

Analyse de données audionumériques

Fiche de Travaux Pratiques numéro 3

NMF

Mathieu Lagrange & Grégoire Lafay

6 mars 2019

Le cours portant sur la NMF est disponible ici :
<https://github.com/mathieulagrange/datasim>

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 précédant. 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".

Dans un premier temps, il s'agira de prendre en main un outil de factorisation de matrice pour la séparation de sources. Une fois cette prise en main effectuée, il s'agira d'utiliser cet outil pour réhausser la partie voix de nos enregistrements de musique.

1 Application de la factorisation de matrices non-négatives

L'objectif ici sera d'utiliser la NMF (*non-negative Matrix Factorization*) afin de séparer les voix et l'accompagnement. On commencera par utiliser la NMF pour séparer les signaux `bent_voice.wav` et `bent_music.wav`, mixés dans un même signal.

La NMF est une opération qui décompose une matrice V en deux matrices W et K appelées respectivement "dictionnaire" et "activation".

$$V \approx WH$$

Ici V est de dimension $freq \times time$, W de dimension $freq \times r$ et H de dimension $r \times time$.

La NMF minimise une fonction de coût exprimée comme une divergence entre la cible à reconstruire V et la reconstruction WH .

$$D(V|WH) = \sum_{i,j} d(V_{ij} | (WH)_{ij})$$

En ce qui concerne le traitement de données audio, la divergence d en question peut être la distance euclidienne (on parle alors de distance du fait de la symétrie), la divergence de Kullback-Leibler, ou encore de la divergence d'Itakura-Saito. Ces trois fonctions impliquent des règles d'update différentes pour la NMF. Cependant, il est possible de les généraliser en utilisant les β -divergences.

Les β -divergences forment une famille de fonctions de distorsions. Pour tout $\beta \in \mathbb{R}$ et tout $x, y \in \mathbb{R}_{++}$, la β -divergence entre x et y est égale à :

$$d_{\beta}(x|y) = \frac{1}{\beta(\beta-1)}(x^{\beta} + (\beta-1)y^{\beta} - \beta xy^{\beta-1})$$

En prenant la limite, on obtient pour $\beta = 0$, $\beta = 1$ et $\beta = 2$, respectivement la divergence d'Itakura-Saito (eq. 1), la divergence de Kullback-Leibler (eq. 2) et la distance euclidienne (eq. 3).

$$d_{\beta=0}(x|y) = \frac{x}{y} - \log \frac{x}{y} - 1 \quad (1)$$

$$d_{\beta=1}(x|y) = x \log \frac{x}{y} + y - x \quad (2)$$

$$d_{\beta=2}(x|y) = \frac{1}{2}(x - y)^2 \quad (3)$$

Les équations d'update de la NMF sont alors :

$$W \leftarrow W \otimes \frac{((WH)^{\beta-2} \otimes V)H^T}{\max((WH)^{\beta-1}H^T, \xi)}$$

$$H \leftarrow H \otimes \frac{W^T((WH)^{\beta-2} \otimes V)}{\max(W^T(WH)^{\beta-1}, \xi)}$$

où \otimes désigne le produit terme à terme ($.*$ en matlab), ξ une petite valeur (utiliser eps en matlab).

2 Utilisation de la NMF en mode oracle

On parle de mode oracle lorsque, comme pour le masque binaire idéal, on dispose des éléments séparés pour constituer une connaissance a priori des éléments à séparer.

1. Afin de pouvoir séparer le signal mixé composé de **bent_voice.wav** et **bent_music.wav**, il nous faut d'abord calculer les templates (éléments du dictionnaire W) de **bent_voice.wav** et **bent_music.wav**. Pour ce faire, implémentez la fonction `[W,H,err]=nmf_beta(V,r,beta)` qui prend en entrée V , un spectrogramme de magnitude, r le nombre d'éléments du dictionnaire, et β . La fonction rend en sortie la reconstruction de $V \approx WH$, ainsi que l'erreur de reconstruction $err = D(V|WH)$.
2. Calculez les templates W_m et W_v pour respectivement les signaux **bent_music.wav** et **bent_voice.wav**. On se limitera à 100 itérations.
3. Créez un dictionnaire complet W_{all} en concaténant W_m et W_v .
4. Nous allons maintenant séparer les signaux de voix et musique du signal mixé. Pour ce faire modifiez la fonction `nmf_beta` afin qu'elle puisse prendre en entrée le dictionnaire W_{all} . Le but ici est de fixer le dictionnaire W , afin de ne calculer que la matrice d'activation H . Il ne faut donc plus mettre à jour W quand W_{all} est passé en paramètre.
5. Appréciez l'impact des variations de β et r en a) affichant err en fonction de β et r , b) en écoutant les résultats, et c) en calculant la métrique snr . Il est demandé d'utiliser les deux méthodes de séparation vues en cours : l'approche synthèse et l'approche filtrage.

2.1 Critère 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.

3 Rehaussement de la partie chantée (approche supervisée)

Le but ici est de reprendre la base de données de chanteurs et d'appliquer la NMF de diverses manières pour rehausser la partie chantée. On prendra la divergence d'Itakura Saïto.

- Apprendre les éléments de dictionnaire sur la partie musicale.
- Décoder la partie voix musicale de la partie chantée avec ces éléments de dictionnaire. La soustraire pour obtenir une estimation de la voix chantée.
- Faire varier le nombre d'éléments de dictionnaire de 10 à 40 par pas de 10 et en apprécier l'influence sur les performances en utilisant la métrique donnée par `precAtK`.

4 Rehaussement de la partie chantée (approche semi-supervisée)

Le but ici est de d'utiliser un paradigme semi-supervisé pour le rehaussement de la partie chantée.

- Prendre les éléments de dictionnaire utilisés dans la section précédente avec le nombre d'éléments ayant amené les meilleurs performances.
- Construire un dictionnaire concaténant les éléments de dictionnaire modélisant l'accompagnement musical M et y ajouter un nombre à déterminer d'éléments de dictionnaire V comportant des valeurs aléatoires tirées dans le même intervalle de valeurs que M . Le dictionnaire final est donc $W = [MV]$.
- Adapter l'algorithme NMF pour qu'à chaque itération, seule la partie dédiée à la voix chantée soit effectivement mise à jour en effectuant l'optimisation complète de W_k , k étant l'indice d'itération, puis en remplaçant la partie M_k par M . Ainsi, seule la partie dédiée à la voix sera mise à jour lors de l'optimisation.
- Faire varier le nombre d'éléments du dictionnaire V de 10 à 40 par pas de 10 et en apprécier l'influence sur les performances en utilisant la métrique donnée par `precAtK`.

5 Questions de réflexion

- Calculez $d_\beta(\lambda x | \lambda y)$. En déduire un avantage certain de la divergence d'Itakura-Saito ($\beta = 0$) pour l'analyse de données audio (Pensez à l'allure générale d'un spectre audio).
- Quel est selon vous le principal défaut des templates utilisés dans la version de la NMF que nous venons de voir ?

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 réplique 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