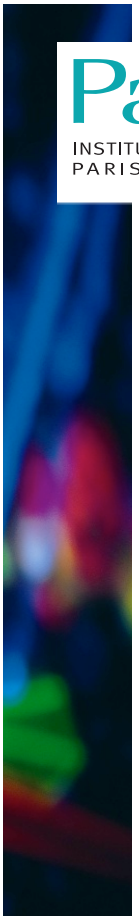
# L'analyse probabiliste en composantes latentes et ses adaptations aux signaux musicaux : application à la transcription automatique de musique et à la séparation de sources

Benoit Fuentes



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0011

EDITE - ED 130

## Doctorat ParisTech

# T H È S E

**pour obtenir le grade de docteur délivré par**

## TELECOM ParisTech

### Spécialité « Signal et Image »

*présentée et soutenue publiquement par*

**Benoit FUENTES**

le 14 mars 2013

# L'analyse probabiliste en composantes latentes et ses adaptations aux signaux musicaux. Application à la transcription automatique de musique et à la séparation de sources.

Directeur de thèse : **Roland BADEAU**
Co-encadrement de la thèse : **Gaël RICHARD**

**Jury**

| | |
|---|---|
| **M. Bruno TORRÉSANI**, Professeur, LATP, Aix-Marseille Université | Président |
| **M. Emmanuel VINCENT**, Chargé de Recherche, INRIA Nancy | Rapporteur |
| **M. Sylvain MARCHAND**, Professeur, Université de Bretagne Occidentale | Rapporteur |
| **M. Tuomas VIRTANEN**, Maître de conférence, MRG, Tampere University of Technology | Examinateur |
| **M. Roland BADEAU**, Maître de conférence, TSI, Télécom ParisTech | Directeur de thèse |
| **M. Gaël RICHARD**, Professeur, TSI, Télécom ParisTech | Directeur de thèse |

T H È S E

**TELECOM ParisTech**
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

*Si tu as une pomme, que j'ai une pomme,*
*et que l'on échange nos pommes,*
*nous aurons chacun une pomme.*
*Mais si tu as une idée, que j'ai une idée*
*et que l'on échange nos idées,*
*nous aurons chacun deux idées.*
— George Bernard Shaw

# Remerciements

Si vous lisez probablement ces lignes en guise de prélude, elles sont pour moi un point final à l'écriture de ce manuscrit et sont l'occasion de revenir sur une idée souvent reçue : non, le doctorat n'est pas le fruit d'un travail solitaire. Je suis loin d'être l'unique responsable de l'existence de ce document ! Sont remerciés ici avec une infinie sincérité toutes celles et ceux qui d'une manière ou d'une autre ont permis la réalité de cette thèse.

En priorité, je souhaiterais remercier mes deux directeurs de thèse **Roland Badeau** et **Gaël Richard** qui m'ont offert cette opportunité exceptionnelle de pouvoir effectuer ma thèse dans leur équipe. **Roland**, selon moi comme beaucoup de monde un brillant et grand chercheur, a été mon interlocuteur privilégié. De part sa grande modestie, sa pédagogie et sa disponibilité, j'ai toujours pu lui demander assistance avec une extrême facilité. La qualité de cette thèse et des publications qui l'ont accompagnée aurait été nettement moindre sans son aide et ses relectures affûtées. Il a su m'enseigner toutes les qualités requises pour un chercheur, et restera assurément un modèle dans mes années futures. **Gaël**, bienveillant à mon égard comme à celui de tous, m'a toujours donné sa confiance et je l'en remercie. C'est essentiellement grâce à lui que j'ai pu vivre toutes mes expériences professionnelles de recherche : d'abord mon stage de recherche aux Pays-Bas quand j'étais encore élève ingénieur à Télécom ParisTech, pour suivre avec mon stage de Master 2 au sein de son équipe et bien entendu, mon doctorat. Je ne me serais très probablement pas lancé dans cette dernière aventure sans son aide, ses encouragements et son implication dans l'obtention d'une bourse. Pendant cette thèse, il a toujours fait preuve d'une incroyable clairvoyance sur mes travaux de recherche et a su donner son soutien quand j'en avais besoin. Travailler avec lui est une vraie partie de plaisir !

Avant eux, il y a eu 21 années d'école et d'enseignement et 13 ans de conservatoire. Jamais (ô grand jamais) je n'aurais acquis cette soif d'apprendre, de sans cesse parfaire mes connaissances, d'explorer quelques terres encore inconnues (aussi étroites soient-elles) et partager mon savoir, si je n'avais pas rencontré des enseignants passionnants et formidables. Par ordre chronologique, je pense à **Mr. Charbonnier**, **Michel Bruzat**, **William Bensimon**, **Élisabeth Douay**, **Eric Durand**, **Pierre-Michel Bédard**, **Philippe Testud**, **Philippe Masse**, **Gérard Blanchet**, **Maurice Charbit**, **Jean-Louis Dessalles**, **Bertrand David**, **Roland Badeau** (encore lui) ou encore **Benoît Fabre**.

L'aventure s'est terminée (ou a commencé véritablement) avec la soutenance, et j'aimerais

également remercier tous les membres du jury qui m'ont fait l'honneur d'apprécier mon travail. En particulier, un grand merci à **Emmanuel Vincent** et à **Sylvain Marchand** qui ont lu et évalué mon manuscrit avec pertinence et soin.

Les trois années qui ont précédé l'écriture du manuscrit se sont déroulées au sein de l'équipe AAO de Télécom ParisTech, constituée des plus exceptionnels amis et collègues qui soient. Aussi bien pour les échanges scientifiques que sur le plan personnel, me lever chaque matin fut d'une simplicité enfantine tant l'ambiance de travail était remarquable. On a quand même été vraiment heureux tous ensemble. Merci donc **Yves**, **Nicolas**, **Gaël, Bertrand**, **Roland**, **Slim**, **Fabrice**, **Nancy** (je ne m'en cacherai pas, ton manuscrit de thèse a été grande source d'inspiration et si je suis, en toute modestie, plutôt content du mien, c'est en essayant d'en produire un d'aussi bonne qualité), **Jean-Louis** (de la méthode Durrieu), **Thomas** (mister Faïon, in Zaïon, like a Laïon), **Benoît**, **Cyril**, **Félicien**, **Mounira** (Mouniraaaaa, notre catalyseur de bon esprit), **Romain** (bien pêchu), **Rémi** (youpiiiiiiiiii), **Manuel** (est-ce bien raisonnable tout ça ?), **Antoine** (mon plus grand collaborateur, tendrement), **Sébastien** (autant pink que dark)**, Angélique** (en termes de portes ouvertes, on fait comment ?), **François** (quelle putain de rock star celui-là !), **Nicolas**, **Aymeric**, **Anne-claire**, **Cécilia**. Et puis même s'ils ne sont pas de notre lab, je les considère également comme des amis et des collègues : merci à **Laurent** (excellent maître de stage), **Valentina** (aaaaaaaaaaaaaaa), **Arnaud**, **Zafar**, **Benjamin**, **Gilles**, **Philippe**, **Pierre**.

En dehors du laboratoire, il y a aussi un monde tout aussi riche, rempli d'amis et de compagnons de route sans qui la vie serait bien morose et sans qui nulle thèse ne vaudrait la peine d'être lancée. Une passion qui remplie ma vie est la musique, et j'aimerais remercier tous mes compagnons musicaux ! Un immense merci à mes tous premiers complices des Domicros, votre amitié est l'un de mes biens les plus précieux : **Alexis**, **Rémi** et **Aurélie**. Merci à mon compère des Babies Genious, **Paul**, notre éphémérité n'a d'égale que la qualité de nos deux uniques chansons ! Merci à mes acolytes de Beubeu&Co, grâce à vous on n'est pas toujours trop seul : **Adrien** et **Arthur**. Merci aussi aux fat PGs de Alarm ! Alarm, on aura fait du fat rock comme on l'aime : **Olivier**, **Arnaud**, **Arthur** et **Manu**. Et enfin, merci Tarsius, c'est avec vous que j'ai décidé aujourd'hui de poursuivre ma route : **Hector**, **Yannick** et **Vincent** (to f.... !). Il y en a aussi tant d'autres : celles et ceux qui sont ou ont été présents dans ma vie et qui importent plus que tout. Vous êtes bien trop nombreux, que ceux que j'oublie me pardonnent : **Guiom** (ma plus forte béquille pendant cette thèse et je l'espère pour plus tard), mon **Paul** (ah oui oui oui), **Marie** (véritable amie et compagne d'expo et autres trucs cools à faire), **Gwendal** (j'espère que cette thèse est suffisamment équitable pour toi), **Céline** (j'ai bien reçu votre paquet), **Baptiste**, **Maroin** (paris paris), **Gros Louis**, **Gros Nico** (des gros pé..aaaaaaaaaa...), **Camille** (rien de cassé ?), **Chloé** (des sacrées soirées per...), **Jeanne** (puddy puddy), **Yann**, **Jean Gay** (on est pas bien là ?), **J-L**, **Patrick**, **Clément, Tômo**, **Djo**, **Perig** (en termes de demi-tours, on est comment ?), **Cyril**, **Manu**, **Juliet**, **Sarah** (allons répandre l'amour)**, Claire** (merciCl tudèch), **Pierre Pa**, **Quentin** (c'est de la marmelade ?), **Papaaaaa** (le mieux, c'est un petit verre de champagne), **Youssef** (mais t'y peux rien), **Caro** (ma petite femme) et tous mes amis

de Léonard Limosin, de Saint-Jean, du conservatoire de musique et de théâtre de Limoges, des stages 10/15, de Ginette, de Télécom, de l'IRCAM, des Coqs En Pâtes, à tous ceux qui sont venus à la soutenance, **merci merci merci merci merci merci merci** !

Pour finir, je voudrais bien entendu remercier du fond du cœur ma famille. Sans eux, je ne serais jamais arrivé là où j'en suis. Merci à mes parents **Pascale** et **Gilles**, ils m'ont toujours encouragé et conduit dans mes choix, et si aujourd'hui je suis heureux dans ma vie professionnelle, artistique et personnelle, je leur dois une très grande part du mérite. Un immense merci à **Emmanuel**, je suis fier d'être son petit frère, et faire sa fierté est aussi un honneur pour moi. Il a toujours été un modèle d'ambition et de courage. Je remercie aussi ma belle-mère **Claire** et mon petit frère **César**, mes grand-mères **Nicole** (la mamaille) et **Jacqueline**, mon grand-père **Reynald** ainsi que **Mara**, mes oncles et tantes **Arnaud**, **Christophe**, **Dana**, **Franck** (une pensée particulière pour toi), **Olivier**. Enfin, je remercie mes cousines et cousins **Livia**, **Adrien**, **Jeanne** et la petite **Naïs**. À tous, faire partie de votre famille est un honneur et il est certain que cela a fortement influencé mes choix, ambitions et réussites.

# Table des matières

# Liste des Algorithmes

# Acronymes

| | |
|---|---|
| **BHAD** | *Blind Harmonic Adaptive Decomposition* (décomposition harmonique, aveugle et adaptative) |
| **CQT** | *Constant-Q Transform* (transformée à Q constant) |
| **EM** | *Expectation-Maximization* (Espérance-Maximisation) |
| **Étape E** | Étape du calcul de l'Espérance |
| **Étape M** | Étape de Maximisation |
| **GUI** | *Graphical User Interface* (Interface graphique) |
| **HALCA** | *Harmonic Adaptive Latent Component Analysis* (Analyse harmonique et adaptative en composantes latentes) |
| **KKT** | Karush-Kuhn-Tucker (conditions de) |
| **LCATS** | *Latent Component Analysis with Temporal Structure* (analyse en composantes latentes avec structure temporelle) |
| **MLCATS** | *Markovian LCATS* (LCATS Markovienne) |
| **MAP** | Maximum *a posteriori* |
| **MV** | Maximum de vraisemblance |
| **NMF** | *Non-negative Matrix Factorisation* (factorisation en matrices positives) |
| **PLCA** | *Probabilistic Latent Component Analysis* (analyse probabiliste en composantes latentes) |
| **RTF** | Représentation temps-fréquence |
| **RTF$^+$** | Représentation temps-fréquence à coefficients positifs ou nuls |

| | |
|---|---|
| **SDR** | Rapport source à distorsion (*Source to Distortion Ratio*) |
| **SAR** | Rapport source à artéfacts (*Source to Artifact Ratio*) |
| **SIR** | Rapport source à interférences (*Source to Interference Ratio*) |
| **SIPLCA** | *Shift-Invariant PLCA* (PLCA avec invariance par translation) |
| **TFCT** | Transformée de Fourier à Court Terme |

# Notations

| | |
|---|---|
| $f$, $F$ | Indice de point fréquentiel, nombre de points fréquentiels |
| $t$, $T$ | Indice de trame temporelle, nombre de trames temporelles |
| $i$, $I$ | Indice de point fréquentiel d'activation temps-fréquence (représente une fréquence fondamentale), nombre de points fréquentiels d'activation temps-fréquence |
| $\mu$, $F$ | Indice de point fréquentiel des noyaux, nombre de points fréquentiels des noyaux |
| $n$, $N$ | Indice de note (ou d'atome), nombre de notes (ou d'atomes) |
| $s$, $S$ | Indice de sources, nombre de sources |
| $z$, $Z$ | Indice de noyau, nombre de noyaux |
| $c$ | Indice de composante ($c = h$ pour la composante harmonique, $c = b$ pour la composante de bruit, $c = m$ pour la composante de mélodie principale et $c = a$ pour la composante d'accompagnement) |
| $j$, $J$ | Indice de tirage dans un processus génératif, nombre de tirages |
| $l$ | Indice d'itération courante dans un algorithme EM |
| $\boldsymbol{X}$ | RTF observée (matrice complexe de coefficients $X_{ft}$) |
| $\boldsymbol{V}$ | RTF$^+$ observée (matrice positive de coefficients $V_{ft}$) |
| $\Lambda$ | Ensemble des paramètres d'un modèle de RTF$^+$ |
| $\boldsymbol{\theta}$ | Sous-ensemble de $\Lambda$ |
| $\beta_{\text{apr}}$ | Hyperparamètre définissant la force de l'apriori *apr* |
| $\beta_{\text{frein}}$ | Hyperparamètre définissant la valeur d'un coefficient de freinage |

| | |
|---|---|
| $\propto$ | Proportionnel à |
| $\nabla L$ | Gradient de la fonction $L$ |
| $H_L(x)$ | Matrice hessienne de la fonction $L$ au point $x$ |
| $\langle x, y \rangle$ | Produit scalaire des vecteurs $x$ et $y$ |

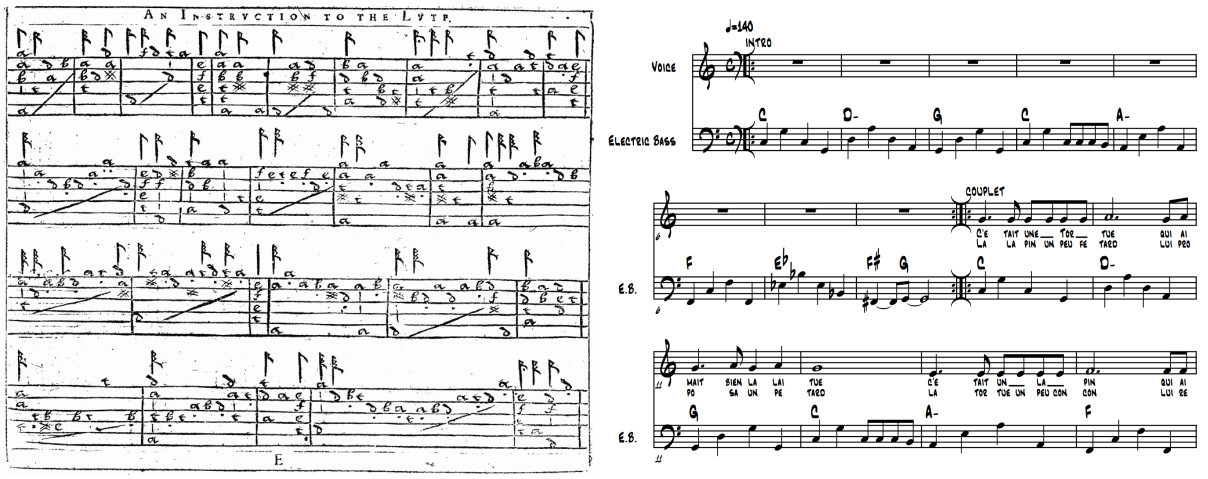
# Introduction

« Aujourd'hui on peut faire de la musique avec des ordinateurs, mais l'ordinateur a toujours existé dans la tête des compositeurs. » En affirmant cela dans *L'Art du Roman*, Milan Kundera pense à Bach, Mozart, Schubert, Stravinsky, Debussy, Schoenberg, ou tout autre compositeur occidental de musique dite savante. Il pense à leur incroyable capacité technique à traiter, ordonner, structurer, combiner, harmoniser toutes sortes de symboles, notes, accords, armures, rythmes, signatures, tout en respectant les règles de composition que leur époque impose, ou qu'ils se sont imposées, afin de traduire une idée et de composer une partition, une œuvre. Il se place en somme dans un paradigme où la composition musicale est purement symbolique, elle est agencement de notes, elle est abstraite. Si cela n'a pas été toujours le cas, c'est la norme du monde occidental du XVII$^{\text{ème}}$ au XX$^{\text{ème}}$ siècle.

Nous allons voir que l'un des objets de ce mémoire en est la cause : la transcription de musique.

## Contexte historique

Évidemment, la musique trouve son origine dans le monde et les objets qui nous entourent. Considérons par exemple les instruments de musique les plus anciens : du son que fait la corde de l'arc quand une flèche est tirée nait la harpe, du vent qui résonne dans les bambous nait la flute, du bruit de l'arbre qui craque naissent les instruments à percussions. De notre gorge sort notre voix. Et voilà que l'homme s'amuse à arranger et ordonner les sons provenant de ces instruments, à y incorporer des silences, à inventer la musique. Par nature, elle est évanescente et n'existe qu'à l'instant où on l'entend. Mais elle est aussi reproductible : il suffit de restituer les mêmes gestes dans le même ordre, avec le même rythme et sur les mêmes instruments. Seulement, la mémoire humaine n'est pas infaillible, et l'on s'est rapidement mis à inventer quelque moyen de notation pour éviter d'oublier. La transcription d'une œuvre musicale est née. Un système de transcription peut aussi bien être constitué de symboles décrivant des gestes musicaux (taper sur une cymbale, placer un doigt de sa main gauche entre deux frettes d'une guitare) que de symboles représentant des caractéristiques acoustiques d'un son musical, considérées comme porteuses d'information (hauteur perçue, durée, intensité, timbre, etc.). Quel que soit le cas, il inclut également des signes décrivant la manière dont ces sons ou gestes sont agencés dans le




(a) Une tablature de Luth

(b) La partition d'une chanson

**Figure 1** – Deux exemples de systèmes de notations. (a) Les symboles représentent des gestes musicaux à reproduire sur un luth. (b) Les symboles représentent des caractéristiques acoustiques du son : hauteur des notes (portée et clefs), phonèmes de la voix (paroles), ensemble de hauteurs (accords), etc.

temps. Sur la figure 1, deux exemples de systèmes de notation musicale sont illustrés.

Si l'on ne sait pas dater précisément l'apparition des premières écritures musicales, il est facile de constater que le besoin de représenter la musique de manière symbolique s'est manifesté dans la plupart des civilisations : en Chine par exemple, à environ dix mille ans avant notre ère ou encore en Grèce au VI$^{\text{ème}}$siècle av. J.-C. Dans l'histoire de la musique occidentale, c'est bien plus tard, au Moyen Âge avec le chant grégorien, que l'on commence véritablement à transcrire la musique grâce à un système de notation qui deviendra plus tard la partition de musique. Si au départ la partition ne sert qu'à décrire certaines caractéristiques sonores d'une œuvre musicale, elle prend une place de plus en plus prépondérante au fil du temps. À la Renaissance, on commence à composer essentiellement de la musique susceptible d'être annotée dans le système de l'époque. Petit à petit, les symboles acquièrent une existence propre et l'on invente des théories musicales qui régissent leur agencement. Là où l'on inventait des notations pour décrire une caractéristique physique du son, c'est le son que l'on adapte désormais pour qu'il concorde à la théorie musicale : la gamme tempérée, pour laquelle l'octave est divisée en douze intervalles égaux appelés demi-tons (*cf.* annexe A page 163), est progressivement adoptée en Occident à partir du XVII$^{\text{ème}}$ siècle pour l'accord de nombreux instruments (claviers, instruments à frettes), de la volonté de pouvoir, sur le papier comme pour l'oreille, moduler librement dans toutes les tonalités. En fin de compte, la partition n'est plus juste transcription : elle devient l'œuvre, la composition. Aussi, il est possible de composer tout en étant sourd, puisque les symboles à eux seuls incarnent la création musicale. Cette primauté du symbolique dans la musique occidentale durera de Bach jusqu'au milieu du XX$^{\text{ème}}$siècle et finira avec Pierre Schaeffer (et sa musique concrète) et un peu plus tard les Beatles : désormais, le compositeur (ou le groupe) travaille directement le son lui-même et l'inscrit sur un support mécanique (bande magnétique, disque

vinyle, CD, etc.) pour réaliser son œuvre. Alors, si l'on souhaite transcrire ces musiques de manière symbolique, il faut réinventer de nouveaux systèmes de notation ou tenter d'utiliser ceux dont on dispose.

Que ce soit pour transcrire une pièce de musique (enregistrée, jouée en concert ou diffusée en direct par exemple), ou pour retrouver la partition à l'origine d'une exécution musicale, la tâche de transcription était, jusqu'à récemment, bien entendu réservée aux acteurs humains. Depuis peu, l'essor de l'informatique ouvre un domaine de recherche vaste et passionnant pour l'automatisation de ce processus à partir d'un enregistrement. Le sujet est en soi un défi majeur pour la recherche scientifique de par sa grande difficulté : les capacités du cerveau humain sont toujours nettement supérieures à ce que l'on sait faire en informatique. Mais il est aussi porteur de nombreuses applications, qu'elles soient pédagogiques, commerciales, artistiques... En voici quelques exemples concrets :

- un logiciel de transcription automatique permettant de mettre sur partition l'improvisation d'un pianiste de jazz : bien utile à celui qui étudie et apprend cette discipline ;
- une application permettant de retrouver automatiquement, dans une grande base de données, le titre de la chanson fredonnée par un utilisateur ;
- un logiciel d'aide à l'apprentissage d'un instrument, qui détecterait et analyserait les erreurs d'un élève ;
- un système interactif pour une œuvre de musique contemporaine.

Maintenant que nous avons introduit la transcription de la musique ainsi que les enjeux de l'automatisation de cette tâche, nous allons désormais véritablement présenter le cahier des charges de ce que nous appelons, dans ce mémoire, transcription automatique.

## La transcription automatique

Vouloir transcrire toute la musique du monde avec un unique système de notation est tout simplement impossible. Elle est trop vaste, et l'information importante peut se cacher derrière différentes caractéristiques acoustiques suivant le genre musical traité. Comme le temps où l'on automatisera le choix adéquat d'un système de transcription en fonction d'un enregistrement de musique est encore loin, nous concentrons nos efforts sur un seul système symbolique, pouvant décrire un ensemble restreint de genres musicaux. Commençons par ces derniers. Sans grande originalité, nous nous restreignons à la musique occidentale, tonale ou non, faisant uniquement l'utilisation de la gamme tempérée. Nous pensons bien évidemment à la musique classique tonale, la musique sérielle ou autre musique savante du XX$^{\text{ème}}$siècle, au jazz, ou encore à toutes sortes de musiques actuelles : rock, pop, reggae... Si pour ces genres, les systèmes de notation les plus adéquats peuvent être de natures différentes (grille d'accords et thème mélodique pour le jazz, partition de chaque instrument pour la musique classique par exemple), une grande partie de l'information symbolique dans un enregistrement est portée par l'ensemble des notes jouées. Afin de rester le plus général possible, nous définissons une note comme un son musical

dont on peut percevoir une hauteur tonale [1]. Alors, une note peut être décrite par un certain nombre d'attributs : temps d'attaque (*onset* en anglais), temps de fin (*offset* en anglais), hauteur, intensité, timbre, instrument l'ayant produite... Comme une définition exacte de ces attributs – qui serait incontestable – n'existe pas, nous les définissons à la lumière de la perception humaine : onset et offset perçus, hauteur perçue, etc. La perception étant subjective, cela pourra poser certains problèmes dont nous parlerons dans la section suivante. Heureusement, dans de nombreux cas, il est possible d'associer la perception d'un attribut à des caractéristiques physiques du son.

La transcription automatique telle que nous l'entendons dans ce mémoire consiste, à partir d'un signal audio (enregistrement numérique monophonique), à estimer automatiquement ces notes via trois des attributs sus-nommés : onsets, offsets et hauteurs. Si nous écartons les autres attributs, ce n'est pas qu'ils soient inintéressants, c'est que le problème que nous venons de définir est déjà, comme nous le verrons, d'une immense difficulté, encore très loin d'être résolu : c'est en soit un défi majeur pour la communauté scientifique. Les sons dont aucune hauteur tonale ne peut être perçue sont également mis de côté. Cela concerne par exemples les sons produits par une batterie, qui sont pourtant porteurs d'informations.

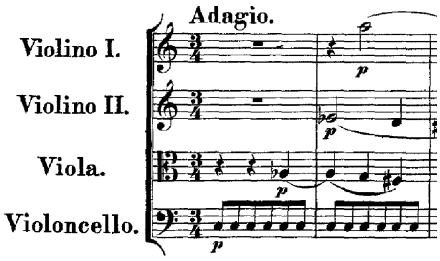
Puisque nous ne considérons que de la musique dont le support des notes est la gamme tempérée, la hauteur est codée par des nombres : nous utilisons l'échelle MIDI (annexe A) qui fait correspondre à chaque touche du piano (chaque note de la gamme) un entier naturel, 69 correspondant au $la_4$, c'est-à-dire le *la* du diapason à 440 Hz. Finalement, nos systèmes de transcription automatique devront prendre en entrée un enregistrement (fichier son au format .wav) et rendre en sortie un fichier texte (format .txt) dans lequel devront être écrits les attributs de l'ensemble des notes. Sur la figure 2, on peut trouver un exemple de fichier texte de sortie, ainsi que d'une représentation graphique quasiment équivalente appelée *Pianoroll* [2] en anglais et que nous appellerons *activations des notes*. Si cette représentation est *quasiment* et non *exactement* équivalente, c'est qu'elle ne permet pas de dissocier deux notes (jouées par des instruments différents par exemple) de même hauteur et ayant un support temporel se chevauchant. Malgré ce défaut, nous utiliserons cette représentation graphique pour illustrer dans ce mémoire des transcriptions de morceaux. Dorénavant, le terme transcription fera référence au format de nos fichiers de sorties : trouver chaque note présente dans un signal musical et décrire son temps d'attaque perçu, son temps de fin perçu, et sa hauteur perçue.

## Évaluer un système de transcription

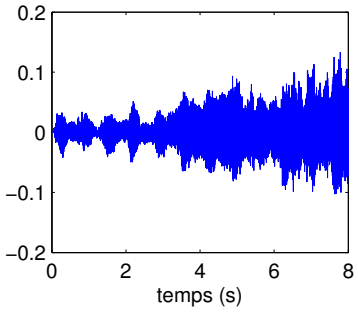
Si décrire les spécificités de la transcription automatique est relativement simple, évaluer la performance d'un système l'est nettement moins. Décrivons d'abord le processus traditionnel

1. Par définition, la hauteur tonale perçue est la fréquence de la sinusoïde pure qu'un auditeur peut associer à un son musical, si tant est qu'il le puisse [DV05].

2. Le terme *pianoroll* est employé car cette représentation ressemble aux rouleaux servant de support d'enregistrement aux pianos mécaniques.



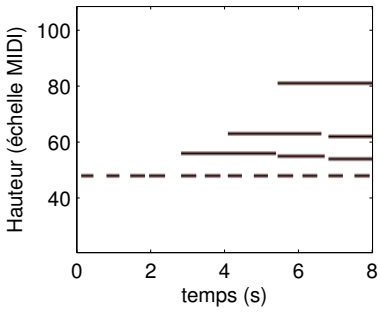

(a) Partition originale




(b) Forme d'onde d'un enregistrement

| <onset (s)> | <offset (s)> | <hauteur (MIDI)> |
|---|---|---|
| 0.13 | 0.47 | 48 |
| 0.81 | 1.16 | 48 |
| 1.45 | 1.86 | 48 |
| 1.97 | 2.40 | 48 |
| 2.83 | 3.24 | 48 |
| 2.83 | 5.38 | 56 |
| 3.47 | 3.89 | 48 |
| 4.08 | 4.46 | 48 |
| 4.08 | 6.61 | 63 |
| 4.78 | 5.17 | 48 |
| 5.42 | 5.84 | 48 |
| 5.42 | 6.70 | 55 |
| 5.42 | 9.33 | 81 |
| 6.10 | 6.49 | 48 |
| 6.80 | 7.19 | 48 |
| 6.80 | 8.07 | 54 |
| 6.80 | 8.07 | 62 |
| 7.50 | 7.90 | 48 |

(c) Transcription




(d) Activations des notes

**Figure 2** – Plusieurs représentations d'une œuvre musicale (les premières notes du quatuor « Dissonance » de W.A. Mozart) : un système de transcription automatique prend en entrée un enregistrement (forme d'onde (b)) et doit rendre en sortie une transcription (c). La représentation graphique (d) permet quant à elle de visualiser la transcription. La partition originale (a) a été ajoutée à titre indicatif.

d'évaluation. On crée en premier lieu une base de données, que nous appelons base d'évaluation, contenant à la fois des enregistrements de musique ainsi que les « véritables » transcriptions correspondantes, que nous appelons vérités terrain : elles peuvent être effectuées par exemple par des experts humains. Un système de transcription automatique analyse l'ensemble des enregistrements et propose pour chacun d'entre eux une estimation des transcriptions. On compare enfin chacune des estimations avec la vérité terrain correspondante : un certain nombre de mesures permettent de noter la qualité d'une transcription estimée, selon des critères arbitraires. Le principal problème du processus que l'on vient de décrire réside dans le terme « véritable » : il n'existe hélas pas nécessairement une unique transcription possible pour un morceau de musique ! Dans de nombreux cas de figures, on se trouve confronté à des ambiguïtés pouvant être appréhendées de différentes manières. Ces ambiguïtés peuvent venir par exemple du caractère subjectif que avons donné aux notions d'onset, d'offset et de hauteur perçus. Nous en listons ici quelques exemples, puis nous concluons sur des solutions possibles pour pallier ces équivoques lors de l'évaluation d'un système.

**Onsets et offsets** Ils peuvent être impossibles à localiser précisément si l'attaque ou l'arrêt d'une note ne s'effectue pas de manière brusque. Prenons l'exemple d'une note de piano que l'on laisse résonner indéfiniment. Son énergie va décroitre petit à petit vers zéro, et si l'on demande à deux experts de situer temporellement son temps de fin, il est probable qu'ils donnent des résultats éloignés puisque leur oreille ne va pas forcément avoir une sensibilité identique. Ceci est d'autant plus vrai si la note se trouve noyée dans un flot d'autres notes et sons. On peut également prendre l'exemple de la réverbération : même si l'on étouffe brusquement cette note de piano, en présence de réverbération, elle peut résonner pendant un certain temps (pouvant aller jusqu'à plusieurs secondes), difficile à apprécier précisément. Pour les attaques de notes, on peut considérer (à tort ?) que le problème est moins prononcé puisque la plupart des instruments (acoustiques du moins) génèrent des notes dont les attaques sont relativement bien définies.

**Superposition de notes** Il peut arriver dans une pièce de musique que plusieurs types d'instruments jouent ensemble une même ligne mélodique, à l'unisson ou à l'octave par exemple, afin de combiner leur timbre et ainsi créer de nouvelles sonorités. Dans l'orchestre par exemple, il arrive souvent que les contrebasses doublent les violoncelles à l'octave inférieure. Dans ce cas, il existe deux possibilités de transcription : ou bien transcrire chaque note jouée par chaque instrument, ou bien considérer le mélange des instruments comme un nouvel instrument et transcrire les notes jouées par ce « super-instrument ». Le même principe de superposition de notes peut se retrouver également au sein d'un même instrument de musique. C'est le cas de l'orgue et de ses jeux par exemple. Suivant les jeux que l'on sélectionne, le jeu d'une touche du clavier va actionner un ou plusieurs tuyaux en même temps. De plus, certains jeux ont pour caractéristique de ne pas produire la note jouée au clavier : les jeux appelés mutations sont des jeux transpositeurs (le *nazard* par exemple qui fait entendre la quinte) ! Là aussi il existe alors deux

possibilités : transcrire chaque note produite par chaque tuyau, ou bien tenter de transcrire les notes résultantes, c'est-à-dire les notes qui ont été jouées sur le clavier, indépendamment des jeux actionnés.

**Répétition d'une même note, ou non** Prenons l'exemple du violon, dont les attaques de notes peuvent être relativement douces. Et plaçons-nous dans le cas où une unique corde est excitée par l'archet. Si le violoniste maintient sa main gauche, qui permet de définir la hauteur de la note jouée, dans une certaine position, et que l'archet dans sa main droite effectue des aller-retour sans jamais quitter la corde, faut-il considérer chaque changement de coup d'archet comme l'attaque d'une nouvelle note ? Là encore deux stratégies sont possibles et il n'existe pas une unique transcription possible. On pourrait bien sûr décréter que si le changement de coup d'archet est suffisamment brusque, l'on décide qu'il y a changement de note, mais comment définir le niveau minimum de vivacité ? Et si l'on considère l'effet de trémolo, les coups d'archets peuvent être brusques sans qu'il soit forcément pertinent de transcrire l'ensemble des attaques : le trémolo peut en effet plutôt être considéré comme un effet de timbre. Un autre exemple de ce type d'ambiguïté, qui finalement traduit la difficulté de définir les frontières d'un son musical, peut aussi se retrouver dans le chant humain : faut-il scinder une note tenue à chaque changement de syllabe si des paroles sont chantées ?

**Stabilité de hauteur** Restons avec la voix humaine, décidément source de nombreuses difficultés. Une de ses particularités (comme de nombreux instruments de musique en réalité) est qu'elle peut moduler sa hauteur de manière continue : vibratos ou glissandos sont deux exemples d'effets exploitant ce phénomène. Alors si un musicien, entre deux notes, effectue un glissement continu, faut-il transcrire également toutes les notes par lesquelles il passe, comme si ce glissement s'effectuait sur un piano par exemple ? Deux possibilités de transcription existent encore ici et il est donc impossible de définir une vérité terrain unique dans ce cas.

**Des solutions ?** Nous venons de souligner le fait que pour une pièce de musique, il n'existe pas nécessairement une unique transcription possible. Ainsi, le but d'un système de transcription automatique n'est pas de trouver *la* transcription, mais plutôt *une* transcription possible pour une œuvre donnée. Alors comment évaluer de manière objective un système de transcription, si l'on considère toutes les ambiguïtés que nous avons relevées ? Une première solution serait de proposer pour chaque enregistrement de la base d'évaluation plusieurs vérités terrain, une pour chaque possibilité de transcription. Pour évaluer une estimation de transcription, on pourrait alors la comparer à chacune des vérités terrain et garder le meilleur score. Le principal problème est qu'aujourd'hui, il n'existe pas de telles bases de données. Une deuxième possibilité serait que la base d'évaluation soit constituée uniquement d'enregistrements ne possédant aucune ambiguïté de transcription (si tant est que cela soit possible). Il n'y aurait ainsi qu'une vérité terrain et il serait facile de se donner une idée des performances d'un système. C'est la solution que nous

avons adoptée, dans la mesure du possible, lors de la création d'une sous-base de la base QUASI [Webf], que nous présentons dans l'annexe C (p.171). Enfin, une dernière solution serait de trouver des métriques d'évaluation insensibles à ces ambiguïtés. Par exemple, pour remédier aux problèmes de localisation d'onsets et d'offsets, ce qui est fait généralement est de considérer que deux transcriptions d'une note présente dans un signal sont identiques si les onsets, et seulement eux, ne sont pas trop éloignés (typiquement un différentiel de 50 ms). La valeur des offsets n'est donc pas prise en compte. Quelles que soient les réponses choisies pour répondre au caractère équivoque de la transcription, il n'existe pas de solution miracle. En revanche, on peut espérer qu'une grande majorité des notes d'une vérité terrain soient présentes de manière incontestable, et que les ambiguïtés restent marginales. Si tel est le cas, alors le principe d'appréciation d'un système de transcription sur une base d'évaluation permet de donner une bonne idée de sa qualité.

## Décompositions de représentations temps-fréquence

Maintenant que nous avons défini la tâche de transcription et introduit une manière dont on peut évaluer un système qui automatiserait ce processus, nous pouvons introduire les solutions apportées à la transcription automatique. Nous pouvons expliquer en quoi ce problème est d'une immense difficulté. Nous pouvons aussi tenter de proposer de nouvelles solutions. Traditionnellement, cette tâche est traitée par la résolution d'un autre problème : l'estimation de hauteurs multiples (ou *multipitch* en anglais, terme que nous emploierons en raison de sa concision). Cela consiste à estimer l'ensemble des hauteurs des notes présentes à un instant donné d'un signal de musique. Si l'on y parvient à des instants réguliers d'un enregistrement, on peut alors construire une matrice d'activations des notes (on trouve une illustration d'une telle matrice sur la figure 2 (d)) et la traiter afin d'obtenir une transcription, au format texte que nous nous sommes imposé. De très nombreuses solutions sont apportées pour résoudre l'estimation *multipitch* ou la transcription automatique, mais ces problèmes restent très ouverts, principalement en raison de deux difficultés majeures. D'abord les sons dont on perçoit une hauteur tonale (les notes de musique en somme) peuvent être de différentes natures : harmoniques, inharmoniques, voire même non déterministes. Il est ainsi difficile de modéliser une note de manière générale, et donc souvent nécessaire d'émettre des hypothèses plus ou moins avérées sur les types de sons que l'on souhaite traiter. La deuxième difficulté réside dans le fait que les sons émis simultanément par les instruments de musique interfèrent aussi bien dans le domaine temporel que dans le domaine spectral : il est impossible de les identifier indépendamment les uns des autres ou même de connaitre de manière certaine leur nombre.

Récemment, de nouvelles méthodes sont apparues, qui consistent à modéliser un signal, ou une transformation de celui-ci, comme une somme d'éléments de base, appelés atomes ou noyaux, que l'on pourrait qualifier d'éléments « mi-niveau », à la fois directement liés au signal physique et porteurs d'information symbolique. Nous pouvons citer par exemple les techniques

de représentations parcimonieuses [AP04, AP06, Lev07, ONP12], ou de décompositions de représentations temps-fréquences positives ($\text{RTF}^+$) [SRS08b, Ber09, GE11]. Ce sont ces dernières qui nous intéressent. Une représentation temps-fréquence (RTF) est une transformation, inversible ou non, d'un signal temporel dans le plan temps-fréquence. Il en existe de nombreux types, la plus connue étant la Transformée de Fourier à Court Terme (TFCT) qui consiste à découper le signal en une série de courtes trames recouvrantes et à calculer la transformée de Fourier discrète de chacune d'entre elles. D'une manière générale, une RTF est composée de valeurs complexes dont le module permet d'exprimer la puissance d'un signal à un temps donné, pour une fréquence donnée. Une $\text{RTF}^+$sera donc typiquement calculée en appliquant une fonction positive à une RTF. Un large panel de méthodes permettant de décomposer ces représentations positives existe dans la littérature. Elles suivent systématiquement un même schéma. En premier lieu, on imagine un modèle de $\text{RTF}^+$. Par exemple on peut supposer que chacune de ses colonnes se décompose comme une somme pondérée de spectres de base (les atomes), chacun d'entre eux représentant le spectre d'une note de musique. Les coefficients de pondération incarnent alors les énergies de chaque note à un temps donné. Dans un second temps, après la construction du modèle, un algorithme pour estimer ses paramètres en fonction d'un signal d'entrée est proposé. Si le modèle est pertinent, et l'algorithme permettant d'estimer ses paramètres suffisamment performant, alors la décomposition qui en résulte pourra être utilisée pour répondre à de nombreux problèmes, comme l'analyse automatique de signaux musicaux (la transcription automatique étant l'application principale qui nous intéresse), ou la séparation de sources, qui consiste à séparer un ou plusieurs instruments d'un mélange de sources.

## Les contributions de la thèse

Dans cette thèse, nous nous concentrons sur ces techniques de décomposition. Comme nous l'avons fait remarquer, on peut étudier celles-ci sous deux angles différents : les modèles de $\text{RTF}^+$ et les algorithmes permettant d'estimer leurs paramètres.

**L'Analyse Probabiliste en Composantes Latentes : étude et amélioration.** Pour définir ces algorithmes, il existe de nombreux cadres mathématiques, qu'ils soient analytiques ou probabilistes, le plus connu étant probablement l'approche analytique de la factorisation de matrices positives (NMF) [LS99, SB03]. Dans cette thèse, c'est un autre outil d'analyse qui est utilisé et étudié, intitulé Analyse Probabiliste en Composantes Latentes (PLCA) [Hof01, SRS08a]. Si ce choix est arbitraire, on peut quand même relever quelques atouts de la PLCA. Elle possède tout d'abord l'avantage de pouvoir facilement dériver des algorithmes d'estimation de paramètres pour n'importe quel modèle (tant qu'il reste linéaire) de $\text{RTF}^+$, et en particulier des modèles avec invariance par translation, utiles pour l'analyse de signaux musicaux. De plus, elle permet aisément d'appliquer des aprioris sur les paramètres d'un modèle, afin de les encourager à converger vers une solution considérée comme plus significative, ou probable. Ainsi, un premier

objet de cette thèse est de proposer, dans le cadre de la PLCA, des outils permettant d'ajouter de l'information sur la nature des paramètres à estimer, et ce indépendamment du modèle de RTF$^+$ considéré. Un certain nombre de nouveaux aprioris (parcimonie, continuité temporelle, ressemblance, monomodalité) sont introduits, pouvant être appliqués au choix aux paramètres d'un modèle. Outre l'ajout d'aprioris, nous étudions la pertinence de pouvoir ralentir la vitesse de convergence d'un sous-ensemble de paramètres. Cela peut s'avérer efficace si, par exemple, on sait que les paramètres dont on ralentit la convergence sont initialisés intelligemment et proches de leur vraie valeur. Enfin, nous introduisons un nouveau méta-modèle de type PLCA permettant de modéliser une structure temporelle pour un sous-ensemble de variables cachées sous-jacentes, en modélisant par exemple leur probabilité de transition d'un instant à l'autre, plutôt que directement leur probabilité à un instant donné. Nous verrons que ce méta-modèle peut se greffer à n'importe quel modèle de RTF.

