SD-TSIA204 : PCA and LASSO

Ekhine Irurozki
Télécom Paris, IP Paris

Lasso : Reminding the model

$$\mathbf{y} = X\boldsymbol{\theta}^{\star} + \boldsymbol{\varepsilon} \in \mathbb{R}^{n}$$

$$X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times p}, \boldsymbol{\theta}^{\star} \in \mathbb{R}^{p}$$

Motivation

In the presence of super-collinearity the OLS estimators can not be given.

Estimators $\hat{\boldsymbol{\theta}}$ with many zero coefficients are useful :

- for interpretation
- for computational efficiency if p is huge

Underlying idea : **variable selection**

Rem: also useful if $\boldsymbol{\theta}^{\star}$ has few non-zero coefficients

Variable selection overview

- Screening : remove the $\mathbf{x}_j$'s whose correlation with $\mathbf{y}$ is weak
  - pros : fast $(+++)$, *i.e.*,one pass over data, intuitive $(+++)$
  - cons : neglect variables interactions $\mathbf{x}_j$, weak theory (- - -)
- Greedy methods aka stagewise / stepwise
  - pros : fast $(++)$, intuitive $(++)$
  - cons : propagates wrong selection forward ; weak theory (-)
- Sparsity enforcing penalized methods (*e.g.*,Lasso)
  - pros : better theory for convex cases $(++)$
  - cons : can be still slow (-)

The ℓ_0 pseudo-norm

The support of $\boldsymbol{\theta} \in \mathbb{R}^p$ is the set of indexes of non-zero coordinates :

$$\mathrm{supp}(\boldsymbol{\theta}) = \{j \in [\![1, p]\!], \theta_j \neq 0\}$$

The ℓ_0 pseudo-norm of a $\boldsymbol{\theta} \in \mathbb{R}^p$ is the number of non-zero coordinates :

$$\|\boldsymbol{\theta}\|_0 = \mathrm{card}\{j \in [\![1, p]\!], \theta_j \neq 0\}$$

<u>Rem</u>: $\|\cdot\|_0$ is not a norm, $\forall t \in \mathbb{R}^*, \|t\boldsymbol{\theta}\|_0 = \|\boldsymbol{\theta}\|_0$

<u>Rem</u>: $\|\cdot\|_0$ it is not even convex, $\boldsymbol{\theta}_1 = (1, 0, 1, \dots, 0)$ $\boldsymbol{\theta}_2 = (0, 1, 1, \dots, 0)$ and $3 = \|\frac{\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2}{2}\|_0 \geqslant \frac{\|\boldsymbol{\theta}_1\|_0 + \|\boldsymbol{\theta}_2\|_0}{2} = 2$

Regularization with the ℓ_0 penalty

First try to get a sparsity enforcing penalty : use ℓ_0 as a penalty (or regularization)

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_0}_{\text{regularization}} \right)}$$

Combinatorial problem ! ! !

Exact solution : require considering all sub-models, *i.e.*,computing OLS for all possible support ; meaning one might need 2^p least squares computation !

Example :

$p = 10$ possible : $\approx 10^3$ least squares

$p = 30$ impossible : $\approx 10^{10}$ least squares

Rem: problem “NP-hard”, can be solved for small problems by mixed integer programming.

Regularization with the ℓ_1 penalty : Lasso

Lasso : *Least Absolute Shrinkage and Selection Operator* Tibshirani (1996)

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_1}_{\text{regularization}} \right)}$$

or $\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^{p} |\theta_j|$ sum of absolute values of the coefficients)

- We recover the limiting cases :

$$\lim_{\lambda\to 0} \hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}} = \hat{\boldsymbol{\theta}}^{\mathrm{OLS}}$$

$$\lim_{\lambda\to +\infty} \hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}} = 0 \in \mathbb{R}^p$$

Constraint point of view

The following problem :

$$\boxed{\hat{\boldsymbol{\theta}}_{\lambda}^{\mathrm{Lasso}} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_1}_{\text{regularization}} \right)}$$

shares the same solutions as the constrained formulation :

$$\begin{cases} \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ \text{s.t. } \|\boldsymbol{\theta}\|_1 \leqslant T \end{cases}$$

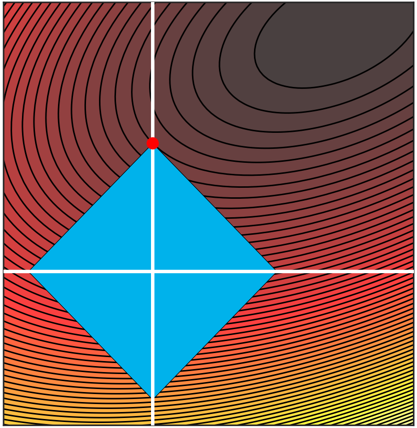
for some $T > 0$.

<u>Rem</u>: unfortunately the link $T \leftrightarrow \lambda$ is not explicit

- If $T \to 0$ one recovers the null vector : $0 \in \mathbb{R}^p$
- If $T \to \infty$ one recovers $\hat{\boldsymbol{\theta}}^{\mathrm{OLS}}$ (unconstrained)

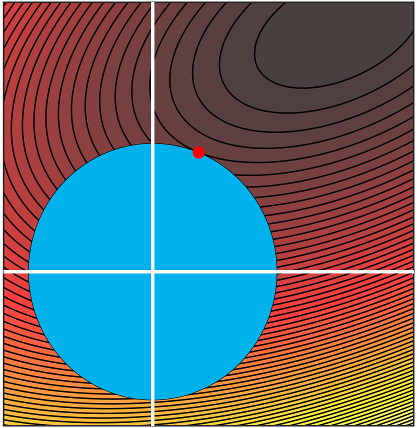
Interpretation : Optimization under ℓ_1 constraint, sparse solution

$$\underset{\boldsymbol{\theta}\in\mathbb{R}^p}{\arg\min} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$$
$$\text{s.t. } \|\boldsymbol{\theta}\|_1 \leqslant T$$



Interpretation : Optimization under ℓ_2 constraint, non-sparse solution

$$\begin{aligned} &\underset{\boldsymbol{\theta}\in\mathbb{R}^p}{\arg\min} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ &\text{s.t. } \|\boldsymbol{\theta}\|_2 \leqslant T \end{aligned}$$



Existance and uniqueness

<u>**Exercise**</u> : the Lasso estimator is not always **unique** for a fixed λ (consider cases with two equals columns in X). However, the prediction is unique. Show these points.

Analytical solution

Non-smooth problem

In general, there is no explicit solution

- Quadratic programming with constraints
- Iterative ridge
- Proximal gradient method

Sub-gradients / sub-differential

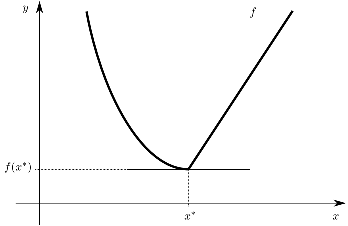
For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a sub-gradient of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The sub-differential is the set of all sub-gradients,

$$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}.$$

<u>Rem</u>: if the sub-gradient is unique, one recovers the standard gradient



Sub-gradients / sub-differential

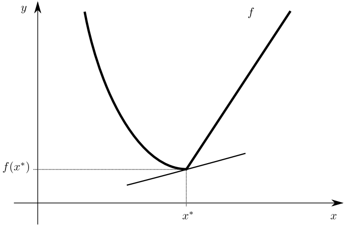
For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a sub-gradient of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The sub-differential is the set of all sub-gradients,

$$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}.$$

Rem: if the sub-gradient is unique, one recovers the standard gradient



Sub-gradients / sub-differential

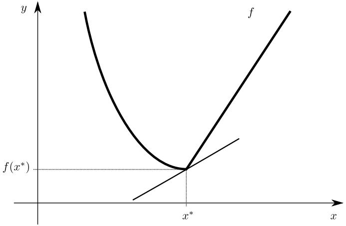
For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a sub-gradient of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The sub-differential is the set of all sub-gradients,

$$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}.$$

Rem: if the sub-gradient is unique, one recovers the standard gradient



Sub-gradients / sub-differential

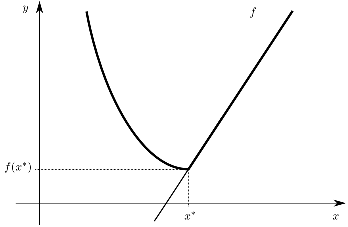
For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a sub-gradient of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The sub-differential is the set of all sub-gradients,

$$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}.$$

Rem: if the sub-gradient is unique, one recovers the standard gradient



Fermat's Rule : optimality of x^*

A point x^* is a minimum of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ if and only if $0 \in \partial f(x^*)$

