E.I.S.T.I - Département Mathématiques
$2^{\text{ème}}$ année Ingénieurs - Mathématiques & Informatique

Compressive Sensing

Algorithmes de reconstruction

Prof : Nisrine Fortin

---

1. Least Absolute Shrinkage and Selection Operator LASSO

Soient $y \in \mathbb{R}^m$, $\alpha \in \mathbb{R}^N$ et $A \in \mathbb{R}^{m \times N}$. On notera

$$A = \big[A_1, A_2, \ldots, A_N\big];$$

A_j désigne une colonne de A.
On cherche à expliquer de manière linéaire la variable y par N variables potentiellement explicatives A_j :

$$\mathbf{y} = \mathbf{A}\alpha + \epsilon$$

où $\epsilon \in \mathbb{R}^m$ est un vecteur de m variables aléatoires i.i.d de moyenne nulle et de variance σ^2.
ϵ correspond au bruit des observations. Les variables A_j n'étant pas toutes pertinentes, l'objectif du LASSO est d'éliminer les variables inutiles et uniquement celles-ci. La méthode LASSO ne fait pas une régression linéaire classique mais une régression régularisée qui rend nuls certains coefficients de l'estimation de α.
Problème LASSO : Estimer pour $\lambda \in \overline{\mathbb{R}}_+$:

$$\widehat{\alpha}(\lambda) = \mathbf{arg}\min_{\alpha \in \mathbb{R}^{\mathbf{N}}} \left(\frac{\mathbf{1}}{\mathbf{2}} \Big\| \mathbf{y} - \mathbf{A}\alpha \Big\|_{\mathbf{2}}^{\mathbf{2}} + \lambda \Big\| \alpha \Big\|_{\mathbf{1}} \right).$$

$\big\|x\big\|_2^2 = \sum_{i=1}^{N} x_i^2$ et $\big\|x\big\|_1 = \sum_{i=1}^{N} |x_i|$.

Le paramètre λ

Le paramètre $\lambda \geq 0$ contrôle la puissance de la régularisation :

- Si $\lambda = 0$ et $N \ll m$, on obtient une régression linéaire classique.
- Si $\lambda = +\infty$, tous les coefficients de $\widehat{\alpha}$ sont nuls.

L'augmentation de λ induit la diminution de certains coefficients de $\widehat{\alpha}(\lambda)$ vers 0.

1.2 Algorithme et conditions d'optimalité et d'unicité

1. L'algorithme du LASSO consiste à déterminer $\widehat{\alpha}(\lambda)$ pour tout $\lambda \geq 0$.
2. Il s'agit ensuite de déterminer le bon λ qui permet de garder uniquement les variables explicatives et d'éliminer les autres.

La problématique

Déterminer pour tout $\lambda > 0$:

$$\widehat{\alpha}(\lambda) = \mathbf{arg}\min_{\alpha \in \mathbb{R}^{\mathbf{N}}} \left(\frac{\mathbf{1}}{\mathbf{2}} \left\| \mathbf{y} - \mathbf{A}\alpha \right\|_{\mathbf{2}}^{\mathbf{2}} + \lambda \left\| \alpha \right\|_{\mathbf{1}} \right). \tag{1}$$

Lemme 1. ***Unicité :*** *Si $A^T A$ est définie positive alors (1) admet une solution unique.*

Exercice 1. *Démontrer le lemme d'unicité de la solution de (1).*

Lemme 2. ***Conditions d'optimalité :*** *α est solution de (1) si et seulement si $\forall j \in [\![1; N]\!]$*

$$\begin{cases} \left| A_j^T (y - A\alpha) \right| \leq \lambda & si \quad \alpha_j = 0 \quad (L_1) \\ \\ A_j^T (y - A\alpha) = \lambda signe(\alpha_j) & si \quad \alpha_j \neq 0 \quad (L_2) \end{cases}$$

Exercice 2. *Démontrer le lemme des conditions d'optimalité du problème (1).*

En supposant connu le signe s d'une solution du problème LASSO, une solution potentielle $\widehat{\alpha}^s(\lambda)$ est donc déduite :

$$\begin{cases} \alpha_j^s(\lambda) = 0 & \text{si} \quad s(j) = 0 \\ \\ \alpha_{\Gamma_s}^s(\lambda) = \left(A_{\Gamma_s}^T A_{\Gamma_s} \right)^{-1} \left(A_{\Gamma_s}^T y - \lambda s_\Gamma \right) & \text{sinon} \end{cases}$$

où

- Γ_s est l'ensemble des indices j tels que $s(j) \neq 0$.
- A_{Γ_s} est la matrice composée des colonnes de la matrice A correspondant aux indices de Γ_s.
- s_Γ est le vecteur s auquel on a retiré les coordonnées nulles.

