Rapport de Stage I5 Rapport de Stage ATIAM

Séparation de la Voix du Locuteur et du Fond Musical dans des Emissions Radiodiffusées

Sylvain ROUSSELLE

13 mars - 13 juillet 2006

Tuteur : Roland BADEAU

ENST
37/39 rue Dareau
75014 PARIS

ESIEE
Cité Descartes - BP 99
2 boulevard Blaise Pascal
93162 NOISY-LE-GRAND Cedex

IRCAM
1 place Igor Stravinsky
75004 PARIS

Remerciements

Merci à Roland Badeau de m'avoir permis de faire ce stage, pour son encadrement et le temps qu'il m'a consacré.

Merci à David, José, Nadir, et Steeve qui m'ont fait une petite place dans leur bureau et sans qui ces quatres mois auraient certainement été moins agréables.

Merci à Fabrice pour sa bonne humeur et ses précieux conseils concernant l'outil informatique.

Enfin, merci à Messieurs Marc Peyrade et Yves Grenier respectivement directeur de l'ENST et responsable du département TSI.

Table des matières

Table des figures

Liste des tableaux

Introduction

Afin de valider mes diplômes d'Ingénieur ESIEE (Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique) et de master recherche ATIAM (Acoustique, Traitement du Signal et Informatique Appliqués à la Musique), j'ai effectué mon stage de fin d'études à l'ENST (Ecole Nationale Supérieure des Télécommunications). J'ai passé quatre mois, du 13 mars au 13 juillet 2006, au sein du département TSI sur le site Dareau de l'école[1] et travaillé, avec mon tuteur Roland Badeau, sur la séparation de sources audio.

Avec l'avènement des nouvelles technologies et l'explosion de l'informatique, le traitement numérique du signal a pris une importance considérable dans le monde d'aujourd'hui. La séparation de sources ne déroge pas à cette règle. Elle consiste, à partir d'un certain nombre d'observations d'une même "scène", à retrouver les différentes sources qui en sont à l'origine. En fonction du domaine d'application, la "scène" peut correspondre à différentes choses. On utilise par exemple la séparation de sources en communications radio. Plusieurs émetteurs émettent des signaux différents, ce sont les sources. Cet ensemble d'ondes forme un signal complexe, la "scène". On le capte à l'aide de plusieurs antennes intelligemment disposées qui nous donnent autant de signaux distincts, les observations. L'enjeu est de reconstituer à partir de ces informations les différents signaux initialement émis. On parle de traitement d'antennes. On utilise aussi la séparation de sources en médecine dans l'analyse des ECGs (ElectroCardioGrammes) pour l'extraction de battements cardiaques fœtaux ou des MEGs (MagnétoEncéphaloGrammes) / EEGs (ElectroEncéphaloGrammes) pour la suppressions des signaux indésirables dans l'étude de l'activité neuronale du cerveau par exemple.

Ce stage est centré sur la séparation des signaux de parole et de musique dans des enregistrements radiophoniques. Les tâches à réaliser étaient les suivantes : étudier les méthodes existantes, choisir celle(s) qui semble(nt) la (les) mieux adaptée(s) au problème, la (les) implémenter, tester et améliorer si possible et nécessaire. Au final, pendant mes quatre mois de stage, je me suis concentré sur une méthode de séparation basée sur des modèles statistiques des sources.

Après une présentation, chapitre 1, du cadre dans lequel j'ai effectué ce stage de fin d'études, je présente, chapitre 2, de manière plus détaillée, la séparation de sources so-

[1] le site principal étant basé rue Barrault.

nores. Je me concentre ensuite sur les différentes étapes de la méthode étudiée : chapitre 3, la manière de modéliser les différents signaux mis en jeu ; chapitre 4, la séparation à proprement parler des signaux lorsqu'on dispose de modèle des sources ; chapitre 5, la méthode d'apprentissage des modèles utilisée et enfin chapitre 6, quelques détails sur l'adaptation des modèles aux mélanges considérés. Le chapitre 7 est lui consacré aux tests et résultats obtenus à l'aide de cette méthode de séparation.

Chapitre 1

Cadre du stage

1.1 Le département Traitement du Signal et de l'Image (TSI)

Le département TSI de l'ENST a pour missions l'enseignement, la recherche et la formation par la recherche dans les domaines du traitement du signal et des images et de leurs applications, en particulier pour les télécommunications.

Il se divise en cinq groupes, chacun contribuant à l'ensemble des missions du département :

- Traitement et Interprétation des Images (TII)
  Ce groupe conduit des recherches sur la mise en oeuvre de schémas complets de traitement, d'analyse et d'interprétation d'images, en particulier de scènes complexes. Les domaines d'application sont l'imagerie médicale, aérienne, satellitaire, radar ou encore la description d'objets tridimensionnels.

- Traitements Statistiques et Applications aux Communications (TSAC)
  Les travaux de ce groupe sont axés sur le signal dans les communications, la séparation de sources ou encore la modélisation statistique des signaux.

- Perception, Apprentissage et Modélisation (PAM)
  Ce groupe étudie le rôle des facteurs humains dans l'accès aux divers types d'informations : parole (reconnaissance), image (psychovision), écrit (structuration de documents), fusion des modalités perceptives dans l'appréhension de l'environnement, interfaces multimodales.

- Codage (COD)
  Le groupe COD s'intéresse aux techniques de compression de sources et à leur adaptation aux applications audiovisuelles/multimédia (compression audio, codage d'images, transmission audiovisuelle, systèmes temps-réel).

- Audio, Acoustique et Ondes (AAO)
  Ce dernier groupe étudie la physique des ondes dans les domaines de l'optique (stockage de l'information) et de l'acoustique (modélisation de la production des

sons, perception, antennes acoustiques).

Mon tuteur Roland Badeau appartient au groupe AAO.

1.2 Le projet INFOM@GIC

1.2.1 Origine

Le projet INFOM@GIC, lancé en décembre 2005, s'inscrit dans le cadre du pôle de compétitivité IMVN (Image, Multimédia et Vie Numérique). Ce dernier est porté par l'Agence Régionale du Développement (ARD) d'Ile de France dont il est le troisième pôle. Consacré aux technologies de l'information et de la communication, il implique de grands groupes (TF1, Lagardère Groupe, France Télécom, Eclair, SFP, TSF), des PMEs, des laboratoires et des institutions (LIP6 Paris VI, l'INA, l'IRCAM, Télécom Paris, CNAM, ESIEE, ENSTA, ENS Louis-Lumière, Gobelins, FEMIS).

1.2.2 Objectif

INFOM@GIC vise à mettre en place, sur une période de trois ans, un laboratoire industriel de sélection, de tests, d'intégration et de validation d'applications opérationnelles des meilleures technologies franciliennes dans le domaine de l'ingénierie des connaissances.

Ce laboratoire s'appuie sur une plate-forme commune qui doit couvrir les grands domaines de l'analyse d'information quelles que soient les sources (données structurées, texte, images et sons) :

- la recherche et l'indexation,
- l'extraction de connaissances,
- la fusion d'informations multimédias.

Elle inclut des applications pour les secteurs de la e-Education et de la gestion des patrimoines culturels numériques.

1.2.3 Acteurs

Les partenaires du projet sont répartis en quatre catégories :

- Industriels : Thalès (coordinateur), EADS, Xerox ;
- PMEs : Bertin, Europlace, FIST (CNRS-ANVAR), Intuilab, Odile Jacob, Pertimm, Temis, Vecsys ;
- Etablissements publics : CEA, CNRS, INA, ONERA ;

– Ecoles et universités : GET/ENST-INT, Paris VI (LIP6, LSTA), Paris VIII (LC&U), Paris IX Dauphine (CEREMADE), Paris XIII (LIPN), Paris-Sud Orsay (LIMSI), Université de Marne-la-Vallée (IGM), CNRS/LACAN.

D'autres partenaires sont en instance d'intégration : Canal+ HiTech, Hi-Store, LIRMM, SINEQUA.

1.2.4 Le sous-projet "Patrimoine Numérisé"

L'ENST intervient principalement dans le sous-projet "Patrimoine Numérisé", piloté par l'INA, pour tout ce qui concerne l'acquisition, la description initiale, l'édition et la publication de contenus audiovisuels. La contribution de l'ENST est répartie dans trois domaines : les outils d'analyse des séquences vidéos, la multimodalité voix-image et les outils de traitement de la bande audio. Ce dernier, auquel mon stage est rattaché, inclut les outils de prétraitement de la bande son, la séparation parole/musique ou encore la segmentation en évènements sonores (parole, musique, applaudissements, etc.).

1.3 Le sujet

Ce stage à pour titre :

Séparation de la voix du locuteur et du fond musical
dans des émissions radiodiffusées.

Cette séparation est une étape dans la transcription de données radiophoniques. En effet, la trancription de signaux radiophoniques est loin d'être aisée parce qu'elle contient des signaux de diverses natures, principalement parole, musique et parole+musique. Une première étape consiste à segmenter, manuellement ou automatiquement, les données en suivant ces trois catégories. Les extraits de parole seule ou de musique seule pourrons dès lors faire l'objet de transcription. Les zones de mélange nécessitent quant à elles un second traitement destiné à extraire les signaux de parole et de musique. C'est à cet endroit de la chaîne de traitement que se positionne mon stage.

Chapitre 2

La séparation de sources

La séparation de sources correspond à une technique de traitement des signaux multicapteurs utilisée dans de nombreux domaines : médecine, radar, audio, etc. On étudie ici uniquement la séparation de sources sonores, objet de ce stage. On distingue deux types de mélanges : les mélanges linéaires instantanés et les mélanges convolutifs. Les méthodes de séparation peuvent être aveugles ou au contraire nécessiter des connaissances *a priori* sur les sources. Enfin, quatre configurations concernant les nombres de sources et d'observations sont possibles :

- déterminée (autant d'observations que de sources),
- sur-déterminée (plus d'observations que de sources),
- sous-déterminée (moins d'observation que de sources),
- monocapteur (un seul capteur quel que soit le nombre de sources).

On se limitera aux mélanges linéaires instantanés, le mélanges convolutifs pouvant faire l'objet d'une étude ultérieure.

Un mélange instantané (invariant temporellement) se traduit par l'équation suivante :

$$x_i(t) = \sum_{j=1}^{N} \lambda_{ij} s_j(t) \tag{2.1}$$

où $x_i(t)$ est l'échantillon à l'instant t de l'observation i, $s_j(t)$ l'échantillon à l'instant t de la source j et λ_{ij} le poids de la source j dans l'observation i.

2.1 Analyse en Composantes Indépendantes (ACI)

L'Analyse en Composantes Indépendantes (ACI) [1, 2] est la méthode utilisée pour la séparation aveugle de sources[1]. Elle est basée sur l'hypothèse, forte mais non dénuée de sens, que les sources sont statistiquement mutuellement indépendantes. Le but est ainsi de trouver les estimations de sources qui présentent une dépendance mutuelle minimale.

[1] Les deux expressions désignent d'ailleurs souvent la même chose.

On dispose de N sources $s_1(t), \ldots, s_N(t)$ et de M observations $x_1(t), \ldots, x_M(t)$. Chaque observation $x_i(t)$ est un mélange des N sources (éq. 2.1). L'indépendance statistique des source s'exprime comme suit :

$$\mathbb{P}(s) = \prod_{i=1}^{N} \mathbb{P}(s_i). \tag{2.2}$$

Considérant le cas déterminé $M = N$, on peut réecrire la formule de mélange sous forme matricielle :

$$x(t) = \Lambda s(t) \tag{2.3}$$

avec $x(t)$ le vecteur $N \times 1$ des observations à l'instant t, $s(t)$ le vecteur $N \times 1$ des sources (supposées indépendantes) à l'instant t et Λ la matrice $N \times N$ de mélange.

Estimer les sources revient dans un premier temps à estimer la matrice de mélange Λ à partir des observations. Ensuite, en considérant que les colonnes de Λ sont linéairement indépendantes, Λ est inversible et on peut facilement obtenir la matrice de séparation $\Psi = \Lambda^{-1}$.

L'ACI recherche les composantes linéaires, d'un ensemble d'observations (cohérentes), dont la dépendance mutuelle est minimale. Elle consiste à résoudre :

$$\hat{\Lambda} = \underset{\Lambda}{\operatorname{argmin}}\, \Gamma(\lambda^{-1} x(t)) \tag{2.4}$$

où $\Gamma(u)$ est une mesure de la dépendance des composantes d'un vecteur aléatoire u.

L'ACI permet de retrouver les sources à quelques indeterminations près, induites par les symétries dans les distributions, concernant le gain et une possible rotation de Λ. De plus, dans le cas stationnaire, l'ACI ne fonctionne pas si plus d'une source est gaussienne (i.e. de densité de probabilité gaussienne). En effet, la résolution de l'équation (2.4) passe par l'estimation d'au minimum deux moments[2] d'ordre pair (le moment d'ordre 1 est nul par hypothèse et les moments d'ordre impaire par symétrie). Pour une source gaussienne, seul le moment d'ordre 2 est non-nul ce qui ne permet pas de résoudre le problème.

