U.E. 4.5 : Sparse Component Analysis for audio source separation
SOIA - 2021

ENSTA Bretagne
2 rue F. Verny
29806 Brest Cedex 9, France

Adrien DEOUX, adrien.deoux@ensta-bretagne.org
Chloé DENIS, chloe.denis@ensta-bretagne.org
Matthieu ROCHE, matthieu.roche@ensta-bretagne.org
Maxime BARRET, maxime.barret@ensta-bretagne.org
Pauline BEAUJARD, pauline.beaujard@ensta-bretagne.org

Remerciements

Nous tenons à remercier Angélique Drémeau et Charles Vanwynsberghe pour leur aide sur ce projet très intéressant et pour leur bonne humeur communicative. Nous tenons également à remercier Matthieu Roche pour ses talents de Beatmaker. Merci également à Maxime Barret pour les soirées passées sur son ordinateur pour le bien de l'équipe. Un grand homme a un jour dit que « généraliste » voulait dire « capable de tout, bon à rien », mais en tant que véritables ingénieurs généralistes à caractère pluridisciplinaire, les membres de cette équipe sont à la fois capables de tout, et bon tout court.

Résumé

En écoutant une chanson, il est parfois possible que nous tombions sur un instrument que nous n'aimons pas. La chanson ayant déjà été enregistré, est-il possible d'enlever le son d'un instrument, tout en gardant les autres sons ? Autrement dit : est-il possible de séparer les sources dans un signal sonore, et de ne garder que les sources utiles ? Une chanson n'est pas un signal parcimonieux, a priori, mais en changeant de domaine il est possible de séparer différentes sources.

Abstract

While listening to a song, we might come across an instrument we're not particularly fond of. Since the song has already been recorded, is it possible to remove the sound of an instrument while keeping the others'? In other words: is it possible to separate the sources in a signal, and keep only the useful ones? A song is not a sparse signal per say, but by changing the domain of study, we might just be able to separate different sources.

Table des matières

Introduction

Ce projet est l'occasion d'utiliser la parcimonie sur un sujet pratique, permettant de mettre en application nos connaissances, de rechercher les informations manquantes dans des articles scientifiques, et de se familiariser avec des problèmes de traitement de données.

1. Situation de départ

1.1. La parcimonie

La représentation parcimonieuse des signaux est la représentation d'un signal avec un faible nombre de coefficients significatifs.

Par définition, un signal est dit parcimonieux lorsque la plupart de ses coefficients sont (approximativement) nuls. Les représentations parcimonieuses consistent en la décomposition du signal sur un dictionnaire comprenant un nombre d'éléments (ou atomes) très supérieur à la dimension du signal. Cette décomposition va introduire dans la nouvelle représentation du signal un grand nombre de valeurs nulles : on parle alors de représentation parcimonieuse.

Les problématiques scientifiques que posent ces représentations sont la définition du critère de parcimonie et la construction du dictionnaire en fonction de l'application à réaliser. Pour représenter un signal parcimonieux, il faut donc bien définir ce dictionnaire et ce critère de parcimonie en fonction de ce que l'on veut faire. Ainsi, l'objectif premier du projet est d'étudier en quelle mesure il est possible d'utiliser une représentation parcimonieuse pour faire de la séparation de sources sur notre signal audio.

1.2. Situation pratique

Dans ce projet, nous partons du principe que nous écoutons un morceau composé par Matthieu Roche. Ce morceau est diffusé grâce à deux haut-parleurs, en stéréo et est composé de 3 sources, c'est à dire de trois instruments différents qui jouent simultanément. Le but est de retrouver au final, 3 signaux audios représentant la mélodie jouée par chacun des instruments, séparément.

De manière plus générale, l'idée serait de généraliser cet algorithme à un ensemble d'instruments plus nombreux et de pouvoir l'appliquer à un signal audio qui serait enregistré, par exemple durant un concert, et dont on souhaiterait isoler l'un des instruments. Le processus pourrait également s'appliquer à des enregistrements de voix uniquement ou à des voix et instruments.

1.3. Mise en équation

L'analyse en composante parcimonieuse peut se ramener au problème inverse suivant :

$$x(t) = A.s(t) + b(t)$$

t représente ici le temps ($0 \leq t \leq T$), x le signal audio reçu, s le signal source, et A la matrice de mélange, de dimension (P, N) avec P le nombre de capteurs et N le nombres de sources.
b représente le bruit, qui sera négligé par la suite du fait de notre signal audio généré directement depuis ordinateur.

Dans notre cas d'étude, nous travaillons en stéréo (2 canaux sonores) avec 3 sources audio (piano, violon et basse), notre matrice de mélange *A* est donc de dimension (2,3).