Proof : use the sub-gradient definition :

- 0 is a sub-gradient of f at x^* if and only if $\forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle 0, x - x^* \rangle$

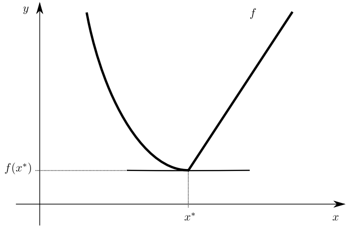
Fermat's Rule : optimality of x^*

A point x^* is a minimum of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ if and only if $0 \in \partial f(x^*)$

Proof : use the sub-gradient definition :

- 0 is a sub-gradient of f at x^* if and only if $\forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle 0, x - x^* \rangle$

Rem:Visually, it corresponds to a horizontal tangent



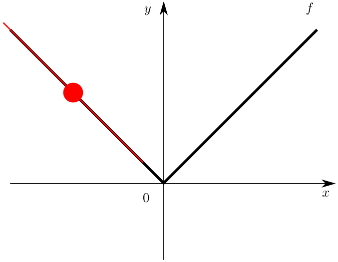
Absolute value sub-differential

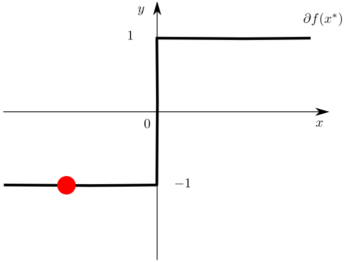
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





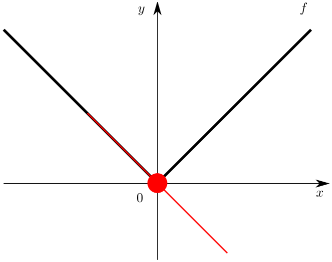
Absolute value sub-differential

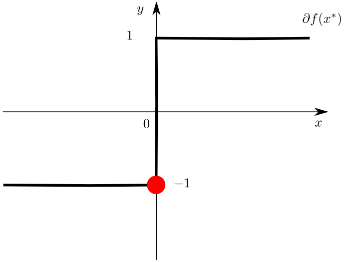
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





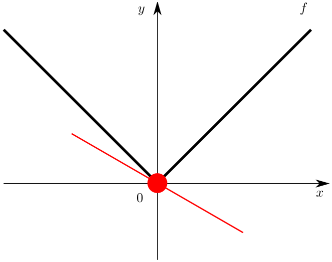
Absolute value sub-differential

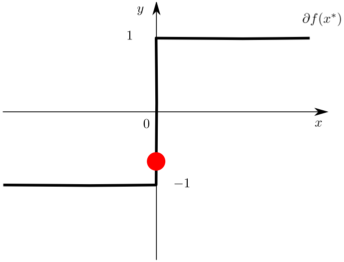
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





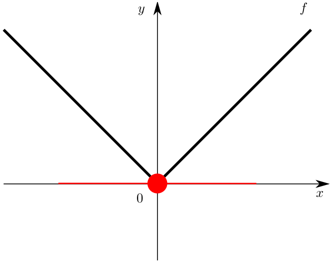
Absolute value sub-differential

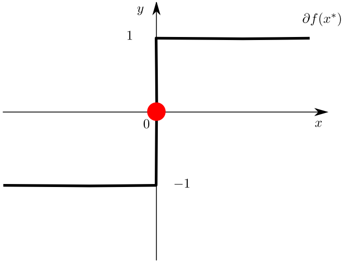
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





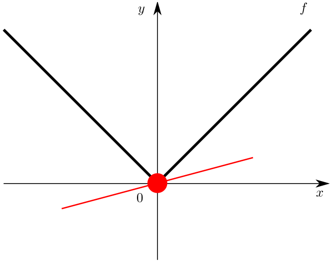
Absolute value sub-differential

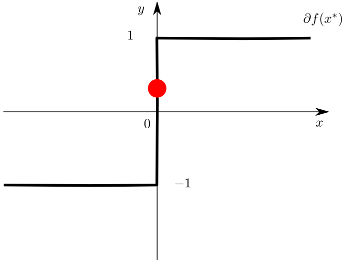
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





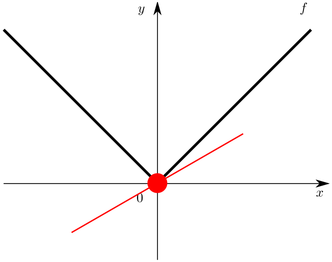
Absolute value sub-differential

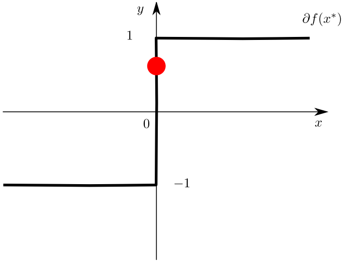
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





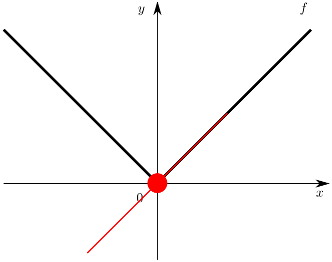
Absolute value sub-differential

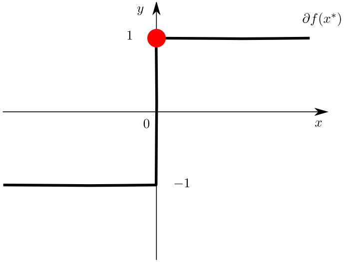
Function (abs) :

$$f:\begin{cases}\mathbb{R} & \to \mathbb{R}\\ x & \mapsto |x|\end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases}\{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0\end{cases}$$





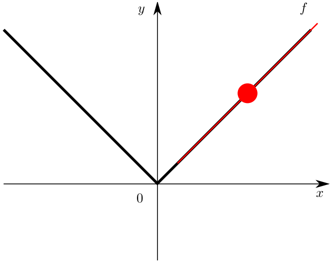
Absolute value sub-differential

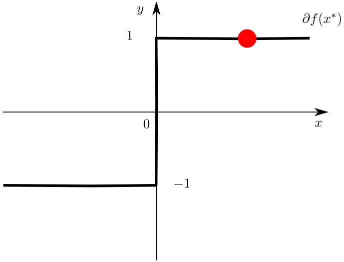
Function (abs) :

$$f : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$





Fermat's rule for the Lasso

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_1}_{\text{regularization}} \right)}$$

Necessary and sufficient optimality (Fermat) :

$$\boxed{\forall j \in [p],\ \mathbf{x}_j^\top \left(\frac{y - X\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}}}{\lambda} \right) \in \begin{cases} \{\text{sign}(\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})_j\} & \text{if} \quad (\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})_j \neq 0, \\ [-1, 1] & \text{if} \quad (\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})_j = 0. \end{cases}}$$

<u>Rem</u>: If $\lambda > \lambda_{\max} := \max_{j\in[\![1,p]\!]} |\langle \mathbf{x}_j, \mathbf{y}\rangle|$, then $\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}} = 0$

Iterative algorithm for Lasso (Sub-gradient descent)

Lasso analysis

Theory : more involved for the Lasso than for least squares / Ridge

Recent reference : Bühlmann and van de Geer (2011)

In a nutshell : add bias to the standard least squares to perform variance reduction

Combining Lasso and Ridge (ℓ_1/ℓ_2 regularization) : Elastic-net

The Elastic-Net, introduced by Zou and Hastie (2005) is the (unique) solution of

$$\hat{\boldsymbol{\theta}}_\lambda = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left[\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\left(\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\frac{\|\boldsymbol{\theta}\|_2^2}{2}\right)\right]$$

<u>Motivation</u> : help selecting all relevant but correlated variable (not only one as for the Lasso)

<u>Rem</u>: two parameters needed, one for global regularization, one trading-off Ridge vs. Lasso

