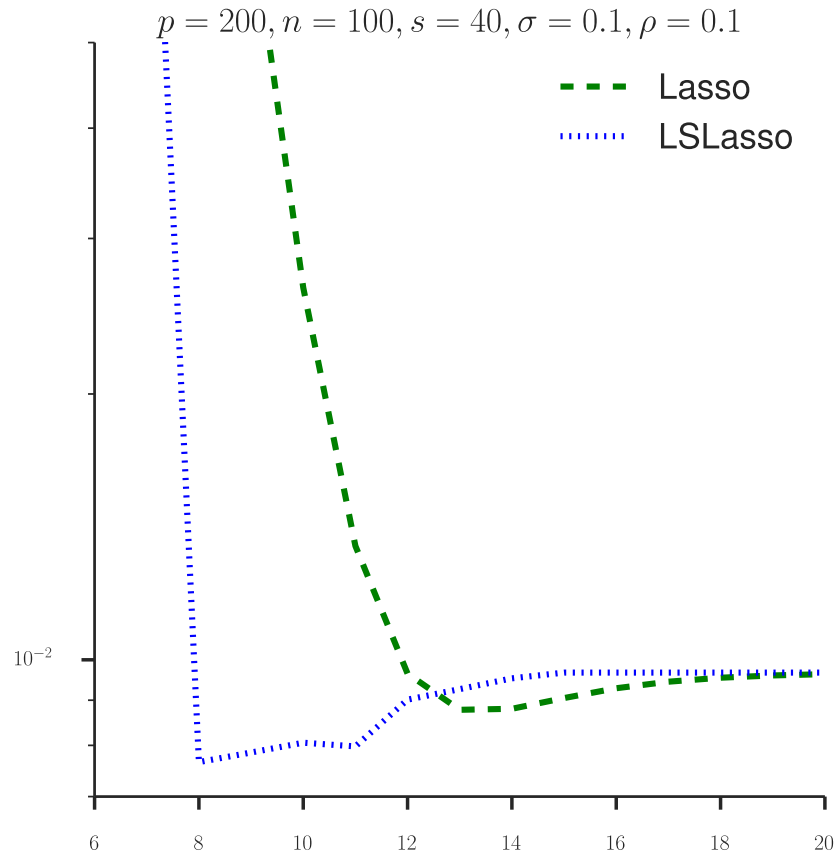
LSLasso

Tuning parameters

In practice, one computes r Lasso for $\lambda_1 > \cdots > \lambda_r$ and searches:

- the best parameter for the LSLasso $\hat{\beta}^{\lambda^*}_{\text{LSLasso}}$
- the best parameter for the Lasso $\hat{\beta}^{\lambda^*}_{\text{Lasso}}$



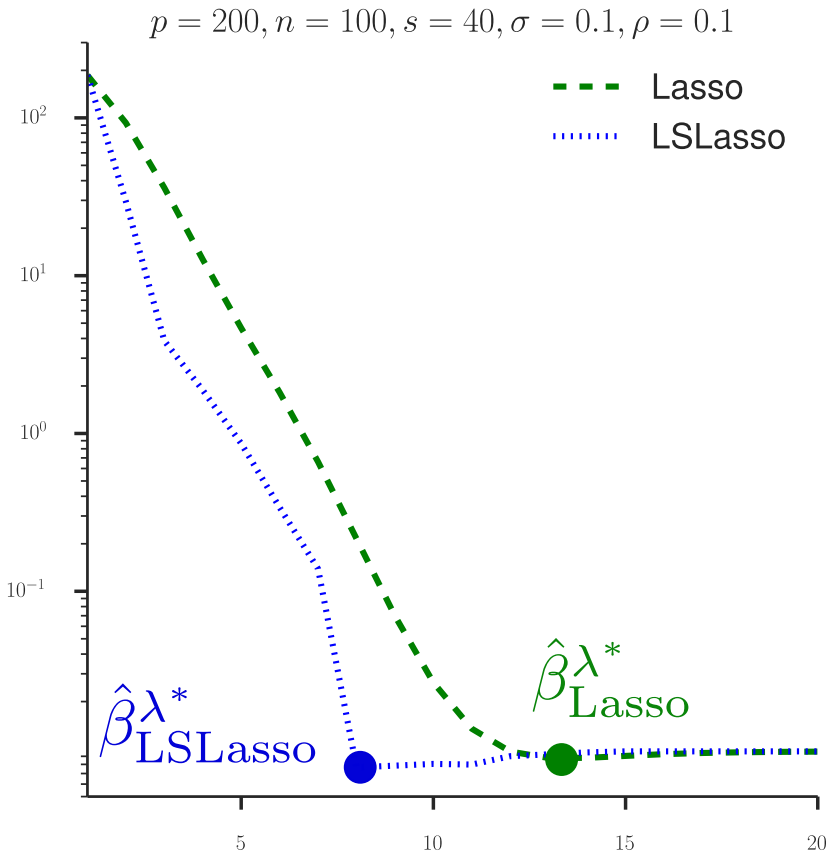


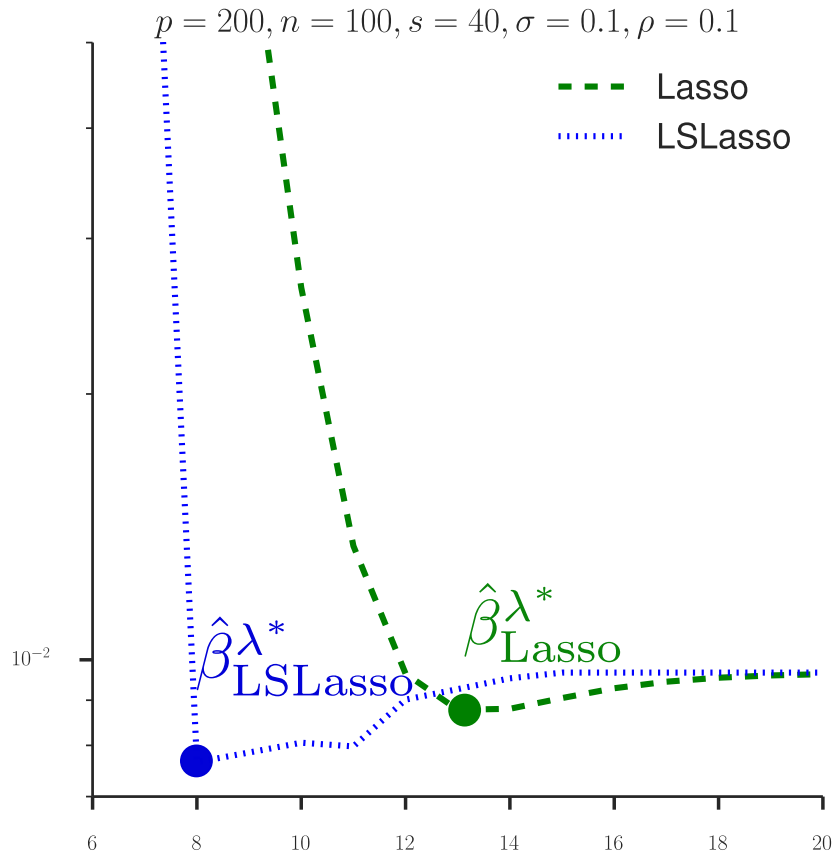
Prediction error $\|X\beta^* - X\hat{\beta}\|_2^2/n$ (here $r = 20$)

Tuning parameters

In practice, one computes r Lasso for $\lambda_1 > \cdots > \lambda_r$ and searches:

- the best parameter for the LSLasso $\hat{\beta}^{\lambda^*}_{\text{LSLasso}}$
- the best parameter for the Lasso $\hat{\beta}^{\lambda^*}_{\text{Lasso}}$

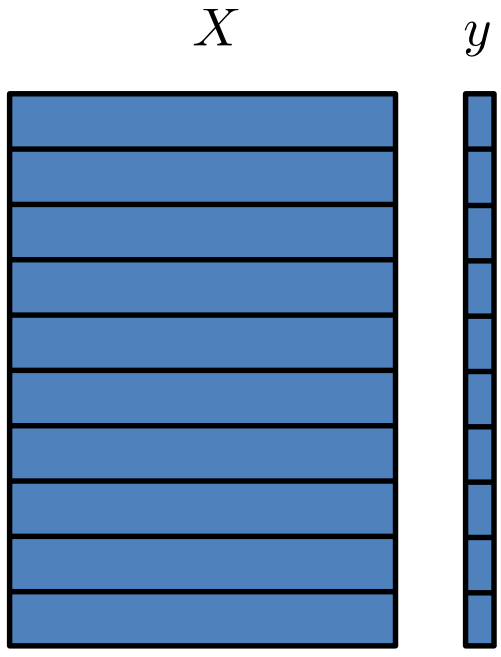




Prediction error $\|X\beta^* - X\hat{\beta}\|_2^2/n$ (here $r = 20$)

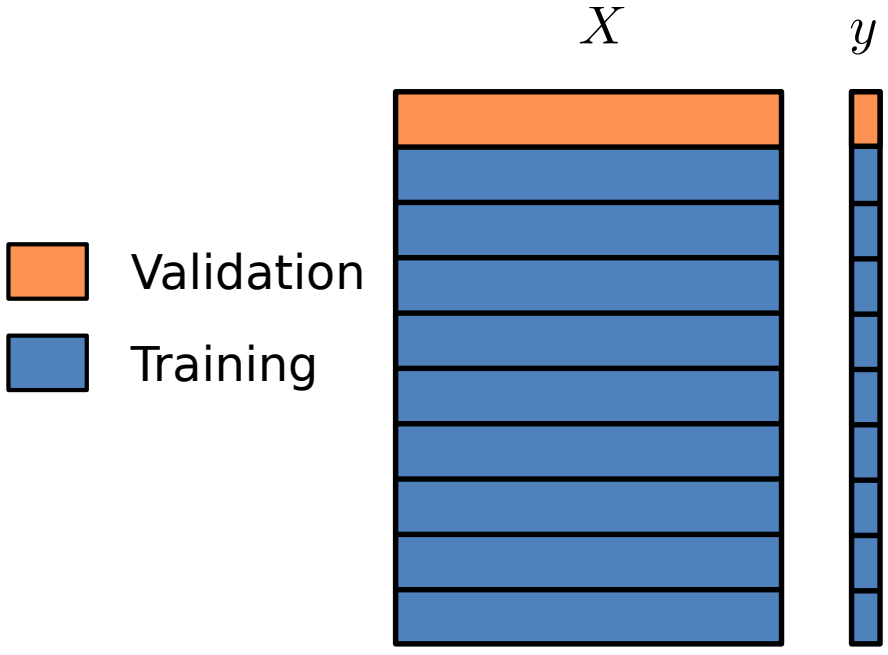
K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples



K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

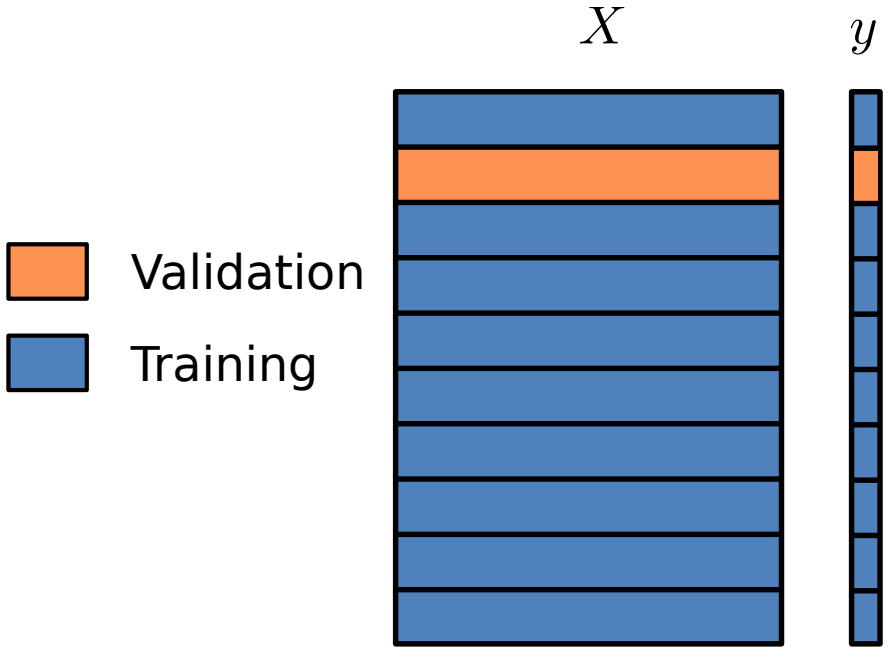


$k = 1$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

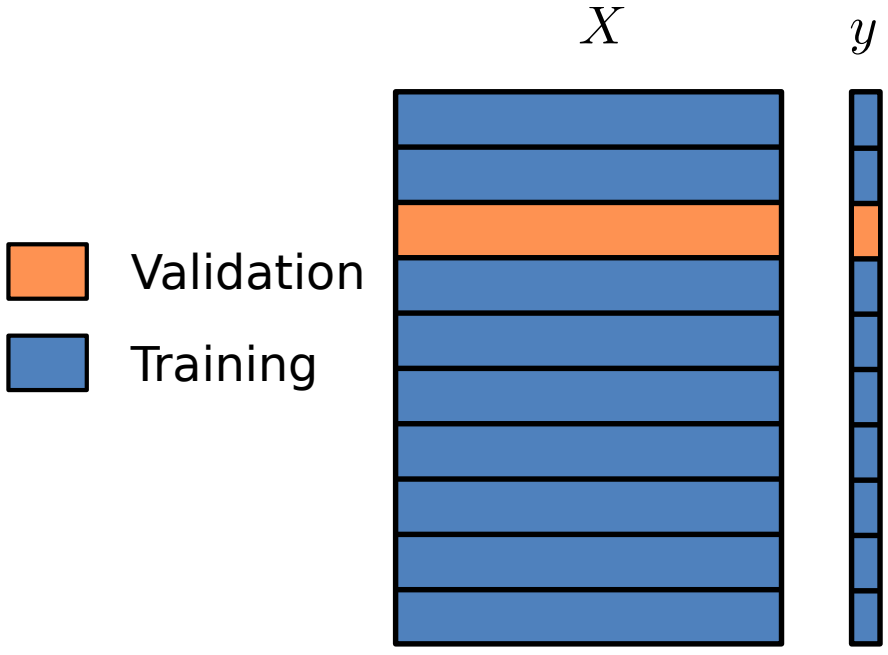


$k = 2$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

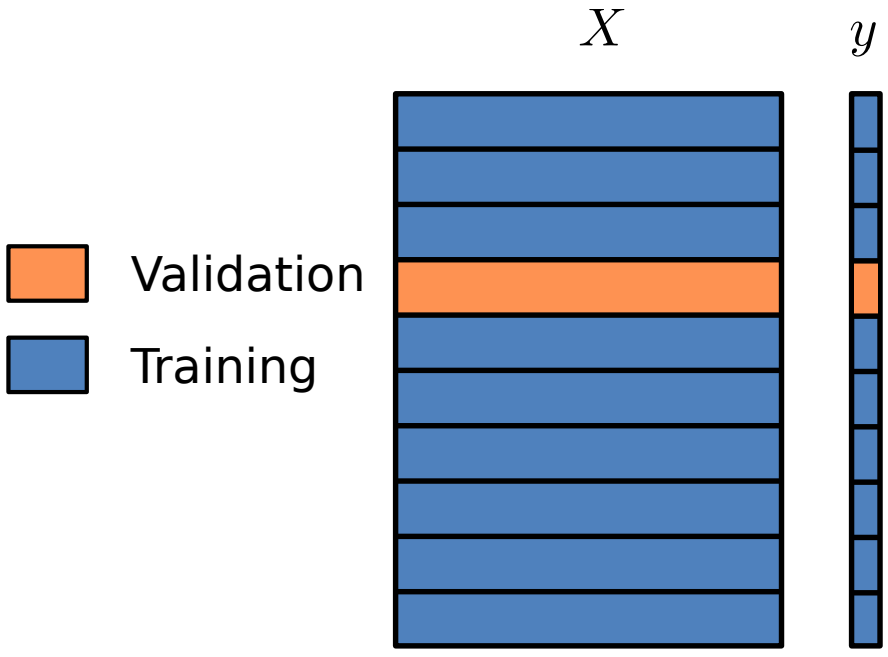


$$k = 3$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

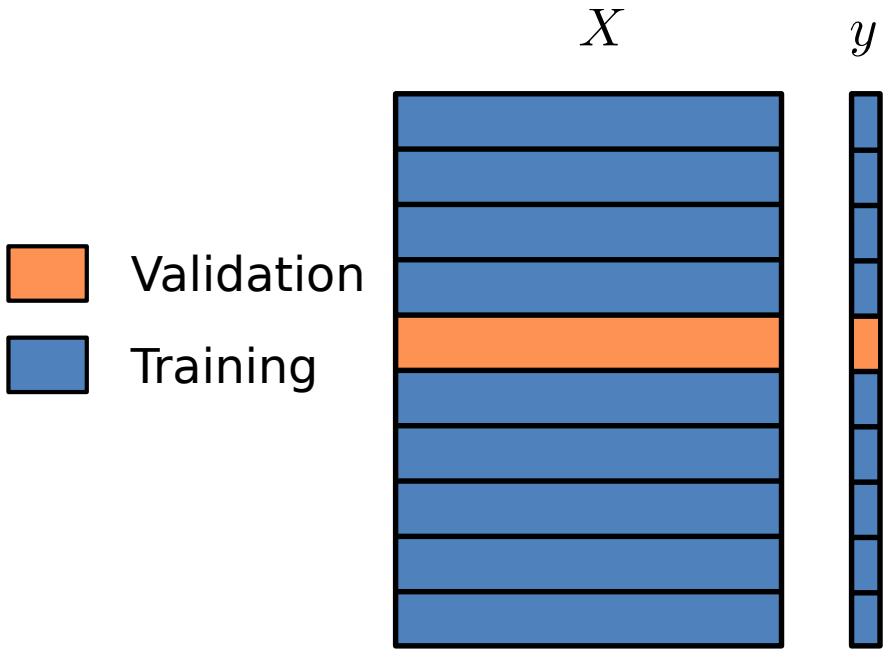


$$k = 4$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

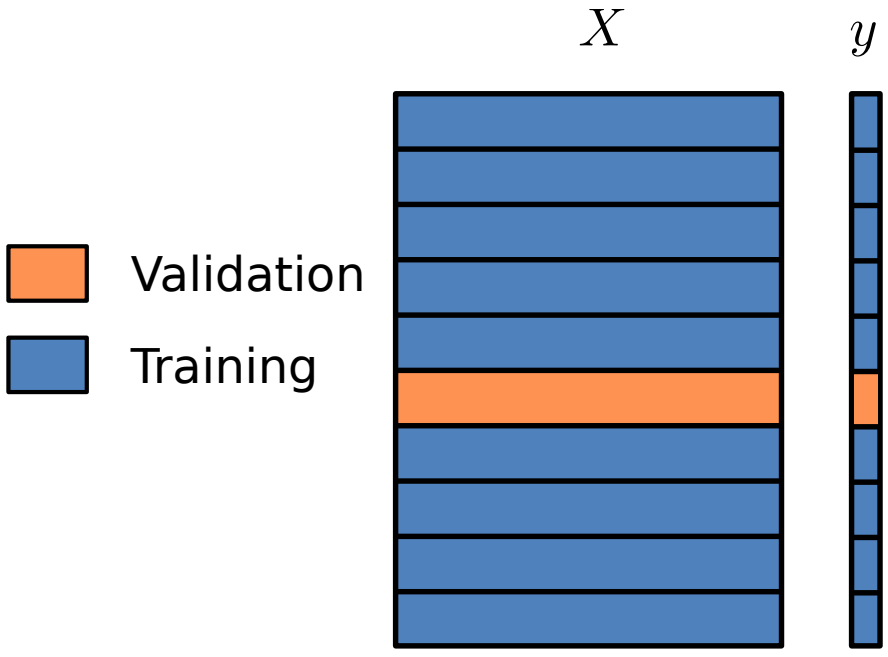


$k = 5$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