La première étape d'une ACI est généralement le blanchiment du vecteur d'observations (on rappelle qu'un vecteur est dit spatialement blanc si sa matrice de covariance est identité). Cette étape permet de réduire la matrice de mélange Λ à une matrice de rotation et simplifie la résolution de l'équation (2.4).

La méthode par ACI peut aussi s'appliquer dans le cas sur-déterminé. La seule différence réside dans le fait que la matrice de mélange ne sera plus carrée et donc inversible. On pourra par exemple utiliser à la place la matrice pseudo-inverse de Moore-Penrose [3].

[2] Généralement on substitue, par soucis de simplicité, les cumulants au moments.

Pour les cas sous-déterminés, l'ACI est un problème mal-posé et l'utilisation d'une information *a priori* sur les sources semble inévitable. L'ACI permet dans certains cas d'estimer une matrice de mélange MxN mais le modèle n'est plus inversible puisqu'on recherche $N > M$ sources. L'enjeu est alors, à partir de M observations et d'une matrice de mélange désormais connue, d'identifier les N sources. On peut par exemple faire l'hypothèse qu'il existe une représentation parcimonieuse (*sparse*) [4, 5] des sources. Dans ce cas, le problème devient solvable parce que, dans un domaine transformé (ex. Fourier), chaque échantillon peut-être, en première approximation, associé à une unique source. On peut aussi estimer les densités des sources puis rechercher la solution qui minimise l'erreur quadratique moyenne ou encore celle qui maximise la loi *a posteriori* [6].

Il existe d'autres méthodes de séparation multicapteurs comme celle détaillée dans [7] qui utilise la Décomposition Modale Empirique (*Empirical Mode Decomposition* - EMD). Cependant ces solutions ne sont pas adaptées au cas très particulier où l'on ne dispose que d'une unique observation.

2.2 Séparation de sources mono-capteur

Voici le cas le plus complexe. Ici on n'a aucune information spatiale puisqu'un seul capteur est témoin de la scène sonore. On peut citer une méthode de séparation aveugle dans ce cas, l'analyse en sous-espaces indépendants (*Independent Subspace Analysis* - ISA)[8]. Cette méthode opère une décomposition en sous-espaces indépendants avant d'appliquer une ACI classique. Cependant, dans le cas général, la séparation ne peut s'effectuer qu'avec certains *a priori* sur les sources à séparer.

Plusieurs méthodes considèrent des critères spectraux pour discriminer deux signaux audios. Karnebäck dans [9] utilise par exemple un critère d'ondulation basse fréquence pour séparer la voix de la musique. Il obtient même de meilleurs résultats en lui associant une représentation MFCC (*Mel Frequency Cepstal Coefficients*). Julie Rosier dans sa thèse [10] caractérise les différentes sources par leur fréquence fondamentale. Dans le même esprit, les auteurs de [11] utilisent des modèles harmoniques des sources pour séparer les différents instruments d'un mélange.

Roweis dans [12] utilise le masquage temps-fréquence. Il divise le signal en différentes sous-bandes fréquentielles et applique des gains différents à chacune d'elles, les valeurs de ces gains étant obtenues par apprentissage sur des signaux séparés. Dans la continuité, une autre méthode utilise des modèles statistiques des sources et un formalisme bayesien [13, 14]. C'est sur cette dernière que l'on décide de s'attarder. En effet, les mélanges audio qui font l'objet de ce stage étant de type radiophonique et surtout monophonique, elle semble être la mieux adaptée à notre problème. La suite de ce rapport permet de la comprendre plus en détails et d'évaluer ses performances. L'ACI présentée plus haut pourra être utilisée dans une phase de post-traitement pour, peut-être, améliorer la séparation des sources.

Le principe général de la méthode choisie est résumé par le schéma Fig. 2.1 pour

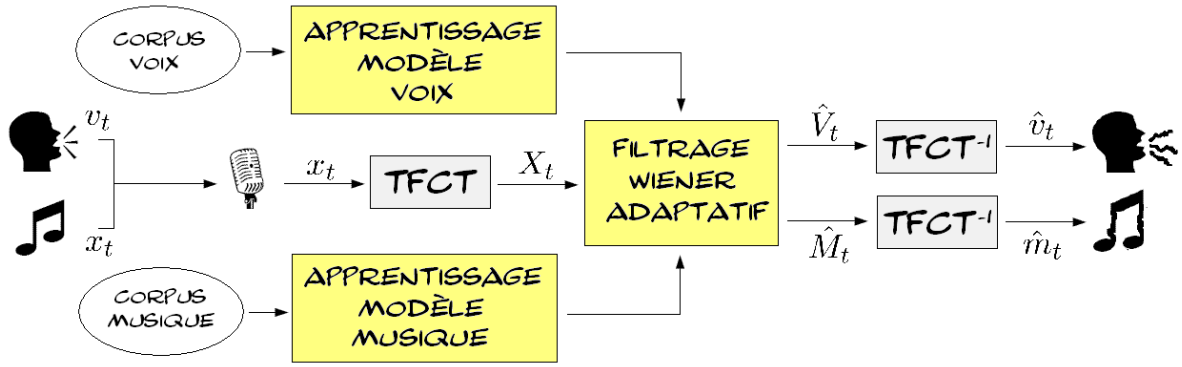


FIG. 2.1 – Schéma de principe de la méthode de séparation monocapteur : v_t une trame de parole, m_t une trame de musique, $x_t = v_t + m_t$ la trame de mélange, X_t le spectre correspondant à la trame de mélange, $\hat{V}_t$ le spectre estimé de la trame de voix, $\hat{M}_t$ le spectre estimé de la trame de musique, $\hat{v}_t$ la trame estimée de voix, $\hat{m}_t$ la trame estimée de musique.

le cas qui nous intéresse : une source voix v et une source musique m. Les différentes étapes seront détaillées dès le chapitre 3 mais on peut d'ores et déjà préciser que le corpus de voix (resp. musique) correspond à un ensemble d'extraits sonores contenant exclusivement de la parole (resp. musique) et sensé être représentatif de la parole (resp. musique) en général. Le bloc "apprentissage" (cf. chap. 5) consiste à calculer les paramètres du modèle de voix (resp. musique). L'opération de TFCT (cf. 3.1) permet de passer dans le domaine fréquentiel dans lequel la séparation s'effectue. L'opération TFCT^{-1} (cf. 4.2) pemet d'en revenir. Le filtrage de Wiener (cf. 4.1) adaptatif est l'étape qui opère concrètement la séparation.

Chapitre 3

Modélisation des signaux

3.1 Transformée de Fourier à Court Terme (TFCT)

La séparation de sources s'effectue dans le domaine fréquentiel. On représente les signaux par leur TFCT dont le calcul peut se décomposer en 3 étapes (Fig. 3.1) :

1. découpage du signal audio en trames de longueur L avec un certain pourcentage de recouvrement ρ ;
2. multiplication des T trames par une fenêtre de pondération w_a ;
3. calcul de la Transformée de Fourier Discrète (TFD) sur N points de chacune d'elles (avec complétion de zéros si $N > L$).

Un recouvrement de 75% signifie par exemple que la trame t+1 contiendra 75% de la trame t et 25% de "nouveauté ".

La formule d'obtention de l'échantillon à la fréquence f de la trame t est :

$$X_t(f) = \sum_{n=1}^{N} w_a(n) x(\tau_a(t) + n) e^{-j2\pi nf} \tag{3.1}$$

avec $\tau_a(t) \triangleq (t-1)N(1 - \frac{\rho}{100})$ l'instant d'analyse.

Le nombre N de points pour la TFD permet de choisir le niveau de précision de la représentation mais doit rester raisonnable pour limiter la complexité. On choisit ici $N = 2048$ qui nous paraît être un bon compromis.

Le choix de la fenêtre et du taux de recouvrement dépend de deux choses :
– la résolution souhaitée (largeur du lobe principal, amplitude des lobes secondaires),
– la possibilité ou non de reconstruction du signal temporel à partir de la TFCT par la méthode OLA (cf. 4.2).

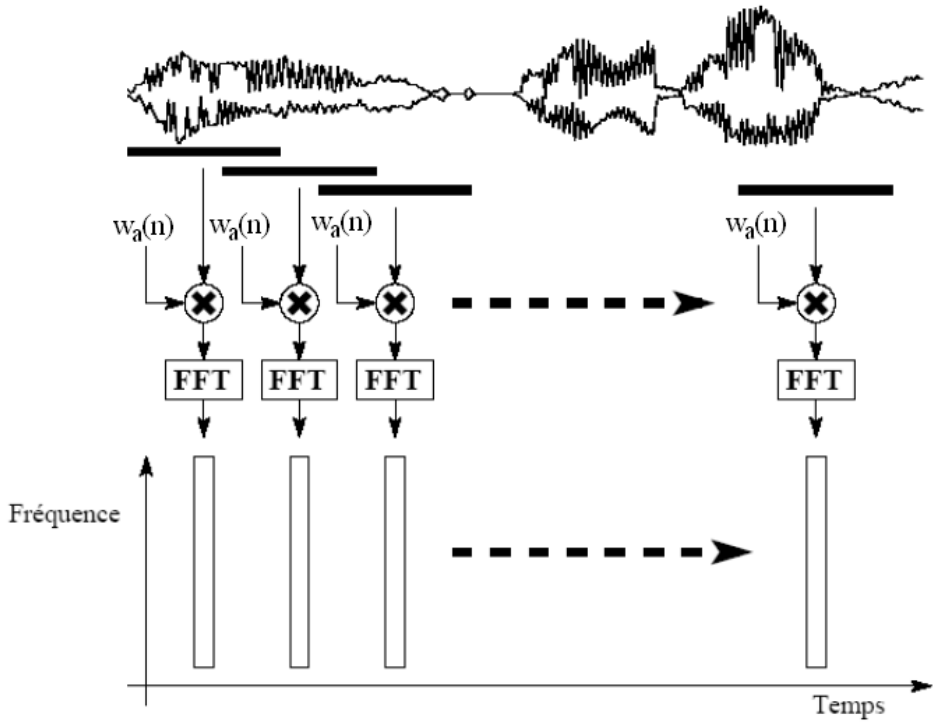


FIG. 3.1 – Schéma de principe de la TFCT.

La fenêtre rectangle est la meilleure en ce qui concerne la largeur du lobe principal mais ses lobes secondaires sont peu atténués. En général, les fenêtres de Hamming ou Hann représentent de bons compromis. Dans [15] , les auteurs mettent d'ailleurs en évidence le fait que les fenêtres de Hann et surtout Hamming donnent de meilleurs résultats pour la séparation de sources que celle de Blackman, par exemple, pourtant souvent appréciée pour sa forte réjection des lobes secondaires. Pour la méthode OLA, il existe une condition (éq. 4.9) sur les fenêtres d'analyse et de synthèse, de reconstruction parfaite. On cherchera à la respecter en choisissant, par exemple, à la fois à l'analyse et à la reconstruction, une fenêtre de type racine carrée de Hamming (FIG. 3.2) avec un recouvrement de 50%.

Pour qu'une représentation en TFCT soit pertinente, il faut que la longueur de la fenêtre d'analyse corresponde plus ou moins à la durée moyenne de stationnarité locale du signal étudié. Les valeurs usuelles pour des signaux audio oscillent entre $10ms$ et $20ms$. Cependant, des tests pratiques de séparation sur plusieurs mélanges de signaux nous ont poussé à choisir $40ms$, la qualité de la séparation diminuant pour des fenêtres plus courtes ou longues. Notons par ailleurs que, dans les articles de référence, les auteurs choisissent $47ms$ [14] voire même $93ms$ [16].

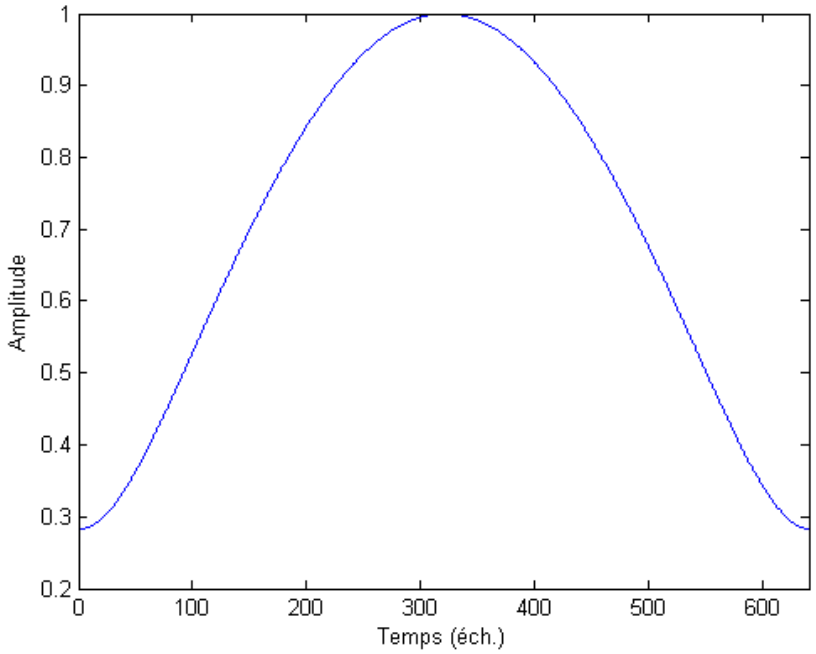


FIG. 3.2 – Fenêtre de Hamming (en racine carrée) de 40ms à 16kHz (soit 640 échantillons).