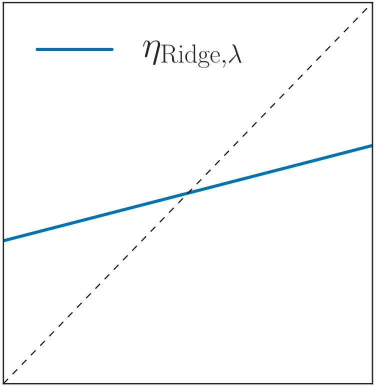
<u>Rem</u>: the solution is unique and the size of the Elastic-Net support is smaller than $\min(n, p)$

Comparing regularizers in 1D : Ridge

Solve :

$$\eta_\lambda(z) = \operatorname*{arg\,min}_{x \in \mathbb{R}} x \mapsto \frac{1}{2}(z - x)^2 + \frac{\lambda}{2}x^2$$

$$\eta_\lambda(z) = \frac{z}{1 + \lambda}$$



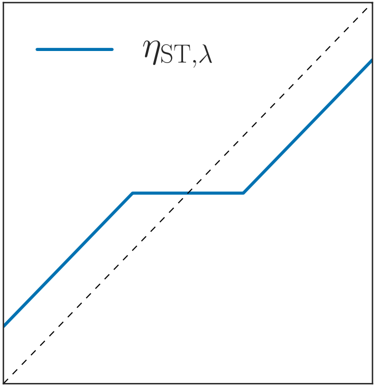
ℓ_2 shrinkage : Ridge

Comparing regularizers in 1D : Lasso

Solve :

$$\eta_\lambda(z) = \operatorname*{arg\,min}_{x \in \mathbb{R}} x \mapsto \frac{1}{2}(z-x)^2 + \lambda|x|$$

$$\eta_\lambda(z) = \mathsf{sign}(z)(|z| - \lambda)_+$$



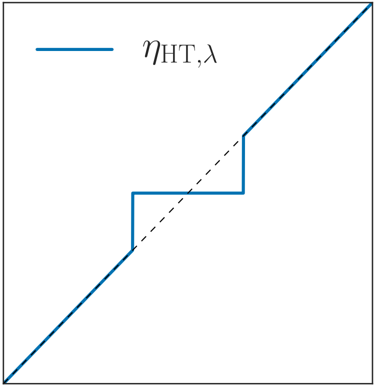
ℓ_1 shrinkage : soft thresholding

Comparing regularizers in 1D : ℓ_0

Solve :

$$\eta_\lambda(z) = \arg\min_{x \in \mathbb{R}} x \mapsto \frac{1}{2}(z-x)^2 + \lambda \mathbb{1}_{x \neq 0}$$

$$\eta_\lambda(z) = z \mathbb{1}_{|z| \geqslant \sqrt{2\lambda}}$$

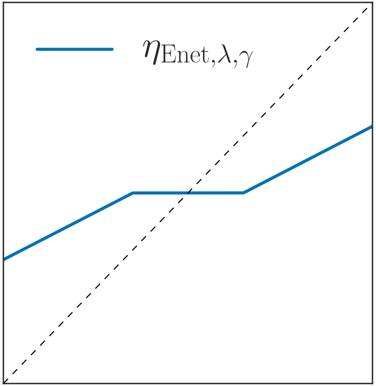


ℓ_0 shrinkage : hard thresholding

Comparing regularizers in 1D : Elastic-Net

Solve :

$$\eta_\lambda(z) = \underset{x\in\mathbb{R}}{\arg\min}\, x \mapsto \frac{1}{2}(z-x)^2+\lambda(\gamma|x|+(1-\gamma)\frac{x^2}{2})$$



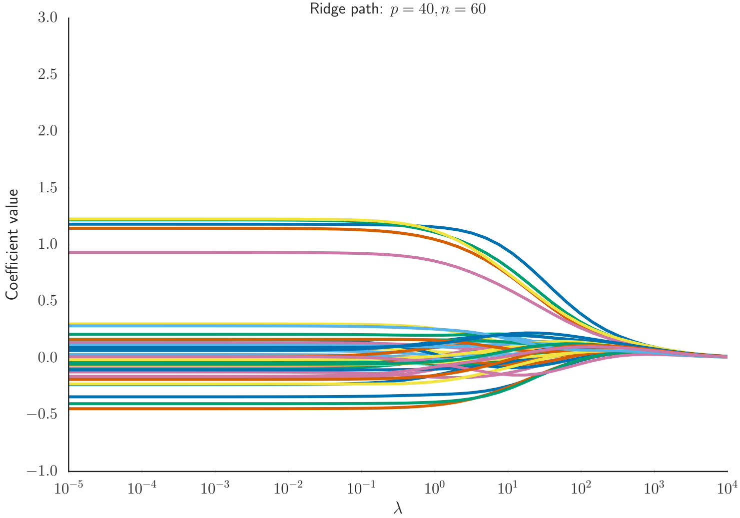
ℓ_1/ℓ_2

Numerical example on simulated data

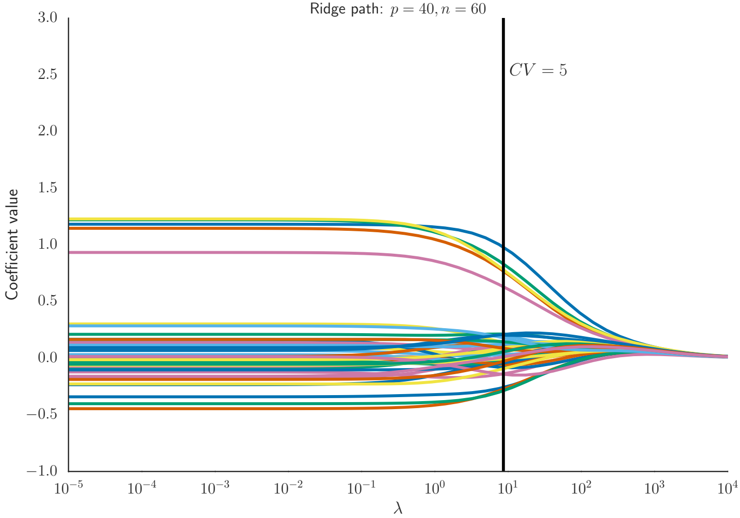
- $\boldsymbol{\theta}^\star = (1, 1, 1, 1, 1, 0, \dots, 0) \in \mathbb{R}^p$ (5 non-zero coefficients)
- $X \in \mathbb{R}^{n \times p}$ has columns drawn according to a Gaussian distribution
- $y = X\boldsymbol{\theta}^\star + \boldsymbol{\varepsilon} \in \mathbb{R}^n$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \operatorname{Id}_n)$
- We use a grid of 50 λ values

