Data Science for Actuaries (ACT6100)

Arthur Charpentier

Supervisé # 2 (Régularisation - Pénalisation - OLS)

automne 2Q20

https://github.com/freakonometrics/ACT6100/

Pénalisation et Lagrangien

En optimisation, le problème d'optimisation sous contrainte

$$\min_{\boldsymbol{x}\in\mathbb{R}^k}\{f(\boldsymbol{x})\}$$
$$\text{sous contrainte } \boldsymbol{x}\in\mathcal{E}$$

peut s'écrire

$$\min_{\boldsymbol{x}\in\mathbb{R}^k}\{f(\boldsymbol{x})+\lambda p(\boldsymbol{x})\}$$

où $\lambda > 0$ est le facteur de pénalisation, et $p(\cdot)$ est une fonction.
En choisissant

$$p(\boldsymbol{x})=\begin{cases}0 \text{ si } \boldsymbol{x}\in\mathcal{E}\\ +\infty \text{ si } \boldsymbol{x}\notin\mathcal{E}\end{cases}$$

Les problèmes sont équivalents.
On dire que p est une fonction de pénalisation exacte si les deux problèmes sont équivalents (toute 'solution' de l'un est solution de l'autre)

Pénalisation et Lagrangien

Classiquement, on cherchera des fonctions de pénalisation continue sur $\mathbb{R}^k$, positives, et telles que $p(\boldsymbol{x}) = 0$ si et seulement si $\boldsymbol{x} \in \mathcal{E}$.

Example si $\mathcal{E} = \mathbb{R}_+ = \{x : x \geq 0\}$, on peut prendre $p(x) = \|x_-\|^2$ (pénalisation quadratique)

Example si $\mathcal{E} = \{x : c(x) \leq 0\}$, on peut prendre $p(x) = \|c(x)_+\|^2$

Example si $\mathcal{E} = \mathbb{R}_+^k = \{\boldsymbol{x} : \boldsymbol{x} \geq \boldsymbol{0}\}$, on peut prendre

$$p(\boldsymbol{x}) = -\sum_{i=1}^{k} \log(x_i) \quad \text{(proposé par Ragnar Frisch, 1955)}$$

Condition de Karush-Kuhn-Tucker

Considérons les problèmes

$$\begin{array}{l}\min_{\boldsymbol{x}\in\mathbb{R}^k}\{f(\boldsymbol{x})\}\\ \text{sous contrainte } g(\boldsymbol{x})=\boldsymbol{0}\end{array} \quad \text{ou} \quad \begin{array}{l}\min_{\boldsymbol{x}\in\mathbb{R}^k}\{f(\boldsymbol{x})\}\\ \text{sous contrainte } g(\boldsymbol{x})\leq\boldsymbol{0}\end{array}$$

La condition de Karush-Kuhn-Tucker est

$$\begin{cases}\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^\star,\boldsymbol{z}^\star)=\boldsymbol{0}\\ \nabla_{\boldsymbol{z}}\mathcal{L}(\boldsymbol{x}^\star,\boldsymbol{z}^\star)=\boldsymbol{0}\end{cases}$$

où

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{z})=f(\boldsymbol{x})+\boldsymbol{z}^\top g(\boldsymbol{x})$$

est le Lagrangien du problème (les paramètres $\boldsymbol{z}$ sont les multiplicateurs)

Si on a des problèmes convexes et différentiables, si $\mathcal{L}(\boldsymbol{x},\boldsymbol{z})$ admet pour minimum global $\boldsymbol{x}^\star$ alors $\boldsymbol{x}^\star$ est solution du problème d'optimisation contraint.

Controlling smoothness with penalization

We want to find $m:\mathbb{R}\to\mathbb{R}$ solution of

$$\sum_{i=1}^{n}\big(y_i-m(x_i)\big)^2+\lambda\int_{\mathbb{R}}m''(u)^2du$$

where the second term penalizes curvature (linear model $=0$)

Proposition Out of all twice-differentiable functions passing through the points (x_i,y_i) the one that minimizes

$$\lambda\int_{\mathbb{R}}m''(u)^2du=\lambda\|m''\|^2$$

is a natural* cubic spline with knots at every unique value of x_i's.

Proposition Out of all twice-differentiable functions, the one that minimizes

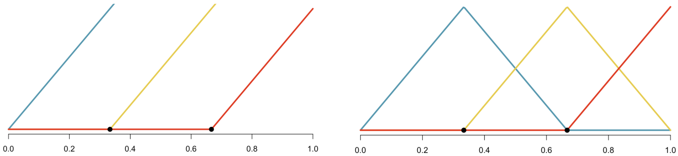
$$\sum_{i=1}^{n}\big(y_i-m(x_i)\big)^2+\lambda\int_{\mathbb{R}}m''(u)^2du$$

is a natural cubic spline with knots at every unique value of x_i's.

Controlling smoothness with penalization

Linear splines (piecewise linear continuous models) are

$$L_1(x) = 1,\ L_2(x) = x,\ L_3(x) = (x - k_1)_+,\ L_4(x) = (x - k_2)_+,\ \dots$$

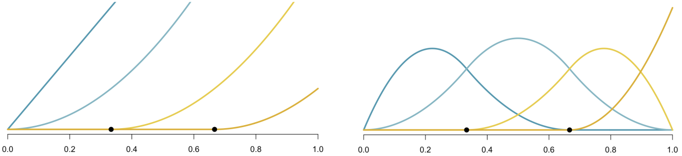



```
> x = sort(runif(n))
> X = bs(x,knots=quantile(x,p=c(1/3,2/3)),degree = 1)
attr(,"degree")
[1] 1
attr(,"knots")
33.33333% 66.66667%
0.3542930 0.7091861
attr(,"Boundary.knots")
[1] 0.003697588 0.989722282
```

Controlling smoothness with penalization

Quadratic splines (piecewise linear continuous models) are

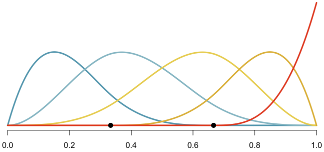
$$L_1(x) = 1,\ L_2(x) = x,\ L_3(x) = x^2,\ L_4(x) = (x - k_1)_+^2,\ \dots$$

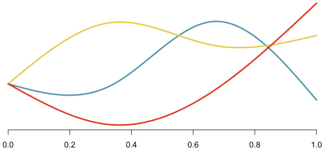



```
> x = sort(runif(n))
> X = bs(x,knots=quantile(x,p=c(1/3,2/3)),degree = 2)
attr(,"degree")
[1] 2
attr(,"knots")
33.33333% 66.66667%
0.3542930 0.7091861
attr(,"Boundary.knots")
[1] 0.003697588 0.989722282
```

Controlling smoothness with penalization

Cubic splines, vs. Natural Splines



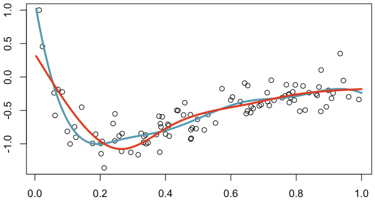


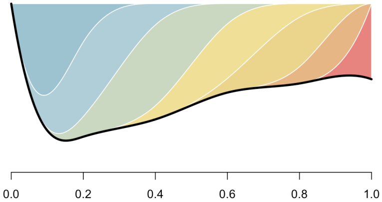

```
> Xb = bs(x,knots=quantile(x,p=c(1/3,2/3)),degree = 3)
> Xn = ns(x,knots=quantile(x,p=c(1/3,2/3)),degree = 3)
```

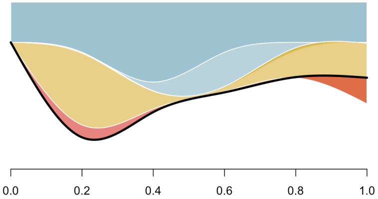
Polynomial models tend to be volatile at the boundaries
So are cubic splines
Natural cubic splines adding constraints that the function is linear beyond the boundaries of the data

Controlling smoothness with penalization

```
> set.seed(1)
> x = sort(runif(100))
> y = sin(log(x))+rnorm(100)/5
> plot(x,y)
> base = data.frame(x,y)
> q = quantile(x,p=c
   (1/5,2/5,3/5,4/5))
> regb = lm(y~bs(x,knots=q),
   data=base)
> regn = lm(y~ns(x,knots=q),
   data=base)
```







Controlling smoothness with penalization

Heuristically, let $(N_j(x))$ denote the natural cubic spline basis with knot x_j.

