Intelligence Artificielle & Machine Learning pour la modélisation de séries temporelles et de signaux

Séance 4 : Apprentissage de représentation et de dictionnaire

Laurent Oudre
laurent.oudre@ens-paris-saclay.fr

Diplôme ARIA
ENS Paris Saclay
2025-2026

Notion de représentation

- Afin de pouvoir faire de l'apprentissage sur des signaux il est nécessaire de les étudier dans un domaine de représentation adapté
- Nous avons déjà vu que le domaine fréquentiel était par exemple très adapté pour transformer, étudier et réparer les signaux
- Nous avons également vu un certain nombre de modèles classiques qui permettent de représenter un signal uniquement sous forme d'une liste de paramètres, donc sous une forme très condensée
- Dans tous les cas, il s'agissait d'utiliser des techniques sur étagère : serait-il possible d'apprendre la meilleure représentation directement à partir d'un ou plusieurs signaux ?

Retour sur la transformée de Fourier discrète

- Etant donné un signal $x[n]$ composé de N échantillons, la transformée de Fourier discrète (TFD) $X[k]$ s'écrit

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi \frac{kn}{N}} \text{ pour } 0 \leq k \leq N-1$$

- A l'inverse, on peut reconstruire le signal à partir des coefficients de Fourier

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, e^{j2\pi \frac{kn}{N}} \text{ pour } 0 \leq n \leq N-1$$

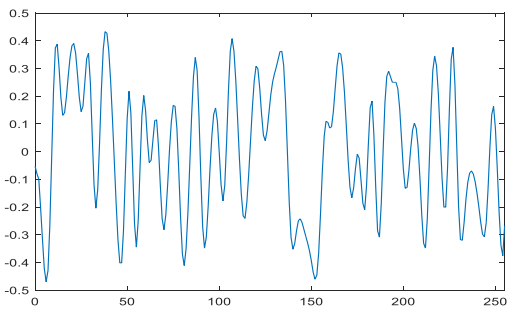
- Si le signal $x[n]$ est réel, en écrivant

$$a_k = \frac{1}{N} \operatorname{Re}\{X[k]\} \quad \text{et} \quad b_k = -\frac{1}{N} \operatorname{Im}\{X[k]\}$$

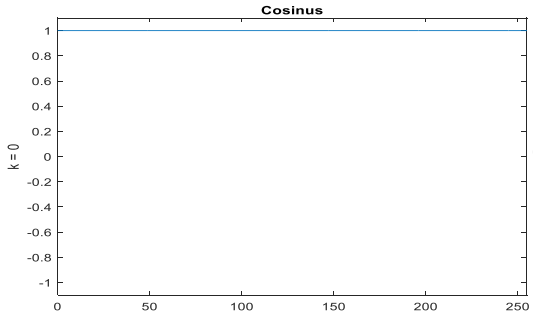
on peut écrire

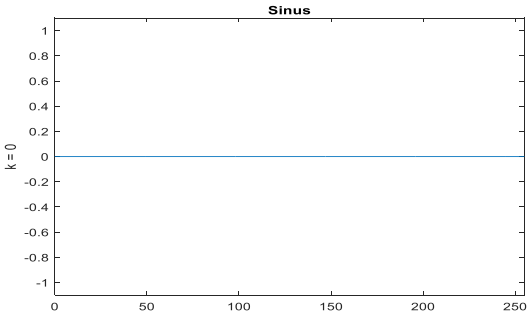
$$x[n] = \sum_{k=0}^{N-1} a_k \cos\left(2\pi \frac{kn}{N}\right) + b_k \sin\left(2\pi \frac{kn}{N}\right)$$

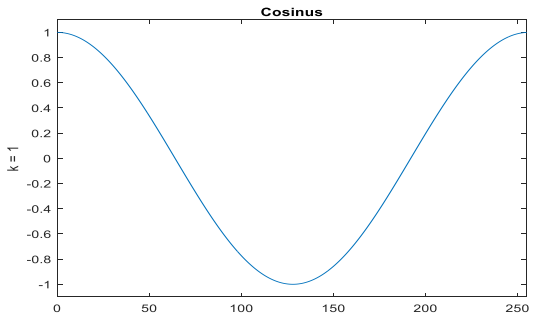
Retour sur la transformée de Fourier discrète

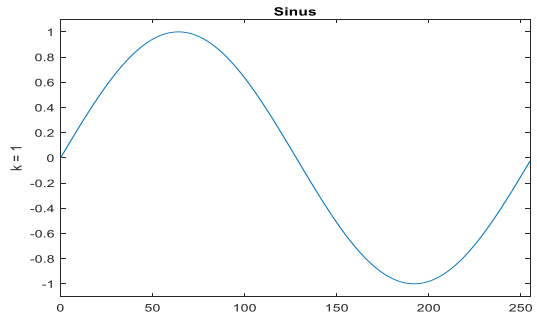


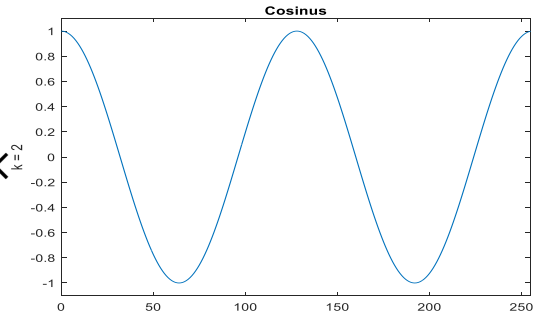
$=$

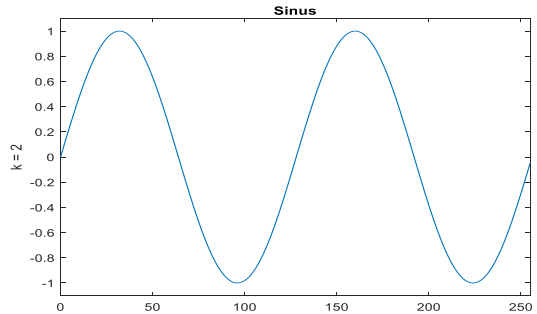
$a_0 \times$ 

$+b_0 \times$ 

$+a_1 \times$ 

$+b_1 \times$ 

$+a_2 \times$ 

$+b_2 \times$ 

$+\ldots$

Autres modèles

- Modèle sinusoïdal

$$x[n] = \sum_{i=1}^{N_h} a_i \sin\left(2\pi i f_0 \frac{n}{Fe} + \phi_i\right)$$

Là ici on suppose qu'on a une certaine collection de signaux $\sin\left(2\pi i f_0 \frac{n}{Fe}\right)$ et $\cos\left(2\pi i f_0 \frac{n}{Fe}\right)$ qui vont nous servir à représenter le signal $x[n]$

- Modèle tendance + saisonnalité

$$x[n] = \underbrace{\alpha_1\beta_1\left(nT_e\right) + \ldots + \alpha_j\beta_j\left(nT_e\right)}_{\text{tendance}} + \underbrace{\alpha_{j+1}\beta_{j+1}\left(nT_e\right) + \ldots + \alpha_d\beta_d\left(nT_e\right)}_{\text{saisonnalité}}$$

Ici ce sont les signaux $\beta_j\left(nT_e\right)$ qui sont utilisés pour la représentation