For this example : $n = 60, p = 40, \sigma = 1$

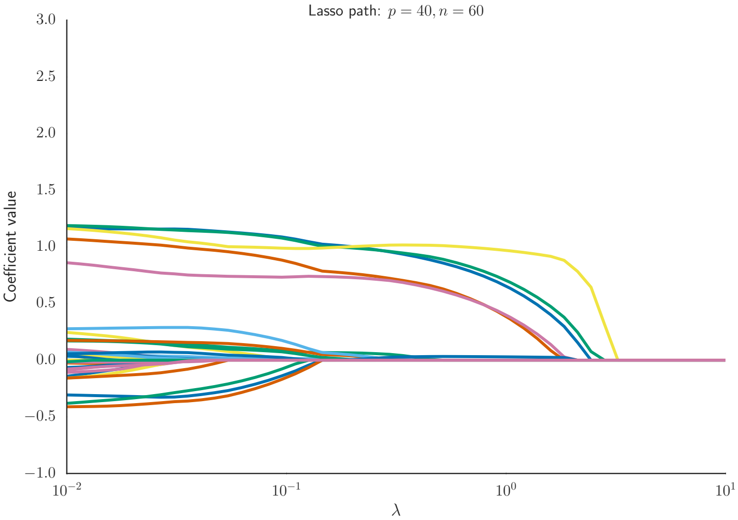
Lasso vs Ridge



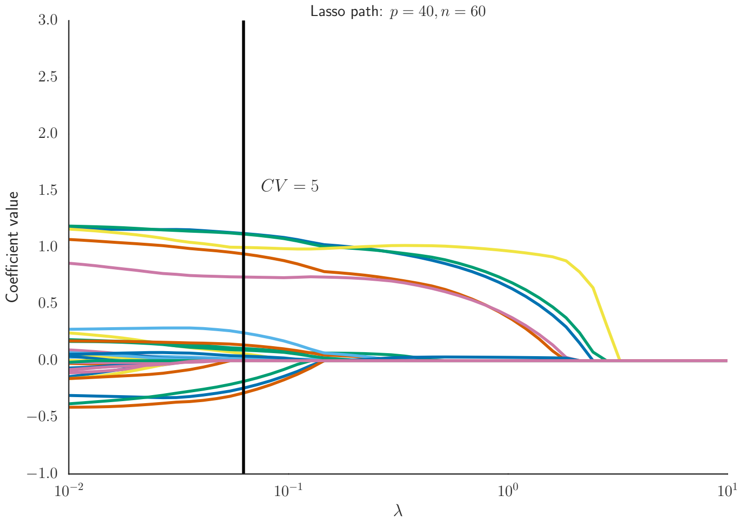
Lasso vs Ridge



Lasso vs Ridge



Lasso vs Ridge



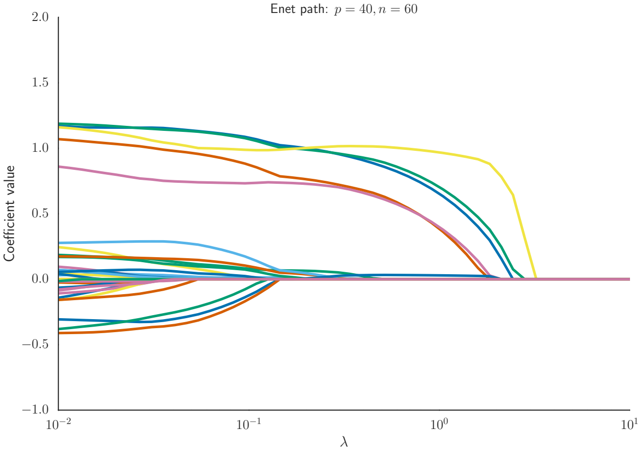
Lasso properties

- Solutions is not necessarily unique
- The analytic form does not necessarily exist
- Numerical aspect : the Lasso is a **convex** problem
- Variable selection / sparse solutions : $\hat{\boldsymbol{\theta}}_{\lambda}^{\mathrm{Lasso}}$ has potentially many zeroed coefficients. The λ parameter controls the sparsity level : if λ is large, solutions are very sparse.

  Example : We got 17 non-zero coefficients for `LassoCV` in the previous simulated example

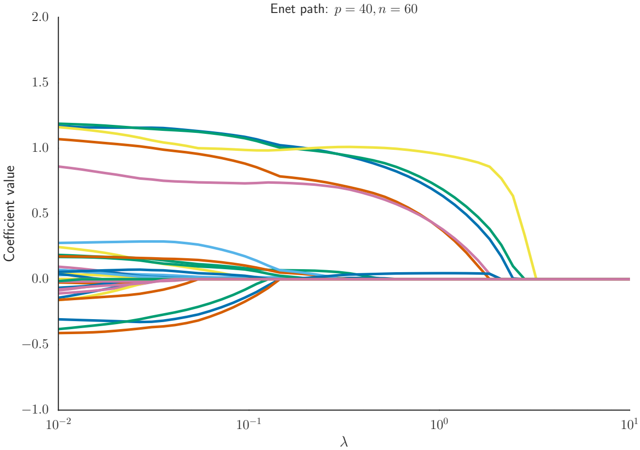
  Rem: RidgeCV has no zero coefficients

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



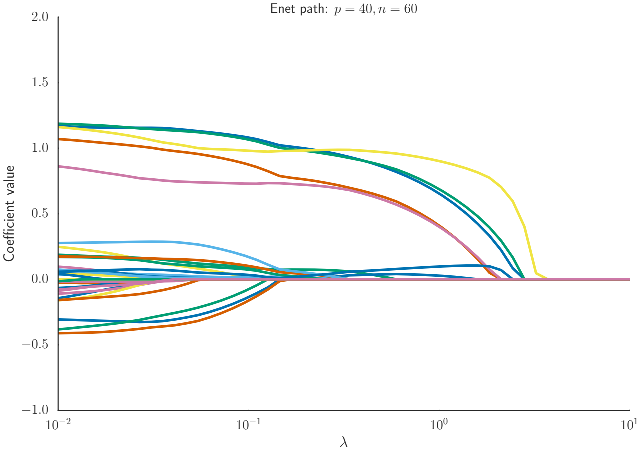
$\gamma = 1.00$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



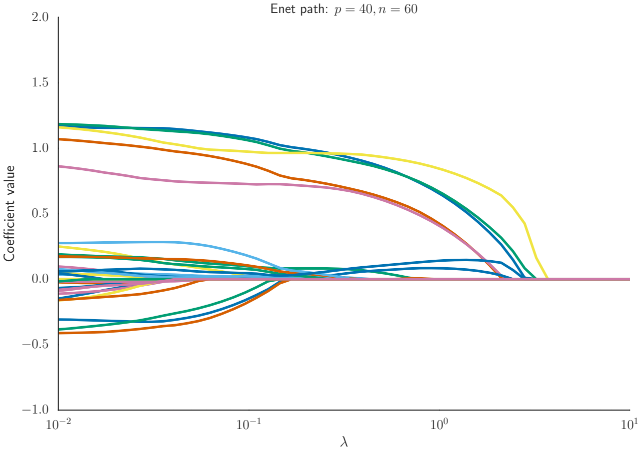
$\gamma = 0.99$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



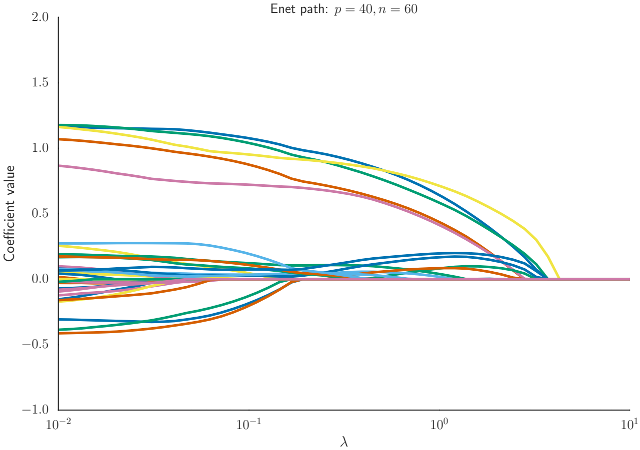
$\gamma = 0.95$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



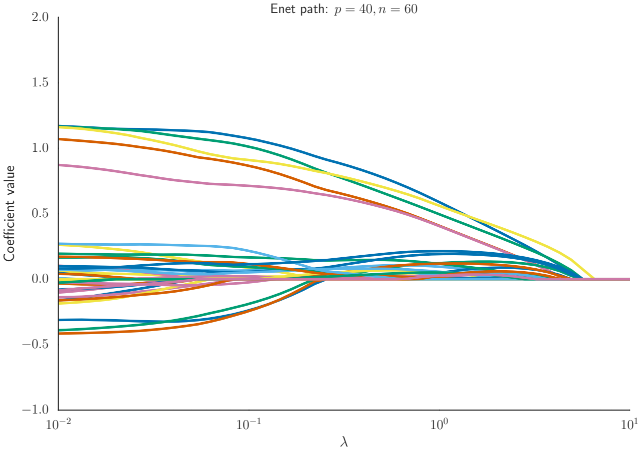
$\gamma = 0.90$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



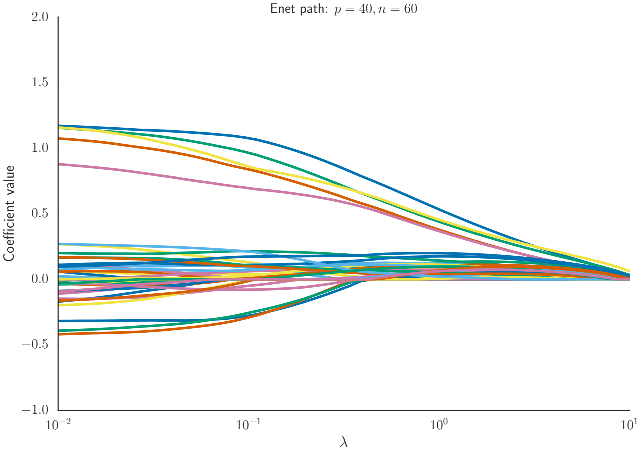
$\gamma = 0.75$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



