# Le Lasso
# La solution component wise

S.Canu
APPS ITI /INSA de Rouen Normandie

2 octobre 2023

# Le Lasso

- Données
  - $X \in \mathbb{R}^{n\times p}$ les $p$ variables observées $n$ fois
  - $y \in \mathbb{R}^n$ la variable réponse
- Inconnues
  - $\beta \in \mathbb{R}^p$
- hypothèses : pas de biais (pas de terme constant)
  - $mean(X) = 0$
  - $\|X_j\|^2 = 1$
  - $mean(y) = 0$

**le cout pénalisé**

$$J_\lambda(\beta) \;=\; \tfrac{1}{2}\|X\beta - y\|^2 + \lambda\|\beta\|_1 \;,$$

**Les conditions d'optimalité**

$$0 \in \partial J_\lambda(\beta) \;=\; X^\top(X\beta - y) + \lambda\partial(\|\beta\|_1) \;,$$

# La notion de chemin de régularisation

$$\min_{\beta \in \mathbb{R}^p} J_\lambda(\beta) \quad \text{avec} \quad J_\lambda(\beta) = \tfrac{1}{2}\|X\beta - y\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$\partial J(\beta) = X^\top (X\beta - y) + \lambda s(\beta) \qquad \text{avec } s_j(\beta) = \begin{cases} 1 & \text{si } \beta_j > 0 \\ [-1, 1] & \text{si } \beta_j = 0 \\ -1 & \text{si } \beta_j < 0 \end{cases}$$

$\beta$ optimal si $0 \in \partial J(\beta)$ c'est-à-dire si pour $g(\beta) = X^\top (X\beta - y)$

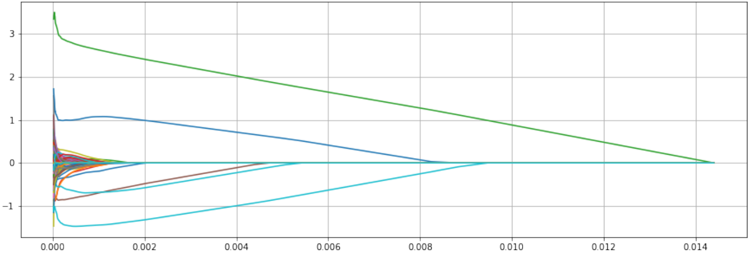
$$\begin{cases} g_j(\beta) + \lambda = 0 & \text{si } \beta_j > 0 \\ -\lambda \leq g \leq \lambda & \text{si } \beta_j = 0 \\ g_j(\beta) - \lambda = 0 & \text{si } \beta_j < 0 \end{cases}$$

Un cas intéressant :

$$\text{si } \lambda \geq \max_j (X^\top y)_j \qquad \text{alors} \qquad \beta = 0$$

# La notion de chemin de régularisation

$\beta(\lambda)$




$\lambda$ large $\Rightarrow \beta = 0$ so that $|X^\top y| \leq \lambda$

the first non zero component for vector $\beta$ appears for $\lambda = \max |X^\top y|$

the regularization path is piecewise linear

# Algo CW pour le Lasso

$$J_\lambda(\boldsymbol{\beta}) \;=\; \tfrac{1}{2}\|X\boldsymbol{\beta} - y\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \;,$$

## Algo CW pour le Lasso

itérer, tant que on n'a pas convergé :

$$\begin{array}{l} \text{Pour } j = 1, p \\ \quad \text{fixer toutes les variables sauf } \beta_j \\ \quad \text{résoudre le Lasso en une dimension :} \\ \qquad \hat{\beta}_j \;=\; \arg\min_{\beta \in \mathbb{R}} J_\lambda^{(j)}(\beta) \end{array} \tag{1}$$

Le cout du Lasso en une dimension

$$J_\lambda^{(j)}(\beta) = \tfrac{1}{2}\|X_j\beta - \left(y - X\beta^{(-j)}\right)\|^2 + \lambda|\beta|$$