3.2 Modèles de Mélange de Gaussiennes (MMG) et de Markov Caché (MMC)

3.2.1 MMG

On modélise la TFCT comme un ensemble de réalisations (i.e. les T spectres) d'un processus vectoriel complexe, gaussien et centré, de matrice de covariance Σ. On rappelle la densité de probabilité d'un tel processus :

$$\mathbb{G}(X_t;\Sigma) = \frac{1}{\pi^N det\Sigma}\exp(-X_t^T\Sigma^{-1}X_t) \tag{3.2}$$

où t est l'indice de la trame considérée et N le nombre de points de TFD, soit la longueur du vecteur aléatoire[1] X_t.

Par la suite, on considère uniquement des matrices Σ diagonales. Ce choix est restrictif mais reste pertinent du fait que Σ est la matrice de covariance d'une TFD et que les valeurs d'une TFD sont asymptotiquement (i.e. quand $N = L \to \infty$) décorrelées si :
- le processus est SSL (Stationnaire au Sens Large),
- $w_a \equiv 1$;

conditions qui ne sont tout de même pas pleinement respectées ici.

[1]Par abus de notation, on note X_t une réalisation du vecteur aléatoire X_t

La diagonale de Σ correspond au périodogramme de x_t, à une constante près, et on notera $\Sigma(f)$ sa valeur à la fréquence f.

Dans un deuxième temps on considère que la densité de probabilité se compose de plusieurs gaussiennes, toutes centrées mais de matrices de covariances différentes Σ_1 à Σ_K. Avec ω_k le poids de la $k^{ème}$ gaussienne du mélange, $\omega = \{\omega_1, \ldots, \omega_K\}$, $\Sigma = \{\Sigma_1, \ldots, \Sigma_K\}$, $\theta = \{\Sigma, \omega\}$, la densité de probabilité du processus devient :

$$\mathbb{P}^{mmg}_{X_t}(X_t, \theta) = \sum_{k=1}^{K} \omega_k \mathbb{G}(X_t; \Sigma_k). \tag{3.3}$$

Concernant la génération d'une réalisation, on peut considérer qu'elle se décompose en deux phases :
- le tirage aléatoire de la gaussienne active $q(t)$, la probabilité d'avoir la $k^{ème}$ gaussienne étant ω_k ;
- le tirage du vecteur aléatoire en considérant que la densité de probabilité de processus est réduite à la gaussienne active.

Un tel modèle, bien que plus intéressant que la simple distribution gaussienne, semble encore trop simple pour espérer modéliser fidèlement un signal audio, entre autres parce qu'il ne prend pas en compte son évolution temporelle. C'est donc logiquement qu'on s'intéresse aux chaînes de Markov.

3.2.2 MMC

Afin d'obtenir une représentation plus fidèle, on choisit de modéliser la TFCT comme les observations d'une Chaîne de Markov Cachée (CMC) d'ordre 1. Les observations sont donc les différents spectres X_t de la TFCT et les données cachées du processus de Markov correspondent aux gaussiennes actives $q(t)$ des MMGs (on conserve la même notation).

Ce qui différencie les MMCs des MMGs est le fait que la probabilité d'avoir une gaussienne k active à l'intant t (i.e. $q(t) = k$) dépend de la gaussienne active à l'instant précédent $t-1$. Cette dépendance se traduit par une matrice de transition a qui donne la probabilité de passer de l'état i à l'état j :

$$a_{i,j} = \mathbb{P}\left(q(t) = j | q(t-1) = i\right) \tag{3.4}$$

avec $i, j \in \{1, \ldots, K\}$.

3.3 Prétraitements

Afin d'améliorer la séparation, on peut envisager de prétraiter le signal. Principalement deux raisons nous poussent à le faire. Premièrement, les MMGs et MMCs sont des modèles simples qui restent éloignés de la réalité d'un signal audio. Un prétraitement

peut permettre de faire s'en rapprocher les signaux considérés. Deuxièmement, la qualité d'une séparation est avant tout perceptive. Le plus important est qu'elle semble efficace à l'écoute. L'oreille humaine présentant une dynamique très importante et une réponse en fréquence non-linéaire, un bon SDR (cf. 4.3) n'est pas gage de qualité auditive. Un prétraitement peut permettre d'y remédier.

3.3.1 Accentuation des hautes-fréquences

En général, la puissance d'un signal audio est beaucoup plus importante en basses fréquences qu'en hautes fréquences. Il y a donc un risque que la séparation soit peu efficace sur ces dernières. Une mesure de type SDR ne s'en trouvera pas nécessairement dégradé mais l'oreille humaine y sera sensible pour les raisons préalablement énoncées. Pour tenter de rendre la séparation plus uniforme, on peut imaginer renforcer les hautes fréquences avec un filtre passe-haut du type (FIG. 3.3) :

$$A(z) = 1 - 0.98z^{-1}. \tag{3.5}$$

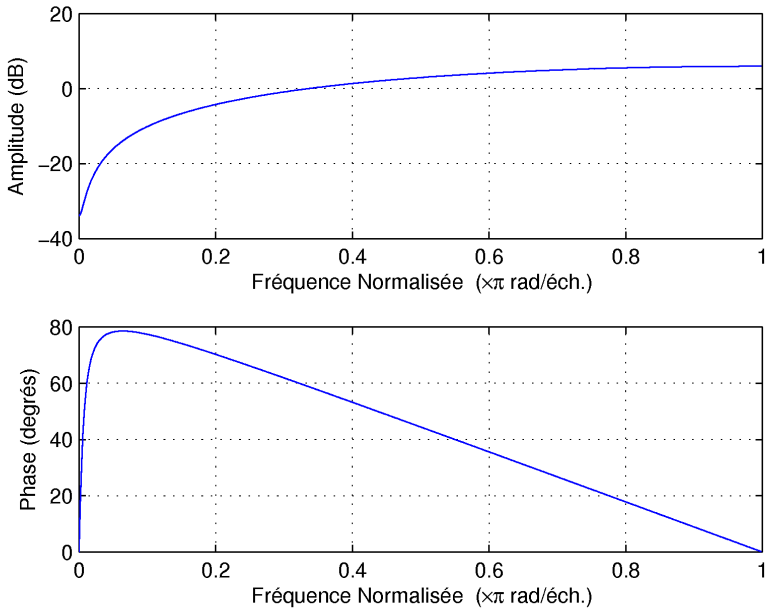


FIG. 3.3 – Module et phase de $A(z)$.

En appliquant ce filtre, on cherche finalement à compenser, très grossièrement, la non-linéarité de l'oreille. Il paraît donc judicieux de s'intéresser à la pondération fréquentielle de type "A" qui a été créée dans ce but.

3.3.2 Pondération type “A” (dBA)

La pondération (en fréquence), ou le filtre (en temps), de type “A” est prévue pour approcher la manière dont l'oreille entend les sons. En quelque sorte, elle permet de mesurer un niveau sonore perceptif et non physique. On peut ainsi utiliser cette pondération pour conditionner nos spectres afin qu'ils soient traités avec prise en compte de la non-linéarité de l'oreille. La courbe de pondération “A” est donnée ci-après (Fig. 3.4) en dB. L'application de la pondération se fait simplement en additionnant nos spectres en dB et cette courbe (exemple Fig. 3.5).

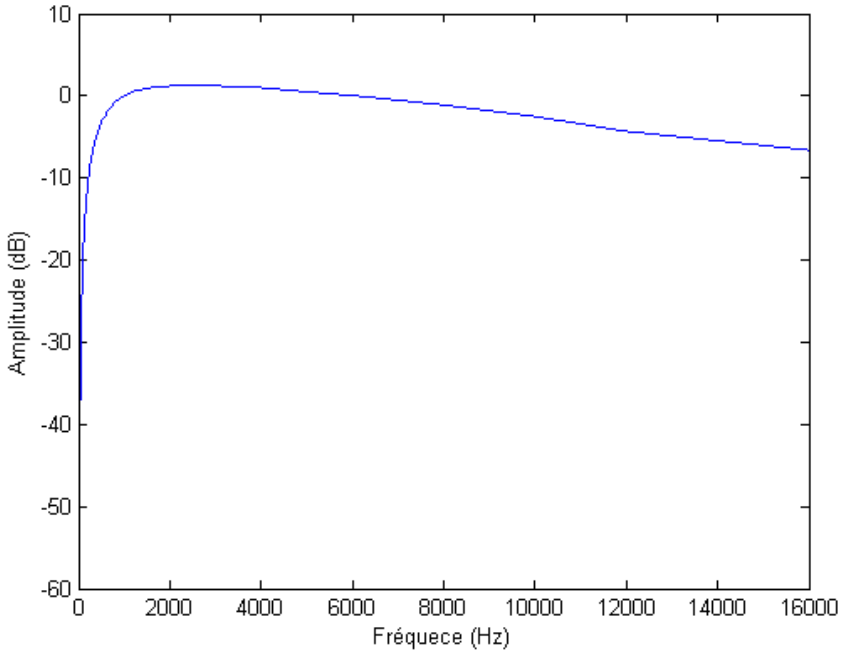
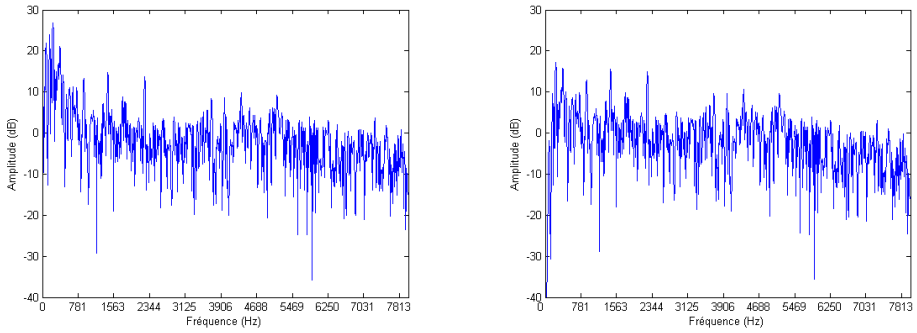


Fig. 3.4 – Courbe de pondération type “A”.

Remarquons que la pondération type “A” est très différente du filtrage passe-haut précédemment introduit. Leur point commun réside dans l'atténuation importante des basses fréquences mais la bande de transition du filtre passe-haut est beaucoup plus large que celle du filtre dBA. On utilisera, bien entendu, soit l'un, soit l'autre, et on peut s'attendre à des résultats assez différents.

3.3.3 Echelle logarithmique

On a vu que la dynamique de l'oreille est très grande. Cependant, il y a peu de chances que la méthode de séparation soit sensible à la même dynamique. Une méthode courante pour compresser la dynamique est le passage au logarithme qui va, à la différence du filtrage passe-haut, transformer uniformément les spectres. On choisit la formule suivante :



(a) (b)

FIG. 3.5 – Exemple de spectre (a) avant et (b) après pondération type "A" (fréquence d'échantillonage $16kHz$, $L = 640$,$N = 2048$).

$$X_t^{log} = (\log(|X_t| + \epsilon) + cst)\, e^{\phi_t} \tag{3.6}$$

où ϕ_t est la phase de X_t, ϵ une constante positive et petite devant X_t qui permet d'éviter la valeur zéro pour laquelle le logarithme n'est pas défini et cst une constante permettant d'assurer que $|X_t^{log}| \geq 0$.

On ne peut pas prendre directement le logarithme du module du spectre, qui donnerait des valeurs dans $\mathbb{R}$, car l'algorithme est prévu pour des spectres avec des valeurs dans $\mathbb{R}^+$ (positives ou nulles) qui sont interprétées comme des variances.

Chapitre 4

Séparation des sources

Dans cette section, on considère que l'on dispose, pour chaque source, d'un modèle statistique obtenu lors d'une phase d'apprentissage détaillée dans le chapitre 5.

4.1 Filtrage de Wiener adaptatif

Positionons-nous dans le cas gaussien simple. Considérons que l'on cherche à séparer un signal de voix v d'un mélange voix/musique $y = v+m$ où m est le signal de musique. On supposera dans un premier temps les signaux stationnaires. Une méthode classique pour séparer deux signaux est le filtrage de Wiener. Il consiste à trouver le filtre w tel que l'energie de l'erreur $E[(\hat{v}(t) - v(t))^2]$, avec $\hat{v}(t) = \{w * y\}(t)$ le filtrage de y par w, soit minimale. Si on considère v et m indépendants, la fonction de transfert du filtre au sens des moindres carrés est donné par :

$$W(f) = \frac{\Sigma^v(f)}{\Sigma^v(f) + \Sigma^m(f)} \tag{4.1}$$

où $\Sigma^v(f)$ (resp. $\Sigma^m(f)$) est la valeur à la fréquence f de la Densité Spectrale de Puissance (DSP) du signal v (resp. m).

