Compte rendu de travaux pratiques

Alexis Guilloteau, Van Phan Quang

29 Novembre 2010

Introduction

Ce travail est une mise en pratique d'une analyse de scène pour l'extraction de caractéristiques audio (ici MFCC). Dans un premier temps, nous observons le phénomène de malédiction de la dimensionalité. Ensuite nous avons implémenté un système simple de recherche par l'exemple. Enfin, nous expérimentons la méthode de masquage binaire à notre système afin d'en améliorer les performances.

1 Malédiction de la dimensionalité

Dans cette partie on cherche a représenter l'histogramme des distances qui sépare chaque point dans des espaces à n-dimension. Avec $n \in [2, 10, 100, 1000]$. La fonction *pdist* permet de calculer les distances euclidiennes entre chaque point disposé de façon aléatoire dans un espace, l'histogramme des distances peut donc être tracé (figure 6).

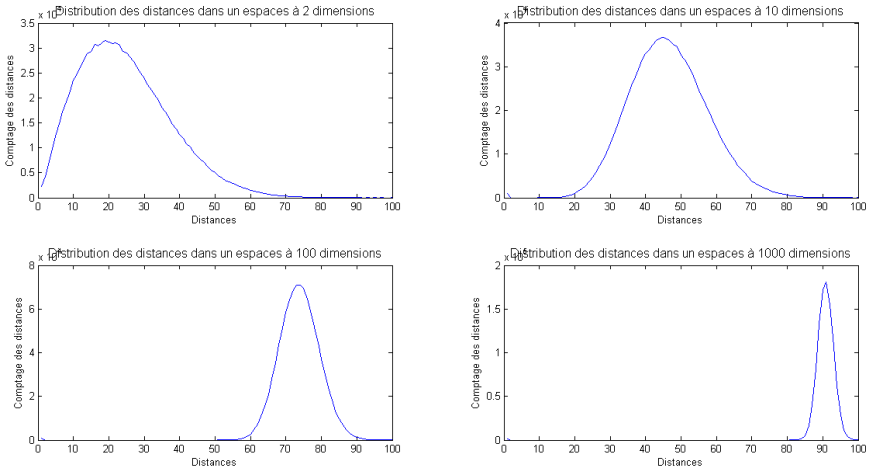

FIGURE 1 – *Distribution des distances de 1000 points choisis aléatoirement selon une distribution normale dans des espaces à 2, 10, 100 et 1000 dimensions. Asymptotiquement, cette distribution donne un dirac.*

On observe que plus la dimensionalité de l'espace augmente plus la distance entre les points est équirépartie. Ceci a une conséquence directe sur les applications pratiques décrites. Les objets utilisés ne doivent pas être représentés par des vecteurs caractéristiques à trop grande dimension, si l'on veut qu'ils soient discriminant.

2 Recherche par similarité en fonction du chanteur

2.1 Exploration

La fonction computeSpec calcule les spectrogrammes des parties musique et chantée de chaque fichier. A partir de ces spectrogrammes, nous changeons l'échelle des fréquences à l'aide de la fonction audspec. Cette fonction permet de passer en échelle logarithmique des fréquences. Ensuite, l'échelle des puissances a été modifiée avec la fonction log. Puis la fonction spec2cep calcule la DCT (Discrete Cosine Transform) du spectre modifié. Enfin, les coefficients obtenus sont moyennés sur toute la durée du signal. On obtient ainsi 13 coefficients MFCC résumant le spectre du signal.

2.2 Métriques d'évaluation

La fonction rankEval permet d'évaluer le taux de réussite de notre système à retrouver les autres chansons d'un même chanteur. Cette fonction compare ses résultats par rapport à la vérité terrain. Cette fonction utilise les matrices de similarité obtenues avec la fonction pdist. La figure suivante présente les matrices de similarité pour la voix et la musique.

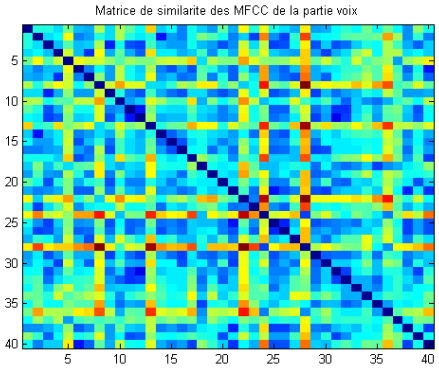

FIGURE 2 – *Matrice de similarité pour les 40 chansons à partir des MFCC calculés sur la partie chantée.*

