

# Régression fused Lasso adaptative et application aux données de microarrays CGH

Mohamad GHASSANY

Responsable :  
Sophie Lambert-Lacroix (LJK - Grenoble)

28 juin 2010

# Plan de la présentation

- 1 Régression fused Lasso adaptative
- 2 Algorithme de type “path”
- 3 Résultats numériques
- 4 Conclusion et Perspectives

# Plan de la présentation

- 1 Régression fused Lasso adaptative
- 2 Algorithme de type “path”
- 3 Résultats numériques
- 4 Conclusion et Perspectives

# Régression linéaire

Soit le modèle de régression linéaire standard

$$y_i = \alpha^* + \mathbf{x}_i^T \beta^* + \epsilon_i, \quad i = 1, \dots, n.$$

Où

- $y = (y_1, \dots, y_n)^T$  est le vecteur des réponses.

- 

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \cdots & \mathbf{x}_{1,p} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \cdots & \mathbf{x}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{n,1} & \mathbf{x}_{n,2} & \cdots & \mathbf{x}_{n,p} \end{pmatrix}$$

est la matrice des régresseurs (matrice du plan d'expérience)

- $\beta^* = (\beta_1^*, \dots, \beta_p^*)^T$  sont les coefficients à estimer.
- $\alpha^*$  est une constante (nul si les  $\mathbf{x}_i$  sont standardisés).
- $\epsilon = (\epsilon_1, \dots, \epsilon_n)^T$  le vecteur des résidus.

- **But:** Expliquer, avec le plus de précision possible, les valeurs prises par  $y_i$  à partir d'une série de variables explicatives  $x_{i1}, \dots, x_{ip}$ .

⇒ Déterminer une estimation des valeurs de  $\beta^*$ .

- **But:** Expliquer, avec le plus de précision possible, les valeurs prises par  $y_i$  à partir d'une série de variables explicatives  $x_{i1}, \dots, x_{ip}$ .

⇒ Déterminer une estimation des valeurs de  $\beta^*$ .

### Méthode de moindres carrés ordinaires MCO:

Soit  $\beta$  le vecteur des estimateurs des  $\beta^*$ ,

$$\begin{aligned} f(\beta) &= (y - \mathbf{X}\beta)^T (y - \mathbf{X}\beta) \\ &= \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij}\beta_j \right)^2 \end{aligned}$$

# Pénalité Lasso

## Limites de l'estimation par MCO:

- L'exactitude des prévision, biais faible et variance large.
- Si  $n \geq p$  la méthode n'est pas performante.

# Pénalité Lasso

## Limites de l'estimation par MCO:

- L'exactitude des prévision, biais faible et variance large.
- Si  $n \geq p$  la méthode n'est pas performante.

**Solution:** Fonction de coût sous contraintes.

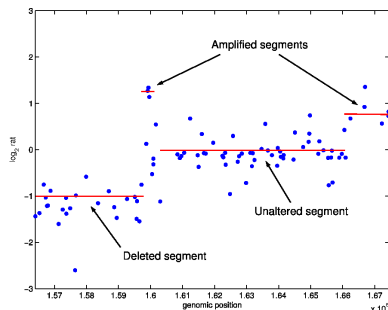
⇒ Pénalité de type  $\ell_1$ : le Lasso (least absolute shrinkage and selection operator).

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p |\beta_j|$$



# Les données microarrays CGH

- Les données de microarrays CGH permettent de décrire l'enchainement du nombre de copies des gènes le long des chromosomes.

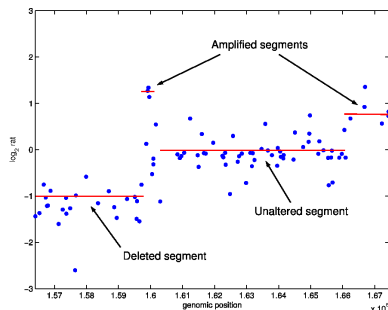


Exemple de profil CGH

# Les données microarrays CGH

- Les données de microarrays CGH permettent de décrire l'enchainement du nombre de copies des gènes le long des chromosomes.