**Les modèles de RTF$^+$.** Un deuxième aspect de cette thèse est de proposer de nouveaux modèles de RTF$^+$, suffisamment expressifs pour pouvoir s'adapter à n'importe quel type d'instrument harmonique ou légèrement inharmonique. En particulier, un important travail est apporté pour pouvoir modéliser des notes possédant simultanément des variations temporelles de fréquence fondamentale et d'enveloppe spectrale, comme cela est le cas par exemple pour celles générées par la voix humaine. A travers cette expressivité du modèle, nous étudions la pertinence de supposer ou non une certaine redondance dans les signaux de musique, surtout quand ceux-ci se complexifient en fonction du nombre d'instruments présents ou du genre musical. Nous avons également porté une grande attention à la sensibilité des modèles proposés aux maxima locaux lors de l'estimation des paramètres.

**Les applications.** En jouant sur les différents modèles imaginés dans ce travail ainsi que sur les outils permettant de personnaliser la PLCA à la nature des données, plusieurs algorithmes sont proposés pour la décomposition de signaux musicaux. L'application principale que nous en faisons est bien entendue la transcription automatique. S'il reste encore beaucoup de recherches à effectuer pour ces problèmes, les résultats obtenus, représentant l'état de l'art pour des signaux de musique réputés difficiles à analyser (musiques actuelles, rock, etc.), permettent de souligner la pertinence des travaux effectués. Les modèles qui sont proposés dans cette thèse ont été créés dans l'optique de traiter le problème de la transcription de musique, mais ils peuvent également être appliqués à un autre problème : la séparation de sources. Aussi, deux applications supplémentaires sont traitées dans ce mémoire. La première est l'extraction automatique de la mélodie principale et la seconde l'extraction de notes assistée par l'utilisateur.

## Plan du document

### Première partie : *État de l'art et cadre de la thèse* (p.13)

Le but de cette première partie n'est pas de dresser un état de l'art des techniques de transcription automatique, mais plutôt de s'intéresser à une classe particulière de méthodes permettant, entre autres, de répondre à ce problème : les décompositions de signaux en éléments significatifs. Plus exactement, nous nous concentrons sur les méthodes de factorisation de RTF. Dans le chapitre 1, on présente ainsi un panorama des méthodes existantes dans la littérature. Elles seront présentées selon deux angles de vues différents : les manières de modéliser une RTF et les outils mathématiques permettant d'estimer les paramètres d'un modèle donné. Le chapitre 2 sera essentiellement consacré aux outils utilisés dans la suite de ce mémoire : la PLCA et sa variante avec invariance par translation, ainsi que la transformée à Q constant (CQT), qui sera la RTF utilisée.

### Deuxième partie : *Aller plus loin avec la PLCA* (p.39)

Dans cette partie nous présentons d'abord de nouveaux outils à appliquer sur les paramètres d'un modèle de type PLCA pour mieux contrôler la manière dont ils convergent lors d'une décomposition. Le chapitre 3 est en effet consacré à l'introduction de quatre nouveaux aprioris, permettant d'incorporer de l'information : aprioris de parcimonie [FBR12a, FBR13], de continuité temporelle [FBR13], de ressemblance et de monomodalité [FBR11a, FBR13]. Dans le chapitre 4, nous proposons une astuce permettant de contrôler la vitesse de convergence de tel ou tel ensemble de paramètres, et nous étudions la pertinence de cette idée. Le chapitre 5 est quant à lui consacré à l'introduction d'un nouveau modèle générique appelé LCATS (pour *Latent Component Analysis with Temporal Structure*), qui permet d'insérer un modèle de structure temporelle dans la PLCA. Une déclinaison markovienne de la LCATS (MLCATS pour *Markovian LCATS)* sera proposée. Ce dernier modèle, en raison de sa relative nouveauté, ne sera pas testé dans les applications de la partie IV.

### Troisième partie : *Les modèles de RTF* (p.85)

La troisième partie est consacrée à l'introduction de nouveaux modèles de CQT permettant d'analyser les structures harmoniques dans un signal audio : le modèle HALCA (pour *Harmonic Adaptive Latent Component Analysis*) [FBR11a, FBR13] au chapitre 6 et le modèle BHAD (pour *Blind Harmonic Adaptive Decomposition*) [FBR12a] au chapitre 7. Ces deux modèles ont été imaginés pour traiter le problème de l'estimation de hauteurs multiples et de la transcription automatique, et leur caractéristique est de pouvoir s'adapter à n'importe quel type de note harmonique ayant des variations de fréquence fondamentale et d'enveloppe spectrale. Nous montrons pour chacun d'entre eux comment les outils présentés dans la deuxième partie peuvent être appliqués, et nous évaluons leurs bénéfices via une tâche d'estimation *multipitch*.

### Quatrième partie : *Applications* (p.123)

La principale application que nous faisons des algorithmes mis au point dans ce mémoire est la transcription automatique de musique polyphonique, telle que définie dans l'introduction. Les différents modèles imaginés dans la partie III, combinés aux outils de la partie II (mise à part le modèle LCATS) nous permettent de décliner plusieurs systèmes de transcription que nous évaluons sur plusieurs bases de données dans le chapitre 8. Si les modèles créés dans cette thèse ont été imaginés dans l'optique de traiter le problème de transcription, nous montrons dans le chapitre 9 qu'ils peuvent être utilisés pour la séparation de sources. Aussi, nous présentons dans ce chapitre deux autres applications : l'extraction de mélodie principale [FLBR12] ainsi que la séparation de notes assistée par l'utilisateur via une interface graphique [FBR12a].

Pour finir, nous concluons et faisons un bilan de l'ensemble de nos travaux. Nous examinons également les perspectives d'amélioration et les axes de recherche futurs.

## Publications en lien avec cette thèse

### Article paru dans une revue internationale

[FBR13] B. Fuentes, R. Badeau et G. Richard : Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Transactions on Audio Speech and Language Processing*, 21(9) :1854-1866, 2013.

### Articles parus dans les actes de conférences

[FBR11a] B. Fuentes, R. Badeau et G. Richard : Adaptive harmonic time-frequency decomposition of audio using shift-invariant PLCA. *In Proc. of ICASSP*, pages 401–404, Prague, République Tchèque, 2011.

[FBR11b] B. Fuentes, R. Badeau et G. Richard : Analyse des structures harmoniques dans les signaux audio : modéliser les variations de hauteur et d'enveloppe spectrale. *In GRETSI*, Bordeaux, France, 2011.

[FLBR12] B. Fuentes, A. Liutkus, R. Badeau et G. Richard : Probabilistic Model for main melody extraction using constant-Q transform. *In Proc. of ICASSP*, pages 5357–5360, Kyoto, Japon, 2012.

[FBR12] B. Fuentes, R. Badeau et G. Richard : Blind Harmonic Adaptive Decomposition Applied to Supervised Source Separation. *In Proc. of EUSIPCO*, pages 93–99, Bucarest, Roumanie, 2012.

# Première partie

# État de l'art et cadre de la thèse

# Chapitre 1

# Factorisations de représentations temps-fréquence pour la transcription automatique

## 1.1 Introduction

Il existe une littérature nombreuse dressant l'état de l'art de la transcription automatique. Une approche historique ainsi qu'un excellent panorama permettant d'obtenir une vision globale du problème peuvent être par exemple lus dans [Ber09]. D'autres bibliographies très complètes sont également consultables dans [Cem04] ou [Emi08]. Des livres traitant de la transcription automatique ou du problème lié d'estimation de hauteurs multiples existent également, parmi lesquels [KD06] ou [CJ09]. Dans ce chapitre, plutôt que du problème de transcription automatique en tant que tel, nous nous concentrons plutôt sur une classe de méthodes particulière permettant, entre autres, de le traiter : les techniques de décomposition en éléments significatifs. Elles consistent à représenter, ou modéliser, un signal audio comme une somme d'éléments de base, porteur de sens et d'information. Parmi elles, on va particulièrement se focaliser sur ce que nous appelons les décompositions de représentations temps-fréquence. De telles décompositions, si elles peuvent répondre aux problèmes d'estimation de hauteurs multiples et de transcription [SB03, VBB08, KNS07, BD11], sont également utilisées dans d'autres domaines comme la séparation de sources [Vir07, CZA06b, OF10], l'extraction de mélodie [DDR11, FLBR12], la reconnaissance d'instruments [GE11], l'estimation de localisation de temps forts [OKS12] ou encore l'alignement audio/partition [Con06]. Nous présenterons d'ailleurs nous-même dans ce mémoire, en sus de la transcription (chapitre 8), le moyen d'appliquer nos recherches à des problématiques de séparation de sources (chapitre 9).

Afin de construire un état de l'art le plus pertinent possible, nous allons procéder de la manière suivante :

—∘ D'abord nous expliquerons le principe général de ces méthodes. Nous tenterons de les

grouper selon plusieurs catégories cohérentes, en mettant en exergue les différentes stratégies possibles.

—◦ Dans une deuxième étape seulement, nous nous concentrerons sur les différents outils mathématiques permettant d'effectuer ces décompositions.

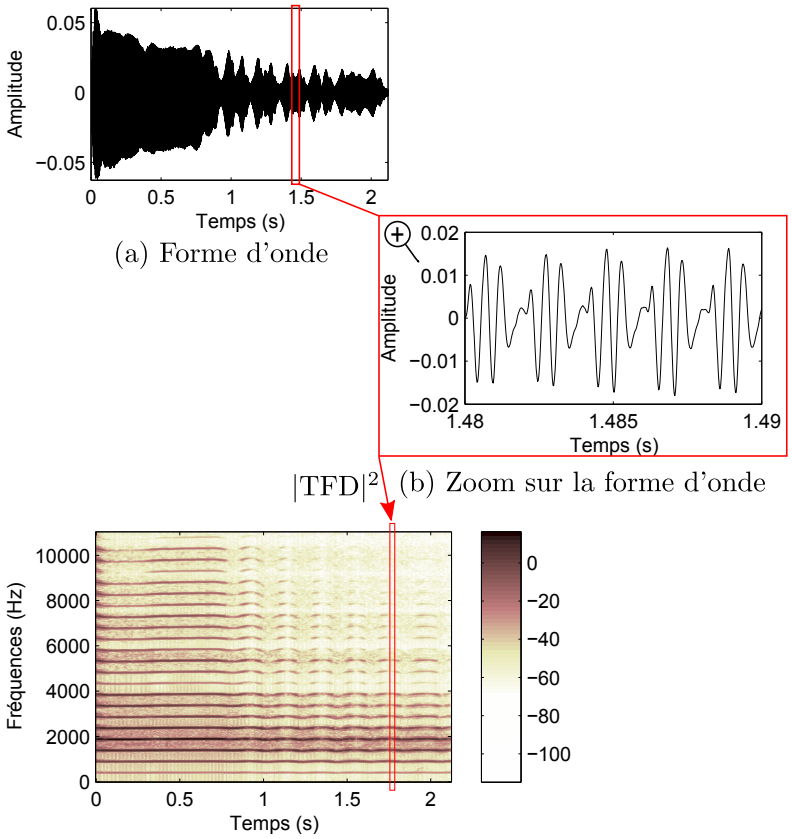
Dans l'ensemble de ce mémoire, nous notons $\boldsymbol{V}$, de coefficients $V_{ft}$ ($f$ et $t$ représentent respectivement l'indice de fréquence et celui de temps) toute représentation temps-fréquence à coefficients positifs (RTF$^+$), qui est généralement calculée à partir d'une représentation temps-fréquence complexe (RTF), notée $\boldsymbol{X}$ et de coefficients $X_{ft}$, en lui appliquant une transformation positive (par exemple $V_{ft} = |X_{ft}|^2$).

Avant de présenter ces méthodes de décomposition, et comme pour tout problème de traitement de données, il nous semble judicieux de discuter certaines caractéristique de nos observations.

## 1.2 Observer des notes de musique

Le but final étant de trouver l'ensemble des notes qui constituent un morceau de musique, il peut être utile d'observer des signaux de notes isolées. Cela nous permettra de comprendre leur nature et d'extraire certaines de leurs caractéristiques. Au hasard, prenons pour commencer une note de trompette. Pour observer ce signal, nous pouvons visualiser sa forme d'onde, qui correspond au signal brut tel qu'il est échantillonné et enregistré à la sortie d'un microphone, mais nous pouvons également lui appliquer une transformation afin de le représenter dans le plan temps-fréquence. Une telle représentation permet de voir l'évolution du contenu spectral de la note au cours du temps. La figure 1.1 présente ces deux représentations. La RTF$^+$ utilisée ici est le spectrogramme, qui consiste à calculer le module au carré de la transformée de Fourier discrète (TFD) sur de courtes trames recouvrantes de signal. Que pouvons-nous alors apprendre de ces observations ? D'abord, en zoomant sur une partie de la forme d'onde, on peut remarquer son caractère périodique (ou pseudo-périodique). C'est la physique qui permet d'expliquer ce phénomène de périodicité [CK08], en réalité partagé aussi par la voix humaine et par de nombreux instruments. On se rend en effet compte, en étudiant l'acoustique des instruments de musique, que les notes produites par bon nombre d'entre eux sont constituées de partiels (sinusoïdes) dont les fréquences sont en rapport harmonique, c'est-à-dire multiples d'une certaine fréquence fondamentale [1]. La théorie de Fourier permet d'établir une équivalence entre périodicité et harmonicité, et en observant la colonne du spectrogramme correspondant à cette partie du signal, on constate bien la présence de pics pour des fréquences multiples d'une certaine fréquence fondamentale. Nous dirons alors de notre signal que c'est une note harmonique, dont les partiels qui la composent sont des *harmoniques*, et dont la fréquence fondamentale instantanée au temps $t$ est $f_0(t)$. Cette fréquence nous intéresse tout particulièrement, car si la notion de hauteur

1. Cela peut venir par exemple du caractère harmonique de la source excitatrice d'un instrument à vent, ou de la géométrie d'un instrument à percussion.




(a) Forme d'onde (b) Zoom sur la forme d'onde

(c) Spectrogramme (dB)

**Figure 1.1** – Illustration d'un signal de note isolée de trompette.

perçue ne peut se référer à aucun attribut physique d'un signal de manière générale, on sait que pour les notes harmoniques, elle est lui étroitement liée [POFP05]. Nous utiliserons d'ailleurs par abus de langage indifféremment les termes *hauteur*, *fréquence fondamentale* ou *pitch* (terme anglais) pour désigner la hauteur perçue d'une note harmonique à un instant donné.

Une deuxième observation est que le contenu spectral de la note évolue au cours du temps. On peut en effet remarquer une évolution temporelle des amplitudes des partiels, et surtout de leur fréquence. On pourrait alors se demander comment donner à cette note une hauteur globale (c'est bien ce que nous souhaitons transcrire) si la trajectoire de fréquence fondamentale fluctue. Et bien à supposer que cette fréquence fondamentale reste toujours dans un intervalle de plus ou moins un quart de ton autour de la hauteur d'une note particulière de l'échelle MIDI, on pourra désigner cette dernière comme étant la note jouée. Nous supposons d'ailleurs par la suite que toutes les notes qui constituent un enregistrement vérifient cette caractéristique. Si jamais la trajectoire $f_0(t)$ sort de cet intervalle, alors on supposera qu'une nouvelle note débute. Ce caractère fluctuant de hauteur n'est pas partagé pas tous les instruments : les partiels des notes de piano, clavecin, orgue et bien d'autres restent stables en fréquence.

Enfin, un dernier fait remarquable qui nous semble important est que l'énergie de la note

n'est pas nécessairement nulle entre ses partiels. Cette énergie est en fait lié essentiellement à la présence de bruit : on entendra le souffle de la trompette ou de la flute, le bruit du marteau du piano, le frottement de l'archet sur la corde du violon.

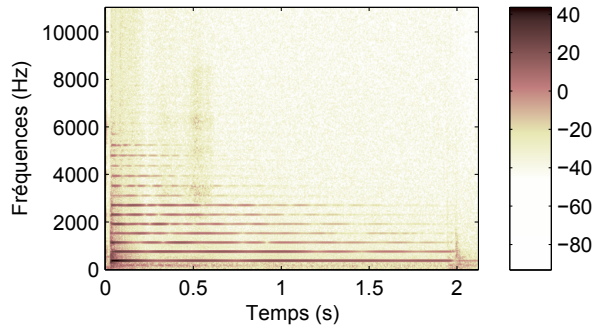
En mettant de côté les sons percussifs dont on ne perçoit aucune hauteur tonale, la plupart des notes rencontrées dans la musique sont harmoniques, et il est possible de déduire leur hauteur en estimant la trajectoire de leur fréquence fondamentale. Mais il existe aussi de nombreux sons dont on peut percevoir une hauteur tonale, sans être pour autant harmoniques. Pour certains, comme les notes de piano, l'inharmonicité est légère (la fréquence des partiels est seulement légèrement déviée par rapport à la fréquence théorique des harmoniques [Fle64]), et la notion de hauteur perçue reste quand même liée à celle d'une fréquence fondamentale. Pour d'autres par contre, comme les sons de cloches, les fréquences des partiels ne sont plus du tout en rapport harmonique, et la hauteur perçue peut alors dépendre de l'auditeur, ou du contexte dans lequel ces sons sont utilisés. Sur la figure 1.2, trois exemples supplémentaires de spectrogrammes de notes de différentes natures sont illustrés.

## 1.3 Modéliser les RTF

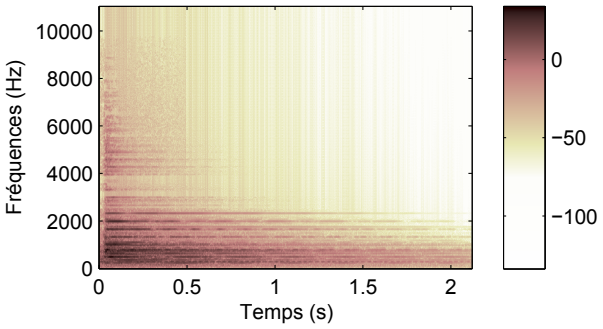
### 1.3.1 Introduction

Maintenant que l'on a observé des notes de musiques et que l'on a pu en extraire quelques caractéristiques, nous avons une meilleure idée des éléments qui constituent un signal musical. Nous allons donc pouvoir nous concentrer sur ce qui nous intéresse ici : les décompositions en éléments significatifs. Rappelons d'abord leur principe, déjà brièvement présenté dans l'introduction. L'idée générale est de modéliser un signal, ou une transformation de celui-ci – nous parlerons essentiellement dans ce chapitre de représentations temps-fréquence à coefficients positifs ($\text{RTF}^+$) – comme une somme d'éléments de base, que nous pouvons aussi appeler atomes ou noyaux. Ces atomes doivent être à la fois des éléments physiques, c'est-à-dire de même nature que le signal (c'est logique puisqu'ils doivent permettre de le décrire) et porteurs d'information. Ainsi, si tel atome est utilisé à un moment donné pour modéliser le signal, nous pourrons déduire que ce dernier porte la même information que celle de l'atome en question. Une des manières les plus classiques et les plus utilisées pour mettre en application cette idée est de modéliser chaque colonne de $\text{RTF}^+$ comme une somme de spectres de base (nos atomes). Si par exemple chaque atome représente une note de la gamme, alors il sera possible de déduire l'ensemble des notes présentes à chaque instant et ainsi de traiter le problème de l'estimation de hauteurs multiples. L'exemple le plus classique de ce type de modèle est la factorisation de matrices positives classique (NMF) [LS99, SB03], où une $\text{RTF}^+$ $\boldsymbol{V}$ (de dimension $F \times T$) à décomposer est modélisée de la manière suivante :

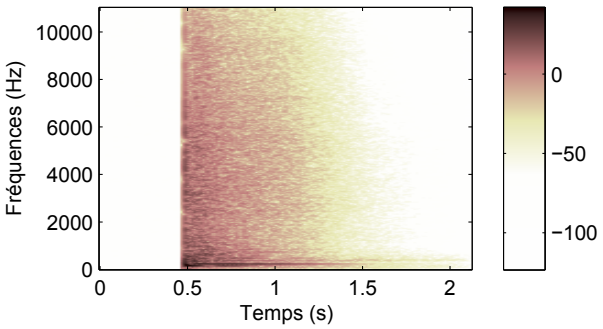
$$\boldsymbol{V} \approx \hat{\boldsymbol{V}} = \boldsymbol{S} \times \boldsymbol{A}, \tag{1.1}$$




(a) Note de piano




(b) Cloche d'église




(c) Coup de caisse claire

**Figure 1.2** – Spectrogrammes de trois notes de musique : une note de piano, de nature légèrement inharmonique ; un son de cloche, dont les partiels ne sont pas en rapport harmonique ; un coup de caisse claire, dont on ne perçoit aucune hauteur tonale.

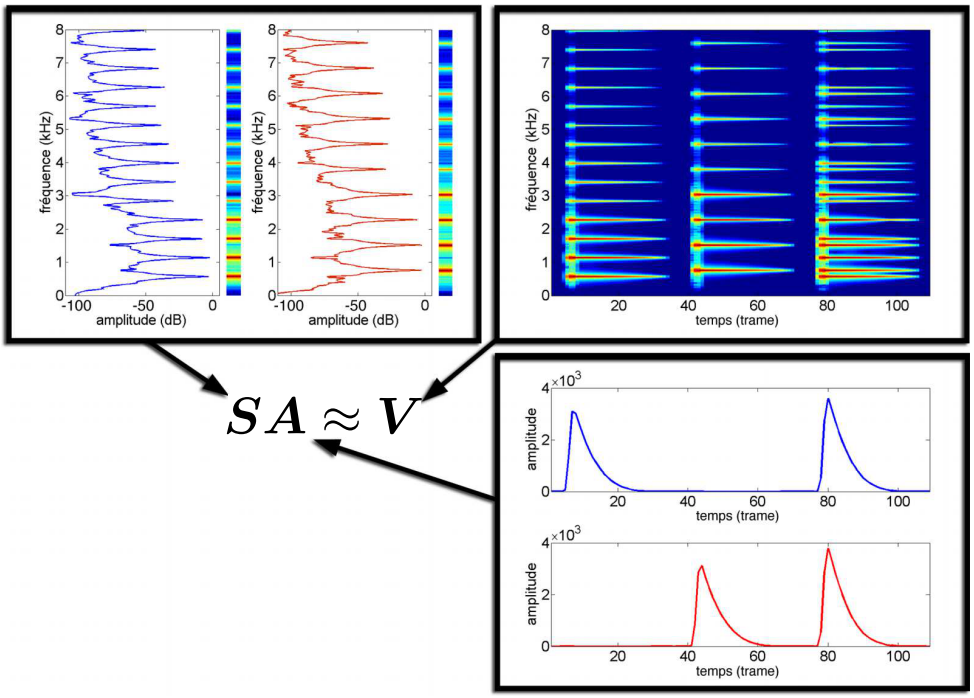



**Figure 1.3** – Principe de la NMF classique, d'après [Hen11]. La RTF$^+$ d'entrée (ici un spectrogramme) est composée de deux notes harmoniques, d'abord jouées successivement, puis simultanément.

soit

$$V_{ft} \approx \hat{V}_{ft} = \sum_{z=1}^{Z} S_{fz} A_{zt}. \tag{1.2}$$

$\boldsymbol{S}$ (de dimension $F \times Z$) représente alors les spectres de base, $\boldsymbol{A}$ (de dimension $Z \times T$) des coefficients de pondération dépendant du temps, autrement appelés *activations temporelles* et $f$, $z$ et $t$ sont les indices de fréquence, de numéro d'atome et de temps. Ce qui rend la NMF puissante est la contrainte qui l'accompagne : les paramètres $\boldsymbol{S}$ et $\boldsymbol{A}$, à l'instar des observations $\boldsymbol{V}$ doivent être à coefficients positifs ou nuls. Cette non-négativité, qui n'autorise que des combinaisons positives d'atomes, eux même contraints à rester dans le même espace que les données à analyser, permet d'assurer au mieux une décomposition significative, comme expliqué dans [LS99]. Pour la transcription automatique, la NMF est intéressante si les spectres de base représentent des spectres de notes dont on connait (ou dont on peut déduire) la hauteur. Le modèle de la NMF est illustré sur la figure 1.3.

De nombreux autres modèles de RTF$^+$ – variantes ou extensions de la NMF par exemple – permettant de mieux prendre en compte certaines caractéristiques des signaux musicaux ont également été proposés dans la littérature. Pour répondre par exemple au cas des notes présentant une évolution d'enveloppe spectrale, on pourra se référer au modèle dynamique [HBD11b]. D'autres travaux ont été menés pour traiter le cas de notes présentant des variations de fréquence fondamentale, en imaginant des modèles avec invariance par homothétie [HBD11c], ou

par translation [SRS08b].

Indépendamment du modèle retenu, il est possible de diviser grossièrement le problème de décomposition de signal en deux sous-tâches, pouvant être effectuées conjointement ou séquentiellement suivant la stratégie adoptée :

- —∘ la création de l'ensemble des noyaux : il faudra qu'ils soient suffisamment expressifs pour pouvoir bien décrire le signal à décomposer, mais également porteurs d'informations ;
- —∘ la décomposition à proprement parler du signal comme une combinaison de ces différents noyaux.

Même s'il n'existe pas de frontière bien définie entre les méthodes existantes, nous avons décidé de les regrouper selon trois sous-catégories, selon qu'elles soient non-supervisées, supervisées, ou semi-supervisées.

### 1.3.2 Les méthodes non-supervisées

Ces méthodes, aussi appelées méthodes aveugles, sont nommées ainsi car elles ne font pas appel à une quelconque étape d'apprentissage, et aucune connaissance sur la nature des signaux n'est fournie. Ici l'ensemble des atomes ainsi que la décomposition du signal sur ceux-ci sont estimés conjointement, grâce à l'exploitation des redondances intrinsèques dans un signal. L'idée principale est que si l'on parvient à décrire, décomposer un signal, aussi complexe soit-il, avec un nombre restreint d'atomes, alors ceux-ci seront significatifs. Pour mieux comprendre, reprenons le cas du modèle classique de NMF (équation (1.2)), et choisissons un nombre d'atomes $Z$ de sorte que $FZ + ZT \ll FT$. Si l'on arrive à estimer automatiquement les matrices $\boldsymbol{S}$ et $\boldsymbol{A}$ (nous verrons comment dans la section suivante) qui expliquent au mieux une observation $\boldsymbol{V}$, alors la NMF se comportera comme un outil de réduction de dimension : c'est de cette manière que la redondance du signal est modélisée. Pour la transcription, on espère bien entendu que ces atomes en petit nombre représentent des notes. Autrement dit, dans le cas de la NMF, on espère que chaque colonne de $\boldsymbol{S}$ représente le spectre d'une note de musique, dont on pourra estimer la fréquence fondamentale par une quelconque technique d'estimation de hauteur simple (par exemple [VP05]). Les activations nous renseigneront alors sur la présence ou l'absence de chaque note au cours du temps, et nous serons capables d'estimer le nombre de notes présentes ainsi que leur hauteur correspondante pour chaque colonne de $\boldsymbol{V}$.

Ces méthodes sont assez séduisantes dans le sens où aucune information sur la nature des signaux autre que la redondance supposée n'est nécessaire pour décomposer le signal. Il est donc théoriquement possible de modéliser n'importe quel évènement sonore, à condition qu'il se répète dans le temps. Cependant, ce type de méthode possède un inconvénient de taille pour la transcription : il est sous-contraint, et rien ne nous assure, comme nous le souhaiterions, que les atomes trouvés représentent des notes.

### 1.3.3 Les méthodes supervisées

Une solution attrayante pour contraindre la décomposition et s'assurer que les atomes correspondent bien à des objets qui nous intéressent, est de réaliser des décompositions supervisées. Cela consiste à décomposer un signal en utilisant des atomes fixes, ayant été appris au préalable lors d'une étape d'apprentissage. C'est par exemple la stratégie adoptée dans [LVRD08] dans le cadre des représentations parcimonieuses de signaux temporels. Dans le cas de la NMF, on peut citer [DCL12] qui propose un système en ligne pour la transcription de musique. Son principe est simple. Lors de l'étape d'apprentissage, des noyaux sont appris sur des signaux de notes isolées d'instruments en leur appliquant une NMF d'ordre $Z_{app} = 1$ (un seul spectre pour représenter la RTF$^+$ d'une note). Pour analyser un signal d'entrée, on construit la matrice $\boldsymbol{S}$ en concaténant l'ensemble des noyaux appris, et seules les activations $\{A_{zt}\}_z$ sont estimées pour représenter à la volée les observations $\{V_{ft}\}_f$ à chaque temps $t$. Quand on procède de cette manière, la taille $Z$ du dictionnaire $\boldsymbol{S}$ ne doit plus être forcément petite devant la dimension des données comme cela devait être le cas pour les méthodes non-supervisées, puisqu'on est assuré de la significativité des atomes. Ce principe de décomposition avec atomes fixes et pré-appris est également celui de [BD11], où le modèle de décomposition est légèrement plus sophistiqué. En effet les noyaux peuvent être légèrement transformés lors de la décomposition pour qu'ils puissent s'adapter à de légères variations de hauteurs de notes.

L'avantage de ces méthodes est qu'elles peuvent être facilement réalisées en ligne puisque la redondance du signal n'est plus nécessairement utilisée lors de la décomposition. Ainsi, si le dictionnaire n'est pas trop fourni, il est possible de les faire tourner en temps réel comme dans [DCL12]. Une autre atout de taille avec l'utilisation d'un dictionnaire fixe d'atomes est que le problème de décomposition peut souvent être modélisé comme un problème d'optimisation convexe, et donc non soumis au problème de minima locaux. En revanche ces techniques possèdent l'inconvénient suivant : leurs performances dépendent fortement de la similarité entre les bases d'apprentissage et les signaux à traiter. Aussi, si elles sont bien adaptées pour des problèmes bien spécifiques (transcription de piano uniquement par exemple), elles le sont moins pour créer des algorithmes génériques, pouvant traiter n'importe quel type de signal musical.

### 1.3.4 Les méthodes semi-supervisées

Un bon compromis entre la rigidité des méthodes supervisées et le manque de contraintes des méthodes non-supervisées peut se trouver dans les méthodes dites semi-supervisées. Le principe ici est d'estimer à la fois les noyaux et la décomposition comme pour les méthodes aveugles, mais de contraindre les atomes à rester dans un certain sous-espace. Ce sous-espace, qui peut être vu comme un sous-espace de solutions possibles et significatives pour les atomes, peut être appris lors d'une étape d'apprentissage. Dans le cadre des décomposition de RTF$^+$, on peut citer [GE11] par exemple, dans lequel est introduite la notion d'*instruments propres hiérarchiques* (ici *propre* est à comprendre comme dans la notion de vecteurs *propres*). Ils représentent des cônes, chacun

permettant de modéliser une classe spécifique d'instruments. Ils sont pré-appris et pendant l'étape de décomposition, un ou plusieurs instruments propres sont utilisés pour modéliser une note présente dans la RTF$^+$. Une deuxième possibilité est de définir manuellement le sous-espace des atomes. Par exemple, les atomes peuvent être modélisés comme une combinaison linéaire de spectres harmoniques à bande étroite, pour les contraindre à pouvoir représenter uniquement des spectres harmoniques à enveloppe spectrale régulière [VBB10]. Plus simplement, dans le cas du modèle classique de NMF, on peut forcer chaque atome à avoir une énergie nulle pour les fréquences comprises entre les fréquences théoriques des partiels d'une certaine note harmonique [Vir06, ROS07b]. On s'assure ainsi du fait que chaque atome représentera bien un spectre harmonique, et donc on espère une note de musique.

Il est intéressant de noter que dans tous les articles cités dans ce paragraphe, l'hypothèse de redondance des signaux est faite. Plus spécifiquement, il est supposé que chaque instrument dans un mélange possède des similitudes d'enveloppe spectrale par note de la gamme tout au long du signal. Cette supposition, qui permet d'adapter la définition du sous-espace des atomes lors de l'étape de décomposition, est une bonne hypothèse pour de nombreux instruments. Malheureusement, ce n'est pas la cas pour tous les instruments, la voix humaine étant le meilleur contre-exemple. En effet, son enveloppe spectrale dépend bien plus des paroles que de la note chantée. Une des problématiques de la thèse sera d'étudier la pertinence ou non de supposer ce type de redondance dans les signaux musicaux.

### 1.3.5 L'ajout de contraintes douces

Indépendamment du modèle considéré, il est possible de contraindre la décomposition d'un signal d'entrée grâce à l'ajout d'aprioris ou de fonctions de pénalité sur ses paramètres (les atomes $\boldsymbol{S}$ ou leur agencement $\boldsymbol{A}$ pour la NMF par exemple). Leur utilisation peut être intéressante puisque qu'ils agissent, lors de l'algorithme de décomposition, comme une incitation pour ces paramètres à converger vers une solution plus vraisemblable, plutôt que de simplement réduire le sous-espace dans lequel ils peuvent reposer. C'est donc un moyen doux d'ajouter de l'information sur la nature des données à analyser. De nombreux aprioris ou termes de pénalité ont été proposés dans la littérature, comme la parcimonie, la décorrélation ou la continuité temporelle des activations d'atomes [Hoy04, SRS08b, DCL12, Vir07, BBV10b, ZF07]. Dans cette thèse, quatre nouveaux aprioris sur les paramètres ainsi que leur dérivation seront proposés au chapitre 3 page 41.

### 1.3.6 Modèles avec structures temporelles

Récemment, une attention particulière a été apportée à l'incorporation de structures temporelles aux modèles de RTF$^+$. En effet, les méthodes de décomposition que nous avons présentées précédemment tirent parti de la redondance du signal ou encore de l'apprentissage d'un dictionnaire, mais jamais de la dynamique spécifique aux signaux musicaux. Plusieurs modèles ont donc

été proposés pour remédier à cette lacune. Citons par exemple les travaux [OFC09, NLK+11], dans lesquels chaque note (ou chaque source) peut être représentée par plusieurs atomes différents, mais un seul à fois, et où l'on modélise la manière dont ces atomes se succèdent dans le temps. Dans [Mys10], on ajoute une couche supplémentaire : une note, ou source, peut être modélisée comme une combinaison linéaire d'atomes appartenant à un unique (parmi plusieurs) dictionnaire. Ici, c'est la manière dont les dictionnaires se succèdent dans le temps qui est modélisée. Dans cette thèse nous proposerons également un modèle permettant de considérer une structure temporelle lors de la décomposition (chapitre 5 page 69). En revanche, contrairement au références citées dans ce paragraphe, nous proposerons un modèle d'activation des notes, plutôt qu'un modèle de transition d'atomes.

## 1.4 Les outils mathématiques

Dans la section précédente, nous avons présenté de manière la plus générique possible les différentes stratégies possibles pour modéliser et décomposer des signaux, et en particulier des RTF+. Dans cette section, nous parlerons des différents cadres mathématiques utilisés pour le développement d'algorithmes de décompositions positives de RTF+.

### 1.4.1 Cadre déterministe

Un premier cadre pour développer des algorithmes de décomposition est de considérer le problème comme un problème analytique et d'employer des algorithmes d'optimisation. Pour cela, on utilise une fonction coût $\mathcal{C}$ à minimiser, qui est fondée sur une distance ou divergence $D$ entre les observations $\boldsymbol{V}$ et le modèle $\hat{\boldsymbol{V}}$, quel qu'il soit ($\hat{\boldsymbol{V}}$ devrait s'écrire $\hat{\boldsymbol{V}}(\Lambda)$ où $\Lambda$ représente l'ensemble des paramètres du modèle, mais pour des raisons de lisibilité, nous omettons de le préciser). Les plus utilisées sont probablement [Ber09] :

- la distance euclidienne au carrée (EUC), définie par :

$$\mathcal{C}(\Lambda) = D_{\text{EUC}}\left(\boldsymbol{V}|\hat{\boldsymbol{V}}\right) = \sum_{f,t}\left(V_{ft} - \hat{V}_{ft}\right)^2 ; \tag{1.3}$$

- la divergence de Kullback-Leibler (KL) [KL51], définie par :

$$\mathcal{C}(\Lambda) = D_{\text{KL}}\left(\boldsymbol{V}|\hat{\boldsymbol{V}}\right) = \sum_{f,t} V_{ft}\ln\frac{V_{ft}}{\hat{V}_{ft}} + \hat{V}_{ft} - V_{ft} ; \tag{1.4}$$

- la divergence d'Itakura-Saito (IS) [IS68], définie par :

$$\mathcal{C}(\Lambda) = D_{\text{IS}}\left(\boldsymbol{V}|\hat{\boldsymbol{V}}\right) = \sum_{f,t}\frac{V_{ft}}{\hat{V}_{ft}} - \ln\frac{V_{ft}}{\hat{V}_{ft}} - 1. \tag{1.5}$$

Si ces trois fonctions de coût sont les plus utilisées dans la littérature, il en existe de nombreuses autres, généralement définies comme des généralisations de ces divergences. Citons par exemple les $\beta$-divergences [EK01], les divergences de Bregman [HBD11a] ou encore les divergences de Csiszár [CZA06a]. Une fois une divergence choisie entre les observations $\boldsymbol{V}$ et le modèle $\hat{\boldsymbol{V}}$, décomposer une RTF$^+$ revient à estimer les paramètres $\Lambda$ de $\hat{\boldsymbol{V}}$ qui la minimisent, sous la contrainte que ces paramètres doivent rester dans leur domaine de définition (par exemple l'orthant positif pour les paramètres $\boldsymbol{S}$ et $\boldsymbol{A}$ du modèle classique de NMF, équation 1.2). Si jamais nous souhaitons ajouter une contrainte douce (*cf.* section 1.3.5) sur les paramètres, alors il faut adjoindre à la divergence le terme $\mathcal{P}(\Lambda)$ de pénalité correspondant. La fonction de coût à minimiser par rapport à $\Lambda$ devient alors

$$\mathcal{C}(\Lambda) = D\left(\boldsymbol{V}|\hat{\boldsymbol{V}}\right) + \mathcal{P}(\Lambda). \tag{1.6}$$

Généralement il n'existe pas de solution analytique, et l'on peut alors utiliser l'un des très nombreux algorithmes d'optimisation existant. Sans trop rentrer dans les détails, et sans avoir la prétention d'être exhaustif, nous pouvons mentionner certaines classes d'algorithmes qui peuvent être rencontrées :

- Les algorithmes de descente de gradient avec projection [Lin07] qui consistent en un algorithme de descente de gradient classique incluant à chaque itération une projection des paramètres dans leur domaine de définition.
- Les méthodes de Newton projetées [ZC07], où la descente de gradient est remplacée par une méthode de Newton du second ordre.
- Les méthodes d'optimisations quadratiques [ZC08, DCL12], appliquées quand la distance EUC est utilisée pour la fonction de coût.
- Les méthodes de mises à jour multiplicatives [LS99, FBD09, BBV10a]. Ce sont certainement les méthodes les plus utilisées pour l'estimation des paramètres d'un modèle si ceux-ci doivent rester positif ou nul (NMF par exemple). Leur principal avantage réside dans la facilité d'assurer la positivité des paramètres à chaque itération.
- Les algorithmes de type Majorisation-Minimisation (ou Minorisation-Maximisation) [FI11, KNS07, NLK$^+$10], qui dans le cas de la NMF classique amènent à des mises à jour multiplicatives pour les paramètres.
- Les algorithmes gloutons [BEZ08a, ONP12] dans le cas des méthodes supervisées, où à chaque itération, on s'autorise à utiliser un atome supplémentaire pour modéliser les observations, et on s'arrête quand le modèle est suffisamment proche des données.

### 1.4.2 Cadres probabilistes

Plutôt que d'envisager le problème d'estimation des paramètres d'un modèle de RTF$^+$ comme un problème analytique d'optimisation, il est possible de le considérer comme un problème d'inférence statistique. Les formulations probabilistes dans la littérature sont très nombreuses, mais reposent généralement sur un même schéma : les observations $\boldsymbol{V}$ sont le fruit d'un processus

génératif, dépendant des paramètres $\Lambda$ du modèle $\hat{\boldsymbol{V}}$ posé. Le but est alors de trouver les valeurs des paramètres qui « expliquent » au mieux les observations, et cela peut s'effectuer par exemple grâce à l'estimation du maximum de vraisemblance (MV) : on tentera de trouver $\Lambda$ tel que $P(\boldsymbol{V}|\Lambda)$ soit maximum.

Parmi les cadres probabilistes que l'on peut trouver, citons :

- les modèles de NMF probabilistes : les cas poissonien [VCG08], gaussien complexe [FBD09], de bruit additif gaussien [SL08] ou encore de bruit multiplicatif gamma [FBD09] ;
- l'analyse probabiliste en composantes latentes (PLCA) et sa version avec invariance par translation [Sha07, SRS08b, MS09] : c'est le cadre mathématique utilisé dans cette thèse ;
- les processus gaussiens [SL08] ;
- les factorisations généralisées de tenseurs couplés (GCTF) [YCS11] ;
- les modèles bayésiens non-paramétriques [NLK$^+$11].
- la NMF probabiliste couplée avec des modèles de Markov cachés [OFC09], ou encore le modèle de Markov caché factoriel et non-negatif [Mys10] permettant de modéliser des structures temporelles.

Les intérêts d'utiliser des cadres probabilistes sont multiples, et chacun d'entre eux peut présenter des avantages propres. Nous tentons ici d'établir un rapide tour d'horizon des bénéfices et caractéristiques de ces cadres probabilistes.