Méthode pour calculer s_i : Soit $i \geq 1$. Supposons connus λ_i et s_{i-1}.

■ On admet que pour $\lambda < \lambda_i$, la fonction $\alpha^{s_{i-1}}(\lambda)$ ne convient plus :

1. Soit signe$(\alpha^{s_{i-1}}(\lambda)) \neq s_{i-1}$. Conséquence : une coordonnée j change de signe, donc s'annule en $\lambda = \lambda_i$.

   On pose $\epsilon_i = 0$.
2. Soit la propriété L_1 n'est plus vérifiée pour un j tel que $s(j) = 0$.

   On pose $\epsilon_i = \text{signe}\left(A_{\Gamma_{s-1}}^T \left(y - A\alpha^{s_{i-1}}(\lambda_i) \right) \right)$

■ On définit alors $s_i = s_{i-1}$ en remplaçant la $j^{\text{ème}}$ coordonnée par ϵ_i.

1.3 Algorithme du LASSO

Pour $\lambda = +\infty$, la solution du Lasso est le vecteur nul. On initialise donc à partir de ces valeurs et on fait décroître λ.

- **Initialisation :** On pose $\lambda_0 = +\infty$, $s_0 = 0_{\mathbb{R}^N}$ et $\alpha^{s_0} = 0_{\mathbb{R}^N}$.

- **Etape 1 :** On fait décroître λ en partant de λ_0 jusqu'à ce qu'une coordonnée j de $\alpha^{s_0}(\lambda)$ ne vérifie plus la propriété L_1.
  - ⋆ On définit λ_1 la valeur de λ pour laquelle la propriété L_1 n'est plus satisfaite.
  - ⋆ La coordonnée j de $\alpha^{s_0}(\lambda)$ n'est donc plus nulle pour $\lambda < \lambda_1$.
  - ⋆ On calcule le signe de cette coordonnée et on le note ϵ_1.
  - ⋆ On pose $s_1 = s_0$ où on a remplacé la j$^{\text{ème}}$ coordonnée par ϵ_1.

- **Etape i ≥ 2 :** On fait décroître λ en partant de λ_{i-1} et on calcule $\alpha^{s_{i-1}}(\lambda)$. On s'arrête :
  - ⋆ soit lorsque la coordonnée j de s_{i-1} n'est pas nulle,
  - ⋆ soit lorsque la condition L_1 n'est plus satisfaite pour une coordonnée telle que $\left(s_{i-1}\right)_j = 0$.
  - ⋆ On définit λ_i la valeur de λ pour laquelle l'une des conditions ci-dessus n'est plus satisfaite.
  - ⋆ On calcule ϵ_i et on pose $s_i = s_{i-1}$ où on a remplacé la j$^{\text{ème}}$ coordonnée par ϵ_i.

1.4 La suite $\left(\lambda_i\right)_i$

Pour la validation de l'algorithme, un choix convenable pour la suite décroissante des paramètres λ_i est donné par la suite récurrente suivante en supposant connu le vecteur des signes s_i et λ_i :

$$\lambda_{i+1} = \max_{j \in \{1,\dots,N\}} \lambda_{\max}(j)$$

où

$$\lambda_{\max}(j) = \begin{cases} \max\limits_{a \in \{-a,a\}} \left\{ \lambda_a(j) \quad | \quad 0 < \lambda_a(j) < \lambda_i \right\} & \text{si} \quad s_i(j) = 0 \\ \dfrac{\left(\left(A_{\Gamma_{s_i}}^T A_{\Gamma_{s_i}}\right)^{-1} \left(A_{\Gamma_{s_i}}^T y\right) \right)_k}{\left(\left(A_{\Gamma_{s_i}}^T A_{\Gamma_{s_i}}\right)^{-1} s_{\Gamma_{s_i}} \right)_k} & \text{si} \quad s_i(j) \neq 0 \end{cases}$$

où $\lambda_a(j) = \dfrac{A_j^T \left(A_{\Gamma_{s_i}} \left(A_{\Gamma_{s_i}}^T A_{\Gamma_{s_i}}\right)^{-1} \left(A_{\Gamma_{s_i}} y\right) - y \right)}{\left(A_j^T A_{\Gamma_{s_i}} \left(A_{\Gamma_{s_i}}^T A_{\Gamma_{s_i}}\right)^{-1} s_{i,\Gamma} \right) + a}$ et k l'indice de la j$^{\text{ème}}$ variable dans Γ_{s_i}.

Exercice 3.