⇒ En plus, il faut prendre en compte l'ordre de ces variables.



Exemple de profil CGH

# Pénalité *fused* Lasso

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}|$$

- $\lambda_1 \geq 0$  force des coefficients à être nuls.
- $\lambda_2 \geq 0$  force deux coefficients consécutives à être égaux.

# Propriété oraculaire

Soit:

- $\hat{\beta}(\delta)$  l'estimateur de  $\beta^*$  par une procédure  $\delta$ .
- $\mathcal{A} = \{1 \leq j \leq p, \beta_j^* \neq 0\}$ .
- $\mathcal{A}^* = \{j : \hat{\beta}_j(\delta) \neq 0\}$ .

# Propriété oraculaire

Soit:

- $\hat{\beta}(\delta)$  l'estimateur de  $\beta^*$  par une procédure  $\delta$ .
- $\mathcal{A} = \{1 \leq j \leq p, \beta_j^* \neq 0\}$ .
- $\mathcal{A}^* = \{j : \hat{\beta}_j(\delta) \neq 0\}$ .

$\hat{\beta}(\delta)$  satisfait la propriété oraculaire si:

- $\hat{\beta}(\delta)$  est consistant en sélection de variables:  $\lim_n \mathbb{P}(\mathcal{A}^* = \mathcal{A}) = 1$ .

## Propriété oraculaire

Soit:

- $\hat{\beta}(\delta)$  l'estimateur de  $\beta^*$  par une procédure  $\delta$ .
- $\mathcal{A} = \{1 \leq j \leq p, \beta_j^* \neq 0\}$ .
- $\mathcal{A}^* = \{j : \hat{\beta}_j(\delta) \neq 0\}$ .

$\hat{\beta}(\delta)$  satisfait la propriété oraculaire si:

- $\hat{\beta}(\delta)$  est consistant en sélection de variables:  $\lim_n \mathbb{P}(\mathcal{A}^* = \mathcal{A}) = 1$ .

Le Lasso ne satisfait pas cette propriété.

$\implies$  Ajouter des poids dans le modèle.

# Lasso adaptative et fused Lasso adaptative

Lasso adaptative:

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p w_j |\beta_j|$$

Où  $w_j = |\hat{\beta}_j^{mco}|^{-\gamma}$ ,  $\gamma \geq 0$ .

# Lasso adaptative et fused Lasso adaptative

Lasso adaptative:

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p w_j |\beta_j|$$

Où  $w_j = |\hat{\beta}_j^{mco}|^{-\gamma}$ ,  $\gamma \geq 0$ .

Fused Lasso adaptative:

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n \left( y_i - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda_1 \sum_{j=1}^p w_j^{(1)} |\beta_j| + \lambda_2 \sum_{j=1}^{p-1} w_j^{(2)} |\beta_j - \beta_{j+1}|$$

Où  $w_j^{(1)} = |\hat{\beta}_j^{mco}|^{-\gamma}$ ,  $j = 1, \dots, p$  et  $w_j^{(2)} = |\hat{\beta}_j^{mco} - \hat{\beta}_{j+1}^{mco}|^{-\gamma}$ ,  $j = 1, \dots, p-1$



## FLSA et A-FLSA

Selon le type des données microarrays CGH,  $\mathbf{X} = \mathbf{I}$  et  $n = p$ .

## FLSA et A-FLSA

Selon le type des données microarrays CGH,  $\mathbf{X} = \mathbf{I}$  et  $n = p$ .

Fused Lasso Signal Approximator (FLSA):

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n (y_i - \beta_i)^2 + \lambda_1 \sum_{i=1}^n |\beta_i| + \lambda_2 \sum_{i=1}^{n-1} |\beta_i - \beta_{i+1}|.$$

## FLSA et A-FLSA

Selon le type des données microarrays CGH,  $\mathbf{X} = \mathbf{I}$  et  $n = p$ .