**Justification de la fonction de coût.** Un premier intérêt est qu'un modèle probabiliste peut permettre de justifier l'utilisation d'une certaine distance ou divergence comme fonction de coût. En effet, on se rend compte dans un certain nombre de cas qu'estimer le MV équivaut à minimiser par exemple une des trois divergences (EUC, KL ou IS) entre les observations $\boldsymbol{V}$ et le modèle $\hat{\boldsymbol{V}}$. Aussi, pour les modèles de bruit gaussien additif, le MV correspond au minimum de la distance EUC, tandis que dans les cas gaussien complexe et de bruit multiplicatif gamma, le MV correspond au minimum de la divergence IS. Le MV dans les cadres de NMF poissonienne ou de PLCA correspond quant à lui au minimum de la divergence KL. Savoir que ces divergences ont une signification dans un certain cadre probabiliste permet de les légitimer.

**Utilisation d'algorithmes génériques.** Un deuxième intérêt est que l'on peut disposer des algorithmes génériques existants pour les problèmes d'inférence. Les modèles génératifs dépendent généralement de variables cachées, et on pense donc particulièrement aux algorithmes permettant de trouver le MV en présence de telles variables : les algorithmes Espérance-Maximisation (EM) et EM généralisé (GEM) [Dem77, Sha07, OF10, Mys10], l'algorithme SAGE (*Space Alternating Generalized EM*) [FH94, FBD09], le *Fisher scoring* [JS76, YCS11] ou encore les méthodes bayésiennes variationnelles [Bea03, NLK$^+$11] sont autant d'algorithmes qui sont utilisés dans la littérature pour les décompositions de RTF.

**Incorporation d'aprioris.** Dans le cas analytique, nous avons vu que l'ajout d'une contrainte douce sur les paramètres pouvait s'effectuer via un terme de pénalité. Dans le cas probabiliste, cela peut se faire grâce à l'ajout d'un apriori $P(\Lambda)$ sur les paramètres. Les paramètres pourront alors être estimés grâce à l'estimateur du maximum *a posteriori* (MAP), qui consiste a maximiser $P(\Lambda|\boldsymbol{V})$. Là encore les algorithmes génériques d'inférence statistique présentés précédemment pourront être utilisés, ce qui rend l'ajout de contraintes douces plutôt simple.

**Caractéristiques propres.** Certains cadres ont des caractéristiques propres qui font qu'ils seront préférablement utilisés dans certains cas, ou pour certaines applications. On donne ici quelques exemples.

- Pour des applications de séparation de sources, il est plus justifié d'utiliser la NMF gaussienne complexe et les processus gaussiens car ils permettent de modéliser (directement ou indirectement) la TFCT complexe (le modèle de décomposition positive s'applique alors aux variances de chaque point temps-fréquence). Ainsi, après l'estimation des paramètres d'un modèle, il est possible d'estimer la TFCT complexe et le signal temporel correspondant de chaque source via filtrage de Wiener [OF10, LBR11].
- Pour des modèles de RTF$^+$convolutifs, c'est-à-dire quand le modèle inclue des convolutions entre atomes et activations, alors la PLCA est particulièrement bien adaptée [SRS08b] pour dériver des algorithmes d'estimation de paramètres, comme nous le comprendrons au chapitre suivant. Il est cependant également possible de trouver des algorithmes d'estimation de paramètres pour ce type de modèles dans d'autres cadre mathématiques.
- Les GCTF permettent quant à eux de gérer facilement des modèles ou plusieurs observations de natures différentes sont décomposées conjointement, en utilisant des paramètres en commun. Dans [SC12] par exemple, on joint au spectrogramme à décomposer une base de données de spectres de notes isolés, considérée aussi comme une observation. Les paramètres à estimer devront alors en même temps servir à décomposer le spectrogramme et à décrire les notes isolées. C'est une manière alternative de contraindre la décomposition.
- Enfin, le dernier exemple que nous donnons concerne les méthodes non paramétriques, qui permettent l'utilisation d'un nombre indéfini d'atomes [NLK$^+$11].

# Chapitre 2

# Outils mathématiques et représentations utilisées

Dans ce chapitre, nous exposons et étudions les outils utilisés dans le cadre de cette thèse. Nous commencerons par présenter la PLCA avec son modèle le plus classique qui est le cadre mathématique dans lequel s'inscrivent les recherches que nous avons effectuées dans cette thèse. Ensuite nous étudierons la manière dont la PLCA peut permettre d'estimer les paramètres d'un modèle convolutif de RTF$^+$. Enfin nous présenterons la transformée à Q constant, qui est le type de RTF utilisé par la suite.

## 2.1 L'analyse probabiliste en composantes latentes

Nous présentons ici le principe général de la PLCA ainsi que son modèle classique. Les calculs sont volontairement très détaillés, puisqu'ils serviront de guide à ceux de la partie III.

### 2.1.1 Le modèle

La PLCA [Sha07] est un outil probabiliste d'analyse de données positives : ici les données sont les coefficients $V_{ft}$ qui composent une RTF$^+$ $\boldsymbol{V}$ d'un signal audio. Cet outil considère $\boldsymbol{V}$ comme l'histogramme du tirage de $J$ variables $(f_j, t_j) \in [\![1, F]\!] \times [\![1, T]\!]$ indépendantes, représentant des points temps-fréquence. Ces variables sont identiquement distribuées selon une loi de probabilité discrète paramétrique $P_\Lambda(f, t)$, $\Lambda$ étant l'ensemble des paramètres. La manière dont $P_\Lambda(f, t)$ est structurée définit la façon dont les données seront décomposées : $\Lambda$ est estimé en maximisant la log-vraisemblance des observations sachant la valeur des paramètres, ou s'il existe une distribution *a priori* $Pr\,(\Lambda)$, en maximisant la probabilité *a posteriori*. Commençons par calculer la fonction de log-vraisemblance (la notation $\bar{x}$ est utilisée pour l'ensemble des tirages

de la variable $x$, soit $\{x_j\}_{j=1\cdots J}$) :

$$\begin{aligned}\mathcal{L}_\Lambda\left(\bar{f},\bar{t}\right) &= \ln\left(P_\Lambda\left(\bar{f},\bar{t}\right)\right)\\ &= \ln\left(\prod_j P_\Lambda\left(f_j,t_j\right)\right)\\ &= \sum_j \ln\left(P_\Lambda(f_j,t_j)\right.\\ &= \sum_j \sum_{f,t} \mathbb{1}_{\{(f,t)=(f_j,t_j)\}} \ln\left(P_\Lambda\left(f,t\right)\right)\\ &= \sum_{f,t} V_{ft} \ln\left(P_\Lambda\left(f,t\right)\right) \end{aligned} \tag{2.1}$$

puisqu'on considère $V_{ft}$ comme le nombre de fois que $f$ et $t$ ont été tirés. Par ailleurs, la log-probabilité *a posteriori* est définie par :

$$\ln\left(P\left(\Lambda|\bar{f},\bar{t}\right)\right) = \mathcal{L}_\Lambda\left(\bar{f},\bar{t}\right) + \ln\left(Pr\left(\Lambda\right)\right) + cst \tag{2.2}$$

où *cst* est une constante additive, indépendante de $\Lambda$. Dans le modèle de base de PLCA, une variable cachée $n$ est introduite ($n$ peut représenter une note MIDI par exemple), $f$ et $t$ sont supposés indépendants conditionnellement à $n$, et $P_\Lambda\left(f,t\right)$ est modélisée comme :

$$P_\Lambda\left(f,t\right) = \sum_n P(n)P(t|n)P(f|n) = \sum_n P(n,t)P(f|n). \tag{2.3}$$

L'ensemble des paramètres est défini par $\Lambda = \{P(n,t), P(f|n)\}_{n,t,f}$. Dans ce modèle, $P(f|n)$ représentent les différents spectres de base (que l'on peut appeler aussi atomes), et $P(n,t)$ leurs activations temporelles, à l'instar du modèle de NMF classique (*cf.* équation (1.2) p.20). Pour une meilleure compréhension du principe de la PLCA, il est possible de détailler le processus génératif de $\boldsymbol{V}$ :

- $\forall\left(f,t\right) \in [\![1,F]\!] \times [\![1,T]\!]$, on pose $V_{ft} = 0$
- Répéter $J$ fois :
  - tirer $(n,t)$ selon $P(n,t)$,
  - tirer $f$ selon $P(f|n)$,
  - poser $V_{ft} = V_{ft} + 1$.

### 2.1.2 Estimation des paramètres du modèle

Pour le moment, nous supposons qu'il n'existe pas d'apriori sur les paramètres, et par conséquent que ces derniers sont estimés grâce à l'estimateur du maximum de vraisemblance (MV), c'est-à-dire en maximisant la fonction de log-vraisemblance (2.1). Il s'avère qu'il n'existe pas de solution analytique à ce problème, et qu'il est alors nécessaire de faire appel à un algorithme d'optimisation. La présence de variables cachées nous incite à utiliser l'algorithme Espérance-

Maximisation (EM) [Dem77], qui permet de définir des règles de mise à jour pour les paramètres telles que la fonction de log-vraisemblance augmente ou reste égale à chaque itération. Commençons par calculer la log-probabilité jointe des variables cachées et observées :

$$\begin{aligned}\mathcal{L}_\Lambda\left(\bar{f},\bar{t},\bar{n}\right) &= \ln\left(P_\Lambda\left(\bar{f},\bar{t},\bar{n}\right)\right)\\ &= \sum_j \ln\left(P_\Lambda\left(f_j,t_j,n_j\right)\right)\\ &= \sum_j \ln\left(P\left(n_j,t_j\right)\right)+\ln\left(P\left(f_j|n_j\right)\right).\end{aligned} \tag{2.4}$$

**Étape du calcul de l'espérance** Lors de l'étape du calcul de l'espérance (étape E) à l'itération $l$, l'espérance conditionnelle de cette log-probabilité sachant les observations et l'estimation courante des paramètres $\Lambda^l = \left\{P^l(n,t), P^l(f|n)\right\}_{n,t,f}$ est évaluée :

$$\begin{aligned}Q_\Lambda &= \mathbb{E}\left[\mathcal{L}_\Lambda\left(\bar{f},\bar{t},\bar{n}\right)|\bar{f},\bar{t};\Lambda^l\right]\\ &= \sum_j\sum_{\bar{n}} P_{\Lambda^l}\left(\bar{n}|\bar{f},\bar{t}\right)\left[\ln\left(P\left(n_j,t_j\right)\right)+\ln\left(P\left(f_j|n_j\right)\right)\right]\\ &= \sum_j\sum_{n_j} P_{\Lambda^l}\left(n_j|f_j,t_j\right)\left[\ln\left(P\left(n_j,t_j\right)\right)+\ln\left(P\left(f_j|n_j\right)\right)\right]\\ &= \sum_{f,t}\sum_{n} V_{ft}P_{\Lambda^l}\left(n|f,t\right)\left[\ln\left(P\left(n,t\right)\right)+\ln\left(P\left(f|n\right)\right)\right].\end{aligned} \tag{2.5}$$

Dans la première étape de l'équation (2.5), on utilise le fait que l'espérance est une application linéaire ; dans la deuxième étape, que la plupart des variables cachées sont marginalisées pour un tirage $n$ donné, et que $n_j$ ne dépend que des observations $f_j$et $t_j$ ; dans la troisième étape on renomme la variable muette $n_j$ en $n$ et on somme sur les points temps-fréquence plutôt que sur les tirages, comme cela est fait dans l'équation (2.1). Les probabilités *a posteriori* des variables cachées connaissant les observations et la valeur courante des paramètres sont données par le théorème de Bayes :

$$P_{\Lambda^l}(n|f,t) = \frac{P(n,t)^l P(f|n)^l}{P_{\Lambda^l}(f,t)}, \tag{2.6}$$

$P_{\Lambda^l}(f,t)$ étant définie dans l'équation (2.3).

**Étape de maximisation** Lors de l'étape de maximisation (étape M), pour obtenir la nouvelle valeur $\Lambda^{l+1}$ des paramètres, on maximise $Q_\Lambda$ par rapport à $\Lambda$ (les probabilités *a posteriori* $P_{\Lambda^l}(n|f,t)$ sont alors fixées), sous la contrainte que toutes les distributions de probabilités restent positives et somment à un. On utilise pour cela des multiplicateurs de Lagrange $\rho = \{\rho^1, \rho_n^2\}$ pour les contraintes d'égalités (on peut vérifier facilement que les conditions de Karush-Kuhn-Tucker (KKT) [KT51] sont réunies et que le maximum global correspond au point stationnaire du Lagrangien). Nous allons voir que les résultats obtenus restent positifs, d'où l'inutilité d'in-

troduire les contraintes de positivité. Le Lagrangien est défini par :

$$L_\rho(\Lambda) = Q_\Lambda + \rho^1\left(1 - \sum_{n,t} P(n,t)\right) + \sum_n \rho_n^2\left(1 - \sum_f P(f|n)\right) \tag{2.7}$$

On annule alors le gradient de $L_\rho(\Lambda)$ pour trouver les arguments qui maximisent $Q_\Lambda$ :

$$\frac{\partial L_\rho(\Lambda)}{\partial P(n,t)} = 0 \Leftrightarrow \sum_f V_{ft} P_{\Lambda^l}(n|f,t) - \rho^1 P(n,t) = 0 \tag{2.8}$$

$$\frac{\partial L_\rho(\Lambda)}{\partial P(f|n)} = 0 \Leftrightarrow \sum_t V_{ft} P_{\Lambda^l}(n|f,t) - \rho_n^2 P(f|n) = 0. \tag{2.9}$$

En sommant sur $n$ et $t$ (resp. sur $f$) l'équation (2.8) (resp. l'équation (2.9)), on a :

$$\rho^1 = \sum_{f,t,n} V_{ft} P_{\Lambda^l}(n|f,t)$$
$$\forall n, \rho_n^2 = \sum_{f,t} V_{ft} P_{\Lambda^l}(n|f,t).$$

Ainsi donc, on obtient les règles de mise à jour pour les paramètres :

$$P(n,t)^{l+1} = \frac{\sum_f V_{ft} P_{\Lambda^l}(n|f,t)}{\sum_{f,t',n'} V_{ft'} P_{\Lambda^l}(n'|f,t')}$$
$$P(f|n)^{l+1} = \frac{\sum_t V_{ft} P_{\Lambda^l}(n|f,t)}{\sum_{f',t} V_{f't} P_{\Lambda^l}(n|f',t)},$$

ou plus simplement

$$P(n,t)^{l+1} \propto \sum_f V_{ft} P_{\Lambda^l}(n|f,t) \tag{2.10}$$

$$P(f|n)^{l+1} \propto \sum_t V_{ft} P_{\Lambda^l}(n|f,t), \tag{2.11}$$

le signe $\propto$ signifiant « proportionnel à ». Après initialisation des paramètres, l'algorithme EM consiste à itérer l'étape E (équations (2.3) et (2.6)) et l'étape M (équations (2.10) et (2.11) suivies de la normalisation des paramètres) jusqu'à convergence de la log-vraisemblance (équation (2.1)). Il est intéressant de noter qu'en fusionnant les étapes E et M de l'algorithme, on obtient des règles de mise à jour multiplicatives pour les paramètres :

$$P(n,t)^{l+1} \propto P(n,t)^l \sum_f \frac{V_{ft}}{P_{\Lambda^l}(f,t)} P(f|n)^l \tag{2.12}$$

$$P(f|n)^{l+1} \propto P(f|n)^l \sum_t \frac{V_{ft}}{P_{\Lambda^l}(f,t)} P(n,t)^l. \tag{2.13}$$

où $P_{\Lambda^l}(f,t)$ est défini dans l'équation (2.3). Comme mentionné dans [SRS08a], ces mises à jours sont en réalité les mêmes que les mises à jours multiplicatives de la NMF quand la divergence de Kullback–Leibler (équation (1.4) page 24) est utilisée comme fonction de coût. Avec cette formulation, l'algorithme EM consiste maintenant, après initialisation des paramètres, à itérer le calcul du modèle (2.3), les mises à jours (2.12) et (2.13), et la normalisation des densités de probabilités, jusqu'à convergence de la fonction de log-vraisemblance.

On peut trouver dans la littérature [SRS08a] une présentation légèrement différente du modèle de PLCA, appelée forme asymétrique :

$$P_\Lambda(f,t) = P(t)\sum_n P(n|t)P(f|n), \tag{2.14}$$

où l'ensemble des paramètres à estimer est $\Lambda = \{P(t), P(n|t), P(f|n)\}_{n,t,f}$. Dans ce cas, il est facile de prouver que

$$P(t) \propto \sum_f V_{ft} \tag{2.15}$$

et que les mises à jour pour les paramètres $P(n|t)$ et $P(f|n)$ sont données par

$$P(n|t)^{l+1} \propto P(n|t)^l \sum_f \frac{V_{ft}}{P_{\Lambda^l}(f,t)} P(t)P(f|n)^l \tag{2.16}$$

$$P(f|n)^{l+1} \propto P(f|n)^l \sum_t \frac{V_{ft}}{P_{\Lambda^l}(f,t)} P(t)P(n|t)^l. \tag{2.17}$$

Ce modèle alternatif est en réalité strictement équivalent au premier, mais nous le présentons ici puisqu'il sera mentionné dans le chapitre 5.

**Maximisation *a posteriori*** Supposons dorénavant que les paramètres sont soumis à des probabilités *a priori* $Pr(\Lambda)$. Alors l'estimateur MV est remplacé par l'estimateur du maximum *a posteriori* (estimateur MAP), c'est-à-dire que les paramètres sont estimés en maximisant l'équation (2.2). Pour cela, l'algorithme EM est adapté de telle sorte que l'étape M est remplacée par une étape MAP : ce n'est plus $Q_\Lambda$ qui est maximisé par rapport à $\Lambda$, mais $Q_\Lambda + \ln(Pr(\Lambda))$. Malheureusement, suivant le type d'apriori utilisé, il n'existe pas nécessairement de solution analytique pour la maximisation de cette fonction, comme nous le verrons dans le chapitre 3. Dans ce cas, l'utilisation d'algorithmes numériques est nécessaire pour la recherche du maximum à chaque étape MAP.

Avant de conclure cette section, on peut noter que les règles de mise à jour (2.12) et (2.13) restent inchangées si $\boldsymbol{V}$ est multiplié par un scalaire quelconque. Il n'est donc pas nécessaire de normaliser une RTF$^+$ d'entrée afin qu'elle n'ait que des coefficients entiers et qu'elle puisse effectivement être considérée comme un histogramme.

Dans un souci de concision et de clarté, sauf nécessité, nous omettons délibérément dans la suite de préciser dans les notations qu'un modèle de distribution de probabilité dépend des

paramètres $\Lambda$. Aussi, une distribution comme $P_\Lambda(f,t)$ sera tout simplement notée $P(f,t)$. De même, nous omettons de préciser que la valeur des paramètres dépend de l'itération en cours via l'exposant $l$.

## 2.2 La PLCA avec invariance par translation

Comme cela sera détaillé dans la section suivante (2.3), pour des signaux de musique, une modulation de fréquence fondamentale d'une note harmonique peut correspondre à une translation en fréquence de son spectre, pour peu qu'on utilise une RTF adaptée. Aussi, il peut être intéressant de proposer des modèles invariants par translation, et cela est possible dans le cadre de la PLCA [MS09]. Dans la PLCA avec invariance par translation (SIPLCA pour *Shift-Invariant* PLCA), chaque variable observée $f$ résulte de la somme de deux variables aléatoires ($f = \mu + i$), et le modèle d'observation d'une source $s_0 \in [\![1, S]\!]$ (la notion de source peut par exemple faire référence à un instrument particulier dans un mélange) est par conséquent défini comme la convolution de deux densités de probabilité : l'une, notée $P(\mu|s_0)_{\mu\in[\![1,F]\!]}$, que l'on appelle noyau, représente la signature spectrale de la source, et l'autre, notée $P(i,t,s_0)_{(i,t)\in[\![0,I-1]\!]\times[\![1,T]\!]}$, correspond à ses activations temps-fréquence. Le modèle SIPLCA peut alors s'écrire de la manière suivante :

$$P_\Lambda(f,t) = \sum_s \sum_i P(i,t,s)P(f-i|s), \tag{2.18}$$

où l'ensemble des paramètres du modèle est défini par $\Lambda = \{P(i,t,s), P(\mu|s)\}_{i,t,s,\mu}$. Ce modèle est illustré sur la figure 2.1, avec une unique source monophonique ($S = 1$). Le processus génératif correspondant au modèle SIPLCA est le suivant :

- $\forall\,(f,t) \in \mathbb{Z} \times [\![1,T]\!]$, on pose $V_{ft} = 0$.
- Répéter $J$ fois :
  - tirer $(i,t,s)$ selon $P(i,t,s)$,
  - tirer $\mu$ selon $P(\mu|s)$,
  - poser $f = \mu + i$,
  - poser $V_{ft} = V_{ft} + 1$.

Puisque que l'on observe $V_{ft}$ pour $f \in [\![1,F]\!]$, et non $f \in \mathbb{Z}$, on suppose simplement que $V_{ft} = 0$ pour $f \notin [\![1,F]\!]$.

Ici aussi, les paramètres sont estimés grâce à l'algorithme EM, qui permet de trouver un maximum local de la fonction de log-vraisemblance des données observées. Sans refaire les calculs en détail (ils sont en fait très similaires à ceux de la section précédente), nous pouvons donner les grandes lignes de la dérivation de l'algorithme EM. Faisons d'abord remarquer que si l'on considère $i$ (resp. $\mu$) comme une variable cachée, alors il est inutile de considérer $\mu$ (resp. $i$) également comme une variable cachée, puisque qu'elle se déduit comme suit : $\mu = f - i$ (resp. $i = f - \mu$), où $f$ est observée. Considérons d'abord que les variables $f$ et $t$ sont observées et que

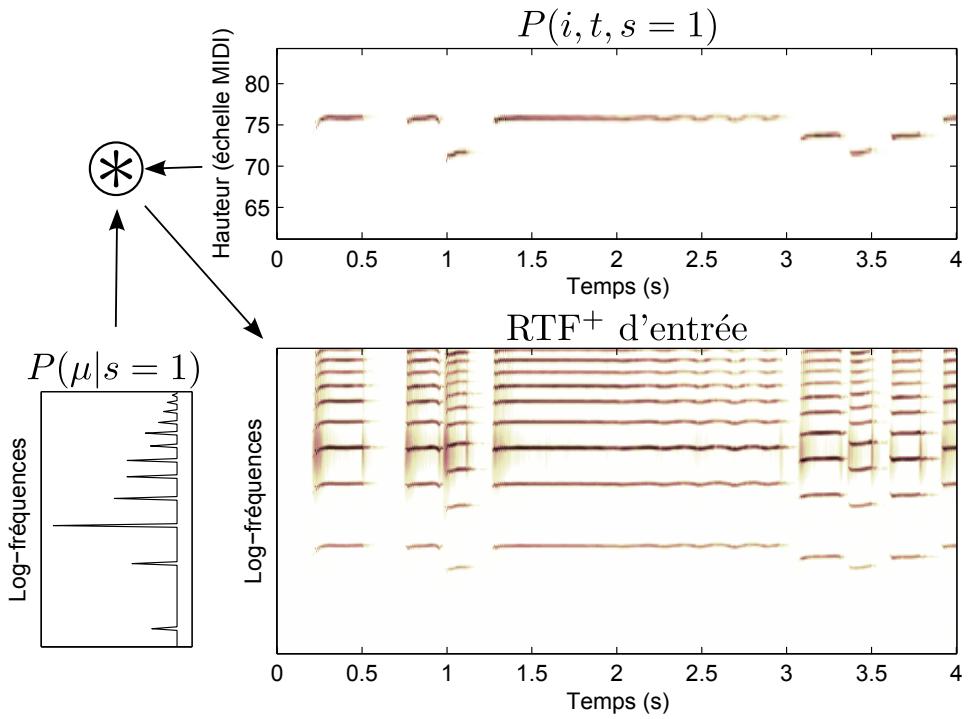



**Figure 2.1** – Illustration de la SIPLCA pour une unique source monophonique : les premières notes de *Summertime* de George Gershwin, jouées à la trompette. Le signe $\circledast$ correspond à l'opérateur de convolution.

$s$ et $i$ sont cachées. La probabilité jointe des variables cachées et observées est donnée par :

$$P(\bar{f},\bar{t},\bar{s},\bar{i}) = \prod_j P(i_j,t_j,s_j)P(f_j - i_j|s_j). \tag{2.19}$$

Alors, l'espérance conditionnelle de $\mathcal{L}_\Lambda(\bar{f},\bar{t},\bar{s},\bar{i}) = \ln\left(P(\bar{f},\bar{t},\bar{s},\bar{i})\right)$ s'exprime comme :

$$Q_\Lambda = \mathbb{E}\left[\mathcal{L}_\Lambda(\bar{f},\bar{t},\bar{s},\bar{i})|\bar{f},\bar{t};\Lambda\right] = \sum_{f,t,i,s} V_{ft}P(i,s|f,t)\left[\ln\left(P(i,t,s)\right) + \ln\left(P(f-i|s)\right)\right]. \tag{2.20}$$

Si maintenant, on considère $\mu$ plutôt que $i$ comme variable cachée, on prouve que $Q_\Lambda$ s'exprime également comme :

$$Q_\Lambda = \sum_{f,t,\mu,s} V_{ft}P(\mu,s|f,t)\left[\ln\left(P(f-\mu,t,s)\right) + \ln\left(P(\mu|s)\right)\right]. \tag{2.21}$$

Lors de l'étape M, les probabilités *a posteriori* sont calculées grâce à la règle de Bayes :

$$P(i,s|f,t) = \frac{P(i,t,s)P(f-i|s)}{P(f,t)},$$
$$P(\mu,s|f,t) = \frac{P(f-\mu,t,s)P(\mu|s)}{P(f,t)}.$$

Lors de l'étape E, on maximise $Q_\Lambda$ en fonction des paramètres (l'équation (2.20) est utilisée pour la mise à jour de $P(i,t,s)$ tandis que l'équation (2.21) est utilisée pour celle de $P(\mu|s)$), sous contrainte que les probabilités somment à un :

$$P(i,t,s) \propto \sum_f V_{ft} P(i,s|f,t), \tag{2.22}$$
$$P(\mu|s) \propto \sum_{f,t} V_{ft} P(\mu,s|f,t). \tag{2.23}$$

On peut également fusionner les étapes E et M, et proposer les règles de mises à jour suivantes :

$$P(i,t,s) \propto P(i,t,s) \sum_f \frac{V_{ft}}{P(f,t)} P(f-i|s), \tag{2.24}$$
$$P(\mu|s) \propto P(\mu|s) \sum_{f,t} \frac{V_{ft}}{P(f,t)} P(f-\mu,t,s), \tag{2.25}$$

$P(f,t)$, étant défini dans l'équation (2.18). On rappelle que les valeurs des paramètres à droite du signe $\propto$ sont fixées (valeurs à l'itération $l$). S'il permet facilement de prendre en compte d'éventuelles variations continues de fréquence fondamentale pour une note de musique donnée, le modèle SIPLCA ne considère pas que son enveloppe spectrale puisse évoluer au cours du temps. Les modèles originaux que nous présentons dans cette thèse (partie III) permettent de s'adapter à ces deux types de non-stationnarités simultanément.

## 2.3 La transformée à Q constant et ses avantages

Pour le moment, nous avons parlé de RTF, sans jamais préciser celle que nous utilisons dans le cadre de cette thèse. Une RTF qui semble bien adaptée aux signaux musicaux est la transformée à Q constant (CQT pour *Constant-Q Transform*) [Bro91, FP12, Pra11]. Cette représentation possède une échelle logarithmique des fréquences, contrairement à la transformée de Fourier à court terme (TFCT) classique. De plus, la résolution fréquentielle est inversement proportionnelle à la fréquence d'analyse. En fait, la CQT peut être considérée comme un banc de filtres dont les fréquences résonnantes sont espacées de manière logarithmique, et dont chaque filtre a un facteur de qualité $Q$ constant. Ces deux caractéristiques présentent un avantage considérable pour les signaux de musique. D'abord, l'espacement entre les partiels d'une note harmonique reste identique, quel que soit sa fréquence fondamentale. Ensuite, pour les signaux

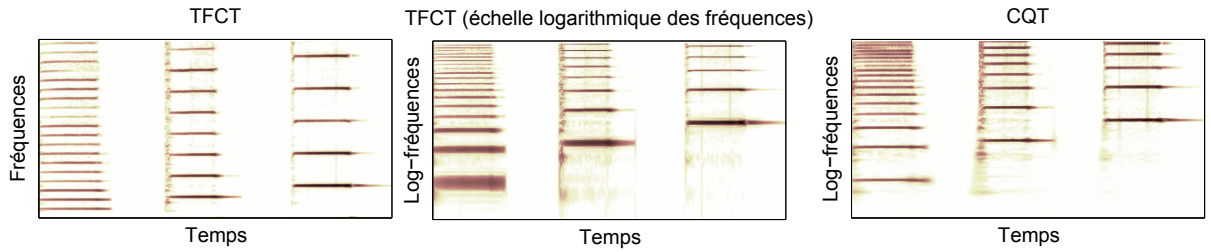



**Figure 2.2** – Trois RTF$^+$différentes : l'amplitude d'une TFCT classique, d'une TFCT avec une échelle logarithmique des fréquences et d'une CQT. Le signal d'entrée correspond à l'enregistrement de trois notes de violoncelle.

localement stationnaires, l'étalement spectral d'un partiel est défini par la valeur du facteur de qualité, et non par sa fréquence, ce qui ne serait pas le cas en utilisant une analyse de Fourier avec une échelle logarithmique. Aussi, une variation de fréquence fondamentale d'une note peut être considérée comme une translation en fréquence de ses harmoniques. Tous les modèles de RTF que nous présentons profitent de cette caractéristique. Afin d'illustrer les propriétés de la CQT, l'amplitude de trois représentations différentes de signaux audio est montrée sur la figure 2.2 : une TFCT classique, une TFCT avec une échelle logarithmique des fréquences et une CQT. Un dernier avantage de cette dernière représentation est que l'on sait l'inverser, au moins approximativement [SK10, Pra11], sinon parfaitement [DHGV11], ce qui permet de l'utiliser pour des problèmes de séparation de sources via masquage temps-fréquence [FLBR12]. Nous verrons en effet au chapitre 9 comment les modèles de RTF$^+$ que nous présentons peuvent directement s'appliquer à ce problème.

La CQT possède cependant quelques inconvénients, essentiellement dus à la taille des fenêtres d'analyse trop grandes dans les basses fréquences. Aussi, quand dans des fréquences plus aigües, le signal peut être considéré comme localement stationnaire (à l'échelle de la taille de la fenêtre d'analyse), ce n'est pas forcément le cas dans les graves. Cela se traduit par deux effets indésirables dans le bas du spectre : un étalement temporel des débuts et fins des évènements sonores, ainsi qu'un étalement fréquentiel des partiels d'une note quand celle-ci n'est pas parfaitement stationnaire. Deux autres problèmes d'ordre plus pratique avec la CQT sont d'une part son temps de calcul élevé et d'autre part la nécessité d'utiliser des pas temporels très faibles si l'on veut garder la propriété d'inversibilité : le pas temporel doit en effet rester plus petit que la fenêtre d'analyse de la plus haute fréquence, qui est généralement très courte (de l'ordre d'une ou deux millisecondes pour une fréquence maximum ne dépassant pas 16 kHz !). On se retrouve alors à manipuler de très grandes matrices, conduisant à une augmentation significative du temps calcul et de la mémoire nécessaire lors de l'exécution d'un algorithme de séparation du sources.

Dans cette thèse, nous avons utilisé l'implémentation de Jacques Prado [Pra11] pour le calcul de la CQT et de son inverse, téléchargeable gratuitement en ligne [Webb]. Sauf dans le chapitre 9 consacrée à la séparation de sources, pour tous les exemples, et tous les algorithmes que nous

proposerons, la RTF$^+$d'entrée $\boldsymbol{V}$ d'un signal temporel se calcule de la manière suivante :

- la CQT (complexe) $\boldsymbol{X}$ d'un signal monophonique est calculée avec 3 points fréquentiels par demi-ton, pour des fréquences allant de $27,5$ Hz à 7040 Hz (ce qui correspond à 8 octaves, on a donc $f \in [\![1, F]\!]$ où $F = 3 \times 8 \times 12 = 288$),
- le pas temporel utilisé est de 10 ms, ce qui signifie que la CQT d'une seconde de signal sera constituée de $T = 100$ colonnes,
- on prend enfin la racine carrée de la valeur absolue ($V_{ft} = \sqrt{|X_{ft}|}$) : prendre la racine carrée équivaut à appliquer une légère compression sur l'ensemble des coefficients et l'expérience nous a montré que les algorithmes d'analyse que nous proposerons dans ce document donnent en général de meilleurs résultats.

Par abus de langage, nous appelons également « CQT » ce type de RTF$^+$.

# Deuxième partie

# Aller plus loin avec la PLCA

# Chapitre 3

# Ajout d'aprioris

## 3.1 Introduction

Comme évoqué précédemment, un premier moyen d'introduire de l'information sur la nature des RTF$^+$ à analyser, afin d'assurer une décomposition significative, est de réduire l'espace dans lequel les paramètres d'un modèle de RTF$^+$ peuvent évoluer. Cette idée peut être facilement mise en pratique par exemple pour définir un sous-espace des solutions possibles pour les spectres de base, mais elle ne peut hélas pas répondre à tout type d'information que l'on souhaiterait ajouter au modèle. De plus, trop restreindre le sous-espace des paramètres peut avoir comme effet de multiplier le nombre de maxima locaux de la vraisemblance des observations en fonction des paramètres, et rendre ainsi l'algorithme EM inefficace. Heureusement, il existe un moyen d'introduire des contraintes douces pour intégrer de la connaissance sur la nature des signaux, et cela peut être effectué via l'introduction d'aprioris sur les paramètres dans le cas de la PLCA. Ce chapitre y est consacré.

L'ajout d'un apriori sur les paramètres, quel qu'il soit, peut permettre deux choses. La première est de rendre le problème plus identifiable. En effet, pour un modèle de RTF$^+$ donné, il peut exister des situations où les observations peuvent être modélisées de plusieurs manières différentes, avec plusieurs jeux de valeurs pour les paramètres. L'ajout d'un apriori permettra alors de choisir la solution la plus vraisemblable. La deuxième est qu'il peut empêcher l'algorithme EM de rester bloqué dans un maximum local non pertinent. Dans ce chapitre nous introduisons un certain nombre d'aprioris dans le cadre de la PLCA classique, ou la SIPLCA, mais ces aprioris sont génériques et peuvent être appliqués à n'importe quel modèle de RTF$^+$ comme nous le ferons plus tard aux modèles de la partie III. Ainsi, nous allons tenter d'utiliser des notations les plus générales possibles.

Considérons n'importe quel modèle d'observation $P(f,t)$, dépendant d'un ensemble de paramètres (représentant des distributions de probabilité) $\Lambda = \{\boldsymbol{\theta}, \Lambda'\}$, où $\boldsymbol{\theta}$ est un sous-ensemble de $\Lambda$. On peut alors remarquer que la fonction $Q_\Lambda$ à maximiser lors de l'étape M de l'algorithme EM (équation (2.5), page 31 pour le modèle PLCA par exemple) peut toujours s'exprimer sous

la forme :

$$Q_\Lambda = \sum_d w_d \ln(\theta_d) + Q_{\Lambda'} \tag{3.1}$$

où $d$ est un ensemble d'indices, $\{\theta_d\}_d$ est l'ensemble des coefficients qui composent $\boldsymbol{\theta}$ et $Q_{\Lambda'}$ ne dépend pas de $\boldsymbol{\theta}$. On note également $D$ la dimension de $\boldsymbol{\theta}$. À titre d'exemple, si $\boldsymbol{\theta}$ représente les activations temporelles $P(n,t)$ du modèle PLCA (*cf.* chapitre 2.1 page 29), alors on aura :

- $d = (n,t)$,
- $D = N \times T$,
- $\boldsymbol{\theta} = \{\theta_d\}_d = \{P(n,t)\}_{n,t}$,
- $\boldsymbol{w} = \{w_d\}_d = \left\{\sum_f V_{ft} P(n|f,t)\right\}_{n,t}$.

Si l'on souhaite appliquer un apriori $Pr(\boldsymbol{\theta})$ sur la distribution $\boldsymbol{\theta}$, l'étape M de l'algorithme EM est remplacée par une étape MAP et la mise à jour pour $\boldsymbol{\theta}$ se calcule en maximisant $Q_\Lambda + \ln(Pr(\boldsymbol{\theta}))$ par rapport à $\boldsymbol{\theta}$. Cela revient à maximiser la fonction suivante, sous la contrainte que les probabilités somment à un :

$$\begin{aligned} \mathcal{M} \ : \ \Omega = ]0,1[^D &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_d w_d \ln(\theta_d) + \ln(Pr(\boldsymbol{\theta})) \,. \end{aligned} \tag{3.2}$$

Nous proposons maintenant quatre différents types d'aprioris, pouvant être utilisés suivant les hypothèses faites sur les données d'entrée : apriori de parcimonie [FBR12a, FBR13], apriori de continuité temporelle [FBR13], apriori de ressemblance (il n'a pas encore fait l'objet d'une publication), et enfin apriori de monomodalité [FBR11a, FBR13, FBR11b].

## 3.2 Aprioris de parcimonie

Dans la littérature, plusieurs solutions ont été suggérées pour renforcer la parcimonie dans le cadre de la PLCA. Dans [GE10] par exemple, un exposant supérieur à 1 est appliqué aux distributions que l'on veut rendre plus parcimonieuses, à chaque itération de l'algorithme EM, juste avant l'étape de normalisation. Il s'agit en effet d'un moyen facile d'accroitre la parcimonie, mais qui ne repose sur aucun résultat théorique, et rien n'assure dans la pratique que la fonction de log-vraisemblance augmente au fil des itérations. Dans [Sha07], une solution plus théorique est proposée, avec l'ajout d'un apriori basé sur la valeur d'entropie des distributions. Cependant, il nous semble que la dérivation de l'algorithme EM avec un tel apriori n'ait pas été complètement résolue dans la littérature : l'étape MAP revient à maximiser une fonction qui n'est plus concave, et qui peut donc présenter plusieurs points stationnaires. En pratique, on remarque que l'algorithme du point fixe proposé ne converge pas nécessairement vers le maximum global de la fonction $Q_\Lambda$. Nous proposons dans cette section une solution viable pour l'utilisation de cet apriori. Nous proposons également deux nouveaux aprioris supplémentaires, l'un fondé sur la norme $l_{1/2}$ et l'autre sur la norme $l_2$. Dans cette section, pour des raisons de simplicité de

notation, $\boldsymbol{\theta}$ est considéré comme un vecteur unidimensionnel de coefficients $\theta_d$, avec $d \in [\![1, D]\!]$, plutôt que comme un tenseur multidimensionnel.

**Parcimonie et norme $l_{1/2}$.** Le premier apriori que nous proposons est le suivant :

$$Pr(\boldsymbol{\theta}) \propto \exp\left(-2\beta_{\text{parci}}\|\boldsymbol{\theta}\|_{1/2}\right) \tag{3.3}$$

où $\|\boldsymbol{\theta}\|_{1/2} = \sum_d \sqrt{\theta_d}$ et $\beta_{\text{parci}} > 0$ est un hyperparamètre positif indiquant la force de l'apriori. Il s'agit d'une fonction croissante d'un des critères de parcimonie étudié dans [HR09]. Avec cet apriori, la fonction (3.2), à maximiser sous la contrainte que $\sum_d \theta_d = 1$, devient alors :

$$\begin{aligned} \mathcal{M} \ : \ \Omega = ]0,1[^D &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_d w_d \ln(\theta_d) - 2\beta_{\text{parci}} \sum_d \sqrt{\theta_d}. \end{aligned} \tag{3.4}$$

Dans l'annexe B page 165, on montre que l'on peut trouver une condition sur la (faible) valeur de $\beta_{\text{parci}}$ pour laquelle on sait trouver l'argument $\hat{\boldsymbol{\theta}}$ du maximum global, et une autre, moins forte, pour laquelle on sait trouver l'argument d'un maximum local. Dans la pratique, une des deux conditions est toujours respectée et $\hat{\boldsymbol{\theta}}$ est alors donné par :

$$\hat{\theta}_d = h_d^+(\rho) = \frac{2w_d^2}{\beta_{\text{parci}} + 2\rho w_d + \beta_{\text{parci}}\sqrt{\beta_{\text{parci}}^2 + 4\rho w_d}}, \tag{3.5}$$

où $\rho$ est l'unique réel supérieur à $\rho_{min} = -\frac{\beta_{\text{parci}}^2}{4\max_d w_d}$ tel que $\sum_d h_d^+(\rho) = 1$. Il peut être trouvé grâce à n'importe quel algorithme numérique de recherche de racine. Dans le cas où $\hat{\boldsymbol{\theta}}$ n'est que l'argument d'un maximum local, il faut alors vérifier qu'il fait bien accroitre la valeur de $\mathcal{M}$ par rapport à l'ancienne valeur de ce paramètre (toujours le cas dans la pratique), pour pouvoir invoquer l'algorithme EM généralisé (GEM) [Dem77] et s'assurer de la convergence de l'algorithme EM.

