TP 7 : Le lasso par le gradient proximal (ISTA et FISTA)

Stéphane Canu

2 novembre 2026, MLA, ITI, INSA Rouen

Le but du TP est d'étudier la solution du Lasso fournie par l'algorithme du gradient proximal et par sa version accélérée[1]. En d'autres termes, construire un algorithme simple, le comparer aux méthodes des semaines précédentes et comprendre ce qui règle sa vitesse de convergence.

On s'intéresse au problème du Lasso

$$\min_{\beta\in\mathbb{R}^p} J_\lambda(\beta) = \tfrac{1}{2}\|X\beta - y\|^2 + \lambda\|\beta\|_1 \tag{1}$$

avec $\lambda = 10^{-3}n$.

Ex. 1 — Le gradient proximal (ISTA)

1. Reprendre les données simulées du TP 2 et vérifier que $\sum_{i=1}^n X_{ij}^2 = 1$, $j = 1, \ldots, p$.

```
import numpy as np
import matplotlib.pyplot as plt
import time

n = 200 # number of examples
p = 2*n # dimensionality of the problem
k = 5 # number of active variables

np.random.seed(0)
X = np.random.randn(n,p) # creating features and normalizing them
X = (X - np.mean(X,axis = 0))/np.std(X,axis = 0)

t = np.arange(0,p)/(p-1) # building the variance matrix
S = np.zeros((p,p))
nn = 0.00001
for i in range(p):
    S[i,:] = np.exp(-(t-t[i])**2/nn)
X = X@(S**.5)
X = X/np.linalg.norm(X,axis=0)

ind = np.random.choice(p, k, replace=False) # generating optimal weights
weights = np.random.randn(k)
weights += 0.1+np.sign(weights) # to get large enough weights
wopt = np.zeros(p)
wopt[ind] = weights

rsnr = 2 # generating output by X@w + noise
z = X[:,ind]@weights
stdnoise = np.std(z)/rsnr
y = z + stdnoise*np.random.randn(n)
```

2. L'opérateur proximal de la norme 1

   a) Montrez que

$$\mathbf{prox}_{\lambda\|\cdot\|_1}(w) = \arg\min_{u\in\mathbb{R}^p} \tfrac{1}{2}\|u - w\|^2 + \lambda\|u\|_1$$

   se calcule composante par composante :

$$\left(\mathbf{prox}_{\lambda\|\cdot\|_1}(w)\right)_j = \operatorname{sign}(w_j)\max(|w_j| - \lambda, 0), \qquad j = 1, \ldots, p.$$

   b) Écrire une fonction `Prox_L1`(w, λ) qui calcule cet opérateur.

[1] https://epubs.siam.org/doi/10.1137/080716542

```python
def Prox_L1(w, lam):
    v = # A COMPLETER
    return v
```

c) Représentez graphiquement avec `matplotlib` $\mathbf{prox}_{\lambda|\cdot|}(w)$ en fonction de $w \in [-3, 3]$ pour $\lambda = 1$. Pourquoi parle-t-on de seuillage doux ?

3. L'algorithme ISTA (Iterative Shrinkage-Thresholding Algorithm)

   $\beta = 0$

   Tant que on n'a pas convergé

   $g = X^\top(X\beta - y)$

   $\beta = \mathbf{prox}_{\rho\lambda\|\cdot\|_1}(\beta - \rho\, g)$

   fin de tant que

   a) Montrez que le gradient de f est L-lipschitzien avec $L = \|X\|_2^2$, la plus grande valeur propre de $X^\top X$. Calculez L pour nos données.

   b) Codez la fonction `ISTA`$(X, y, \lambda, \rho,$`nb_iter`$)$ avec le pas $\rho = 1/L$. Elle renverra la solution et la suite des valeurs $J_\lambda(\beta_k)$.

   c) Représentez $J_\lambda(\beta_k) - J_\lambda^\star$ en fonction de k en échelle logarithmique, où $J_\lambda^\star$ est la valeur optimale obtenue avec cvxpy au TP 3.

   d) Vérifiez numériquement que la solution obtenue est optimale, c'est-à-dire que $0 \in \partial J_\lambda(\beta)$, avec la fonction de vérification écrite au TP 3.

   e) Comparez avec le fonction ISTA produite par votre IA préférée (et donnez votre prompt).

Ex. 2 — Comparaison des méthodes de résultation du Lasso

1. Proposez des expériences permettant de comparer différents algorithmes pour résoudre le lasso (Sklearn, programmation quadratique (cvcpy), chemin de régularisation, component wise et gradient proximal...).
2. En fonction du résultats de vos expériences, proposez des recommandations sur l'usage de ces algorithmes.
3. Quelles autres méthodes pour obtenir la solutions existe-t-il (donnée le ou les prompts utilisés, laes réponse de l'IA et votre analyse de cet échange).

Ex. 3 — Extra balle : L'accélération de Nesterov (FISTA)

1. L'algorithme FISTA

   $\beta = \gamma = 0, \quad s = 1$

   Tant que on n'a pas convergé

   $\beta^+ = \mathbf{prox}_{\rho\lambda\|\cdot\|_1}(\gamma - \rho\, X^\top(X\gamma - y))$

   $s^+ = \frac{1}{2}(1 + \sqrt{1 + 4s^2})$

   $\gamma = \beta^+ + \frac{s-1}{s^+}(\beta^+ - \beta), \quad \beta = \beta^+, \quad s = s^+$

   fin de tant que

   a) Codez la fonction `FISTA`$(X, y, \lambda, \rho,$`nb_iter`$)$ et ajoutez sa courbe de convergence sur la figure précédente.

   b) La suite $J_\lambda(\beta_k)$ est-elle toujours décroissante ? Comment l'expliquer ?

   c) Résoudre, à l'aide d'une IA générative, le problème (1) par FISTA. Vous donnerez le prompt et le LLM que vous avez utilisés. Le code obtenu est-il correct ? Est-il plus rapide que le vôtre ?

2. Influences

   a) Quelle est l'influence du pas ρ sur la vitesse de convergence ? Essayez $\rho = 0.5/L$, $1/L$, $2/L$, $3/L$ et $4/L$. La théorie ne garantit la convergence que pour $\rho < 2/L$: qu'observez-vous au-delà ? Comment l'expliquer ?

b) Quelle est l'influence de la matrice X ? Reprenez les données sans la multiplication par $S^{1/2}$ (variables non corrélées) et comparez les vitesses de convergence.