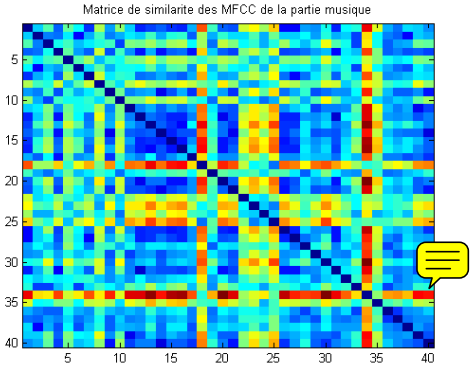

FIGURE 3 – *Matrice de similarité pour les 40 chansons à partir des MFCC calculés sur la partie musicale.*

2.3 Premiers résultats

Dans cette partie, nous faisons varier les différents paramètres de manière automatique à l'aide des fonctions fournies getVariants, iterateOverVariants et singerSimularity. Les paramètres sur lesquels nous jouons sont le choix de l'observation (musique ou voix), le passage ou non en échelle de fréquence mel, le passage ou non des puissances en décibel, le calcul ou non de la DCT, avec ou sans coefficients delta, avec ou sans variance des coefficients MFCC, en normalisant ou pas la représentation obtenue, en changeant le calcul de la distance dans la fonction pdist. Les meilleurs résultats ont été obtenus en calculant les MFCC sur la voix, en rajoutant la variance des MFCC et en normalisant les différents coefficients. Le taux de réussite ainsi obtenu est de 0.1917.

observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=1	variance=0	normalized=0	distance=1	Ranking precision: 0.1083
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=0	variance=0	normalized=0	distance=1	Ranking precision: 0.1167
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=0	variance=1	normalized=1	distance=1	Ranking precision: 0.1250
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=0	variance=1	normalized=0	distance=1	Ranking precision: 0.1417
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=1	variance=1	normalized=0	distance=1	Ranking precision: 0.1500
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=0	variance=0	normalized=1	distance=1	Ranking precision: 0.1667
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=1	variance=0	normalized=1	distance=1	Ranking precision: 0.1667
observation=0	mel=1	decibel=1	dct=0	dynamic=1	delta=1	variance=1	normalized=1	distance=1	Ranking precision: 0.1917

FIGURE 4 – *Sur ces résultats, on observe qu'il faut normaliser les coefficients, rajouter la variance, conserver mel et décibel.*

observation=1	mel=0	decibel=0	dct=0	dynamic=1	delta=0	variance=0	normalized=1	distance=1	Ranking precision: 0.0833
observation=1	mel=0	decibel=0	dct=0	dynamic=1	delta=1	variance=0	normalized=1	distance=1	Ranking precision: 0.1000
observation=1	mel=0	decibel=0	dct=1	dynamic=1	delta=0	variance=0	normalized=1	distance=1	Ranking precision: 0.1000
observation=1	mel=0	decibel=0	dct=1	dynamic=1	delta=1	variance=0	normalized=1	distance=1	Ranking precision: 0.1000
observation=1	mel=0	decibel=0	dct=0	dynamic=1	delta=0	variance=1	normalized=1	distance=1	Ranking precision: 0.1167
observation=1	mel=0	decibel=0	dct=1	dynamic=1	delta=1	variance=1	normalized=1	distance=1	Ranking precision: 0.1417
observation=1	mel=0	decibel=0	dct=0	dynamic=1	delta=1	variance=1	normalized=1	distance=1	Ranking precision: 0.1500
observation=1	mel=0	decibel=0	dct=1	dynamic=1	delta=0	variance=1	normalized=1	distance=1	Ranking precision: 0.1917

FIGURE 5 – *Le tableau de résultats suivant présente le taux de réussite du système en choisissant comme observation la musique. Sur ces résultats on note qu'il est préférable d'enlever les deltas, conserver la variance et les dct, alors que les mel et les décibels ne jouent aucun rôle sur le score final.*

3 Masquage binaire

Dans cette partie il est question de mettre en pratique l'utilisation d'un masque binaire afin de séparer la partie vocale et l'accompagnement d'un signal mixé. Dans un premier temps ce masque est déterminé à partir des oracles mis à disposition. Seulement en réalité ces oracles ne sont pas disponible, il est donc préférable de fournir une méthode de masquage qui soit applicable a des cas réels.