1. *Implémenter la première étape de la méthode Lasso sur Scilab.*
2. *Implémenter l'algorithme de la méthode Lasso sur Scilab.*

2. Matching Pursuit

Le matching Pursuit est un algorithme itératif qui recherche des approximations du signal $y \in \mathbb{R}^m$ par optimisation unidimensionnelle. On cherche le vecteur s-parcimonieux $\alpha \in \mathbb{R}^n$ vérifiant

$$y = A\alpha$$

La matrice A est égale à $A = \Phi\Psi$, A est d'ordre $m \times N$. On peut écrire

$$y = \alpha_1 A_1 + \alpha_2 A_2 + \ldots + \alpha_N A_N,$$

où A_j désigne la $j^{\text{ème}}$ colonne de la matrice A.

2.1 Principe du Matching Pursuit

Le principe du Matching Pursuit est de décomposer itérativement le signal y dans un dictionnaire qui correspond le mieux au signal à reconnaître :

$$y = \underbrace{\widehat{y}^{(n)}}_{\text{Approximation de y après n itérations}} + \underbrace{R^{(n)}}_{\text{Résiduel après n itérations}}.$$

1. **A l'étape 0**, on considère que $\widehat{y}^{(0)} = 0_{\mathbb{R}^m}$ et donc le résiduel est $R^{(0)} = y$.

2. **A l'itération 1**, on cherche l'atôme A_{m_1} qui contribue le plus au résiduel de l'étape précédente $R^{(0)}$:

   $$R^{(0)} = \underbrace{\alpha_{m_1} A_{m_1}}_{\text{Contribution de l'atôme } A_{m_1}} + \underbrace{R^{(1)}}_{\text{Résiduel après 1 itération}}.$$

   Mise à jour de l'approximation : $\widehat{y}^{(1)} = \widehat{y}^{(0)} + \alpha_{m_1} A_{m_1}$.

3. **A l'itération k**, on cherche l'atôme A_{m_k} qui contribue le plus au résiduel de l'étape précédente $R^{(k-1)}$:

   $$R^{(k-1)} = \underbrace{\alpha_{m_k} A_{m_k}}_{\text{Contribution de l'atôme } A_{m_k}} + \underbrace{R^{(k)}}_{\text{Résiduel après k itérations}}.$$

   Mise à jour de l'approximation : $\widehat{y}^{(k)} = \widehat{y}^{(k-1)} + \alpha_{m_k} A_{m_k}$.

4. **Critère d'arrêt :** On arrête les itérations selon un critère d'arrêt prédéfini.

   (a) *Arrêt après un nombre fini d'itérations.*

   (b) *Arrêt lorsque l'amplitude du résiduel est inférieure à un seuil prédéfini.*

5. **En sortie :** Notons K le nombre d'itérations utilisées pour cette approximation, en sortie, le signal original s'exprime sous la forme :

   $$y = \sum_{i=1}^{K} \alpha_{m_i} A_{m_i} + R^{(K)}.$$

2.2 Algorithme du Matching Pursuit

Le principe du Matching Pursuit repose sur une séléction d'atômes dont la contribution au résiduel est la plus grande. Pour cela, il faudra choisir une fonction de corrélation qui permettra de mesurer la corrélation entre le signal y et les différents atômes du dictionnaire. Nous utiliserons le produit scalaire comme fonction de corrélation.

`Algorithme MP` : $\alpha =$`MP(y,A)`

---

1. `Initialisation : initialiser le résiduel` R `avec le vecteur de mesure` y

   `et l'approximation du signal` $\widehat{y}$ `par un vecteur nul :`

   $$\mathbf{R_0} \leftarrow \mathbf{y} \text{ et } \mathbf{\widehat{y}_0} \leftarrow \mathbf{0}$$

2. `A l'itération k :`

   `Pour j=1,...,N,`

   `calculer` $\langle \mathbf{R_{k-1}}, \mathbf{A_j} \rangle$

   `Fin pour`

   `séléctionner` $\mathbf{m_k} = \text{argmax} \left| \langle \mathbf{R_{k-1}}, \mathbf{A_j} \rangle \right|$

   `Mise à jour :`

   $$\begin{aligned} \mathbf{\widehat{y}_k} &= \mathbf{\widehat{y}_{k-1}} + \langle \mathbf{R_{k-1}}, \mathbf{A_{m_k}} \rangle \mathbf{A_{m_k}} \\ \mathbf{R_k} &= \mathbf{R_{k-1}} - \langle \mathbf{R_{k-1}}, \mathbf{A_{m_k}} \rangle \mathbf{A_{m_k}} \end{aligned}$$