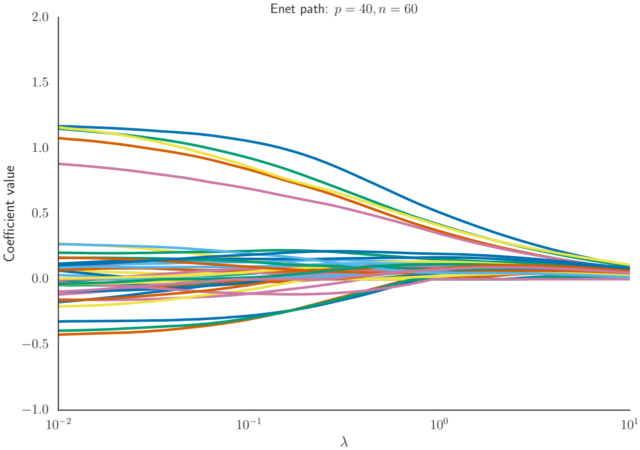
$\gamma = 0.50$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



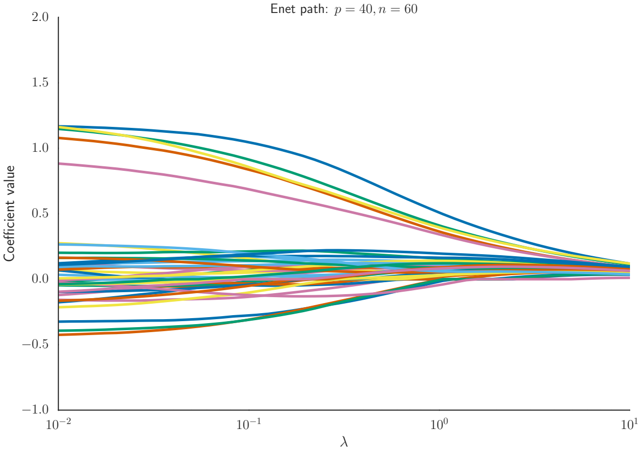
$\gamma = 0.25$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



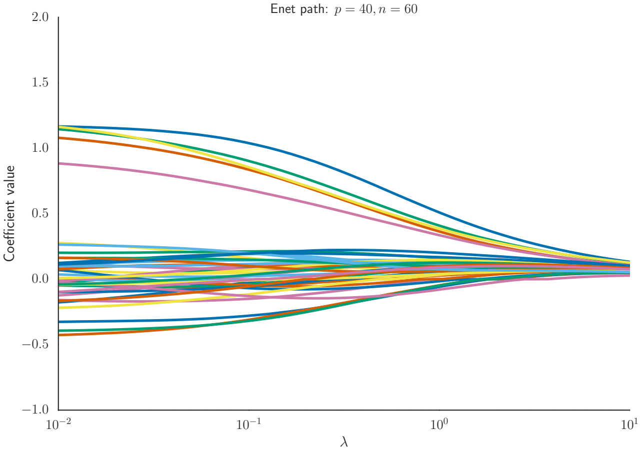
$\gamma = 0.1$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



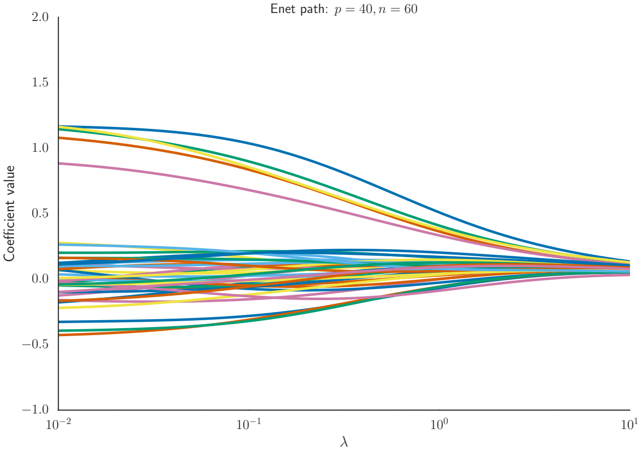
$\gamma = 0.05$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



$\gamma = 0.01$

Elastic-Net : $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



$\gamma = 0.00$

The Lasso bias

The Lasso is biased : it shrinks large coefficients towards 0

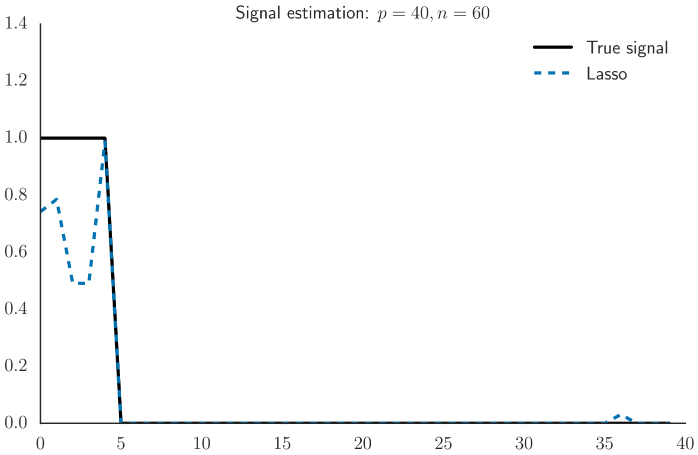


Illustration over the previous example

The Lasso bias

The Lasso is biased : it shrinks large coefficients towards 0

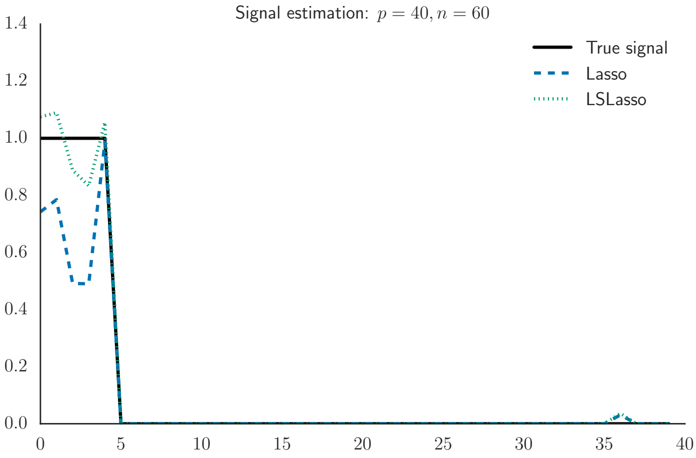


Illustration over the previous example

The Lasso bias : a simple remedy

How to rescale shrunk coefficients ?

LSLasso (Least Square Lasso)

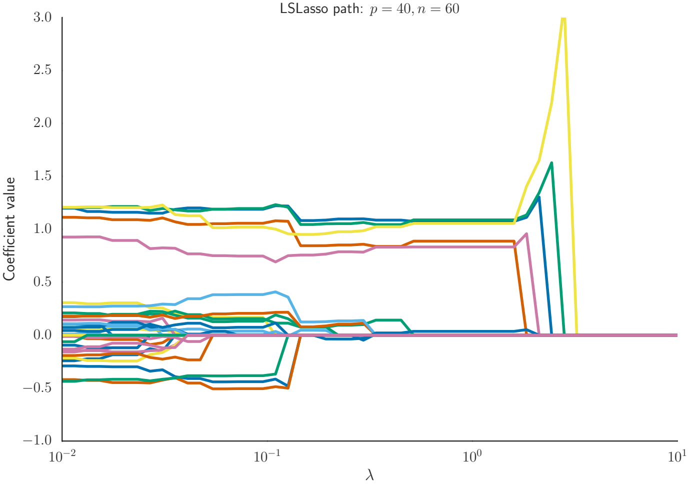
1. Lasso : compute $\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}}$
2. Perform least squares over selected variables : $\operatorname{supp}(\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})$

$$\hat{\boldsymbol{\theta}}_\lambda^{\text{LSLasso}} = \mathop{\arg\min}_{\substack{\boldsymbol{\theta}\in\mathbb{R}^p \\ \operatorname{supp}(\boldsymbol{\theta})=\operatorname{supp}(\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})}} \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$$

<u>Rem</u>: perform CV for the double step procedure ; choosing λ by LassoCV and then performing OLS keeps too many variables