$m(x) = \sum_{j=1}^{n} \gamma_j N_j(x)$, or $m(\boldsymbol{x}) = \boldsymbol{N}\gamma$, and the penalized objective is

$$(\boldsymbol{y} - \boldsymbol{N}\gamma)^\top(\boldsymbol{y} - \boldsymbol{N}\gamma) + \lambda\gamma^\top\boldsymbol{\Omega}\gamma$$

where $\boldsymbol{\Omega}_{ij} = \int_{\mathbb{R}} N_i''(u) N_j''(u) du$

And the solution is $\widehat{\gamma} = (\boldsymbol{N}^\top\boldsymbol{N} + \lambda\boldsymbol{\Omega})^{-1}\boldsymbol{N}^\top\boldsymbol{y}$

Penalized Inference and Shrinkage

Consider a parametric model, with true (unknown) parameter θ, then

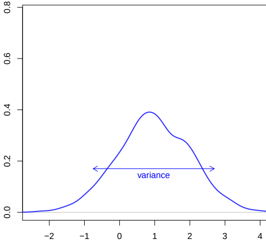
$$\text{mse}(\hat\theta) = \mathbb{E}\left[(\hat\theta-\theta)^2\right] = \underbrace{\mathbb{E}\left[(\hat\theta-\mathbb{E}[\hat\theta])^2\right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[(\mathbb{E}[\hat\theta]-\theta)^2\right]}_{\text{bias}^2}$$

One can think of a shrinkage of an unbiased estimator,

Let $\widetilde{\theta}$ denote an unbiased estimator of θ. Then

$$\hat\theta = \frac{\theta^2}{\theta^2+\text{mse}(\widetilde{\theta})}\cdot\widetilde{\theta}$$

satisfies $\text{mse}(\hat\theta)\leq\text{mse}(\widetilde{\theta})$.



Penalized Inference and Shrinkage

Consider a parametric model, with true (unknown) parameter θ, then

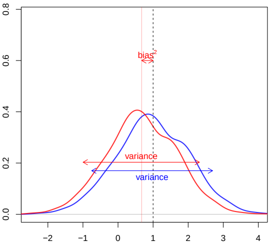
$$\text{mse}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta}-\theta)^2\right] = \underbrace{\mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[(\mathbb{E}[\hat{\theta}]-\theta)^2\right]}_{\text{bias}^2}$$

One can think of a shrinkage of an unbiased estimator,

Let $\widetilde{\theta}$ denote an unbiased estimator of θ. Then

$$\hat{\theta} = \frac{\theta^2}{\theta^2 + \text{mse}(\widetilde{\theta})} \cdot \widetilde{\theta}$$

satisfies $\text{mse}(\hat{\theta}) \leq \text{mse}(\widetilde{\theta})$.



Penalized Inference and Shrinkage

Consider a parametric model, with true (unknown) parameter θ, then

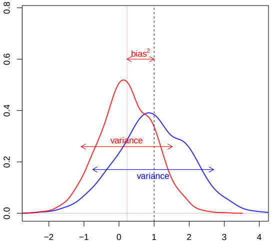
$$\text{mse}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta}-\theta)^2\right] = \underbrace{\mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[(\mathbb{E}[\hat{\theta}]-\theta)^2\right]}_{\text{bias}^2}$$

One can think of a shrinkage of an unbiased estimator,

Let $\widetilde{\theta}$ denote an unbiased estimator of θ. Then

$$\hat{\theta} = \frac{\theta^2}{\theta^2 + \text{mse}(\widetilde{\theta})} \cdot \widetilde{\theta}$$

satisfies $\text{mse}(\hat{\theta}) \leq \text{mse}(\widetilde{\theta})$.



Penalized Inference and Shrinkage

Consider a parametric model, with true (unknown) parameter θ, then

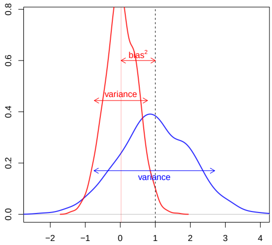
$$\text{mse}(\hat{\theta}) = \mathbb{E}\left[(\hat{\theta}-\theta)^2\right] = \underbrace{\mathbb{E}\left[(\hat{\theta}-\mathbb{E}[\hat{\theta}])^2\right]}_{\text{variance}} + \underbrace{\mathbb{E}\left[(\mathbb{E}[\hat{\theta}]-\theta)^2\right]}_{\text{bias}^2}$$

One can think of a shrinkage of an unbiased estimator,

Let $\widetilde{\theta}$ denote an unbiased estimator of θ. Then

$$\hat{\theta} = \frac{\theta^2}{\theta^2 + \text{mse}(\widetilde{\theta})} \cdot \widetilde{\theta}$$

satisfies $\text{mse}(\hat{\theta}) \leq \text{mse}(\widetilde{\theta})$.



Linear Regression Shortcoming

Least Squares Estimator $\widehat{\beta} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$
Unbiased Estimator $\mathbb{E}[\widehat{\beta}] = \boldsymbol{\beta}$, with variance $\text{Var}[\widehat{\beta}] = \sigma^2(\boldsymbol{X}^\top\boldsymbol{X})^{-1}$
which can be (extremely) large when $\det[(\boldsymbol{X}^\top\boldsymbol{X})] \sim 0$.

$$\boldsymbol{X} = \begin{bmatrix} 1 & -1 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & -1 \\ 1 & 1 & 0 \end{bmatrix} \quad \boldsymbol{X}^\top\boldsymbol{X} = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 6 & -4 \\ 2 & -4 & 6 \end{bmatrix} \quad \boldsymbol{X}^\top\boldsymbol{X}+\mathbb{I} = \begin{bmatrix} 5 & 2 & 2 \\ 2 & 7 & -4 \\ 2 & -4 & 7 \end{bmatrix}$$

eigenvalues : $\{10, 6, 0\}$ $\{11, 7, 1\}$

More generally, eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I} = \{10+\lambda, 6+\lambda, \lambda\}$

Ad-hoc strategy: use $\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I}$, for some $\lambda \geq 0$.

Ridge Regression

One could consider

$$\widehat{\beta}_\lambda^{\text{ridge}} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

which can be also seen as the solution of

$$\widehat{\beta}_\lambda^{\text{ridge}} = \text{argmin}\left\{\sum_{i=1}^n (y_i - \boldsymbol{x}_i^\top\beta)^2 + \lambda\sum_{j=1}^p \beta_j^2\right\}$$

$$\widehat{\beta}_\lambda^{\text{ridge}} = \text{argmin}\left\{\underbrace{\|\boldsymbol{y} - \boldsymbol{X}\beta\|_{\ell_2}^2}_{=\text{criteria}} + \underbrace{\lambda\|\beta\|_{\ell_2}^2}_{=\text{penalty}}\right\}$$

$\lambda \geq 0$ is a tuning parameter.