Fused Lasso Signal Approximator (FLSA):

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n (y_i - \beta_i)^2 + \lambda_1 \sum_{i=1}^n |\beta_i| + \lambda_2 \sum_{i=1}^{n-1} |\beta_i - \beta_{i+1}|.$$

Adaptive Fused Lasso Signal Approximator (A-FLSA):

$$f(\beta) = \frac{1}{2} \sum_{i=1}^n (y_i - \beta_i)^2 + \lambda_1 \sum_{i=1}^n w_i^{(1)} |\beta_i| + \lambda_2 \sum_{i=1}^{n-1} w_i^{(2)} |\beta_i - \beta_{i+1}|,$$

Où  $w_j^{(1)} = |\hat{\beta}_j^{mco}|^{-\gamma}$ ,  $j = 1, \dots, n$  et  $w_j^{(2)} = |\hat{\beta}_j^{mco} - \hat{\beta}_{j+1}^{mco}|^{-\gamma}$ ,  $j = 1, \dots, n-1$

# Plan de la présentation

- 1 Régression fused Lasso adaptative
- 2 Algorithme de type "path"**
- 3 Résultats numériques
- 4 Conclusion et Perspectives

# Algorithme "path"

Avantage des algorithmes de type "path":

Trouver la solution pour n'importe quel valeur de  $\lambda$ .

# Algorithme "path"

Avantage des algorithmes de type "path":

Trouver la solution pour n'importe quel valeur de  $\lambda$ .

Soit à trouver

$$\hat{\beta}(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} L(y, X\beta) + \lambda J(\beta),$$

Où:

- $L: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ .
- $J: \mathbb{R}^p \rightarrow \mathbb{R}$ , avec  $J(0) = 0$ .

## Algorithme "path"

Conditions:

- $\frac{\partial \hat{\beta}(\lambda)}{\partial \lambda} / \left\| \frac{\partial \hat{\beta}(\lambda)}{\partial \lambda} \right\|$  est constante par morceaux.
- $L$  est quadratique par morceaux en fonction de  $\beta$  le long du chemin optimal  $\hat{\beta}(\lambda)$ .
- $J$  est linéaire par morceaux en fonction de  $\beta$  le long du chemin.

## Algorithme "path"

Conditions:

- $\frac{\partial \hat{\beta}(\lambda)}{\partial \lambda} / \left\| \frac{\partial \hat{\beta}(\lambda)}{\partial \lambda} \right\|$  est constante par morceaux.
- $L$  est quadratique par morceaux en fonction de  $\beta$  le long du chemin optimal  $\hat{\beta}(\lambda)$ .
- $J$  est linéaire par morceaux en fonction de  $\beta$  le long du chemin.

$\hat{\beta}(\lambda)$  est linéaire par morceaux en fonction de  $\lambda$ ,

$\implies$

Il existe:

- $\lambda^{(0)} = 0 < \lambda^{(1)} < \dots < \lambda^{(m)} = \infty$ .
- $\gamma_0, \gamma_1, \dots, \gamma_{m-1} \in \mathbb{R}^p$ .

tels que,

- $\hat{\beta}(\lambda) = \hat{\beta}(\lambda^{(k)}) + (\lambda - \lambda^{(k)})\gamma_k$  pour  $\lambda^{(k)} \leq \lambda \leq \lambda^{(k+1)}$ .



A-FLSA:

$$\hat{\beta}(\lambda_1, \lambda_2) = \operatorname{argmin}_{\beta \in \mathbb{R}^p} L(y, X\beta) + \lambda_1 J_1(\beta) + \lambda_2 J_2(\beta).$$

## A-FLSA:

$$\hat{\beta}(\lambda_1, \lambda_2) = \operatorname{argmin}_{\beta \in \mathbb{R}^p} L(y, X\beta) + \lambda_1 J_1(\beta) + \lambda_2 J_2(\beta).$$

## Théorème 1:

Soit  $\hat{\beta}(0, \lambda_2)$  la solution pour  $\lambda_1 = 0$  et  $\lambda_2 > 0$ . Alors la solution pour  $\lambda_1 > 0$  est

$$\begin{aligned} \hat{\beta}_i(\lambda_1, \lambda_2) &= S(\hat{\beta}_i(0, \lambda_2), w_1^{(i)} \lambda_1) \text{ pour } i = 1, \dots, n \\ &= \operatorname{sign}(\hat{\beta}_i(0, \lambda_2)) (|\hat{\beta}_i(0, \lambda_2)| - w_1^{(i)} \lambda_1)^+ \end{aligned}$$

où  $y^+ = \max(0, y)$ .