**Parcimonie et entropie.** L'apriori entropique est défini comme une fonction décroissante de l'entropie de Shannon :

$$Pr(\boldsymbol{\theta}) \propto \exp\left(\beta_{\text{parci}} \sum_d \theta_d \ln(\theta_d)\right), \tag{3.6}$$

$\beta_{\text{parci}} > 0$ définissant la force de l'apriori. Il a été à notre connaissance pour la première fois introduit dans [Bra99], dans le cadre de l'estimation des paramètres d'une distribution multinomiale, puis réutilisé dans le cadre de la PLCA ou de la SI-PLCA dans [Sha07, SRS08b]. La fonction $\mathcal{M}$ à maximiser avec un tel apriori, sous la contrainte que $\sum_d \theta_d = 1$, est alors la

suivante :

$$\begin{aligned}\mathcal{M} \ : \ \Omega =]0,1[^D &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_d w_d \ln(\theta_d) + \beta_{\text{parci}} \sum_d \theta_d \ln(\theta_d)\,.\end{aligned} \tag{3.7}$$

Dans la littérature, la recherche du maximum s'effectue uniquement via la recherche d'un point stationnaire du Lagrangien correspondant. Or, puisque $\mathcal{M}$ n'est pas une fonction concave, rien n'assure que l'algorithme du point fixe proposé dans les articles sus-mentionnés converge. Et s'il converge ce n'est pas nécessairement vers un maximum local. Dans l'annexe B page 165, on montre comme pour l'apriori précédent, que l'on peut trouver deux conditions sur la valeur de $\beta_{\text{parci}}$ (une des deux étant toujours vérifiée dans la pratique) pour lesquelles on sait calculer l'argument $\hat{\boldsymbol{\theta}}$ soit du maximum global, soit d'un maximum local :

$$\hat{\theta}_d = \mathrm{h}_d^{-1}(\rho) = \frac{w_d/\beta_{\text{parci}}}{-\mathcal{W}_{-1}\left(-\frac{w_d}{\beta_{\text{parci}}}\exp\left(1-\rho/\beta_{\text{parci}}\right)\right)}, \tag{3.8}$$

où $\mathcal{W}_{-1}$ est la branche $-1$ de la fonction multivaluée de Lambert [CGH$^+$96] et où $\rho$ est l'unique réel supérieur à $\rho_{min} = \beta\left(\ln\left(\max_d(w_d)/\beta\right)+2\right)$ tel que $\sum_d h_d^{-1}(\rho) = 1$. Il peut être trouvé grâce à n'importe quel algorithme numérique de recherche de racine.

**Parcimonie et norme $l_2$.** Pour finir, le dernier apriori de parcimonie que l'on propose, basé sur la norme $l_2$ est le suivant :

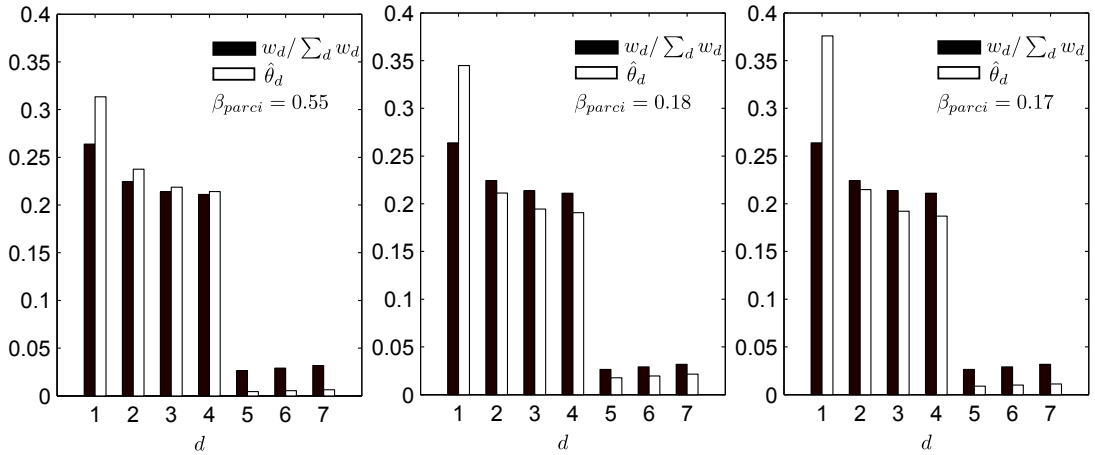
$$Pr(\boldsymbol{\theta}) \propto \exp\left(\frac{\beta_{\text{parci}}}{2}\sum_d \theta_d^2\right). \tag{3.9}$$

Ici encore, il s'agit d'une fonction croissante d'un des critères de parcimonie étudié dans [HR09]. Pour résoudre l'étape MAP avec cet apriori, on arrive aussi à trouver (*cf.* annexe B page 165) des conditions pour lesquelles on sait calculer l'argument $\hat{\boldsymbol{\theta}}$ du maximum global ou d'un maximum local :

$$\hat{\theta}_d = \mathrm{h}_d^{-}(\rho) = \frac{\rho - \sqrt{\rho^2 - 4\beta w_d}}{2\beta} = \frac{2w_d}{\rho + \sqrt{\rho^2 - 4\beta w_d}}, \tag{3.10}$$

où $\rho$ est l'unique réel supérieur à $\rho_{min} = 2\sqrt{\beta \max_d(w_d)}$ tel que $\sum_d h_d^{-}(\rho) = 1$ pouvant être estimé grâce à un algorithme numérique de recherche de racine.

**Étude empirique des différents aprioris.** Avant d'appliquer l'un de ces aprioris à la PLCA, il est intéressant de les observer expérimentalement et de comparer leur comportement. Pour cela, nous avons créé un vecteur $\boldsymbol{w} = (w_1, \cdots, w_D)$, et l'avons utilisé comme entrée aux trois aprioris décrits dans cette section. Pour chacun des aprioris, $\beta_{\text{parci}}$ a été fixé arbitrairement pour que les sorties aient à vue d'œil les mêmes niveaux de parcimonie. Dans la figure 3.1, nous avons illustré les différentes sorties (en blanc) ainsi que la sortie si $\beta_{\text{parci}}$ était égal à 0 (le vecteur $\boldsymbol{w}$ normalisé, en noir). D'abord, on remarque bien que les aprioris ont renforcé la parcimonie




(a) Parcimonie et norme $l_{1/2}$ (b) Parcimonie et entropie (c) Parcimonie et norme $l_2$

**Figure 3.1** – Étude expérimentale du comportement des différents aprioris de parcimonie.

du vecteur d'entrée, agissant comme une fonction de contraste sur les coefficients de $\boldsymbol{w}$. En regardant de plus près, on observe aussi que suivant le type d'apriori utilisé, ce ne sont pas les mêmes coefficients qui ont été augmentés ou diminués. En fait l'apriori fondé sur la norme $l_{1/2}$ a rabaissé les trois plus petits coefficients ($d = 5, 6$ et 7) et augmenté les 4 autres, alors que les deux autres aprioris ont augmenté le plus grand coefficient ($d = 1$) au détriment de tous les autres. Ainsi selon le cas, soit l'on encourage les plus petites valeurs à se rapprocher de 0, soit l'on encourage la (ou les) plus grande(s) valeur(s) à se rapprocher de 1. Dans cette thèse, nous souhaitons utiliser l'apriori de parcimonie sur les activations temporelles des notes ($P(n,t)$ par exemple pour le modèle classique de PLCA) et il nous semble plus judicieux d'utiliser celui fondé sur la norme $l_{1/2}$ : il permet d'écraser les faibles valeurs vers zéro, tout en garantissant une certaine conservation des rapports relatifs entre les grandes valeurs. Dorénavant, ce sera cet apriori que nous désignerons lorsque nous mentionnerons l'apriori de parcimonie.

**Indépendance à la taille des données.** Dans la pratique nous souhaitons pouvoir fixer la force de l'apriori de parcimonie, quelle que soit la taille des données observées. Pour cela, nous avons considéré l'idée qu'en dupliquant les données à analyser, les résultats d'une décomposition restent les mêmes (c'est-à-dire qu'ils soient juste, eux aussi, dupliqués), sans avoir à changer la valeur de $\beta_{\text{parci}}$. En étudiant ce problème, on peut prouver facilement que la valeur de $\beta_{\text{parci}}$ doit être proportionnelle à $\sqrt{D}$ ($D$ étant est le nombre de coefficients qui composent la distribution $\boldsymbol{\theta}$ soumise à l'apriori). C'est alors le coefficient de proportionnalité qui est fixé. Dorénavant, $\beta_{\text{parci}}$ se réfèrera à ce coefficient (autrement dit, nous redéfinissons l'apriori de parcimonie (3.3) comme $Pr(\boldsymbol{\theta}) \propto \exp\left(-2\sqrt{D}\beta_{\text{parci}}\|\boldsymbol{\theta}\|_{1/2}\right)$). En guise de synthèse, la mise à jour des paramètres avec l'apriori de parcimonie est résumée dans l'Algorithme 1 : nous y avons également inclus la condition sur la valeur de $\beta_{\text{parci}}$ qui assure que $\hat{\boldsymbol{\theta}}$ est l'argument du maximum global (*cf.* annexe B page 165).

**Algorithme 1:** $\hat{\boldsymbol{\theta}} = \text{Parci}\,(\boldsymbol{w}, \beta_{\text{parci}})$ : mise à jour des paramètres avec l'apriori de parcimonie.

· vérifier que $\sum_d h_d^+(\rho_0) > 1$ (toujours vrai dans la pratique), avec :

$$h_d^+(\rho) = \frac{2w_d^2}{D\beta_{\text{parci}}^2 + 2\rho w_d + \sqrt{D}\beta_{\text{parci}}\sqrt{D\beta_{\text{parci}}^2 + 4\rho w_d}},$$
$$\rho_0 = \max_d(w_d) - \beta_{\text{parci}},$$

· trouver l'unique $\rho > \rho_0$ tel que

$$\sum_d h_d^+(\rho) = 1$$

grâce à un algorithme de recherche de racine (dans la pratique, nous utilisons la fonction Matlab *fzero.m*);
· pour chaque indice $d$, poser :

$$\hat{\theta}_d = h_d^+(\rho).$$

**Détail d'implémentation.** L'apriori de parcimonie dépend de l'hyperparamètre $\beta_{\text{parci}}$ qui définit sa force, et qui doit être fixé manuellement. Après expérimentation, nous nous rendons compte que si sa valeur est trop grande lors des premières itérations, l'algorithme a tendance à converger trop rapidement vers un maximum local. Ainsi nous préconisons une stratégie en deux étapes pour l'algorithme EM : une première étape de « post-initialisation » où $\beta_{\text{parci}}$ augmente de 0 vers sa valeur finale sur quelques itérations (typiquement quelques dizaines) et une deuxième étape classique où $\beta_{\text{parci}}$ reste constant jusqu'à convergence de l'algorithme.

**Application à la PLCA.** A titre d'exemple et d'illustration, nous souhaitons utiliser l'apriori de parcimonie sur les activations temporelles du modèle PLCA classique (section 2.1 page 29), où chaque colonne d'une CQT est modélisée comme une somme pondérée de spectres de base, représentant dans notre cas des spectres de notes de musique. Les activations temporelles $P(n,t)$ représentent alors l'énergie de chaque note au cours du temps, et appliquer un apriori de parcimonie dessus revient à supposer qu'il est préférable d'expliquer une observation avec le moins de notes possibles. Afin d'aider l'algorithme à converger vers une solution significative, on personnalise la PLCA de la manière suivante :

- —∘ les 88 premiers spectres de base sont réservés pour les spectres harmoniques et ils représentent chacun une note sur l'échelle MIDI,
- —∘ pour forcer le spectre $P(f|n)$ à rester harmonique (avec $n \in [\![1, 88]\!]$), on l'initialise avec des valeurs nulles pour les points fréquentiels éloignés de plus ou moins un quart de ton des harmoniques théoriques de la note $n$, à l'instar du modèle de NMF présenté dans [ROS07b] : puisque les mises à jours sont multiplicatives, ces valeurs restent nulles au fil des itérations,

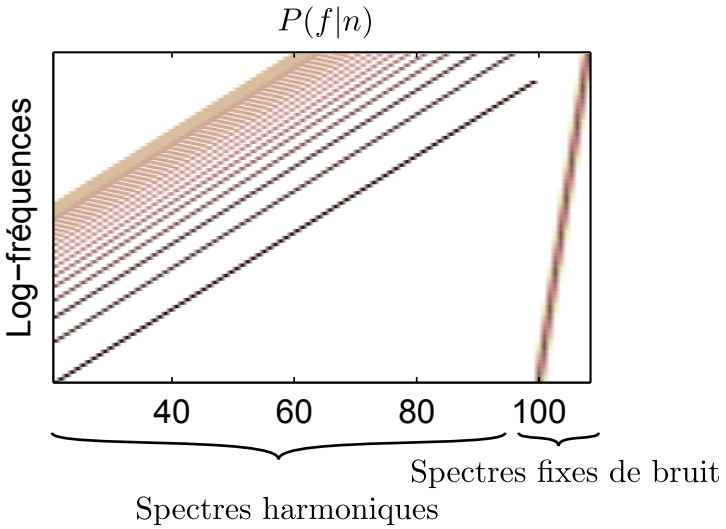


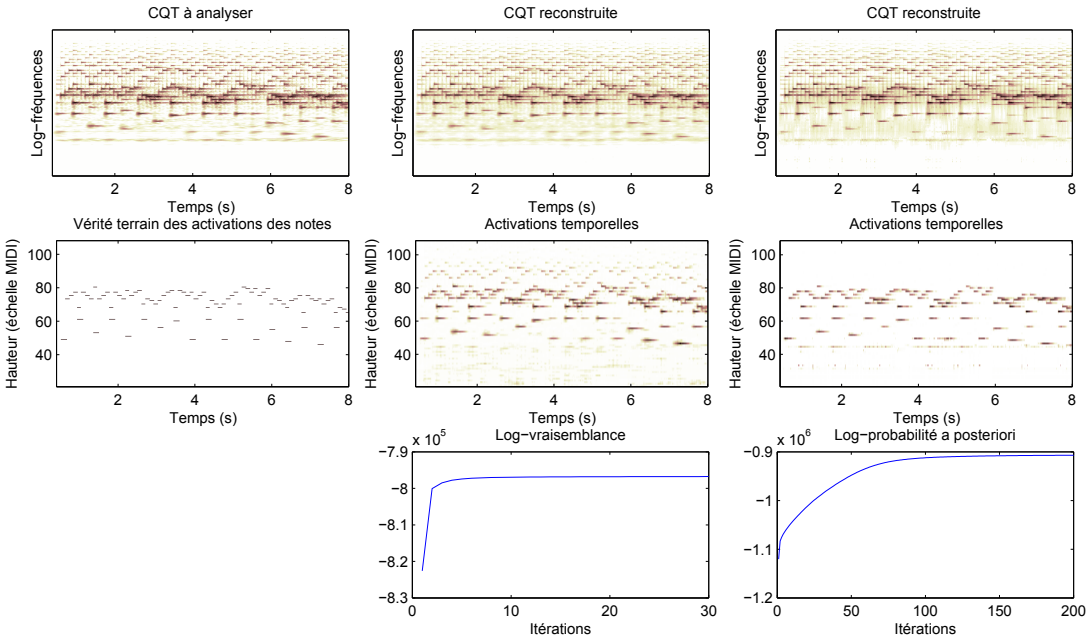

**Figure 3.2** – Initialisation des spectres de base en vu de l'illustration de l'apriori de parcimonie dans le cas de la PLCA.

- 10 spectres de base supplémentaires, ayant la forme de fenêtres à bandes étroites, sont utilisés pour modéliser le bruit : ils ne sont pas mis à jour durant l'algorithme, et seuls leurs activations sont estimées,
- la distribution $P(n,t)$ représentant les activations temporelles est initialisée de manière uniforme.

L'initialisation des spectres de base peut être visualisée sur la figure 3.2. Sur la figure 3.3, on a illustré l'effet de l'utilisation de l'apriori de parcimonie. Le signal d'entrée correspond à morceau de piano : nous prenons cet instrument car l'hypothèse selon laquelle le spectre d'une note donnée reste identique à tout instant, hypothèse implicite dans le modèle PLCA, est assez raisonnable avec le piano (même si elle n'est pas juste). Cela ne serait pas le cas avec des instruments n'ayant pas une fréquence fondamentale fixe par note (le violon et son vibrato par exemple).

## 3.3 Apriori de continuité temporelle

Dans le modèle de PLCA, si l'on décide d'effectuer une permutation des colonnes d'une $\text{RTF}^+$ d'observation $V_{ft}$, cela ne change strictement rien à la décomposition obtenue (pour peu qu'on effectue la permutation inverse des colonnes de $P(n,t)$ après l'algorithme). En fait, cela vient de la supposition que chaque tirage du processus génératif est indépendant, et *a fortiori*, que les observations au temps $t$ sont indépendantes des observations au temps $t+1$. Or, cette hypothèse peut être remise en question pour deux raisons. D'abord, la musique est généralement produite par des instruments régis par les lois de la physique, ainsi peu ou pas de discontinuités dans les paramètres acoustiques sont observées : on a donc toutes les raisons de penser que les observations au temps $t+1$ vont ressembler à celles au temps $t$. La deuxième raison, plus pragmatique, est que lorsque l'on calcule une RTF (CQT ou STFT par exemple),




(a) Sans apriori (b) Avec apriori de parcimonie

**Figure 3.3** – Illustration de l'utilisation de l'apriori de parcimonie. Le signal d'entrée, d'une durée de 8 secondes, est un extrait du Prélude en Ré majeur BWV 850 de J.S. Bach (piano). Les activations temporelles correspondent aux énergies des spectres harmoniques ($P(n,t)$ pour $n \in [\![1, 88]\!]$). On voit bien que si la CQT reconstruite $P(f,t)$ reste quasiment inchangée, les activations temporelles deviennent plus parcimonieuses avec l'apriori.

les fenêtres d'analyses se chevauchent généralement d'une trame à l'autre et qu'il est donc impossible que deux colonnes contigües soient indépendantes. L'hypothèse que les observations suivent une certaine continuité temporelle est donc une information supplémentaire sur la nature des données, qui pourrait être utile pour la décomposition. Par exemple, pour le modèle classique de PLCA, cela peut se traduire en contraignant chaque colonne de $P(n,t)$ à évoluer lentement au cours du temps.

De nombreuses solutions ont été suggérées pour renforcer la continuité temporelle des paramètres, dans le cadre de la PLCA comme dans celui de la NMF. On peut mentionner par exemple [Vir07, BBV10b], où une contrainte de régularité est imposée via un terme de pénalité dans la fonction de coût de la NMF, ou via un apriori sur les paramètres dans le cadre de la NMF Bayésienne. Dans [MS09], la continuité temporelle est imposée grâce à l'application d'un filtre de Kalman régularisant sur les activations temps-fréquence entre deux itérations de l'algorithme, mais avec cette solution, on sort un peu du cadre Bayésien de l'algorithme EM.

Nous proposons ici un nouvel apriori fondé sur le rapport des moyennes géométrique et arithmétique des paramètres que l'on souhaite voir évoluer lentement au cours du temps. Soit $\boldsymbol{\theta}$ une matrice $N \times T$ représentant une distribution bidimensionnelle (par exemple les activations $P(n,t)$ du modèle classique de PLCA), dont la deuxième dimension est la dimension temporelle. Pour une meilleure intelligibilité des notations, ses coefficients $\theta_d = \theta_{nt}$ sont notés $\theta_n^t$ (de même

que pour les coefficients $w_d$ correspondants, que l'on note $w_n^t$). L'apriori de continuité temporelle que nous introduisons est défini comme :

$$Pr(\boldsymbol{\theta}) \propto \left( \prod_n \prod_{t=2}^{T} 2\frac{\sqrt{\theta_n^t \theta_n^{t-1}}}{\theta_n^t + \theta_n^{t-1}} \right)^{\beta_{\text{temp}}} \tag{3.11}$$

où $\beta_{\text{temp}}$ est un hyperparamètre positif définissant la force de l'apriori. Un tel apriori favorise en effet une lente évolution des coefficients de chaque colonne de $\boldsymbol{\theta}$ puisque deux nombres sont d'autant plus proches que le rapport entre leurs moyennes géométrique et arithmétique est grand. La fonction 3.2, à maximiser sous la contrainte que $\sum_{n,1} \theta_n^t = 1$, devient alors :

$$\begin{aligned} \mathcal{M} \ : \ \Omega = ]0,1[^{N\times T} &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_n \sum_{t=1}^{T} w_n^t \ln\left(\theta_n^t\right) + \beta_{\text{temp}} \sum_n \sum_{t=2}^{T} \ln\left( 2\frac{\sqrt{\theta_n^t \theta_n^{t-1}}}{\theta_n^t + \theta_n^{t-1}} \right). \end{aligned} \tag{3.12}$$

Soit $\hat{\boldsymbol{\theta}}$ l'argument qui maximise $\mathcal{M}$ tout en vérifiant la contrainte $\sum_{n,t} \hat{\theta}_n^t = 1$. On sait que $\hat{\boldsymbol{\theta}}$ existe puisque $\mathcal{M}$ est majorée par 0 sur $\Omega$. De plus, il vérifie les conditions de KKT : il existe un unique $\rho \in \mathbb{R}$ tel que

$$\forall n,\ \hat{\theta}_n^t = \begin{cases} \frac{w_n^t + \beta_{\text{temp}}}{\rho + \frac{\beta_{\text{temp}}}{2\hat{\theta}_n^t} + \frac{\beta_{\text{temp}}}{\hat{\theta}_n^{t+1} + \hat{\theta}_n^t}} & \text{si } t = 1 \\ \frac{w_n^t + \beta_{\text{temp}}}{\rho + \frac{\beta_{\text{temp}}}{\hat{\theta}_n^{t-1} + \hat{\theta}_n^t} + \frac{\beta_{\text{temp}}}{\hat{\theta}_n^{t+1} + \hat{\theta}_n^t}} & \text{si } t \in [\![2, T-1]\!] \\ \frac{w_z^t + \beta_{\text{temp}}}{\rho + \frac{\beta_{\text{temp}}}{\hat{\theta}_n^{t-1} + \hat{\theta}_n^t} + \frac{\beta_{\text{temp}}}{2\hat{\theta}_n^t}} & \text{si } t = T \end{cases} . \tag{3.13}$$

Malheureusement, il n'existe pas de solution analytique à cette équation, et l'on n'est pas sûr non plus qu'il existe une unique solution. Cependant, les simulations numériques montrent que l'algorithme du point fixe proposé dans l'Algorithme 2, linéaire en la taille des données, converge toujours vers une solution qui fait accroitre la valeur de $\mathcal{M}$. L'algorithme GEM permet alors d'assurer la convergence du critère (la probabilité *a posteriori* des paramètres sachant les observations).

Dans cette section, jusqu'à présent, $\boldsymbol{\theta}$ représentait une distribution bidimensionnelle (par exemple les activations $P(n,t)$ du modèle PLCA), c'est-à-dire qu'on avait $\sum_{n,t} \theta_n^t = 1$. Or, par la suite, on voudra éventuellement appliquer l'apriori de continuité à un ensemble de distributions unidimensionnelles : par exemple les activations $P(n|t)$ du modèle PLCA alternatif (modèle 2.14 page 33). Dans ce cas, la contrainte de normalisation lors de l'étape de maximisation est remplacée par $\forall t, \quad \sum_n \theta_n^t = 1$, et l'Algorithme 2 a besoin d'être légèrement modifié, comme décrit dans l'Algorithme 3.

**Algorithme 2:** $\hat{\boldsymbol{\theta}} = \mathrm{Temp}\,(\boldsymbol{w}, \beta_{\mathrm{temp}})$ : méthode du point fixe pour l'apriori de continuité temporelle (contrainte : $\sum_{n,t} \hat{\theta}_n^t = 1$).

$\forall\,(n,t) \in [\![1,N]\!] \times [\![1,T]\!], \hat{\theta}_n^t \leftarrow \frac{w_n^t}{\sum_{\tilde{n},\tilde{t}} w_{\tilde{n}}^{\tilde{t}}}$ ;

**répéter**

- $\forall n \in [\![1,N]\!],\ s_n^1 \leftarrow \beta_{\mathrm{temp}}/(2\hat{\theta}_n^1)$;
- $\forall (n,t) \in [\![1,N]\!] \times [\![2,T]\!],\ s_n^t \leftarrow \beta_{\mathrm{temp}}/(\hat{\theta}_n^{t-1} + \hat{\theta}_n^t)$;
- $\forall n \in [\![1,N]\!],\ s_n^{T+1} \leftarrow \beta_{\mathrm{temp}}/(2\hat{\theta}_n^T)$;
- trouver l'unique $\rho$ tel que $\sum_{n,t} \frac{w_n^t+\beta_{\mathrm{temp}}}{\rho+s_n^t+s_n^{t+1}} = 1$ et $\forall n,t,\ \frac{w_n^t+\beta_{\mathrm{temp}}}{\rho+s_n^t+s_n^{t+1}} \geq 0$ (il est possible d'utiliser par exemple la méthode de Laguerre [Mek01]);
- $\forall (n,t) \in [\![1,N]\!] \times [\![1,T]\!],\ \hat{\theta}_n^t \leftarrow \frac{w_n^t+\beta_{\mathrm{temp}}}{\rho+s_n^t+s_n^{t+1}}$;

**jusqu'à** *convergence*;

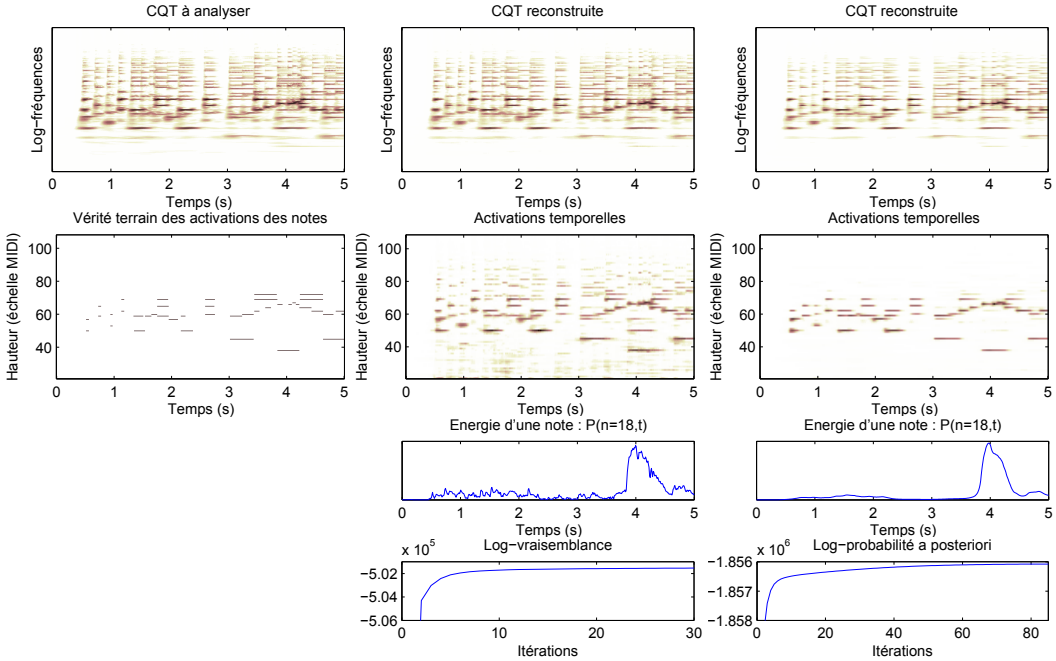
**Algorithme 3:** $\hat{\boldsymbol{\theta}} = \mathrm{Temp2}\,(\boldsymbol{w}, \beta_{\mathrm{temp}})$ : méthode du point fixe pour l'apriori de continuité temporelle (contraintes : $\forall t,\ \sum_n \hat{\theta}_n^t = 1$).

$\forall\,(n,t) \in [\![1,N]\!] \times [\![1,T]\!], \hat{\theta}_n^t \leftarrow \frac{w_n^t}{\sum_{\tilde{n}} w_{\tilde{n}}^t}$ ;

**répéter**

- $\forall n \in [\![1,N]\!],\ s_n^1 \leftarrow \beta_{\mathrm{temp}}/(2\hat{\theta}_n^1)$;
- $\forall (n,t) \in [\![1,N]\!] \times [\![2,T]\!],\ s_n^t \leftarrow \beta_{\mathrm{temp}}/(\hat{\theta}_n^{t-1} + \hat{\theta}_n^t)$;
- $\forall n \in [\![1,N]\!],\ s_n^{T+1} \leftarrow \beta_{\mathrm{temp}}/(2\hat{\theta}_n^T)$;
- $\forall t \in [\![1,T]\!]$, trouver l'unique $\rho^t$ tel que $\sum_n \frac{w_n^t+\beta_{\mathrm{temp}}}{\rho^t+s_n^t+s_n^{t+1}} = 1$ et $\forall n,\ \frac{w_n^t+\beta_{\mathrm{temp}}}{\rho^t+s_n^t+s_n^{t+1}} \geq 0$ (il est possible d'utiliser par exemple la méthode de Laguerre [Mek01]);
- $\forall (n,t) \in [\![1,N]\!] \times [\![1,T]\!],\ \hat{\theta}_n^t \leftarrow \frac{w_n^t+\beta_{\mathrm{temp}}}{\rho^t+s_n^t+s_n^{t+1}}$;

**jusqu'à** *convergence*;




(a) Sans apriori (b) Avec apriori de continuité temporelle

**Figure 3.4** – Illustration de l'utilisation de l'apriori de continuité temporelle sur les activations temporelles du modèle classique de PLCA. Le signal d'entrée, d'une durée de 5 secondes, est extrait de la Suite bergamasque (Menuet) de C. Debussy (piano). Les activations temporelles correspondent aux énergies des spectres harmoniques ($P(n,t)$ pour $n \in [\![1, 88]\!]$).

**Détail d'implémentation.** Contrairement à l'apriori de parcimonie, l'expérience montre qu'il est inutile de passer par une phase où $\beta_{\text{temp}}$ augmente à chaque itération jusqu'à sa valeur finale. Dans la pratique, $\beta_{\text{temp}}$ peut donc être fixé dès la première itération de l'algorithme EM.

**Application à la PLCA.** Pour illustrer cet apriori, on reprend exactement le même cadre expérimental que pour l'apriori de parcimonie (page 46), en remplaçant ce dernier apriori par l'apriori de continuité temporelle. La figure 3.4 illustre son utilisation : on observe bien une plus lente évolution de l'énergie d'une note donnée au cours du temps. Il est très intéressant de constater dans cet exemple que l'apriori de continuité temporelle a également amplifié le caractère parcimonieux des activations temporelles. Cela peut être dû au fait qu'au départ, comme l'énergie du signal est nulle, aucune note n'est active. Ainsi, renforcer la continuité temporelle a permis de conserver au mieux cet état d'inactivité : le moins de notes possibles s'activent donc au fil du temps.

## 3.4 Apriori de ressemblance

Dans l'interprétation spécifique du modèle classique de PLCA que nous faisons depuis le début de ce mémoire, chaque spectre de base représente le spectre d'une certaine note de la gamme : on pose la contrainte qu'une note, quand elle est présente, possède systématiquement

le même spectre (à une constante près). C'est de cette manière que l'on suppose qu'un signal de musique est intrinsèquement redondant, et la PLCA permet de tirer profit de cette redondance supposée pour décomposer intelligemment les données observées. Cependant, cette hypothèse de redondance parfaite des spectres pour une note donnée est une hypothèse très forte, qui s'avère fausse pour la plupart des signaux musicaux. Une note de piano par exemple verra ses partiels les plus aigus décroitre plus rapidement que les plus graves, et l'on peut donc considérer que la forme de son spectre dépend du temps. Ou encore si l'on considère deux instruments différents jouant une même note à des moments différents, les spectres composant cette note ne seront certainement pas identiques à tout instant.

On peut souhaiter relâcher cette contrainte de redondance parfaite, et s'autoriser à utiliser plusieurs atomes pour modéliser le spectre d'une note à un instant donné. En jouant sur les coefficients de pondération de ces atomes, on pourra alors modéliser tout un ensemble de spectres de notes. Une manière très simple de mettre cette idée en application est de multiplier l'ordre $N$ du modèle PLCA (équation (2.3) p.30) par $Z$ et de considérer $Z$ colonnes adjacentes de $P(f|n)$ comme des atomes de base pouvant représenter une certaine note. Seulement si l'on n'ajoute aucune contrainte, rien ne permet d'assurer que cela sera le cas lors de la décomposition d'une CQT d'entrée, et une idée que nous proposons est de forcer ces atomes adjacents à se « ressembler ». Nous introduisons donc un apriori de ressemblance.

Celui s'applique à un ensemble de distributions unidimensionnelles $\boldsymbol{\theta}$ de coefficients $\theta_d = \theta_f^z$ (par exemple pour la PLCA, $\theta_f^z$ peut représenter $Z$ colonnes adjacentes de $P(f|n)$). Le but de l'apriori est d'encourager cet ensemble de paramètres à vérifier :

$$\forall f, \quad \theta_f^1 \approx \theta_f^2 \approx \cdots \theta_f^Z. \tag{3.14}$$

Pour cela, nous utilisons, à l'instar de l'apriori de continuité temporelle, une mesure qui calcule le rapport des moyennes géométriques et arithmétiques (aussi utilisée pour calculer la platitude spectrale [Joh88]). L'apriori de ressemblance est ainsi défini comme :

$$Pr(\boldsymbol{\theta}) \propto \left( \prod_f \frac{\sqrt[Z]{\prod_z \theta_f^z}}{\frac{1}{Z}\sum_z \theta_f^z} \right)^{Z\beta_{\text{res}}} \tag{3.15}$$

où $\beta_{\text{res}}$ est l'hyperparamètre permettant de contrôler la force de l'apriori. Avec un tel apriori, la fonction (3.2) (p.42), à optimiser sous la contrainte que $\forall t, \quad \sum_f \theta_f^z = 1$, devient :

$$\begin{aligned} \mathcal{M} \ : \ \Omega = ]0,1[^{F\times Z} &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_{f,z} w_f^z \ln\left(\theta_f^z\right) + Z\beta_{\text{res}} \sum_f \left( \frac{1}{Z}\sum_z \ln\left(\theta_f^z\right) - \ln\left(\frac{1}{Z}\sum_z \theta_f^z\right)\right). \end{aligned} \tag{3.16}$$

On sait que l'argument $\hat{\boldsymbol{\theta}}$, qui maximise $\mathcal{M}$ tout en vérifiant la contrainte $\forall t, \quad 1 - \sum_f \theta_f^z = 0$, existe puisque $\mathcal{M}$ est majorée par 0 sur $\Omega$. De plus, il vérifie les conditions de KKT : pour tout

$z$, il existe un unique $\rho^z \in \mathbb{R}$ tel que

$$\forall f, \quad \hat{\theta}_f^z = \frac{w_f^z + \beta_{\text{res}}}{\rho^z + \frac{Z\beta_{\text{res}}}{\sum_z \hat{\theta}_f^z}}. \tag{3.17}$$

Malheureusement, de même que pour l'apriori de continuité temporelle, il n'existe pas de solution analytique à cette équation, et l'on n'est pas sûr non plus qu'il en existe une unique. On peut alors proposer un algorithme du point fixe (Algorithme 4, linéaire en la taille des données) et vérifier empiriquement qu'il converge systématique vers une solution qui fait augmenter la valeur de $\mathcal{M}$. C'est bel et bien le cas dans la pratique, et d'après l'algorithme GEM, on est assuré que la log-probabilité *a posteriori* augmente à chaque itération.

**Algorithme 4:** Méthode du point fixe pour l'apriori de ressemblance.

$\forall\,(f,z) \in [\![1,F]\!] \times [\![1,Z]\!], \hat{\theta}_f^z \leftarrow \frac{w_f^z}{\sum_{\tilde{f}} w_{\tilde{f}}^z}$;

**répéter**

- $\forall f \in [\![1,F]\!],\ S_f \leftarrow Z\beta_{\text{res}} / \sum_z \hat{\theta}_f^z$;
- $\forall t \in [\![1,Z]\!]$, trouver l'unique $\rho^z$ tel que $\sum_f \frac{w_f^z + \beta_{\text{res}}}{\rho^z + S_f} = 1$ et $\forall f,\ \frac{w_f^z + \beta_{\text{res}}}{\rho^z + S_f} \geq 0$ (il est possible d'utiliser par exemple la méthode de Laguerre [Mek01]);
- $\forall (f,z) \in [\![1,F]\!] \times [\![1,Z]\!],\ \hat{\theta}_f^z \leftarrow \frac{w_f^z + \beta_{\text{res}}}{\rho^z + S_f}$;

**jusqu'à** *convergence*;

**Détail d'implémentation.** Avec cet apriori, contrairement à celui de parcimonie, on observe que diminuer la valeur de $\beta_{\text{res}}$ d'une valeur assez élevée jusqu'à sa valeur prédéfinie à chaque itération lors d'une première étape dite de « post-initialisation » est une bonne stratégie pour éviter les maxima locaux.

**Application à la PLCA.** Revenons donc au modèle PLCA et appliquons l'apriori de ressemblance aux paramètres $P(f|n)$ pour chaque sous-ensemble de $Z$ colonnes adjacentes. Les mises à jour obtenues après dérivation de l'algorithme EM quand on applique l'apriori devient (la notation $\tilde{P}$ est utilisée pour les probabilités qui ne sont pas encore normalisées) :

$$P(n,t) \propto \sum_f V_{ft} P(n|f,t) \tag{3.18}$$

$$\tilde{P}(f|n) = \sum_t V_{ft} P(n|f,t) \tag{3.19}$$

$$\forall n_0 \in [\![0, \frac{N}{Z} - 1]\!], \quad P(f|n_0 Z + z) = \text{Res}\left(\tilde{P}(f|n_0 Z + z), \beta_{\text{res}}\right),\ z = 1 \cdots Z \quad \text{(algorithme 4)}, \tag{3.20}$$

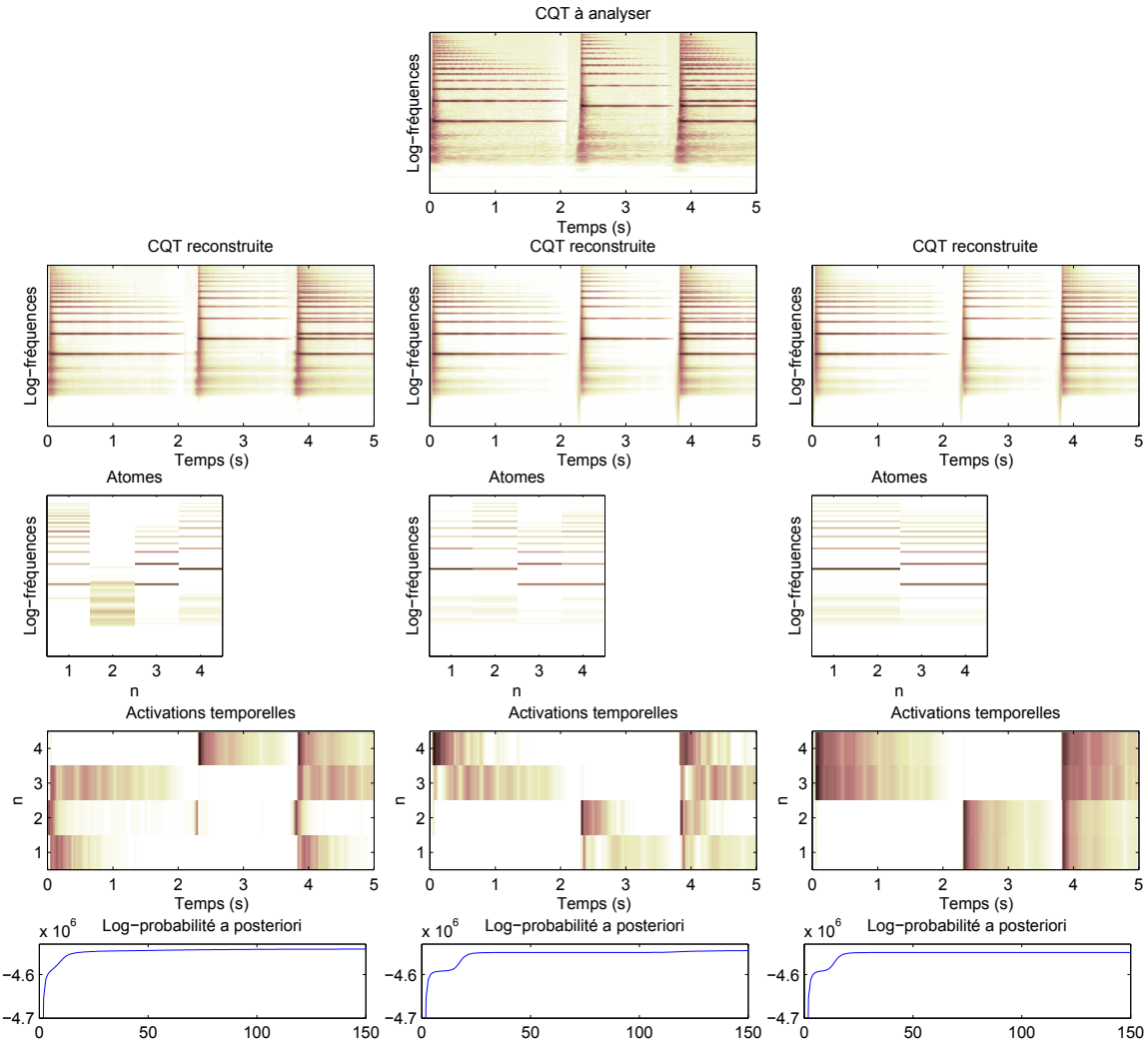
avec

$$P(n|f,t) = \frac{P(n,t)P(f|n,t)}{\sum_n P(n,t)P(f|n,t)}. \tag{3.21}$$

Pour illustrer l'apriori, nous prenons un exemple simple qui permet de bien visualiser son influence, même s'il ne présente pas d'intérêt applicatif. Une CQT d'entrée est constituée de deux notes de piano, d'abord jouées l'une à la suite de l'autre, puis en même temps. Pour l'analyser, on utilise non pas un mais deux atomes par notes, soit quatre atomes. Chaque atome est initialisé aléatoirement, tandis que les activations temporelles $P(n,t)$ sont initialisés uniformément. L'apriori de ressemblance est appliqué indépendamment aux deux premiers atomes $P(f|n=1,2)$ ainsi qu'aux deux suivants $P(f|n=3,4)$, nous lançons l'algorithme avec trois valeurs de $\beta_{\text{res}}$ différentes. De plus, comme pour l'exemple de l'application de l'apriori de parcimonie (section 3.2), un petit nombre d'atomes fixes supplémentaires, représentant des fenêtres à bande étroite, sont utilisés pour modéliser le bruit. Sur la figure 3.5, on peut observer le comportement de l'algorithme suivant la valeur de $\beta_{\text{res}}$ (exactement la même initialisation est faite pour chaque expérience). Quand $\beta_{\text{res}} = 0$, c'est-à-dire quand l'apriori n'est pas pris en compte, on remarque bien que les atomes 1 et 2 (ou 3 et 4) ne représentent pas la même note de musique, contrairement à ce que nous souhaitions. En effet, on devine dans cet exemple que les atomes 1 et 3 sont utilisés pour modéliser la première note, que l'atome 4 est utilisé pour modéliser la deuxième note, et que l'atome 2 est utilisé pour modéliser du bruit. Inversement, si $\beta_{\text{res}} = 5000$, c'est-à-dire s'il est très grand, alors on a une parfaite égalité des atomes 1 et 2, ainsi que des atomes 3 et 4. La cas $\beta_{\text{res}} = 50$ semble être un bon compromis : chaque paire d'atomes représente bien une même note (ils se ressemblent, sans être parfaitement égaux).