Le fait qu'un signal soit considéré comme parcimonieux se traduit par le fait que, dans une certaine base (ou plus généralement, un certain dictionnaire), seul certains de ses coefficients sont significatifs, les autres étant égaux à 0. Mathématiquement, cela se traduit par :

$$\mathrm{x} = \mathrm{C_x}.\Phi$$

avec $\mathrm{C_x}$ le vecteur reçu parcimonieux, et Φ le dictionnaire choisi, de dimension (K, T), K étant le nombre d'atomes. Cette équation traduit l'étape de synthèse du signal.

Lors du passage dans le domaine où le signal est parcimonieux, nous introduisons C_s, défini par :

$$C_x \approx A.C_s$$

C_s représente le signal source dans le domaine choisi précédemment. Dans cette équation, l'approximation vient du fait que nous négligeons la représentation du bruit dans le domaine choisi.

Nous obtenons $\mathrm{C_s}$ par résolution de problème inverse, et nous pouvons alors synthétiser le signal source :

$$\mathrm{s} = \mathrm{C_s}.\Phi$$

2. Analyse de nos signaux

Dans cette partie, nous détaillerons les procédés par lesquels nous avons traités notre signal de départ.

2.1. Génération du signal audio

Afin de travailler sur la séparation de source audio, nous avons décidé de générer un signal nous-même afin de pouvoir le manipuler et tester tous nos algorithmes. Nous avons donc, à l'aide du logiciel FL Studio 20, généré différentes mélodies pour chaque instrument et à l'aide du logiciel Audacity, nous avons compiler ces différents instruments en les positionnant spatialement à des endroits différents.

Concrètement et après plusieurs tests, nous avons décidé de prendre 3 instruments qui se distinguent relativement facilement les uns des autres, un piano, une basse et un violon. Le piano se situe à gauche, le violon au centre et la basse à droite (avec des angles de -70° et 70°). Les signaux sont générés au format .wav afin d'être ensuite utilisé sur Matlab. Nous avons donc à notre disposition les signaux bruts avec un seul instrument pour pouvoir comparer, par la suite, ces enregistrements bruts avec les signaux que nous allons récupérer suite à la séparation de sources.

2.2. Changement de domaine

Les signaux audios ne sont à fortiori pas parcimonieux, ce qui rend leur étude plus difficile. Pour enfoncer le clou, notre signal est un problème à dominantes multiples : les sources sont actives simultanément. Afin de se ramener à un cas *sparse*, nous allons utiliser une transformation qui rendra notre signal parcimonieux.

Plusieurs options se présentent à nous : la *Modified Discrete Cosine Transform*, ou la *Short-term Fourier Transform*. Ces options semblent toutes deux viables au préalable, ainsi nous allons les comparer tout au long des étapes de notre script.

Sur Matlab, la STFT est assurée par la fonction spectrogram(), tandis que la MDCT est assurée par la fonction éponyme. Notons tout de même que ces fonctions nécessitent l'utilisation d'une fenêtre de calcul, pour séparer l'évaluation de la transformée sur le signal. Dans un premier temps, nous prendrons une simple fenêtre rectangulaire à 1024 composants.

Contrairement aux notations suivies dans le handbook de Rémi Gribonval et Sylvain Lesage, les coefficients Cx obtenus sont sous la forme d'une matrice : en effet, cette matrice représente un vecteur de coefficients par fenêtre évaluée.

Nous sommes partis du principe que la MDCT et la STFT rendent nos données parcimonieuses, mais nous avons le moyen de le vérifier. En affichant les histogrammes de nos coefficients, nous constatons un pic dans un proche voisinage de 0, mais aussi quelques coefficients non nuls : sans trop d'approximations, nous pouvons affirmer que nos données sont belles et bien parcimonieuses.