## A-FLSA:

$$\hat{\beta}(\lambda_1, \lambda_2) = \operatorname{argmin}_{\beta \in \mathbb{R}^p} L(y, X\beta) + \lambda_1 J_1(\beta) + \lambda_2 J_2(\beta).$$

## Théorème 1:

Soit  $\hat{\beta}(0, \lambda_2)$  la solution pour  $\lambda_1 = 0$  et  $\lambda_2 > 0$ . Alors la solution pour  $\lambda_1 > 0$  est

$$\begin{aligned} \hat{\beta}_i(\lambda_1, \lambda_2) &= S(\hat{\beta}_i(0, \lambda_2), w_1^{(i)} \lambda_1) \text{ pour } i = 1, \dots, n \\ &= \operatorname{sign}(\hat{\beta}_i(0, \lambda_2)) (|\hat{\beta}_i(0, \lambda_2)| - w_1^{(i)} \lambda_1)^+ \end{aligned}$$

où  $y^+ = \max(0, y)$ .

## Fonction de coût:

$$L(y, \beta) = \frac{1}{2} \sum_{i=1}^n (y_i - \beta_i)^2 + \lambda_2 \sum_{i=1}^{n-1} w_i^{(2)} |\beta_i - \beta_{i+1}|,$$

## Théorème 2:

Soit  $\beta_k(0, \lambda_2)$  la solution optimale pour un coefficient  $k$  et un certain  $\lambda_2$ .  
Si pour un certain  $k$  et  $\lambda_2^0$  on a  $\beta_k(0, \lambda_2^0) = \beta_{k+1}(0, \lambda_2^0)$ , alors  $\forall \lambda_2 > \lambda_2^0$ ,  
 $\beta_k(0, \lambda_2) = \beta_{k+1}(0, \lambda_2)$ .

## Théorème 2:

Soit  $\beta_k(0, \lambda_2)$  la solution optimale pour un coefficient  $k$  et un certain  $\lambda_2$ .  
Si pour un certain  $k$  et  $\lambda_2^0$  on a  $\beta_k(0, \lambda_2^0) = \beta_{k+1}(0, \lambda_2^0)$ , alors  $\forall \lambda_2 > \lambda_2^0$ ,  
 $\beta_k(0, \lambda_2) = \beta_{k+1}(0, \lambda_2)$ .

## Notations:

Soit  $F_i$ ,  $i = 1, \dots, n_F(\lambda_2)$ , les ensembles des indices des coefficients fusionnés pour  $\lambda_2$ .

- De la forme:  $F_i = \{k | l_i \leq k \leq u_i\}$
- $\cup_{i=1}^{n_F(\lambda_2)} F_i = 1, \dots, n$ .
- $F_i \cap F_j = \emptyset$ ,  $i \neq j$ .
- $\forall k, l \in F_i$  on a  $\beta_k(\lambda_2) = \beta_l(\lambda_2)$ .
- Si  $k \in F_i, l \in F_{i+1}$ ,  $\beta_k(\lambda_2) \neq \beta_l(\lambda_2)$ .

On écrit  $\beta_{F_i}(\lambda_2)$  pour tout  $\beta_k(\lambda_2)$  avec  $k \in F_i$ .