Extension

- En réalité, un grand nombre des représentations vues dans ce cours peuvent se mettre dans un même formalisme
- Un signal $x[n]$ va être représenté comme une combinaison linéaire d'un certain nombre de signaux de référence, qui seront rangés dans un **dictionnaire**
- K : nombre d'éléments du dictionnaire
- $\{d_k[n]\}_{1 \leq k \leq K}$: signaux du dictionnaire ou **atomes**

$$x[n] = \sum_{k=1}^{K} z_k d_k[n]$$

Approche par dictionnaire

$$\begin{pmatrix} x[0] \\ x[1] \\ \vdots \\ \vdots \\ \vdots \\ x[N-1] \end{pmatrix} = \begin{pmatrix} d_1[0] & d_2[0] & \cdots & d_K[0] \\ d_1[1] & d_2[1] & \cdots & d_K[1] \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ d_1[N-1] & d_2[N-1] & \cdots & d_K[N-1] \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{pmatrix}$$

$$\mathbf{x} = \mathbf{D}\mathbf{z}$$

- K : taille du dictionnaire
- $\mathbf{D}$: dictionnaire
- $\mathbf{z}$: vecteur d'activations

Notion de représentation

- La représentation par dictionnaire consiste à considérer un dictionnaire $\mathbf{D}$ (connu ou non) et à représenter le signal $\mathbf{x}$ grâce au vecteur d'activations
- La représentation fréquentielle ainsi que certains modèles vus dans le cours se ramènent exactement à cette formulation
- Bien représenter un signal, c'est trouver un dictionnaire et/ou des activations qui font que le signal composé à la base de N échantillons, puisse être expliqué par un petit nombre de valeurs

Programme de la séance

- Connaître les techniques d'apprentissage pour la représentation des séries temporelles
- Apprentissage de représentation et de dictionnaire

Session 4 : Representation and dictionary learning

Plan du cours

1. Codage parcimonieux

2. Apprentissage de dictionnaire

Plan du cours

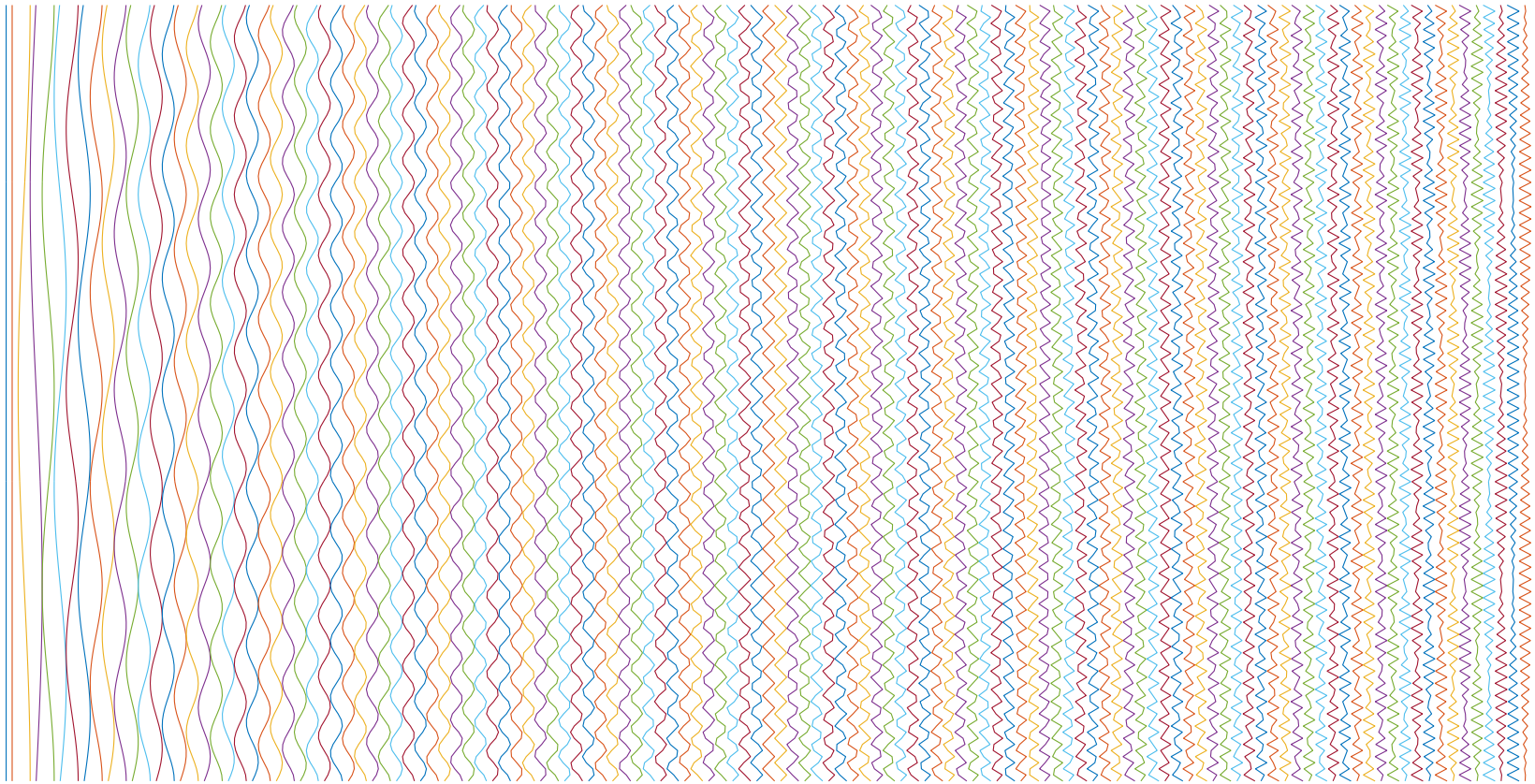
1. Codage parcimonieux

2. Apprentissage de dictionnaire

Dictionnaire redondant

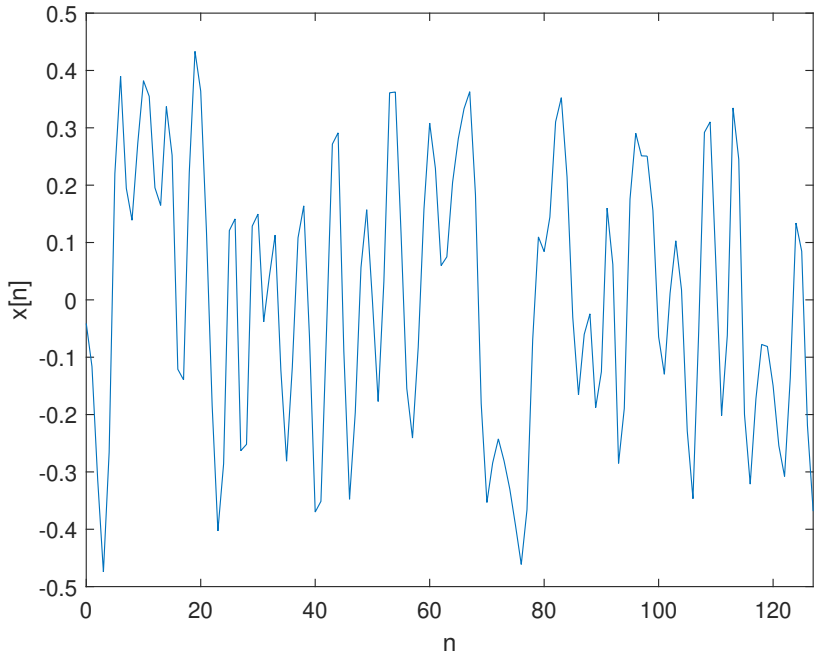
- Nous allons considérer dans un premier temps que le dictionnaire $\mathbf{D}$ est connu et qu'il contient un grand nombre d'éléments $K \geq N$
- C'est le cas par exemple du dictionnaire de Fourier constitué des fonctions cosinus et sinus
- Dans ce cas, si l'on représente le signal sous forme d'un vecteur d'activations de dimension K, on le représente avec encore plus de valeurs que le nombre d'échantillons N....