Pour le cas d'un mélange de gaussiennes, si on connait les indices q_v et q_m des gaussiennes actives dans les deux sources, le filtrage de Wiener peut s'appliquer directement en substituant dans la formule les covariances $\Sigma^v_{q_v}$ et $\Sigma^m_{q_m}$ aux covariances Σ^v et Σ^m . Lorsque (q_v, q_m) sont inconnus, on calcule le filtre comme une somme pondérée par les probabilités γ_{k_v,k_m} (éq. 4.3) des différents filtres de Wiener possibles [13]. On obtient la formule suivante :

$$W(f) = \sum_{k_v=1}^{K_v} \sum_{k_m=1}^{K_m} \gamma_{k_v,k_m} \frac{\Sigma^v_{k_v}(f)}{\Sigma^v_{k_v}(f) + \Sigma^m_{k_m}(f)} \tag{4.2}$$

avec

$$\begin{aligned} \gamma_{k_v,k_m} &\triangleq \mathbb{P}(q_v = k_v, q_m = k_m | X_t, \Sigma_v, \Sigma_m) \\ &\propto \omega^v_{k_v} \omega^m_{k_m} \mathbb{G}(X_t; \Sigma^v_{k_v} + \Sigma^m_{k_m}). \end{aligned} \tag{4.3}$$

Pour avoir l'égalité dans l'équation 4.3, il faut normaliser les γ_{k_v,k_m} par leur somme $S = \sum_{i=1}^{K_v} \sum j = 1^{K_m} \omega_i^v \omega_j^m \mathbb{G}(X_t; \Sigma_j^v + \Sigma_j^m)$ sur tous les couples (k_v, k_m).

Dans notre application, les signaux utilisés ne sont pas complètement stationnaires. On considère tout de même qu'ils le sont localement et on utilise une représentation en TFCT. Pour chaque spectre X_t de la TFCT du mélange voix/musique, les probabilités γ_{t,k_v,k_m}, qui dépendent maintenant de t, sont réévaluées et le filtre appliqué. On a donc un filtrage de type Wiener qui varie dans le temps et de manière adaptative puisque les γ_{t,k_v,k_m} dépendent des données y elle-mêmes. Les formules d'estimation de la voix et de la musique pour chaque trame t sont données respectivement équation (4.4) et (4.5) :

$$\hat{V}_t(f) = \left[\sum_{k_v=1}^{K_v} \sum_{k_m=1}^{K_m} \gamma_{t,k_v,k_m} \frac{\Sigma_{k_v}^v(f)}{\Sigma_{k_v}^v(f) + \Sigma_{k_m}^m(f)} \right] X_t(f), \tag{4.4}$$

$$\hat{M}_t(f) = \left[\sum_{k_v=1}^{K_v} \sum_{k_m=1}^{K_m} \gamma_{t,k_v,k_m} \frac{\Sigma_{k_m}^m(f)}{\Sigma_{k_v}^v(f) + \Sigma_{k_m}^m(f)} \right] X_t(f) \tag{4.5}$$

où $\hat{V}_t(f)$ (resp. $\hat{M}_t(f)$) représente la valeur à la fréquence f de la trame t de la TFCT estimée du signal de voix v (resp. de musique m).

Ces formules correspondent à l'estimateur de la Moyennne *a Posteriori* (*Posterior mean* - PM). On peut aussi utiliser l'estimateur du Maximum *A Posteriori* (MAP) qui consiste à choisir le filtre correspondant au $\gamma_{t,k_v,k_m}(t)$ maximum, ce qui revient à estimer les gaussiennes actives. Les nouvelles expressions sont données par les équations (4.6) et (4.7) :

$$\hat{V}_t(f) = \left[\frac{\Sigma_{\hat{q}_v(t)}^v(f)}{\Sigma_{\hat{q}_v(t)}^v(f) + \Sigma_{\hat{q}_m(t)}^m(f)} \right] X_t(f), \tag{4.6}$$

$$\begin{aligned} \hat{M}_t(f) &= \left[\frac{\Sigma_{\hat{q}_m(t)}^m(f)}{\Sigma_{\hat{q}_v(t)}^x(f) + \Sigma_{\hat{q}_m(t)}^m(f)} \right] X_t(f) \\ &= X_t(f) - \hat{V}_t(f) \end{aligned} \tag{4.7}$$

avec $(\hat{q}_v(t), \hat{q}_m(t)) = \underset{k_v,k_m}{\operatorname{argmax}} \gamma_{t,k_v,k_m}$.

Concrètement, dans le cas MAP, cela consiste, trame par trame, fréquence par fréquence, à répartir l'énergie de la TFCT du mélange entre la voix estimée et la musique estimée selon des proportions imposées par les valeurs à la fréquence f des gaussiennes actives de chaque modèle.

Les deux estimateurs donnent des résultats très similaires, autant à l'écoute qu'en valeur de NSDR (cf.4.3), avec cependant un léger avantage pour l'estimateur MAP. Ce dernier étant aussi le plus simple à calculer, tous les tests à venir l'utiliseront exclusivement.

4.2 Reconstruction par addition-recouvrement (*Overlap and Add* - OLA)

On dispose d'une TFCT X de T trames, on veut reconstituer le signal temporel $\hat{x}$ correspondant. Le principe de reconstruction OLA (Fig. 4.1), dont la formule est rappelée par l'équation (4.8), est relativement simple. On peut le décomposer en trois étapes :

1. calcul des TFDs inverses $\tilde{x}_t$ de chaque spectre X_t ;
2. multiplication de chaque $\tilde{x}_t$ par une fenêtre de pondération w_s dûment choisie ;
3. addition des signaux en tenant compte du recouvrement (le même qu'au calcul de la TFCT).

$$\hat{x}(n) = \sum_{t=1}^{T} w_s(n - \tau_a(t))\tilde{x}_t(n - \tau_a(t)) \tag{4.8}$$

La fenêtre de synthèse w_s doit, pour une reconstruction parfaite, i.e. $\hat{x}_t = x_t$ si X_t n'a pas subi de modification, respecter la condition suivante :

$$\sum_{t=1}^{T} w_a(n - \tau_a(t))w_s(n - \tau_a(t)) \equiv 1, \quad \forall n \tag{4.9}$$

où on rappelle que w_a est la fenêtre d'analyse et N le nombre de points des TFDs.

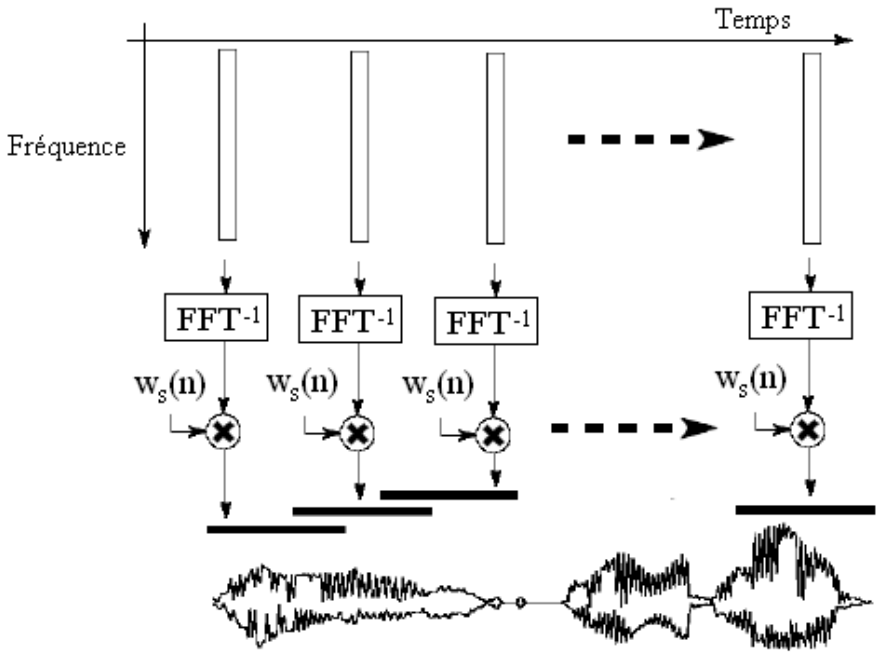


Fig. 4.1 – Schéma de principe de la méthode OLA.

4.3 Mesure de performance

La qualité d'une séparation de sources audio est avant tout perceptuelle. C'est à dire que le plus important est que l'auditeur percoive un minimum de défauts dans les signaux séparés. Néanmoins il reste difficile de mesurer la qualité d'une séparation sur ce seul critère perceptif, d'abord parce qu'il reste subjectif et ensuite parce qu'il faudrait avoir à disposition un ensemble d'auditeurs dès que l'on teste une nouvelle séparation. Il est donc utile d'avoir un moyen numérique de comparer entre elles plusieurs séparations.

On décide d'utiliser le Rapport Source à Distorsion (*Source to Distortion Ratio* - SDR) [17] défini comme suit :

$$SDR(\hat{s}, s) = 10 \log_{10} \left[\frac{\langle \hat{s}, s \rangle^2}{\|\hat{s}\|^2 \|s\|^2 - \langle \hat{s}, s \rangle^2} \right] \tag{4.10}$$

où s est une source quelconque, $\hat{s}$ son estimée, $\langle \cdot, \cdot \rangle$ représente le produit scalaire de deux vecteurs et $\| \cdot \|^2$ l'énergie d'un signal. Le SDR mesure la qualité d'une estimation. Lorsque l'estimée $\hat{s}$ tend vers le signal original s, le dénominateur tend vers zéro et le SDR vers l'infini. Un SDR élevé traduira donc une bonne estimation.

Pour évaluer la qualité d'une séparation, on utilise le SDR Normalisé (*Normalized SDR* - NSDR). Il mesure l'amélioration du SDR lorsqu'on passe du mélange à la source estimée. Celà se traduit, en logarithme, par la différence entre le SDR calculé avec l'estimée $\hat{s}$ et celui calculé en considérant le mélange x comme "estimée" :

$$NSDR(\hat{s}, x, s) = SDR(\hat{s}, s) - SDR(x, s). \tag{4.11}$$

Chapitre 5

Apprentissage des modèles

Comme vu précédemment, la séparation nécessite l'apprentissage d'un modèle par source, c'est à dire le calcul de ses paramètres θ. Celui-ci s'effectue, pour une source, à partir d'un corpus d'extraits représentatifs et en utilisant l'algorithme EM (*Expectation-Maximization* ou Espérance-Maximisation).

L'algorithme EM est un algorithme itératif dont le but est de trouver les paramètres qui maximisent la vraisemblance d'un modèle. Cependant, il ne garantit pas le maximum global. Il se décompose en deux étapes, calcul de l'espérance et maximisation de la vraisemblance, répétées alternativement jusqu'à la convergence.

Une des propriétés fondamentales de l'algorithme EM est que la vraisemblance augmente (ou reste stable) d'une itération à l'autre sans jamais décroître. De ce fait, on dira que l'algorithme converge lorsque l'augmentation de la vraisemblance devient faible, i.e. inférieure à un ε donné.

Dans cette section on considère l'apprentissage du modèle d'une seule source, le calcul étant identique pour toutes les sources.

5.1 MMG

On dispose de T réalisations de notre processus MMG, i.e. tous les spectres qui composent les TFCTs des extraits représentatifs. Notons que dans le cas de plusieurs extraits distincts, on construit un unique signal formé par leur juxtaposition. On cherche l'estimation $\hat{\theta}$ des paramètres qui maximise localement la log-vraisemblance des observations donnée par l'équation (5.1) :

$$\log(\mathbb{P}^{mmg}_{X_1,\dots,X_T}(X_1,\dots,X_T;\theta)) = \sum_{t=1}^{T} log\left(\sum_{k=1}^{K} \omega_k \mathbb{G}_k(X_t;\Sigma_k)\right). \tag{5.1}$$

5.1.1 Espérance

La première étape est le calcul des probabilités *a posteriori* $\gamma_{t,k}^{(l)}$ des gaussiennes k connaissant la réalisation t où l est l'indice d'itération :

$$\begin{aligned} \gamma_{t,k}^{(l)} &\triangleq \mathbb{P}(q(t) = k | X_t, \Sigma_k^{(l)}) \\ &= \frac{\mathbb{G}_k(X_t; \Sigma_k^{(l)}) \omega_k^{(l)}}{\sum_{j=1}^{K} \omega_j^{(l)} \mathbb{G}_j(X_t; \Sigma_j^{(l)})}. \end{aligned} \tag{5.2}$$

5.1.2 Maximisation

La seconde étape consiste en la réestimation des paramètres du modèle, les probabilité *a priori* des classes ω_k :

$$\omega_k^{(l)} = \frac{\sum_{t=1}^{T} \gamma_{t,k}^{(l-1)}}{T}, \tag{5.3}$$

et les matrices de covariances Σ_k :

$$\Sigma_k^{(l)} = \frac{\sum_{t=1}^{T} \gamma_{t,k}^{(l-1)} X_t X_t^T}{\sum_{t=1}^{T} \gamma_{t,k}^{(l-1)}}. \tag{5.4}$$

Etant donné que l'on considère uniquement des matrices de covariances diagonales, on impose la diagonalité de la réestimation en mettant à zéro tous les coefficients hors-diagonale dans $\Sigma_k^{(l)}$. Du point de vue de l'implémentation, et pour limiter les erreurs de précision, on calcul directement la diagonale comme suit :

$$diag[\Sigma_k^{(l)}] = \frac{\sum_{t=1}^{T} \gamma_{t,k}^{(l-1)} (X_t \odot X_t)}{\sum_{t=1}^{T} \gamma_{t,k}^{(l-1)}} \tag{5.5}$$

où $\odot$ désigne le produit terme à terme de deux vecteurs.