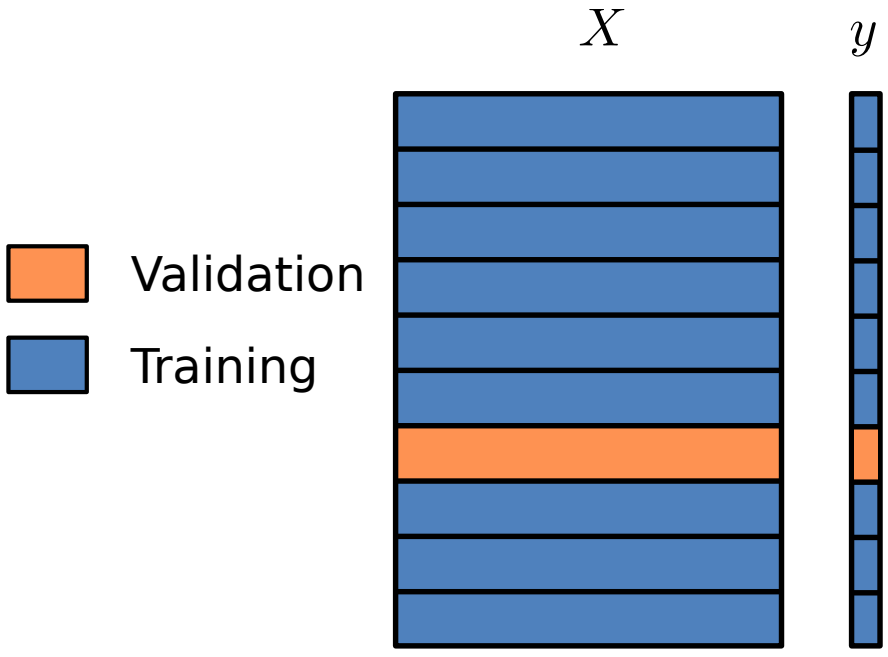


$$k = 6$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

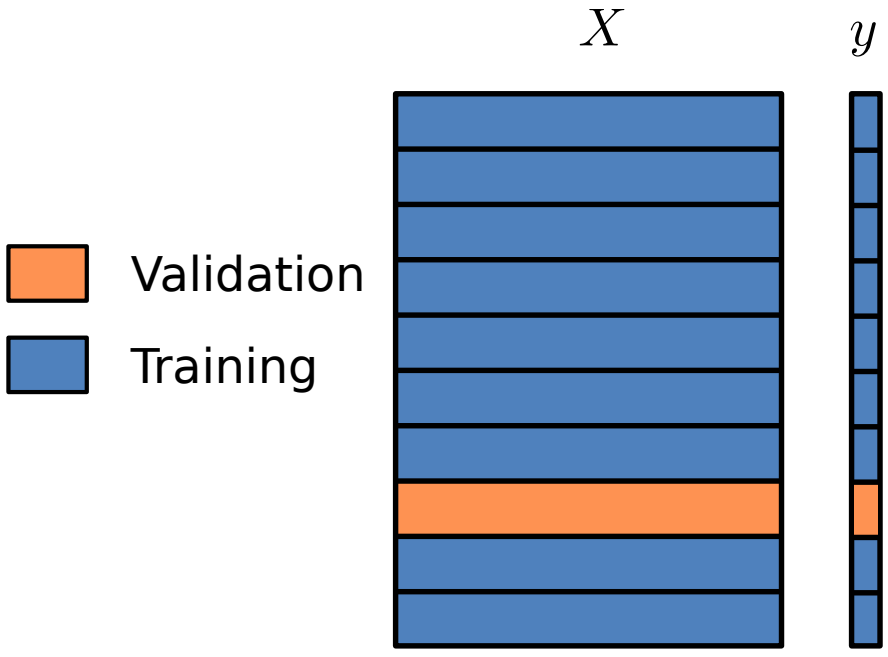


$$k = 7$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

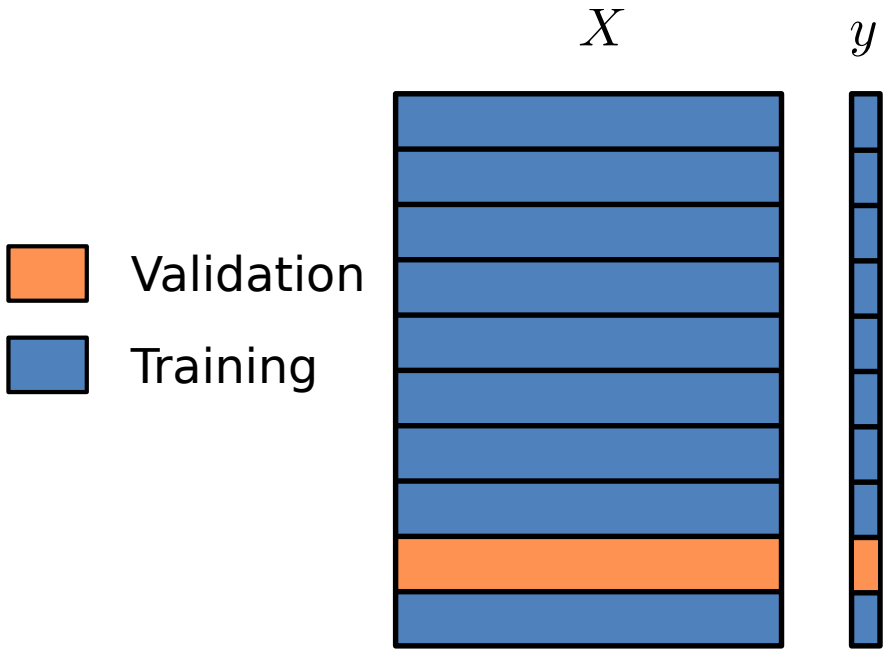


$$k = 8$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

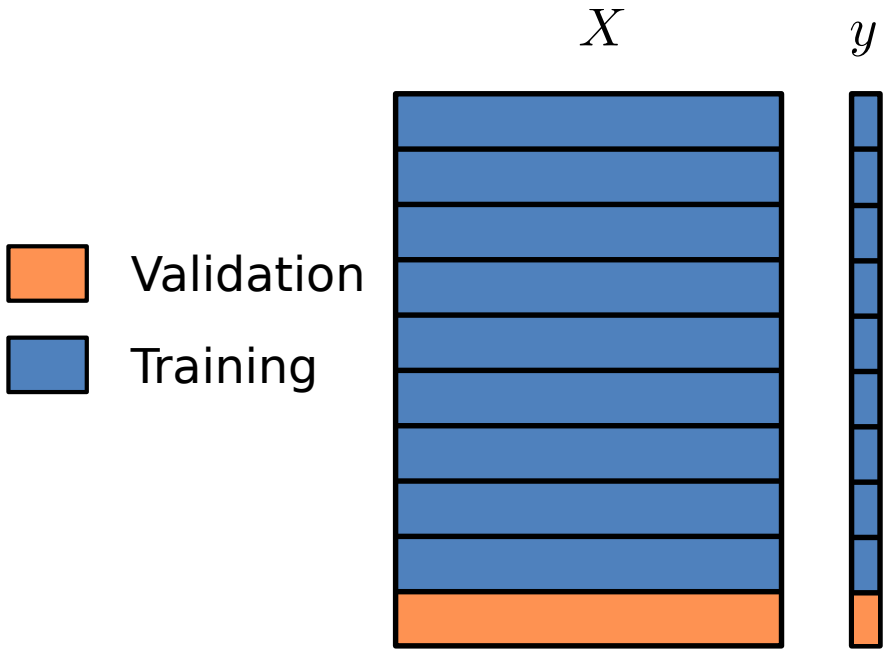


$$k = 9$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples

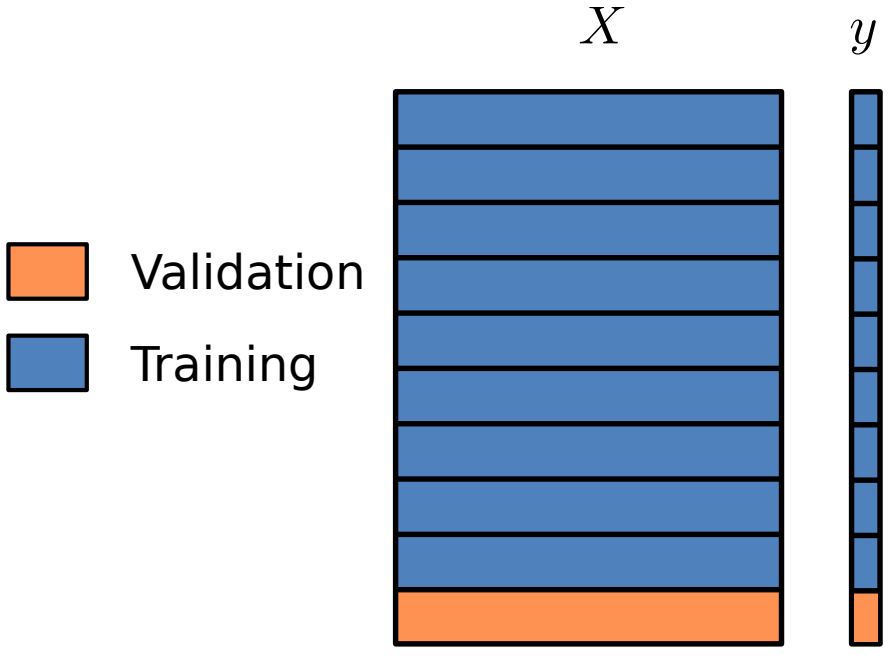


$$k = 10$$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\text{Error}_1^k, \ldots, \text{Error}_r^k$