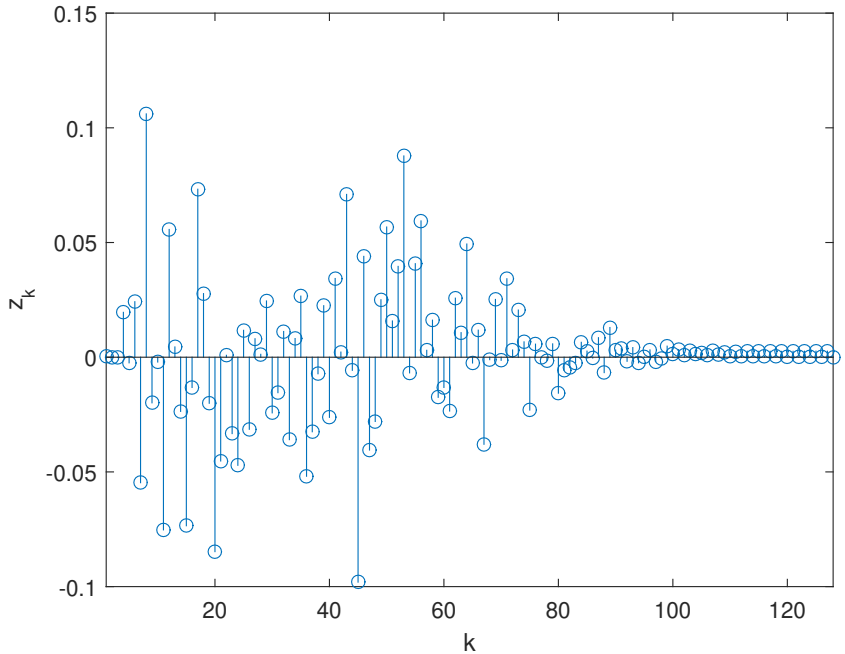
Dictionnaire de Fourier



$$N = K = 128$$

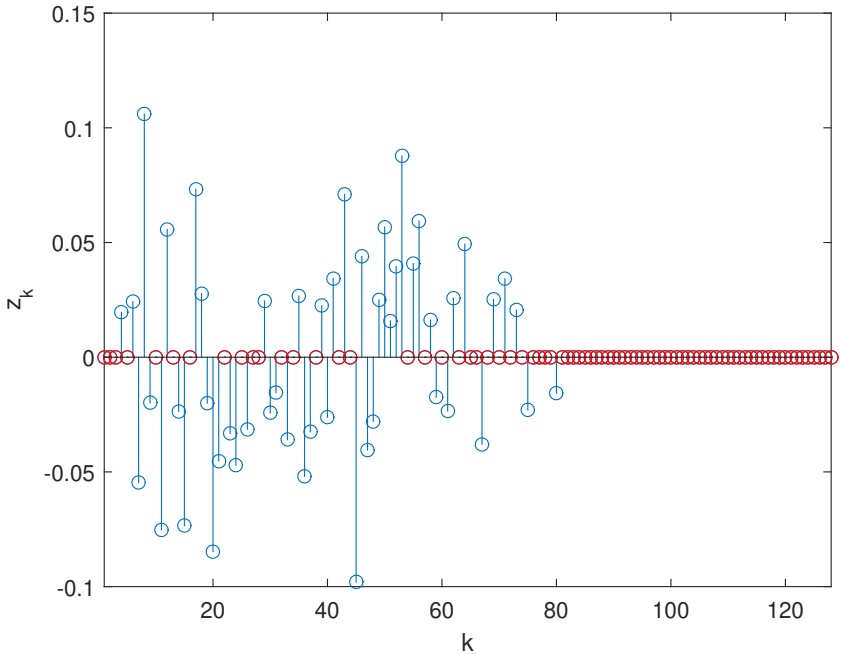
Exemple

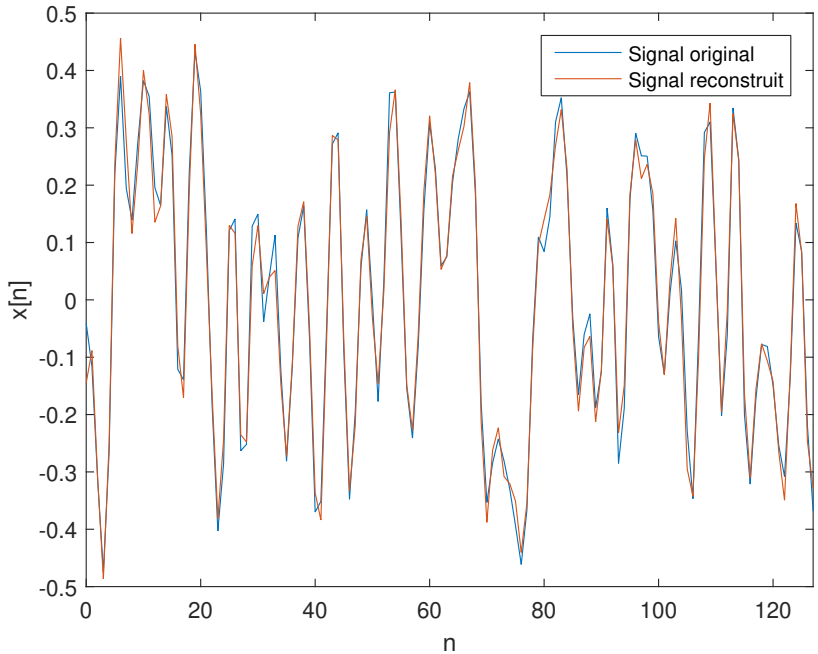




En réalité, beaucoup d'activations z_k sont proches de 0

Exemple





On garde uniquement les 50 plus grandes activations

Principe de la parcimonie

- En réalité, si $K \geq N$, le dictionnaire contient de la redondance, et donc beaucoup d'activations z_k vont être très proches de 0
- Trouver une bonne représentation du signal, c'est donc trouver un vecteur d'activations parcimonieux, c'est à dire qui contient beaucoup de valeurs nulles
- Dans l'exemple précédent, il était par exemple possible de représenter un signal contenant 128 échantillons grâce à 50 valeurs
- Retrouver un vecteur d'activation $\mathbf{z}$ parcimonieux à partir d'un signal $\mathbf{x}$ et d'un dictionnaire $\mathbf{D}$ s'appelle le **codage parcimonieux**