5.2 MMC

Dans le MMC, on considère R séries de réalisations, une série correspondant à la TFCT d'un extrait cohérent temporellement (à l'inverse du MMG, on ne considère pas un unique "méga-extrait" composé de tous les extraits). On note $r \in \{1, \ldots, R\}$ l'indice de la série et T_r le nombre de réalisations qu'elle comporte.

5.2.1 Espérance

Pour cette première étape, il y a deux probabilités *a posteriori* à calculer :

– la probabilité $\gamma_{t,j,r}^{(l)}$ à l'itération l que l'état de la réalisation t de la série r soit j connaissant toute la série r :

$$\gamma_{t,j,r}^{(l)} \triangleq \mathbb{P}(q^r(t) = j | X_1^r, \ldots, X_{T_r}^r), \tag{5.6}$$

– la probabilité $\xi_{t,i,j,r}^{(l)}$ à l'itération l que les état des réalisations t et $t-1$ de la série r soient respectivement j et i, connaissant toute la série r :

$$\xi_{t,i,j,r}^{(l)} \triangleq \mathbb{P}(q^r(t) = j, q^r(t-1) = i | X_1^r, \ldots, X_{T_r}^r). \tag{5.7}$$

Ce calcul nécessite l'utilisation de l'algorithme de Baum-Welch ou *forward-backward*.

5.2.2 Maximisation

La seconde étape est très similaire au cas du MMG. On réestime les ω_k qui correspondent maintenant aux probabilités initiales de la chaîne de Markov :

$$\omega_k^{(l+1)} = \frac{\sum_{r=1}^{R} \gamma_{1,k,r}^{(l)}}{\sum_{j=1}^{K} \sum_{r=1}^{R} \gamma_{1,j,r}^{(l)}}, \tag{5.8}$$

les matrices de covariance :

$$\Sigma_k^{(l+1)} = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{t,k,r}^{(l-1)} X_t^r (X_t^r)^T}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{t,k,r}^{(l-1)}}, \tag{5.9}$$

mais on calcule en plus la matrice de transitions :

$$a_{i,j}^{(l+1)} = \frac{\sum_{r=1}^{R} \sum_{t=2}^{T_r} \xi_{t,i,j,r}^{(l)}}{\sum_{j=1}^{K} \sum_{r=1}^{R} \sum_{t=2}^{T_r} \xi_{t,i,j,r}^{(l)}}. \tag{5.10}$$

Comme pour le MMG, on réestime directement la diagonale de $\Sigma_k^{(l+1)}$:

$$diag[\Sigma_k^{(l+1)}] = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{t,k,r}^{(l-1)} (X_t^r \odot X_t^r)}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{t,k,r}^{(l-1)}}. \tag{5.11}$$

En pratique, la formule de réestimation de ω_k nous a posé des problèmes. Le type de données traitées font que les valeurs dégénèrent rapidement (cf. 5.3). Pour y remédier, on effectue la réestimation de ces probabilités directement à partir de la matrice de transitions a en choisissant son vecteur propre associé à sa plus grande valeur propre comme nouveau vecteur ω (en prenant soin de le normaliser pour qu'il corresponde à des probabilités).

Montrons la validité de cette solution. On a dans un premier temps :

$$\begin{aligned}\omega_j &\triangleq \mathbb{P}(q^r(t) = j) \\ &= \sum_j \mathbb{P}(q^r(t) = j | q^r(t-1) = i)\mathbb{P}(q^r(t-1) = i) \\ &= \sum_j a_{i,j}\omega_i.\end{aligned}$$

Si on considère l'expression matricielle, on a :

$$\omega = a\omega$$

et on définit le vecteur $\omega^{(l)}$ comme le vecteur propre de la matrice $a^{(l)}$ associé à la valeur propre 1.

La valeur propre 1 existe nécessairement puisque la somme de chaque ligne vaut 1 (somme des probabilités d'un même événement). En effet, en notant $\mathbb{1}$ le vecteur unitaire de dimension K le nombre de sources, on a

$$a^{(l)}\mathbb{1} = \mathbb{1}$$

et donc 1 est valeur propre de $a^{(l)}$.

Prouvons maintenant que 1 est la valeur propre maximale de $a^{(l)}$. Supposons qu'il existe une valeur propre $\nu > 1$ associée à un vecteur propre p et posons i l'indice du plus grand coefficient de p :

$$i = \text{argmax}(|p_i|).$$

Le coefficient d'indice i de $|a^{(l)}p|$, noté $(a^{(l)}p)_i$, est $\nu|p_i| > |p_i|$. Comme la somme de chaque ligne de $a^{(l)}$ vaut 1, $(a^{(l)}p)_i$ est une moyenne pondérée des coefficients de p. En particulier, puisque p_i est le le plus grand coefficient de p,

$$|(a^{(l)}p)_i| \leq |p_i|.$$

Ainsi, on a

$$\nu p_i > p_i \quad \text{et} \quad |(a^{(l)}p)_i| \leq |p_i|,$$

ce qui est contradictoire avec $(a^{(l)}p)_i = \nu p_i$.

Nécessairement, le cas $\nu > 1$ est impossible et donc 1 est la valeur propre maximale de $a^{(l)}$.

Cette méthode de réestimation des ω_k permet d'éviter quasiment tout dégénérescence inopinée.

5.3 Problèmes numériques

Les densités de probabilités que l'on calcule ont des valeurs très faibles, en partie à cause de la grande dimension des vecteurs d'observations. Ces densités multipliées entre elles donnent des valeurs qui tendent rapidement vers zéro et font dégénérer l'algorithme. Pour palier à ce problème, on utilise deux méthodes complémentaires :

- Dès que possible, les calculs sont effectués dans le domaine logarithmique et non linéaire. Par exemple, pour le calcul des déterminants des matrices de covariance diagonales, on préfèrera calculer la somme des logarithmes des éléments de la diagonale plutôt que le produit de ces éléments.

- Les différentes valeurs sont calculées à des facteurs d'échelle près. On peut se le permettre dès que ces valeurs sont destinées à être normalisées, comme par exemple les probabilités dont la somme doit être égale à 1.

Appliquons ces solutions au calcul de $\gamma_{t,k}^{(l)}$ (éq. 5.2). Les matrices de covariance $\Sigma_k^{(l)}$ étant diagonales, la densité de probabilité gaussienne (éq. 3.2) peut se calculer comme suit :

$$\mathbb{G}_k(X_t; \Sigma_k^{(l)}) = \frac{1}{\pi^N \prod_f \Sigma_k^{(l)}(f)} \exp\left(- \sum_f \frac{|X_t(f)|^2}{\Sigma_k^{(l)}} \right). \tag{5.12}$$

Afin de privilégier le domaine logarithmique, on intègre le facteur multiplicatif dépendant de k et l à l'exponentielle :

$$\mathbb{G}_k(X_t; \Sigma_k^{(l)}) = \frac{1}{\pi^N} \exp\left(- \sum_f \left[\frac{|X_t(f)|^2}{\Sigma_k^{(l)}} + \log(\Sigma_k^{(l)}(f)) \right] \right). \tag{5.13}$$

On notera :

$$\mathbb{G}_k^*(X_t; \Sigma_k^{(l)}) = \exp\left(- \sum_f \left[\frac{|X_t(f)|^2}{\Sigma_k^{(l)}} + \log(\Sigma_k^{(l)}(f)) \right] \right). \tag{5.14}$$

On peut ensuite mettre $\frac{1}{\pi^N}$ en facteur du dénominateur de $\gamma_{t,k}^{(l)}$ puisqu'il est indépendant de k et l :

$$\sum_{j=1}^{K} \omega_j^{(l)} \mathbb{G}_j(X_t; \Sigma_j^{(l)}) = \frac{1}{\pi^N} \sum_{j=1}^{K} \omega_j^{(l)} \mathbb{G}_j^*(X_t; \Sigma_j^{(l)}). \tag{5.15}$$

Au final, on a :

$$\gamma_{t,k}^{(l)} = \frac{\mathbb{G}_k^*(X_t; \Sigma_k^{(l)}) \omega_k^{(l)}}{\sum_{j=1}^{K} \omega_j^{(l)} \mathbb{G}_j^*(X_t; \Sigma_j^{(l)})}. \tag{5.16}$$

Ce type de modifications est réitéré dès que possible. Notons que ces problèmes numériques ne sont pas anodins puisque l'implémentation de l'algorithme de Baum-Welch utilisée les prend initialement en compte en incluant des facteurs d'échelle.

Chapitre 6

Adaptations aux données

6.1 Adaptation des MMGs aux données

L'apprentissage est effectué sur un ensemble d'extraits de chaque source et nous donne des modèles "généraux". Le problème est qu'il existe une multitude de musiques et voix différentes. Notre modèle général ne sera pas nécessairement un bon modèle pour les sources à séparer. L'idée donnée dans [16] est d'adapter ces modèles aux données à séparer. Ce qui suit ne s'applique qu'au MMG mais pourrait être étendu au MMC.

6.1.1 Modification des modèles eux mêmes

Si l'extrait à séparer comporte des segments dans lesquels une source est seule, et des segments dans lesquels l'autre source est à son tour seule, le meilleur moyen d'avoir des modèles adéquats est d'en faire les apprentissages sur ces segments sans mélange. On aura donc des modèles adaptés et c'est théoriquement dans ce cas que l'on aura la meilleure séparation, si l'on exclut le cas empirique où les données d'apprentissage sont exactement celles du mélange.

Si seulement une des deux sources se trouve isolée sur certains segments, on peut tout de même en profiter. Considérons par exemple que l'on dispose de segments de musique isolée. Les auteurs de [16] proposent d'apprendre le modèle de musique sur ces segments puis de s'en servir pour adapter le modèle général de parole aux données. La méthode utilisée est assez classique, il s'agit d'estimer le modèle de parole par maximum de vraisemblance (6.1) :

$$\theta_v^{adapt} = \underset{\theta_v}{\operatorname{argmax}}\, \mathbb{P}(X_t|\theta^v, \theta^m). \tag{6.1}$$

De nouveau, on utilise l'algorithme EM mais uniquement sur les trames de mélange. La première étape (espérance) consiste à calculer les $\gamma^{(l)}_{t,k_v,k_m}$ en suivant la formule (4.3). La seconde (maximisation) consiste en :

– la réestimation des poids

$$\omega_{k_v}^{(l+1)} = \frac{1}{T}\sum_t \sum_{k_m} \gamma^{(l)}_{t,k_v,k_m} \tag{6.2}$$

où T représente ici le nombre de trames de mélange,
– la réestimation des covariances

$$\Sigma_{k_v}^{v(l+1)}(f) = \frac{\sum_t \sum_{k_m} \gamma_{t,k_v,k_m}^{(l)} \langle |V_t(f)|^2 \rangle_{k_v,k_m}^{(l)}}{\sum_t \sum_{k_m} \gamma_{t,k_v,k_m}^{(l)}} \tag{6.3}$$

avec

$$\begin{aligned} \langle |V_t(f)|^2 \rangle_{k_v,k_m}^{(l)} &= \mathbb{E}\left[|V_t(f)|^2 \mid X_t, q_v(t), q_m(t), \theta^{v(l)}, \theta^m \right] \\ &= \frac{\Sigma_{k_v}^{v(l)}(f)\Sigma_{k_m}^m(f)}{\Sigma_{k_v}^{v(l)}(f) + \Sigma_{k_m}^m(f)} + \left| \frac{\Sigma_{k_v}^{v(l)}(f)}{\Sigma_{k_v}^{v(l)}(f) + \Sigma_{k_m}^m(f)} X_t(f) \right|^2 . \end{aligned} \tag{6.4}$$

On pourra bien sûr effectuer le même genre d'adaptation si l'on dispose, non pas de musique, mais de voix isolée.

6.1.2 Adaptation par filtrage

Toujours dans [16], une autre méthode d'adaptation est proposée, l'adaptation par filtrage.

Cette approche est basée sur l'idée que deux enregistrements distincts présentent des conditions d'enregistrement différentes (microphone, acoustique de la salle, etc.) et que la modification de ces dernières peut être modélisée par un filtre causal, linéaire et temporellement invariant. Pour notre application, on peut donc chercher à adapter le modèle de voix (resp. musique) à l'extrait de mélange traité, en considérant que le modèle de musique (resp. voix) est adéquat, i.e. appris par exemple sur des passages sans voix (resp. musique) du mélange comme au 6.1.1.

La recherche du filtre adapté H_v pour la voix sur les extraits de mélange est résumée par l'équation (6.5) :

$$H_v = \underset{H}{\operatorname{argmax}}\, \mathbb{P}\left((X_t)_{\forall t} | \mathcal{H}\theta^v, \theta^m \right) \tag{6.5}$$

où $\mathcal{H} = diag[|H(f)|^2]_f$ est la matrice diagonale formée par les coefficients en module au carré du vecteur H.