La fonction de coût devient:

$$L_{F,\lambda_2}(y, \beta) = \frac{1}{2} \sum_{i=1}^{n_F(\lambda_2)} \left( \sum_{j \in F_i} (y_j - \beta_{F_i})^2 \right) + \lambda_2 \sum_{i=1}^{n_F(\lambda_2)-1} \frac{|\beta_{F_i} - \beta_{F_{i+1}}|}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

La fonction de coût devient:

$$L_{F,\lambda_2}(y, \beta) = \frac{1}{2} \sum_{i=1}^{n_F(\lambda_2)} \left( \sum_{j \in F_i} (y_j - \beta_{F_i})^2 \right) + \lambda_2 \sum_{i=1}^{n_F(\lambda_2)-1} \frac{|\beta_{F_i} - \beta_{F_{i+1}}|}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

$$\frac{\partial L_{F,\lambda_2}}{\partial \beta_{F_i}} = (\#F_i)\beta_{F_i} - \sum_{j \in F_i} y_j + \lambda_2 \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i-1}})}{|\hat{\beta}_{\max F_{i-1}} - \hat{\beta}_{\min F_i}|} + \lambda_2 \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i+1}})}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

La fonction de coût devient:

$$L_{F,\lambda_2}(y, \beta) = \frac{1}{2} \sum_{i=1}^{n_F(\lambda_2)} \left( \sum_{j \in F_i} (y_j - \beta_{F_i})^2 \right) + \lambda_2 \sum_{i=1}^{n_F(\lambda_2)-1} \frac{|\beta_{F_i} - \beta_{F_{i+1}}|}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

$$\frac{\partial L_{F,\lambda_2}}{\partial \beta_{F_i}} = (\#F_i)\beta_{F_i} - \sum_{j \in F_i} y_j + \lambda_2 \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i-1}})}{|\hat{\beta}_{\max F_{i-1}} - \hat{\beta}_{\min F_i}|} + \lambda_2 \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i+1}})}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

$$\beta_{F_i}(\lambda_2) = \frac{1}{\#F_i} \left[ \sum_{j \in F_i} y_j - \lambda_2 \left( \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i-1}})}{|\hat{\beta}_{\max F_{i-1}} - \hat{\beta}_{\min F_i}|} + \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i+1}})}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|} \right) \right].$$



La fonction de coût devient:

$$L_{F, \lambda_2}(y, \beta) = \frac{1}{2} \sum_{i=1}^{n_F(\lambda_2)} \left( \sum_{j \in F_i} (y_j - \beta_{F_i})^2 \right) + \lambda_2 \sum_{i=1}^{n_F(\lambda_2)-1} \frac{|\beta_{F_i} - \beta_{F_{i+1}}|}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

$$\frac{\partial L_{F, \lambda_2}}{\partial \beta_{F_i}} = (\#F_i) \beta_{F_i} - \sum_{j \in F_i} y_j + \lambda_2 \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i-1}})}{|\hat{\beta}_{\max F_{i-1}} - \hat{\beta}_{\min F_i}|} + \lambda_2 \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i+1}})}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|}.$$

$$\beta_{F_i}(\lambda_2) = \frac{1}{\#F_i} \left[ \sum_{j \in F_i} y_j - \lambda_2 \left( \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i-1}})}{|\hat{\beta}_{\max F_{i-1}} - \hat{\beta}_{\min F_i}|} + \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i+1}})}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|} \right) \right].$$

$$\frac{\partial \beta_{F_i}}{\partial \lambda_2} = -\frac{1}{\#F_i} \left( \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i-1}})}{|\hat{\beta}_{\max F_{i-1}} - \hat{\beta}_{\min F_i}|} + \frac{\text{sign}(\beta_{F_i} - \beta_{F_{i+1}})}{|\hat{\beta}_{\max F_i} - \hat{\beta}_{\min F_{i+1}}|} \right).$$

$$h_{i,i+1}(\lambda_2) = \frac{\beta_{F_i}(\lambda_2) - \beta_{F_{i+1}}(\lambda_2)}{\frac{\partial \beta_{F_{i+1}}}{\partial \lambda_2} - \frac{\partial \beta_{F_i}}{\partial \lambda_2}} + \lambda_2 \text{ for } i = 1, \dots, n_F(\lambda_2) - 1.$$

$$h_{i,i+1}(\lambda_2) = \frac{\beta_{F_i}(\lambda_2) - \beta_{F_{i+1}}(\lambda_2)}{\frac{\partial \beta_{F_{i+1}}}{\partial \lambda_2} - \frac{\partial \beta_{F_i}}{\partial \lambda_2}} + \lambda_2 \text{ for } i = 1, \dots, n_F(\lambda_2) - 1.$$