## 3.5 Apriori de monomodalité

Considérons dans cette section le modèle SIPLCA (modèle (2.18) page 34), et supposons que le signal d'entrée est la somme de $S$ instruments harmoniques monodiques (ne jouant qu'une unique note à la fois), $S$ étant supposé connu. Dans la SIPLCA, la CQT d'une source $s_0$ est modélisée comme la convolution d'un spectre de base (ou noyau) $P(\mu|s_0)$ et de ses activations temps-fréquence $P(i,t,s_0)$. Comme la source est censée être monodique, idéalement, à la fin de l'algorithme EM au temps $t_0$, le vecteur $P(i,t_0,s_0)_i$ devrait être unimodal (ne possédant qu'un unique maximum local), la valeur du mode correspondant à la différence relative entre la fréquence fondamentale du noyau et celle de la note jouée à $t_0$ par $s_0$. Or, comme rien ne contraint les activations temps-fréquence à converger comme tel, ce n'est bien évidement pas le cas, et en pratique, on se retrouve avec des vecteurs multimodaux. Pour étayer nos dires, prenons un exemple simple : le cas d'une trompette solo jouant quelques notes de musique. Pour analyser le signal, nous décomposons la CQT du signal avec le modèle SIPLCA en utilisant deux noyaux : un pour la composante harmonique de la trompette, et l'autre pour le bruit. A l'instar des expériences faites pour les aprioris de parcimonie et de continuité temporelle dans le cadre du




(a) $\beta_{res} = 0$ (b) $\beta_{res} = 50$ (c) $\beta_{res} = 5000$

**Figure 3.5** – Illustration de l'apriori de ressemblance. Les atomes fixes réservés pour la modélisation du bruit ne sont pas illustrés ici.

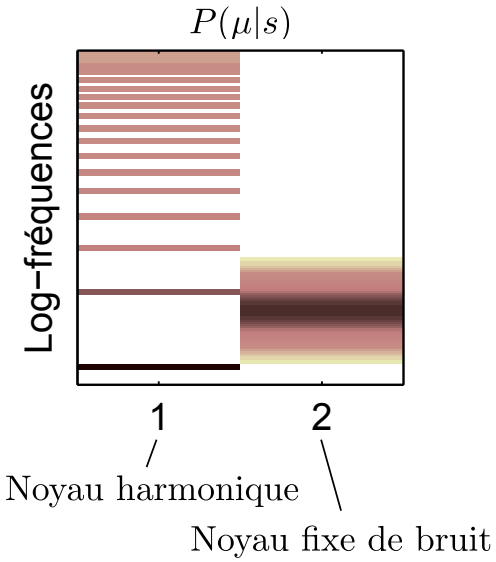


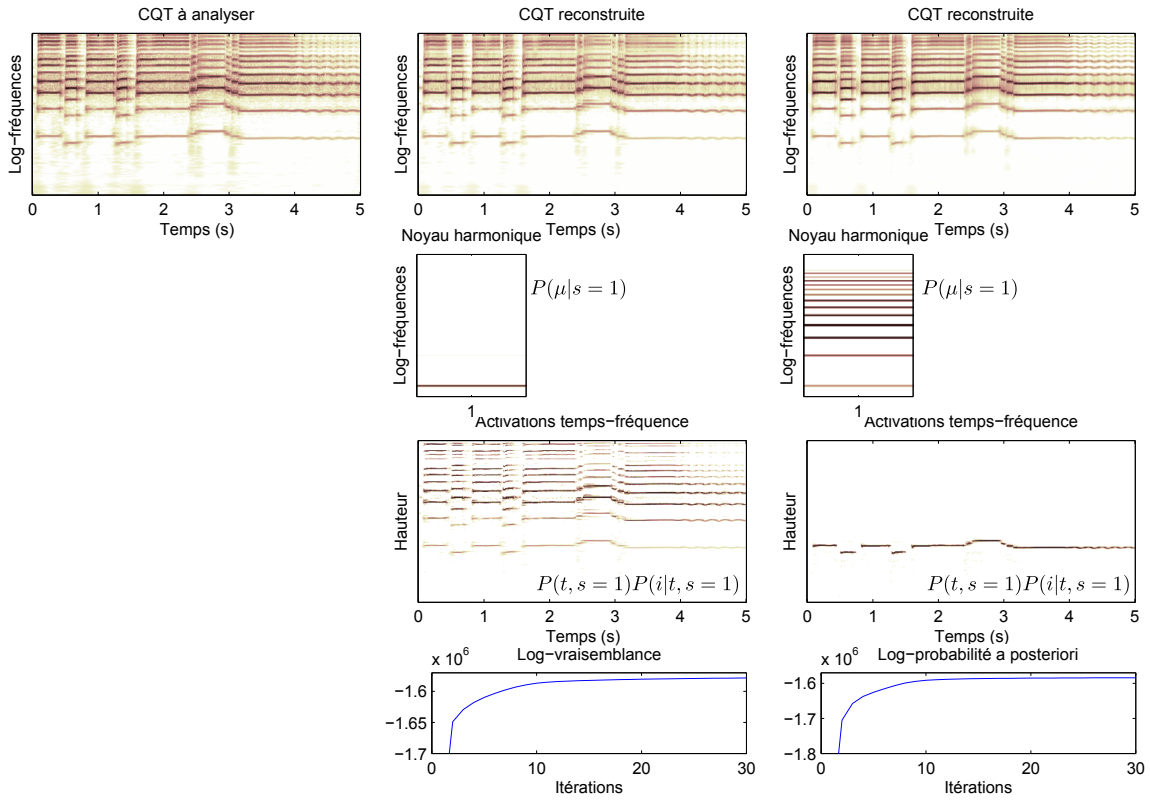

**Figure 3.6** – Initialisation des noyaux, en vu de l'illustration de l'apriori de monomodalité dans le cas de la SIPLCA.

modèle PLCA, le premier noyau est initialisé comme un spectre harmonique avec des zéros entre ses partiels, tandis que le deuxième, qui n'est pas mis à jour pendant l'algorithme, est modélisé comme une fenêtre à bande étroite. La figure 3.6 permet de visualiser l'initialisation de ces deux noyaux. Sur la figure 3.7 dans la colonne (a), on peut remarquer que l'algorithme SIPLCA n'a pas convergé tel que nous le souhaitions, puisque le noyau harmonique $P(\mu|s = 1)$ estimé n'est plus qu'une impulsion tandis que l'activation temps-fréquence correspondante possède des maxima aux fréquences des partiels de la note jouée. Puisque l'on souhaite garder uniquement le mode de plus basse fréquence, un apriori de monomodalité est introduit, forçant chaque colonne de la distribution d'impulsions (qui, lorsqu'elles sont normalisées, sont aussi des distributions de probabilité) à avoir à la fois une faible variance et une faible moyenne. Avant d'aller plus loin, on peut mentionner que dans [MS09], le cas de la monomodalité est également traité, grâce à un apriori de Dirichlet dont les paramètres ont une forme gaussienne (ils sont paramétrisés comme des gaussiennes, définies par deux hyperparamètres : valeur du mode et variance). Le souci est que l'hyperparamètre de la valeur du mode est redéfini à chaque itération en fonction de la valeur courante des paramètres du modèle (c'est-à-dire des données observées). Il ne s'agit donc pas véritablement d'une distribution *a priori* des paramètres, dans le sens strict du terme. Ici, les hyperparamètres de l'apriori de monomodalité que nous proposons sont fixés avant l'algorithme et sont donc indépendants des observations à analyser.

Notre apriori s'applique à des distributions unidimensionnelles, et par conséquent, le modèle SIPLCA a besoin d'être légèrement adapté : la distribution des activations temps-fréquence est décomposée comme

$$P(i, t, s) = P(t, s)P(i, |t, s) \tag{3.22}$$

où $P(t, s)$ et $P(i|t, s)$ représentent respectivement l'énergie de l'instrument $s$ au temps $t$ et la




(a) Sans apriori (b) Avec apriori de monomodalité

**Figure 3.7** – Illustration avec deux sources (une pour la partie harmonique, l'autre pour le bruit) de l'utilisation de l'apriori de monomodalité dans le cas du modèle SIPLCA : si les CQTs estimées restent quasiment inchangées, les activations temps-fréquence deviennent monomodales à chaque temps (l'apriori est juste appliqué aux activations temps-fréquence de la première source). Dans les deux cas, le critère de convergence croît au fil des itérations. Le signal d'entrée est un extrait de Summertime (G. Gershwin) joué à la trompette.

distribution de hauteur correspondante. Si aucun apriori n'était ajouté, alors l'équation (2.22) deviendrait l'ensemble d'équations

$$P(i|t,s) \propto \sum_f V_{ft} P(i,s|f,t) \tag{3.23}$$

$$P(t,s) \propto \sum_{f,i} V_{ft} P(i,s|f,t). \tag{3.24}$$

Si la mise à jour de $P(t,s)$ va bel et bien s'exprimer de la sorte, celle de $P(i|t,s)$ va changer puisque l'apriori de monomodalité proposé lui est appliqué. Celui-ci est fondé sur une mesure adéquate que nous appelons variance asymétrique (par souci de simplicité, nous fixons une source $s$ et un temps $t$, et définissons $\boldsymbol{\theta}$ comme le vecteur de coefficients $\theta_d = \theta_i = P(i|t,s)$, c'est-à-dire $d = i$) :

$$\begin{aligned} \text{avar}_\gamma(\boldsymbol{\theta}) &= \sum_i \left( \exp(\gamma i) - \exp\left( \gamma \sum_{\tilde{i}} \tilde{i}\theta_{\tilde{i}} \right) \right) \theta_i \\ &= \left( \sum_i \exp(\gamma i)\, \theta_i \right) - \exp\left( \gamma \sum_i i\theta_i \right) \text{ puisque } \sum_i \theta_i = 1. \end{aligned} \tag{3.25}$$

Cette mesure dépend de l'hyperparamètre $\gamma > 0$ qui définit la force de l'asymétrie. On peut prouver, grâce à la convexité stricte de la fonction exponentielle, que

$$\text{avar}_\gamma(\boldsymbol{\theta}) \geq 0,$$

et que

$$\text{avar}_\gamma(\boldsymbol{\theta}) = 0 \Leftrightarrow \exists\, i_0 \mid \forall i,\ \theta_i = 1 \text{ if } i = i_0 \text{ et } 0 \text{ sinon.}$$

Dans le but de contraindre $\text{avar}_\gamma(\boldsymbol{\theta})$ à être faible lors de l'exécution de l'algorithme, on introduit l'apriori suivant :

$$Pr(\boldsymbol{\theta}) \propto \exp\left(-\beta_{\text{mono}} \text{avar}_\gamma(\boldsymbol{\theta})\right) \tag{3.26}$$

où $\beta_{\text{mono}} > 0$ est un hyperparamètre indiquant la force de l'apriori. L'étape MAP consiste alors à maximiser sous la contrainte $\sum_i \theta_i = 1$ l'équation (3.2), qui devient dans ce cas :

$$\begin{aligned} \mathcal{M} \ :\ \Omega = ]0,1[^I &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_i w_i \ln(\theta_i) - \beta_{\text{mono}} \left( \sum_i \exp(\gamma i)\, \theta_i \right) + \beta_{\text{mono}} \exp\left( \gamma \sum_i i\theta_i \right), \end{aligned} \tag{3.27}$$

où $w_i = \sum_f V_{ft} P(i,s|f,t)$ dans le cas de la SIPLCA. Si l'on note $\hat{\boldsymbol{\theta}}$ le maximum que nous cherchons, alors selon les conditions de KKT [KT51], il existe un unique $\nu \in \mathbb{R}$ tel que

$$\forall i \in \mathbb{Z},\ \frac{\partial}{\partial \theta_i} \left( \mathcal{M}(\boldsymbol{\theta}) + \nu \left( 1 - \sum_i \theta_i \right) \right) = 0,$$

soit

$$\forall i \in \mathbb{Z},\ \hat{\theta}_i = \frac{w_i}{\beta_{\text{mono}} \left( \exp(\gamma i) - \gamma i \exp\left( \gamma \sum_{\tilde{i}} \tilde{i} \hat{\theta}_{\tilde{i}} \right) \right) + \nu}. \tag{3.28}$$

Malheureusement, de même que pour l'apriori de continuité, il n'y a pas de solution analytique à cette équation, et l'on ne sait pas non plus s'il n'existe qu'une unique solution. Cependant, les simulations numériques ont montré que la méthode du point fixe décrite dans l'Algorithme 5 converge toujours vers une solution qui fait croître la valeur de $\mathcal{M}$. Même si on ne peut pas être sûr qu'elle corresponde au maximum global, l'algorithme GEM nous permet quand même d'assurer la croissance de la log-probabilité *a posteriori* des paramètres sachant les observations (équation (2.2) p.30) au fil des itérations. Cet algorithme est lancé de manière indépendante aux valeurs de $t$ et $s$ souhaitées. La figure 3.7 illustre l'effet de l'utilisation de cet apriori, ainsi que la croissance du critère de convergence au fil des itérations. Finalement, l'algorithme MAP

**Algorithme 5:** Méthode du point fixe pour l'apriori de monomodalité.

$\forall\, i \in [\![0, I-1]\!], \hat{\theta}_i \leftarrow \frac{w_i}{\sum_{\tilde{i}} w_{\tilde{i}}}$ ;

**répéter**

- $m \leftarrow \sum_i i\hat{\theta}_i$;
- $\forall i \in [\![1, I]\!], c_i \leftarrow \beta_{\text{mono}} \left( e^{\gamma i} - \gamma i e^{\alpha m} \right)$;
- trouver l'unique réel $\nu$ tel que $\sum_i \frac{w_i}{c_i + \upsilon} = 1$ et $\forall i, \frac{w_i}{c_i+\upsilon} \geq 0$ (il est possible d'utiliser par exemple la méthode de Laguerre [Mek01]);
- $\forall\, i \in [\![0, I-1]\!],\ \hat{\theta}_i \leftarrow \frac{w_i}{c_i+\upsilon}$;

**jusqu'à** *convergence*;

n'est différent de l'algorithme EM que lors de la dernière étape de normalisation : les paramètres soumis à l'apriori de monomodalité sont « normalisés » via l'Algorithme 5.

**Détail d'implémentation** Avec cet apriori, comme avec celui de parcimonie, on observe qu'augmenter la valeur de $\beta_{\text{mono}}$ de 0 jusqu'à sa valeur prédéfinie à chaque itération lors d'une première étape dite de « post-initialisation » est une bonne stratégie pour éviter les maxima locaux.

## 3.6 Conclusion

Dans ce chapitre, nous avons introduit quatre types d'aprioris, permettant de mieux prendre en compte la nature des données à analyser. D'abord, nous avons étudié trois aprioris de parcimonie, pour lesquels nous avons trouvé des solutions analytiques (à une recherche numérique de racine près) pour la résolution de l'étape MAP. Une rapide analyse expérimentale nous a permis de retenir l'un des trois aprioris, fondé sur la norme $l_{1/2}$, qui semblait présenter un comportement intéressant. Trois aprioris supplémentaires de continuité temporelle, de ressemblance et de monomodalité ont également été introduits. Pour ceux-là, nous n'avons hélas pas réussi à trou-

ver des solutions analytiques pour la résolution de l'étape MAP, mais nous avons proposé des algorithmes du point fixe qui, dans la pratique, semblent toujours converger vers une solution. Nous n'avons pas encore réussi à trouver des preuves de convergence pour ces algorithmes, mais c'est un problème auquel nous souhaiterions nous atteler dans un travail futur.

Nous avons donné des exemples d'applications de ces aprioris à la PLCA ou la SIPLCA, ce qui nous a montré leur efficacité. Ils seront réutilisés dans la partie III, consacrée à des nouveaux modèles de CQT de signaux musicaux.

# Chapitre 4

# Maîtriser la vitesse de convergence des paramètres

## 4.1 Introduction

Dans le chapitre précédent, nous avons vu que l'ajout d'aprioris permettait de mieux contrôler la décomposition, pour que les paramètres convergent vers une solution plus significative. Dans ce chapitre, nous étudions une nouvelle idée, très simple, qui consiste à ralentir la convergence de certains paramètres. Très facile à mettre en œuvre, nous allons voir qu'elle permet d'insérer de l'information sur la nature des données, et cela de deux manières différentes :

- freiner un ensemble de paramètres peut permettre de supposer que leurs valeurs à la fin de l'algorithme doivent être proches de leurs initialisations ;
- si les données sont parcimonieuses et si les paramètres sont initialisés de manière non-parcimonieuse, alors freiner un sous-ensemble permet de choisir quels paramètres seront parcimonieux : ceux qui ne sont pas freinés.

Freiner un ensemble de paramètres, tout en restant dans le cadre de l'algorithme EM, repose sur une astuce lors du processus génératif : tirer un certain nombre de variables (cachées), correspondant aux paramètres dont on veut ralentir la convergence, puis les jeter. Elles ne sont donc liées à aucune observation. Même si cela n'a aucun intérêt en soi, puisqu'on reste dans le cadre de l'algorithme EM, on est sûr que la vraisemblance (ou la probabilité *a posteriori* des paramètres) augmente toujours au fil des itérations. Ici, nous allons introduire cette astuce et étudier expérimentalement ses implications dans le cadre du modèle classique de la PLCA, mais elle pourra s'appliquer à n'importe quel type de modèle de RTF$^+$, comme nous le ferons dans la partie III. Le contenu de ce chapitre n'a pas encore fait l'objet d'une publication.

## 4.2 Une astuce simple

Nous restons donc dans le cadre de la PLCA classique, dont le modèle est donné, pour rappel, par

$$P(f,t) = \sum_n P(n,t)P(f|n) \tag{4.1}$$

et modifions légèrement le processus génératif des observations $V_{ft}$ décrit dans la section 2.1.1 page 29 pour y insérer notre astuce :

- ∀ $(f,t) \in [\![1,F]\!] \times [\![1,T]\!]$, on pose $V_{ft} = 0$
- Répéter $J$ fois :
  - tirer $(n,t)$ selon $P(n,t)$,
  - tirer $f$ selon $P(f|n)$,
  - poser $V_{ft} = V_{ft} + 1$,
- répéter $\beta^1_{\text{frein}}$ fois :
  - tirer $(n^0,t^0)$ selon $P(n^0,t^0)$ et ne rien faire de ces variables,
- pour chaque $n$, répéter $\beta^2_{\text{frein}}$ fois :
  - tirer $f^n$ selon $P(f^n|n)$ et ne rien faire non plus de cette variable.

Pour la dérivation de l'algorithme EM, on procède de manière classique. La log-probabilité jointe est donnée par :

$$\begin{aligned}
\mathcal{L}_\Lambda\left(\bar{f},\bar{t},\bar{n},\bar{n^0},\bar{t^0f^1},\cdots,\bar{f^N}\right) &= \ln\left(P\left(\bar{f},\bar{t},\bar{n},\bar{n^0},\bar{t^0f^1},\cdots,\bar{f^N}\right)\right) \\
&= \sum_{j=1}^{J} \ln\left(P\left(n_j,t_j\right)\right) + \ln\left(P\left(f_j|n_j\right)\right) \\
&\quad + \sum_{j=1}^{\beta^1_{\text{frein}}} \ln\left(P\left(n_j^0,t_j^0\right)\right) + \sum_n \sum_{j=1}^{\beta^2_{\text{frein}}} \ln\left(P\left(f_j^n|n\right)\right)
\end{aligned} \tag{4.2}$$

et son espérance conditionnelle par :

$$\begin{aligned}
Q_\Lambda = &\sum_{\bar{n},\bar{n^0},\bar{t^0f^1},\cdots,\bar{f^N}} P(\bar{n},\bar{n^0},\bar{t^0f^1},\cdots,\bar{f^N}|\bar{f},\bar{t})\Big[\sum_{j=1}^{J} \ln\left(P(n_j,t_j)\right) + \ln\left(P(f_j|n_j)\right) \\
&+ \sum_{j=1}^{\beta^1_{\text{frein}}} \ln\left(P\left(n_j^0,t_j^0\right)\right) + \sum_n \sum_{j=1}^{\beta^2_{\text{frein}}} \ln\left(P\left(f_j^n|n\right)\right)\Big] \\
= &\sum_{j=1}^{J}\sum_{n_j} P(n_j|f_j,t_j)\left[\sum_{j=1}^{J} \ln\left(P(n_j,t_j)\right) + \ln\left(P(f_j|n_j)\right)\right] \\
&+ \sum_{j=1}^{\beta^1_{\text{frein}}} \sum_{n_j^0,t_j^0} P(n_j^0,t_j^0|-) \ln\left(P(n_j^0,t_j^0)\right) + \sum_n \sum_{j=1}^{\beta^2_{\text{frein}}} \sum_{f_j^n|} P(f_j^n|-) \ln\left(P(f_j^n|n)\right)
\end{aligned}$$

soit

$$Q_\Lambda = \sum_{f,t}\sum_{n} V_{ft} P\left(n|f,t\right)\left[\ln\left(P(n,t)\right) + \ln\left(P(f|n)\right)\right] + \beta^1_{\text{frein}} \sum_{n,t} P(n,t|-)\ln\left(P(n,t)\right) + \beta^2_{\text{frein}} \sum_{n}\sum_{f} P^n(f|-)\ln\left(P(f|n)\right). \tag{4.3}$$

La notation $P(x|-)$ est utilisée pour signifier qu'il s'agit bien d'une probabilité *a posteriori* mais que la variable cachée $x$ ne dépend d'aucune observation. De plus la notation $P^n(f|-)$ signifie que la probabilité dépend de la valeur de $n$.

Lors de l'*étape E*, on calcule les probabilités *a posteriori* en fonction des paramètres du modèle :

$$P(n|f,t) = \frac{P(n,t)P(f|n)}{P(f,t)} \tag{4.4}$$

$$P(n,t|-) = P(n,t) \tag{4.5}$$

$$P^n(f|-) = P(f|n) \tag{4.6}$$

où $P(f,t)$ est donné par l'équation (4.1). Lors de l'*étape M*, on maximise $Q_\Lambda$ par rapport aux paramètres du modèle, en se souvenant que les probabilités doivent sommer à un. Cela donne :

$$P(n,t) \propto \sum_f V_{ft} P(n|f,t) + \beta^1_{\text{frein}} P(n,t|-), \tag{4.7}$$

$$P(f|n) \propto \sum_t V_{ft} P(n|f,t) + \beta^2_{\text{frein}} P^n(f|-). \tag{4.8}$$

En regroupant les étapes E et M, on peut déduire des mises à jours multiplicatives pour les paramètres :

$$P(n,t) \propto P(n,t)\left[\sum_f \frac{V_{ft}}{P\left(f,t\right)} P(f|n) + \beta^1_{\text{frein}}\right] \tag{4.9}$$

$$P(f|n) \propto P(f|n)\left[\sum_t \frac{V_{ft}}{P\left(f,t\right)} P(n,t) + \beta^2_{\text{frein}}\right]. \tag{4.10}$$

On se retrouve donc avec quasiment les mêmes mises à jours que les équations (2.12) et (2.13) page 32, mis à part ces deux coefficients $\beta^1_{\text{frein}}$ et $\beta^2_{\text{frein}}$. En fait, ce sont ces coefficients qui jouent le rôle de frein dans la convergence des paramètres : plus ils sont grands, plus la valeur des paramètres à une itération donnée sera proche de celle de l'itération précédente. Nous les appellerons donc coefficients de freinage. L'idée sous-jacente est qu'en jouant sur la valeur de ces deux coefficients, on maitrise la vitesse de convergence de tel ou tel jeu de paramètres.

Avant de parler des applications possibles dans la section suivante, on peut remarquer que si l'on multiplie $V_{ft}$, $\beta^1_{\text{frein}}$ et $\beta^2_{\text{frein}}$ par un même scalaire positif quelconque, les mises à jour

| Expérience | $\beta^1_{\text{frein}}$ (coef. de freinage de $P(n,t)$) | $\beta^2_{\text{frein}}$ (coef. de freinage de $P(f\|n)$) |
|---|---|---|
| *Exp1* | 0 | 0 |
| *Exp2* | 375 | 0 |
| *Exp3* | 0 | 250 |

**Table 4.1** – Expérience sur la maîtrise de la vitesse de convergence dans le cas de la PLCA : valeurs des coefficients de freinage.

restent parfaitement identiques. Il n'est donc pas nécessaire de fixer $\beta^1_{\text{frein}}$ et $\beta^2_{\text{frein}}$ à des valeurs entières, de même qu'il n'est pas nécessaire de mettre à l'échelle une observation $V_{ft}$ pour que les coefficients qui la composent soient entiers.

## 4.3 Étude expérimentale

Nous explorons ici l'utilisation qui peut être faite de la maitrise de la vitesse de convergence des paramètres dans un modèle de type PLCA. Partons pour cela d'une expérience : on applique une PLCA à une même CQT d'entrée, mais avec différentes valeurs des coefficients de freinage, et l'on observe les résultats. Pour ne pas qu'ils soient dépendants des initialisations, on initialise $P(n,t)$ à la distribution uniforme et $P(f|n)$ aléatoirement, l'initialisation restant la même pour chacune des expériences. Le signal d'entrée utilisé est un extrait de piano, et le nombre de spectres de base est fixé à 88 (on reste dans l'optique où chaque spectre de base représente une des notes MIDI). On peut préciser que dans le signal d'entrée, la totalité des 88 notes MIDI n'est pas présente, comme cela est quasiment toujours le cas dans un morceau de musique. Dans la Table 4.1, on définit les différentes valeurs de $\beta^1_{\text{frein}}$ et $\beta^2_{\text{frein}}$ pour chacune des trois expériences et sur la figure 4.1, on illustre les spectres de base et les activations temporelles estimés correspondant après convergence de l'algorithme EM.

On observe bien une différence notable entre les différentes estimations. En fait, la principale différence réside dans la parcimonie des paramètres. Le fait est que la CQT d'entrée est, par nature, plutôt parcimonieuse et par conséquent, cette parcimonie se retrouve nécessairement dans les paramètres. Ce dont on se rend compte, selon que l'on freine la convergence de tel ou tel paramètre, est que la parcimonie se retrouve soit dans les spectres de base (*Exp2*), soit dans les activations (*Exp3*). Pour expliquer ce phénomène, prenons le cas qui nous intéresse le plus : l'expérience *Exp3*, où ce sont les paramètres représentant les spectres de base $P(f|n)$ qui sont freinés. A l'initialisation, les paramètres $P(n,t)$ et $P(f|n)$ ne sont pas parcimonieux. Puisqu'on force $P(n,t)$ à converger plus rapidement, c'est cet ensemble de paramètres qui va en premier s'adapter aux données et ainsi devenir parcimonieux, à l'image des observations. Une propriété remarquable que l'on observe dans ce cas, est que tous les spectres de base ne sont pas mis à contribution pour modéliser la CQT d'entrée. On le remarque grâce à la visualisation de $P(n)$ (figure 4.1, colonne (c)) : de nombreux noyaux ne sont en réalité jamais activés, contrairement aux autres expériences. Cette propriété est intéressante dans le cadre de

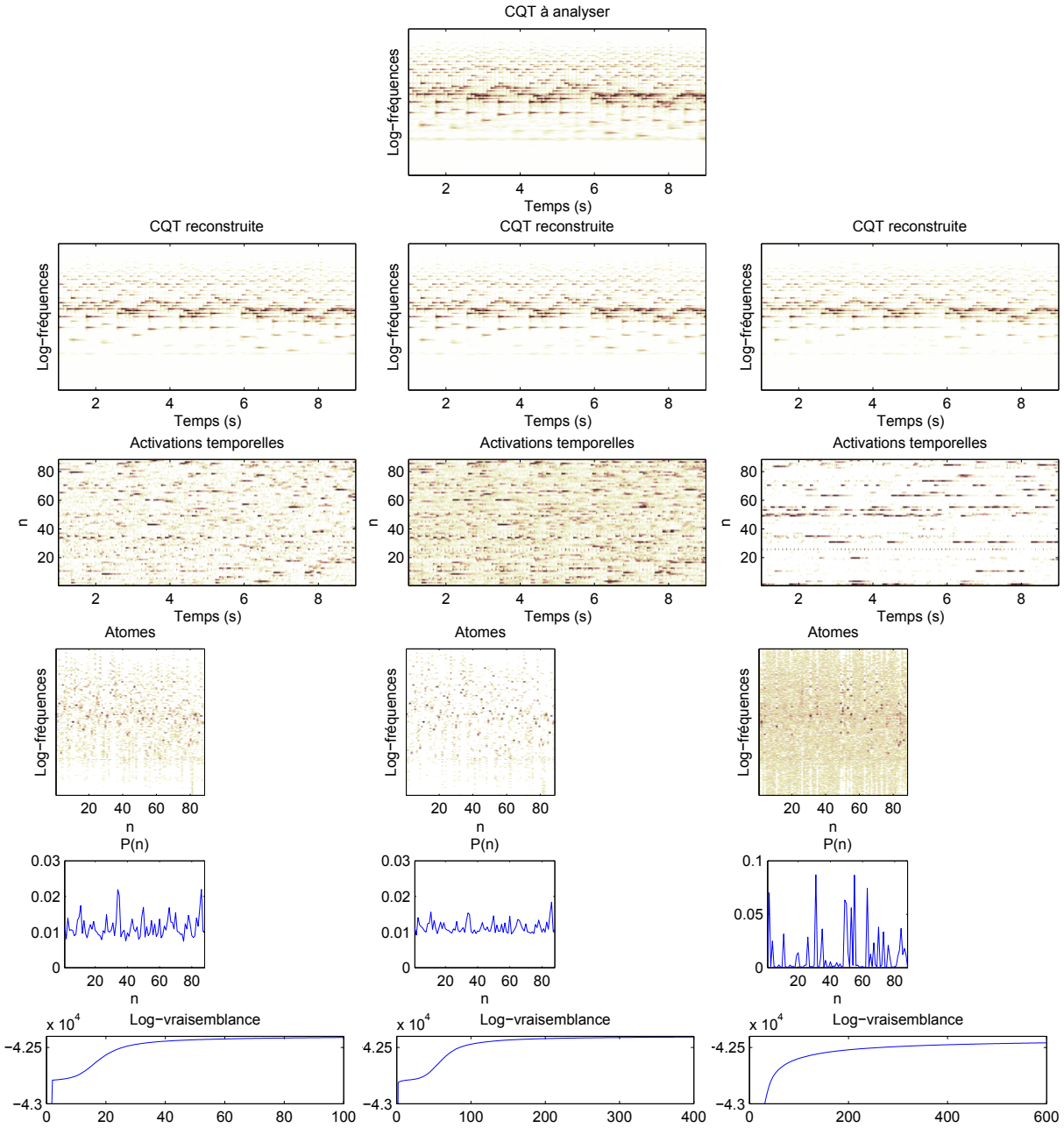



**Figure 4.1** – Maîtrise de la vitesse de convergence : illustration de trois différents scénarii (*cf.* Table 4.1). Le signal d'entrée, d'une durée de 8 secondes, est un extrait du Prélude en Ré majeur BWV 850 de J.S. Bach (piano). Les activations temporelles sont les paramètres $P(n,t)$, les noyaux $P(f|n)$ et $P(n) = \sum_t P(n,t)$ représentent l'énergie totale de chaque atome dans le signal.

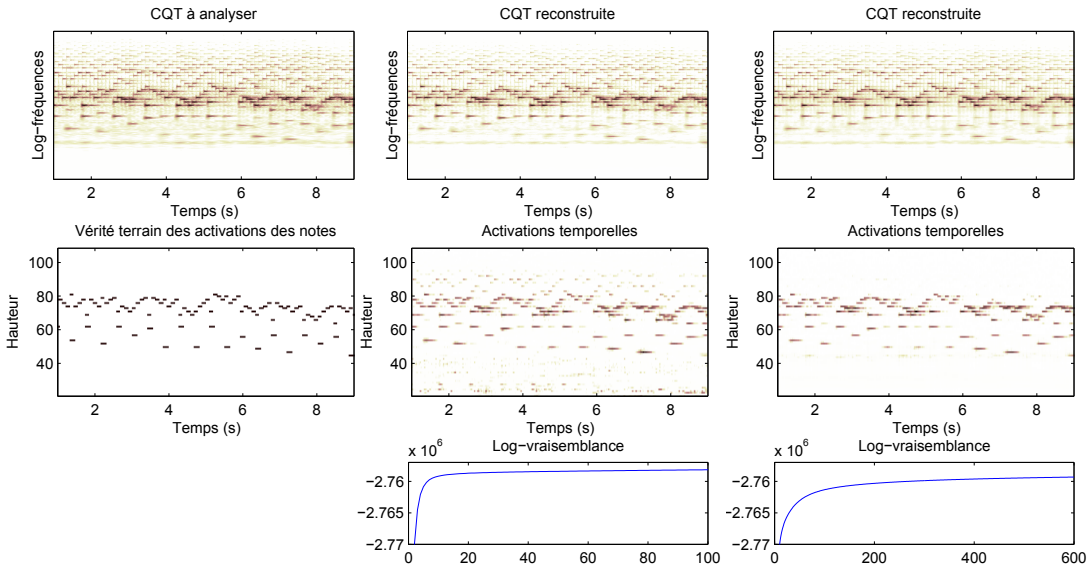
la transcription automatique, puisque comme nous l'avons mentionné précédemment, la totalité des 88 notes MIDI n'est pas nécessairement présente dans un signal de musique.

Nous retenons donc qu'il est intéressant de pouvoir freiner la convergence des spectres de base puisque cela rend les activations plus parcimonieuses. On peut tout de même se questionner sur la convergence de la log-vraisemblance. Avec l'astuce que nous proposons dans ce chapitre, on reste dans le cadre théorique de l'algorithme EM, et par conséquent, on est assuré de la non décroissance de la log-vraisemblance des observations au fil des itérations. En revanche, on remarque bien sur la figure 4.1 une différence flagrante des vitesses de convergence. Bien entendu, en freinant la convergence d'un sous-ensemble de paramètres, on ne s'attendait pas à ce que l'algorithme converge plus rapidement, mais le cas de l'expérience *Exp3* nous interpelle particulièrement. En effet, si la vitesse de convergence entre les expériences *Exp1* et *Exp2* ne semble différer que d'un facteur d'échelle (facteur 4 environ dans cet exemple), pour l'expérience *Exp3* c'est l'ordre de la vitesse de convergence qui semble différent. Nous n'avons pas cherché à trouver une explication théorique à ce phénomène, mais le lecteur pourra se référer à [BBV10a], où sont analysés stabilité et convergence des algorithmes à mises à jour multiplicatives. Une autre question nous vient à l'esprit en regardant les courbes de log-vraisemblance des trois expériences : pourquoi l'expérience *Exp3* donnerait-elle des résultats plus satisfaisants, alors que la log-vraisemblance des observations est clairement plus faible, même après 600 itérations ? Pour répondre à cela, nous nous réfèrerons à [Ber09, section IV.2.2, p. 74], où il est observé qu'aucune corrélation n'existe entre la valeur finale du critère à la fin de l'algorithme et la pertinence sémantique des paramètres estimés. Nous nous apercevrons nous-même de l'amélioration des résultats de transcription automatique grâce à cette astuce, lorsque nous l'appliquerons aux modèles des chapitres 6 et 7.

Dans la figure 4.2, on compare les résultats des expériences *Exp1* (les coefficients de freinage sont nuls) et *Exp3* (les spectres de base sont freinés), dans le cas où $P(f|t)$ est initialisé comme dans l'exemple proposé pour traiter l'apriori de parcimonie (section 3.2 page 42). Pour rappel, 88 noyaux harmoniques, ayant des zéros entre leurs partiels, sont utilisés pour représenter les notes, et 10 noyaux fixes, représentant des fenêtres lisses à bande étroite, sont utilisés pour modéliser le bruit (*cf.* figure 3.2 page 47). On remarque que le coefficient de freinage $\beta^2_{\text{frein}}$ joue un rôle similaire à l'apriori de parcimonie sur les activations temps-fréquence. Ici, on peut donner une autre interprétation à l'ajout de ce coefficient. En effet, freiner la convergence $P(f|n)$ peut être vu comme la connaissance *a priori* que la vraie valeur de $P(f|n)$ est proche de son initialisation. C'est finalement un moyen assez simple d'ajouter un apriori sur les paramètres.

## 4.4 Conclusion

Dans ce chapitre, une astuce simple a été introduite pour freiner la vitesse de convergence d'un ensemble de paramètres. Elle consiste à tirer un certain nombre de variables cachées, qui ne sont liées à aucune observation. Si nous ne savons *a priori* pas pourquoi cela ralentit la vitesse




(a) *Exp1* (b) *Exp3*

**Figure 4.2** – Maîtrise de vitesse de convergence : illustration des deux scénarii *Exp1* et *Exp3* (*cf.* Table 4.1), dans le cas où les noyaux sont initialisés selon la figure 3.2 page 47). Le signal d'entrée, d'une durée de 8 secondes, est un extrait du Prélude en Ré majeur BWV 850 de J.S. Bach (piano). Les activations temporelles représentent les paramètres $P(n,t)$ pour $n \in [\![1, 88]\!]$.

de convergence, nous remarquons, en dérivant l'algorithme EM, que cette astuce a pour effet l'introduction de coefficients de freinage dans la mise à jour des paramètres. Nous avons vu, grâce à deux exemples, que freiner un sous-ensemble de paramètres pouvait permettre deux choses : accentuer la parcimonie des paramètres non freinés dans le cas où un modèle d'observation est sur-dimensionné, et encourager les paramètres freinés à converger vers une solution proche de leur initialisation.

Si nous n'avons étudié que le cas du modèle PLCA classique, cette idée de maîtrise de vitesse de convergence peut s'appliquer à n'importe quel modèle d'observation, et nous l'utiliserons dans les modèles de la partie III.

# Chapitre 5

# PLCA avec structure temporelle : LCATS

## 5.1 Les motivations

Dans le modèle PLCA classique (ou NMF classique), c'est-à-dire quand chaque colonne de $\text{RTF}^+$ est modélisée comme une somme pondérée de spectres de base, si l'on intervertit aléatoirement les colonnes des données d'entrée, cela ne change rien à la décomposition : les colonnes des activations seront juste interchangées en conséquence. Dans le cas de la PLCA, cela est dû au caractère indépendant de chaque tirage des points temps-fréquence $(f, t)$. On peut alors légitimement se dire que ce modèle n'est pas assez complet, car il ne permet pas de modéliser l'évolution dynamique d'un signal. Bien entendu, il est possible de remédier à ce défaut via l'incorporation d'un apriori de continuité temporelle (comme celui présenté à la section 3.3 page 47), mais ici nous souhaiterions disposer d'un modèle permettant de modéliser directement les variations temporelles.

Cette volonté d'inclure une dimension temporelle dans les modèles de décomposition de $\text{RTF}^+$ est depuis récemment source de nombreuses recherches [OFC09, NLK+10, Mys10], comme nous l'avons annoncé dans l'état de l'art (section 1.3 page 18). Ces méthodes citées reposent sur des modèles à états, permettant de changer de dictionnaire de spectres au cours du temps pour représenter des notes ou des sources dans un mélange, et de modéliser la manière dont les dictionnaires se succèdent. Un problème de ces modèles est que leur complexité est factorielle relativement aux nombres d'états et de sources considérés ce qui les rend dans la pratique inutilisables quand ceux-ci sont trop élevés, à moins d'utiliser des méthodes approchées de type méthodes variationnelles [MS12].

Dans ce chapitre nous proposons une nouvelle manière d'introduire une structure temporelle dans un modèle de type PLCA, qui n'est pas un modèle factoriel. La motivation originale a été d'imaginer un modèle où la probabilité que la variable cachée $n$ ait pour valeur $n_0$ au temps $t$ dépendait de la probabilité qu'elle ait cette même valeur au temps $t-1$. Ainsi, nous avons

essayé d'imaginer des processus génératifs de RTF$^+$ de type PLCA où les tirages n'étaient pas indépendants, et cela sans ajouter une couche de variables cachées d'états comme dans [Mys10]. La solution que nous avons trouvée est légèrement différente : les tirages sont bien indépendants, en revanche, pour chacun d'entre eux, on tire une série temporelle de variables cachées, présentant, elles, des dépendances.

Afin de laisser place à d'éventuels futurs travaux, nous allons d'abord présenter un méta-modèle, appelé analyse en composantes latentes avec structure temporelle (LCATS pour *Latent Component Analysis with Temporal Structure*), où nous n'allons pas préciser de modèle de structure temporelle. Nous allons le présenter comme une extension de la PLCA classique, que nous exprimons sous sa forme asymétrique (*cf.* section 2.1) :

$$P(f,t) = P(t)\sum_n P(n|t)P(f|n) \tag{5.1}$$

Il sera ensuite décliné selon deux cas différents : le cas indépendant, qui nous permettra de vérifier si LCATS est oui ou non une généralisation de la PLCA, et le cas markovien, intitulé MLCATS (pour *Markovian LCATS*) et qui est le cas qui nous intéresse le plus. Ces travaux sont assez récents et ils ne seront donc pas utilisés pour une application précise dans la suite de ce mémoire. Ils n'ont pas non plus encore été publiés.

## 5.2 Méta-modèle

### 5.2.1 Processus génératif et log-vraisemblance

Contrairement à l'approche faite pour la présentation de la PLCA (section 2.1), nous commençons ici directement par la description du processus génératif pour une TFR$^+$ $V_{ft}$ :

- pour tout $(f,t) \in [\![1,F]\!] \times [\![1,T]\!]$, on pose $V_{ft} = 0$,
- répéter $J$ fois :
  - tirer $t$ selon $P_\Lambda(t)$,
  - tirer $\eta = (n^1, n^2, \cdots n^T)$ selon $P_\Lambda(\eta) = P_\Lambda(n^1, n^2, \cdots n^T)$,
  - tirer $f$ selon $P_\Lambda(f|n^1, n^2, \cdots n^T, t)$,
  - poser $V_{ft} = V_{ft} + 1$.