3. $\mathbf{k \leftarrow k+1}$

4. `Critère d'arrêt`

Exemple et mise en pratique

Pour comparer la contribution de chaque colonne A_j de la matrice A au signal y, on doit en premier lieu normaliser les colonnes de la matrice A. La procédure qu'on implémentera consiste à :

1. Calculer la matrice $A^T A$.
2. Extraire les éléments diagonaux d_j de $A^T A$. Pour $j = 1, \ldots, N$, $d_j = A_j^T A_j = \|A_j\|_2^2$.
3. Normaliser les vecteurs colonnes de la matrice A :

$$A_n = \left[\frac{A_1}{\sqrt{d_1}}, \frac{A_2}{\sqrt{d_2}}, \ldots, \frac{A_N}{\sqrt{d_N}} \right]$$

`Algorithme MP amélioré` : $\alpha =$`MPA`(y,A)

---

1. `Normaliser la matrice A`

   `Calculer` $\mathbf{A1} \leftarrow \mathbf{A}'\mathbf{A}$

   `Extraire la diagonale` $\mathbf{D} \leftarrow \mathbf{diag}(\mathbf{A1})$

   `Normaliser` $\mathbf{A} \leftarrow \mathbf{A}./\sqrt{\mathbf{D}}$

2. `Appliquer l'algorithme MP`(y,A)

Exercice 4. *On considère la matrice*

$$A = \begin{pmatrix} \frac{1}{2}\sqrt{2} & \frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{6} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} & -\frac{1}{6}\sqrt{6} & \frac{2}{3} & -\frac{2}{3} \\ 0 & -\frac{1}{3}\sqrt{3} & \frac{1}{6}\sqrt{6} & \frac{1}{3} & \frac{2}{3} \end{pmatrix}$$

Soit $\alpha = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 2 \\ 0 \end{pmatrix}$ *une représentation 3-parcimonieuse du vecteur* $y = \begin{pmatrix} \frac{4}{3} - \frac{1}{2}\sqrt{2} \\ \frac{4}{3} + \frac{1}{2}\sqrt{2} \\ \frac{2}{3} \end{pmatrix}$ *dans le dictionnaire* A *:* $y = A\alpha$.

1. *Reconstruire "à la main", une approximation de la représentation* α *en utilisant l'algorithme du Matching Pursuit. On choisira un critère d'arrêt relatif au seuil du résiduel :* $\|R\| \leq \epsilon$ *où* $\epsilon = 10^{-6}$.

2. *Implémenter l'algorithme du reconstruction du Matching Pursuit MPA sur Scilab.*

3. Orthogonal Matching Pursuit

Le Matching Pursuit Orthogonal s'inspire de l'algorithme du Maching Pursuit. La différence entre les deux algorithmes réside dans la mise à jour des approximations du signal $y \in \mathbb{R}^m$ et du résiduel, on utilisera la projection orhogonale d'où le nom de la méthode.

3.1 Principe de l'Orthogonal Matching Pursuit

Le principe de l'Orthogonal Matching Pursuit est de décomposer itérativement le signal y dans un dictionnaire qui minimise l'erreur quadratique moyenne.

1. **A l'étape 0**, on considère que $\widehat{y}^{(0)} = 0_{\mathbb{R}^m}$ et donc le résiduel est $R^{(0)} = y$.
2. **A l'itération 1**, on cherche l'atôme A_{m_1} qui contribue le plus au résiduel de l'étape précédente $R^{(0)}$.

   Mise à jour du dictionnaire actif : $D^{(1)} = \left[A_{m_1}\right]$.

   Mise à jour de l'approximation : $\widehat{y}^{(1)} = arg\min_x \left\|y - D^{(1)}x\right\|_2^2$.

   Mise à jour du résiduel : $R^{(1)} = y - A_{m_1}\widehat{y}^{(1)}$
3. **A l'itération k**, on cherche l'atôme A_{m_k} qui contribue le plus au résiduel de l'étape précédente $R^{(k-1)}$.

   Mise à jour du dictionnaire actif : $D^{(k)} = [D^{(k-1)}, A_{m_k}]$.

   Mise à jour de l'approximation : $\widehat{y}^{(k)} = arg\min_x \left\|y - D^{(k)}x\right\|_2^2$.

   Mise à jour du résiduel : $R^{(k)} = y - A_{m_k}\widehat{y}^{(k)}$

3.2 Algorithme du "Orthogonal Matching Pursuit"

Le principe du Matching Pursuit repose sur une séléction d'atômes dont la contribution au résiduel est la plus grande et une minimisation d'erreur.