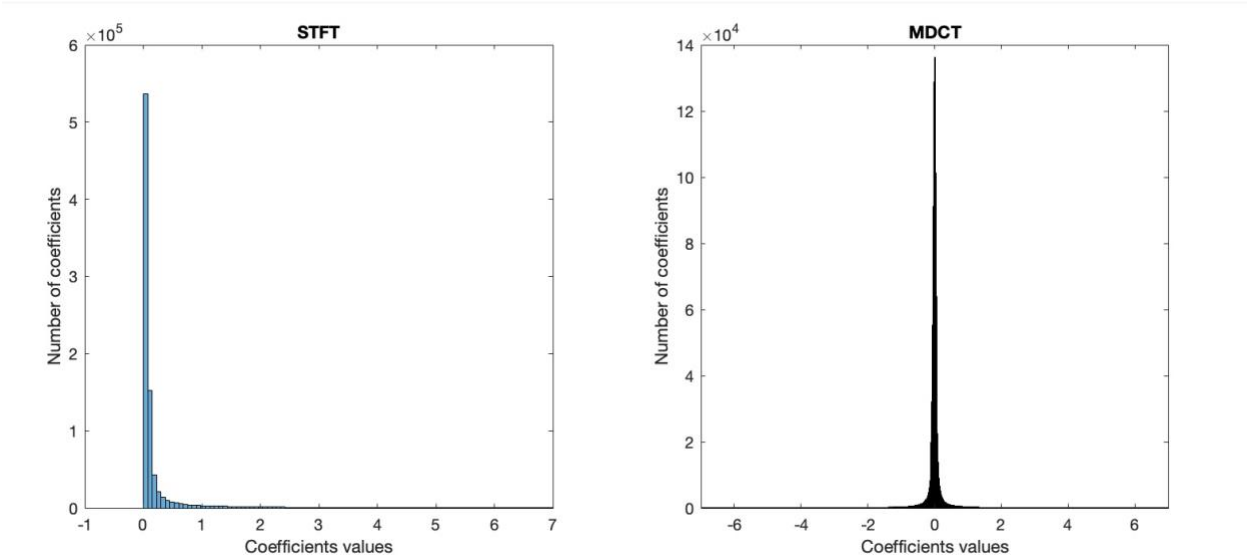

Figure 1: Histogrammes des coefficients issus de nos transformations (nombre d'occurrences en fonction de la valeur)

Après avoir obtenus nos coefficients, et s'être assuré de leur parcimonie, nous allons afficher les scatter plot d'un canal en fonction de l'autre. De ces nuages de points, nous devrions pouvoir distinguer autant de droites s'entrecoupant en (0,0) que de sources à estimer : c'est particulièrement visible dans notre cas, étant donné que le fichier audio a été pensé pour exacerber ce phénomène.

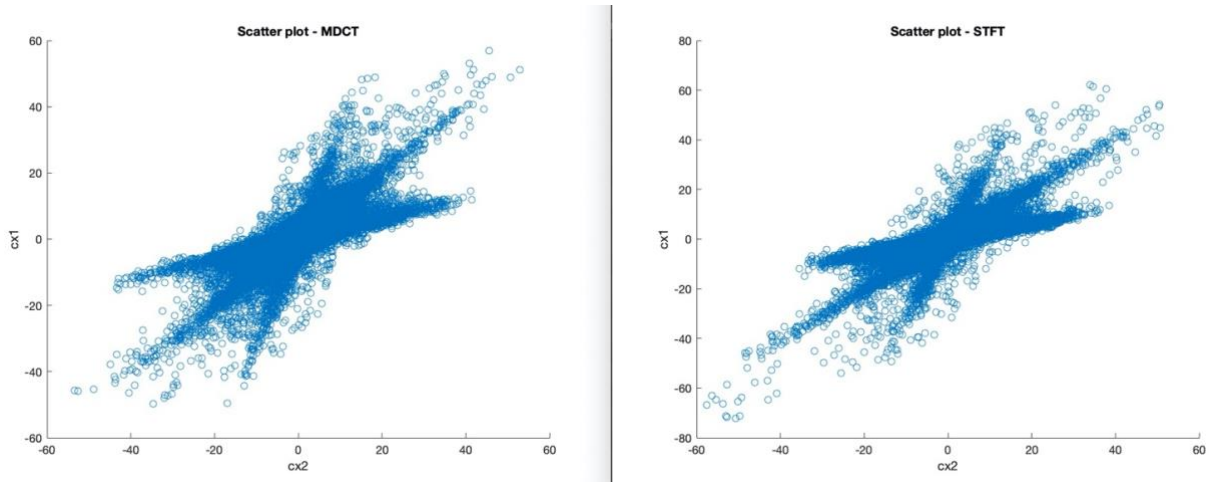

Figure 2: Scatter plot *(cx1 en fonction de cx2)*

Il nous faut maintenant estimer la matrice de mélange, liées aux directions trouvées sur ces scatter plot.

