

Le Lasso

La solution component wise

S.Canu
MLA ITI /INSA de Rouen Normandie

6 octobre 2025

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)
 - ▶ $\text{mean}(X) = 0$
 - ▶ $\|X_j\|^2 = 1$
 - ▶ $\text{mean}(y) = 0$

le cout pénalisé

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

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

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

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

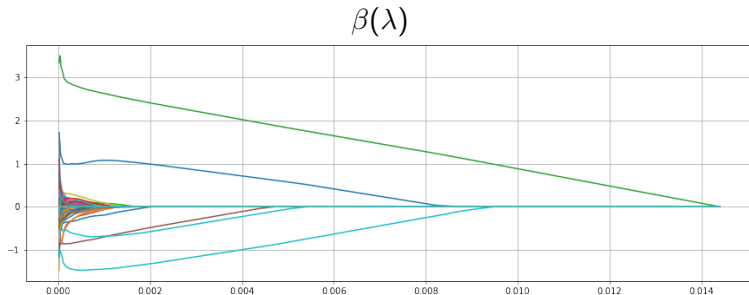
β optimal si $0 \in \partial J(\beta)$ c'est-à-dire si pour $g(\beta) = X^\top(X\beta - y)$

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

Un cas intéressant :

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

La notion de chemin de régularisation



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

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

the regularization path is piecewise linear

Algo CW pour le Lasso

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

Algo CW pour le Lasso

itérer, tant que on n'a pas convergé :

Pour $j = 1, p$

fixer toutes les variables sauf β_j

résoudre le Lasso en une dimension :

$$\hat{\beta}_j = \arg \min_{\beta \in \mathbb{R}} J_\lambda^{(j)}(\beta)$$

(1)

Le cout du Lasso en une dimension

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

Le Lasso en 1d

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

$$\partial_{\beta} J_{\lambda}^{(1d)}(\beta) = \mathbf{x}^{\top}(\mathbf{x}\beta - \mathbf{r}) + \begin{cases} \lambda\alpha & \text{if } \beta = 0 \\ \lambda \text{sign}(\beta) & \text{else.} \end{cases}$$

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

The differential part :

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

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

Le Lasso en 1d

The differential part :

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

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

if $\|\mathbf{x}\|^2 = 1$

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

Or in one line :

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

Algo CW pour le Lasso

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

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

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

Maths

Pour $j = 1, p$

$$\beta^{(-j)} = (\beta_1, \dots, \beta_{j-1}, 0, \beta_{j+1}, \dots, \beta_p)$$

$$r = y - X\beta^{(-j)}$$

$$\hat{\beta}_j = \arg \min_{\beta \in \mathbb{R}} J^{(j)}(\beta, X_j, z)$$

fin de pour

Info

for j in range(p) :

$$bj = \text{beta}; bj[j] = 0;$$

$$r = y - X @ bj$$

$$\text{beta}[j] = \text{sign}(x^T r) \max((|x^T$$

end

Codage

- ❶ let's begin with $\beta = 0$
- ❷ in that case $r \leftarrow y$ since $X\beta = 0$
- ❸ fit a single (the j th) component of vector β :
$$\beta[j] = \text{sign}(\mathbf{x}^\top \mathbf{r}) \max(|\mathbf{x}^\top \mathbf{r}| - \lambda, 0)$$
- ❹ update $r : r \leftarrow r - X[:,j]\beta[j]$
- ❺ choose another component j'
- ❻ $r \leftarrow r + X[:,j']\beta[j']$
- ❼ fit the other component $j' : \beta[j'] = \text{sign}(\mathbf{x}^\top \mathbf{r}) \max(|\mathbf{x}^\top \mathbf{r}| - \lambda, 0)$
- ❽ update $r : r \leftarrow r - X[:,j']\beta[j']$
- ❾ ...iterate (go to 5)

LLM ?

Expliquer l'algorithme component-wise pour résoudre le problème de la régression Lasso Réécrire le problème d'optimisation du lasso comme une fonction d'une seule variable

❖ Sources

[PDF] Le LASSO -
Université de Lille
math - 1

[PDF] Régression
pénalisée : le Lasso
pbil - 2

Lasso (statistiques) -
Wikipédia
fr.wikipedia - 3

View 5 more

🌀 Perplexity

Pour expliquer l'algorithme component-wise (ou par coordonnées) pour résoudre le problème de la régression Lasso, commençons par réécrire le problème d'optimisation du Lasso comme une fonction d'une seule variable.

Réécriture du problème Lasso

Le problème d'optimisation du Lasso s'écrit généralement sous la forme suivante :

$$\min_{\beta} \frac{1}{2n} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

où Y est le vecteur des observations, X la matrice des variables explicatives, β le vecteur des coefficients à estimer, n le nombre d'observations, et λ le paramètre de régularisation ①.

Pour réécrire ce problème en fonction d'une seule variable, nous allons nous concentrer sur un coefficient β_j à la fois, en considérant tous les autres coefficients comme fixes. Cela nous donne :

$$\min_{\beta_j} \frac{1}{2n} \|r_j - x_j \beta_j\|_2^2 + \lambda |\beta_j|$$

où $r_j = Y - \sum_{k \neq j} x_k \beta_k$ est le résidu partiel (c'est-à-dire le résidu sans prendre en compte la j -ème variable), et x_j est la j -ème colonne de X ②.

Algorithme component-wise

L'algorithme component-wise pour résoudre le problème Lasso fonctionne comme suit :

1. **Initialisation** : Commencer avec un vecteur initial $\beta^{(0)}$, généralement tous les coefficients sont mis à zéro.
2. **Itération** : Pour chaque itération t et pour chaque coefficient j :

Sklearn et le Lasso

https://scikit-learn.org/stable/

de Adele Drive cantine Galaxie_CS Eval2020_25Norma... Commitee_IA DARI Microsoft Word - m... retraite Gaëlle_J Firefox Monitor quentin les articles

 Install User Guide API Examples More ^

scikit-learn

Machine Learning in Python

Getting Started Release Highlights for 0.23 GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...

Clustering

Automatic grouping of similar objects

Applications: Customer segment comes

Algorithms: k-Means, spectral clustering

- 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

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

Elastic Net

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

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

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

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

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

Reweighted least square (Lasso and Ridge)

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

the idea : iterate the weighted ridge towards a fixed point

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

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

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

$W = \text{Identit }$

Tantque on n'a pas converg 

$$\beta = \arg \min \frac{1}{2} \|X\beta - y\|^2 + \lambda \beta^{\top} W \beta$$

$$W = \text{diag}(1/|\beta|)$$

fin de tantque

The adaptive Lasso

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

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

```
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