# Le Lasso en 1d

$$\min_{\beta \in \mathbb{R}} J_\lambda^{(1d)}(\beta) \qquad \text{avec} \qquad J_\lambda^{(1d)}(\beta) = \frac{1}{2}\|\boldsymbol{x}\beta - \boldsymbol{r}\|^2 + \lambda|\beta|$$

$$\partial_\beta J_\lambda^{(1d)}(\beta) = \boldsymbol{x}^\top(\boldsymbol{x}\beta - \boldsymbol{r}) \; + \left\{ \begin{array}{ll} \lambda\alpha & \text{if } \beta = 0 \\ \lambda \text{sign}(\beta) & \text{else.} \end{array} \right.$$

$$0 \in \partial_\beta J_\lambda^{(1d)}(\beta) \quad \Leftrightarrow \quad \left\{ \begin{array}{ll} \exists \alpha \in [-1, 1], -\boldsymbol{x}^\top\boldsymbol{r} + \lambda\alpha = 0 & \text{if } \beta = 0 \\ \|\boldsymbol{x}\|^2\beta - \boldsymbol{x}^\top\boldsymbol{r} + \lambda\text{sign}(\beta) = 0 & \text{else.} \end{array} \right.$$

The differential part :

$$\left\{ \begin{array}{lll} \text{if } \beta > 0 & \beta = \frac{\boldsymbol{x}^\top\boldsymbol{r} - \lambda}{\|\boldsymbol{x}\|^2} & \text{it works when } \boldsymbol{x}^\top\boldsymbol{r} > \lambda \\ \text{if } \beta < 0 & \beta = \frac{\boldsymbol{x}^\top\boldsymbol{r} + \lambda}{\|\boldsymbol{x}\|^2} & \text{it works when } \boldsymbol{x}^\top\boldsymbol{r} < -\lambda \end{array} \right.$$

It remains the non differential part : when $-\lambda < \boldsymbol{x}^\top\boldsymbol{r} < \lambda$ then $\beta = 0$

# Le Lasso en 1d

The differential part :

$$\begin{cases} \text{if } \beta > 0 & \beta = \frac{\boldsymbol{x}^\top \boldsymbol{r} - \lambda}{\|\boldsymbol{x}\|^2} & \text{it works when } \boldsymbol{x}^\top \boldsymbol{r} > \lambda \\ \text{if } \beta < 0 & \beta = \frac{\boldsymbol{x}^\top \boldsymbol{r} + \lambda}{\|\boldsymbol{x}\|^2} & \text{it works when } \boldsymbol{x}^\top \boldsymbol{r} < -\lambda \end{cases}$$

It remains the non differential part : when $-\lambda < \boldsymbol{x}^\top \boldsymbol{r} < \lambda$ then $\beta = 0$

**if $\|\boldsymbol{x}\|^2 = 1$**

$$\hat{\beta}_{Lasso1d} = \begin{cases} 0 & \text{if } |\boldsymbol{x}^\top \boldsymbol{r}| \leq \lambda \\ \text{sign}(\boldsymbol{x}^\top \boldsymbol{r})(|\boldsymbol{x}^\top \boldsymbol{r}| - \lambda) & \text{else.} \end{cases}$$

Or in one line :

$$\text{sign}(\boldsymbol{x}^\top \boldsymbol{r}) \max\big((|\boldsymbol{x}^\top \boldsymbol{r}| - \lambda), 0\big)$$

# Algo CW pour le Lasso

$$\text{Pour } j = 1, p \;:\; \hat{\beta}_j \;=\; \arg\min_{\beta\in\mathbb{R}} J^{(j)}(\beta)$$

$$\begin{aligned} J^{(j)}(\beta) &= \tfrac{1}{2}\|y - X\beta^{(-j)} - X_j\beta\|^2 + \lambda|\beta| \\ &= \tfrac{1}{2}\|\boldsymbol{r} - X_j\beta\|^2 + \lambda|\beta| \end{aligned} \tag{2}$$