Ridge Regression

In an OLS context, we want to solve

Ridge Estimator (OLS)

$$\widehat{\beta}_\lambda^{\text{ridge}} = \underset{\beta\in\mathbb{R}^p}{\text{argmin}} \left\{ \frac{1}{2}\sum_{i=1}^n (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

or more generally (when maximizing the log-likelihood)

Ridge Estimator (GLM)

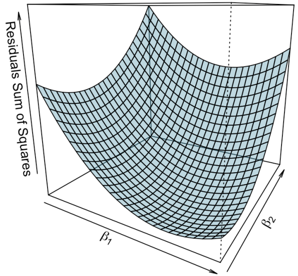
$$\widehat{\beta}_\lambda^{\text{ridge}} = \underset{\beta\in\mathbb{R}^p}{\text{argmin}} \left\{ -\sum_{i=1}^n \log f(y_i|\mu_i = g^{-1}(\boldsymbol{x}_i^\top \boldsymbol{\beta})) + \lambda \sum_{j=1}^p \beta_j^2 \right\}$$

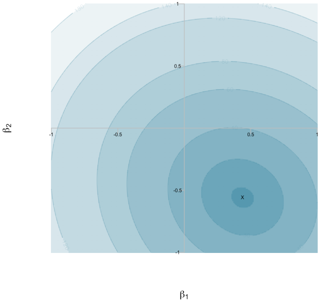
see an Wieringen (2018) for (much) more results

Ridge Regression

To make sense, we should standadize variables $\boldsymbol{x}$ (and y)

```
> chicago=read.table("http://
    freakonometrics.free.fr/chicago
    .txt",header=TRUE,sep=";")
> standardize <-  function(x)  {(x-
    mean(x))/sd(x)}
> y = standardize(chicago[,"Fire"])
> x1 =standardize(chicago[,"X_2"])
> x2 =standardize(chicago[,"X_2"])
> RSS = function(beta){
+ sum((y-beta[1]*x1-beta[2]*x2)^2)
+ }
>summary(lm(y~x1+x2-1)

Coefficients:
      x1         x2
 0.4386   -0.5576
```





Ridge Regression

$$\mathcal{L}_\lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

$$\frac{\partial \mathcal{L}_\lambda(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^\top \boldsymbol{y} + 2(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbb{I})\boldsymbol{\beta}$$

$$\frac{\partial^2 \mathcal{L}_\lambda(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\top} = 2(\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbb{I})$$

where $\boldsymbol{X}^\top \boldsymbol{X}$ is a semi-positive definite matrix, and $\lambda \mathbb{I}$ is a positive definite matrix, and

$$\widehat{\boldsymbol{\beta}}_\lambda^{\text{ridge}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \mathbb{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

Ridge Regression

$$\widehat{\beta}_\lambda^{\text{ridge}} = \text{argmin}\left\{\|\boldsymbol{y} - (\beta_0 + \boldsymbol{X}\beta)\|_{\ell_2}^2 + \lambda\|\boldsymbol{\beta}\|_{\ell_2}^2\right\}$$

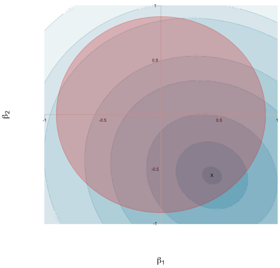
can be seen as a constrained optimization problem

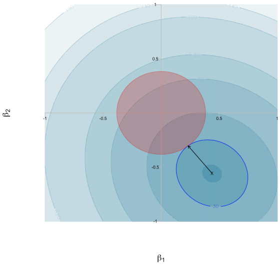
$$\widehat{\beta}_\lambda^{\text{ridge}} = \underset{\|\boldsymbol{\beta}\|_{\ell_2}^2 \le h_\lambda}{\text{argmin}}\left\{\|\boldsymbol{y} - (\beta_0 + \boldsymbol{X}\beta)\|_{\ell_2}^2\right\}$$

Explicit solution

$$\widehat{\beta}_\lambda^{\text{ridge}} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

If $\lambda \to 0$, $\widehat{\beta}_0^{\text{ridge}} = \widehat{\beta}^{\text{ols}}$
If $\lambda \to \infty$, $\widehat{\beta}_\infty^{\text{ridge}} = \boldsymbol{0}$.





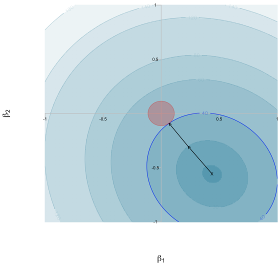
Ridge Regression

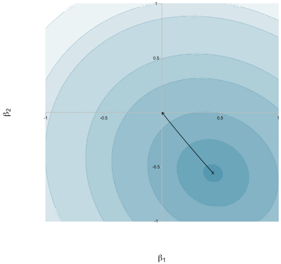
This penalty can be seen as rather unfair if components of $\boldsymbol{x}$ are not expressed on the same scale

- center: $\overline{\boldsymbol{x}}_j = 0$, then $\widehat{\beta}_0 = \overline{\boldsymbol{y}}$
- scale: $\boldsymbol{x}_j^\top \boldsymbol{x}_j = 1$

Then compute

$$\widehat{\beta}_\lambda^{\text{ridge}} = \text{argmin}\left\{\underbrace{\|\boldsymbol{y}-\boldsymbol{X}\beta\|_{\ell_2}^2}_{=\text{loss}} + \underbrace{\lambda\|\boldsymbol{\beta}\|_{\ell_2}^2}_{=\text{penalty}}\right\}$$



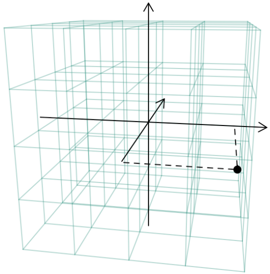


Ridge Regression

Observe that if $\boldsymbol{x}_{j_1} \perp \boldsymbol{x}_{j_2}$, then

$$\widehat{\beta}_{\lambda}^{\text{ridge}} = [1+\lambda]^{-1}\widehat{\beta}_{\lambda}^{\text{ols}}$$

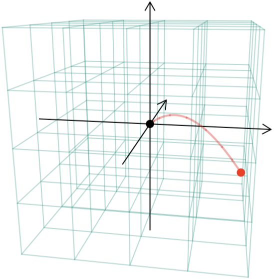
which explain relationship with shrinkage.
But generally, it is not the case...



Smaller mse

There exists λ such that

$$\text{mse}[\widehat{\beta}_{\lambda}^{\text{ridge}}] \leq \text{mse}[\widehat{\beta}_{\lambda}^{\text{ols}}]$$



The Bayesian Interpretation

From a Bayesian perspective,

$$\underbrace{\mathbb{P}[\boldsymbol{\theta}|\boldsymbol{y}]}_{\text{posterior}} \propto \underbrace{\mathbb{P}[\boldsymbol{y}|\boldsymbol{\theta}]}_{\text{likelihood}} \cdot \underbrace{\mathbb{P}[\boldsymbol{\theta}]}_{\text{prior}} \quad \text{i.e.} \quad \log \mathbb{P}[\boldsymbol{\theta}|\boldsymbol{y}] = \underbrace{\log \mathbb{P}[\boldsymbol{y}|\boldsymbol{\theta}]}_{\text{log likelihood}} + \underbrace{\log \mathbb{P}[\boldsymbol{\theta}]}_{\text{penalty}}$$

If β has a prior $\mathcal{N}(\mathbf{0}, \tau^2\mathbb{I})$ distribution, then its posterior distribution has mean

$$\mathbb{E}[\beta|\boldsymbol{y}, \boldsymbol{X}] = \left(\boldsymbol{X}^\top \boldsymbol{X} + \frac{\sigma^2}{\tau^2}\mathbb{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}.$$

Properties of the Ridge Estimator

$$\widehat{\beta}_\lambda^{\text{ridge}} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

$$\mathbb{E}[\widehat{\beta}_\lambda^{\text{ridge}}] = \boldsymbol{X}^\top\boldsymbol{X}(\lambda\mathbb{I} + \boldsymbol{X}^\top\boldsymbol{X})^{-1}\beta \neq \beta$$

Set $\boldsymbol{W}_\lambda = (\mathbb{I} + \lambda[\boldsymbol{X}^\top\boldsymbol{X}]^{-1})^{-1}$. One can prove that

$$\text{Var}[\widehat{\beta}_\lambda] = \boldsymbol{W}_\lambda \text{Var}[\widehat{\beta}^{\text{ols}}]\boldsymbol{W}_\lambda^\top$$

and

$$\text{Var}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\top\boldsymbol{X}[(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}]^\top.$$

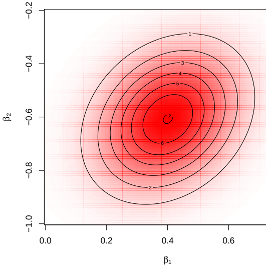
Observe that

$$\text{Var}[\widehat{\beta}^{\text{ols}}] - \text{Var}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2\boldsymbol{W}_\lambda[2\lambda(\boldsymbol{X}^\top\boldsymbol{X})^{-2} + \lambda^2(\boldsymbol{X}^\top\boldsymbol{X})^{-3}]\boldsymbol{W}_\lambda^\top \geq \boldsymbol{0}.$$