On peut parler d'une estimation du type Régression Linéaire du Maximum de Vraisemblance (*Maximum Likelihood Linear Regression* - MLLR). Pour le calcul, l'algorithme EM est une nouvelle fois utilisé. La formule de réestimation est donnée, toujours pour la voix, par l'équation (6.6) :

$$|H_v^{(l+1)}(f)|^2 = \frac{1}{T} \sum_{t=1}^{T} \sum_{k_v=1}^{K_v} \frac{\sum_{k_m=1}^{K_m} \langle |V_t(f)|^2 \rangle_{k_v,k_m}^{(l)} \gamma_{t,k_v,k_m}^{(l)}}{\Sigma_{k_v}^v(f)} \tag{6.6}$$

avec $\gamma_t, k_v, k_m{}^{(l)} \triangleq \mathbb{P}(q_v(t) = k_v, q_m(t) = k_m)$.

6.2 Apprentissage des MMGs avec filtres adaptés

Dans la continuité du paragraphe précédent, les auteurs de [16] proposent de faire l'apprentissage en utilisant ces mêmes filtres adaptés. L'idée est simple : rendre les signaux d'apprentissage comparables afin de construire le modèle le plus représentatif possible. La méthode consiste à réévaluer, à chaque itération de l'algorithme EM, à la fois les paramètres du modèle mais aussi les filtres adaptés à chaque extrait du corpus. Pour ce, ils utilisent l'algorithme *Space-Alternating Generalized Expectation-maximization* (SAGE) [18] qui facilite la double réestimation en la séparant en deux phases. A chaque itération de SAGE correspondent deux itérations EM. Au final, cela revient à estimer alternativement (i.e. une itération EM sur deux) les filtres ou les paramètres du modèles, les données non réestimées étant considérées constantes.

On donne les équations de réestimation des deux phases, l est l'indice d'itération de l'algorithme SAGE, r l'indice de l'extrait considéré[1].

1. Réestimations des filtres :

$$|H^{r(l+1)}(f)|^2 = \frac{1}{T_r} \sum_{t=1}^{T_r} \sum_{k=1}^{K} \gamma_{t,k,r}^{(l)} \frac{|X_t^r(f)|^2}{\Sigma_k^{(l)}(f)} \tag{6.7}$$

   où $\gamma_{t,k,r}^{(l)}$ est donné par l'équation (5.2) en remplaçant $\Sigma_k^{(l)}$ par $\mathcal{H}^{r(l)}\Sigma_k^{(l)}$.

2. Réestimation des paramètres du modèle :

$$\omega_k^{(l+1)} = \frac{1}{\sum_{r=1}^{R} T_r} \sum_{r=1}^{R} \sum_{t=1}^{T_r} \tilde{\gamma}_{t,k,r}^{(l)} \tag{6.8}$$

   et

$$\Sigma_k^{(l+1)}(f) = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \tilde{\gamma}_{t,k,r}^{(l)} \frac{|X_t^r(f)|^2}{|H^{r(l+1)}(f)|^2}}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} \tilde{\gamma}_{t,k,r}^{(l)}} \tag{6.9}$$

   où $\tilde{\gamma}_{t,k,r}^{(l)}$ est donné par l'équation (5.2) en remplaçant $\Sigma_k^{(l)}$ par $\mathcal{H}^{r(l+1)}\Sigma_k^{(l)}$.

[1] On utilise la même notation que pour le MMC dans lequel on doit distinguer les différents extraits utilisés pour l'apprentissage (cf. 3.2.2)

Chapitre 7

Tests et résultats

7.1 MMG

Dans cette section, on utilise le MMG pour modéliser les sources. Les signaux audio sont échantillonnés à 16kHz. Les TFCTs sont calculées avec une fenêtre de pondération de type racine carrée de Hamming de $40ms$, un recouvrement de 50% et 2048 points de TFD (cf. 3.1). L'estimateur utilisé pour le filtrage de Wiener adaptatif est le MAP (cf. 4.1).

On utilise 8 secondes d'un signal de musique et d'un signal de parole (présentateur radio) pour l'apprentissage, les 2 secondes suivantes de chacun d'eux servant à former le mélange que l'on souhaite séparer. Les signaux utilisés pour l'apprentissage et pour la séparation étant issus des mêmes extraits, on dira que les modèles sont adaptés.

Pour la première série d'expériences, on choisit une musique de type *rock* (guitare - basse - batterie) et on fait varier les nombres de classes K_v de la voix et K_m de la musique. Pour la seconde, on choisit une musique de type *classique* jouée par un orchestre symphonique. Les NDSRs sont donnés dans Tab. 7.1. Les lignes grisées représentent les meilleurs résultats de chaque série.

La première chose que l'on remarque est que le nombre de classes, dès lors qu'il n'est pas trop petit, influence assez peu la qualité de la séparation. En effet, pour la série avec musique *rock* par exemple (Tab. 7.1(a)), de 8 à 128 classes, les NSDRs de la voix se situent entre $4,08$dB et $5,07$dB et ceux de la musique entre $4,74$dB et $5,89$dB. Globalement, le NSDR reste dans un intervalle de 1dB quand le nombre de classes augmente très significativement. Il en est de même pour la série sur la musique *classique* (Tab. 7.1(b)). On peut donc se limiter à un nombre de classes raisonnable sans pour autant dégrader la séparation.

Ensuite, on voit que les NSDRs avec musique *rock* et musique *classique* sont très différents. Là où, comme on vient de le voir, ceux de la voix du cas *rock* oscillent entre entre $4,08$dB et $5,07$dB, ceux du cas classique se situent entre $6,98$dB et $8,08$dB. Cela signifie que l'extraction de la parole est plus efficace avec la musique classique. Mais

PAROLE *Présentateur radio*		MUSIQUE *Rock*	
K_v	NSDR (dB)	K_m	NSDR (dB)
2	3,29	2	3,39
4	3,76	4	4,16
8	4,08	8	4,74
16	4,37	16	4,89
16	4,34	32	5,12
32	4,19	16	5,17
32	4,61	32	5,45
32	4,62	64	5,30
48	5,07	16	5,89
48	5,00	48	5,84
48	4,92	64	5,73
64	4,71	32	5,56
64	4,55	48	5,62
64	4,62	64	5,44
128	4,02	128	5,01

(a)

PAROLE *Présentateur radio*		MUSIQUE *Classique*	
K_v	NSDR (dB)	K_m	NSDR (dB)
2	5,02	2	1,59
4	6,19	4	2,70
8	8,08	8	4,36
16	7,91	16	4,03
16	8,01	32	4,19
32	7,58	16	3,97
32	7,64	32	4,04
32	7,42	64	3,71
48	7,64	16	4,05
48	7,96	48	4,07
48	7,96	64	4,07
64	7,88	32	4,30
64	7,62	48	3,88
64	7,68	64	4,00
128	6,98	128	3,46

(b)

TAB. 7.1 – Résultats de séparations MMG avec une fenêtre d'analyse en racine carrée de Hamming, $L = 640$ ($40ms$), $\rho = 50\%$, $N = 2048$ sur un mélange (a) *présentateur radio + rock*. (b) *présentateur radio + classique*.

restons prudents, cela ne signifie pas que le signal de parole estimé contient moins de bruit ou d'artefacts lorsqu'il est extrait de musique classique. Le NSDR mesure bien une amélioration du SDR qui lui mesure les défauts dans l'estimée. Justement, dans notre exemple, les SDRs pour la parole sont plus élevés avec la musique *rock* (cf. TAB. 7.2). A l'inverse pour la musique, les NSDR sont meilleurs pour le cas *classique* quand les SDRs sont plus élevés pour le cas *rock*.

Il ressort du paragraphe précédent que l'efficacité de la séparation dépend directement des signaux considérés. En plus de présenter des valeurs de NSDR différentes, on remarque aussi que la meilleure séparation n'est pas obtenue pour la même combinaison de nombres de classes dans les deux séries. La série *rock* donne son maximum pour $K_v = 48$ et $K_m = 16$, la série classique pour $K_v = K_m = 8$. Il est donc *a priori* difficile de trouver des nombres de classes "optimaux". Il reste que, étant donné la faible variabilité des résultats pour différents K_v et K_m, cette différence n'est peut-être pas significative.

Enfin, si on observe les enveloppes temporelles des sources et de leurs estimées (FIG. 7.1) obtenues dans le meilleur des cas (MMG, $K_v = 48$, $K_m = 16$), on voit que globalement la séparation est assez efficace. Les signaux sont très proches et les artéfacts, bien qu'indéniablement présents à l'écoute, n'ont pas une grande influence

PAROLE *Présentateur radio*		MUSIQUE *Rock*	
K_v	SDR (dB)	K_m	SDR (dB)
16	7,51	16	1,76
32	7,75	32	2,32
48	8,14	48	2,70
64	7,76	64	2,31

(a)

PAROLE *Présentateur radio*		MUSIQUE *Classique*	
K_v	SDR (dB)	K_m	SDR (dB)
16	5,39	16	6,54
32	5,12	32	6,55
48	5,45	48	*6,57
64	5,17	64	6,50

(b)

TAB. 7.2 – Quelques valeurs de SDRs correpondant aux NDSRs de TAB. 7.1.

sur la forme d'onde.

7.2 MMC

Pour ces tests avec le MMC, les données sont les mêmes que pour le MMG. Les NSDRs des deux séries de tests (*rock* et *classique*) sont donnés TAB. 7.3. Les valeurs de K_v et K_m choisies ne sont pas exactement les mêmes pour deux raisons : le temps de calcul beaucoup plus important et les enseignements du cas MMG.

Assez logiquement, les remarques faites pour le cas MMG sont toujours d'actualité. Si on exclut la première ligne ($Kv = K_m = 4$), l'intervalle de valeurs est encore plus restreint, de l'ordre de $0,6$dB. Concernant les différences *rock*/*classique*, on retrouve le même type de comportement.

Comparons maintenant ces résultats avec ceux du MMG. Les NSDRs sont assez semblables d'une méthode à l'autre, les différences sont de l'ordre du dixième de dB. A nombre de classes égal, le MMC est alternativement légèrement moins ou plus performant que le MMG. Plus gênant, la meilleure séparation avec MMC est moins bonne que le meilleure du cas MMG. On pouvait espérer une amélioration significative avec l'introduction des MMCs, ça n'est visiblement pas le cas. D'autant que si on prend en compte le surplus de complexité et donc temps de calcul induit par l'utilisation du MMC, ce modèle ne paraît pas être particulièrement intéressant dans notre application. Par la suite on se contentera d'effectuer des tests en utilisant le MMG, qui, même s'il peut ponctuellement donner des résultats moins bons que le MMC, est globalement plus satisfaisant.

7.3 Tests avec apprentissage sur corpus

Nous avons vu, avec les premiers tests, que la séparation donne des résultats encourageants lorsque les modèles sont "intrinsèquement" adaptés. Mais qu'en est-il lorsque les modèles sont généraux ? C'est ce qu'on se propose d'étudier dans cette partie.

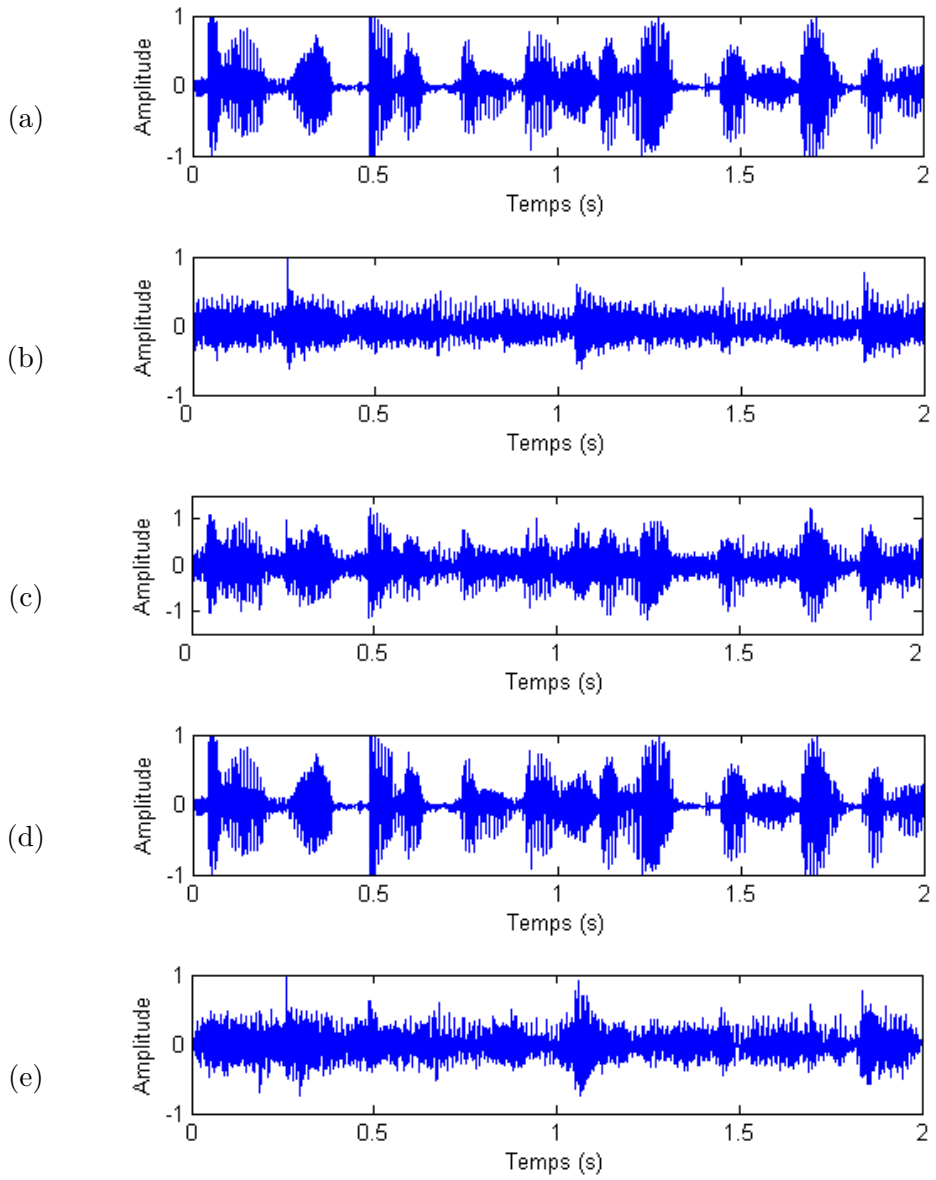


FIG. 7.1 – Enveloppes temporelles du test de séparation *présentateur radio / rock* (MMG, $K_v = 48$, $K_m = 16$) : (a) voix, (b) musique, (c) mélange, (d) voix estimée, (e) musique estimée.