Méthode naïve

- Pour trouver les paramètres $\mathbf{z}$ qui permettent de reconstruire le signal $\mathbf{x}$, on peut utiliser un critère des moindres carrés

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2$$

- Ce problème a une solution exacte (déjà vue) que l'on peut écrire :

$$\mathbf{z}^* = \left(\mathbf{D}^T\mathbf{D}\right)^{-1}\mathbf{D}^T\mathbf{x}$$

- Afin de rendre le vecteur d'activation parcimonieux, on peut décider que si l'on souhaite K_0 valeurs non nulles (c'est à dire $\|\mathbf{z}\|_0 = K_0$) on va choisir les K_0 plus grandes valeurs de $\mathbf{z}^*$ et annuler les autres : permet une reconstruction approchée mais pas forcément optimale
- Cette façon de procéder nécessite également l'inversion de la matrice $\mathbf{D}^T\mathbf{D}$ qui peut être compliquée

Codage parcimonieux

- Afin de résoudre ce problème de façon efficace, on va introduire des algorithmes de **codage parcimonieux** qui vont encoder un signal sur un dictionnaire d'atomes, en utilisant le moins d'atomes possibles
- On se basera sur la résolution de problèmes d'optimisation contraints et on introduira pour cela deux critères différents de parcimonie
- Norme $\ell 0$

$$\|\mathbf{z}\|_0 = \text{nb de coeff non nuls}$$

- Norme $\ell 1$

$$\|\mathbf{z}\|_1 = \sum_{k=1}^{K} |z_k|$$

Codage parcimonieux

Il y a plusieurs formulations pour le problème de codage parcimonieux :

- Régularisation $\ell 0$

$$\mathbf{z}^* = \underset{\substack{\mathbf{z} \\ \|\mathbf{z}\|_0 = K_0}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2$$

  - Iterative Hard Thresholding [Blumensath et al., 2008]
  - Matching Pursuit [Mallat et al., 1993]

- Régularisation $\ell 1$ (appelée aussi LASSO : Least Absolute Shrinkage and Selection Operator)

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1$$

  - Iterative Soft Thresholding Algorithm (ISTA) [Daubechies et al., 2004]

Régularisation $\ell 0$: Iterative Hard Thresholding

$$\mathbf{z}^* = \underset{\|\mathbf{z}\|_0 = K_0}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2$$

Algorithm 1: Iterative Hard Thresholding

Inputs : Signal $\mathbf{x} \in \mathbb{R}^N$
Dictionnaire $\mathbf{D} \in \mathbb{R}^{N \times K}$
Nombre de coefficients non-nuls K_0

Output: Vecteur d'activations $\mathbf{z} \in \mathbb{R}^K$ avec K_0 activations non nulles

$\mathbf{z} = \mathbf{0}_K$;

while $n_{iter} < n_{max}$ **do**

  Descente de gradient;

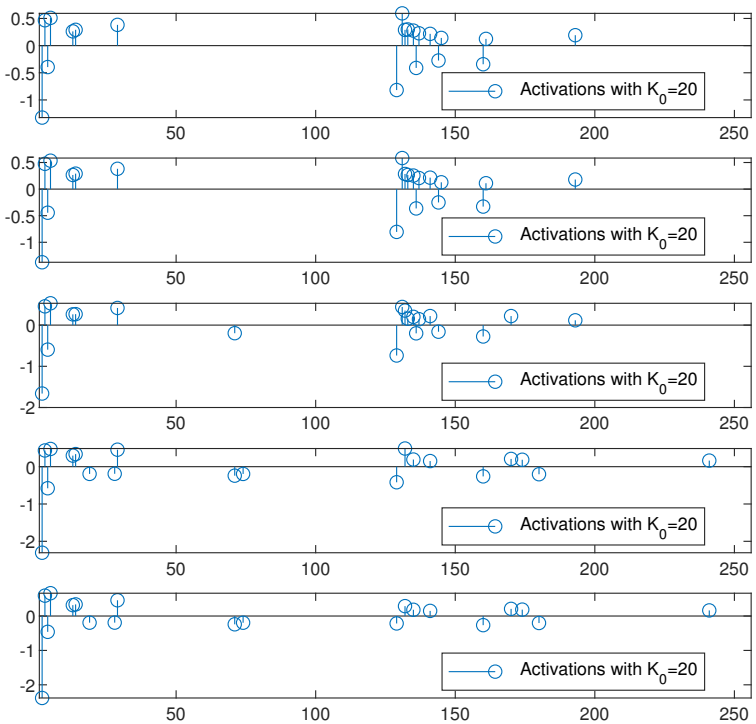
  $\mathbf{z} \leftarrow \mathbf{z} - \mu \mathbf{D}^T (\mathbf{D}\mathbf{z} - \mathbf{x})$;

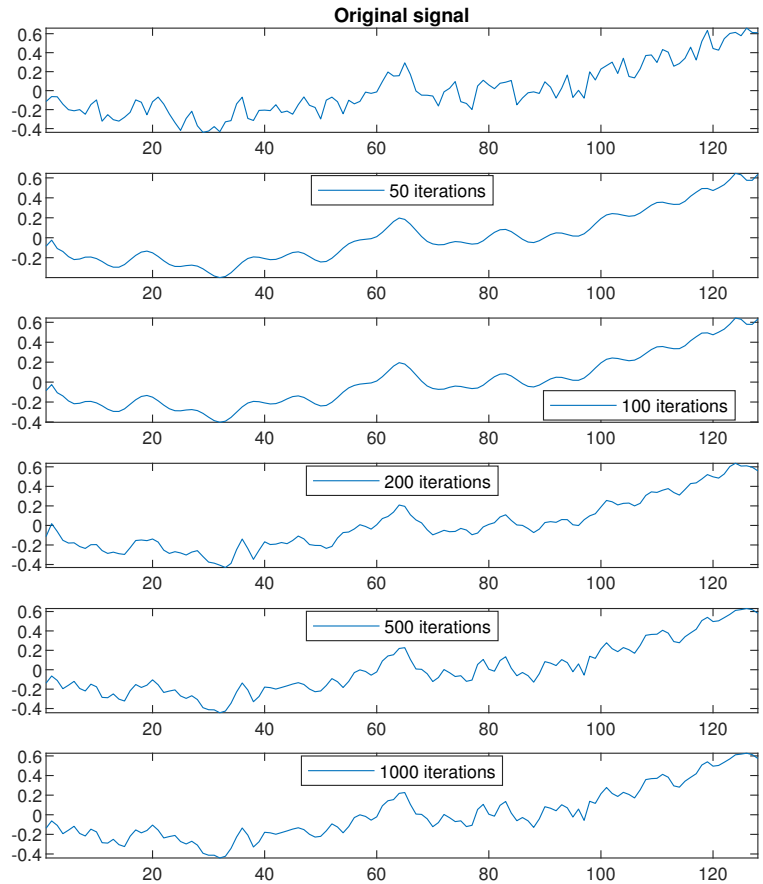
  Etape de Hard Thresholding;

  Projection pour ne garder que les K_0 plus grandes valeurs du vecteur $\mathbf{z}$;

end

Régularisation $\ell 0$: Iterative Hard Thresholdingg





Dictionary : Fourier + Wavelet (db4 - level 5), $K_0 = 20$

Régularisation $\ell 0$: Matching Pursuit

$$\mathbf{z}^* = \underset{\|\mathbf{z}\|_0 = K_0}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2$$