3.1 Chaîne d'analyse / synthèse et masquage

Les fichiers *.wav*, correspondant à la partie vocale et à l'accompagnement, sont extraits puis mixés. Les spectrogrammes des oracles sont alors calculés à partir de la fonction $stft.m$. Un masque est ensuite obtenu en appliquant la structure conditionnel $S_{Voix} > S_{Musique}$. Il est préférable de tester la taille de la fenêtre qui donne le meilleur rendu à l'écoute après masquage. Ce résultat peut également être vérifié à l'aide le la fonction $SpectralSnr.m$. Cette fonction permet d'évaluer la différence entre 2 spectrogrammes supposés identiques. Cette chaine est réalisée et donne le masque de la figure suivante pour la voix.

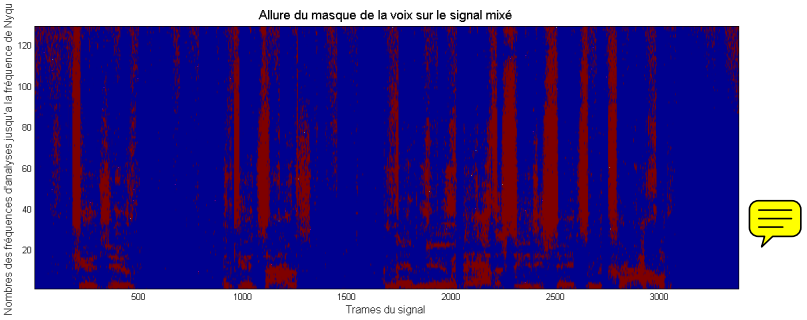

FIGURE 6 – *Matrice de masquage permettant de récupérer la voix du signal mixé.*

Ensuite, à l'aide de la fonction $SpectralSnr.m$, la similitude de la voix obtenue après masquage avec l'oracle est apprécié et donne 0.46 (un 0 signifierait que les signaux sont identiques). Le résultat à l'écoute est plutôt satisfaisant. Bien que le signal de voix résultant soit assez bruité il est correctement séparé de l'accompagnement.

3.2 Réduction des a priori

Dans cette partie la séparation doit être réalisé sans oracles. Etant donné que l'accompagnement est toujours présent, contrairement à la voix, le spectre moyen de l'accompagnement est calculé, à partir du signal mixé en découpant ses parties sans voix, afin de servir de masque par la suite. Il est nécessaire d'affecter un pondération α au signal mixé avant le calcul du masque, et de le comparer au spectrogramme moyen de l'accompagnement. Ensuite il est nécessaire de déterminer la valeur de α qui minimise le résultat de la fonction $spectralSnr.m$ (figure 7).

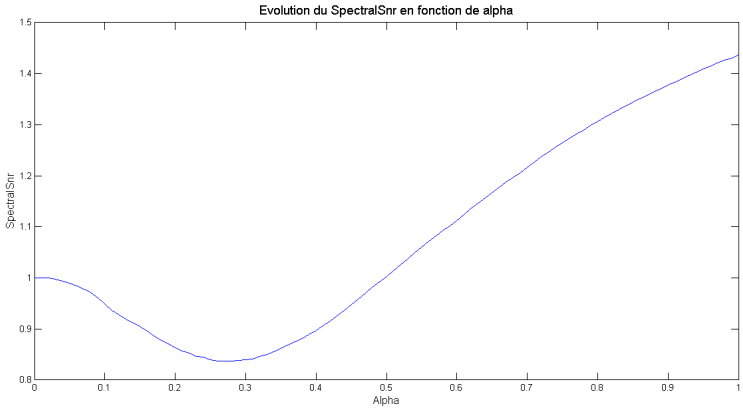

FIGURE 7 – *Evolution des résultats de la fonction spectralSnr.m en fonction du paramètre α. Cette fonction est minimisé pour une valeur de 0.26.*

En faisant varier α de 0 à 1 par pas de 0.01 on obtient le meilleur masque pour une value de 0.26. A l'écoute le signal est nettement plus dégradé que précédément cependant la séparation est bien réalisée. On notera que les signaux très marqués temporellement (coup de caisse-claire) ne sont pas tout à fait retirés.

Conclusion

Dans ce TP nous avons pu voir d'utilisation des MFCC pour caractériser la musique et la voix chantée. La reconnaissance d'un chanteur à partir des MFCC est plus performante en examinant la partie chantée seulement. En rajoutant d'autres caractéristiques, telles que la variance notamment, et en normalisant les coefficients, nous avons obtenu un taux de réussite d'environ 19 %. Cependant, le système n'est performant que dans 80 % des cas. Afin d'améliorer les performances du système la technique du masquage binaire est utilisée pour mieux séparer le signal de la voix à celui de la musique.