Les modèles généraux sont appris sur 9 extraits de $20s$ chacun. On est loin de pouvoir forger un modèle réellement général étant donnée la grande diversité de voix et musique existante. Cependant, le temps de calcul dépendant directement de la quantité d'information utilisée pour l'apprentrissage, ce petit corpus représente un bon compromis. Notons que pour apprendre un tel modèle (parole ou musique) de type MMG avec 48 classes, avec la méthode standard, il faut compter 24 heures sur un ordinateur grand

PAROLE *Présentateur radio*		MUSIQUE *Rock*	
K_v	NSDR (dB)	K_m	NSDR (dB)
4	3,85	4	4,26
6	4,18	6	4,73
8	4,06	8	4,60
12	4,11	6	4,67
12	4,31	12	4,93
12	4,57	16	5,19
16	4,61	12	5,27
16	4,70	16	5,35
32	4,32	32	4,91
64	4,34	32	5,08

(a)

PAROLE *Présentateur radio*		MUSIQUE *Classique*	
K_v	NSDR (dB)	K_m	NSDR (dB)
4	6,14	4	2,64
6	7,84	6	4,13
8	7,81	8	4,11
12	7,90	6	4,15
12	8,41	12	4,63
12	8,08	16	4,33
16	8,18	12	4,42
16	7,81	16	4,10
32	7,73	32	4,01
64	7,74	32	4,01

(b)

TAB. 7.3 – Résultats de séparations MMC avec une fenêtre d'analyse en racine carrée de Hamming, $L = 640$ $(40ms)$, $\rho = 50\%$, $N = 2048$ sur un mélange (a) *présentateur radio + rock*. (b) *présentateur radio + classique*.

public[1] quasiment dédié.

7.3.1 Apprentissage standard

Dans ce paragragraphe on utilise des MMGs de 48 classes appris comme détaillé précédemment. On choisit 6 extraits de musique rock et 3 *jingles* de radio (ces mêmes modèles seront utilisés un peu plus loin pour des tests sur des données réelles de radio). On ne mélange pas de styles très différents afin que le modèle soit un minimum pertinent. Pour la parole, des voix d'hommes et femmes de la radio sont utilisées. On teste la séparation sur des mélanges *présentateur radio*/*rock* d'extraits n'ayant pas servi à l'apprentissage. Les deux locuteurs utilisés ($voix_1$ et $voix_2$) sont tous deux des hommes, les musiques de type rock ($rock_1$, $rock_2$ et $rock_3$) contiennent au minimum le trio typique guitare-basse-batterie. Les résultats, relatés dans TAB. 7.5, sont assez mauvais.

Comme on l'a déjà vu, les résultats dépendent énormément des signaux utilisés. Par exemple,$rock_3$ permet une bien meilleure extraction de la parole que les deux autres, $rock_2$ a un léger avantage concernant l'extraction de la musique. Si on regarde globalement les résultats, on remarque deux valeurs négatives qui signifient que la source est moins bien séparée que dans le mélange ! Deux autres valeurs restent inférieures à 1dB. Le mélange $rock_1$ + $voix_1$ correspond à celui utilisé dans les premiers test sur MMG (cf. 7.1) et MMC (cf. 7.2). Les résultats sont logiquement bien moins bon avec le corpus.

Comme pour la musique, on aurait pu considérer uniquement des voix d'hommes ou uniquement des voix de femmes à la fois pour le mélange et l'apprentissage. Le temps

[1] PIV 3GHz, 512Mo DDR, 80Go SATA

	NDSR (dB)	
Mélange	PAROLE	MUSIQUE
$voix_1$ + $rock_1$	0,72	2,01
$voix_1$ + $rock_2$	0,17	4,31
$voix_1$ + $rock_3$	3,95	2,07
$voix_2$ + $rock_1$	-0,31	1,51
$voix_2$ + $rock_2$	-2,09	2,72
$voix_2$ + $rock_3$	3,58	2,59
moyenne	1,00	2,54

TAB. 7.4 – Résultats de séparations sur corpus, apprentissage standard.

à manqué pour pouvoir tester ces cas plus particuliers mais il serait intéressant de voir si cela améliore les résultats, comme nous le pensons.

7.3.2 Apprentissage avec filtres adaptés

On utilise maintenant des MMGs appris avec filtres adaptés. Ce type d'apprentissage est plus gourmant en temps de calcul[2], on aurait pu diminuer le nombre de classes pour rester dans des temps raisonnables. Les corpus sont inchangés, seule la méthode d'apprentissage varie.

	NDSR (dB)	
Mélange	PAROLE	MUSIQUE
$voix_1$ + $rock_1$	2,18	3,46
$voix_1$ + $rock_2$	2,71	6,73
$voix_1$ + $rock_3$	5,74	2,88
$voix_2$ + $rock_1$	0,57	1,6
$voix_2$ + $rock_2$	-0,72	4,31
$voix_2$ + $rock_3$	4,41	2,83
moyenne	2,48	3,64

TAB. 7.5 – Résultats de séparations sur corpus, apprentissage avec filltres adaptés.

Clairement, cet apprentissage avec filtres adaptés améliore la séparation. D'abord globalement les moyennes des NSDRs de voix et musique passe respectivement de 1dB à $2,48$dB et de $2,54$dB à $3,64$dB. Si on regarde les résultats du premier mélange, on est plus proche des cas sans corpus (cf. 7.1 et 7.2) qu'avec l'apprentissage standard mais l'écoute nous rapelle clairement qu'une différence de 2 ou 3dB subsiste.

[2] A peu près 48h par modèle avec le même matériel

7.4 Tests avec adaptations

On vient de voir que la séparation à partir des modèles généraux est beaucoup moins évidente. Nous allons maintenant voir si l'adaptation des modèles au données permet d'améliorer ces résultats.

Pour pouvoir utiliser cette adaptation, il faut qu'un des deux modèles soit déjà "adapté", i.e. appris sur une partie du signal utilisé pour le mélange. Cependant, rien ne nous empêche de considérer qu'un des deux modèles généraux est adapté et de tenter l'adaptation du second. Les deux premières expériences traitent de ce cas, les deux suivantes utilisent des modèles "intrinsèquement" adaptés.

1. Modèle général de voix et adaptation pour la musique :

	NSDR (dB)			
	apprentissage standard		*appr. filtres adaptés*	
	PAROLE	MUSIQUE	PAROLE	MUSIQUE
Adaptation directe du modèle	0,98	2,42	2,11	3,97
Adaptation par filtrage	0,62	2,08	2,17	3,41

2. Modèle général de musique et adaptation pour la voix :

	NSDR (dB)			
	apprentissage standard		*appr. filtres adaptés*	
	PAROLE	MUSIQUE	PAROLE	MUSIQUE
Adaptation directe du modèle	0,25	1,67	1,06	1,78
Adaptation par filtrage	0,59	2,12	2,16	3,46

3. Modèle adapté de voix et adaptation pour la musique :

	NSDR (dB)			
	apprentissage standard		*appr. filtres adaptés*	
	PAROLE	MUSIQUE	PAROLE	MUSIQUE
Adaptation directe du modèle	1,41	3,49	2,52	3,96
Adaptation par filtrage	1,47	3,65	2,3	3,74

4. Modèle adapté de musique et adaptation pour la voix :

	NSDR (dB)			
	apprentissage standard		*appr. filtres adaptés*	
	PAROLE	MUSIQUE	PAROLE	MUSIQUE
Adaptation directe du modèle	5,36	7,07	5,2	6,91
Adaptation par filtrage	5,04	6,74	4,76	6,31

On note que l'adaptation joue bien son rôle. Lorsqu'un des modèles est déjà adapté, elle permet de se rapprocher des cas quasiment idéaux des premiers tests (deux modèles initialement adaptés). En considérant, à tort, un modèle général comme adapté, l'adaptation du second ne permet pas d'atteindre des sommets mais améliore sensiblement la séparation.

Si on compare les deux méthodes d'adaptation, réestimation directe du modèle ou filtrage, on note que la première semble légèrement plus efficace. Ceci est assez logique puisque le filtrage n'a pour but que de rendre comparables le modèle et le mélange en "uniformisant" les conditions d'acquisition des signaux. Le poids des gaussiennes n'est d'ailleurs pas modifié. Si dans les mêmes conditions d'acquisition, les sources et le modèle sont de toute façon très différents, l'adaptation par filtrage ne donnera pas un bon résultat.

7.5 Post-traitement par ACI

On désire ici tenter une ACI sur des signaux séparés par la méthode étudiée. En effet, une source estimée contient en majorité le signal de la source réelle mais aussi des restes plus ou moins importants de l'autre source. On peut alors considérer qu'on se trouve dans un cas déterminé de séparation multicapteurs : 2 sources et 2 observations (les estimations). Logiquement, on peut espérer qu'une ACI séparera davantage les sources en faisant tendre les estimées vers les sources réelles.

Pour ces tests on utilise l'algorithme JADE de Jean-François Cardoso et la fonction Matlab [19] qu'il propose sur son site. JADE est un algorithme d'ACI basé sur la diagonalisation conjointe des matrices de cumulants [20].

Etonnament, l'ACI n'améliore pas la séparation des sources, elle la dégrade ! La perte est de l'ordre de quelques de dixièmes de dB. Certes les NSDRs semblent assez peu diminuer mais la comparaison auditive est plus parlante. L'estimée de parole contient réellement plus de musique que sans post-traitement. Idem pour la musique. Nous avons testé d'autres algorithmes d'ACI comme FastICA [21, 22] mais les résultats sont du même acabit.

Le post-traitement par ICA paraissait être une bonne idée, il se révèle inefficace.

7.6 Tests sur signaux réels

Dans cette dernière partie, les mélanges ne sont plus construits manuellement. On prend des mélanges existants et on tente d'en extraire la parole ou la voix chantée et le fond musical. Ici, le seul moyen de mesurer la qualité de la séparation, c'est d'écouter ! En effet, on ne possède pas les sources initiales...

7.6.1 Séparation chant/musique sur des enregistrements du commerce

On a effectué des tests de séparation chant/musique sur des musiques du commerce, entre autres pour comparer avec ceux qu'Alexey Ozerov [23] présente sur son site internet. Dans un premier temps, on utilise exclusivement nos modèles "généraux" et dans un deuxième temps, on adapte le modèle de voix chantée à partir d'un modèle de musique appris sur les zones sans voix.

Comme on pouvait le prévoir, les séparations avec modèles généraux uniquement ne donnent rien de bon. Lorsqu'on adapte le modèle de voix à partir d'un modèle de musique initialement adapté, les résultats sont bien meilleurs. L'estimée de voix contient majoritairement la voix et celle de musique, majoritairement la musique. Néanmoins on reste en dessous des résultats d'Ozerov. Les deux principaux défauts, présents chez Ozerov mais dans des proportions moindres, sont :

- la présence non négligeable de voix dans la musique, même si elle l'est beaucoup moins que dans le mélange, on la distigue sans problème ;
- la présence de nombreux artefacts dans l'estimée de voix, dus à des résidus de musique qui créent un effet "aquatique " peu agréable.

Ces différences de qualité peuvent avoir plusieurs origines, comme un modèle original de voix peu pertinent, qui même s'il est réestimé est un mauvais point de départ (n'oublions pas que l'algorithme EM permet d'atteindre un maximum local et non global de vraisemblance) ou encore des paramètres de séparation différents.