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples



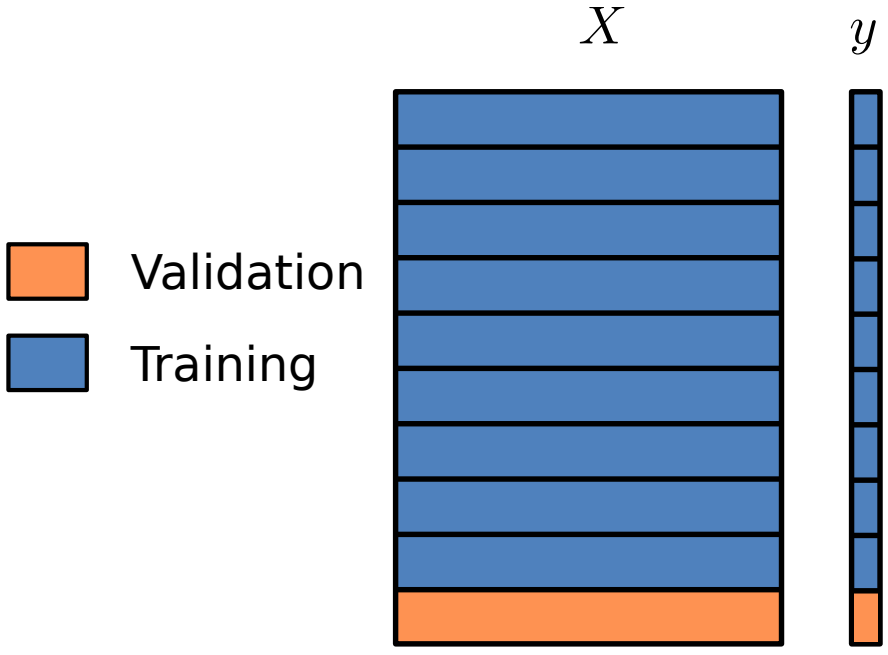
$k = 10$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\mathrm{Error}_1^k, \ldots, \mathrm{Error}_r^k$

Final Step: average the prediction errors get $\widehat{\mathrm{Error}}_1, \ldots, \widehat{\mathrm{Error}}_r$ and choose $\hat{i}^{\mathrm{CV}} \in [r]$ achieving the smallest error

K-fold Cross Validation ($K = 10$)

Divide (X, y) in K fold along the samples



$k = 10$

1. Compute over the training set the estimators for $\lambda_1 > \cdots > \lambda_r$, get $\hat{\beta}^{\lambda_1}, \ldots, \hat{\beta}^{\lambda_r}$
2. Compute the prediction errors using the validation set and get $\mathrm{Error}_1^k, \ldots, \mathrm{Error}_r^k$

Final Step: average the prediction errors get $\widehat{\mathrm{Error}}_1, \ldots, \widehat{\mathrm{Error}}_r$ and choose $\hat{i}^{\mathrm{CV}} \in [r]$ achieving the smallest error

Final Step (bis): compute $\hat{\beta}^{\lambda_i}$ over the whole (X, Y) for $i = \hat{i}^{\mathrm{CV}}$

Drawbacks of K-fold Cross Validation

Say $K = 10$

Computational limits

Naive method:

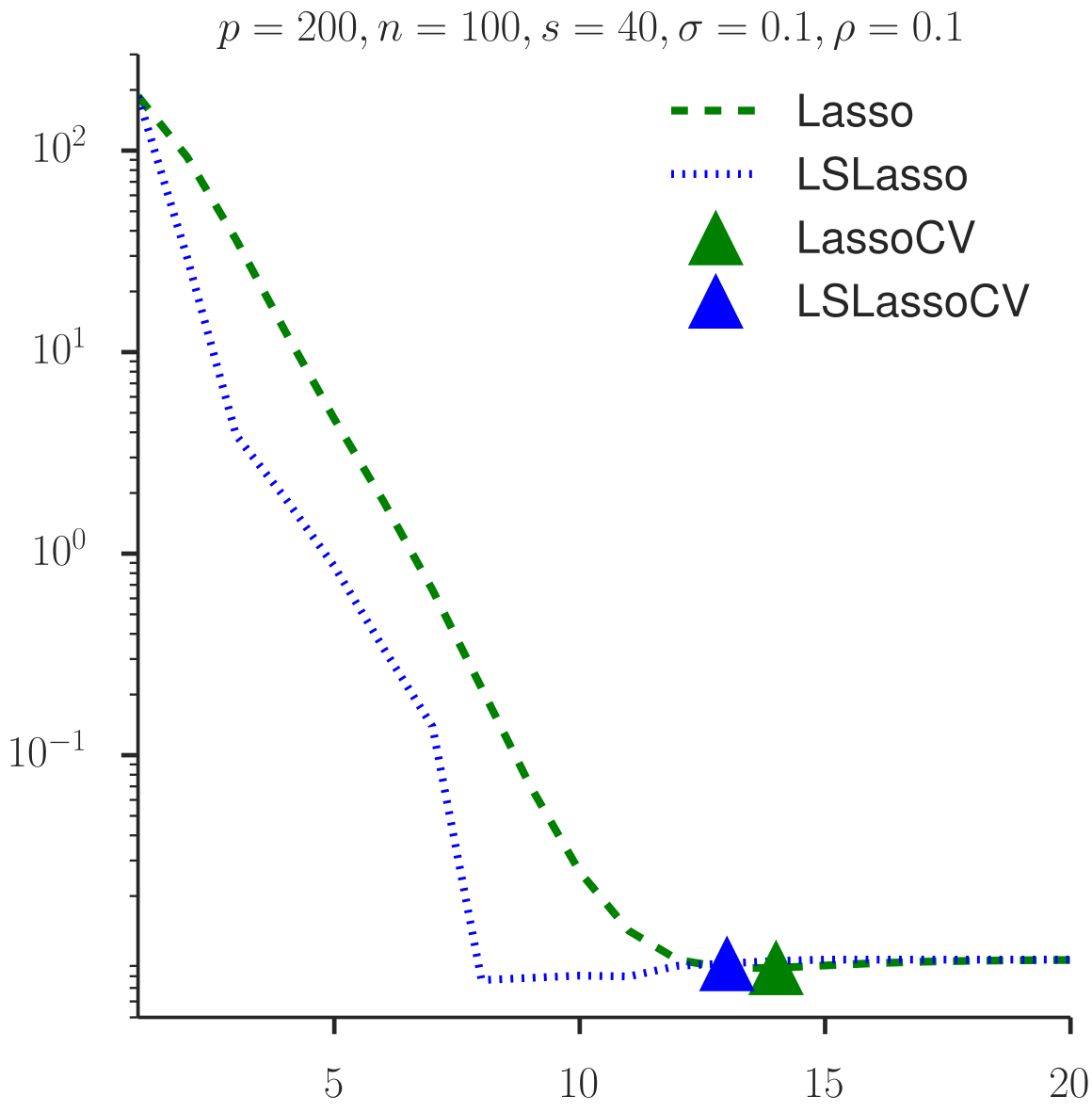
- compute 10 times the estimator over datasets of size 90% of the original one
- compute 1 time the estimator over 100% of the dataset

Theoretical limits

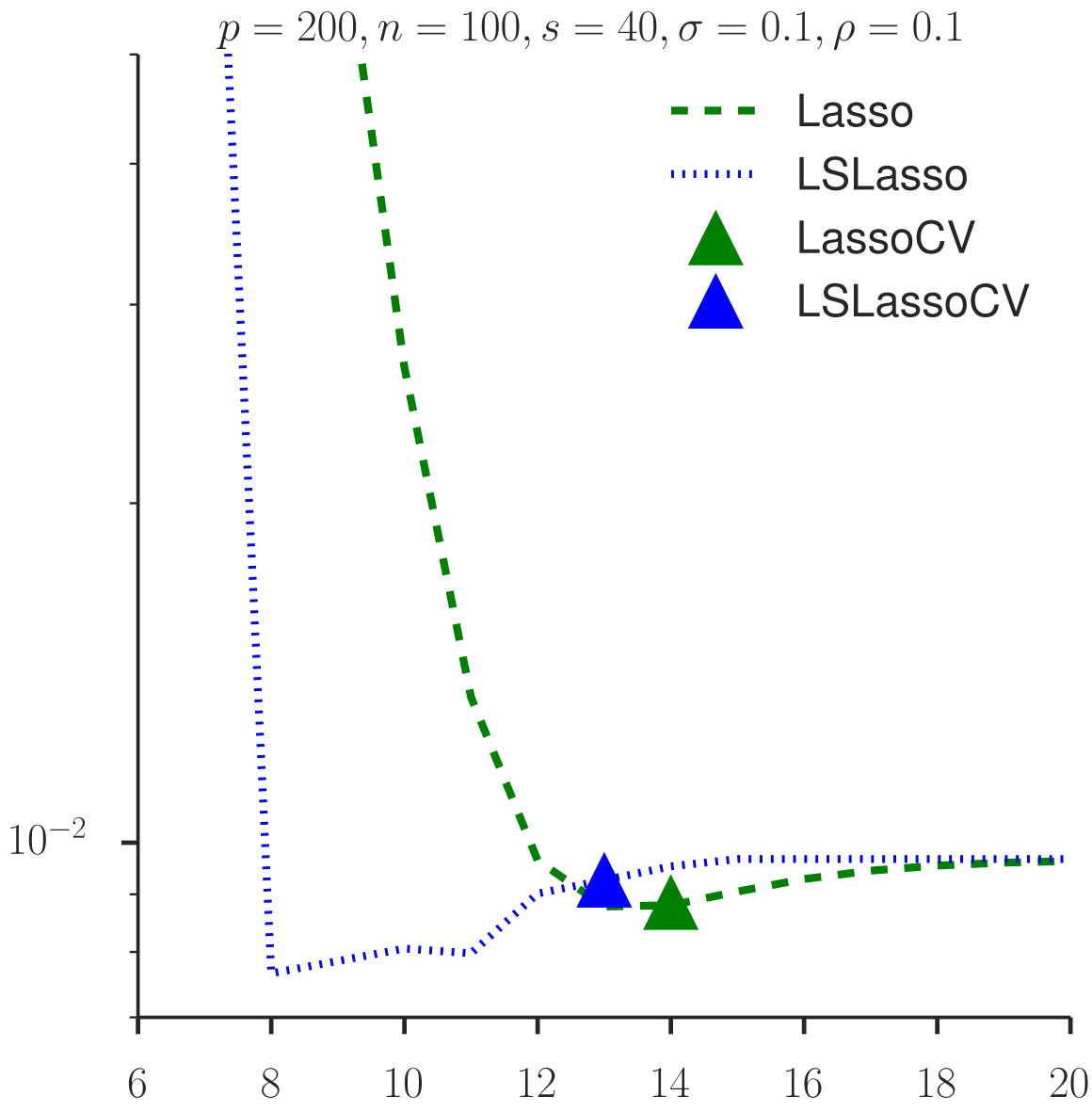
- basic results suppose estimators are (almost) "independent"
- little is known on the performance of cross-validated Lasso / LSLasso type methods

cf. Arlot and Celisse (2010)