`Algorithme MP` : $\alpha =$`OMP`(y,A)

---

1. `Initialisation :` $\mathbf{R}_0 \leftarrow \mathbf{y}$, $\widehat{\mathbf{y}}_\mathbf{0} \leftarrow \mathbf{0}$ et $D_0 = \emptyset$
2. `A l'itération k :`

   `Pour j=1,...,n,`

   `calculer` $\langle \mathbf{R_{k-1}}, \mathbf{A_j} \rangle$

   `Fin pour`

   `séléctionner` $\mathbf{m_k} = \text{argmax} \left| \langle \mathbf{R_{k-1}}, \mathbf{A_j} \rangle \right|$

   `Mise à jour :`

   $$\begin{aligned} \mathbf{D_k} &= [\mathbf{D_{k-1}}, \mathbf{A_{m_k}}] \\ \widehat{\mathbf{y}}_\mathbf{k} &= \mathbf{arg}\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D_k x}\| \\ \mathbf{R_k} &= \mathbf{y} - \mathbf{D_k}\widehat{\mathbf{y}}_\mathbf{k} \end{aligned}$$

3. $\mathbf{k} \leftarrow \mathbf{k} + \mathbf{1}$
4. `Critère d'arrêt`

Exercice 5. *Soient* $y \in \mathbb{R}^N$ *et* $V = \boldsymbol{Vect}\{v_1, v_2, \ldots, v_l\}$.

1. *Résoudre le problème des moindres carrées :*

$$\widehat{y} = arg \min_{x \in R^l} \left\| y - \sum_{i=1}^{l} x_i v_i \right\|_2^2.$$

2. *Appliquer l'algorithme OMP à la reconstruction du signal de l'exercice 1.*
3. *Implémenter l'algorithme OMP sur Scilab.*

Références

1. An Introduction To Compressive Sampling, Emmanuel J. Candès and Michael B. Wakin, EEE SIGNAL PROCESSING MAGAZINE, MARCH 2008.

2. Compressive sampling, Emamnuel J. Candès, Proceedings of the International Congress of Mathematicians, Madrid, Spain, 2006, European Mathematical Society.

3. Thèse : Représentations parcimonieuses pour les signaux multivariés - Quentin Barthélemy

4. Une petite introduction aux représentations parcimonieuses de signaux et d'images -Jérôme Landré 01-07-2008

5. Allen Y. Yang. Compressed sensing meets machine learning - classification of mixture subspace models via sparse representation. Mini Lectures in Image Processing, TRUST Center Seminar, University of California, Berkeley, USA, 2008.

6. Analyse Multi-Résolution et Ondelettes (AMO) Représentations des signaux 1D et 2D, Nicolas Thome Laboratoire d'Informatique de Paris 6 (LIP6) Université Pierre et Marie Curie (UPMC) Master 2 Informatique - Spécialité IMA

7. Compressive Sensing Richard G.Baraniuk IEE SIGNAL PROCESSING MAGAZINE [118] JULY 2007

8. R.G. Baraniuk. Compressive Sensing [lecture notes]. IEEE Signal Processing Magazine,

9. [DE03] David L. Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. Proceedings of the National Academy of Sciences,

10. Approche déterministe de l'acquisition comprimée et la reconstruction des signaux issus de capteurs intelligents distribués THESE présentée et soutenue publiquement le 9 novembre 2015 pour l'obtention du Doctorat de l'Université de Lorraine (mention systèmes électroniques) par Andrianiaina Ravelomanantsoa

11. A User's Guide to Compressed Sensing for Communications Systems Kazunori HAYASHI, Masaaki NAGAHARA, and Toshiyuki TANAHA, IEICE TRANS COMMUN, VOL.E96-B, NO.3 MARCH 2013

12. D.L Donoho, "Compressed sensing", IEEE Tran Inf. Theory, vol52, no.4, pp. 1289-1306 April 2006.

13. E.J. Candes and T.Tao, "Decoding by linear programming" ; IEEE Trans. Inf. Theory, Vol.51, no12, pp.4203-4215, Dec.2005

14. Regression Shrinkage and Selection via the Lasso, Robert Tibshirami, journal of the Royal Statistical Siciety. Series B (Methodological), Volume 58, Issue 1 (1996), 267-288

15. Compressed Sensing with Coherent and Redundant Dictionaries, Emmanuel J. Candes, Yonina C. Eldar, Deanna Needell, Paige Randall, Applied and Computational Harmonic Analysis, Volume 31, Issue 1, July 2011, Pages 59-73

16. Introduction au Compressed Sensing, notions de complexité algorithmique et relaxation convexe, Guillaume Lecué, CNRS, CREST, ENSAE.