Le réseau bayésien équivalent pour un tirage $j$ peut être visualisé sur la figure 5.1. L'ensemble des variables désignées par $n_t$ sont latentes alors que les variables désignées par $f$ et $t$ sont observées via $V_{ft}$. On note

$$\eta_j = \left(n_j^1, n_j^2, \cdots n_j^T\right)$$

l'ensemble des variables cachées au tirage $j \in [\![1,J]\!]$, et

$$\bar{\eta} = (\eta_1, \eta_2, \cdots \eta_J)$$

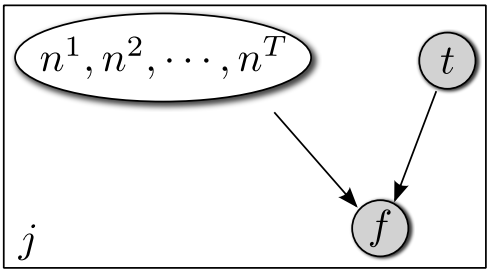



**Figure 5.1** – Modèle graphique du modèle LCATS au $j^{\text{ème}}$ tirage. Les variables grisées sont observées.

l'ensemble de toutes les variables cachées. $\bar{f} = (f_1, \cdots, f_J)$ et $\bar{t} = (t_1, \cdots, t_J)$ représentent quant à eux l'ensemble de toutes les variables observées. Enfin, $\Lambda$ désigne l'ensemble des paramètres du modèle, que pour l'instant nous n'explicitons pas. De même que pour n'importe quel modèle fondé sur la PLCA, la log-vraisemblance des données en fonction des paramètres est donnée par

$$\mathcal{L}_\Lambda(\bar{f}, \bar{t}) = \sum_{f,t} V_{ft} \ln\left(P_\Lambda(f,t)\right). \tag{5.2}$$

Ici, $P_\Lambda(f,t)$ est donné par :

$$P(f,t) = \sum_{n^1,\cdots,n^T} P_\Lambda(t) P_\Lambda(n^1, \cdots n^T) P_\Lambda(f|n^1, \cdots n^T, t). \tag{5.3}$$

Pour éviter la surcharge d'indices et d'exposants, nous omettons délibérément désormais de préciser (par l'indice $\Lambda$) que les probabilités dépendent des paramètres du modèle.

### 5.2.2 Algorithme EM

Ici encore, l'algorithme EM nous permet de définir des mises à jour afin de trouver un maximum local de la log-vraisemblance. Calculons en premier lieu la log-probabilité des variables cachées et observées :

$$\begin{aligned}\mathcal{L}(\bar{f}, \bar{t}, \bar{\eta}) &= \ln\left(\prod_j P(\eta_j, f_j, t_j)\right) \\ &= \sum_j \ln\left(P(t_j)\right) + \ln\left(P(\eta_j)\right) + \ln\left(P(f_j|\eta_j, t_j)\right).\end{aligned}$$

Lors de l'étape E, l'espérance conditionnelle de cette log-probabilité sachant les observations et les paramètres du modèle est calculée :

$$\begin{aligned} Q_\Lambda &= \mathbb{E}\left[\mathcal{L}(\bar{f},\bar{t},\bar{\eta})|\bar{f},\bar{t},\Lambda\right] \\ &= \sum_{\bar{\eta}} P(\bar{\eta}|\bar{f},\bar{t})\mathcal{L}(\bar{f},\bar{t},\bar{\eta}) \\ &= \sum_j \sum_{\eta_j} P(\eta_j|f_j,t_j)\left[\ln\left(P(t_j)\right)+\ln\left(P(\eta_j)\right)+\ln\left(P(f_j|\eta_j,t_j)\right)\right]. \end{aligned}$$

Dans la dernière étape de l'équation précédente, de nombreuses variables cachées ont été marginalisées et l'on a utilisé le fait que $\eta_j$ dépend uniquement des observations $f_j$ et $t_j$. Si maintenant on somme sur les points temps-fréquences plutôt que sur les tirages et si l'on renomme la variable muette $\eta_j$ en $\eta$, alors on obtient :

$$Q_\Lambda = \sum_{f,t}\sum_{\eta} V_{ft}P(\eta|f,t)\left[\ln\left(P(t)\right)+\ln\left(P(\eta)\right)+\ln\left(P(f|\eta,t)\right)\right]. \tag{5.4}$$

Les probabilités *a posteriori* $P(\eta|f,t)$ sont calculées grâce à la règle de Bayes :

$$P(\eta|f,t) = \frac{P(t)P(\eta)P(f|\eta,t)}{\sum_{\eta'} P(t)P(\eta\prime)P(f|\eta',t)} = \frac{P(t)P(\eta)P(f|\eta,t)}{P(f,t)}. \tag{5.5}$$

Lors de l'étape M, $Q_\Lambda$ est maximisée par rapport à chaque élément, sous la contrainte que les probabilités somment à 1.

Nous pouvons désormais proposer différents modèles pour $P(\eta)$ et $P(f|\eta,t)$.

## 5.3 Le cas indépendant

### 5.3.1 Le modèle

Le premier modèle que nous présentons est le cas où les variables cachées au tirage $j$ sont totalement indépendantes, et où la variable $f$ quand elle est conditionnée par $t$ ne dépend que de $n^t$. Le modèle est alors le suivant :

$$P(n^1,n^2,\cdots n^T) = \prod_\tau P^\tau(n^\tau), \tag{5.6}$$

$$P(f|n^1,n^2,\cdots n^T,t) = P(f|n^t). \tag{5.7}$$

L'ensemble des paramètres est alors défini par :

$$\Lambda = \left\{P(t),P^1(n),\cdots P^T(n),P(f|n)\right\}. \tag{5.8}$$

L'exposant $t$ (resp. $\tau$) quand il est utilisé dans les expressions du type $P^t(.)$ (resp. $P^\tau(.)$) signifie que la distribution de probabilité dont il est question dépend du temps $t$ (resp. $\tau$). Même si cela peut porter à confusion, il faut bien comprendre que quand $t$ ou $\tau$ sont utilisés en exposant,

ils ne représentent plus des variables aléatoires, mais juste des indices temporels. Ce modèle ne nous intéresse pas en tant que tel puisque l'on n'introduit aucune structure temporelle, mais il est intéressant de savoir si l'on se retrouve avec le modèle classique de PLCA, et ainsi pouvoir affirmer ou non que le modèle LCATS est une généralisation de la PLCA.

### 5.3.2 Dérivation de l'algorithme EM

Avec ce modèle, l'équation (5.4), pour peu qu'on marginalise au maximum les variables cachées, devient :

$$\begin{aligned} Q_\Lambda = &\sum_{f,t} V_{ft} \ln\left(P(t)\right) \\ &+ \sum_{f,t,\tau} \sum_{n^\tau} V_{ft} P(n^\tau|f,t) \ln\left(P^\tau(n^\tau)\right) \\ &+ \sum_{f,t} \sum_{n^t} V_{ft} P(n^t|f,t) \ln\Big(P(f|n^t)\Big), \end{aligned}$$

ou, en interchangeant les variables muettes $t$ et $\tau$ dans le second terme :

$$\begin{aligned} Q_\Lambda = &\sum_{f,t} V_{ft} \ln\left(P(t)\right) \\ &+ \sum_{f,t,\tau} \sum_{n^\tau} V_{f\tau} P(n^t|f,\tau) \ln\Big(P^t(n^t)\Big) \\ &+ \sum_{f,t} \sum_{n^t} V_{ft} P(n^t|f,t) \ln\Big(P(f|n^t)\Big). \end{aligned} \tag{5.9}$$

Les probabilités *a posteriori* sont données par :

$$P(n^t|f,\tau) = P^t(n^t) \quad \text{si} \quad \tau \neq t, \tag{5.10}$$

$$P(n^t|f,t) = \frac{P(t)P^t(n^t)P(f|n^t)}{P(f,t)}, \tag{5.11}$$

avec, en fusionnant l'équation (5.3) et le modèle courant,

$$P(f,t) = P(t) \sum_{n^t} P^t(n^t) P(f|n^t). \tag{5.12}$$

Lors de l'étape M, on cherche à maximiser $Q_\Lambda$ en fonction des paramètres, sous la contrainte que les probabilités somment à 1. Cela donne les mises à jour suivantes :

$$\begin{aligned} P(t) &\propto \sum_f V_{ft}, \\ P^t(n^t) &\propto \sum_{\tau,f} V_{f\tau} P(n^t|f,\tau), \end{aligned}$$

$$P(f|n^t) \propto \sum_t V_{ft} P(n^t|f,\tau),$$

ou encore

$$P(t) \propto \sum_f V_{ft}, \tag{5.13}$$

$$P^t(n) \propto P^t(n) \left( \sum_{\tau \neq t} V_{f\tau} + \sum_f \frac{V_{ft}}{P(f,t)} P(t) P(f|n) \right), \tag{5.14}$$

$$P(f|n) \propto P(f|n) \sum_t \frac{V_{ft}}{P(f,t)} P(t) P^t(n). \tag{5.15}$$

Par comparaison, on rappelle que le modèle équivalent de la PLCA classique et les mises à jour de ses paramètres sont respectivement donnés par l'équation (2.14) et les équations (2.15), (2.16) et (2.17) (page 33). On se retrouve donc avec quasiment les mêmes mises à jour : la seule différence réside dans terme $\sum_{\tau \neq t} V_{f\tau}$ de l'équation (5.14). On ne peut ainsi pas vraiment affirmer que le modèle LCATS est une extension de la PLCA. En réalité, ce terme additionnel est à comprendre à la lumière du chapitre 4 : il agit comme un frein à la convergence de $P^t(n)$, dû au fait qu'à chaque tirage $j$, l'ensemble des variables cachées $\left\{n_j^\tau\right\}_{\tau \neq t_j}$ n'est lié à aucune observation [1].

## 5.4 Le cas markovien : MLCATS

### 5.4.1 Le modèle

Dans la version markovienne du modèle LCATS (MLCATS pour *Markovian-LCATS)*, nous ne modélisons plus directement les activations temporelle $P^t(n)$ mais les probabilités de transition d'un atome à l'autre entre les temps $t-1$ et $t$ :

$$P(n^1, n^2, \cdots n^T) = P^1(n^1) \prod_\tau P^\tau(n^\tau|n^{\tau-1}), \tag{5.16}$$

$$P(f|n^1, n^2, \cdots n^T, t) = P(f|n^t). \tag{5.17}$$

L'ensemble des paramètres est alors défini comme :

$$\Lambda = \left\{ P^1(n^1), P^2(n^2|n^1), \cdots, P^T(n^T|n^{T-1}), P(f|z) \right\}. \tag{5.18}$$

Les paramètres $\left\{P^t(n^t|n^{t-1})\right\}_t$ sont appelés probabilités de transition. Le modèle graphique du réseau bayésien correspondant pour le tirage $j$ est montré sur la figure 5.2. Pour le calcul du modèle d'observation $P(f,t)$ (équation (5.3)), on peut aisément prouver qu'il est possible de

1. Pour l'anecdote, c'est en étudiant ce cas de figure que l'idée de la maitrise de la vitesse de convergence du chapitre 4 nous est venue.

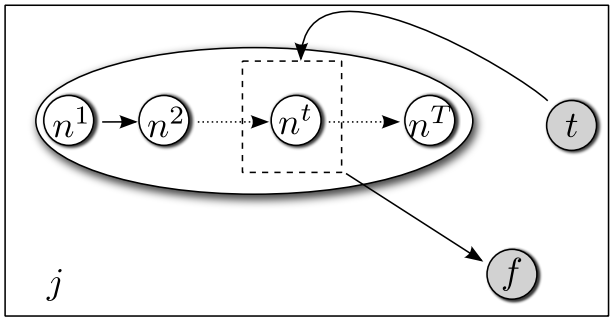



**Figure 5.2** – Modèle graphique pour le modèle MLCATS, à la $j^{\text{ème}}$ itération. Les variables grisées sont observées.

procéder de la manière suivante :

$$P(f,t) = P(t) \sum_{n^t} P^t(n^t) P(f|n^t) \tag{5.19}$$

où $P^t(n^t)$ peut se calculer de manière itérative :

$$\forall t > 1, \quad P^t(n^t) = \sum_{n^{t-1}} P^{t-1}(n^{t-1}) P^t(n^t|n^{t-1}). \tag{5.20}$$

A chaque temps, la probabilité $P^t(n)$ correspond à l'énergie relative des différents atomes $n$. Si l'on souhaite obtenir leurs activations temporelles, il suffit de calculer :

$$P(n,t) = P(t) P^t(n). \tag{5.21}$$

### 5.4.2 Dérivation de l'algorithme EM

Dans cette section, nous détaillons les calculs pour la dérivation de l'algorithme EM. Ils sont assez difficiles à suivre, et le lecteur pourra s'il le souhaite, directement se reporter au résumé des mises à jour dans la section suivante (section 5.4.3).

Avec le modèle ainsi défini, l'équation (5.4), en marginalisant selon le maximum de variables cachées, devient :

$$\begin{aligned} Q_\Lambda = & \sum_{f,t} V_{ft} \ln\left(P(t)\right) \\ & + \sum_{f,t} \sum_{n^1} V_{ft} P(n^1|f,t) \ln\left(P^1(n^1)\right) + \sum_{f,t,\tau>1} \sum_{n^{\tau-1},n^\tau} V_{ft} P(n^{\tau-1},n^\tau|f,t) \ln\left(P^\tau(n^\tau|n^{\tau-1})\right) \\ & + \sum_{f,t} \sum_{n^t} V_{ft} P(n^t|f,t) \ln\left(P(f|n^t)\right), \end{aligned}$$

ou encore, en interchangeant les variables muettes $t$ et $\tau$ dans le deuxième terme de cette équa-

tion, et en réorganisant les sommes,

$$\begin{aligned}
Q_\Lambda = & \sum_{f,t} V_{ft} \ln\left(P(t)\right) \\
& + \sum_{n^1} \sum_{f,\tau} V_{f\tau} P(n^1|f,\tau) \ln\left(P^1(n^1)\right) \\
& + \sum_{n^{t-1},n^t,t>1} \sum_{f,\tau<t} V_{f\tau} P(n^{t-1},n^t|f,\tau) \ln\left(P^t(n^t|n^{t-1})\right) \\
& + \sum_{n^{t-1},n^t,t>1} \sum_{f,\tau\geq t} V_{f\tau} P(n^{t-1},n^t|f,\tau) \ln\left(P^t(n^t|n^{t-1})\right) \\
& + \sum_{n^t,f} \sum_{t} V_{ft} P(n^t|f,t) \ln\left(P(f|n^t)\right). \qquad (5.22)
\end{aligned}$$

Nous pouvons maintenant décrire le moyen de calculer les probabilités *a posteriori* de manière efficace (de complexité linéaire en la taille des données).

**Calcul de $\sum_{f,\tau<t} V_{f\tau} P(n^{t-1},n^t|f,\tau)$ pour $t>1$ :**

$$\begin{aligned}
\sum_{f,\tau<t} V_{f\tau} P(n^{t-1},n^t|f,\tau) &= \sum_{\tau=1}^{t-1} \sum_{f} V_{f\tau} P(n^{t-1}|f,\tau) P^t(n^t|n^{t-1}) \\
&= P^t(n^t|n^{t-1}) R^{t-1}(n^{t-1}), \qquad (5.23)
\end{aligned}$$

où $R^{t-1}(n^{t-1}) = \sum_{\tau=1}^{t-1} \sum_f V_{f\tau} P(n^{t-1}|f,\tau)$ peut se calculer récursivement en fonction des paramètres et des observations :

$$\begin{aligned}
R^1(n^1) &= \sum_f V_{f1} P(n^1|f,\tau=1) \\
&= \sum_f \frac{V_{f1}}{P(f,\tau=1)} P(\tau=1) P^1(n^1) P(f|n^1) \\
&= P^1(n^1) T^1(n^1), \qquad (5.24)
\end{aligned}$$

et, pour $t>1$,

$$\begin{aligned}
R^t(n^t) &= \sum_{\tau=1}^{t} \sum_f V_{f\tau} P(n^t|f,\tau) \\
&= \sum_f V_{ft} P(n^t|f,t) + \sum_{\tau=1}^{t-1} \sum_f V_{f\tau} P(n^t|f,\tau) \\
&= \sum_f \frac{V_{ft}}{P(f,t)} P(t) P^t(n^t) P(f|n^t) + \sum_{n^{t-1}} \sum_{\tau=1}^{t-1} \sum_f V_{f\tau} P(n^{t-1}|f,\tau) P^t(n^t|n^{t-1})
\end{aligned}$$

soit

$$R^t(n^t) = P^t(n^t)T^t(n^t) + \sum_{n^{t-1}} P^t(n^t|n^{t-1})R^{t-1}(n^{t-1}). \tag{5.25}$$

Dans les équations (5.24) et (5.25), on a posé :

$$T^t(n^t) = \sum_f \frac{V_{ft}}{P(f,t)} P(t)P(f|n^t). \tag{5.26}$$

**Calcul de $\sum_{f,\tau\geq t} V_{f\tau}P(n^{t-1}, n^t|f,\tau)$ :**

$$\sum_{f,\tau\geq t} V_{f\tau}P(n^{t-1}, n^t|f,\tau) = \sum_{\tau=t}^{T}\sum_f \frac{V_{f\tau}}{P(f,\tau)} P(\tau)P^{t-1}(n^{t-1})P^t(n^t|n^{t-1})P(f|n^t,\tau) \tag{5.27}$$

$$= P^{t-1}(n^{t-1})P^t(n^t|n^{t-1})S^t(n^t), \tag{5.28}$$

où $S^t(n^t) = \sum_{\tau=t}^{T}\sum_f \frac{V_{f\tau}}{P(f,\tau)} P(\tau)P(f|n^t,\tau)$ peut se calculer récursivement de la manière suivante :

$$\begin{aligned} S^T(n^T) &= \sum_f \frac{V_{fT}}{P(f,T)} P(T)P(f|n^T) \\ &= T^T(n^T) \end{aligned} \tag{5.29}$$

et, pour $t < T$,

$$\begin{aligned} S^t(n^t) &= \sum_{\tau=t}^{T}\sum_f \frac{V_{f\tau}}{P(f,\tau)} P(\tau)P(f|n^t,\tau) \\ &= \sum_f \frac{V_{ft}}{P(f,t)} P(t)P(f|n^t) + \sum_{\tau=t+1}^{T}\sum_f \frac{V_{f\tau}}{P(f,\tau)} P(\tau)P(f|n^t,\tau) \\ &= T^t(n^t) + \sum_{n^{t+1}} P^{t+1}(n^{t+1}|n^t) \sum_{\tau=t+1}^{T}\sum_f \frac{V_{f\tau}}{P(f,\tau)} P(\tau)P(f|n^{t+1},\tau) \\ &= T^t(n^t) + \sum_{n^{t+1}} P^{t+1}(n^{t+1}|n^t)S^{t+1}(n^{t+1}). \end{aligned} \tag{5.30}$$

**Calcul de $\sum_{f,\tau} V_{f\tau}P(n^1|f,\tau)$ :**

$$\begin{aligned} \sum_{f,\tau} V_{f\tau}P(n^1|f,\tau) &= \sum_{f,\tau} \frac{V_{f\tau}}{P(f,\tau)} P(\tau)P^1(n^1)P(f|n^1,\tau) \\ &= P^1(n^1)S^1(n^1) \end{aligned} \tag{5.31}$$

**Calcul de $\sum_t V_{ft} P(n^t|f,t)$ :**

$$\sum_t V_{ft} P(n^t|f,t) = \sum_t \frac{V_{ft}}{P(f,t)} P(t) P^t(n^t) P(f|n^t). \tag{5.32}$$

Lors de l'étape M, $Q_\Lambda$ est maximisée par rapport aux paramètres, sous la contrainte que les probabilités somment à un. Les mises à jours obtenues sont les suivantes :

$$P^1(n^1) \propto \sum_{\tau,f} V_{f\tau} P(n^1|f,\tau) \tag{5.33}$$

$$P^t(n^t|n^{t-1}) \propto \sum_{f,\tau<t} V_{f\tau} P(n^{t-1},n^t|f,\tau) + \sum_{f,\tau\geq t} V_{f\tau} P(n^{t-1},n^t|f,\tau) \tag{5.34}$$

$$P(f|n^t) \propto \sum_t V_{ft} P(n^t|f,t). \tag{5.35}$$

### 5.4.3 Résumé des mises à jour pour l'estimation des paramètres du modèle MLCATS

Nous résumons dans l'Algorithme 6 l'ensemble des mises à jour pour l'estimation des paramètres du modèle en fonction des observations. Comme nous le verrons lors des tests préliminaires à la fin de ce chapitre, la convergence de $P^t(n^t|n^{t-1})$ s'avère très lente dans la pratique, et il peut être utile par conséquent de ralentir également la convergence de $P(f|n)$ afin d'encourager la parcimonie des activations temporelles des atomes (*cf.* chapitre 4). Nous incluons donc dans ce résumé un coefficient $\beta_{\text{frein}}$ dans la mise à jour de ces derniers paramètres, selon l'astuce proposée au chapitre (4).

## 5.5 Combinaison de modèles

La différence entre la PLCA classique et la MLCATS réside dans la modélisation des activations temporelles : le modèle des atomes de base $P(f|n)$ est resté inchangé (*cf.* équation (5.17)). En réalité, on peut imaginer d'autres modèles pour $P(f|n^1,\cdots,n^T,t)$ et en particulier des modèles de type

$$P(f|n^1,\cdots,n^T,t) = P(f|n^t,t) \tag{5.47}$$

où $P(f|n^t,t)$ peut lui même être décomposé selon un modèle défini. Une autre manière de voir les choses est que l'on peut substituer le modèle MLCATS à n'importe quelle distribution bidimensionnelle d'un modèle de type PLCA. Par exemple, si $P(f,t)$ se décompose comme

$$P(f,t) = \sum_n P(n,t) P(f|n,t), \tag{5.48}$$

où, $P(f|n,t)$ suit un modèle quelconque, alors $P(n,t)$ peut être remplacée par le modèle MLCATS.

**Algorithme 6:** Résumé des mises à jours des paramètres pour le modèle MLCATS.

**répéter**

**Étape E**

$$\forall t > 1, \quad P^t(n^t) = \sum_{n^{t-1}} P^{t-1}(n^{t-1}) P^t(n^t|n^{t-1}) \tag{5.36}$$

$$P(f,t) = P(t) \sum_n P^t(n) P(f|n) \tag{5.37}$$

$$T^t(n^t) = P(t) \sum_f \frac{V_{ft}}{P(f,t)} P(f|n^t) \tag{5.38}$$

$$S^T(n^T) = T^T(n^T) \tag{5.39}$$

$$\forall t < T, \quad S^t(n^t) = T^t(n^t) + \sum_{n^{t+1}} P^{t+1}(n^{t+1}|n^t) S^{t+1}(n^{t+1}) \tag{5.40}$$

$$R^1(n^1) = P^1(n^1) T^1(n^1) \tag{5.41}$$

$$\forall t > 1, \quad R^t(n^t) = P^t(n^t) T^t(n^t) + \sum_{n^{t-1}} P^t(n^t|n^{t-1}) R^{t-1}(n^{t-1}) \tag{5.42}$$

**Étape M**

$$P(t) \propto \sum_f V_{ft} \tag{5.43}$$

$$P^1(n^1) \propto P^1(n^1) S^1(n^1) \tag{5.44}$$

$$P^t(n^t|n^{t-1}) \propto P^t(n^t|n^{t-1}) \left( R^{t-1}(n^{t-1}) + P^{t-1}(n^{t-1}) S^t(n^t) \right) \tag{5.45}$$

$$P(f|n) \propto P(f|n) \sum_t \left( \frac{V_{ft}}{P(f,t)} P(t) P^t(n) + \beta_{\text{frein}} \right) \tag{5.46}$$

où $\beta_{\text{frein}} \geq 0$ est un coefficient de freinage pour $P(f|n)$.

**jusqu'à** *convergence*;

## 5.6 MLCATS : expérience

Le modèle MLCATS que nous venons de présenter est un nouveau cadre permettant de modéliser les probabilités de transition pour les variables cachées $n$ entre les temps $t-1$ et $t$ plutôt que directement leur probabilité au temps $t$. L'intérêt d'un tel modèle est que l'on a désormais un contrôle direct sur la manière dont les activations des atomes évoluent. À titre d'exemple, on peut proposer deux scénarios d'application, suivant la manière d'initialiser les probabilités de transition $\{P^t(n^t|n^{t-1})\}_{t>1}$ (que l'on peut considérer comme un ensemble de matrices carrées) :

- initialiser ou contraindre chaque matrice $P^t(n^t|n^{t-1})$ (pour chaque $t$) de sorte que les valeurs de sa diagonale soient fortes permet d'encourager la régularité des probabilités $P^t(n)$ au fil du temps ;
- initialiser chaque matrice $P^t(n^t|n^{t-1})$ de sorte que ses valeurs soient nulles (ou au moins faibles) quand $n^t$ est trop éloigné de $n^{t-1}$ (c'est-à-dire les coefficients loin de la diagonale) peut permettre de considérer la proximité de deux notes consécutives dans une ligne mélodique, caractère souvent rencontré dans la musique.

Dans l'expérience que nous menons ici, nous nous en tenons au premier de ces deux scénarios. La CQT d'entrée à analyser correspond à un extrait d'enregistrement de piano, et nous comparons les estimations données avec trois systèmes différents : la PLCA classique, la MLCATS, et la MLCATS avec freinage des spectres de base $P(f|n)$. Dans les trois cas on utilise 88 spectres harmoniques (un pour chaque note) et 10 spectres fixes pour modéliser le bruit. Ces spectres sont initialisés de la même manière que pour l'exemple d'application de l'apriori de parcimonie (figure 3.2 page 47). Pour la PLCA, les activations temps-fréquence $P(n,t)$ sont initialisées uniformément. Pour la MLCATS, les probabilités de transition sont initialisées de manière un peu plus complexe :

- les valeurs de la diagonale sont fortes pour les atomes harmoniques : cela permet d'encourager la continuité temporelle des activations de ces atomes ;
- les valeurs des transitions entre atomes harmonique/bruit, bruit/harmonique, bruit/bruit sont également assez fortes pour permettre les échanges d'énergie entre les notes harmoniques et le bruit ;
- toutes les autres valeurs sont assez faibles.

On peut voir une illustration de cette initialisation de probabilités de transition sur la figure 5.3. Sur la figure 5.4, on peut comparer les résultats obtenus pour chacun des trois systèmes considérés. On peut d'abord remarquer que la MLCATS permet d'obtenir des probabilités $P^t(n)$ assez régulières au cours du temps. Par conséquent, en raison de la relative régularité de la puissance du signal $P(t)$, les activations temporelles, obtenues grâce à l'équation (5.21), sont au final plus régulières que celles de la PLCA classique. Avec la MLCATS sans frein, on remarque que les activations temporelles obtenues sont particulièrement non-parcimonieuses. En fait, la convergence des paramètres $\{P^t(n^t|n^{t-1})\}_{t>1}$ s'avère être assez lente et par conséquent

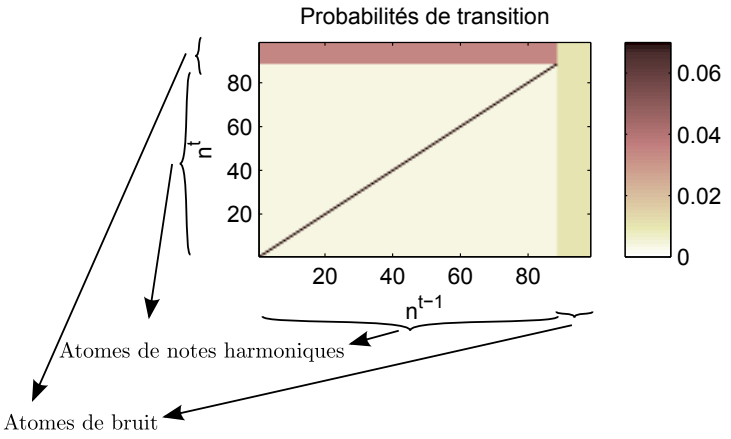
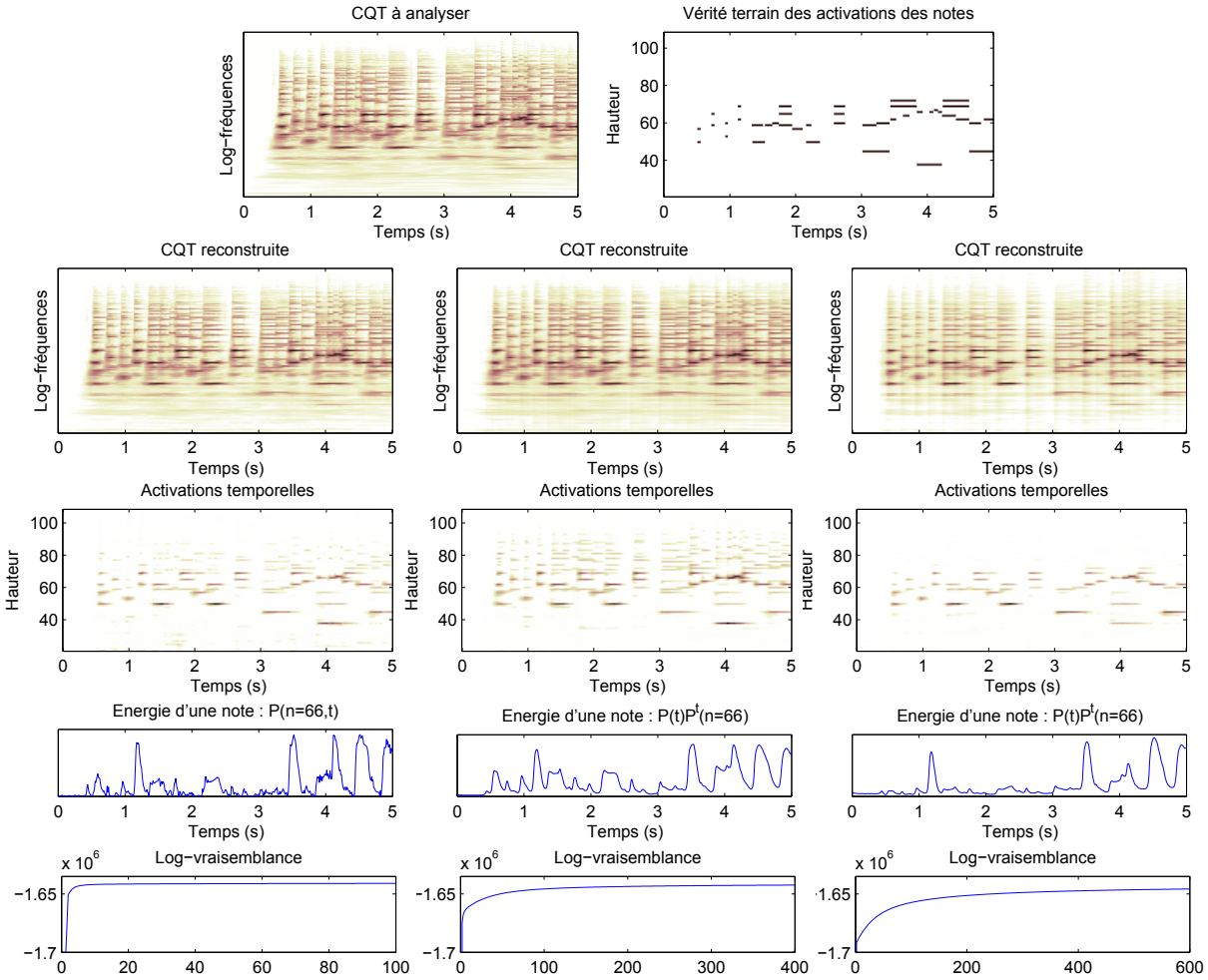



**Figure 5.3** – Initialisation des probabilités de transition $P^t(n^t|n^{t-1})$. Elle est identique pour chaque $t$.

la parcimonie intrinsèque aux données à analyser se retrouve dans les paramètres $P(f|n)$. Ralentir la convergence de ces derniers paramètres (avec l'utilisation d'un coefficient de freinage), permet de contre-balancer ce phénomène, et l'on observe alors des activations nettement plus parcimonieuses et proches de la vérité terrain. En revanche cela ralentit encore plus la vitesse de convergence de la log-vraisemblance.

## 5.7 Conclusion et discussion

Dans ce chapitre, nous avons introduit un nouveau méta-modèle, appelé LCATS, permettant d'introduire une structure temporelle pour les activations des atomes dans un modèle de type PLCA. La première déclinaison à laquelle nous nous sommes intéressés, le cas indépendant, nous a permis de conclure que la LCATS n'était pas tout à fait une généralisation de la PLCA. Nous avons également décliné ce modèle dans un cas markovien (modèle MLCATS), où plutôt que de modéliser directement la probabilité des variables cachées à chaque temps $t$, nous avons modélisé les probabilités de transition entre les temps $t-1$ et $t$. Nous avons proposé un premier exemple d'application, consistant à initialiser ces probabilités de transition avec de fortes valeurs sur leur diagonale et permettant de constater qu'ainsi, les activations temporelles des atomes après la décomposition d'une CQT d'entrée avaient tendance à être plus régulières au cours du temps. Ceci étant dit, le modèle MLCATS souffre d'un problème que nous n'avons pas encore discuté : sa surparamétrisation. En effet, alors que son pouvoir d'expression n'est pas plus grand que celui de la PLCA, le nombre de ses paramètres est multiplié par le nombre $N$ d'atomes. Ce n'est pas un problème en soi tant que l'on ne s'intéresse qu'aux activations temporelles $P(n,t)$ que l'on peut déduire des paramètres (qu'importe s'il existe plusieurs jeux de paramètres qui donnent les mêmes activations temporelles), mais si pour une application donnée, l'estimation des probabilités de transition nous intéresse vraiment, la MLCATS ne possèdera pas le caractère nécessaire d'identifiabilité. Nous envisageons donc de trouver, dans un travail futur, un moyen




(a) PLCA (b) MLCATS (c) MLCATS + Frein

**Figure 5.4** – Activations temporelles des notes obtenues avec la MLCATS et la PLCA ainsi que la log-vraisemblance au fil des itérations. Le signal d'entrée correspond à un extrait de 5 secondes de la Suite bergamasque (Menuet) de C. Debussy, joué au piano. Les activations des atomes fixes de bruit ne sont pas illustrées ici.

de réduire l'espace dans lequel ces matrices de transition peuvent reposer.

# Troisième partie

# Les modèles de RTF

# Chapitre 6

# Modèle par source : HALCA

## 6.1 Introduction

Dans la précédente partie de ce mémoire, nous avons introduit des moyens de mieux contrôler et d'améliorer la décomposition de données d'entrée par la PLCA. Mais nous avons toujours gardé le modèle de base consistant à décomposer chaque colonne de $\text{RTF}^+$ comme une somme pondérée de spectres de base. Pour une application à la transcription, les spectres de base sont censés représenter des spectres harmoniques de notes. En observant les activations de chaque spectre, on peut donc déduire à chaque temps l'ensemble des notes présentes dans le signal. Malheureusement, comme nous avons pu nous en apercevoir en observant une RTF d'une note réelle de trompette dans la section 1.2 page 16, il est peu réaliste de représenter le spectre d'une note de musique par un unique spectre de base, puisqu'une note peut, par exemple, présenter des variations de fréquence fondamentale ou d'enveloppe spectrale. De plus, en l'absence de contraintes, rien ne nous assure que les atomes estimés à la fin d'un algorithme de décomposition seront harmoniques et représenteront ainsi des notes. On peut alors imaginer des nouveaux modèles de $\text{RTF}^+$ permettant de mieux prendre en compte les caractéristiques des signaux musicaux : c'est le but des chapitres de cette troisième partie. En particulier, l'ensemble des modèles de CQT que nous allons présenter dans les chapitres 6 et 7 ont été imaginés dans le but d'analyser les notes harmoniques possédant les deux types de variations que nous venons de mentionner. De plus, ils ont été pensés pour être robuste au bruit.

Dans ce chapitre, nous introduisons un premier modèle, appelé Analyse harmonique et adaptative en composantes latentes (HALCA pour *Harmonic Adaptive Latent Component Analysis*). Nous le caractérisons comme un modèle par source, puisqu'ici on va considérer un signal comme un mélange de signaux provenant de différents instruments de musique, monodiques ou polyphoniques. Nous commencerons par présenter le modèle HALCA et décrirons l'algorithme permettant d'estimer ses paramètres. Nous expliquerons ensuite comment initialiser les paramètres et comment appliquer les outils développés dans les chapitres 3 et 4 pour améliorer notre modèle. Enfin, nous montrerons quelques exemples de décompositions et discuterons le comportement

de l'algorithme, pour finir avec une évaluation préliminaire de ses performances dans une tâche d'estimation de hauteurs multiples. Ce modèle a été introduit dans [FBR11a, FBR11b] et plus amplement étudié dans [FBR13].

## 6.2 Présentation du modèle

Introduisons en premier lieu une première variable cachée $c$ afin de décomposer la CQT $V_{ft}$ d'un signal de musique comme la somme d'un signal polyphonique harmonique (dans ce cas $c = h$ pour *harmonique*) et d'un signal de bruit (alors $c = b$ pour *bruit*) (les notations $P_h(.)$ et $P_b(.)$ sont utilisés pour $P(.|c = h)$ et $P(.|c = b)$) :

$$P(f,t) = P(c = h)P_h(f,t) + P(c = b)P_b(f,t). \tag{6.1}$$

$P_h(f,t)_{(f,t)\in\mathbb{Z}\times[\![1,T]\!]}$et $P_b(f,t)_{(f,t)\in\mathbb{Z}\times[\![1,T]\!]}$ représentent alors respectivement les CQTs du signal polyphonique et du signal de bruit. $P(c)_{c=h,b}$ correspond quant à lui à l'énergie normalisée de chaque signal. On décompose maintenant la partie polyphonique comme la somme de $S$ sources, désignées par la variable aléatoire $s$, chacune d'entre elles représentant un instrument spécifique :

$$P_h(f,t) = \sum_s P_h(f,t,s). \tag{6.2}$$

Chaque colonne de $P_h(f,t,s)$ figure alors le spectre d'une ou plusieurs notes harmoniques, jouées par un instrument. Nous allons maintenant expliquer comment chaque colonne de $P_h(f,t,s)$ et de $P_b(f,t)$ est modélisée.

**Modèle d'instrument.** Nous souhaitons pouvoir prendre en compte la nature non-stationnaire des instruments harmoniques, aussi bien en terme de fréquence fondamentale que d'enveloppe spectrale. À cette fin, nous nous inspirons du modèle introduit dans [VBB10], où un spectre harmonique est décomposé comme une somme pondérée de noyaux harmoniques à bande étroite fixés, partageant la même fréquence fondamentale. Dans le modèle HALCA, pour une source $s$ donnée, les $Z$ noyaux harmoniques notés $P_h(\mu|z)_{(\mu,z)\in[\![1,F]\!]\times[\![1,Z]\!]}$ sont tout d'abord pondérés pour chaque $t$ par les poids $P_h(z|t,s)$, puis sommés. Les spectres résultants sont alors convolués par une distribution d'activations temps-fréquence $P_h(i,t,s)_{(i,t)\in[\![0,I-1]\!]\times[\![1,T]\!]}$ qui définira au temps $t$, pour la source $s$, le nombre de notes ainsi que leur hauteur respective. Le modèle d'une source est alors le suivant :

$$P_h(f,t,s) = \sum_{z,i} P_h(i,t,s)P_h(f-i|z)P_h(z|t,s). \tag{6.3}$$

Les paramètres possèdent les caractéristiques suivantes :

- les noyaux $P_h(\mu|z)_{z\in[\![1,Z]\!]}$ partagent la même fréquence fondamentale de référence, mais ont leur énergie concentrée sur un partiel donné (leur conception sera discutée en détail ultérieurement dans cette section),

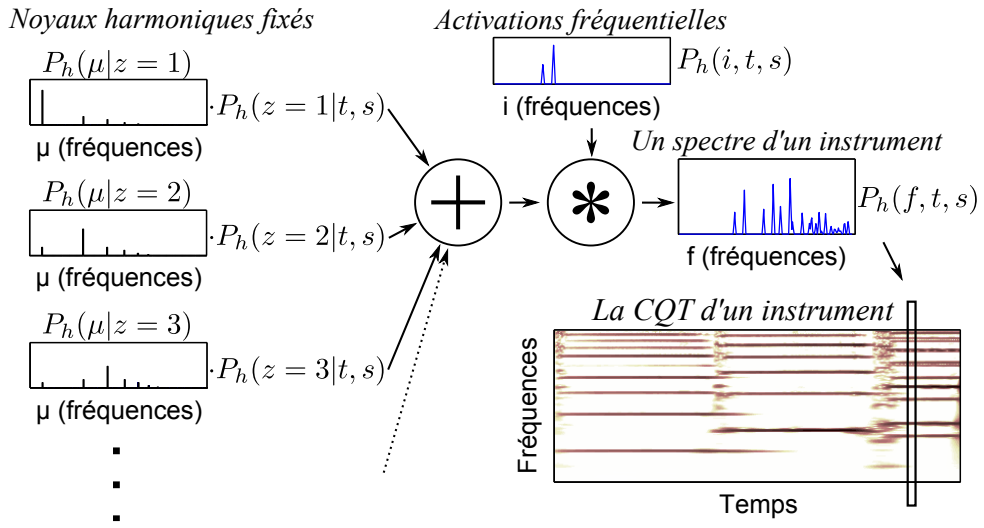



**Figure 6.1** – Modèle de spectre d'une source $s$ au temps $t$.

- les poids $P_h(z|t,s)$ définissent les amplitudes des harmoniques des notes jouées par la source $s$ au temps $t$ : nous les appelons coefficients d'enveloppe,
- chaque colonne des activations temps-fréquence $P_h(i,t,s)_{i\in[\![0,I-1]\!]}$ peut être monomodale ou multimodale, chaque mode correspondant à la hauteur d'une des notes,
- pour s'assurer que le modèle puisse s'adapter à n'importe quel étalement spectral des partiels (par exemple, une variation continue de fréquence fondamentale implique un plus grand étalement à un temps donné), les noyaux contiennent de l'énergie uniquement aux points fréquentiels coïncidant avec les multiples de leur fréquence fondamentale, et l'étalement est ainsi pris en compte dans les activations temps-fréquence.