Properties of the Ridge Estimator

Hence, the confidence ellipsoid of ridge estimator is indeed smaller than the OLS,
If $\boldsymbol{X}$ is an orthogonal design matrix,

$$\text{Var}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2(1+\lambda)^{-2}\mathbb{I}.$$

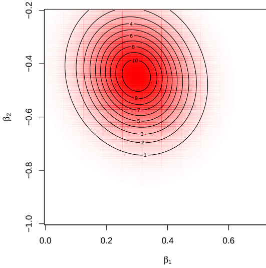


$$\text{mse}[\widehat{\beta}_\lambda] = \sigma^2\text{trace}(\boldsymbol{W}_\lambda(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{W}_\lambda^\top) + \beta^\top(\boldsymbol{W}_\lambda - \mathbb{I})^\top(\boldsymbol{W}_\lambda - \mathbb{I})\beta.$$

If $\boldsymbol{X}$ is an orthogonal design matrix,

$$\text{mse}[\widehat{\beta}_\lambda^{\text{ridge}}] = \frac{p\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2}{(1+\lambda)^2}\beta^\top\beta$$

Properties of the Ridge Estimator



Hence, the confidence ellipsoid of ridge estimator is indeed smaller than the OLS,
If $\boldsymbol{X}$ is an orthogonal design matrix,

$$\text{Var}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2(1+\lambda)^{-2}\mathbb{I}.$$

$$\text{mse}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2\text{trace}(\boldsymbol{W}_\lambda(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{W}_\lambda^\top)+\boldsymbol{\beta}^\top(\boldsymbol{W}_\lambda-\mathbb{I})^\top(\boldsymbol{W}_\lambda-\mathbb{I})\boldsymbol{\beta}.$$

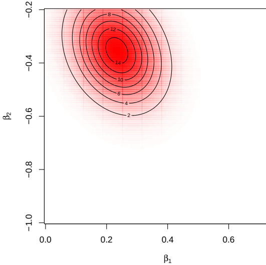
If $\boldsymbol{X}$ is an orthogonal design matrix,

$$\text{mse}[\widehat{\beta}_\lambda^{\text{ridge}}] = \frac{p\sigma^2}{(1+\lambda)^2} + \frac{\lambda^2}{(1+\lambda)^2}\boldsymbol{\beta}^\top\boldsymbol{\beta}$$

Properties of the Ridge Estimator

Hence, the confidence ellipsoid of ridge estimator is indeed smaller than the OLS,
If $\boldsymbol{X}$ is an orthogonal design matrix,

$$\text{Var}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2(1+\lambda)^{-2}\mathbb{I}.$$



$$\text{mse}[\widehat{\beta}_\lambda^{\text{ridge}}] = \sigma^2\text{trace}(\boldsymbol{W}_\lambda(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{W}_\lambda^\top)+\boldsymbol{\beta}^\top(\boldsymbol{W}_\lambda-\mathbb{I})^\top(\boldsymbol{W}_\lambda-\mathbb{I})\boldsymbol{\beta}.$$

If $\boldsymbol{X}$ is an orthogonal design matrix,

$$\text{mse}[\widehat{\beta}_\lambda^{\text{ridge}}] = \frac{p\sigma^2}{(1+\lambda)^2}+\frac{\lambda^2}{(1+\lambda)^2}\boldsymbol{\beta}^\top\boldsymbol{\beta}, \text{ which s minimal for } \lambda^\star = \frac{p\sigma^2}{\boldsymbol{\beta}^\top\boldsymbol{\beta}}$$

SVD decomposition

Consider the singular value decomposition of $\boldsymbol{X}$, $\boldsymbol{X} = \boldsymbol{UDV}^\top$.
Then

$$\widehat{\beta}^{\text{ols}} = \boldsymbol{V}\underbrace{\boldsymbol{D}^{-2}\boldsymbol{D}}\boldsymbol{U}^\top\boldsymbol{y}$$

$$\widehat{\beta}_\lambda^{\text{ridge}} = \boldsymbol{V}\underbrace{(\boldsymbol{D}^2 + \lambda\mathbb{I})^{-1}\boldsymbol{D}}\boldsymbol{U}^\top\boldsymbol{y}$$

Observe that

$$\boldsymbol{D}_{i,i}^{-1} \geq \frac{\boldsymbol{D}_{i,i}}{\boldsymbol{D}_{i,i}^2 + \lambda}$$

hence, the ridge penalty shrinks singular values.
Set now $\boldsymbol{R} = \boldsymbol{UD}$ ($n \times n$ matrix), so that $\boldsymbol{X} = \boldsymbol{RV}^\top$,

$$\widehat{\beta}_\lambda^{\text{ridge}} = \boldsymbol{V}(\boldsymbol{R}^\top\boldsymbol{R} + \lambda\mathbb{I})^{-1}\boldsymbol{R}^\top\boldsymbol{y}$$

see Golub & Reinsh (1970).

Hat matrix and Degrees of Freedom

Recall that with OLS, $\widehat{\boldsymbol{Y}} = \boldsymbol{H}\boldsymbol{Y}$ with

$$\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$$

Similarly, with Ridge estimator, $\widehat{\boldsymbol{Y}} = \boldsymbol{H}_\lambda\boldsymbol{Y}$ with

$$\boldsymbol{H}_\lambda = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\mathbb{I})^{-1}\boldsymbol{X}^\top$$

$$\text{trace}[\boldsymbol{H}_\lambda] = \sum_{j=1}^{p} \frac{\boldsymbol{D}_{j,j}^2}{\boldsymbol{D}_{j,j}^2 + \lambda} \rightarrow 0, \text{ as } \lambda \rightarrow \infty.$$

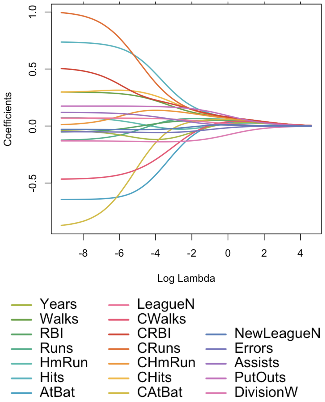
Régression Ridge avec R

On peut utiliser

```
> library(MASS)
> ?lm.ridge
```

ou

```
> library(ISLR)
> library(glmnet)
> Hitters = na.omit(Hitters)
> x = model.matrix(Salary~.,
    Hitters)[,-1]
> y = Hitters$Salary
> ridge_mod = glmnet(x, y, alpha =
    0, family = "gaussian")
> plot(ridge_mod, var="lambda")
```

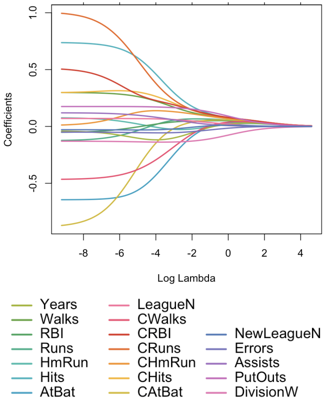


Régression Ridge avec R

L'option "gaussian" fait que les variables sont centrées et réduites, par défaut
i.e. on centre et on réduit les variables explicatives

$$x_j \mapsto \frac{x_j - \overline{x}_j}{s_{x_j}}$$

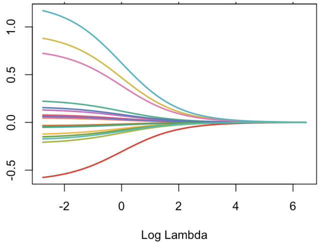
```
> ys = (y-mean(y))/sd(y)
> xs = x
> for(i in 1:ncol(x)) xs[,i] = (x[,
    i]-mean(x[,i]))/sd(x[,i])
> ridge_mod_s = glmnet(xs, ys,
    alpha = 0)
> plot(ridge_mod_s, xvar="lambda")
```