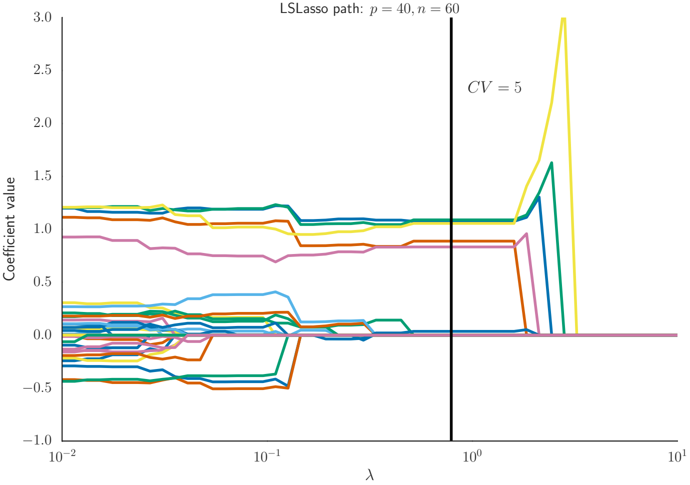
<u>Rem</u>: LSLasso is not coded in standard packages

De-biasing



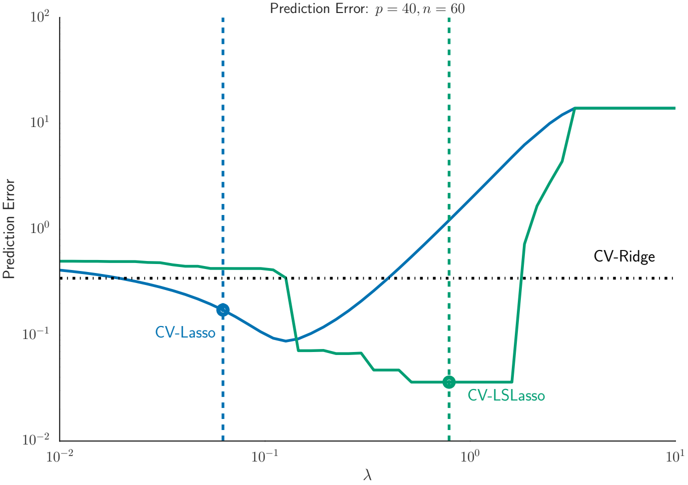
LSLasso path: $p = 40, n = 60$
Coefficient value
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
10^{-2}
10^{-1}
10^{0}
10^{1}
λ

De-biasing



LSLasso path: $p = 40, n = 60$
$CV = 5$
Coefficient value
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
10^{-2}
10^{-1}
10^{0}
10^{1}
λ

Prediction : Lasso vs. LSLasso



LSLasso evaluation

Pros

- the "true" large coefficients are less shrunk
- CV recovers less "parasite" variables (improve interpretability)
  e.g.,in the previous example the LSLassoCV recovers exactly the 5 "true" non zero variables, up to a single false positive

LSLasso : especially useful for <u>estimation</u>

Cons

- the difference in term of prediction is not always striking
- requires (slightly) more computation : needs to compute as many OLS as λ's

Principal components analysis, PCA

What is it ?

- PCA is an unsupervised learning technique : the goal is to find a lower dimensional representation of the data that keeps as much of the variance of the original data. Can be used as a preprocessing for Clustering
- We use it here as a preprocessing for the OLS (aka PCA before OLS, aka PCRegression, ...)

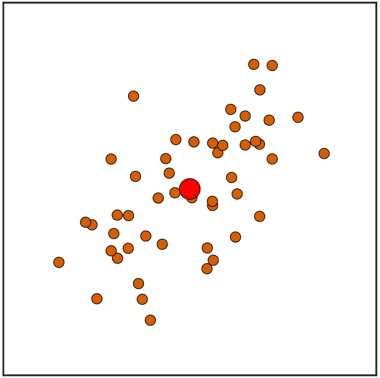
Goal : Reduce the dimensionality while keeping the variance in the data

High level idea : remove

- Super-collinearity
- Close to 0 variance features

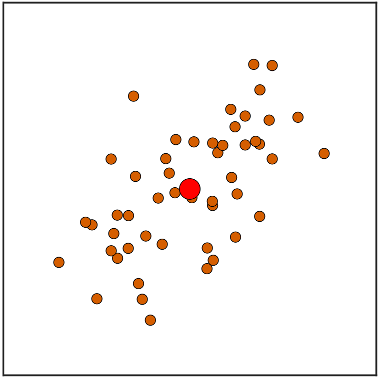
Graphical representation (not to be confused with OLS)