Dans ce modèle, l'enveloppe spectrale de chaque note peut évoluer au cours du temps. Cependant, à $t$ et $s$ donnés, on est obligé de supposer que les notes jouées simultanément par un unique instrument ont les mêmes amplitudes relatives pour leurs harmoniques. Ce modèle d'instrument est illustré dans la figure 6.1.

**Modèle de bruit.** Afin de prendre en compte la présence de bruit dans une CQT, celui-ci est modélisé comme la convolution d'une fenêtre à bande étroite $P_b(\mu)_{\mu\in[\![1,F]\!]}$ (nous utilisons dans la pratique une fenêtre de Hann) et d'une distribution temps-fréquence de bruit $P_b(i,t)_{(i,t)\in[\![0,I-1]\!]\times[\![1,T]\!]}$ :

$$P_b(f,t) = \sum_i P_b(i,t)P_b(f-i) \tag{6.4}$$

comme illustré sur la figure 6.2. La taille de la fenêtre de Hann détermine le niveau de régularité des spectres de bruit que le modèle peut prendre en compte. Dans la pratique, on fixe arbitrairement le support de la fenêtre à une largeur correspondant à une octave.

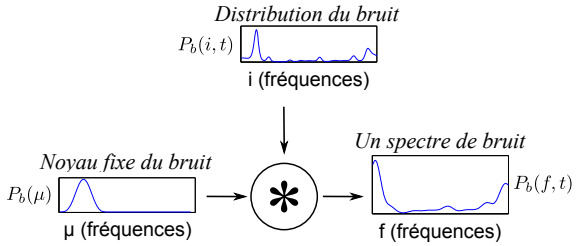



**Figure 6.2** – Modèle de spectre pour le bruit au temps $t$

| Paramètres | Définition sémantique |
|---|---|
| $P(c=h)$ | Énergie relative de la composante polyphonique harmonique |
| $P(c=b)$ | Énergie relative de la composante de bruit |
| $P_h(i,t,s)$ | Activations temps-fréquence de chaque source |
| $P_h(i,t) = \sum_s P_h(i,t,s)$ | Activations temps-fréquence de l'ensemble des sources |
| $P_h(\mu\|z)$ | $z^{\text{ème}}$ noyau harmonique à bande étroite |
| $P_h(z\|t,s)$ | Coefficients d'enveloppe de la source $s$ au temps $t$ |
| $P_b(i,t)$ | Distribution temps-fréquence du bruit |
| $P_b(\mu)$ | Noyaux réguliers à bande étroite du bruit |

**Table 6.1** – Les différents paramètres du modèle HALCA.

**Modèle complet.** En regroupant les équations (6.1),(6.2), (6.3) et (6.4), on peut formuler le modèle HALCA complet :

$$P(f,t) = P(c=h) \sum_{s,i,z} P_h(i,t,s) P_h(f-i|z) P_h(z|t,s) + P(c=b) \sum_i P_b(i,t) P_b(f-i). \quad (6.5)$$

On liste dans la Table 6.1 l'ensemble des paramètres du modèle avec leur signification sémantique.

Nous pouvons également en déduire le processus génératif d'une CQT :

- $\forall (f,t) \in \mathbb{Z} \times [\![1,T]\!]$, on pose $V_{ft} = 0$.
- Répéter $N$ fois :
  - tirer $c$ selon $P(c)_{c=h,b}$,
  - si $c = h$ :
    - tirer $(i,t,s)$ selon $P_h(i,t,s)$
    - tirer $z$ selon $P(z|t,s)$,
    - tirer $\mu$ selon $P_h(\mu|z)$,
    - poser $f = \mu + i$,
    - poser $V_{ft} = V_{ft} + 1$,
  - si $c = b$ :
    - tirer $(i,t)$ selon $P_b(i,t)$,
    - tirer $\mu$ selon $P_b(\mu)$,
    - poser $f = \mu + i$,
    - poser $V_{ft} = V_{ft} + 1$.

De la même manière que pour le modèle SIPLCA (section 2.2), on considère que $V_{ft} = 0$ pour $f \notin [\![1, F]\!]$. $\Lambda = \{P(c), P_h(i,t,s), P_h(z|t,s), P_b(i,t)\}_{i,t,s,z,c}$ représente l'ensemble des paramètres du modèle HALCA à estimer, duquel on pourra déduire l'information utile. Aussi, les activations temps-fréquence des sources $P_h(i,t,s)$, et plus exactement les activations temps-fréquence globales définies par $P_h(i,t) = \sum_s P_h(i,t,s)$, nous permettent d'estimer la hauteur de l'ensemble des notes pour chaque trame de CQT. Il est ainsi possible d'utiliser cet ensemble de paramètres pour une tâche d'estimation de hauteurs multiples ou de transcription automatique, comme nous le verrons dans la section 6.6, ou au chapitre 8. Nous définissons d'ailleurs la qualité d'une décomposition de CQT à la lumière de ces deux applications, puisque nous ne savons pas *a priori* s'il existe une corrélation entre vraisemblance (ou log-probabilité *a posteriori*) des données en fonction des paramètres et performance de l'algorithme dans ces applications.

**Conception des noyaux.** Dans le modèle HALCA, on souhaite modéliser le spectre d'une note harmonique réelle comme la combinaison linéaire de plusieurs vecteurs de base, appelés noyaux. En raison de la nature convolutive du modèle, les noyaux peuvent être conçus indépendamment de la hauteur d'une note, ainsi que de l'étalement spectral de ses partiels. De nombreuses approches sont possibles pour construire ces noyaux. Une première approche serait de les apprendre sur des signaux réels d'instruments isolés comme cela est fait dans [GE11]. Mais une limite de cette approche est que rien n'assure qu'un nouvel instrument, non présent dans la base d'apprentissage, puisse correctement être modélisé. Aussi, nous préférons garder un contrôle sur la conception des noyaux et les paramétrer manuellement. Dans cette optique, une possibilité serait de définir des noyaux pour lesquels l'énergie est concentrée sur un unique partiel. Cette solution serait similaire à l'approche retenue dans [KNS07], où un spectre harmonique est modélisé comme une somme de gaussiennes représentant les partiels. Cela permet de tenir compte d'amplitudes quelconques pour les partiels d'un spectre de note. Cependant, si aucun apriori sur les paramètres n'est ajouté, cela peut facilement amener à des erreurs, de type erreur d'octave, puisqu'une note de fréquence fondamentale $f_0$ pourrait être modélisée par une note de fréquence fondamentale $f_0/2$ pour laquelle toutes les harmoniques impaires sont nulles. Pour éviter ce genre d'erreur, on peut prendre un plus petit nombre de noyaux dans lesquels plusieurs harmoniques adjacentes sont présentes, à l'instar de [VBB10]. Mais l'utilisation de tels noyaux n'est pas appropriée pour modéliser des notes dont des partiels sont manquants, à l'exemple de la clarinette dont toutes les harmoniques paires sont quasi-nulles.

Un choix qui nous parait plus satisfaisant est obtenu en trouvant un compromis entre ces deux dernières possibilités : le nombre de noyaux est défini comme étant égal au nombre maximum de partiels considérés (typiquement $Z = 16$) et l'essentiel de l'énergie de chaque noyau est concentré sur une harmonique donnée. Le reste de l'énergie est partagé entre quelques harmoniques adjacentes. Le nombre de partiels non-nuls de chaque noyau, ainsi que la puissance relative de son partiel principal sont définis manuellement. Sans préciser l'exacte définition que nous avons arbitrairement donnée à chacun des noyaux (elle peut être trouvée dans le code Mat-

lab mis en ligne [Fue13]), nous pouvons préciser que l'on a fait appel à la fenêtre de Hamming afin de s'assurer que les partiels du spectre harmonique résultant de la somme de tous les noyaux aient tous la même valeur. On peut se donner une idée de leur forme sur la figure 6.1.

## 6.3 Estimation des paramètres

Étant donnée une CQT d'observation $V_{ft}$ et un ensemble de noyaux fixés $P_h(\mu|z)$ et $P_b(\mu)$, nous souhaitons estimer l'ensemble des paramètres $\Lambda$ qui maximise la vraisemblance des observations (pour le moment, nous n'ajoutons pas d'apriori sur $\Lambda$). L'algorithme EM permet de définir des règles de mise à jour pour les paramètres, de sorte qu'à chaque itération la log-vraisemblance $L_\Lambda(\bar{f}, \bar{t})$, donnée par l'équation (2.1) (page 30) augmente ou reste égale.

Dans le modèle HALCA, les variables $f$ et $t$ sont observées alors que $i$, $s$, $c$ et $z$ sont des variables cachées (la variable $\mu$ est observée via $f$). On peut démontrer que l'espérance conditionnelle de la log-vraisemblance des variables observées et cachées $\ln\left(P(\bar{f}, \bar{t}, \bar{i}, \bar{s}, \bar{c}, \bar{z})\right)$, sachant les observations et les paramètres, est donnée par (le calcul est très similaire à celui détaillé dans la section 2.1 page 29) :

$$\begin{aligned} Q_\Lambda = &\sum_{i,s,z} V_{ft} P(i,s,z,c=h|f,t) \left[\ln\left(P(c=h)\right) + \ln\left(P_h(i,t,s)\right) + \ln\left(P_h(z|t,s)\right) + \ln\left(P_h(f-i|z)\right)\right] \\ &+ \sum_{i} V_{ft} P(i,c=b|f,t) \left[\ln\left(P(c=b)\right) + \ln\left(P_b(i,t)\right) + \ln\left(P_b(f-i)\right)\right]. \end{aligned} \tag{6.6}$$

Lors de l'étape E, les probabilités *a posteriori* des variables cachées sont calculées grâce au théorème de Bayes :

$$P(i,s,z,c=h|f,t) = \frac{P(c=h)P_h(i,t,s)P_h(f-i|z)P_h(z|t,s)}{P(f,t)}, \tag{6.7}$$

$$P(i,c=b|f,t) = \frac{P(c=b)P_b(i,t)P_b(f-i)}{P(f,t)}, \tag{6.8}$$

$P(f,t)$ étant définie à l'équation (6.5).

Puis, dans l'étape M, $Q_\Lambda$ est maximisée par rapport à $\Lambda$ sous la contrainte que les probabilités somment à un. Cela amène aux mises à jour suivantes :

$$P(c=h) \propto \sum_{f,t,i,s,z} V_{ft} P(i,s,z,c=h|f,t), \tag{6.9}$$

$$P_h(i,t,s) \propto \sum_{f,z} V_{ft} P(i,s,z,c=h|f,t), \tag{6.10}$$

$$P_h(z|t,s) \propto \sum_{f,i} V_{ft} P(i,s,z,c=h|f,t), \tag{6.11}$$

$$P(c=b) \propto \sum_{f,t,i} V_{ft} P(i,c=b|f,t), \tag{6.12}$$

$$P_b(i,t) \propto \sum_f V_{ft} P(i, c = b|f, t). \tag{6.13}$$

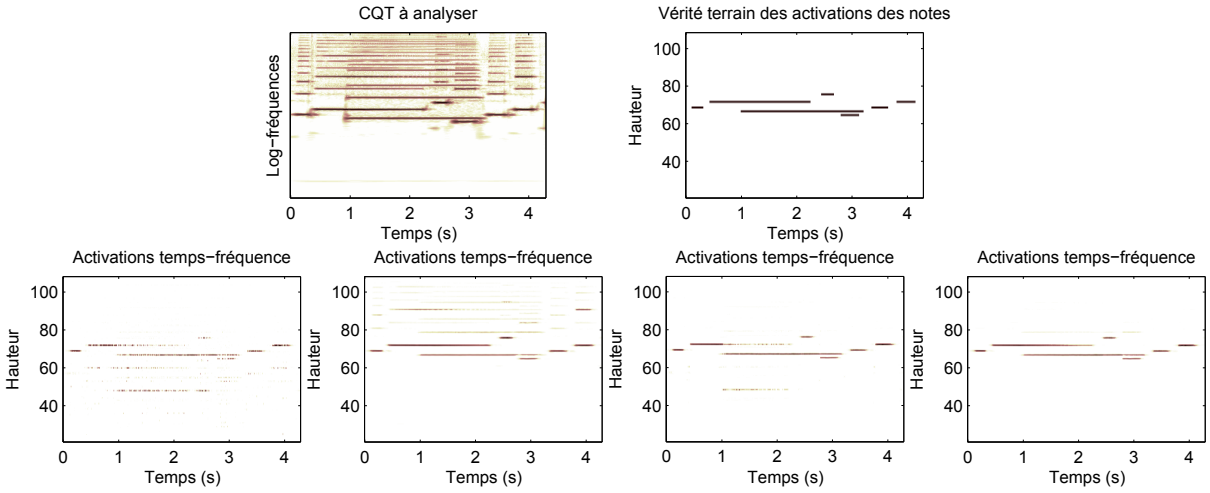
L'algorithme EM consiste alors à initialiser les paramètres, puis à itérer le calcul du modèle (équation (6.5)), l'étape E (équations (6.7) et (6.8)) et l'étape M (équations (6.9) à (6.13), suivies de la normalisation des paramètres), jusqu'à convergence de la fonction de log-vraisemblance. Par extension, nous appelons cet algorithme l'algorithme HALCA.

## 6.4 Initialisation et ajout de modules

Dans cette section, nous nous concentrons sur la manière d'initialiser les paramètres dans la pratique, ainsi que sur les différents modules que l'on peut ajouter au modèle HALCA, à savoir l'ajout d'aprioris ou le freinage de la convergence de certains paramètres lors de l'exécution de l'algorithme.

**Initialisation des paramètres.** Comme pour tous les algorithmes itératifs d'optimisation, l'initialisation des arguments influe sur les résultats obtenus, et donc dans notre cas, sur la qualité de la décomposition. D'une certaine manière, l'initialisation peut être vue comme l'ajout d'une connaissance *a priori* sur la valeur des paramètres, puisque ceux-ci auront de fortes chances de converger vers l'argument d'un optimum local proche de l'initialisation. Ce phénomène s'accentuera si l'on utilise un coefficient de freinage sur un ensemble de paramètres (voir chapitre 4). Bien initialiser les paramètres est donc primordial, comme cela est illustré sur la figure 6.3. Après expérimentations et observations, on peut fournir les recommandations suivantes :

- L'initialisation aléatoire est à proscrire : cela reviendrait à incorporer des aprioris non choisis sur la valeur des paramètres (*cf.* figure 6.3 (a)).
- Afin de rester le plus générique possible, il est conseillé d'initialiser $P_h(i,t,s)$ et $P_b(i,t)$ de manière uniforme : ainsi, aucun apriori sur la distribution des notes ou de l'énergie du bruit n'est fait (on peut mettre quand même à zéro $P_h(i,t,s)$ pour les valeurs de $i$ n'étant pas incluses dans le segment des fréquences fondamentales que l'on suppose possibles).
- L'expérience montre que l'initialisation de $P(c)$ influe très peu sur les résultats : peut-être l'algorithme a-t-il moins tendance à s'engouffrer vers des minima locaux si $P(c = b) > P(c = h)$ à l'initialisation.
- Pour que les paramètres de chaque source prennent des chemins différents, il faut initialiser, pour un temps $t$ donné, les coefficients d'enveloppes $P_h(z|t,s)$ différemment pour chaque source $s$ : on peut par exemple les définir de manière décroissante par rapport à $z$ (on observe souvent dans les instruments de musique une décroissance des énergies des partiels en fonction de la fréquence), avec un coefficient de décroissance différent selon $s$. L'initialisation de ces paramètres est très importante. Si la décroissance est trop forte, cela pourra amener à une sur-évaluation du nombre de notes (une note pour chaque harmonique, *cf.* figure 6.3 (b)), et si elle est trop faible, à des erreurs d'octaves inférieures




(a) Initialisation aléatoire (b) Initialisation trop pentue des coefficients d'enveloppes (c) Initialisation pas assez pentue des coefficients d'enveloppes (d) Bonne initialisation

**Figure 6.3** – Illustration de l'importance de l'initialisation pour le modèle HALCA (avec $S = 2$). Les activations temps-fréquence correspondent à la distribution $P_h(i,t) = \sum_s P_h(i,t,s)$. Le signal d'entrée correspond à quelques notes de saxophone et de cor.

ou de notes fantômes dont la fréquence fondamentale est sous-multiple commune aux hauteurs de plusieurs notes réelles (*cf.* figure 6.3 (c)).

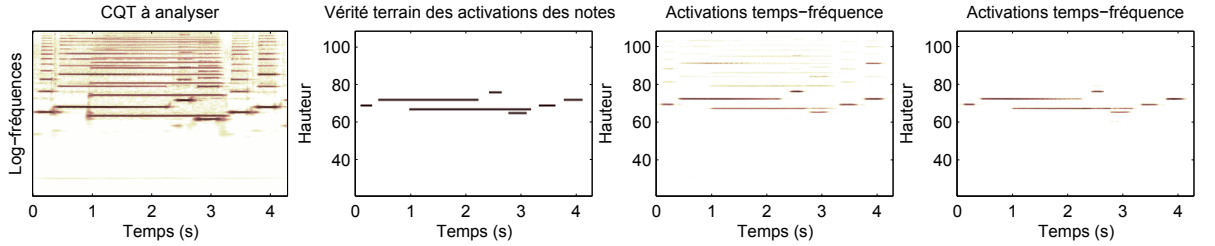
La variabilité des signaux musicaux rend incertaine l'existence d'une initialisation optimale. Nous pouvons alors appliquer des aprioris sur les paramètres, ou maitriser leur vitesse de convergence. Nous aidons ainsi l'algorithme à converger vers des solutions pertinentes, le rendant moins sensible à une initialisation sous-optimale. Pour illustrer les bénéfices des modules que nous allons ajouter à l'algorithme HALCA, nous garderons le même signal d'entrée que sur la figure 6.3, et nous initialiserons les paramètres identiquement aux expériences (b) ou (c) correspondantes, selon le cas qui nous intéresse.

**Apriori de monomodalité.** Dans le cas où l'on connait le nombre $S$ de sources présentes dans le signal, et si l'on sait que chacune d'entre elles est monodique (c'est-à-dire que chaque vecteur $P_h(i,t,s)_i$ doit être monomodal), alors on peut utiliser l'apriori de monomodalité (section 3.5 p.54). Pour ce faire, comme avec son application à la SIPLCA, il est nécessaire d'adapter un peu le modèle HALCA en décomposant les activations temps-fréquence de la manière suivante :

$$P_h(i,t,s) = P_h(t,s)P_h(i,|t,s). \tag{6.14}$$

La mise à jour (6.10) est alors remplacée par les étapes suivantes (la notation $\tilde{P}$ est utilisée pour les paramètres qui ne sont pas encore normalisés) :

$$P_h(t,s) \propto \sum_{f,z,i} V_{ft} P(i,s,z,c=h|f,t), \tag{6.15}$$




(a) Sans apriori (b) Avec apriori de monomodalité

**Figure 6.4** – Illustration de l'apriori de monomodalité sur les activations temps-fréquence du modèle HALCA. L'initialisation des paramètres est identique dans les deux cas.

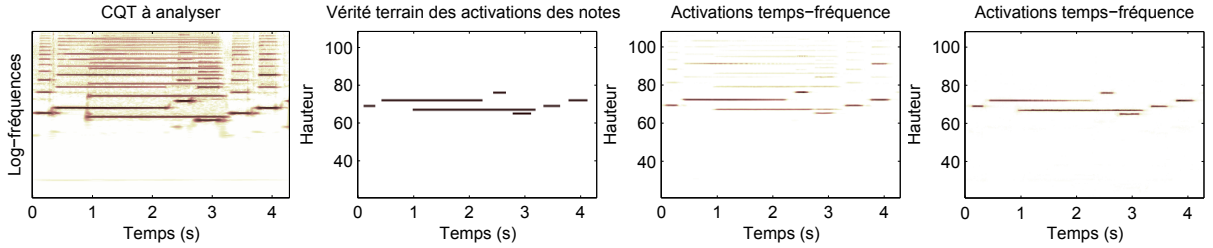
$$\tilde{P}_h(i|t,s) = \sum_{f,z} V_{ft} P(i,s,z,c=h|f,t), \tag{6.16}$$

$$P_h(i|t,s) = \text{Mono}\left(\tilde{P}_h(i|t,s), \beta_{\text{mono}}, \gamma\right). \tag{6.17}$$

$\beta_{\text{mono}}$ et $\gamma$ sont deux hyperparamètres caractérisant l'algorithme Mono $(.,.)$ (Algorithme 5 page 59). Sur la figure 6.4, l'effet de l'utilisation de cet apriori est illustré : dans les deux cas (avec ou sans apriori), la même initialisation des coefficients d'enveloppe est utilisée, à savoir celle trop pentue que nous avions faite lors de l'expérience (b) de la figure 6.3. Après expérimentations, on peut se rendre compte de l'efficacité de l'apriori quand le nombre de sources ne dépasse pas $S = 2$ (il sera employé dans une tâche d'estimation de hauteur simple dans la section suivante). En revanche, lorsque nous avons affaire à un plus grand degré de polyphonie, l'utilisation de cet apriori mène à des résultats non pertinents, en raison d'un trop grand nombre de maxima locaux. De plus, le cas où toutes les sources sont monodiques est trop restrictif et nous voudrions pouvoir analyser des instruments polyphoniques comme le piano ou la guitare. L'apriori de monomodalité ne sera donc utilisé dans aucun des systèmes de transcription automatique que nous proposerons au chapitre 8 page 125.

**Apriori de parcimonie.** Dans le cas où les sources d'un signal d'entrée sont polyphoniques, l'apriori de monomodalité ne peut plus s'appliquer, et nous pouvons le remplacer par un apriori de parcimonie (section 3.2 page 42). L'idée est de supposer qu'il est plus probable d'expliquer une même observation avec un plus petit nombre de notes. Dans le cas du modèle HALCA, cet apriori peut être appliqué à $\theta_d = \theta_{its} = P_h(i,t,s)$. Ce faisant, on considère en même temps plusieurs niveaux de parcimonie :

- peu de notes sont présentes à un temps donné,
- peu d'instruments contribuent à la production d'une note donnée,
- une source n'est pas nécessairement active à tout instant.




(a) Sans apriori (b) Avec apriori de parcimonie

**Figure 6.5** – Illustration de l'apriori de parcimonie sur les activations temps-fréquence du modèle HALCA. L'initialisation des paramètres est identique dans les deux cas.

Pour appliquer l'apriori de parcimonie, il suffit de remplacer la mise à jour (6.10) par :

$$\tilde{P}_h(i,t,s) = \sum_{f,z} V_{ft} P(i,s,z,c=h|f,t) \tag{6.18}$$

$$P_h(i,t,s) = \text{Parci}\left(\tilde{P}_h(i,t,s), \beta_{\text{parci}}\right) \tag{6.19}$$

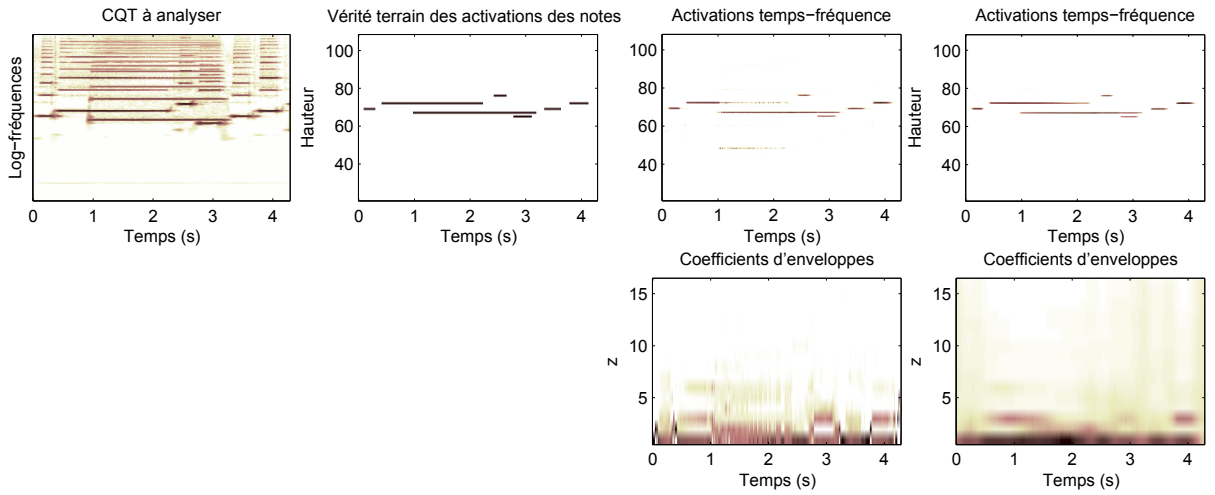
où $\beta_{\text{parci}}$ est l'hyperparamètre qui définit la force de l'apriori et Parci $(.,.)$ l'Algorithme 1 page 46.

On peut voir un exemple de l'utilisation de cet apriori sur la figure 6.5 : ici encore il est utilisé pour pallier une initialisation trop pentue des coefficients d'enveloppe.

**Apriori de continuité temporelle.** Le modèle HALCA permet de modéliser des notes de musique pouvant présenter des variations temporelles d'énergie, de hauteur et d'enveloppe spectrale. Cependant, rien ne contraint ces attributs à évoluer lentement au cours du temps, comme on l'observe généralement dans les sons musicaux réels. Prendre en considération une telle information pourrait aider l'algorithme EM à éviter certains minima locaux. Ici, on peut appliquer l'apriori de continuité temporelle sur les paramètres incarnant le timbre des sources, soit $P_h(z|t,s)$. Nous faisons ainsi l'hypothèse raisonnable que l'énergie relative des partiels d'un instrument à la trame $t$ est proche de celle à la trame $t-1$. Deux raisons principales ont motivé notre choix d'appliquer cet apriori sur ces paramètres. D'abord, contrairement à une contrainte de continuité sur des activations temps-fréquence, ici on ne défavorise pas des attaques des notes pouvant être brutales, ni des phénomènes de variations de fréquence fondamentale comme les vibratos ou les glissandos. De manière plus pragmatique, nous avons déjà appliqué l'apriori de parcimonie ou de monomodalité sur $P_h(i,t,s)$ et deux aprioris différents sur le même ensemble de paramètres pourraient mener à des difficultés en terme de résolution de l'étape MAP. Pour appliquer l'apriori de continuité temporelle, il faut remplacer la mise à jour (6.11) par les étapes suivantes :

$$\tilde{P}_h(z|t,s) = \sum_{f,t,i} V_{ft} P(i,s,z,c=h|f,t), \tag{6.20}$$

$$\forall s \in [\![1,S]\!],\ P_h(z|t,s) = \text{Temp2}\left(\tilde{P}_h(z|t,s), \beta_{\text{temp}}\right), \tag{6.21}$$




(a) Sans apriori (b) Avec apriori de continuité temporelle

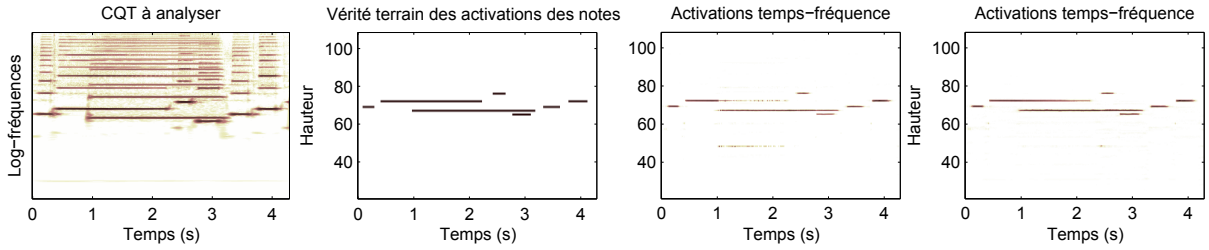
**Figure 6.6** – Illustration de l'apriori de continuité temporelle sur les coefficients d'enveloppes du modèle HALCA. L'initialisation des paramètres est identique dans les deux cas. Les coefficients d'enveloppe sont ceux de la source $s = 1$ : $P_h(z|t, s = 1)$.

Temp2 $(.,.)$ correspondant à l'Algorithme 3 page 50 et $\beta_{\text{temp}}$ à la force de l'apriori.

Après expérimentation, on observe que l'ajout de cet apriori aide l'algorithme EM à sortir de maxima locaux, particulièrement quand on se trouve dans la même situation que l'expérience (c) de la figure 6.3, où l'on observe des maxima dans les activations temps-fréquence pour les fréquences sous-multiples communes de hauteurs de plusieurs notes réellement présentes. Pour illustrer nos affirmations, on montre sur la figure 6.6 comment l'ajout de l'apriori de temporalité permet de corriger ce type d'erreur.

**Apriori de ressemblance.** Plutôt que l'apriori de continuité temporelle, on pourrait appliquer l'apriori de ressemblance (section 3.4) sur les coefficients d'enveloppe $P_h(z|t, s_0)$ de chaque source $s_0$ : cela reviendrait à supposer que l'énergie relative des partiels d'un instrument varie peu au cours du temps, et ce quelle que soit la note jouée. Cependant, cette hypothèse est assez loin de la réalité et pour de nombreux instruments, le rapport des énergies des harmoniques ne peut être considéré constant sur toute leur tessiture. De plus, nous avons motivé la création du modèle HALCA justement pour pouvoir considérer ces variations de signature spectrale. Par conséquent, nous n'appliquons pas l'apriori de ressemblance au modèle HALCA.

**Frein.** On peut envisager de freiner la convergence des coefficients d'enveloppe. Cela nous permettrait de supposer que la valeur optimale de ces paramètres n'est pas très éloignée de leur initialisation. Contrairement au ralentissement de la convergence des spectres de base de la PLCA, qui avait pour conséquence une plus grande parcimonie des activations de notes (*cf.* chapitre 4 page 61), nous n'avons pas observé un tel phénomène sur les activations temps-fréquence. Typiquement, dans le cas de l'expérience (b) de la figure 6.3, ralentir la convergence




(a) Sans frein

(b) Avec frein sur les coefficients d'enveloppes

**Figure 6.7** – Maitrise de la vitesse de convergence : un frein appliqué aux coefficients d'enveloppes dans l'algorithme EM peut permettre à celui-ci de ne pas converger trop vite vers un maximum local. L'initialisation des paramètres est identique dans les deux cas.

de $P_h(z|i,t)$ ne change rien à l'allure de $\sum_s P_h(i,t,s)$ en fin d'algorithme. En revanche, on peut observer dans certains cas que l'algorithme ne se précipite pas vers un maximum local dont il ne pourra plus sortir : comme on le voit sur la figure 6.7, le problème de l'expérience (c) de la figure 6.3 est contourné en freinant la convergence des coefficients d'enveloppes.

**Résumé de l'algorithme HALCA.** Nous résumons dans l'Algorithme 7 les différentes étapes de l'algorithme HALCA, en prenant en compte les conseils et propositions faites dans cette section. Nous faisons remarquer que les mises à jour des paramètres que nous proposons sont une fusion des étapes E et M, comme nous l'avons fait pour la PLCA classique (équations (2.12) et (2.13) page 32).

## 6.5 Exemples et discussion

Dans cette section nous testons le modèle HALCA sur deux exemples de signaux. Le premier, monodique, est un extrait d'une voix chantée d'homme *a cappella*. Nous avons choisi cet extrait car la voix est un instrument typique dont les notes possèdent des variations continues de hauteur et d'enveloppe spectrale. L'algorithme HALCA a été appliqué avec un nombre de sources fixé à $S = 1$ et sans l'ajout de module. Certains paramètres après convergence de l'algorithme sont illustrés sur la figure 6.8 : on peut lire sur les activations temps-fréquences $P_h(i,t,s=1)$ la trajectoire de la hauteur du chant et l'on constate bien l'évolution des coefficients d'enveloppe au fil du temps qui traduit les variations d'enveloppe spectrale de la voix. Concernant la distribution temps-fréquence du bruit $P_b(i,t)$, on se rend compte qu'elle possède des maxima pour les fréquences correspondant aux partiels du signal d'entrée : en fait, le noyau de bruit aide aussi à modéliser leur étalement spectral.

Le deuxième signal sur lequel nous testons l'algorithme HALCA est un signal polyphonique. Pour que l'on puisse bien observer son comportement, le signal est plutôt simple : seulement deux instruments monodiques, un cor et une clarinette, jouent quelques notes de musique. Dans cet exemple, on ajoute au modèle HALCA les modules suivants : parcimonie des activations

**Algorithme 7:** Algorithme HALCA

**Initialisation**

· $P_h(i,t,s)$ et $P_b(i,t)$ sont initialisées uniformément ;
· $P_h(z|t,s)$ est initialisée différemment pour chaque source $s$ (par exemple décroissant en $z$) ;
· $P(c)$ est initialisée de sorte que $P(c=h) < P(c=b)$ ;

**Algorithme EM**

**répéter**

· calcul du modèle :

$$P(f,t) = P(c=h)\sum_{s,i,z} P_h(i,t,s)P_h(f-i|z)P_h(z|t,s) + P(c=b)\sum_i P_b(i,t)P_b(f-i); \tag{6.22}$$

· mises à jour multiplicatives :

$$\tilde{P}(c=h) = P(c=h)\sum_{f,t,i,s,z}\frac{V_{ft}}{P(f,t)}P_h(i,t,s)P_h(f-i|z)P_h(z|t,s), \tag{6.23}$$

$$\tilde{P}_h(i,t,s) = P_h(i,t,s)P(c=h)\sum_{f,z}\frac{V_{ft}}{P(f,t)}P_h(f-i|z)P_h(z|t,s), \tag{6.24}$$

$$\tilde{P}_h(z|t,s) = P_h(z|t,s)\left(P(c=h)\sum_{f,i}\frac{V_{ft}}{P(f,t)}P_h(i,t,s)P_h(f-i|z) + \beta_{\text{frein}}\right) \tag{6.25}$$

$\beta_{\text{frein}}$ étant le coefficient de freinage pour $P_h(z|t,s)$,

$$\tilde{P}(c=b) = P(c=b)\sum_{f,t,i}\frac{V_{ft}}{P(f,t)}P_b(i,t)P_b(f-i), \tag{6.26}$$

$$\tilde{P}_b(i,t) = P_b(i,t)P(c=b)\sum_f\frac{V_{ft}}{P(f,t)}P_b(f-i); \tag{6.27}$$

· normalisation et application des aprioris :

$$P(c) \propto \tilde{P}(c),$$
$$P_b(i,t) \propto \tilde{P}_b(i,t);$$
$$\forall s \in [\![1,S]\!],\ P_h(z|t,s) = \text{Temp2}\left(\tilde{P}_h(z|t,s), \beta_{\text{temp}}\right) \quad \text{(Algorithme 3 page 50)},$$

– si on utilise l'apriori de parcimonie :

$$P_h(i,t,s) = \text{Parci}\left(\tilde{P}_h(i,t,s), \beta_{\text{parci}}\right) \quad \text{(Algorithme 1 page 46)},$$

– si on utilise l'apriori de monomodalité :

$$P_h(t,s) \propto \sum_i \tilde{P}_h(i,t,s)$$
$$\forall s,\ \forall t\ P_h(i|t,s) = \text{Mono}\left(\tilde{P}_h(i,t,s), \beta_{\text{mono}}\right) \quad \text{(Algorithme 5 page 59)},$$
$$P_h(i,t,s) = P_h(t,s)P_h(i|t,s)$$

**jusqu'à** *convergence*;

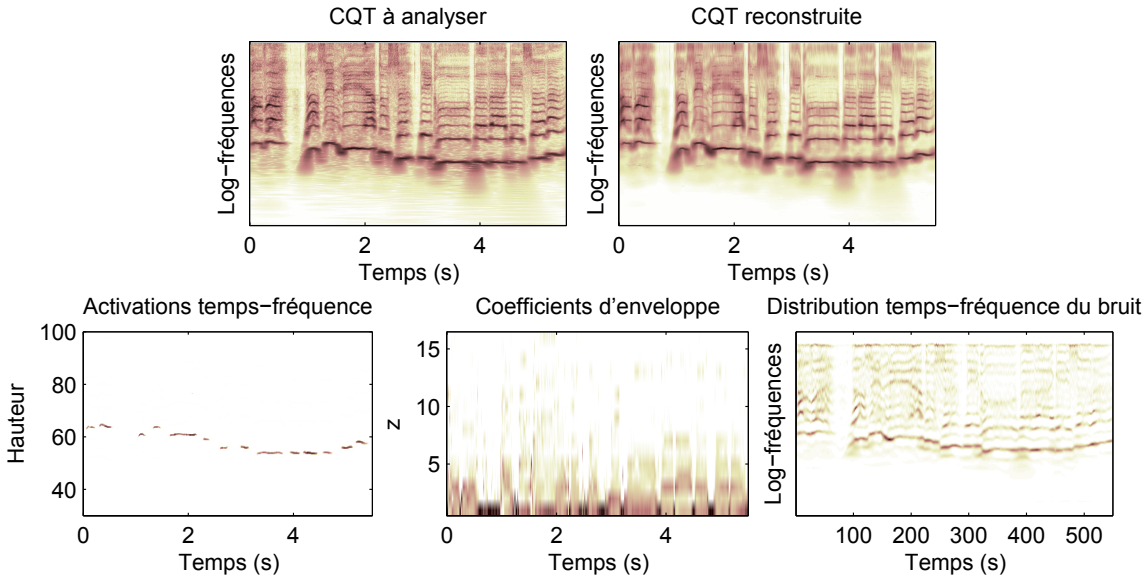



**Figure 6.8** – Illustration de la décomposition de la CQT d'un signal de voix chantée par le modèle HALCA (avec $S = 1$ source).

temps-fréquence ainsi que continuité temporelle et freinage des enveloppes spectrales. On peut observer le résultat de la décomposition sur la figure 6.9. Même si ce sont les activations temps-fréquence résultantes $P_h(i,t) = \sum_s P_h(i,t,s)$ qui nous intéressent pour la transcription, il est intéressant d'observer sur cette figure le détail de la décomposition pour chaque source $s$. On peut constater que chaque source du modèle ne correspond pas vraiment à un instrument particulier : on observe que le cor est modélisé grâce aux deux sources. Par exemple, au temps $t = 2$ sec., on se rend compte que la source $s = 1$ est utilisée essentiellement pour modéliser les harmoniques 1 et 3 du cor (en plus de la clarinette) tandis que la source $s = 2$ est utilisée pour modéliser les autres harmoniques. En fait, nous avons imaginé le modèle HALCA tel que chaque source est censée représenter un unique instrument, mais dans la pratique l'algorithme ne converge pas ainsi. Les sources estimées sont plutôt des « méta-instruments », plusieurs d'entre eux étant combinés pour représenter chaque instrument. Et cela malgré les aprioris de parcimonie et de continuité temporelle. Cela implique que l'algorithme HALCA ne pourra pas être utilisé pour obtenir des transcriptions individuelles pour chaque instrument. En revanche, on n'est pas obligé de connaitre ou d'estimer le nombre réel d'instruments dans un enregistrement : un nombre fixe de sources peut être suffisant pour modéliser un nombre inconnu d'instruments. De plus, nous n'avons plus besoin d'émettre l'hypothèse de la section 6.2, sur le fait que les notes d'un instrument polyphonique devraient posséder la même forme spectrale. Dans la section suivante, nous allons étudier, entre autres, l'effet du nombre de sources sur les performances dans une tâche d'estimation *multipitch*.

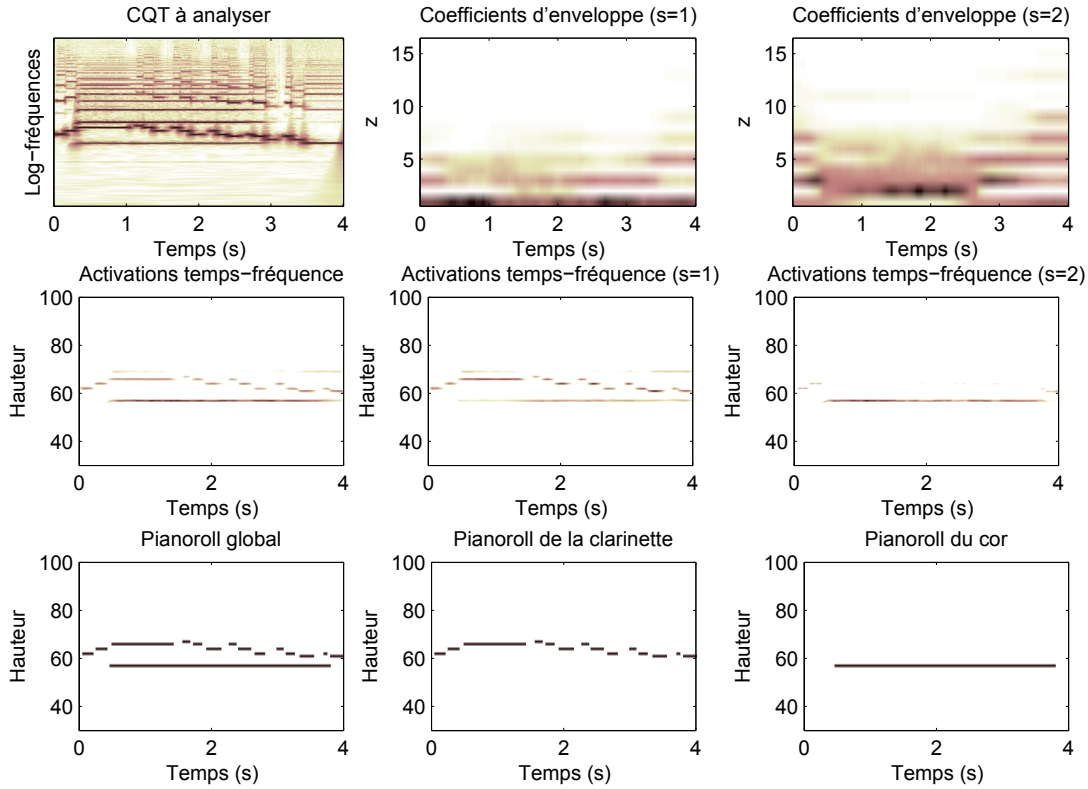



**Figure 6.9** – Illustration de la décomposition de la CQT d'un signal polyphonique par le modèle HALCA (avec $S = 2$ sources).