. . . but a practical interest



. . . but a practical interest



Notation for defining a new validation method

Parameters: $\lambda_1 \quad \cdots \quad \lambda_r$

Notation for defining a new validation method

Parameters:	λ_1	$\cdots$	λ_r
	$\downarrow$	$\cdots$	$\downarrow$
Lasso estimators:	$\hat{\beta}^{\lambda_1}_{\text{Lasso}}$	$\cdots$	$\hat{\beta}^{\lambda_r}_{\text{Lasso}}$

Notation for defining a new validation method

Parameters:	λ_1	$\cdots$	λ_r
	$\downarrow$	$\cdots$	$\downarrow$
Lasso estimators:	$\hat{\beta}^{\lambda_1}_{\text{Lasso}}$	$\cdots$	$\hat{\beta}^{\lambda_r}_{\text{Lasso}}$
	$\downarrow$	$\cdots$	$\downarrow$
Active sets:	$\hat{S}^1$	$\cdots$	$\hat{S}^r$

Notation for defining a new validation method

Parameters:	λ_1	$\cdots$	λ_r
	$\downarrow$	$\cdots$	$\downarrow$
Lasso estimators:	$\hat{\beta}^{\lambda_1}_{\text{Lasso}}$	$\cdots$	$\hat{\beta}^{\lambda_r}_{\text{Lasso}}$
	$\downarrow$	$\cdots$	$\downarrow$
Active sets:	$\hat{S}^1$	$\cdots$	$\hat{S}^r$
	$\downarrow$	$\cdots$	$\downarrow$
LSLasso estimators:	$\hat{\beta}^{\lambda_1}_{\text{LSLasso}}$	$\cdots$	$\hat{\beta}^{\lambda_r}_{\text{LSLasso}}$

Notation for defining a new validation method

$$
\begin{array}{lccc}
\text{Parameters:} & \lambda_1 & \cdots & \lambda_r \\
& \downarrow & \cdots & \downarrow \\
\text{Lasso estimators:} & \hat{\beta}^{\lambda_1}_{\text{Lasso}} & \cdots & \hat{\beta}^{\lambda_r}_{\text{Lasso}} \\
& \downarrow & \cdots & \downarrow \\
\text{Active sets:} & \hat{S}^1 & \cdots & \hat{S}^r \\
& \downarrow & \cdots & \downarrow \\
\text{LSLasso estimators:} & \hat{\beta}^{\lambda_1}_{\text{LSLasso}} & \cdots & \hat{\beta}^{\lambda_r}_{\text{LSLasso}}
\end{array}
$$

For simplicity, denote:

- $\overline{\beta}^i = \hat{\beta}^{\lambda_i}_{\text{LSLasso}}$ (*i.e.*, the least square estimator over the variables in $\hat{S}^i$)
- $\overline{\beta}^{i,j}$ the least square estimator over the variable in $\hat{S}^i \cup \hat{S}^j$

Rem: there are at most r^2 such estimators; not all needed for our purpose

Adaptive Validation in Prediction ($\mathsf{AV_p}$)

Assume the active sets are ordered such that: $|\hat{S}^1| \leq \ldots \leq |\hat{S}^r|$

Definition $\mathsf{AV_p}$

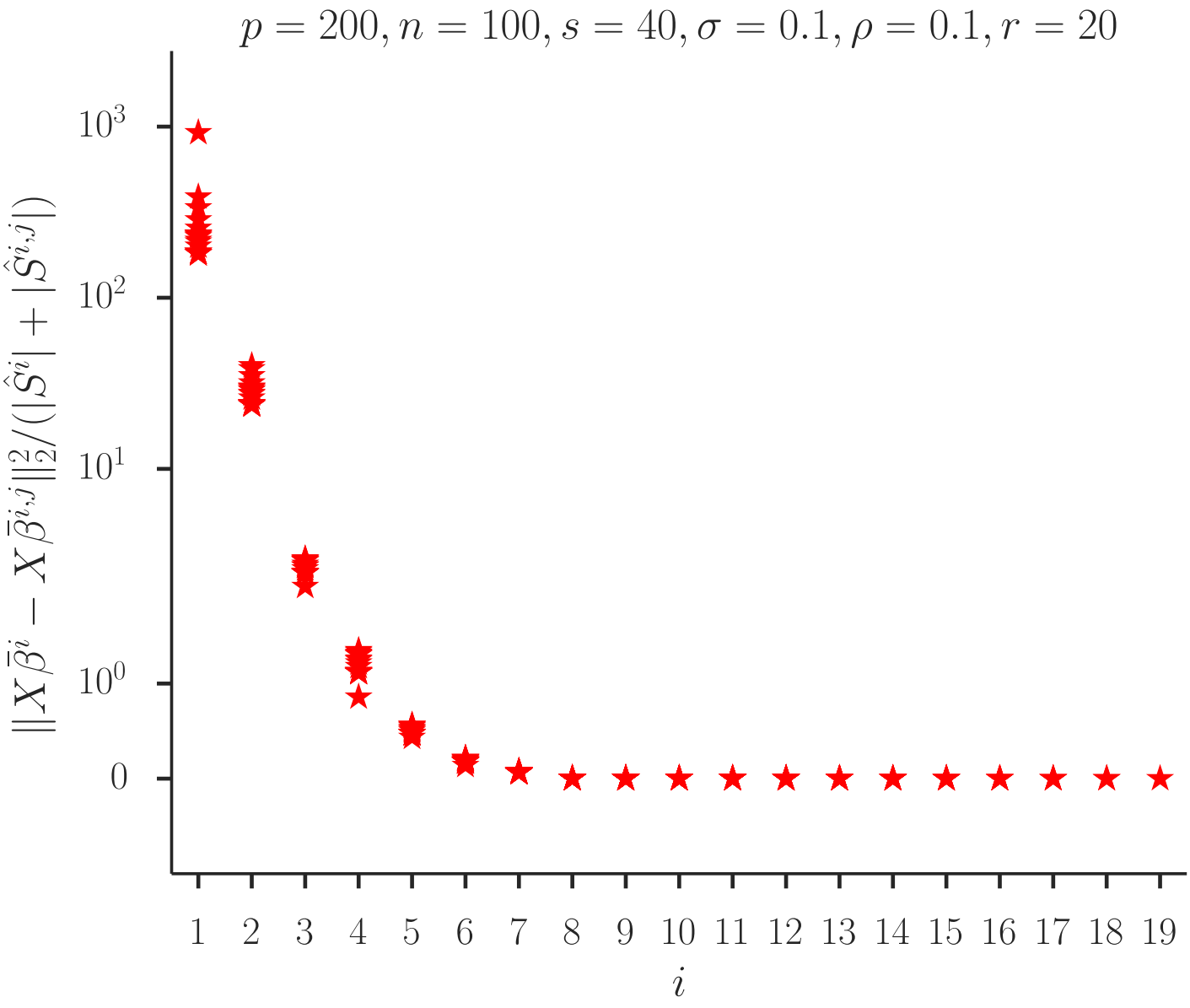
Let $a \geq 0$. The $\mathsf{AV_p}$ is the estimator $\overline{\beta} := \overline{\beta}^{\hat{i}}$ with active set $\hat{S} = \hat{S}^{\hat{i}}$, where

$$\hat{i} := \min \left\{ i \in [r-1] \,\middle|\, \frac{\|X\overline{\beta}^i - X\overline{\beta}^{i,j}\|_2^2}{|\hat{S}^i| + |\hat{S}^{i,j}|} \leq a, \forall j \in [r] : |\hat{S}^j| \geq |\hat{S}^i| \right\}$$