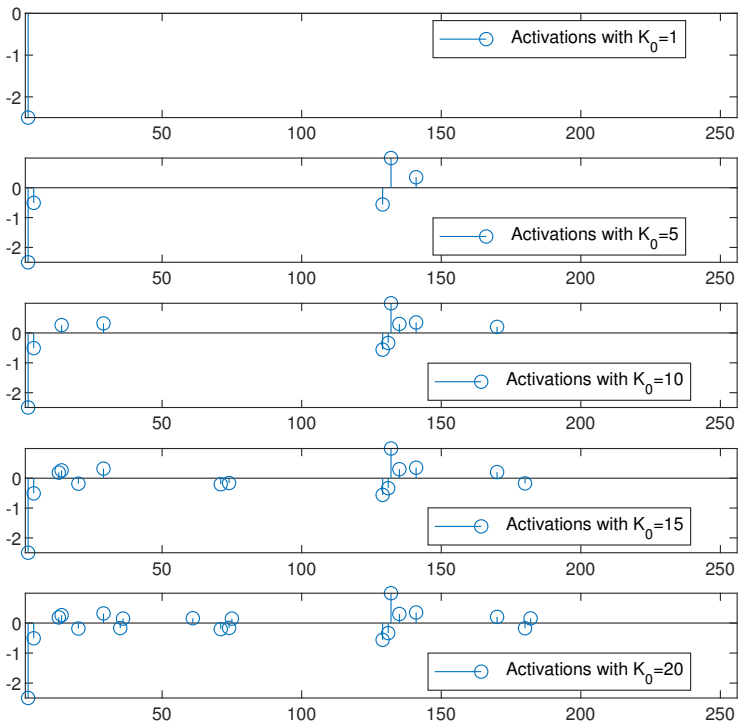
Algorithm 2: Matching Pursuit

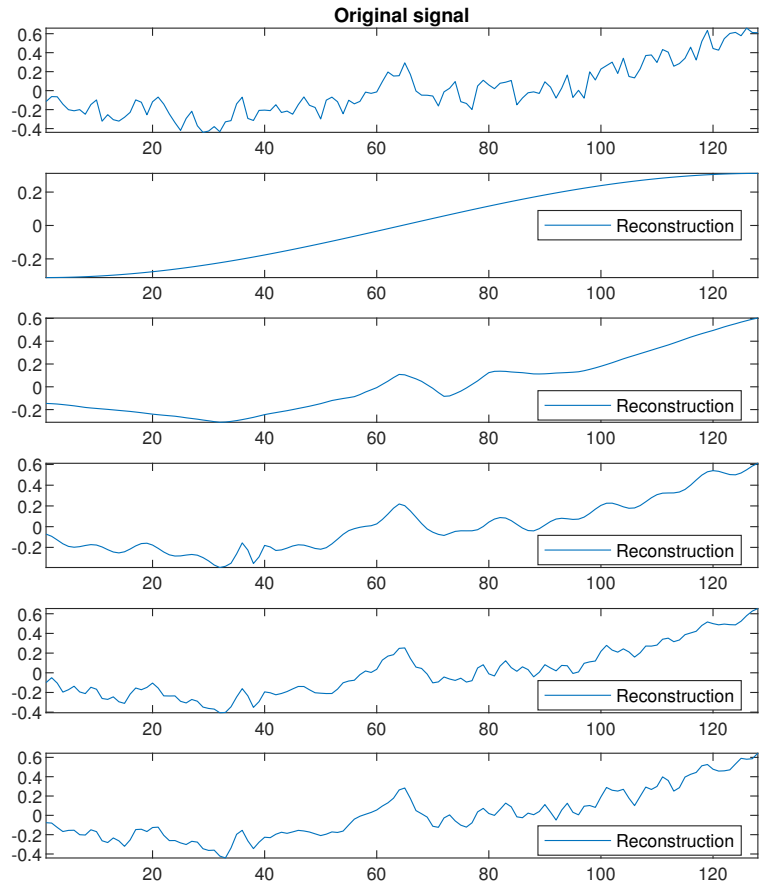
Inputs : Signal $\mathbf{x} \in \mathbb{R}^N$
Dictionnaire $\mathbf{D} \in \mathbb{R}^{N \times K}$ avec des colonnes normalisées
Nombre de coefficients non-nuls K_0

Output: Vecteur d'activations $\mathbf{z} \in \mathbb{R}^K$ avec K_0 activations non nulles

$\mathbf{z} = \mathbf{0}_K$;
$\mathbf{r} = \mathbf{x}$;
while $n_{iter} < K_0$ **do**
    Trouver le produit scalaire maximal;
    $k^* = \operatorname{argmax}_k |\langle \mathbf{r}, \mathbf{d}_k \rangle|$;
    Etape de projection;
    $z_{k^*} = \langle \mathbf{r}, \mathbf{d}_{k^*} \rangle$;
    $\mathbf{r} = \mathbf{r} - z_{k^*} \mathbf{d}_{k^*}$;
end

Régularisation $\ell 0$: Matching Pursuit





Dictionary : Fourier + Wavelet (db4 - level 5), $K_0 = 20$

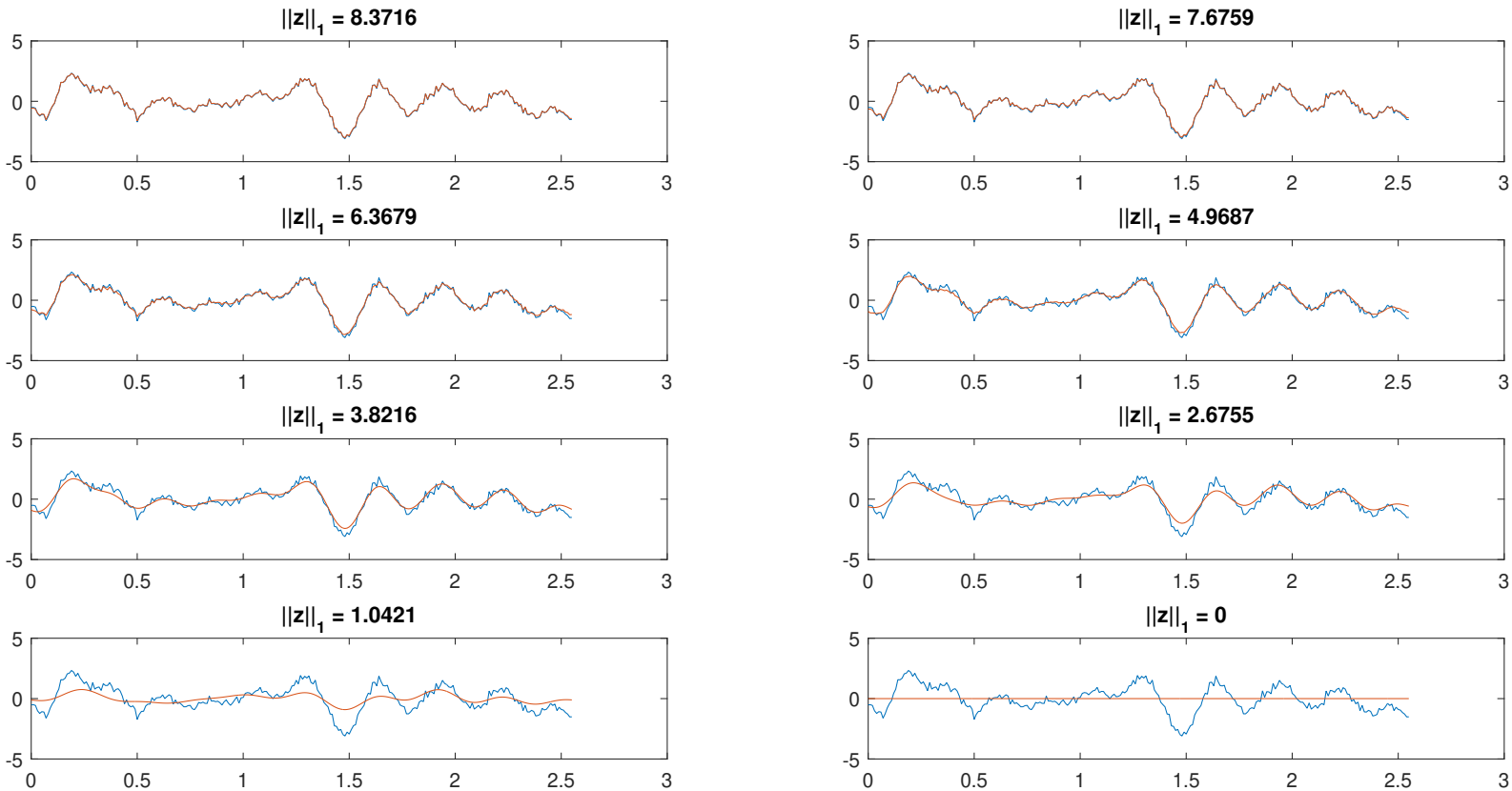
Régularisation $\ell 1$: Iterative Soft Thresholding Algorithm (ISTA)