Régression Ridge avec R

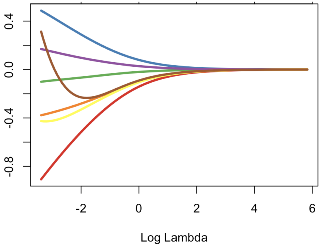
Pour avoir des variables explicatives orthogonales, on peut utiliser une ACP, sur `Hitters`

```
> library(FactoMineR)
x = model.matrix(Salary~., Hitters)
    [,-1]
y = Hitters$Salary
ys = (y-mean(y))/sd(y)
pca = PCA(x,ncp=ncol(x))
pca_x = get_pca_ind(pca)$coord
ridge_pca = glmnet(pca_x, ys, alpha
     = 0,family="gaussian")
plot(ridge_pca, xvar="lambda")
```



ou sur la base `myocarde`

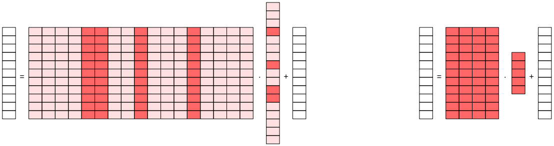
```
pca = PCA(X,ncp=ncol(X))
pca_X = get_pca_ind(pca)$coord
glm_ridge = glmnet(pca_X, y, alpha
    =0, family="binomial")
plot(glm_ridge, xvar="lambda")
```



Sparsity

In several applications, k can be (very) large, but a lot of features are just noise: $\beta_j = 0$ for many j's. Let s denote the number of relevant features, with $s << k$, cf Hastie, Tibshirani & Wainwright (2015),

$$s = \text{card}\{\mathcal{S}\} \text{ where } \mathcal{S} = \{j; \beta_j \neq 0\}$$

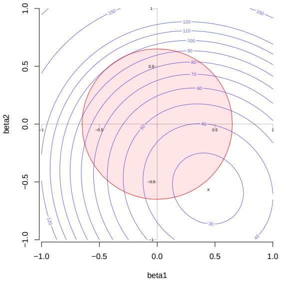


The model is now $y = \boldsymbol{X}_{\mathcal{S}}^{\top}\boldsymbol{\beta}_{\mathcal{S}} + \varepsilon$, where $\boldsymbol{X}_{\mathcal{S}}^{\top}\boldsymbol{X}_{\mathcal{S}}$ is a full rank matrix.

Variable Selection

The Ridge regression problem was to solve

$$\widehat{\beta}^{\text{ridge}} = \underset{\beta\in\{\|\beta\|_{\ell_2}\leq s\}}{\text{argmin}}\{\|\boldsymbol{Y}-\boldsymbol{X}^\top\beta\|_{\ell_2}^2\}$$



Define $\|\boldsymbol{a}\|_{\ell_0}=\sum\mathbf{1}(|a_i|>0)$.
Here $\dim(\beta)=k$ but $\|\beta\|_{\ell_0}=s$.
We wish we could solve

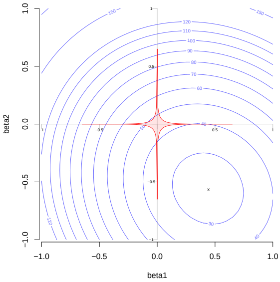
$$\widehat{\beta}^{\text{selec}} = \underset{\beta\in\{\|\beta\|_{\ell_0}=s\}}{\text{argmin}}\{\|\boldsymbol{Y}-\boldsymbol{X}^\top\beta\|_{\ell_2}^2\}$$

Problem: it is usually not possible to describe all possible constraints, since $\binom{s}{k}$ coefficients should be chosen here (with k (very) large).

Variable Selection

The Ridge regression problem was to solve

$$\widehat{\beta}^{\text{ridge}} = \underset{\beta\in\{\|\beta\|_{\ell_2}\leq s\}}{\text{argmin}} \{\|\boldsymbol{Y}-\boldsymbol{X}^\top\beta\|_{\ell_2}^2\}$$



Define $\|\boldsymbol{a}\|_{\ell_0} = \sum \mathbf{1}(|a_i|>0)$.
Here $\dim(\beta)=k$ but $\|\beta\|_{\ell_0}=s$.
We wish we could solve

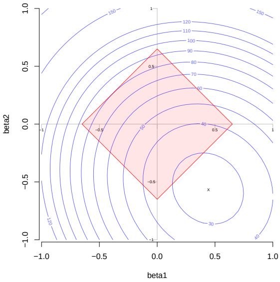
$$\widehat{\beta}^{\text{selec}} = \underset{\beta\in\{\|\beta\|_{\ell_0}=s\}}{\text{argmin}} \{\|\boldsymbol{Y}-\boldsymbol{X}^\top\beta\|_{\ell_2}^2\}$$

Problem: it is usually not possible to describe all possible constraints, since $\binom{s}{k}$ coefficients should be chosen here (with k (very) large).

Variable selection

The Ridge regression problem was to solve

$$\widehat{\beta}^{\text{ridge}} = \underset{\beta \in \{\|\beta\|_{\ell_2} \leq s\}}{\text{argmin}} \{\|\boldsymbol{Y} - \boldsymbol{X}^\top \beta\|_{\ell_2}^2\}$$



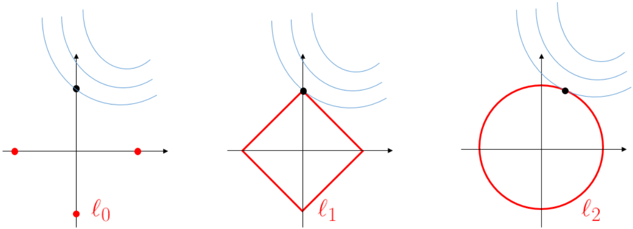
Define $\|\boldsymbol{a}\|_{\ell_0} = \sum \mathbf{1}(|a_i| > 0)$.
Here $\dim(\beta) = k$ but $\|\beta\|_{\ell_0} = s$.
We wish we could solve

$$\widehat{\beta}^{\text{ridge}} = \underset{\beta \in \{\|\beta\|_{\ell_0} = s\}}{\text{argmin}} \{\|\boldsymbol{Y} - \boldsymbol{X}^\top \beta\|_{\ell_2}^2\}$$

Problem: it is usually not possible to describe all possible constraints, since $\binom{s}{k}$ coefficients should be chosen here (with k (very) large).

Sparsity

We might convexify the ℓ_0 "norm", $\|\cdot\|_{\ell_0}$.



On $[-1,+1]^k$, the convex hull of $\|\beta\|_{\ell_0}$ is $\|\beta\|_{\ell_1}$
On $[-a,+a]^k$, the convex hull of $\|\beta\|_{\ell_0}$ is $a^{-1}\|\beta\|_{\ell_1}$
Hence, why not solve

$$\widehat{\beta} = \underset{\beta;\|\beta\|_{\ell_1}\leq\tilde{s}}{\text{argmin}} \{\|\boldsymbol{Y}-\boldsymbol{X}^\top\beta\|_{\ell_2}\}$$

which is equivalent (Kuhn-Tucker theorem) to the Lagragian optimization problem

$$\widehat{\beta} = \text{argmin}\{\|\boldsymbol{Y}-\boldsymbol{X}^\top\beta\|_{\ell_2}^2+\lambda\|\beta\|_{\ell_1}\}$$

LASSO *Least Absolute Shrinkage and Selection Operator*

In an OLS context, we want to solve

LASSO Estimator (OLS)

$$\widehat{\beta}_{\lambda}^{\text{lasso}} = \text{argmin}\left\{\frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^\top\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\}$$

or more generally (when maximizing the log-likelihood)

LASSO Estimator (GLM)

$$\widehat{\beta}_{\lambda}^{\text{lasso}} = \text{argmin}\left\{-\sum_{i=1}^{n}\log f(y_i|\mu_i = g^{-1}(\boldsymbol{x}_i^\top\beta)) + \lambda\sum_{j=1}^{p}|\beta_j|\right\}$$

LASSO with only 1 covariate

Consider a simple regression $y_i = x_i\beta + \varepsilon$, with ℓ_1-penalty and a ℓ_2-loss function. ($\ell 1$) becomes

$$\min\left\{\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{y}^\top\boldsymbol{x}\beta + \beta\boldsymbol{x}^\top\boldsymbol{x}\beta + 2\lambda|\beta|\right\}$$

First order condition can be written

$$-2\boldsymbol{y}^\top\boldsymbol{x} + 2\boldsymbol{x}^\top\boldsymbol{x}\widehat{\beta}\pm 2\lambda = 0.$$

(the sign in $\pm$ being the sign of $\widehat{\beta}$). Assume that least-square estimate ($\lambda = 0$) is (strictly) positive, i.e. $\boldsymbol{y}^\top\boldsymbol{x} > 0$. If λ is not too large $\widehat{\beta}$ and $\widehat{\beta}^{\text{ols}}$ have the same sign, and

$$-2\boldsymbol{y}^\top\boldsymbol{x} + 2\boldsymbol{x}^\top\boldsymbol{x}\widehat{\beta} + 2\lambda = 0.$$

with solution $\widehat{\beta}_\lambda^{\text{lasso}} = \dfrac{\boldsymbol{y}^\top\boldsymbol{x} - \lambda}{\boldsymbol{x}^\top\boldsymbol{x}}$.