Main axis : variance maximization



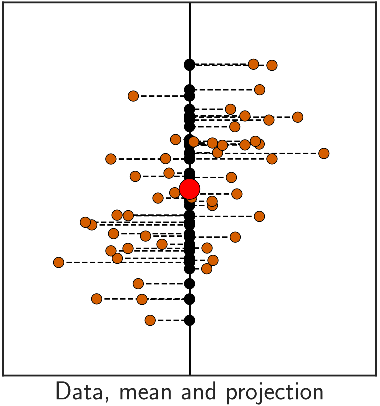
Data and mean

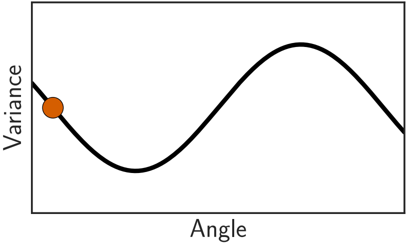
Main axis : variance maximization



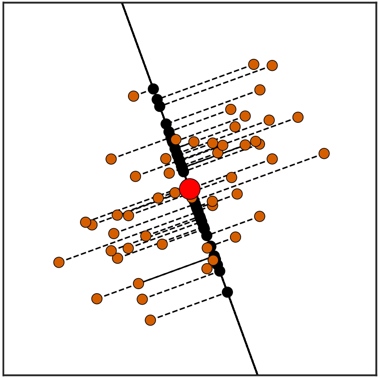
Data and mean

Main axis : variance maximization

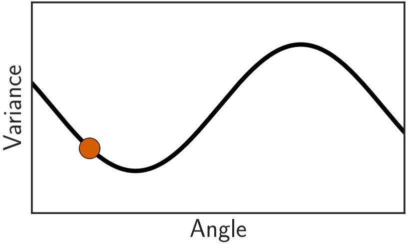




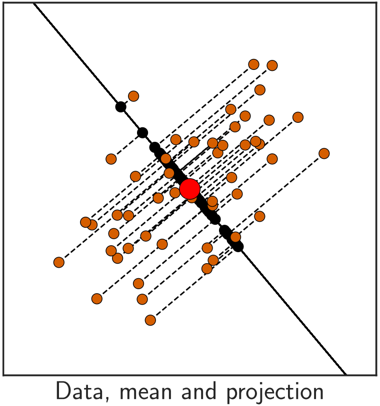
Main axis : variance maximization

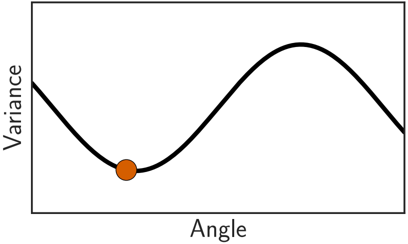


Data, mean and projection

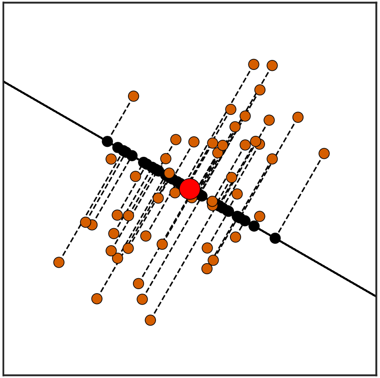


Main axis : variance maximization

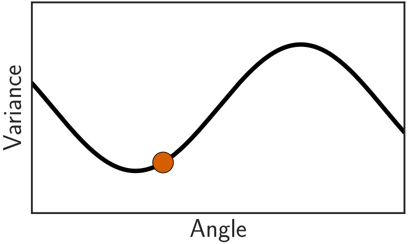




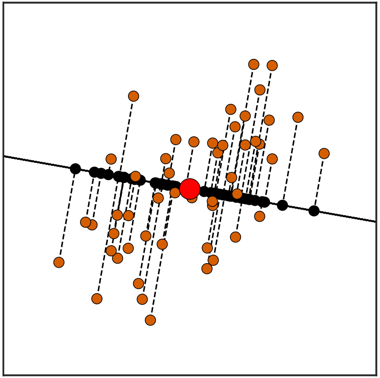
Main axis : variance maximization



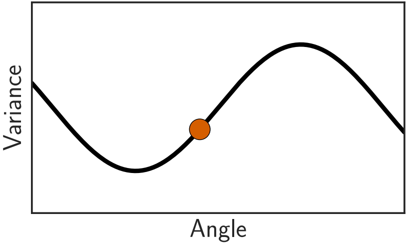
Data, mean and projection



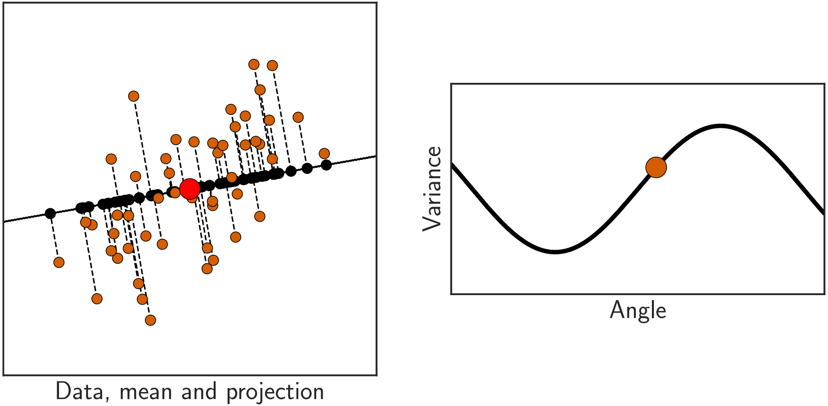
Main axis : variance maximization



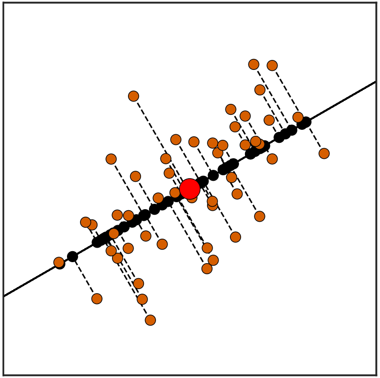
Data, mean and projection



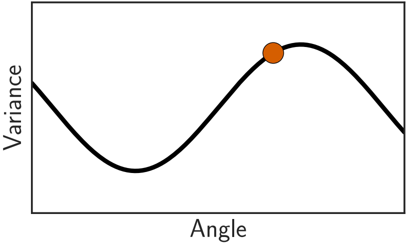
Main axis : variance maximization



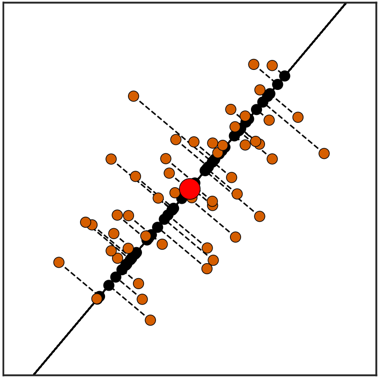
Main axis : variance maximization



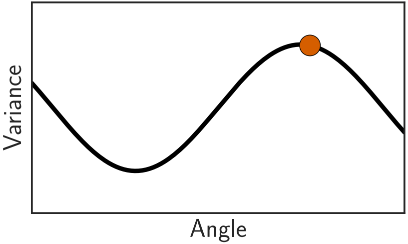
Data, mean and projection



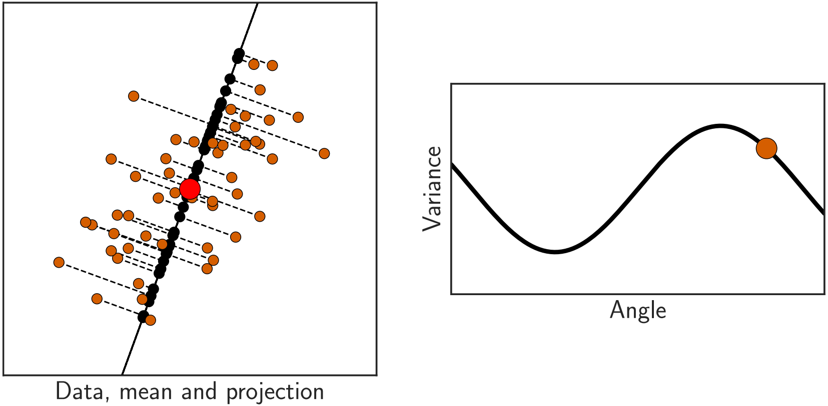
Main axis : variance maximization



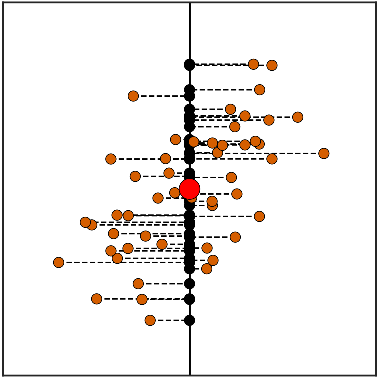
Data, mean and projection

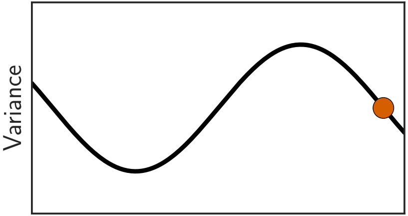


Main axis : variance maximization

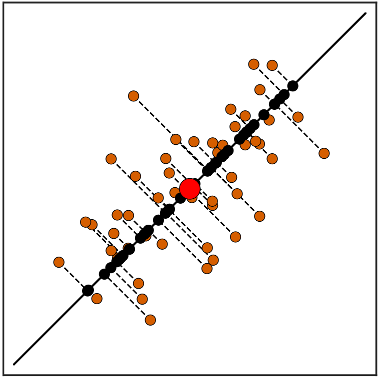


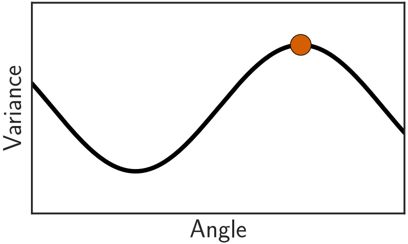
Main axis : variance maximization





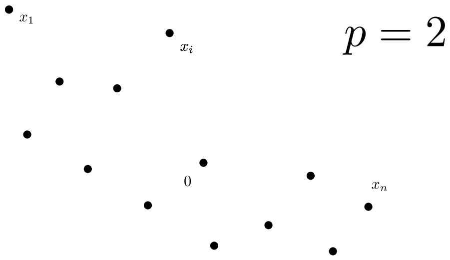
Main axis : variance maximization


Principal direction (main axis)



Variance of the distances along direction v

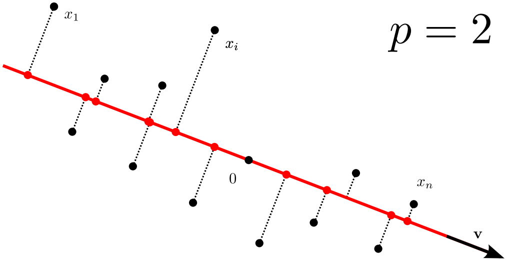
We observe n points $x_1, \ldots, x_n$, i.e., $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times p}$, n observations (rows), p *features* (columns)



Rem: we have to center and scale the dataset : the points have a zero average $X \leftarrow [x_1 - \overline{x}_n, \ldots, x_n - \overline{x}_n]^\top = X - \mathbf{1}_n \overline{x}_n^\top$ and variance 1.
Rem: The distance from x_i to the origin is $x_i^\top v$, and the variances are $\sum_{i=1}^n (x_i^\top v_1)^2$

Variance of the distances along direction v

We observe n points $x_1, \dots, x_n$, i.e., $X = [x_1, \dots, x_n]^\top \in \mathbb{R}^{n \times p}$, n observations (rows), p *features* (columns)

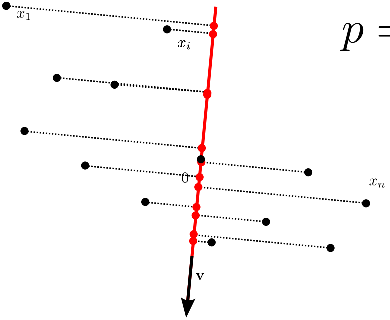


$$p = 2$$

Rem: we have to center and scale the dataset : the points have a zero average $X \leftarrow [x_1 - \overline{x}_n, \dots, x_n - \overline{x}_n]^\top = X - \mathbf{1}_n \overline{x}_n^\top$ and variance 1.
Rem: The distance from x_i to the origin is $x_i^\top v$, and the variances are $\sum_{i=1}^n (x_i^\top v_1)^2$

Variance of the distances along direction v

We observe n points $x_1, \ldots, x_n$, i.e., $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times p}$, n observations (rows), p *features* (columns)



$$p = 2$$

Rem: we have to center and scale the dataset : the points have a zero average $X \leftarrow [x_1 - \overline{x}_n, \ldots, x_n - \overline{x}_n]^\top = X - \mathbf{1}_n \overline{x}_n^\top$ and variance 1.