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{D}\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_1$$

Algorithm 3: Iterative Hard Thresholding Algorithm (ISTA)

Inputs : Signal $\mathbf{x} \in \mathbb{R}^N$
Dictionnaire $\mathbf{D} \in \mathbb{R}^{N \times K}$
Penalité λ
Output: Vecteur d'activations parcimonieuses $\mathbf{z} \in \mathbb{R}^K$
$\mathbf{z} = \mathbf{0}_K$;
while $n_{iter} < n_{max}$ **do**
    Descente de gradient;
    $\mathbf{z} \leftarrow \mathbf{z} - \mu \mathbf{D}^T (\mathbf{D}\mathbf{z} - \mathbf{x})$;
    Etape de Soft Thresholding;
    $\mathbf{z} = \mathcal{S}_{\lambda\mu}(\mathbf{z}) = \text{sign}(\mathbf{z}) \times \max(|\mathbf{z}| - \lambda\mu, 0)$;
end

Régularisation $\ell 1$: Iterative Soft Thresholding Algorithm (ISTA)



Dictionnaire de Fourier

Le paramètre λ permet de doser le degré de parcimonie que l'on souhaite

Choix du dictionnaire

- Le choix du dictionnaire à utiliser dépend des phénomènes observés dans le signal
- Base de Fourier : dictionnaire non informatif qui permet de reconstruire parfaitement le signal si toutes les activations sont non nulles
- Fonctions polynomiales : modélisation de la tendance et des variations lentes
- Cosinus et sinus pour des multiples de la fréquence fondamentale : modélisation des aspects harmoniques (ou saisonnalité)
- Ondelettes : permet de gérer les aspects multi-échelles + les phénomènes locaux

Plan du cours

1. Codage parcimonieux

2. Apprentissage de dictionnaire

Choix du dictionnaire

- Serait-il possible d'apprendre automatiquement le dictionnaire qui permet de représenter au mieux le signal ?
- Notion d'**apprentissage de dictionnaire** : apprendre directement à partir des signaux le meilleur dictionnaire possible
- Problème : si l'on dispose d'un seul signal il existe un dictionnaire trivial qui sera parfait : le signal lui-même
- On va donc considérer ici une collection de M signaux $\mathbf{x_1}, \ldots, \mathbf{x_M}$: ces signaux sont servir d'entrée pour apprendre le dictionnaire $\mathbf{D}$

Notations

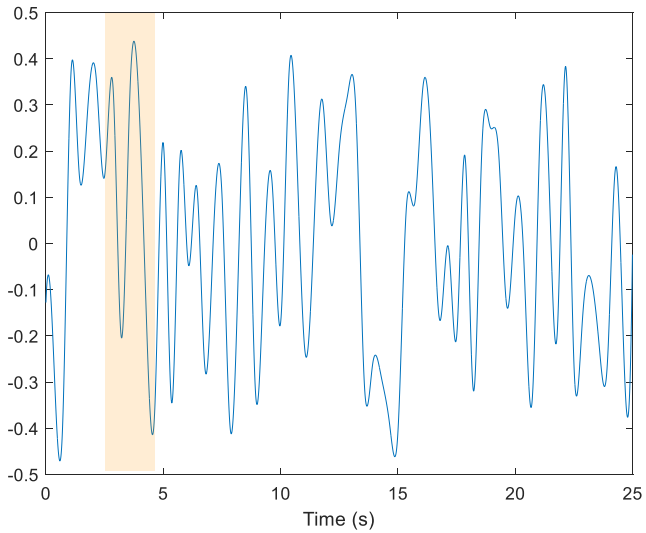
- On a donc en entrée une matrice $\mathbf{X}$ contenant les M signaux

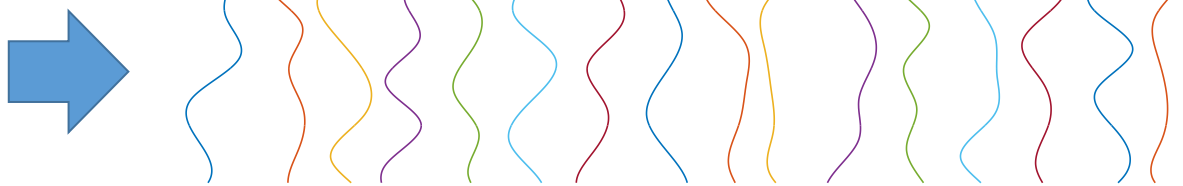
$$\mathbf{X} = \begin{pmatrix} x_1[0] & x_2[0] & \cdots & x_M[0] \\ x_1[1] & x_2[1] & \cdots & x_M[1] \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ x_1[N-1] & x_2[N-1] & \cdots & x_M[N-1] \end{pmatrix}$$

- Avec cette configuration le dictionnaire $\mathbf{D}$ est toujours de taille $N \times K$, mais le vecteur d'activation devient une matrice d'activation $\mathbf{Z}$ de taille $K \times M$

$$\mathbf{D} = \begin{pmatrix} d_1[0] & d_2[0] & \cdots & d_K[0] \\ d_1[1] & d_2[1] & \cdots & d_K[1] \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ d_1[N-1] & d_2[N-1] & \cdots & d_K[N-1] \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,M} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,M} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ z_{K,1} & z_{K,2} & \cdots & z_{K,M} \end{pmatrix}$$

Exemple





$\mathbf{X}$

Si l'on ne dispose pas de plusieurs signaux on peut diviser notre signal original en trames pour former la matrice $\mathbf{X}$