LASSO with only 1 covariate

Increase λ so that $\widehat{\beta}_\lambda = 0$.
Increase slightly more, $\widehat{\beta}_\lambda$ cannot become negative, because the sign of the first order condition will change, and we should solve

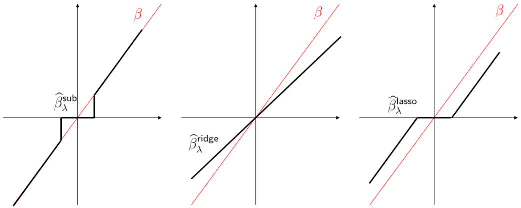
$$-2\boldsymbol{y}^\top\boldsymbol{x} + 2\boldsymbol{x}^\top\boldsymbol{x}\widehat{\beta} - 2\lambda = 0.$$

and solution would be $\widehat{\beta}_\lambda^{\text{lasso}} = \dfrac{\boldsymbol{y}^\top\boldsymbol{x} + \lambda}{\boldsymbol{x}^\top\boldsymbol{x}}$. But that solution is positive (we assumed that $\boldsymbol{y}^\top\boldsymbol{x} > 0$), to we should have $\widehat{\beta}_\lambda < 0$.
Thus, at some point $\widehat{\beta}_\lambda = 0$, which is a corner solution.
In higher dimension, see Tibshirani & Wasserman (2016) or Candès & Plan (2009)
With some additional technical assumption, that LASSO estimator is "sparsistent" in the sense that the support of $\widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}$ is the same as $\boldsymbol{\beta}$,

ℓ_0, ℓ_1 and ℓ_2 penalty

Thus, LASSO can be used for variable selection - see Hastie *et al.* (2001).
Generally, $\widehat{\beta}_\lambda^{\text{lasso}}$ is a biased estimator but its variance can be small enough to have a smaller least squared error than the OLS estimate.
With orthonormal covariates, one can prove that

$$\widehat{\beta}_{\lambda,j}^{\text{sub}} = \widehat{\beta}_j^{\text{ols}} \mathbf{1}_{|\widehat{\beta}_{\lambda,j}^{\text{sub}}|>b}, \quad \widehat{\beta}_{\lambda,j}^{\text{ridge}} = \frac{\widehat{\beta}_j^{\text{ols}}}{1+\lambda} \text{ and } \widehat{\beta}_{\lambda,j}^{\text{lasso}} = \text{sign}[\widehat{\beta}_j^{\text{ols}}] \cdot (|\widehat{\beta}_j^{\text{ols}}| - \lambda)_+.$$



OLS pénalisé

Recall that the subdifferential of $x \mapsto |x|$ is

$$\partial|x| = \begin{cases} \{-1\} & \text{if } x < 0 \\ [-1, +1] & \text{if } x = 0 \\ \{+1\} & \text{if } x > 0 \end{cases}$$

Here, we want to find $\min\{\|\boldsymbol{y} - \boldsymbol{X}\beta\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\}$, the *first order condition* is

$$\boldsymbol{0} \in -2\boldsymbol{X}^\top\boldsymbol{y} + 2\boldsymbol{X}^\top\boldsymbol{X}\beta^\star + \lambda\partial\|\boldsymbol{\beta}^\star\|_1$$

i.e., for the (univariate) jth condition, if all variables are orthogonal

$$0 \in \; -\widehat{\boldsymbol{\beta}}_j^{\text{ols}} + \boldsymbol{\beta}_j^\star + \frac{\lambda}{2}\partial|\boldsymbol{\beta}_j^\star|.$$

i.e.

$$\boldsymbol{\beta}_j^\star = \begin{cases} \widehat{\boldsymbol{\beta}}_j^{\text{ols}} + \lambda/2 \text{ if } \boldsymbol{\beta}_j^\star < 0 \\ \widehat{\boldsymbol{\beta}}_j^{\text{ols}} - \lambda/2 \text{ if } \boldsymbol{\beta}_j^\star > 0 \end{cases}$$

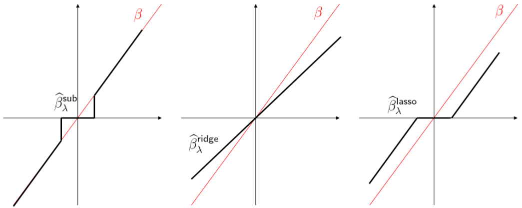
OLS pénalisé

Let us define the soft-thresholding function,

$$S_\gamma(z) = \text{sign}(z) \cdot (|z| - \gamma)_+$$

then $\beta_j^\star = S_{\lambda/2}(\widehat{\beta}_j^{\text{ols}})$.

$$\widehat{\beta}_{\lambda,j}^{\text{sub}} = \widehat{\beta}_j^{\text{ols}} \mathbf{1}_{|\widehat{\beta}_{\lambda,j}^{\text{sub}}|>b}, \quad \widehat{\beta}_{\lambda,j}^{\text{ridge}} = \frac{\widehat{\beta}_j^{\text{ols}}}{1+\lambda} \quad \text{and} \quad \widehat{\beta}_{\lambda,j}^{\text{lasso}} = \text{sign}[\widehat{\beta}_j^{\text{ols}}] \cdot (|\widehat{\beta}_j^{\text{ols}}| - \lambda)_+$$



OLS pénalisé

In a general context, set

$$\mathbf{r}_j = \mathbf{y} - \left(\beta_0\mathbf{1} + \sum_{k\neq j}\beta_k\mathbf{x}_k\right) = \mathbf{y} - \widehat{\mathbf{y}}^{(j)}$$

so that the optimization problem can be written, equivalently

$$\min\left\{\frac{1}{2n}\sum_{j=1}^{p}[\mathbf{r}_j - \beta_j\mathbf{x}_j]^2 + \lambda|\beta_j|\right\}$$

hence

$$\min\left\{\frac{1}{2n}\sum_{j=1}^{p}\beta_j^2\|\mathbf{x}_j\| - 2\beta_j\mathbf{r}_j^T\mathbf{x}_j + \lambda|\beta_j|\right\}$$

and one gets $\beta_{j,\lambda} = \dfrac{1}{\|\mathbf{x}_j\|^2}S(\mathbf{r}_j^T\mathbf{x}_j, n\lambda)$ or, if we develop

$$\beta_{j,\lambda} = \frac{1}{\sum_i x_{ij}^2}S_{n\lambda}\left(\sum_i x_{i,j}[y_i - \widehat{y}_i^{(j)}]\right)$$

WLS pénalisé

or, $\beta_{j,\lambda,\omega} = \dfrac{1}{\sum_i \omega_i x_{ij}^2} S_{n\lambda}\left(\sum_i \omega_i x_{i,j}[y_i - \widehat{y}_i^{(j)}]\right)$, with weights