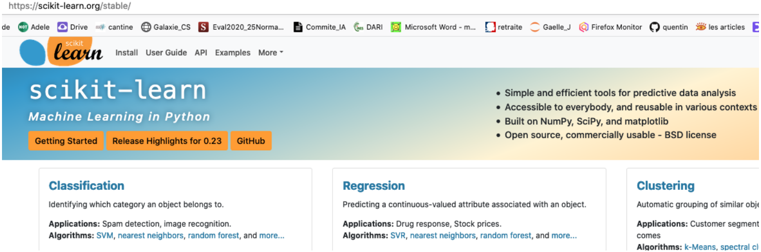
avec $\boldsymbol{r} = y - X\beta^{(-j)}$ et $\beta^{(-j)} = (\beta_1, \dots, \beta_{j-1}, 0, \beta_{j+1}, \dots, \beta_p)$

| Maths | Info |
|---|---|
| Pour $j = 1, p$ | for $j$ in range($p$) : |
| $\quad \beta^{(-j)} = (\beta_1, \dots, \beta_{j-1}, 0, \beta_{j+1}, \dots, \beta_p)$ | $\quad bj = beta; bj[j] = 0;$ |
| $\quad r = y - X\beta^{(-j)}$ | $\quad r = y - X@bj$ |
| $\quad \hat{\beta}_j \;=\; \arg\min_{\beta\in\mathbb{R}} J^{(j)}(\beta, X_j, z)$ | $\quad beta[j] = \text{sign}(\boldsymbol{x}^\top \boldsymbol{r}) \max((\lvert\boldsymbol{x}^\top$ |
| fin de pour | end |

# Codage

1. let's begin with $\beta = 0$
2. in that case $r \leftarrow y$ since $X\beta = 0$
3. fit a single (the $j$th) component of vector $\beta$ :
   $\beta[j] = \text{sign}(\boldsymbol{x}^\top \boldsymbol{r}) \max\big((|\boldsymbol{x}^\top \boldsymbol{r}| - \lambda), 0\big)$
4. update $r$ : $r \leftarrow r - X[:, j]\beta[j]$
5. choose another component $j'$
6. $r \leftarrow r + X[:, j']\beta[j']$
7. fit the other component $j'$ : $\beta[j'] = \text{sign}(\boldsymbol{x}^\top \boldsymbol{r}) \max\big((|\boldsymbol{x}^\top \boldsymbol{r}| - \lambda), 0\big)$
8. update $r$ : $r \leftarrow r - X[:, j']\beta[j']$
9. ...iterate (go to 5)

# Sklearn et le Lasso




- Lasso
- LassoCV
- lasso_path

- LassoLars
- LassoLarsCV
- lars_path

- sklearn.decomposition.sparse_encode

# sklearn.linear_model.Lasso

*class* `sklearn.linear_model.Lasso`(*alpha=1.0, *, fit_intercept=True, normalize=False, precompute=False, copy_X=True, max_iter=1000, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*) [source]

Linear Model trained with L1 prior as regularizer (aka the Lasso)

The optimization objective for Lasso is:

```
(1 / (2 * n_samples)) * ||y - Xw||^2_2 + alpha * ||w||_1
```

Technically the Lasso model is optimizing the same objective function as the Elastic Net with `l1_ratio=1.0` (no L2 penalty).

Read more in the User Guide.

**Parameters:**

**alpha : *float, default=1.0***
Constant that multiplies the L1 term. Defaults to 1.0. `alpha = 0` is equivalent to an ordinary least square, solved by the `LinearRegression` object. For numerical reasons, using `alpha = 0` with the `Lasso` object is not advised. Given this, you should use the `LinearRegression` object.

**fit_intercept : *bool, default=True***
Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (i.e. data is expected to be centered).

**normalize : *bool, default=False***
This parameter is ignored when `fit_intercept` is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use `sklearn.preprocessing.StandardScaler` before calling `fit` on an estimator with `normalize=False`.