Problème

- Le but du problème revient à faire en sorte que $\mathbf{DZ}$ soit le plus proche possible du signal $\mathbf{X}$
- L'apprentissage de dictionnaire peut donc se résoudre grâce à une minimisation des moindres carrés

$$\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{DZ}\|_F^2$$

où $\|.\|_F$ est la norme de Frobenius définie par

$$\|\mathbf{Y}\|_F = \sqrt{\sum_{i,j} Y_{i,j}^2}$$

- Il existe une solution exacte à ce problème, si l'on suppose les activations $\mathbf{Z}$ connues

$$\mathbf{D}^* = \mathbf{X}\mathbf{Z}^T \left(\mathbf{Z}\mathbf{Z}^T\right)^{-1}$$

- Exactement comme pour le codage parcimonieux, cette résolution peut être complexe surtout si la matrice $\mathbf{Z}$ est de grande taille

Méthode itérative

- Afin d'éviter les problèmes de conditionnement et d'accroître l'interprétabilité, on va rajouter une contrainte sur le problème, en forçant chaque élément du dictionnaire $\mathbf{d_k}$ à être de norme 2 plus petite que 1
- Le problème à résoudre devient

$$\mathbf{D}^* = \underset{\forall k, \|\mathbf{d_k}\|_2 = 1}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{DZ}\|_F^2$$

- Pour le résoudre, on peut une méthode itérative basée sur une descente de gradient
  1. Initialisation de $\mathbf{D}^0$ aléatoire et normalisé
  2. Itérations
     - Mise à jour des coefficients

$$\mathbf{D}^{\ell+1} \leftarrow \mathbf{D}^{\ell} - \mu \left(\mathbf{D}^{\ell}\mathbf{Z} - \mathbf{X}\right) \mathbf{Z}^T$$

     - Projection pour faire en sorte que chaque élément doit de norme 2 égale à 1

$$\mathbf{d_k}^{\ell+1} = \frac{\mathbf{d_k}^{\ell+1}}{\left\|\mathbf{d_k}^{\ell+1}\right\|_2}$$

Problème général

- Pour calculer les activations... il faut le dictionnaire. Pour calculer le dictionnaire... il faut les activations.
- Les tâches d'apprentissage de dictionnaire et de codage parcimonieux sont donc intrinsèquement liées
- Dans la pratique on va donc alterner entre ces deux tâches jusqu'à convergence

Résolution par descente de gradient

Initialisation

Paramètres d'entrée

- K : nombre d'atomes, N : taille des signaux, M : nombre d'observations
- $\mathbf{X} \in \mathbb{R}^{N \times M}$: données
- K_0 : nombre de composantes que l'on souhaite garder

Initialisations

- $\mathbf{D} \in \mathbb{R}^{N \times K}$ aléatoire avec $\forall k, \quad \|\mathbf{d}_k\|_2 = 1$
- $\mathbf{Z} \in \mathbb{R}^{K \times M}$ avec tous les coefficients nuls

Résolution par descente de gradient

Mise à jour Z (ici L0)

$$\mathbf{Z}^{\ell+1} \leftarrow \mathbf{Z}^{\ell} - \mu \mathbf{D}^T \left(\mathbf{D}\mathbf{Z}^{\ell} - \mathbf{X} \right)$$

Mise à zéro de tous les coefficients sauf les K_0 plus grands

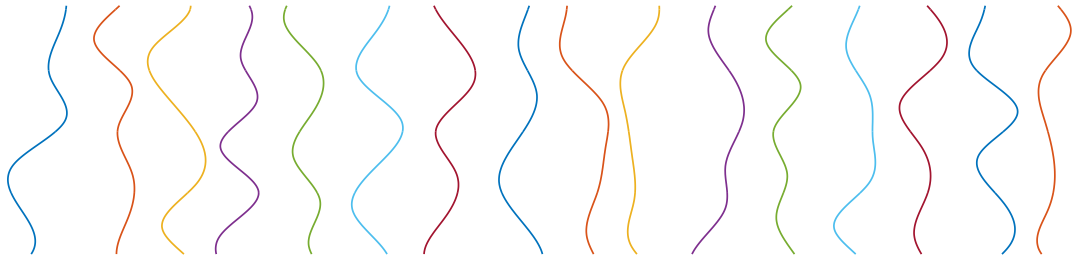
$\updownarrow$ alternance

Mise à jour D

$$\mathbf{D}^{\ell+1} \leftarrow \mathbf{D}^{\ell} - \mu \left(\mathbf{D}^{\ell}\mathbf{Z} - \mathbf{X} \right) \mathbf{Z}^T$$

$$\forall k, \quad \mathbf{d_k}^{\ell+1} = \frac{\mathbf{d_k}^{\ell+1}}{\left\| \mathbf{d_k}^{\ell+1} \right\|_2}$$