7.6.2 Séparation parole/musique sur des extraits d'émissions radiophoniques

Pour terminer, on effectue des tests sur des enregistrements radiophoniques. Le modèle de parole a été appris sur un corpus composé exclusivement de présentateurs de la station dont sont extraits ces enregistrements. On choisit, pour les tests, des extraits dont le locuteur n'appartient pas au corpus. Le modèle de musique provient lui d'un corpus qui n'a rien à voir avec le fond musical de la station. Entre autres, les *jingles* utilisés sont assez différents des musiques du corpus. On peut donc s'attendre à des résultats peu convaincants. Cependant on a à notre disposition de nombreuses minutes de discussions sans musique et en particulier avec le locuteur choisi pour les tests. On peut donc dans un deuxième temps tester l'adaptation du modèle de musique à partir d'un modèle de voix adapté, appris sur ces trames.

Les résultats et observations sont similaires au cas *chant/musique* précédent. L'adaptation à partir d'un modèle initialement adapté de voix permet d'obtenir des résultats beaucoup plus intéressants que le cas sur corpus seuls. On retrouve les mêmes défauts de présence de voix dans la musique et d'artéfacts dans la voix.

Conclusion

La séparation de sources est un problème très en vogue mais loin d'être complètement résolu. Ce stage a permis de comprendre et tester une méthode de séparation de signaux audios dans le cas monophonique ainsi que diverses extensions et améliorations. Le but était la séparation de la voix et du fond musical dans des documents radiophoniques, concrètement il en résulte un ensemble de fonctions Matlab fonctionnelles qui le permettent plus ou moins efficacement et qui surtout permettront une poursuite du travail.

On a vu dans ce rapport, que le cas monophonique, et monocapteur en général, est particulièrement compliqué puisqu'il requiert des connaissances *a priori* sur les sources dont la pertinence est fondamentale pour obtenir une bonne séparation. La méthode sur laquelle on s'est concentré comporte initialement deux étapes. La première consiste justement à apprendre les modèles de musique et de parole. Deux possibilités on fait l'objet de tests, le MMG et le MMC. Seul le premier nous a semblé intéressant du point de vue de la complexité et des performances. L'apprentissage standard d'un modèle, réalisé grace à l'algorithme EM, permet d'obtenir les modèles qui maximisent la vraisemblance des signaux d'apprentissage. Dans le cas général, ces derniers doivent être aussi représentatif que possible du type de signal modélisé. Il est évident que le temps de calcul rend difficile l'apprentissage de gros corpus qui permettraient peut-être d'avoir des modèles plus généraux. L'implémentation de cette première partie a été rendue difficile par la présence de nombreux problèmes numériques. Les différentes formules de réestimations n'ont pu être codées telles quelles, l'algorithme dégénérant. La seconde étape est la séparation à proprement parler qui s'effectue par filtrage de Wiener adaptatif. Deux estimateurs étaient à notre disposition, PM et MAP, mais le second s'est révélé plus simple et plus performant.

Plusieurs améliorations ont ensuite été implémentées. Tout d'abord, une méthode améliorée d'apprentissage, l'apprentissage avec filtres adaptés, basée sur la modélisation des conditions d'acquisition de signaux par des filtres linéaires. Cette dernière permet de rendre comparables les signaux du corpus en faisant abstraction des différences dûes à leur environnement. On note avec cette méthode une amélioration non négligeable des NSDRs à contrebalancer avec un temps d'apprentissage en moyenne deux fois supérieur. Cette modélisation des conditions d'acquisition est aussi à l'origine d'une modification au niveau de la séparation. On rend comparables l'un des deux modèles (voix ou musique) et le mélange en calculant le filtre adapté au mélange et au second modèle

considéré adapté[3]. Cette adaptation par filtre adapté (aussi appelée adaptation MLLR) se traduit par un léger gain de NSDR. La dernière amélioration est l'adaptation, ou réestimation, des paramètres du modèle de voix ou musique en fonction des données de mélange et de l'autre modèle (musique ou voix). On considère un des deux modèles parfaitement adapté au mélange et on réestime, le connaissant, le second modèle de sorte qu'il maximise la vraisemblance du mélange. Cette méthode est particulièrement efficace (plus que l'adaptation MLLR sur les test effectués) lorsque l'un des deux modèles est réellement adapté, i.e. appris sur une partie du mélange ne contenant qu'une des deux sources. Ces deux méthodes d'adaptation, puisqu'elle utilisent aussi l'algorithme EM, introduisent une augmentation de complexité et donc de calcul assez importante. Néanmoins le gain en performance nous incite, dès que possible (i.e. dès qu'un des deux modèles est initialement bien adapté) à user d'une des deux.

Comme on a pu le constater, la séparation n'est pas encore parfaite. Il reste donc du chemin à parcourir et de nombreuses voies à explorer ! Dans un premier temps il pourrait être intéressant de faire l'apprentissage de corpus plus conséquents. Attention, si un modèle général pertinent de la voix semble envisageable, cela paraît beaucoup plus difficile pour la musique tant sa diversité est grande. Ensuite, la phase de post-traitement, qui fut ici limitée au test de l'ACI, pourrait être approfondie avec notamment d'autres méthodes de séparation aveugle comme celle qui utilise l'EMD [7].

D'un point de vue personnel, ce stage m'a beaucoup apporté. J'ai pu expérimenter des problèmes liés au développement d'un algorithme à partir d'un article scientifique. Les détails d'implémentation y sont rarement traités mais les aléas toujours rencontrés. En grande partie cela s'est traduit par des problèmes numériques et une dégénérescence de l'algorithme. J'ai aussi appronfondi certains aspects et techniques de traitement statistique du signal avec lesquels je n'étais pas particulièrement à l'aise. Enfin, l'objectif du stage est atteint même si la qualité de séparation obtenue reste en dessous des attentes. Au final, ce stage a confirmé mon intention de travailler dans la recherche audio.

[3] dans le sens où il modélise très fidèlement le signal de musique ou voix contenu dans le mélange.

Annexe A

Algorithme de Baum-Welch ou *Forward-Backward*

① **Définitions** :

– variable *forward*

$$\alpha_{r,t,k} \triangleq \mathbb{P}(X_1^r, \ldots, X_t^r, q^r(t) = k)$$

– variable *backward*

$$\beta_{r,t,k} \triangleq \mathbb{P}(X_{T+1}^r, \ldots, X_{T_r}^r | q^r(t) = k)$$

– vraisemblance de l'échantillons t de la réalisation r

$$b_{r,t,k} \triangleq \mathbb{G}(X_t^r; \Sigma_k)$$

② **Pour** $r = 1$ à R **Faire**

➲ **Pour** $k = 1$ à K **Faire** $\alpha_{r,1,k} = \omega_k b_{r,1,k}$ **FinPour**

➲ $c(1, r) = \sum_k \alpha_{r,1,k}$

➲ **Pour** $k = 1$ à K **Faire** $\tilde{\alpha}_{r,1,k} = \frac{\alpha_{r,1,k}}{c(1,r)}$ **FinPour**

➲ **Pour** $t = 1$ à $T_r - 1$ **Faire**

Pour $k = 1$ à K **Faire** $\bar{\alpha}_{r,t+1,k} = \left(\sum_{i=1}^K \tilde{\alpha}_{r,t,i} a_{i,k}\right) b_{r,t+1,k}$ **FinPour**

$c(t + 1, r) = \sum_{k=1}^K \bar{\alpha}_{r,t+1,k}$

Pour $k = 1$ à K **Faire** $\tilde{\alpha}_{r,t+1,k} = \frac{\bar{\alpha}_{r,t+1,k}}{c(t+1,r)}$ **FinPour**

FinPour

➲ **Pour** $k = 1$ à K **Faire** $\quad \tilde{\beta}_{r,T_r,k} \quad$ **FinPour**

➲ **Pour** $k = 1$ à K **Faire**
Pour $t = T_r - 1$ à 1 **Faire**
$\bar{\beta}_{r,t,k} = \sum_{i=1}^{K} a_{k,i} \tilde{\beta}_{r,t+1,i} b_{r,t+1,i}$
$\tilde{\beta}_{r,t,k} = \frac{\bar{\beta}_{r,t,k}}{c(t,r)}$
FinPour
FinPour

➲ **Pour** $t = 1$ à T_r **Faire**
Pour $k = 1$ à K **Faire** $\quad \bar{\gamma}_{r,t,k} = \tilde{\alpha}_{r,t,k} \tilde{\beta}_{r,t,k} \quad$ **FinPour**
Pour $k = 1$ à K **Faire** $\quad \gamma_{r,t,k} = \frac{\bar{\gamma}_{r,t,k}}{\sum_{i=1}^{K} \bar{\gamma}_{r,t,i}} \quad$ **FinPour**
FinPour

➲ **Pour** $t = 2$ à T_r **Faire**
Pour $i = 1$ à K **Faire**
Pour $j = 1$ à K **Faire** $\quad \bar{\xi}_{r,t,i,j} = \tilde{\alpha}_{r,t-1,i} \tilde{A}_{i,j} \beta_{r,t,j} b_{r,t,j} \quad$ **FinPour**
Pour $j = 1$ à K **Faire** $\quad \xi_{r,t,i,j} = \frac{\bar{\xi}_{r,t,i,j}}{\sum_{m=1}^{K} \sum_{n=1}^{K} \bar{\xi}_{r,t,m,n}} \quad$ **FinPour**
FinPour
FinPour

FinPour

③ **Calcul de la log-vraisemblance** :

$$LV\left((X_1, \ldots X_{T_1}), \ldots, (X_1, \ldots X_{T_R})\right) = \sum_{r=1}^{R} \sum_{t=1}^{N_r} \log(c(t,r))$$

Bibliographie

[1] J.-F. Cardoso, « Analyse en composantes indépendantes », *in Actes des XXXIVèmes Journées de Statistique, JSBL*, 2002.

[2] J.-F. Cardoso, « Blind signal separation : statistical principles », *Proc. IEEE*, vol. 9, p. 2009–2025, oct. 1998.

[3] R. Penrose, « A Generalized Inverse for Matrices », *Proc. of the Cambridge Philosophical Society*, vol. 51, p. 406–413, 1955.

[4] T.-W. Lee, S. M. Lewicki, M. Girolami et J. T. Sejnowski, « Blind Source Separation of More Sources Than Mixtures Using Overcomplete Representations », *IEEE Signal Proc. Letters*, vol. 6, no. 4, p. 87–90, 1999.

[5] M. Zibulevsky et B. Pearlmutter, « Blind Source Separation by sparse decomposition in a signal dictionary », *Neural Computations*, vol. 13, no. 4, p. 863–882, 2001.

[6] O. Bermond et J.-F. Cardoso, « Méthodes de séparation de sources dans le cas sous-déterminé », *in GRETSI*, 1999.

[7] A. Aissa-El-Bey, K. Abed-Meraim et Y. Grenier, « Séparation aveugle sous-déterminée de sources audio par la méthode EMD (Empirical Mode Decomposition) », *in GRETSI*, sept. 2005.

[8] A. M. Casey, « Separation of Mixed Audio Sources by Independent Subspace Analysis », *in GRETSI*, sept. 2001.

[9] S. Karnebäck, « Expanded examinations of a low frequency modulation feature for speech/music discrimination », *in ICSLP*, 2002.

[10] J. Rosier, *Méthodes d'estimation de fréquences fondamentales multiples pour la séparation de signaux de parole et musique*. Thèse doctorat, ENST, 2003.

[11] E. Vincent et M. D. Plumbley, « Single-channel mixture decomposition using bayesian harmonic models », *in ICA*, 2006.

[12] S. T. Roweis, « One microphone source separation », *in NIPS*, p. 793–799, déc. 2000.

[13] L. Benaroya et F. Bimbot, « Wiener based source separation with HMM/GMM using a single sensor », *in ICA*, avril 2003.

[14] L. Benaroya, F. Bimbot et R. Gribonval, « Audio source separation with a single sensor », *IEEE Trans. Audio, Speech, Language Processing*, vol. 14, jan. 2006.

[15] A. Ozerov, R. Gribonval, P. Philippe et F. Bimbot, « Séparation voix / musique à partir d'enregistrements mono : quelques remarques sur le choix et l'adaptation des modèles », *in GRETSI*, sept. 2005.

[16] A. Ozerov, R. Gribonval, P. Philippe et F. Bimbot, « One microphone singing voice separation using source-adapted models », *in WASPAA*, oct. 2005.

[17] R. Gribonval, L. Benaroya, E. Vincent et C. Févotte, « Proposals for performance measurement in source separation », *in ICA*, avril 2003.

[18] J. A. Fessler et A. O. Hero, « Space-Alternating Generalized Expectation-Maximization Algorithm », *IEEE Trans. Signal Processing*, vol. 42, oct. 1994.

[19] `http://www.tsi.enst.fr/~cardoso/Algo/Jade/jadeR.m`.

[20] J.-F. Cardoso et A. Souloumiac, « Blind beaforming for non gaussian signals », *IEE Proceedings-F*, vol. 140, p. 362–370, déc. 1993.

[21] A. Hyvärinen, « Fast and robust fixed-point algorithms for indepedent component analysis », *IEEE Trans. Neural Networks*, vol. 10, no. 3, p. 626–634, 1999.

[22] `http://www.cis.hut.fi/projects/ica/fastica/`.

[23] `http://www.irisa.fr/metiss/ozerov/`.