# 3.2.4.1.3. sklearn.linear_model.LassoCV

*class* `sklearn.linear_model.LassoCV`(**, eps=0.001, n_alphas=100, alphas=None, fit_intercept=True, normalize=False, precompute='auto', max_iter=1000, tol=0.0001, copy_X=True, cv=None, verbose=False, n_jobs=None, positive=False, random_state=None, selection='cyclic'*) [source]

Lasso linear model with iterative fitting along a regularization path.

See glossary entry for cross-validation estimator.

The best model is selected by cross-validation.

The optimization objective for Lasso is:

```
(1 / (2 * n_samples)) * ||y - Xw||^2_2 + alpha * ||w||_1
```

Read more in the User Guide.

**Parameters:**

**eps : *float, default=1e-3***
Length of the path. `eps=1e-3` means that `alpha_min / alpha_max = 1e-3`.

**n_alphas : *int, default=100***
Number of alphas along the regularization path

# Lasso et Elastic Net

Le Lasso

$$J_\lambda(\beta) \; = \; \tfrac{1}{2}\|X'\beta - y'\|^2 + \lambda\|\beta\|_1 \; ,$$

Elastic Net

$$J_{\mathrm{el}}(\beta) \; = \; \tfrac{1}{2}\|X\beta - y\|^2 + \lambda\|\beta\|_1 + \gamma\|\beta\|_2^2 \; ,$$

Lasso $=$ Elastic Net with $\gamma = 0, X = X', y = y'$

**Elastic Net $=$ Lasso with $X' = (X^\top, \sqrt{\gamma} I_p)^\top, y' = (y^\top, 0_p)^\top$**

$$\|X\boldsymbol{\beta} - y\|_2^2 + \underbrace{\gamma\,\|\boldsymbol{\beta}\|_2^2}_{\|\sqrt{\gamma}\boldsymbol{\beta}\|_2^2} = \left\| \underbrace{\begin{bmatrix} X \\ \sqrt{\gamma} I_p \end{bmatrix}}_{=:X'} \boldsymbol{\beta} - \underbrace{\begin{bmatrix} y \\ 0_p \end{bmatrix}}_{=:\underline{y}'} \right\|_2^2 = \|X'\boldsymbol{\beta} - y'\|_2^2 .$$

Regularization and variable selection via the elastic net H Zou, T Hastie, 2005

# Reweigthed least square (Lasso and Ridge)

$$J_\lambda(\beta) \;=\; \tfrac{1}{2}\|X\beta - y\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \frac{\beta_j^2}{|\beta_j|} \;,$$

the idea : iterate the weighted rigde towards a fixed point

$$\beta^{(k+1)} \;=\; \arg\min_{\beta} \tfrac{1}{2}\|X\beta - y\|^2 + \lambda \sum_{j=1}^{p} \frac{\beta_j^2}{|\beta_j^{(k)}|} = w_j \; \beta_j^2 \;,$$

with $w_j = 1/|\beta_j^{(k)}|$, that is

$$\beta^{(k+1)} \;=\; (X^\top X + \lambda W)^{-1} X^\top y \qquad \text{with } diag(W) = \frac{1}{|\beta_j^{(k)}|}$$

W = Identité
Tantque on n'a pas convergé
  $\beta = \arg\min \tfrac{1}{2}\|X\beta - y\|^2 + \lambda \beta^\top W \beta$
  $W = diag(1/|\beta|)$
fin de tantque

# The adaptive Lasso

$$J_\lambda(\beta) \;=\; \tfrac{1}{2}\|X\beta - y\|^2 + \lambda \sum_{j=1}^{p} w_j|\beta_j|\;,$$

$$w_j = \frac{1}{|\hat{\beta}_j^{(\mathrm{ls})}|^\gamma}$$

```
w = np.linalg.solve(X.T@X,X.T@y)
w = 1/np.sqrt(np.abs(w))
X_c = X/w
c = mon_lasso(X_c,y,lambda)
beta = c/w
```

The Adaptive Lasso and Its Oracle Properties, H. Zou, JASA, 2012