

# Analyse de données audionumériques

## Fiche de Travaux Pratiques numéro 2

### Masquage Binaire

Mathieu Lagrange & Grégoire Lafay

27 février 2019

Ce document, les scripts Matlab ainsi que les fichiers audionumériques sont disponibles ici : <https://github.com/mathieulagrange/datasim/tps/tpIbm>

Par convention, les fonctions directement disponibles via Matlab sont en italique. Une liste des commandes Matlab utiles pour ce TP est donnée en appendice.

Ce TP est la suite directe du TP précédent. Il s'agira ici de séparer l'accompagnement des extraits musicaux possédant une partie chantée, afin d'améliorer la capacité de notre espace de description à séparer les extraits musicaux de "voix" et "musique".

## 1 Masquage binaire

On essaie ici de se rendre compte de la pertinence du masquage binaire pour extraire des signaux à partir d'une polyphonie.

### 1.1 Chaîne d'analyse / synthèse

En utilisant les fonctions `stft(y,nfft,nfft,0,sr)` et `istft(Y,nfft)`, implantez une chaîne d'analyse / synthèse. Ici  $y$  est un signal ayant pour spectre  $Y$ ,  $nfft$  est la taille de la fft

Testez cette chaîne en :

- en implémentant un effet "voix robotique" en mettant le spectre de phase à 0.
- en implémentant la fonction `fltr` qui prend en entrée un spectre de magnitude et rend en sortie ce même spectre de magnitude filtré. La fonction devra réaliser un filtrage passe-bande à la fréquence centrale  $f_c$  (paramètre d'entrée de la fonction `fltr`).

Que se passe-t-il si vous n'utilisez pas la phase (fonction *angle*) lors de la re-synthèse?

### 1.2 Masque

En utilisant comme oracle les spectrogrammes de la voix et de l'accompagnement, respectivement issus des fichiers `bent_voice.wav` et `bent_music.wav`, générer le masque optimal permettant d'obtenir une estimation du signal de voix. Appliquez le au signal de mixé et écoutez le résultat.

### 1.3 Critères de qualité

Afin de comparer la voix estimée grâce à cette méthode avec la voix originale, implémentez la fonction `spectralSnr` qui prend en entrée le spectrogramme du signal d'origine et le spectrogramme du signal estimé, et donne une valeur *snr* rendant compte de la qualité de la séparation.

$$snr = 20 \log \left( \frac{\sum_{i,j} s_o(i,j)^2}{\sum_{i,j} (s_o(i,j) - s_e(i,j))^2} \right)$$

Avec  $s_o$  le spectrogramme de magnitude du signal d'origine et  $s_e$  le spectrogramme de magnitude du signal estimé. La valeur obtenue fournit une borne supérieure car dans ce cas, le masque est optimal.

## 1.4 Réduction des a priori

On suppose ici que nous ne disposons pas de connaissances sur les propriétés spectrales de la voix chantée. En revanche, on dispose comme a priori d'un spectre moyen de l'accompagnement (Ce qui est vrai dans le cas des extraits avec et sans voix chantée).

Calculer ce spectre moyen et l'utiliser pour générer un masque binaire selon la formule suivante :

$$m(i, j) = s_{mix}(i, j) * \alpha > s_m(i)$$

où  $s_m(i)$  est le spectre moyen de musique à l'indice de fréquence  $i$  et  $\alpha$  est un paramètre à déterminer.

Faire varier  $\alpha$  de 0 à 1 par pas de 0.1 pour en déterminer la valeur optimale au sens de la métrique  $snr$ . Affichez l'évolution de  $snr$  en fonction de  $\alpha$ . Écoutez les résultats pour plusieurs valeurs de  $\alpha$ . La métrique est-elle en accord avec votre perception ?

## 2 Amélioration de l'espace de description en utilisant le masque binaire

Dans cette partie, il s'agira d'améliorer l'espace de description créée lors du TP 1. Il faut donc avoir fini le TP 1 avant de passer à cette partie, du moins pour les spectres et les Mel. Finissez ce TP si ce n'est pas le cas.

### 2.1 Utilisation directe du masque binaire

1. Appliquez le masque binaire aux extraits musicaux possédant une partie chantée, afin d'en supprimer l'accompagnement. Créez une nouvelle banque de données que vous sauvegarderez dans le dossier singerDb2. Ce dossier contiendra les 40 extraits sans partie chantée et les 40 extraits avec partie chantée obtenus après application du masque binaire.
2. Faites varier le type de descripteur en utilisant le spectrogramme, les mels et les MFCCs, et déterminez la meilleure configuration. Pour ce faire vous utiliserez la distance euclidienne (et non cosinusoidale) ainsi que la valeur  $\alpha$  trouvée dans la section 1.4.
3. Une fois la meilleure configuration déterminée, optimisez la valeur  $\alpha$  en utilisant la métrique donnée par precAtK. Est-ce que le fait de supprimer l'accompagnement améliore les performances mesurées par precAtK ? Vérifiez ce point en refaisant les sections 2.2 et 2.3 du tp précédent.

ATTENTION, l'application du masque binaire fait que certaines valeurs du spectre sont mises à zéro. Si la représentation choisie est le spectre, on ne doit pas utiliser ces valeurs lors du calcul des distances. Modifiez mydist en conséquence, en lui passant en paramètre le descripteur utilisé.

## A Commandes Matlab utiles

### A.1 Debug

Il est **indispensable** d'utiliser la commande *dbstop if error*. Elle permet à matlab de passer automatiquement en mode "Debug" lorsqu'une erreur est rencontrée.

### A.2 Divers

- load : permet de charger des données à partir d'un fichier ".mat"
- dir : charge la liste des fichiers présents dans un répertoire
- hist : permet de calculer un histogramme des valeurs d'une matrice
- repmat : permet la réplication de matrices
- imagesc : permet de visualiser les valeurs d'une matrice
- audioplayer : permet d'écouter un signal
- unique : sélectionne les éléments uniques d'un vecteur

### A.3 Regroupement

- pdist : produit un vecteur de similarité
- squareform : convertit un vecteur de similarité en une matrice
- kmeans : implantation native de l'algorithme "k-means"

### A.4 Analyse en composantes principales

- (pc,score) = princomp(V)
- score est une matrice de la même taille que V contenant les nouvelles coordonnées des données dans le système défini par les composantes principales
- plot(score(:,1),score(:,2),'+') affichera les données en fonction des deux axes expliquant le mieux la variance de ces dernières.

### A.5 Documentation

- doc <command> : affiche des informations à propos de la commande <command>.
- lookfor <keyword> : liste les fonctions ayant le mot-clé <keyword> au sein de son nom ou de sa description