Exemple



d_1 

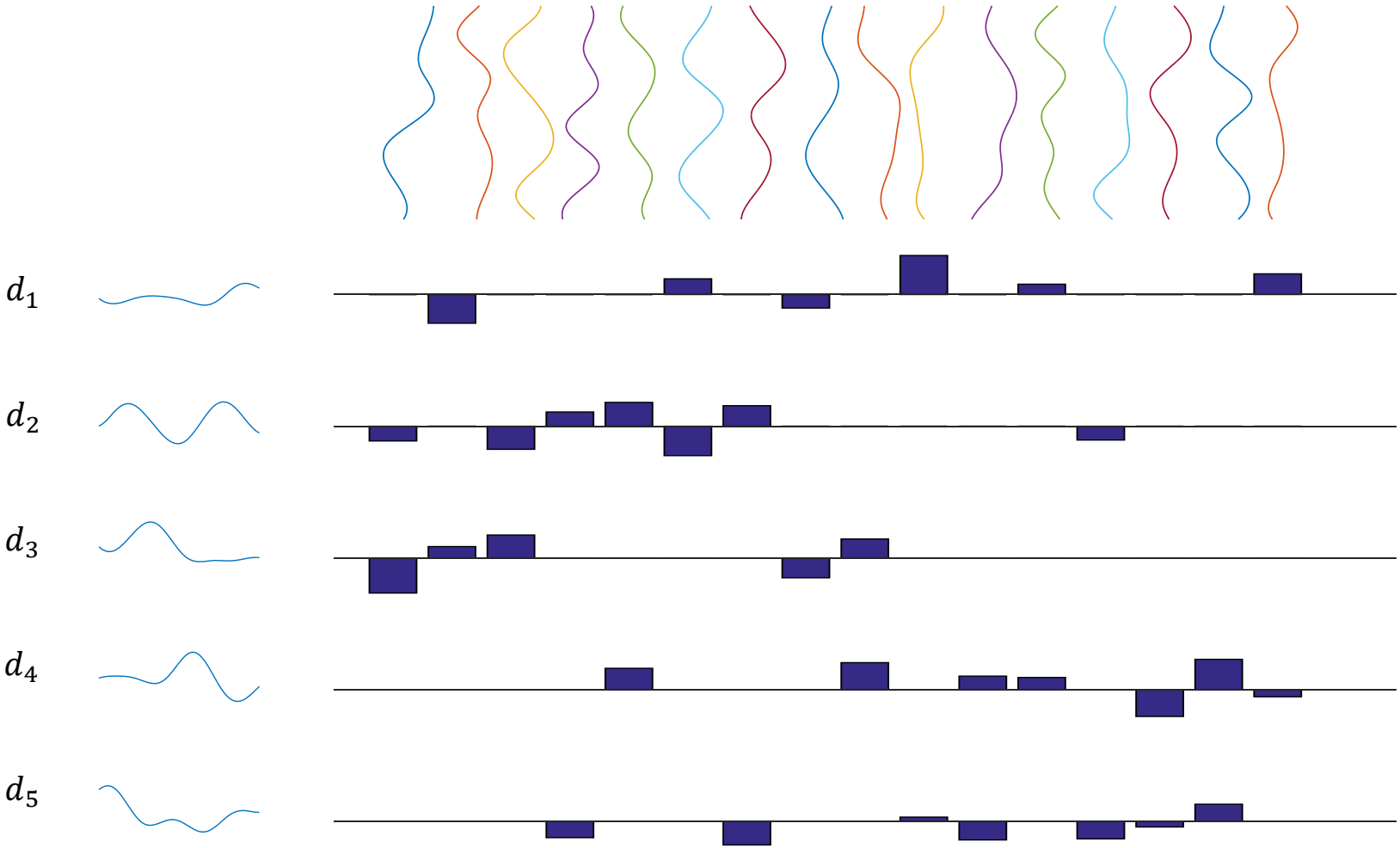
d_2 

d_3 

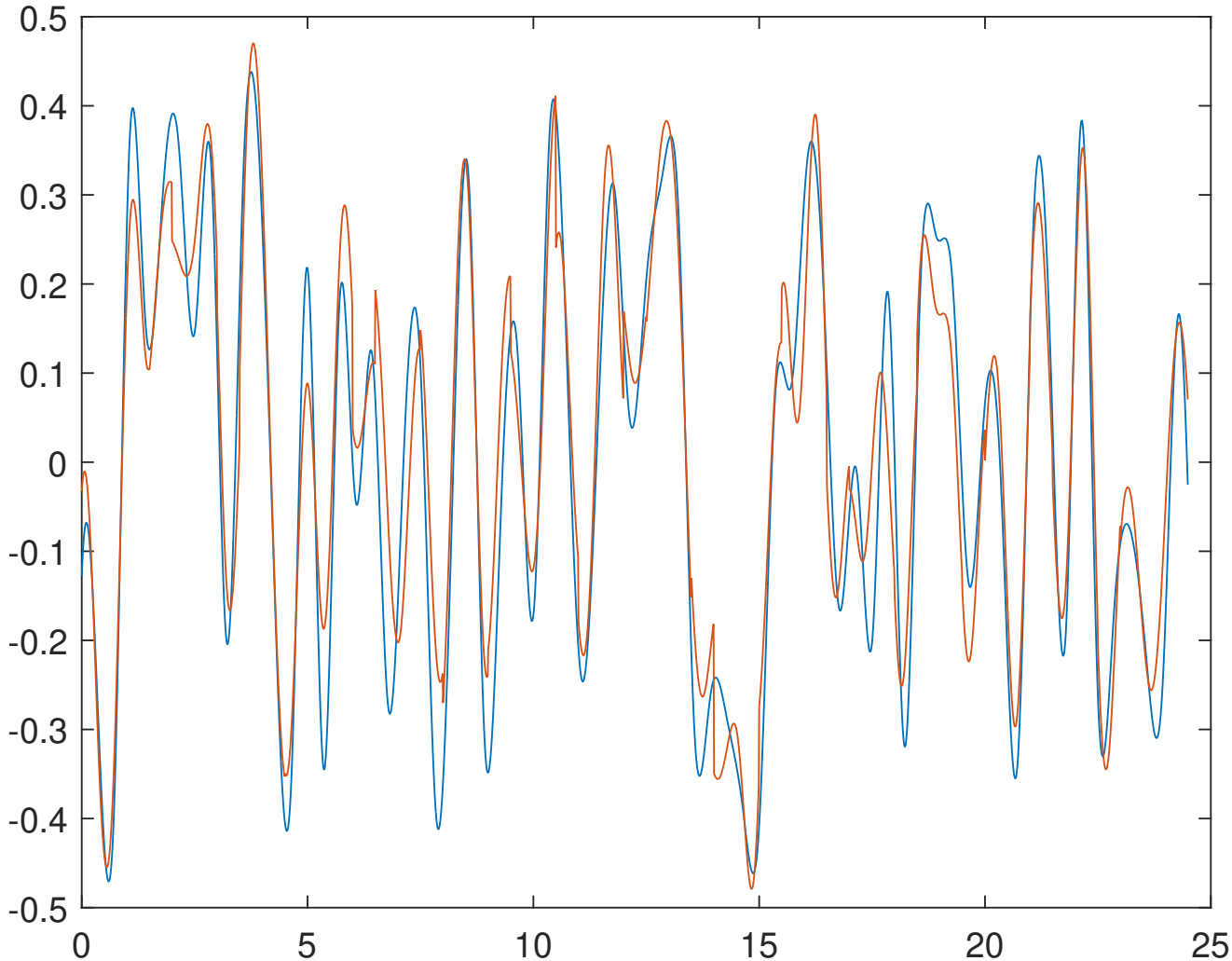
d_4 

d_5 

Exemple



Exemple



Autres méthodes de résolution

- Il existe en réalité un très grand nombre de méthodes pour résoudre les étapes Z et D
- On peut notamment pour l'étape Z utiliser une approche de type LASSO déjà vue pour le codage parcimonieux
- D'autres contraintes peuvent également être prises en compte pour la normalisation du dictionnaire

Exemples d'application

- L'apprentissage de dictionnaire/codage parcimonieux peut être vu comme un apprentissage non supervisé sur les signaux
- Il permet d'apprendre une représentation adaptée aux signaux
- Les activations peuvent être utilisées comme paramètres pour représenter ou indexer les signaux
- En regroupant les atomes de façon astucieuse, on peut également réaliser des tâches de débruitage, de séparation de sources, de suppression de tendance... (voir séance sur les pré-traitements)

Références

- G. Peyré. Sparsity and compressed sensing
  http://www.numerical-tours.com/
  https://mathematical-tours.github.io/book-sources/chapters-pdf/sparse-regularization.pdf
  https://mathematical-tours.github.io/book-sources/chapters-pdf/compressed-sensing.pdf
- I. Tosic and P. Frossard. "Dictionary learning : What is the right representation for my signal ?." IEEE Signal Processing Magazine 28.ARTICLE (2011) : 27-38.
  https://infoscience.epfl.ch/record/161378/files/spm2011.pdf
- R. Gribonval and M. Nielsen. "Sparse representations in unions of bases." IEEE transactions on Information theory 49.12 (2003) : 3320-3325.
  https://hal.inria.fr/inria-00071943/document
- J. Mairal et al. "Online dictionary learning for sparse coding." Proceedings of the 26th annual international conference on machine learning. ACM, 2009.
  https://apps.dtic.mil/dtic/tr/fulltext/u2/a513243.pdf
- K. Kreutz-Delgado et al. "Dictionary learning algorithms for sparse representation." Neural computation 15.2 (2003) : 349-396.
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2944020/