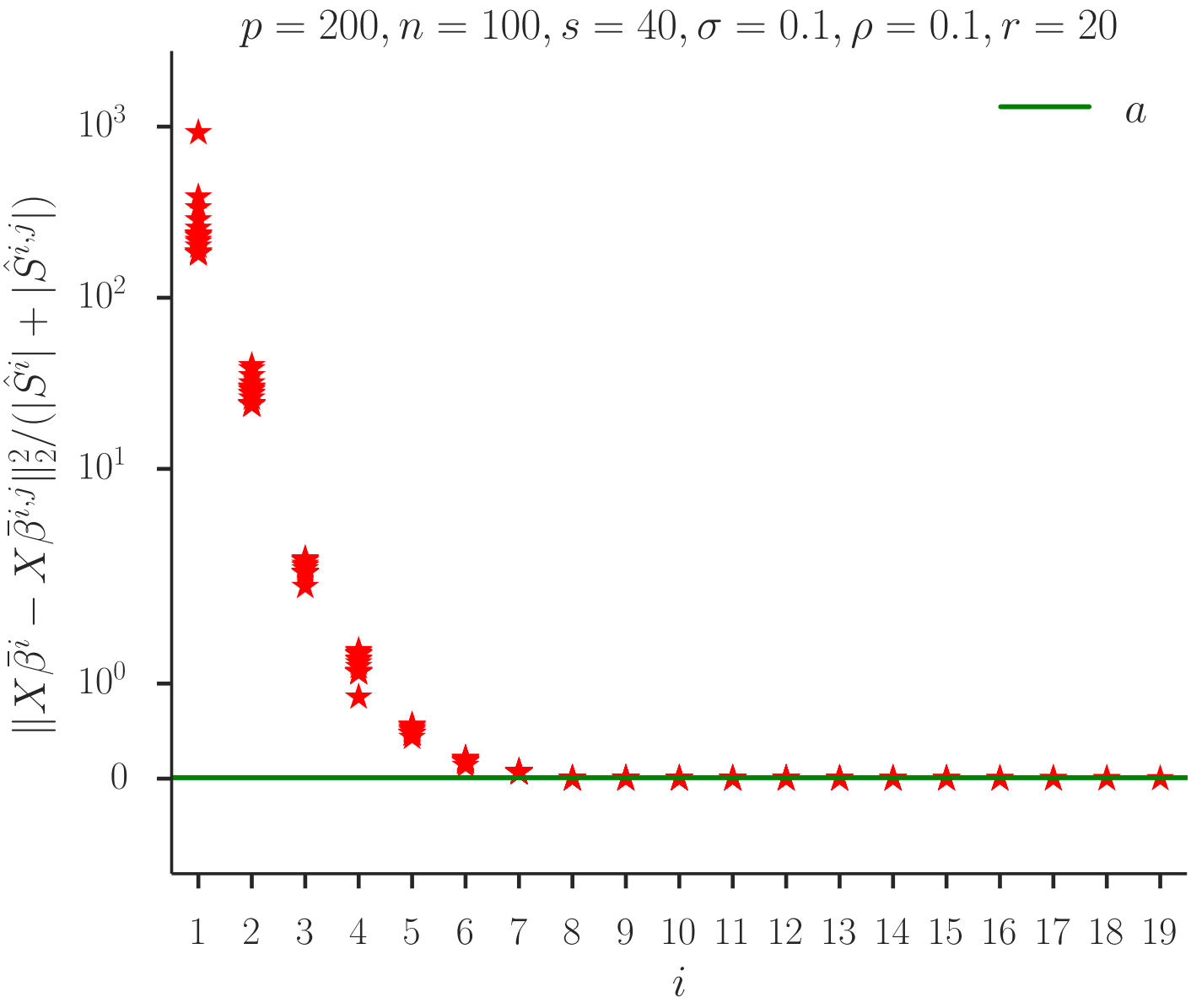
when the minimum exists, and $\hat{i} = r$ otherwise.

- a is a tuning parameter, as was K for Cross-Validation (to be discuss later)
- $\|X\overline{\beta}^i - X\overline{\beta}^{i,j}\|_2^2$ measures the variation in prediction obtained by adding the variable from the set $\hat{S}^j$ to $\hat{S}^i$

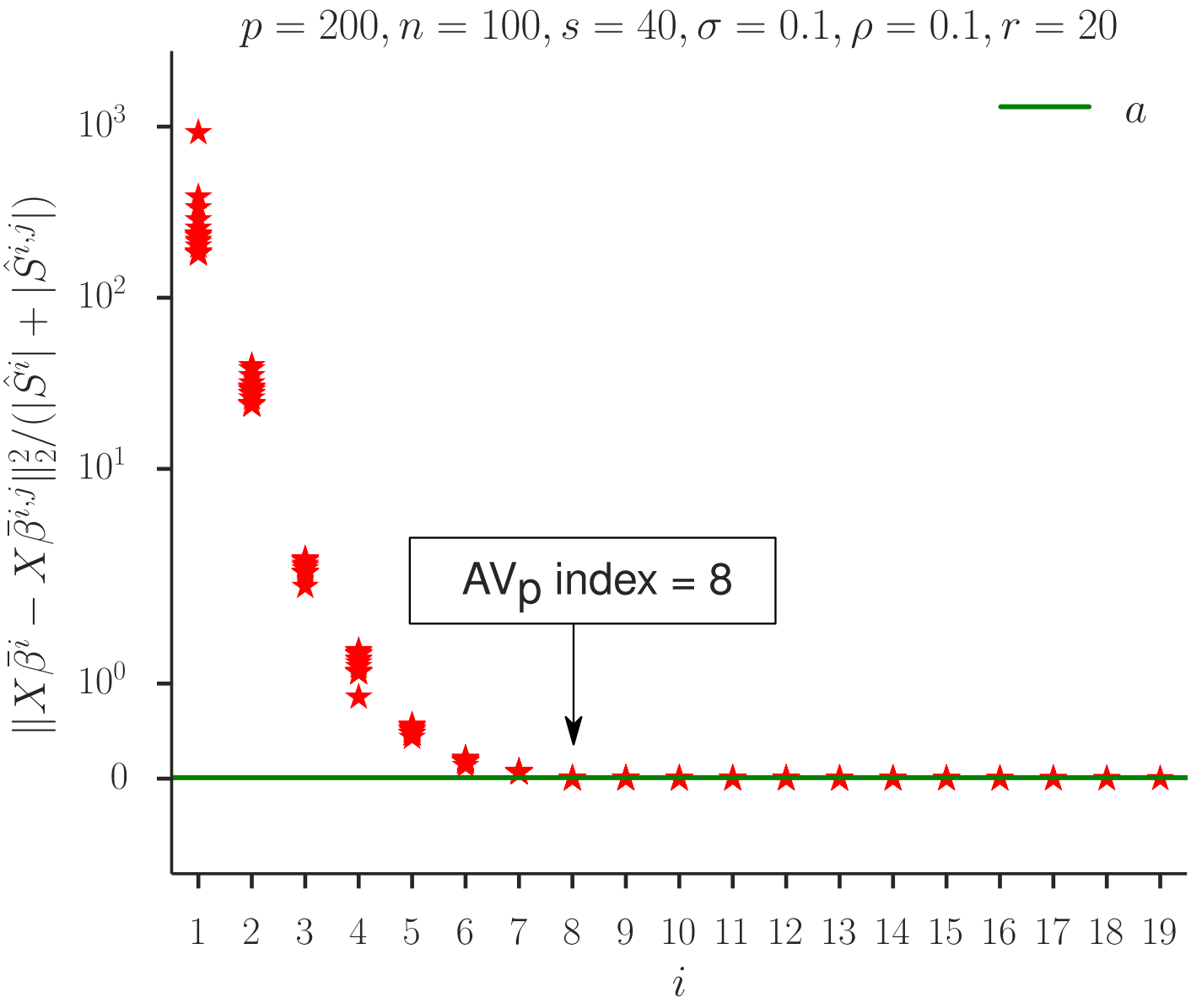
What does it mean?



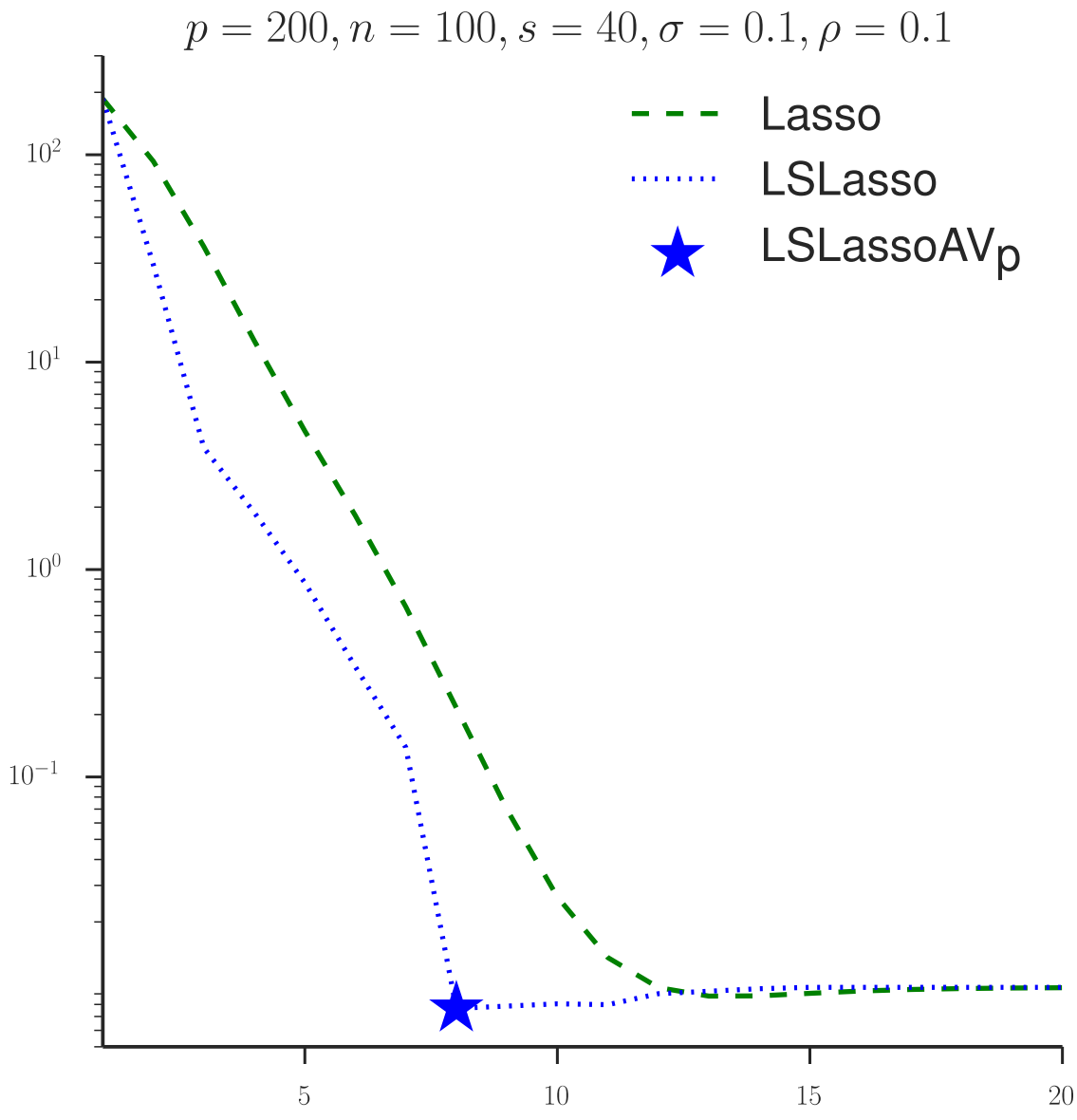
What does it mean?