Hitting time:

$$h(\lambda_2) = \min_{h_{i,i+1}(\lambda_2) > \lambda_2} h_{i,i+1}(\lambda_2).$$

**Algorithm 1:** A-FLSA path algorithm**begin**

$$\lambda_2 = 0;$$

$$\beta_k = y_k \text{ for } k = 1, \dots, n;$$

$$F_i = \{i\} \text{ for } i = 1, \dots, n;$$

$$n_F = n;$$

**end****while**  $n_F > 1$  **do**Calculate next hitting time  $h(\lambda_2)$ ;Let  $(i_0(\lambda_2), i_0(\lambda_2) + 1) = \operatorname{argmin}_{h_{i,i+1} > \lambda_2} h_{i,i+1}(\lambda_2)$  be the indices of the sets to be fuse next;Fuse the two sets  $F_{i_0(\lambda_2)}$  and  $F_{i_0(\lambda_2)+1}$ ;Set:  $\lambda_2 = h(\lambda_2)$ ;Update the values for  $\beta_k(\lambda_2)$ ,  $\frac{\partial \beta_k(\lambda_2)}{\partial \lambda_2}$  and set  $n_F = n_F - 1$  ;**end**

# Plan de la présentation

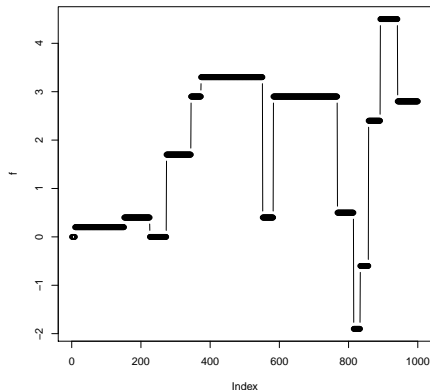
- 1 Régression fused Lasso adaptative
- 2 Algorithme de type “path”
- 3 Résultats numériques**
- 4 Conclusion et Perspectives

# Simulations

- $\mathbf{X} = \mathbf{I}$  et  $p = n$ ,

$$y_i = \beta_i^* + \epsilon_i \quad i = 1, \dots, n.$$

- $\epsilon_i \hookrightarrow \mathcal{N}(0, \sigma^2)$ .



Simulations de  $\beta^*$ ,  $n=1000$

# Simulations

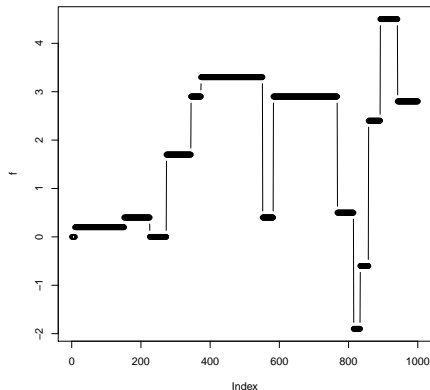
- $\mathbf{X} = \mathbf{I}$  et  $p = n$ ,

$$y_i = \beta_i^* + \epsilon_i \quad i = 1, \dots, n.$$

- $\epsilon_i \hookrightarrow \mathcal{N}(0, \sigma^2)$ .

$$\sigma^2 = \frac{1}{n} \frac{\|\beta^*\|_2^2}{\text{SNR}^2}.$$

- $\|\beta^*\|_2^2 = \sum_{i=1}^n (\beta_i^*)^2$ .
- $\text{SNR} = 3, 5$  ou  $7$ .