2.3. Estimation de la *Mixing Matrix*

Nous avons dû manipuler une matrice de mélange que l'on notera A. Ses coefficients renferment des informations relatives aux différentes directions dans lesquelles se trouvent chacune des sources (dans notre cas 3 sources). Maintenant que les échantillons sont parcimonieux, nous devons regrouper les points de la matrice des coefficients (désormais en temps-fréquence) par un algorithme de clustering (regroupement). Le résultat sera alors une nouvelle matrice Cx entièrement déterminé par les sources une fois parcimonieuses (cx1 et cx2)

$$C_{\mathbf{x}}(k) = \rho(k) \cdot [\cos\theta(k), \sin\theta(k)]^T$$

avec

$$\begin{cases} \rho(k) := (-1)^{\varepsilon} \sqrt{|c_{\mathbf{x}_1}(k)|^2 + |c_{\mathbf{x}_2}(k)|^2}, \\ \theta(k) := \arctan(c_{\mathbf{x}_2}(k)/c_{\mathbf{x}_1}(k)) + \varepsilon\pi, \end{cases}$$

Un histogramme des valeurs de thêta permet graphiquement d'isoler spécifiquement 3 directions desquelles sont émises nos 3 signaux sources (dans notre cas nous obtenons le schéma suivant après **pondération et lissage**).

Ce lissage est effectué conformément au Handbook, selon :

$$\text{for a set of discrete angles on a grid } \theta_\ell = \ell\pi/L, 0 \le \ell \le L$$

$$H(\theta_\ell) := \sum_{k=1}^{K} f(|\rho(k)|) \cdot w(\theta_\ell - \theta(k))$$

Avec f($|rho|$) = $|rho|$ et w(x) = $exp(-\frac{x^2}{sigma})$, sigma = 0.001

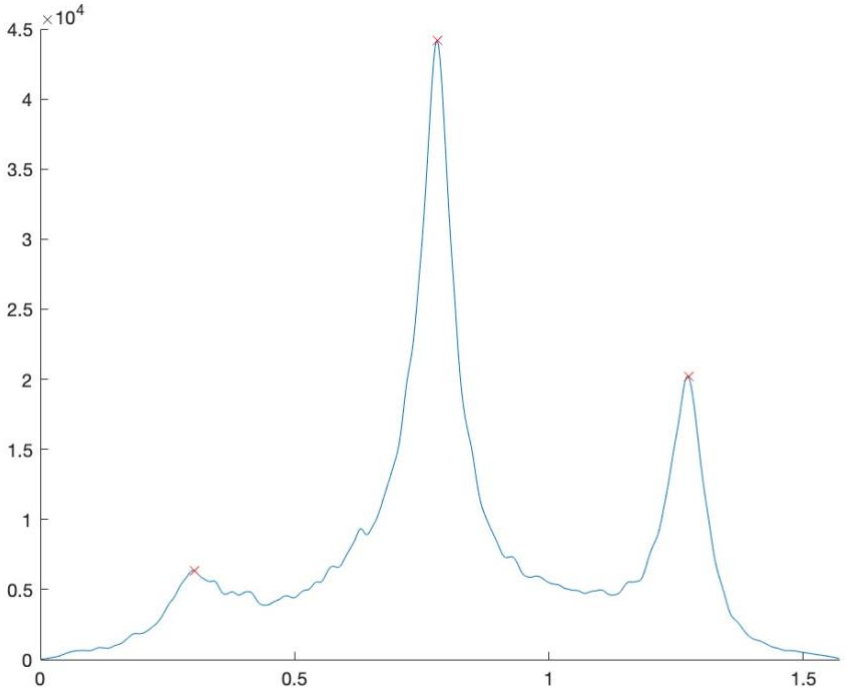

Figure 3: Histogramme de thêta (nbr d'occurrences en fonction de la direction en rad)

Dans le cadre des manipulations MDCT que nous effectuons, nous pouvons alors facilement faire une estimation de la matrice de mélange Â telle que :

$$\hat{\mathbf{A}}_n := [\cos\hat{\theta}_n \sin\hat{\theta}_n]^T$$

Malheureusement, nous n'avons pas assez d'informations sur les conditions initiales du problème physique pour pouvoir estimer la matrice de mélange correspondant à la STFT. Pour les étapes suivantes, nous utiliserons la MDCT uniquement.

2.4. Séparation dans le domaine parcimonieux

Pour séparer les différentes sources, nous avons utilisé deux méthodes. Naïvement, et pour s'assurer que nous faisions un pas dans la bonne direction, nous avons évalué les coefficients Cs de nos sources avec la pseudo-inverse de notre matrice de mélange (*Least-Squares*). Ainsi, on obtient une matrice 3xT, une ligne par source.

Ensuite, nous avons utilisé l'Orthogonal Matching Pursuit, là encore pour obtient une matrice 3xT.