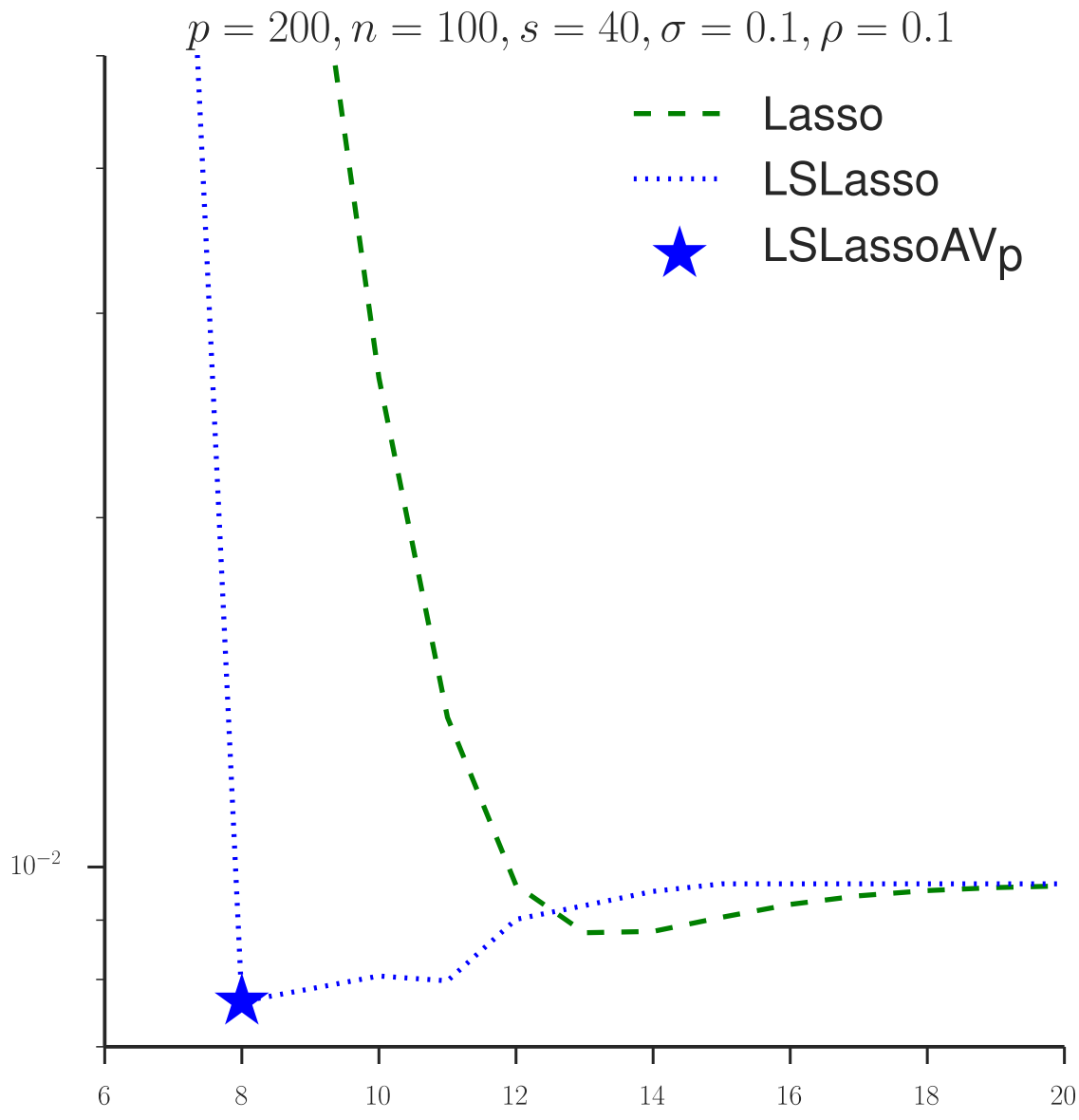
What does it mean?