Rem: The distance from x_i to the origin is $x_i^\top v$, and the variances are $\sum_{i=1}^n (x_i^\top v_1)^2$

Connection between PCA and variance (sketch), first step

Goal : find the direction v_1 that maximizes the variance of the data

- The data is centered and standardized
- Direction $v_1 \in \mathbb{R}^p$ is a linear combination of the original dimensions of X and $\|v\| = 1$
- The distance from the origin to the projection of x_i onto v_1 is $x_i^\top v_1$
- The variance along v_i of the projections is $\sum_{i=1}^n (x_i^\top v_1)^2 = \|Xv_1\|^2 = v_1^\top X^\top X v_1$
- Gram matrix : $G = (n-1)^{-1} X^\top X$, a symmetric covariance matrix
- We rewrite the variance $\sum_{i=1}^n (x_i^\top v_1)^2 \propto v_1^\top G v_1$
- Optimization problem : the direction v_1 that maximizes the variance of the data is

$$v_1 = \underset{v \in \mathbb{R}^p, \|v\|=1}{\arg\max} \sum_{i=1}^n (x_i^\top v)^2 = \underset{v \in \mathbb{R}^p, \|v\|=1}{\arg\max}\ v^\top G v$$

Connection between PCA and variance, first step

By the method of Lagrange multipliers the solution of $\mathbf{v_1} = \arg\max_{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1} v^\top G v$ is

$$G\mathbf{v_1} = \lambda_1 \mathbf{v_1}$$

- $\lambda_1, \mathbf{v_1}$ are the eigenvalue/vector
- λ_1 is also the variance
- v_1 is the eigenvector associated to the largest eigenvalue

To summarize, we have found that if we wish to find a 1-dimensional subspace with with to approximate the data, we should choose v to be the principal eigenvector of G.

Then, to represent $x^{(i)}$ in this basis, we need only compute the corresponding scalar :

$$v_1^T x^{(i)} \in \mathbb{R}.$$

Further components

In the following "iterations", find $\mathbf{v}_2$, a direction $\perp \mathbf{v}_1$ that maximizes the variance.

Let λ_i, v_i the i-th largest eigenvalue and its associated eigenvector. Then $\mathbf{v}_i \perp \mathbf{v}_{i-1}$ for $i > 1$ (since G is symmetric p.s.d.) and maximizes the variance

If we wish to project our data into a k-dimensional subspace ($k < d$), we should choose $\mathbf{v}_1, \ldots, \mathbf{v}_k$ to be the top k eigenvectors of G. The $\mathbf{v}_i$'s now form a new, orthogonal basis for the data.

Then, to represent $x^{(i)}$ in this basis, we need only compute the corresponding vector

$$\begin{bmatrix} \mathbf{v}_1^T x^{(i)} \\ \mathbf{v}_2^T x^{(i)} \\ \vdots \\ \mathbf{v}_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

Lower dimensional representation of X

- The axes (of direction) $\mathbf{v}_1, \dots, \mathbf{v}_p \in \mathbb{R}^p$ are called **principal components**
- The new variables $\mathbf{c}_j = X\mathbf{v}_j, j = 1, \dots, p$ are called scores

New representation (order k) :

- The matrix XV_k (with $V_k = [\mathbf{v}_1, \dots, \mathbf{v}_k]$) is the matrix representing the data in the base of the first k eigenvectors

Reconstruction in the original space (debruiter) :

- "Perfect" reconstruction for $\mathbf{x} \in \mathbb{R}^p$: $\mathbf{x} = \sum_{j=1}^{p} (\mathbf{x}^\top \mathbf{v}_j)\mathbf{v}_j$
- Reconstruction with loss of information : $\hat{\mathbf{x}} = \sum_{j=1}^{k} (\mathbf{x}^\top \mathbf{v}_j)\mathbf{v}_j$

PCA before OLS

Algorithme : PCA before OLS

Entrées : $X \in \mathbb{R}^{n \times p}$, itérations K

$V_k \leftarrow k$-th eigenvectors assoc to the k largest eigenvalues

$Z = XV_k$ is the new (projected) dataset

OLS in Z

When does it work ?

For practical reasons, we usually prefer to use the SVD of X than the eigen-decomposition of $X^T X$

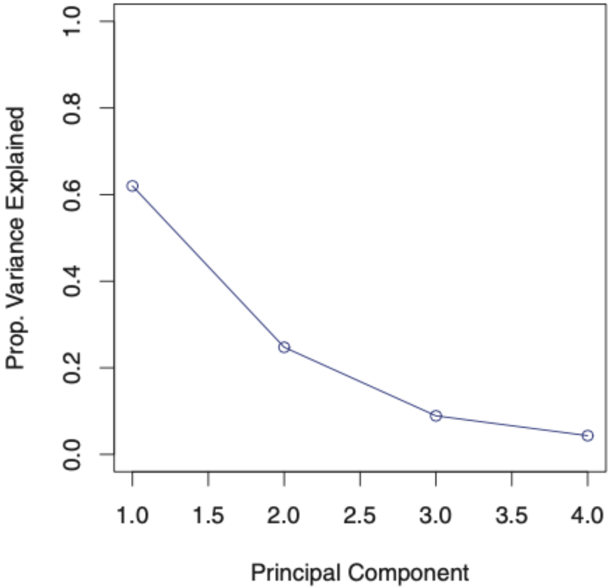
Exercise: Show that the i-th singular value of X, σ_i, and the i-th eigenvalue of $X^\top X$, λ_i, are related as follows $\lambda_i = (n-1)^{-1}\sigma_i^2$

Understanding the projection/direction, dataset USArrests

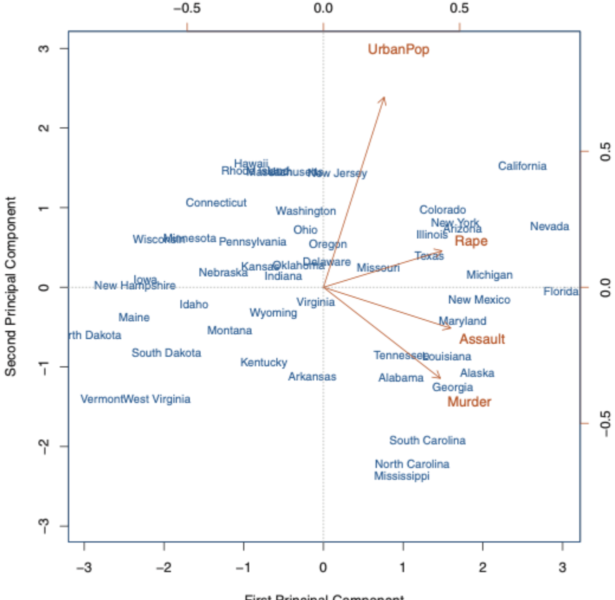
		Murder	Assault	UrbanPop	Rape
0	Alabama	13.2	236	58	21.2
1	Alaska	10.0	263	48	44.5
2	Arizona	8.1	294	80	31.0
3	Arkansas	8.8	190	50	19.5
4	California	9.0	276	91	40.6

. . .

Percentage of variance explained



Principal components



Conclusions

- PCA is an unsupervised technique
- Dimensionality reduction (more than a feature subset selection method)
- When the target **y** is correlated with the variance directions then its useful
- Interpretation of the proportion of variance explained
- Projection to low dimensions
- No interpretability on lower dimensions