2.5. Retour dans le domaine de base – Reconstruction

Après avoir généré, analysé, et traité notre signal, nous avons pu isoler différentes sources. Il est maintenant temps de revenir dans notre domaine de départ, le domaine temporel, afin de pouvoir par exemple écouter chacune de nos sources individuellement, et les comparer avec les signaux dont ils sont issus. La reconstruction s'établit à partir de l'estimation des différents vecteurs Cs estimés dans l'étape précédente, par la relation :

$$s = C_s.\Phi$$

En pratique, cela se traduit par une opération inverse en MATLAB (imdct ou istft, selon le dictionnaire Φ choisi). Dans notre cas, le dictionnaire se confond avec la matrice de mélange A

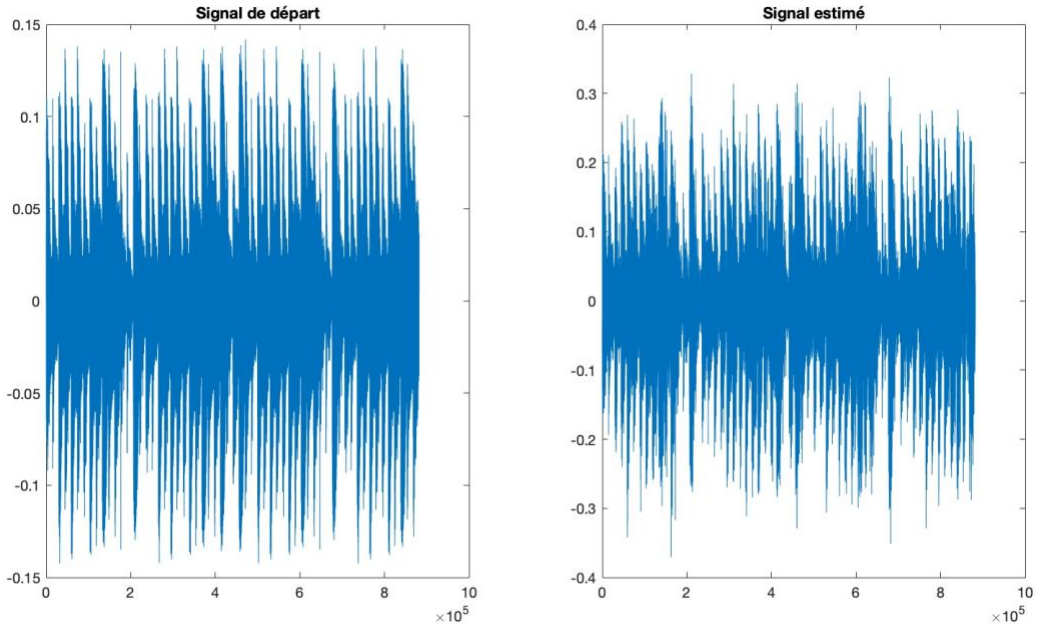

Figure 4: Comparaison source seule / source reconstruite - Basse

Le résultat est encore plus parlant à l'oreille nue, où l'on entend seulement la source voulu (bien que bruitée).

3. Ouverture – Autres applications, autres méthodes, produits existants

3.1. Compression

Pour compresser certains signaux audios, une méthode consiste à sélectionner les informations non perceptibles, donc non pertinentes, et de les supprimer. La plupart des algorithmes de compression avec perte utilisent des transformées telles que la transformée en cosinus discret modifiée (MDCT).

3.2. Réduction de bruit et isolation de la voix

Une autre méthode pour la séparation de sources serait d'utiliser un réseau de neurones, entrainé à séparer différent type de sources, et à ne reconstruire que les sources d'intérêt. Avec assez de puissance de calcul et une latence extrêmement faible, il est possible de faire ce traitement en temps-réel.

La société NVIDIA® conçoit des processeurs graphiques, utilisés dans les ordinateurs grand public ainsi que dans les serveurs de calculs intensifs. Leurs derniers processeurs haut de gamme incluent des unités de calculs dédiées au machine learning, avec plusieurs applications grand public, dont une solution pour les utilisateurs de micro en milieu bruyant. RTX Sound® est une solution logicielle permettant d'utiliser les unités de calculs pour isoler la voix d'un utilisateur lors d'une diffusion en live. Les résultats sont impressionnants.

Conclusion

Ce sujet a été particulièrement intéressant car il nous a permis d'aborder une technologie toujours en développement et dont les applications ne se limitent pas uniquement aux sources audios mais à de nombreux autres domaines que nous rencontrons sûrement dans nos futures vies d'ingénieurs. Le fait que nous ayons obtenus des résultats non pas seulement acceptables mais aussi dépassant nos attentes nous donne le sentiment d'un travail abouti dont nous sommes très fiers.