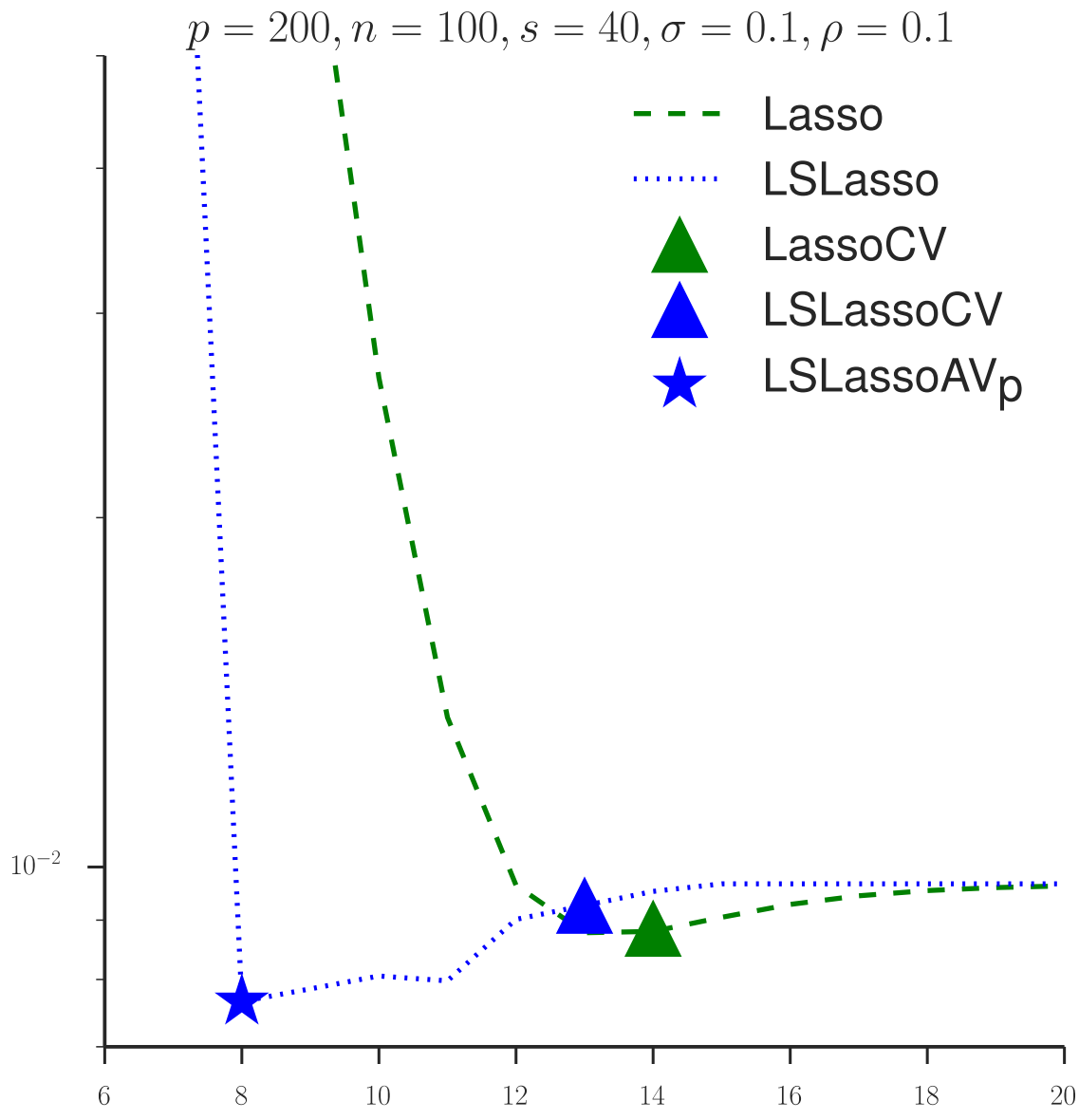
Performance in prediction



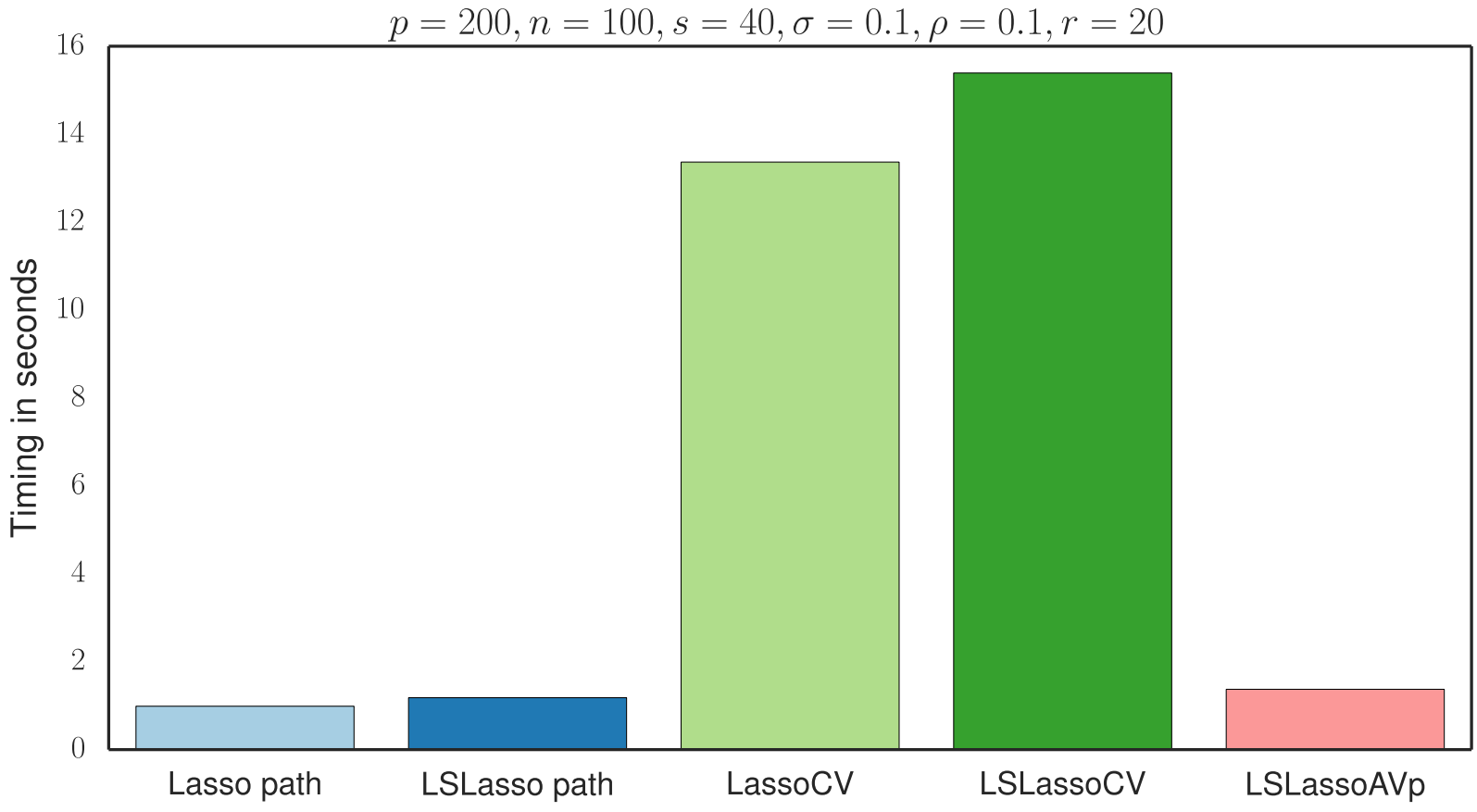
Performance in prediction



Performance in prediction



Time efficiency



$\mathbf{AV_p}$: the algorithm

Data: $Y, X, \hat{S}^1, \ldots, \hat{S}^r, a$
Result: $\widehat{i} \in [r], \overline{\beta} \in \mathbb{R}^p$
Initialize index: $i \leftarrow 1$
while $i \leq r-1$ **do**
  Initialize stopping $TestFailure \leftarrow False$ and compute $\overline{\beta}^i$
  Initialize comparisons: $j \leftarrow \min\{k \in [i] |\ |\hat{S}^k| \geq |\hat{S}^i|\}$
  while *($j \leq r$) & (TestFailure==False)* **do**
    Compute $\hat{S}^{i,j}$ and $\overline{\beta}^{i,j}$
    if $\|X\overline{\beta}^i - X\overline{\beta}^{i,j}\|_2^2 \leq a|\hat{S}^i| + a|\hat{S}^{i,j}|$ **then**
      $j \leftarrow j+1$
    else
      $TestFailure \leftarrow True$
    end
  end
  if $TestFailure == True$ **then**
    $i \leftarrow i+1$
  else
    break
  end
end

$\widehat{i} \leftarrow i$ and $\overline{\beta} \leftarrow \overline{\beta}^{\widehat{i}}$

Choice of a

When σ^2 (noise level) is known

In theory: a should be chosen proportionally to σ^2
In practice: $a = \sigma^2$ works fine ... when σ^2 is known

When σ^2 (noise level) is known

Need to estimate σ^2 with often $p > n$

- ℓ_1-penalized maximum likelihood over (β, σ^2) Stadler *et al.* 2010
- Square root Lasso / Scaled Lasso Antoniadis (2010) , Belloni *et al.* (2011) , Sun and Zhang (2012)

Choice of a

When σ^2 (noise level) is known

In theory: a should be chosen proportionally to σ^2
In practice: $a = \sigma^2$ works fine ... when σ^2 is known

When σ^2 (noise level) is known

Need to estimate σ^2 with often $p > n$

- ℓ_1-penalized maximum likelihood over (β, σ^2) Stadler *et al.* 2010
- Square root Lasso / Scaled Lasso Antoniadis (2010) , Belloni *et al.* (2011) , Sun and Zhang (2012)

$$\text{Lasso} \quad \begin{cases} \hat{\beta}^\lambda \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\boxed{\frac{1}{2}\|y - X\beta\|_2^2} + \lambda \|\beta\|_1 \right) \\ \hat{\sigma}^2 = \|\hat{\beta}^\lambda - y\|_2^2 / n \end{cases}$$

Choice of a

When σ^2 (noise level) is known

In theory: a should be chosen proportionally to σ^2
In practice: $a = \sigma^2$ works fine ... when σ^2 is known

When σ^2 (noise level) is known

Need to estimate σ^2 with often $p > n$

- ℓ_1-penalized maximum likelihood over (β, σ^2) Stadler *et al.* 2010
- Square root Lasso / Scaled Lasso Antoniadis (2010) , Belloni *et al.* (2011) , Sun and Zhang (2012)

Square root Lasso

$$\begin{cases} \hat{\beta}^\lambda \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\boxed{\frac{1}{2}\|y - X\beta\|_2} + \lambda \|\beta\|_1 \right) \\ \hat{\sigma}^2 = \|\hat{\beta}^\lambda - y\|_2^2 / n \end{cases}$$

Choice of a

When σ^2 (noise level) is known

In theory: a should be chosen proportionally to σ^2
In practice: $a = \sigma^2$ works fine ... when σ^2 is known

When σ^2 (noise level) is known

Need to estimate σ^2 with often $p > n$

- ℓ_1-penalized maximum likelihood over (β, σ^2) Stadler *et al.* 2010
- Square root Lasso / Scaled Lasso Antoniadis (2010) , Belloni *et al.* (2011) , Sun and Zhang (2012)

Square root Lasso

$$\begin{cases} \hat{\beta}^\lambda \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\boxed{\frac{1}{2}\|y - X\beta\|_2} + \lambda \|\beta\|_1 \right) \\ \hat{\sigma}^2 = \|\hat{\beta}^\lambda - y\|_2^2 / n \end{cases}$$

Rem: Reasonable rough estimation of $\hat{\sigma}^2$ for $\lambda = \sqrt{2n \log p}$

$\mathsf{AV_p}$: theory behind

- Non-parametric statistics: $\mathsf{AV_p}$ idea $\approx$ the **Lepski method** Lepski (1990), Lepski, Mammen and Spokoiny (1997), Chichignoud, Lederer and Wainwright (2014)
- Image processing: popularized as the **ICI** for Intersection of Intervals of Confidence Katkovnik (1999)

Theorem for the $\mathsf{AV_p}$ applied to the Lasso: $\overline{\beta} = \overline{\beta}^{\widehat{i}}$

For a large enough, under some technical assumption over the estimators $\overline{\beta}^{i,j}$, then **with high probability** it holds that

$$\|X\overline{\beta} - X\beta\|_2^2 \leq 8a|\widetilde{S}|$$

where $\widetilde{S}$ being the smallest active set among $\hat{S}^1, \ldots, \hat{S}^r$ containing the true support $\mathrm{supp}(\beta^*)$

Conclusion

Take home message

- New technique for tuning parameters of Lasso but not only: Sqrt-root Lasso, Thresholded Ridge/Tikohnov Regression.
- (Partial) theoretical guarantees
- Computationally more efficient than CV

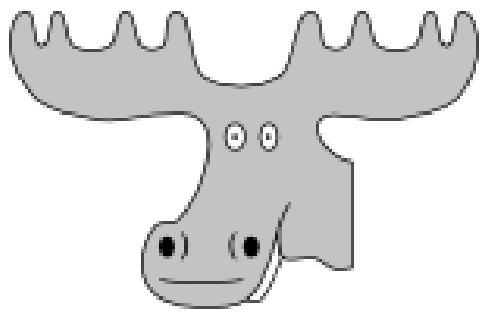
Future work

- Improve the noise estimation step
- Encompass more method in the framework
- Generalized to other noise model

More info

Website: `http://josephsalmon.eu`

- Article on ArXiV next week
- Python Code on demand (soon on authors webpage)
- Slides online after the talk



Powered with **MooseTeX**

Références I

- S. Arlot and A. Celisse.
  A survey of cross-validation procedures for model selection.
  Statistics surveys, 4:40–79, 2010.

- A. Antoniadis.
  Comments on: ℓ_1-penalization for mixture regression models.
  TEST, 19(2):257–258, 2010.

- A. Belloni, V. Chernozhukov, and L. Wang.
  Square-root Lasso: Pivotal recovery of sparse signals via conic programming.

  Biometrika, 98(4):791–806, 2011.

- S. S. Chen, D. L. Donoho, and M. A. Saunders.
  Atomic decomposition by basis pursuit.
  SIAM J. Sci. Comput., 20(1):33–61 (electronic), 1998.

- M. Chichignoud, J. Lederer, and M. Wainwright.
  Tuning lasso for sup-norm optimality.
  ArXiv e-prints, 2014.

Références II

- V. Katkovnik.
  A new method for varying adaptive bandwidth selection.
  IEEE Trans. Image Process., 47(9):2567–2571, 1999.

- O. V. Lepski.
  On a problem of adaptive estimation in gaussian white noise.
  Theory of Probability and its Applications, 35(3):454–466, 1990.

- O. V. Lepski, E. Mammen, and V. G. Spokoiny.
  Optimal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selectors.
  Ann. Statist., 25(3):929–947, 1997.

- J. Mairal, F. Bach, J. Ponce, and G. Sapiro.
  Online learning for matrix factorization and sparse coding.
  J. Mach. Learn. Res., pages 19–60, 2010.

- N. Städler, P. Bühlmann, and Sara s van de Geer.
  ℓ_1-penalization for mixture regression models.
  TEST, 19(2):209–256, 2010.

Références III

- T. Sun and C.-H. Zhang.

  Scaled sparse linear regression.

  Biometrika, 99(4):879–898, 2012.

- R. Tibshirani.

  Regression shrinkage and selection via the lasso.

  J. Roy. Statist. Soc. Ser. B, 58(1):267–288, 1996.