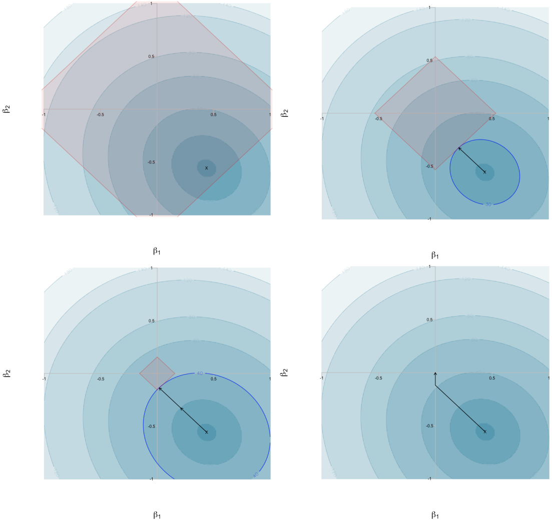
Algorithm 1: OLS LASSO

Initialisation: $\boldsymbol{\beta}^{(0)}$ and $\beta_0^{(0)} \leftarrow n^{-1}\sum_i (y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}^{(0)})$;
for t=1,2,... **do**
- $\alpha_0 \leftarrow \overline{y}$ and $\alpha_j \leftarrow \widehat{\beta}_j^{(t-1)}$ for $j = 1, 2, \cdots, k$;
- **for** j=1,2,...,k **do**
  - **for** i=1,2,...,n **do**
    - $r_{i,j} \leftarrow \boldsymbol{z}_i^{(t)} - \alpha_0 - \sum_\ell \alpha_\ell x_{i\ell}$
  - $u_j^{(t)} \leftarrow \sum_i \omega_i^{(t)} r_{ij} x_{ij}$ and $v_j^{(t)} \leftarrow \sum_i \omega_i^{(t)} x_{ij}^2$;
  - $\alpha_j = \text{sign}(u_j^{(t)}) \left(\dfrac{|u_j^{(t)} - \lambda|}{v_j^{(t)}} \right)_+$;
- $\widehat{\beta}_0^{(t)} \leftarrow \alpha_0$ and $\widehat{\boldsymbol{\beta}}_j^{(t)} \leftarrow \alpha_j$

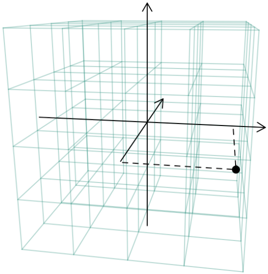
LASSO Regression

No explicit solution...

If $\lambda \to 0$, $\widehat{\beta}_0^{\text{lasso}} = \widehat{\beta}^{\text{ols}}$
If $\lambda \to \infty$, $\widehat{\beta}_\infty^{\text{lasso}} = \mathbf{0}$.
For some λ, there are k's such that $\widehat{\beta}_{k,\lambda}^{\text{lasso}} = 0$.
Further, $\lambda \mapsto \widehat{\beta}_{k,\lambda}^{\text{lasso}}$ is piecewise linear



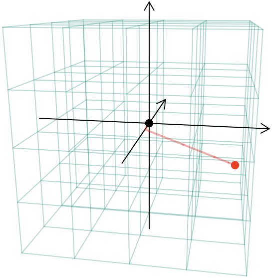
LASSO Regression



In the orthogonal case, $\boldsymbol{X}^\top\boldsymbol{X}=\mathbb{I}$,

$$\widehat{\beta}^{\text{lasso}}_{k,\lambda}=\text{sign}(\widehat{\beta}^{\text{ols}}_{k})\left(|\widehat{\beta}^{\text{ols}}_{k}|-\frac{\lambda}{2}\right)$$

i.e. the LASSO estimate is related to the soft threshold function...



Optimal LASSO Penalty

Use cross validation, e.g. K-fold,

$$\widehat{\beta}_{(-k)}(\lambda) = \text{argmin}\left\{\sum_{i\notin\mathcal{I}_k}[y_i - \boldsymbol{x}_i^\top\beta]^2 + \lambda\|\beta\|_{\ell_1}\right\}$$

then compute the sum of the squared errors,

$$Q_k(\lambda) = \sum_{i\in\mathcal{I}_k}[y_i - \boldsymbol{x}_i^\top\widehat{\beta}_{(-k)}(\lambda)]^2$$

and finally solve

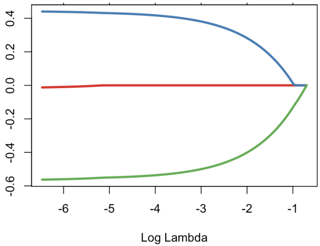
$$\lambda^\star = \text{argmin}\left\{\overline{Q}(\lambda) = \frac{1}{K}\sum_k Q_k(\lambda)\right\}$$

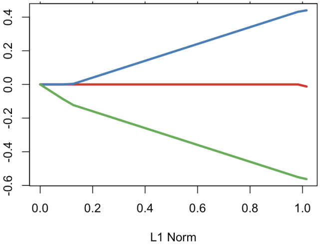
Note that this might overfit, so Hastie, Tibshiriani & Friedman (2009) suggest the largest λ such that

$$\overline{Q}(\lambda) \le \overline{Q}(\lambda^\star) + \text{se}[\lambda^\star] \text{ with } \text{se}[\lambda]^2 = \frac{1}{K^2}\sum_{k=1}^K[Q_k(\lambda) - \overline{Q}(\lambda)]^2$$

LASSO with R

```
> library(glmnet)
> chicago=read.table("http://freakonometrics.free.fr/chicago.txt",header=TRUE,sep=";")
> standardize <-  function(x) {(x-mean(x))/sd(x)}
y = chicago[,1]
y = standarize(y)
X = chicago[,2:4]
> for(i in 1:3) X[,i] <- standardize(X[, i])
X = as.matrix(X)
> library(glmnet)
> glm_lasso = glmnet(X, y, alpha=1, family="gaussian", stardardize=TRUE)
> plot(glm_lasso,xvar="lambda")
> plot(glm_lasso,xvar="norm")
```

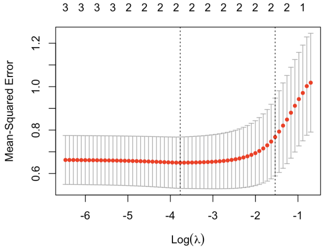




LASSO with R

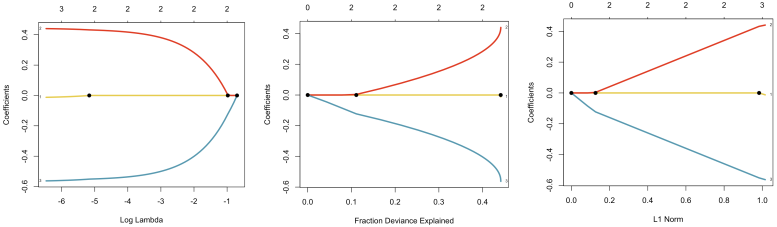
```
> glm_lasso$beta[,10]
       X_1        X_2        X_3
 0.0000000  0.1897653 -0.3087704
 > glm_lasso$beta[,60]
       X_1        X_2        X_3
-0.0108099  0.4393318 -0.5612430
> plot(glm_lasso,xvar="lambda")
```

```
> cvmfit = cv.glmnet(X, y,
   family = "gaussian",alpha=1)
> plot(cvmfit)
> cvmfit

Measure: Mean-Squared Error

     Lambda Measure     SE Non
min 0.02306   0.6497 0.1184   2
1se 0.21507   0.7678 0.1793   2
```

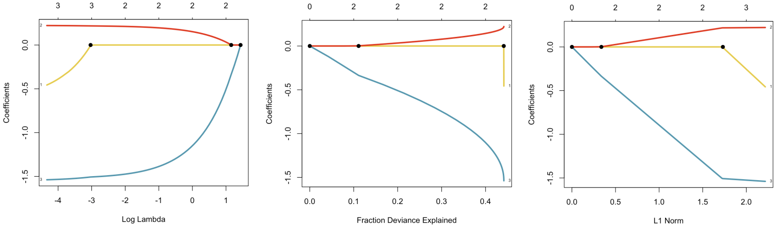


LASSO with R

Lasso with normalized (centered and scaled) variables



Lasso without normalization



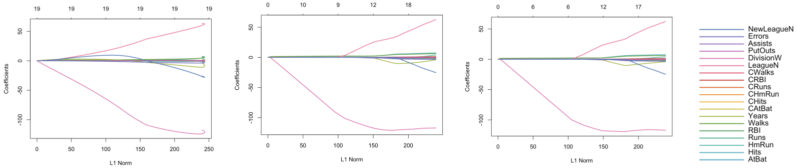
Elastic Net

Singularities at the vertexes (sparsity) and strict convex edges.