Simulations de  $\beta^*$ ,  $n=1000$

## Validation croisée

- **But:** Trouver  $\lambda_1$  et  $\lambda_2$  pour lesquels  $\hat{\beta}$  est optimale.
- **Procédure:**

$$y^{(T)} = \{y_i; i = 2k; k = 1, \dots, (\frac{n}{2})\}.$$

$$y^{(L)} = \{y_i; i = 2k + 1; k = 0, \dots, (\frac{n}{2} - 1)\} \implies \hat{\beta}_{i_{\lambda_1, \lambda_2}}$$



## Validation croisée

- **But:** Trouver  $\lambda_1$  et  $\lambda_2$  pour lesquels  $\hat{\beta}$  est optimale.
- **Procédure:**

$$y^{(T)} = \{y_i; i = 2k; k = 1, \dots, (\frac{n}{2})\}.$$

$$y^{(L)} = \{y_i; i = 2k + 1; k = 0, \dots, (\frac{n}{2} - 1)\} \implies \hat{\beta}_{i_{\lambda_1, \lambda_2}}$$

$$MSE_{(\lambda_1, \lambda_2)} = \sum_{i=1}^{\frac{n}{2}} (y_i^{(T)} - \hat{\beta}_{i_{\lambda_1, \lambda_2}}^{(L)})^2 \quad \text{pour tout pair } (\lambda_1, \lambda_2).$$

## Validation croisée

- **But:** Trouver  $\lambda_1$  et  $\lambda_2$  pour lesquels  $\hat{\beta}$  est optimale.
- **Procédure:**

$$y^{(T)} = \{y_i; i = 2k; k = 1, \dots, (\frac{n}{2})\}.$$

$$y^{(L)} = \{y_i; i = 2k + 1; k = 0, \dots, (\frac{n}{2} - 1)\} \implies \hat{\beta}_{i_{\lambda_1, \lambda_2}}$$

$$MSE_{(\lambda_1, \lambda_2)} = \sum_{i=1}^{\frac{n}{2}} (y_i^{(T)} - \hat{\beta}_{i_{\lambda_1, \lambda_2}}^{(L)})^2 \quad \text{pour tout pair } (\lambda_1, \lambda_2).$$

$$(\lambda_1, \lambda_2) = \operatorname{argmin}_{(\lambda_1, \lambda_2)} MSE.$$

# False Discovery Rate

- **But:** Comparer la performance de plusieurs méthodes.

$$\text{False discovery rate} \left\{ \begin{array}{l} FDR_i = \frac{FP_i}{FP_i + TP_i} \quad i = 1, 2 \end{array} \right.$$

# False Discovery Rate

- **But:** Comparer la performance de plusieurs méthodes.

$$\text{False discovery rate} \left\{ \begin{aligned} FDR_i &= \frac{FP_i}{FP_i + TP_i} \quad i = 1, 2 \end{aligned} \right.$$

$$\text{False positives} \left\{ \begin{aligned} FP_1 &= \#\{j : \beta_j^* = 0 \ \& \ \hat{\beta}_j \neq 0\} \\ FP_2 &= \#\{j : \beta_j^* = \beta_{j+1}^* \ \& \ \hat{\beta}_j \neq \hat{\beta}_{j+1}\} \end{aligned} \right.$$

$$\text{True positives} \left\{ \begin{aligned} TP_1 &= \#\{j : \beta_j^* \neq 0 \ \& \ \hat{\beta}_j \neq 0\} \\ TP_2 &= \#\{j : \beta_j^* \neq \beta_{j+1}^* \ \& \ \hat{\beta}_j \neq \hat{\beta}_{j+1}\} \end{aligned} \right.$$

# False Negative Rate

- **But:** Comparer la performance de plusieurs méthodes.

$$\text{False negative rate} \left\{ \begin{array}{l} FNR_i = \frac{FN_i}{FN_i + TN_i} \quad i = 1, 2 \end{array} \right.$$

# False Negative Rate

- **But:** Comparer la performance de plusieurs méthodes.