Elastic-net (α) Estimator (OLS)

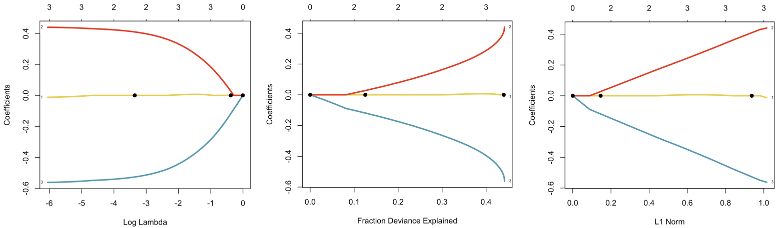
$$\widehat{\beta}_{\lambda}^{\text{en}-\alpha} = \text{argmin}\left\{\sum_{i=1}^{n}(y_i - \beta_0 - \beta^\top x_i)^2 + \lambda\left[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1\right.\right.$$

Comparison of ridge, elastic-net, Lasso

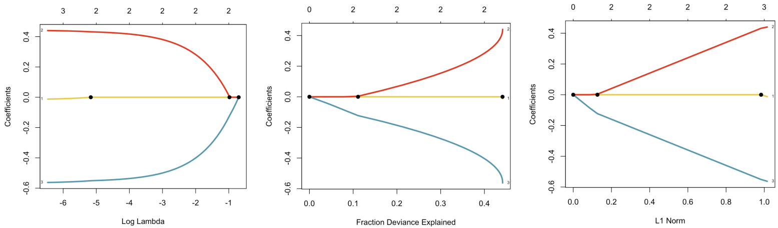


Elastic Net

Elastic-net with normalized (centered and scaled) variables



Lasso with normalized (centered and scaled) variables



GAM, splines and Ridge regression

Consider a univariate nonlinear regression problem, so that $\mathbb{E}[Y|X=x]=m(x)$.

Given a sample $\{(y_1,x_1),\cdots,(y_n,x_n)\}$, consider the following penalized problem

$$m^\star = \underset{m\in\mathcal{C}^2}{\text{argmin}}\left\{\sum_{i=1}^n (y_i - m(x_i))^2 + \lambda\int_{\mathbb{R}} m''(x)dx\right\}$$

with the Residual sum of squares on the left, and a penalty for the roughness of the function.

The solution is a natural cubic spline with knots at unique values of x, see Eubanks (1999).

Consider some spline basis $\{h_1,\cdots,h_n\}$,

$$m(x)=\sum_{i=1}^n \beta_i h_i(x)$$

Let $\boldsymbol{H}$ and $\boldsymbol{\Omega}$ be the $n\times n$ matrices $H_{i,j}=h_j(x_i)$, and

$$\Omega_{i,j}=\int_{\mathbb{R}} h_i''(x)h_j''(x)dx$$

GAM, splines and Ridge regression

Then the objective function can be written

$$(\boldsymbol{y}-\boldsymbol{H}\beta)^\top(\boldsymbol{y}-\boldsymbol{H}\beta)+\lambda\beta^\top\boldsymbol{\Omega}\beta$$

Recognize here a generalized Ridge regression, with solution

$$\widehat{\beta}_\lambda=\big(\boldsymbol{H}^\top\boldsymbol{H}+\lambda\Omega\big)^{-1}\boldsymbol{H}^\top\boldsymbol{y}.$$

Note that predicted values are linear functions of the observed value since

$$\widehat{\boldsymbol{y}}=\boldsymbol{H}\big(\boldsymbol{H}^\top\boldsymbol{H}+\lambda\Omega\big)^{-1}\boldsymbol{H}^\top\boldsymbol{y}=\boldsymbol{S}_\lambda\boldsymbol{y},$$

with degrees of freedom trace$(\boldsymbol{S}_\lambda)$.
One can obtain the so-called Reinsch form by considering the singular value decomposition of $\boldsymbol{H}=\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$.

GAM, splines and Ridge regression

Here $\boldsymbol{U}$ is orthogonal since $\boldsymbol{H}$ is square ($n \times n$), and $\boldsymbol{D}$ is here invertible. Then

$$\boldsymbol{S}_\lambda = (\mathbb{I} + \lambda \boldsymbol{U}^\top \boldsymbol{D}^{-1} \boldsymbol{V}^\top \boldsymbol{\Omega} \boldsymbol{V} \boldsymbol{D}^{-1} \boldsymbol{U})^{-1} = (\mathbb{I} + \lambda \boldsymbol{K})^{-1}$$

where $\boldsymbol{K}$ is a positive semidefinite matrix, $\boldsymbol{K} = \boldsymbol{B}\boldsymbol{\Delta}\boldsymbol{B}^\top$, where columns of $\boldsymbol{B}$ are know as the Demmler-Reinsch basis.

In that (orthonormal) basis, $\boldsymbol{S}_\lambda$ is a diagonal matrix,

$$\boldsymbol{S}_\lambda = \boldsymbol{B}\big(\mathbb{I} + \lambda \boldsymbol{\Delta}\big)^{-1} \boldsymbol{B}^\top$$

Observe that $\boldsymbol{S}_\lambda \boldsymbol{B}_k = \dfrac{1}{1 + \lambda \Delta_{k,k}} \boldsymbol{B}_k$.

Here again, eigenvalues are shrinkage coefficients of basis vectors.

With more covariates, consider an additive problem

$$(h_1, \cdots, h_p)^\star = \underset{h_1,\cdots,h_p \in \mathcal{C}^2}{\text{argmin}} \left\{ \sum_{i=1}^n \left(y_i - \sum_{j=1}^p m(x_{i,j}) \right)^2 + \lambda \sum_{j=1}^p \int_{\mathbb{R}} m_j''(x) dx \right\}$$

GAM, splines and Ridge regression

which can be written

$$\min\left\{(\boldsymbol{y}-\sum_{j=1}^{p}\boldsymbol{H}_j\boldsymbol{\beta}_j)^\top(\boldsymbol{y}-\sum_{j=1}^{p}\boldsymbol{H}_j\boldsymbol{\beta}_j)+\lambda\big(\boldsymbol{\beta}_1^\top\sum_{j=1}^{p}\boldsymbol{\Omega}_j\boldsymbol{\beta}_j\big)\right\}$$

where each matrix $\boldsymbol{H}_j$ is a Demmler-Reinsch basis for variable x_j.
Chouldechova & Hastie (2015)
Assume that the mean function for the jth variable is
$m_j(x)=\alpha_j x+\boldsymbol{m}_j(x)^\top\boldsymbol{\beta}_j$. One can write

$$\min\Big\{(\boldsymbol{y}-\alpha_0-\sum_{j=1}^{p}\alpha_j x_j-\sum_{j=1}^{p}\boldsymbol{H}_j\boldsymbol{\beta}_j)^\top(\boldsymbol{y}-\alpha_0-\sum_{j=1}^{p}\alpha_j x_j-\sum_{j=1}^{p}\boldsymbol{H}_j\boldsymbol{\beta}_j)$$

$$+\lambda\big(\gamma|\alpha_1|+(1-\gamma)\|\boldsymbol{\beta}_j\|_{\Omega_j}\big)+\big(\psi_1\boldsymbol{\beta}_1^\top\boldsymbol{\Omega}_1\boldsymbol{\beta}_1+\cdots+\psi_p\boldsymbol{\beta}_p^\top\boldsymbol{\Omega}_p\boldsymbol{\beta}_p\big)\Big\}$$

where $\|\boldsymbol{\beta}_j\|_{\Omega_j}=\sqrt{\boldsymbol{\beta}_j^\top\boldsymbol{\Omega}_j\boldsymbol{\beta}_j}$.

GAM, splines and Ridge regression

The second term is the selection penalty, with a mixture of ℓ_1 and ℓ_2 (type) norm-based penalty

The third term is the end-to-path penalty (GAM type when $\lambda = 0$).

For each predictor x_j, there are three possibilities

- zero, $\alpha_j = 0$ and $\boldsymbol{\beta}_j = \mathbf{0}$
- linear, $\alpha_j \neq 0$ and $\boldsymbol{\beta}_j = \mathbf{0}$
- nonlinear, $\boldsymbol{\beta}_j \neq \mathbf{0}$