$$\text{False negative rate} \left\{ \begin{aligned} FNR_i &= \frac{FN_i}{FN_i + TN_i} \quad i = 1, 2 \end{aligned} \right.$$

$$\text{False negatives} \left\{ \begin{aligned} FN_1 &= \#\{j : \beta_j^* \neq 0 \ \& \ \hat{\beta}_j = 0\} \\ FN_2 &= \#\{j : \beta_j^* \neq \beta_{j+1}^* \ \& \ \hat{\beta}_j = \hat{\beta}_{j+1}\} \end{aligned} \right.$$

$$\text{True negatives} \left\{ \begin{aligned} TN_1 &= \#\{j : \beta_j^* = 0 \ \& \ \hat{\beta}_j = 0\} \\ TN_2 &= \#\{j : \beta_j^* = \beta_{j+1}^* \ \& \ \hat{\beta}_j = \hat{\beta}_{j+1}\} \end{aligned} \right.$$

# Mean Squared Error

$$MSE(\hat{\beta}_k) = \text{Var}(\hat{\beta}_k) + (\text{Bias}(\hat{\beta}_k, \beta_k^*))^2 \quad \text{pour } k = 1, \dots, n.$$

# Mean Squared Error

$$MSE(\hat{\beta}_k) = \text{Var}(\hat{\beta}_k) + (\text{Bias}(\hat{\beta}_k, \beta_k^*))^2 \quad \text{pour } k = 1, \dots, n.$$

$$\text{Var}(\hat{\beta}_k) = \frac{1}{Nr} \sum_{nr=1}^{Nr} (\hat{\beta}_k^{nr})^2 - \frac{1}{Nr} \left( \sum_{nr=1}^{Nr} \hat{\beta}_k^{nr} \right)^2 \quad \text{pour } k = 1, \dots, n.$$

$$(\text{Bias}(\hat{\beta}_k, \beta_k^*))^2 = \frac{1}{Nr} \sum_{nr=1}^{Nr} (\hat{\beta}_k^{nr} - \beta_k^*)^2 \quad \text{pour } k = 1, \dots, n.$$



# Comparaison

Table: Comparaison des résultats,  $n=100$ .

	A-FLSA			FLSA		
SNR	7	5	3	7	5	3
FDR 1	0.03483	0.04457	0.05687	0.06454	0.06647	0.06634
FDR 2	0.48719	0.44719	0.48073	0.42865	0.48213	0.55175
FNR 1	0.3901	0.04457	0.24545	0.10625	0.06647	0.15863
FNR 2	0.02848	0.44719	0.04237	0.02735	0.48213	0.03863
$\bar{\lambda}_1$	0.0642	0.0538	0.0538	0.0244	0.0268	0.0458
$\bar{\lambda}_2$	0.634	0.782	1.062	0.5016	0.5404	0.6432
MSE	0.05323	0.07095	0.13653	0.03454	0.05153	0.11681

# Comparaison

Table: Comparaison des résultats,  $n=1000$ .

	A-FLSA			FLSA		
SNR	7	5	3	7	5	3
FDR 1	0.05689	0.05504	0.05029	0.05589	0.05426	0.05195
FDR 2	0.69368	0.73686	0.81771	0.83268	0.83230	0.82899
FNR 1	0.00146	0.05504	0.11707	0.00301	0.05426	0.07479
FNR 2	0.00344	0.73686	0.00542	0.00269	0.83230	0.00465
$\bar{\lambda}_1$	0.0002	0.0038	0.0202	0.0028	0.0086	0.0226
$\bar{\lambda}_2$	1.1432	1.9828	4.3256	1.1104	1.5324	2.5496
MSE	0.01173	0.02679	0.08950	0.01012	0.01824	0.04351

# Plan de la présentation

- 1 Régression fused Lasso adaptative
- 2 Algorithme de type “path”
- 3 Résultats numériques
- 4 Conclusion et Perspectives

# Conclusion et perspectives

- Résultats presque équivalents en cas de  $\mathbf{X} = \mathbf{I}$  et  $n = p$ .
- Critère du choix de  $\lambda_1$  et  $\lambda_2$ .
- Possibilité de pénaliser le premier terme seulement.
- Adapter un algorithme pour le cas de matrice  $\mathbf{X}$  quelconque.
- Application sur des données réelles.
- Démontrer les propriétés théoriques.

Merci de votre attention