## 6.6 Tests préliminaires

Dans les sections précédentes, nous avons tenté d'illustrer le comportement de l'algorithme HALCA sur quelques exemples simples de signaux. Nous souhaitons dans cette section procéder à une étude rigoureuse de l'algorithme et de ses différentes instanciations (valeurs des hyperparamètres et ordre $S$ du modèle) sur une tâche bien définie. La tâche que nous retenons est l'évaluation de hauteurs multiples (*multipitch*), qui consiste à estimer à des temps réguliers la hauteur MIDI de l'ensemble des notes présentes dans le signal. En d'autres termes, cela consiste à estimer le pianoroll correspondant au signal. Même si cette tâche n'est pas tout à fait l'application principale que nous voulons faire de nos modèles (transcription automatique), elle semble plus appropriée pour l'étude du comportement de l'algorithme et de l'influence des valeurs des hyperparamètres, principalement en raison du fait que, contrairement à la transcription automatique, elle ne nécessite quasiment aucun post-traitement.

Avant d'aller plus loin, disons juste un mot sur la manière dont on va construire les symboles permettant de désigner les différentes instanciations de l'algorithme HALCA : $H_S - option$ où $S$ est le nombre $S$ de sources du modèle et $-option$ désigne les modules dont les hyperparamètres sont non-nuls : $m$ pour l'apriori de monomodalité, $p$ pour l'apriori de parcimonie, $t$ pour l'apriori de continuité temporelle et $f$ pour le frein sur les coefficients d'enveloppe. Par exemple $H_1 - mt$ désigne l'algorithme HALCA où $S$ est fixé à 1 et où on utilise les aprioris de monomodalité et de continuité temporelle.

### 6.6.1 Estimation de hauteur simple

Avant de nous atteler à l'analyse des signaux polyphoniques, considérons en premier lieu des enregistrements ne comportant pas plus d'une note à la fois : la première tâche que nous examinons est l'estimation de hauteur simple. Cela nous donnera un premier aperçu de la pertinence du modèle HALCA, et de sa capacité à s'adapter à n'importe quel instrument de musique. Cela nous permettra également d'étudier l'efficacité de l'apriori de monomodalité qui s'applique uniquement si l'on sait que les sources sont monodiques. La base de données utilisée pour cette évaluation est constituée de 3307 notes isolées, sans silences, provenant de la base Iowa [Webg]. Elle contient des enregistrements de plusieurs instruments, jouant sur l'ensemble de leur tessiture, selon différents modes de jeu et avec plusieurs nuances. Pour chaque signal, la CQT est calculée puis analysée par l'algorithme HALCA avec un nombre de sources fixé à $S = 1$. La hauteur de la note jouée est alors déduite à chaque trame temporelle du maximum des activations temps-fréquence $P_h(i, t, s = 1)$ et arrondie à la hauteur MIDI la plus proche. Nous profitons de cette évaluation pour comparer plusieurs réalisations de l'algorithme HALCA : on les définit avec les valeurs correspondantes de leurs hyperparamètres dans le tableau 6.2. Ces valeurs ont été fixées manuellement, d'après l'observation des résultats sur quelques exemples. Pour s'assurer que les activations à chaque instant correspondent bien à des vecteurs monomodaux, l'apriori de monomodalité est utilisé systématiquement. Nos méthodes sont également comparées à l'algo-

| Symbole | $\beta_{\text{mono}}$ | $\beta_{\text{temp}}$ | $\beta_{\text{frein}}$ |
|---|---|---|---|
| $H_1 - m$ | $10^4$ | 0 | 0 |
| $H_1 - mf$ | $10^4$ | 0 | 5 |
| $H_1 - mt$ | $10^4$ | $10^4$ | 0 |

**Table 6.2** – Les différentes instanciations du modèle HALCA pour l'estimation de hauteur simple. L'hyperparamètre $\gamma$ qui permet de définir la force de l'asymétrie est fixé à $\gamma = 0.001$ (*cf.* section 3.5 page 54).

rithme YIN [dCK02], téléchargeable gratuitement en ligne [Weba], qui représente l'état de l'art des algorithmes d'estimation *monopitch*, et que nous utilisons comme référence de performance. Nous avons fixé la taille des fenêtres d'analyse pour YIN à 100 ms, soit à la taille minimum nécessaire si l'on souhaite considérer des hauteurs allant de 21 à 108 sur l'échelle MIDI.

Nous mesurons, pour chaque instrument et chaque système, la proportion de trames pour lesquelles la hauteur est mal estimée. La figure 6.10 présente ces résultats. Nous pouvons tirer plusieurs conclusions de ces résultats. D'abord, quel que soit le système que nous proposons, on peut constater qu'ils sont du même ordre de grandeur que ceux de l'algorithme YIN pour la plupart des instruments, et que pour le hautbois et le basson, ils sont meilleurs. Ces deux instruments ont en effet des centroïdes spectraux assez élevés, indépendamment de la hauteur des notes, et là où YIN fait des erreurs d'octave, le modèle HALCA s'adapte aux formes spectrales des notes. Ensuite, on peut comparer les résultats donnés par nos différents algorithmes. On remarque que le coefficient de freinage sur les coefficients d'enveloppe n'a pas d'effet significatif sur les résultats : suivant l'instrument considéré il peut très légèrement améliorer ou baisser la performance de l'algorithme. En fait, on a vu qu'ajouter ce frein permettait de supposer que les bonnes valeurs des coefficients d'enveloppe étaient proches de leur initialisation. Comme l'initialisation est restée la même, cette supposition s'est avérée vraie, ou fausse selon l'instrument et la note à analyser. L'apriori de continuité temporelle a, en revanche, un impact positif sur les résultats, ce qui confirme sa pertinence dans le cas monodique.

### 6.6.2 Estimation de hauteurs multiples

**Description de l'algorithme.** Nous avons constaté que le modèle HALCA permettait de modéliser n'importe quel instrument harmonique et de bien traiter le problème d'estimation de hauteur simple. Nous souhaitons dorénavant effectuer quelques tests préliminaires de détection de hauteurs multiples, ce qui nous permettra de mieux comprendre le comportement de l'algorithme avec ou sans les différents modules. Décrivons tout d'abord le système complet :

- Pour un signal d'entrée donné, la CQT est calculée puis analysée grâce à l'algorithme HALCA (30 itérations sont suffisantes pour la convergence).
- Les activations temps-fréquence sont déduites des paramètres par $P_h(i,t) = \sum_s P_h(i,t,s)$.
- On détecte ensuite les pics de chaque colonne de $P_h(i,t)$.
- Au temps $t$, un pic $i_0$ donné est associé à la note MIDI $n_0$ la plus proche.

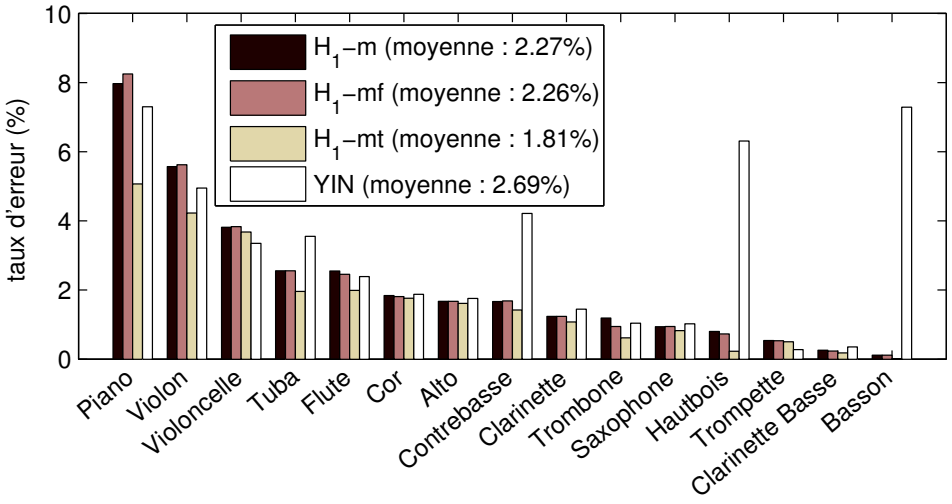



**Figure 6.10** – Estimation de hauteur simple : taux d'erreur moyen pour chaque instrument de la base de données.

- On peut alors obtenir la vélocité de la note MIDI $n_0$ au temps $t$ en sommant les coefficients de $P_h(i,t)$ pour les $i$ qui sont éloignés de moins d'un quart de ton de $i_0$, et l'on obtient ainsi une matrice $P(n,t)$ de puissance des notes : puisque la CQT est calculée avec trois points fréquentiels par demi-ton, $P(n_0,t)) = P_h(i_0,t) + P_h(i_0-1,t) + P_h(i_0+1,t)$.
- On normalise ensuite $P(n,t)$ de la manière suivante : $P(n,t) \leftarrow P(n,t)/\max_{n,t} P(n,t)$.
- Pour finir, on binarise la matrice $P(n,t)$ en appliquant un seuil $P_{\min}$ pour obtenir une estimation de la matrice d'activation des notes : $\hat{A}(n,t) = 1$ si $P(n,t) > P_{\min}$, et à $\hat{A}(n,t) = 0$ sinon.

**Métrique.** Pour évaluer la qualité des activations $\hat{A}(n,t)$ estimées pour un signal donné, on la compare à la vérité terrain $A(n,t)$ selon 3 mesures classiques [vR79] :

- le rappel $\mathcal{R}$ qui mesure le taux de hauteurs correctement estimées sur le nombre total de hauteurs de la vérité terrain,
- la précision $\mathcal{P}$ qui mesure le taux de hauteurs correctement estimées sur le nombre total de hauteurs estimées,
- la F-mesure $\mathcal{F}$ qui combine ces deux scores pour donner une mesure globale de qualité.

Dans notre cas, ils sont calculés de la manière suivante (on rappelle que $\hat{A}$ et $A$ sont des matrices binaires) :

$$\mathcal{R} = \frac{\sum_{n,t} \hat{A}(n,t)A(n,t)}{\sum_{n,t} A(n,t)},$$
$$\mathcal{P} = \frac{\sum_{n,t} \hat{A}(n,t)A(n,t)}{\sum_{n,t} \hat{A}(n,t)},$$

$$\mathcal{F} = \frac{2\mathcal{R}\mathcal{P}}{\mathcal{R}+\mathcal{P}}.$$

**Base de données.** La base de données que nous utilisons ici est la base d'apprentissage $BD_{\text{app}}$ décrite en détails dans l'annexe C page 171. Elle est constituée de cinq extraits de 30 secondes de musique classique, issus de la base RWC [GN03]. Le nombre d'instruments dans les fichiers de cette base varie de 1 à 5, avec des degrés de polyphonies oscillant de 1 à 10.

**Nombre de sources.** Nous avons soulevé dans la section (6.5) le fait que dans la pratique, les différentes sources du modèle HALCA ne correspondaient pas aux différents instruments après convergence de l'algorithme. S'il n'est donc pas nécessaire d'estimer ou de connaître le vrai nombre d'instruments dans un signal à analyser, on peut quand même étudier l'influence de la valeur de $S$ sur les résultats. Pour cela nous avons appliqué le modèle HALCA (sans ajout de module) avec différentes valeurs pour $S$, allant de 1 à 5, à chaque fichier de $BD_{\text{app}}$, et calculé la F-mesure en fonction du seuil $P_{\text{min}}$. Sur la figure 6.11, on peut voir les courbes moyennes de F-mesure en fonction du seuil de détection pour chacun des systèmes. On peut observer que la qualité de l'estimation augmente avec le nombre de sources, et ce quelque soit le seuil considéré. Il semble donc que la qualité de la décomposition est fonction croissante du nombre de sources, c'est-à-dire de l'expressivité du modèle. En revanche, il apparait que les performances croissent de moins en moins au fur et à mesure que l'on considère des sources supplémentaires. Comme une grande valeur de $S$ induit une plus grande complexité (le temps de calcul de chaque système est reporté dans la Table 6.3), il nous semble que la valeur $S = 4$ est un bon compromis entre la performance et la rapidité de l'algorithme. Nous garderons donc ce nombre de sources pour la suite des expériences.

**Ajout de modules.** Maintenant que nous avons fixé le nombre de sources, nous pouvons étudier l'influence des différents modules sur les performances de l'algorithme (apriori de parcimonie, de continuité temporelle et freinage des coefficients d'enveloppe). Afin de mener une étude la plus exhaustive possible, nous avons d'abord considéré l'ajout de chaque module séparément, puis les différentes combinaisons de deux modules, et enfin l'effet des trois modules ensemble. Les différentes instanciations des algorithmes, leurs valeurs correspondantes d'hyperparamètres, fixées à la main après observations sur quelques exemples, ainsi que les temps de calcul sont consultables dans la Table 6.3. Les courbes illustrant la valeur des métriques $\mathcal{R}$, $\mathcal{P}$ et $\mathcal{F}$ en fonction du seuil $P_{\text{min}}$ sont représentées sur la figure 6.12.

De nombreuses conclusions peuvent être tirées de ces résultats, nous listons celles qui nous semblent les plus importantes :

—∘ Observons tout d'abord le temps de calcul de chaque système. On peut constater que l'apriori de continuité temporelle ralentit considérablement l'algorithme. Il s'avère en effet que le sous-algorithme (Algorithme 3 page 50) permettant de résoudre chaque étape MAP demande un certain nombre d'itérations (plusieurs dizaines) pour converger. Le

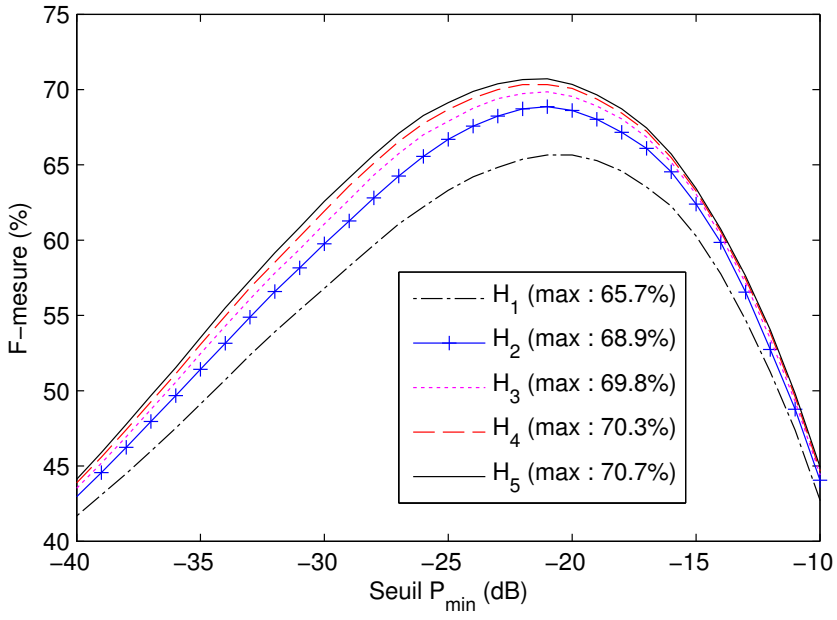
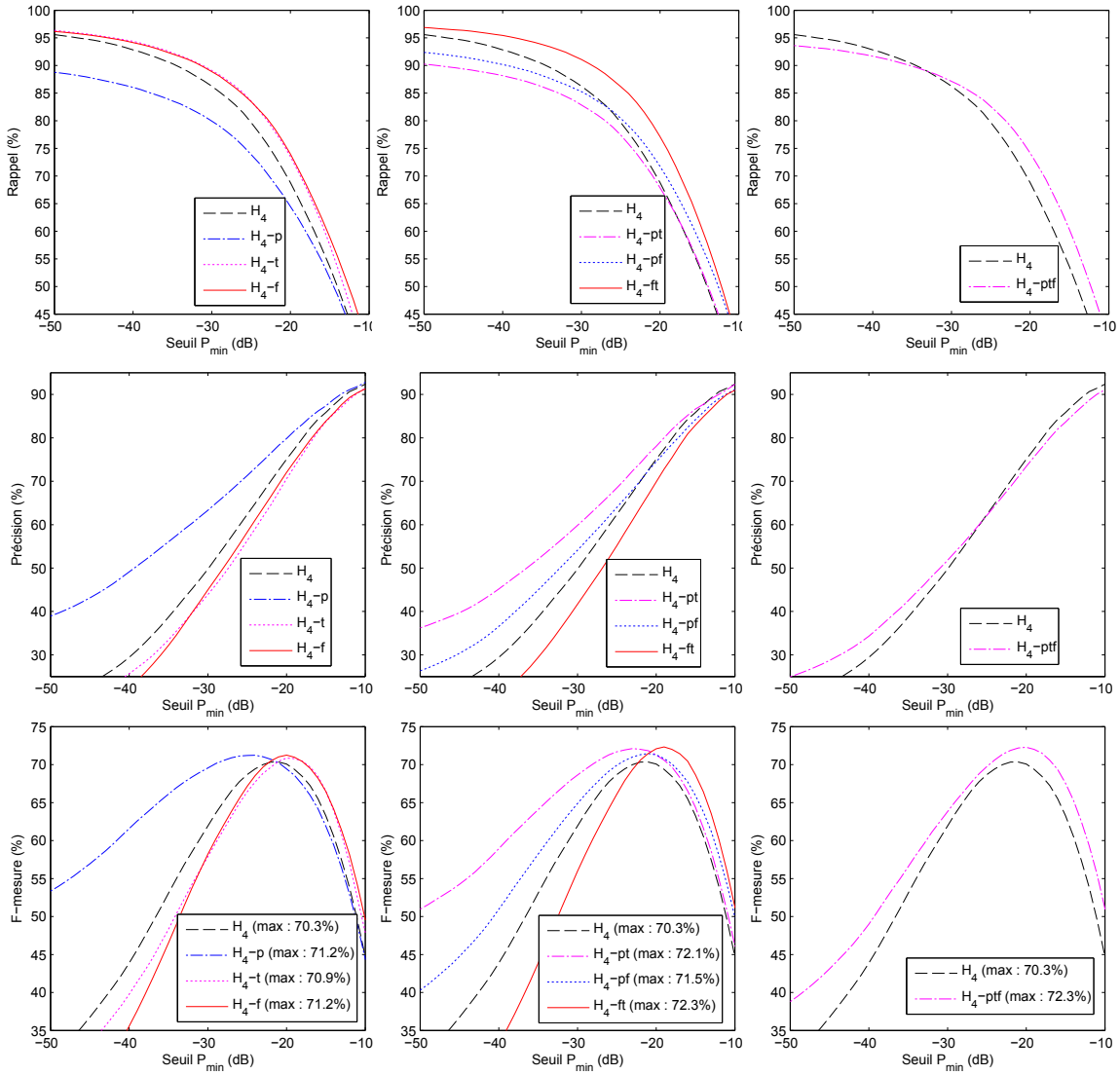



**Figure 6.11** – Étude de l'influence du nombre $S$ de source du modèle HALCA : F-mesure moyenne en fonction du seuil de détection $P_{\text{min}}$.

| Symbole | $S$ | $\beta_{\text{parci}}$ | $\beta_{\text{temp}}$ | $\beta_{\text{frein}}$ | Temps de calcul ($\times$ temps réel) |
|---|---|---|---|---|---|
| $H_1$ | 1 | 0 | 0 | 0 | 1.5 |
| $H_2$ | 2 | 0 | 0 | 0 | 3 |
| $H_3$ | 3 | 0 | 0 | 0 | 4 |
| $H_4$ | 4 | 0 | 0 | 0 | 4.5 |
| $H_5$ | 5 | 0 | 0 | 0 | 5.5 |
| $H_4 - p$ | 4 | 0.02 | 0 | 0 | 9 |
| $H_4 - t$ | 4 | 0 | $10^4$ | 0 | 21 |
| $H_4 - f$ | 4 | 0 | 0 | 5 | 4.5 |
| $H_4 - pt$ | 4 | 0.02 | $10^4$ | 0 | 27 |
| $H_4 - pf$ | 4 | 0.02 | 0 | 5 | 9 |
| $H_4 - ft$ | 4 | 0 | $10^4$ | 5 | 21 |
| $H_4 - ptf$ | 4 | 0.02 | $10^4$ | 5 | 27 |

**Table 6.3** – Les différentes instanciations du modèle HALCA pour l'estimation de hauteurs multiples, ainsi que le temps de calcul approximatif correspondant (mesuré en utilisant une version 64 bits de Matlab, avec un processeur à deux cœurs de 3.1GHz).




(a) Ajout d'un unique module (b) Combinaison de deux modules (c) Ajout de trois modules

**Figure 6.12** – Étude de l'influence des modules sur le modèle HALCA pour une tâche d'estimation de hauteurs multiplies : résultats moyens en fonction du seuil de détection $P_{\text{min}}$.

ralentissement dû à l'apriori de parcimonie est lui nettement moins important puisqu'il existe une solution quasi-analytique (seule une racine doit être estimée numériquement) pour la résolution de l'étape MAP (Algorithme 1 page 46). Pour finir, le temps de calcul supplémentaire consommé par l'ajout du frein est quant à lui négligeable.

- Observons maintenant l'effet des modules quand ils sont utilisés indépendamment (figure 6.12 (a)). L'apriori de parcimonie, comme on s'y attendait, permet un gain notable de la précision, même si cela se fait au détriment du rappel pour un seuil donné. La courbe de F-mesure résultante montre une courbe en cloche plus large que celle du système $H_4$ : cette variante permet ainsi d'être moins sensible à une valeur sous-optimale du seuil $P_{\min}$. Concernant les modules de continuité temporelle, et de freinage, il est intéressant de constater qu'ils ont quasiment la même influence sur les résultats : un meilleur rappel au détriment d'une précision plus faible pour un seuil donné. Mais au final, les courbes de $\mathcal{F}$ correspondantes ont quasiment la même forme que celle du système brut $H_4$. On constate tout de même, pour chacun des modules, une augmentation du maximum de la courbe de F-mesure.
- Combiner deux modules semble beaucoup plus intéressant (figure 6.12 (b)). En effet, pour les systèmes $H_4 - pt$ et $H_4 - pf$, la courbe de $\mathcal{F}$ est au dessus de celle du système $H_4$, quelle que soit la valeur du seuil, ce qui assure de meilleures performances pour n'importe quelle valeur de $P_{\min}$. Le système $H_4 - ft$ possède quant à lui la meilleure valeur de F-mesure pour un seuil optimal. En revanche, on remarque une courbe en cloche plus étroite, c'est-à-dire que la qualité de l'estimation est plus sensible à la valeur de $P_{\min}$.
- Finalement, combiner les trois modules offre dans cette expérience à la fois la meilleure valeur de $\mathcal{F}$ pour un seuil optimal, et l'assurance d'avoir de meilleures performances que l'algorithme simple $H_4$ pour n'importe quelle valeur de $P_{\min}$.

D'après ces observations, et en considérant à la fois la qualité de l'estimation de hauteurs simples et le temps de calcul, on retiendra pour l'application du modèle HALCA à la transcription automatique (chapitre 8 page 125) trois systèmes :

- $H_4$ qui servira d'algorithme de référence et permettra d'évaluer les bénéfices éventuels de l'ajout de modules pour la transcription,
- $H_4 - pf$ qui présente un bon rapport performance/rapidité,
- $H_4 - ptf$ qui dans cette expérience présente la meilleure valeur maximale de F-mesure.

## 6.7 Conclusion

Nous avons présenté dans ce chapitre un nouveau modèle de RTF, le modèle HALCA, qui a pour caractéristique de pouvoir modéliser des instruments harmoniques pouvant jouer des notes montrant des variations temporelles de hauteur et d'enveloppe spectrale. Il permet également de considérer la présence de bruit. De plus, nous avons introduit différentes améliorations possibles

via l'ajout de modules : aprioris et maîtrise de la vitesse de convergence. Nous avons observé aussi que les sources dans le modèle HALCA ne représentaient pas vraiment des instruments dans la pratique, mais plutôt des méta-instruments servant eux-mêmes à modéliser les vrais instruments. Un premier test que nous avons effectué est une application à la tâche d'estimation de hauteur simple. Les résultats ont permis de souligner la pertinence du modèle pour pouvoir prendre en compte tout type d'instrument de musique. Ensuite, nous avons appliqué l'algorithme HALCA à une tâche d'estimation de hauteurs multiples. Nous avons pu constater que les performances augmentaient avec le nombre $S$ de sources. Nous avons ensuite étudié l'influence de l'ajout de modules, et avons sélectionné deux instanciations de l'algorithme que nous utiliserons prochainement pour une tâche de transcription.

L'observation du comportement du modèle permet de soulever deux questions :

- Si les sources ne permettent pas dans la pratique de modéliser chaque instrument séparément, ne pouvons-nous pas tenter d'imaginer un modèle de CQT sans introduire la notion de source ?
- Puisqu'on a observé une augmentation des performances quand on multipliait le nombre de sources, n'est-il pas possible de créer un modèle encore plus expressif que le modèle HALCA ?

En réponse à ces questions, nous proposons dans le chapitre suivant un nouveau modèle, encore plus expressif que celui proposé dans ce chapitre, où la notion de source est abandonnée.

# Chapitre 7

# Modèle par note : BHAD

## 7.1 Introduction

En conclusion du chapitre précédent, nous avons émis l'idée qu'il serait intéressant d'imaginer un modèle de RTF$^+$ qui ne fasse pas appel à la notion de source : dans le cas du modèle HALCA, les sources ne correspondaient pas vraiment aux instruments présents dans le signal, d'où leur remise en question. Dans ce chapitre, nous proposons donc un nouveau modèle, que l'on appelle « décomposition aveugle, harmonique et adaptative » (BHAD pour *Blind Harmonic Adaptive Decomposition*), n'introduisant pas la notion de source. Le modèle BHAD, à l'instar du modèle HALCA, permet de modéliser des notes présentant des variations de hauteur et d'enveloppe spectrale. Nous allons voir en effet que chaque colonne de CQT va être modélisée comme une somme pondérée de spectres harmoniques (l'ensemble de toutes les fréquences fondamentales est considéré, avec éventuellement des poids nuls), chaque spectre possédant sa propre enveloppe spectrale dépendante du temps. Ce modèle est donc plus expressif que le modèle HALCA, pour lequel on avait supposé que toutes les notes d'une même source à un temps donné possédaient les mêmes coefficients d'enveloppe. Nous verrons également que le modèle BHAD permet de modéliser des spectres légèrement inharmoniques. Après avoir présenté ce modèle, et décrit l'algorithme permettant d'estimer ses paramètres, nous étudierons les manières d'ajouter différents modules (aprioris ou frein). Enfin, en guise de tests préliminaires et pour mieux analyser les différentes instanciations de l'algorithme BHAD, nous l'appliquerons à une tâche d'estimation de hauteurs multiples. Le modèle BHAD a été l'objet de la publication [FBR12a].

## 7.2 Présentation du modèle

Nous décrivons ici le modèle de $P(f,t)$, la distribution de probabilité qui représente une CQT d'entrée. Similairement au modèle HALCA (chapitre 6 page 87), une première variable cachée $c$ est introduite afin de décomposer un signal musical comme la somme d'un signal polyphonique harmonique (alors $c = h$) et d'un signal de bruit ($c = b$) (les notations $P_h(.)$ et $P_b(.)$ sont utilisées

pour $P(.|c=h)$ et $P(.|c=b)$) :

$$P(f,t) = P(c=h)P_h(f,t) + P(c=b)P_b(f,t), \tag{7.1}$$

où $P_h(f,t)_{(f,t)\in\mathbb{Z}\times[\![1,T]\!]}$ et $P_b(f,t)_{(f,t)\in\mathbb{Z}\times[\![1,T]\!]}$ représentent respectivement les CQTs normalisées du signal polyphonique et du signal de bruit. Le modèle de chacune des composantes est maintenant détaillé.

**Modèle polyphonique.** Pour un temps $t$ donné, le spectre de la composante polyphonique, représenté par $P_h(f,t)$, est décomposé comme une somme pondérée de différents spectres harmoniques normalisés $P_h(f|i,t)$, chacun d'entre eux possédant sa propre enveloppe spectrale (dépendant du temps) et sa propre hauteur $i \in [\![0, I-1]\!]$ (nouvelle variable cachée représentant une hauteur sur une échelle logarithmique discrète). Puisque le nombre de notes à un temps donné est inconnu, on suppose présent l'ensemble de tous les spectres possibles, leur poids pouvant être nul :

$$P_h(f,t) = \sum_i P_h(i,t)P_h(f|i,t). \tag{7.2}$$

$P_h(i,t)$ représente alors la puissance du spectre $P_h(f|i,t)$. Cette distribution peut donc être considérée comme les activations temps-fréquences des spectres harmoniques.

Afin de tenir compte de la nature harmonique de $P_h(f|i,t)$, de même que de son enveloppe spectrale spécifique, le même principe que pour le modèle HALCA est retenu : un tel spectre est décomposé comme une combinaison linéaire de $Z$ noyaux harmoniques à bande étroite fixés, notés $P_h(f|z,i)$, partageant la même fréquence fondamentale $i$, et ayant leur énergie concentrée sur la $z^{\text{ème}}$ harmonique :

$$P_h(f|i,t) = \sum_z P_h(z|i,t)P_h(f|z,i). \tag{7.3}$$

On suppose donc que $f$ est indépendant de $t$ conditionnellement à $z$ et $i$. Les poids qui sont appliqués, incarnés par $P_h(z|i,t)$ et appelés coefficients d'enveloppe, définissent l'enveloppe spectrale du spectre harmonique de hauteur $i$ au temps $t$. On peut maintenant utiliser l'avantage de travailler avec la CQT : une modulation de fréquence fondamentale pour un spectre harmonique correspond à une translation de ses partiels. Aussi, pour $i$ et $z$ donnés, $P_h(f|z,i)$ peut être déduit d'un unique noyau $P_h(\mu|z)$, comme suit :

$$P_h(f|z,i) = P_h(f-i|z). \tag{7.4}$$

Ici, $P_h(\mu|z)$ est également un noyau harmonique à bande étroite, ayant son énergie concentrée sur la $z^{\text{ème}}$ harmonique, et dont la fréquence fondamentale est $i=0$. Nous utilisons en réalité les mêmes noyaux que ceux définis dans le modèle HALCA (section 6.2 page 91). Comme les noyaux possèdent de l'énergie uniquement sur les points fréquentiels multiples de leur fréquence fondamentale, l'étalement spectral des partiels d'une note est pris en compte par l'activation des

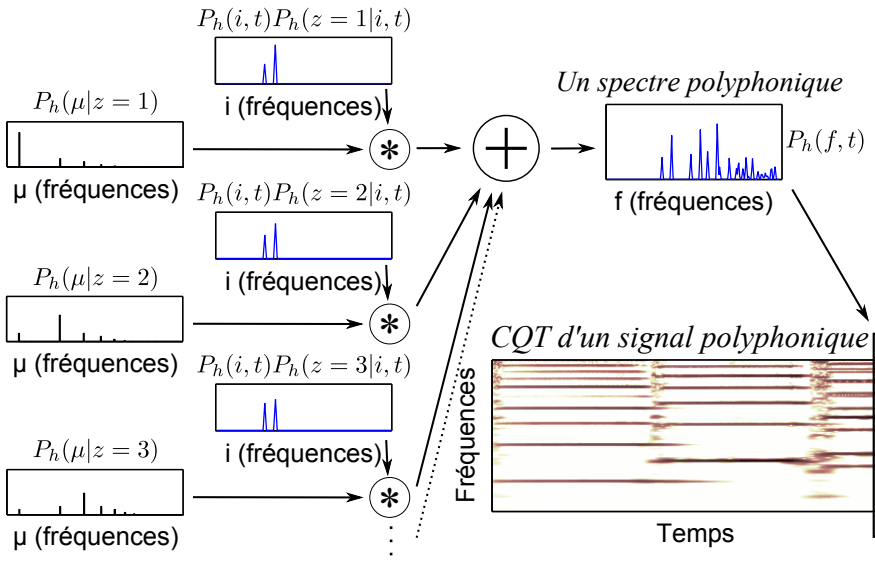



**Figure 7.1** – Modèle pour la composante polyphonique du modèle BHAD au temps $t$.

spectres : plusieurs valeurs de $i$ sont donc nécessaires pour modéliser le spectre d'une note réelle. Ainsi, le modèle BHAD peut s'adapter à n'importe quelle taille d'étalement spectral des partiels, celui-ci pouvant être variable suivant le niveau de non-stationnarité de fréquence fondamentale d'une note.

Finalement, le modèle de la composante polyphonique peut s'écrire comme :

$$P_h(f,t) = \sum_{i,z} P_h(i,t)P_h(z|i,t)P_h(f-i|z). \tag{7.5}$$

On peut remarquer que l'on se retrouve avec un modèle convolutif, ce qui signifie que la variable $f$ est définie comme la somme de deux variables aléatoires $\mu$ et $i$. Ce modèle est illustré sur la figure 7.1.

On peut noter également que ce modèle permet de décrire des spectres de notes légèrement inharmoniques, comme pour le piano où l'on observe une déviation vers les hautes fréquences des partiels les plus aigus [Fle64]. En effet, comme illustré sur la figure 7.2, une note inharmonique de fréquence fondamentale $i$ pourra être décomposée comme la somme d'un spectre de hauteur $i$ ayant de l'énergie uniquement pour ses premiers partiels, et d'un spectre de hauteur $i+1$ dont l'énergie est concentrée sur les partiels plus aigus, etc.

**Modèle de bruit.** Celui-ci est strictement identique au modèle de bruit décrit dans la section 6.2 page 89 :

$$P_b(f,t) = \sum_{i} P_b(i,t)P_b(f-i), \tag{7.6}$$

où $P_b(\mu)_{\mu\in[\![1,F]\!]}$ représente une fenêtre à bande étroite, et $P_b(i,t)_{(i,t)\in[\![0,I-1]\!]\times[\![1,T]\!]}$ la distribution temps-fréquence du bruit. Le lecteur est renvoyé à la figure 6.2 page 90 pour l'illustration d'un

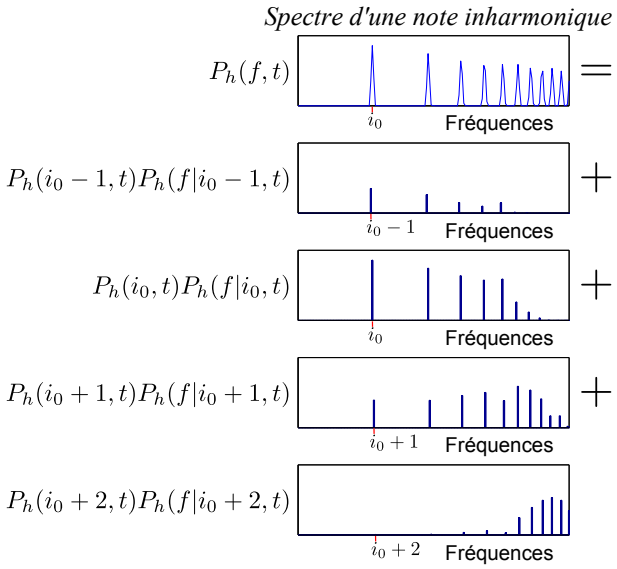



**Figure 7.2** – Exemple de décomposition d'un spectre légèrement inharmonique comme une somme de spectres harmoniques ayant différentes hauteurs et enveloppes spectrales.

| Paramètres | Définition sémantique |
|---|---|
| $P(c=h)$ | Énergie relative de la composante polyphonique harmonique |
| $P(c=b)$ | Énergie relative de la composante de bruit |
| $P_h(i,t)$ | Activations temps-fréquence des spectres harmoniques |
| $P_h(\mu\|z)$ | $z^{\text{ème}}$ noyau harmonique à bande étroite |
| $P_h(z\|i,t)$ | Coefficients d'enveloppe du spectre harmonique de hauteur $i$ au temps $t$ |
| $P_b(i,t)$ | Distribution temps-fréquence du bruit |
| $P_b(\mu)$ | Noyau régulier à bande étroite du bruit |

**Table 7.1** – Les différents paramètres du modèle BHAD.

tel modèle.

**Modèle complet.** En regroupant les équations (7.1), (7.5) et (7.6), le modèle BHAD se formule de la manière suivante :

$$P(f,t) = P(c=h)\sum_{i,z} P_h(i,t)P_h(z|i,t)P_h(f-i|z) + P(c=b)\sum_{i} P_b(i,t)P_b(f-i). \qquad (7.7)$$

On liste dans la Table 7.1 l'ensemble des paramètres du modèle avec leur signification sémantique.

On peut également en déduire le processus génératif d'une CQT :

- $\forall\,(f,t) \in \mathbb{Z} \times [\![1,T]\!]$, on pose $V_{ft} = 0$.
- Répéter $N$ fois :
  - tirer $c$ selon $P(c)_{c=h,b}$,
  - si $c = h$ :
    - tirer $(i,t)$ selon $P_h(i,t)$

- tirer $z$ selon $P(z|i,t)$,
- tirer $\mu$ selon $P_h(\mu|z)$,
- poser $f = \mu + i$,
- poser $V_{ft} = V_{ft} + 1$,

* si $c = b$ :
  - tirer $(i,t)$ selon $P_b(i,t)$,
  - tirer $\mu$ selon $P_b(\mu)$,
  - poser $f = \mu + i$,
  - poser $V_{ft} = V_{ft} + 1$.

On suppose encore que $V_{ft} = 0$ pour $f \notin [\![1, F]\!]$.

## 7.3 Estimation des paramètres

Dans cette section, on détaille la dérivation de l'algorithme EM, qui permet d'estimer l'ensemble les paramètres $\Lambda = \{P(c), P_h(i,t), P_h(z|i,t), P_b(i,t)\}_{i,t,z,c}$ étant données les variables $\left(\bar{f}, \bar{t}\right)$ observables via $V_{ft}$. L'espérance de la log-probabilité jointe des variables observées et cachées, conditionnellement aux observations, est d'abord calculée :

$$\begin{aligned} Q_\Lambda = &\sum_{f,t}\sum_{i,z} V_{ft} P(i,z,c=h|f,t) \left[\ln\left(P(c=h)\right) + \ln\left(P_h(i,t)\right) + \ln\left(P_h(z|i,t)\right) + \ln\left(P_h(f-i|z)\right)\right] \\ &+ \sum_{f,t}\sum_{i} V_{ft} P(i,c=b|f,t) \left[\ln\left(P(c=b)\right) + \ln\left(P_b(i,t)\right) + \ln\left(P_b(f-i)\right)\right]. \end{aligned} \tag{7.8}$$

Lors de l'étape E, on calcule les probabilités *a posteriori* des variables cachées grâce au théorème de Bayes :

$$P(i,z,c=h|f,t) = \frac{P(c=h)P_h(i,t)P_h(z|i,t)P_h(f-i|z)}{P(f,t)}, \tag{7.9}$$

$$P(i,c=b|f,t) = \frac{P(c=b)P_b(i,t)P_b(f-i)}{P(f,t)}, \tag{7.10}$$

$P(f,t)$ étant défini dans l'équation (7.7).

Dans l'étape M, on maximise $Q_\Lambda$ par rapport aux paramètres, sous la contrainte que les probabilités somment à un. Ce qui mène aux mises à jours suivantes :

$$P(c=h) \propto \sum_{f,t,i,z} V_{ft} P(i,z,c=h|f,t), \tag{7.11}$$

$$P_h(i,t) \propto \sum_{f,z} V_{ft} P(i,z,c=h|f,t), \tag{7.12}$$

$$P_h(z|i,t) \propto \sum_f V_{ft} P(i,z,c=h|f,t), \tag{7.13}$$

$$P(c=b) \propto \sum_{f,t,i} V_{ft} P(i,c=b|f,t), \tag{7.14}$$

$$P_b(i,t) \propto \sum_f V_{ft} P(i,c=b|f,t). \tag{7.15}$$

Après l'initialisation des paramètres, l'algorithme EM consiste donc à itérer le calcul du modèle (équation (7.7)), l'étape E (équations (7.9) et (7.10)) et l'étape M (équations (7.11) à (7.14), suivies de la normalisation des paramètres), jusqu'à convergence de la fonction de log-vraisemblance (équation (2.1)).

## 7.4 Initialisation et ajout de modules

Dans cette section, nous étudions la manière d'initialiser intelligemment les paramètres, ainsi que les possibilités d'ajout de modules (aprioris et frein) au modèle BHAD. Beaucoup d'idées ici sont similaires à celles de la section jumelle consacrée au modèle HALCA (section 6.4 page 93), et seront donc introduites plus directement. Nous conseillons donc au lecteur de se rapporter à cette section antérieure pour plus de détails et d'explications.

**Initialisation.** Après expérimentation, nous préconisons les initialisations suivantes des différents paramètres.

- Les activations temps-fréquence $P_h(i,t)$, ainsi que la distribution de bruit $P_b(i,t)$ sont initialisées uniformément.
- Les coefficients d'enveloppes $P_h(z|i,t)$ sont initialisés selon une pente descendante en $z$, puisque l'on constate souvent pour les instruments de musique une décroissance de l'énergie des partiels d'une note en fonction de leur fréquence. De même que pour le modèle HALCA, il faut trouver un bon compromis sur la pente utilisée : une initialisation trop pentue entrainera beaucoup d'erreurs de surestimation de notes (une pour chaque partiel d'un signal), tandis qu'une initialisation trop plate entrainera des erreurs de notes fantômes, sous-multiples de notes réelles.
- Enfin pour l'énergie relative des composantes « harmonique » et « bruit » $P(c)$, on observe qu'en initialisant de manière à ce que $P(c=h) < P(c=b)$, l'algorithme EM a tendance à s'engouffrer moins rapidement vers des maxima locaux. L'impact reste cependant assez faible.

**Apriori de monomodalité.** Si l'on souhaite appliquer cet apriori au modèle BHAD (à utiliser alors seulement si le signal d'entrée est monodique), la distribution d'impulsions devra être décomposée :

$$P_h(i,t) = P_h(t) P_h(i|t), \tag{7.16}$$

et l'apriori s'appliquera à chaque distribution $P_h(i|t)$. Puisque nous avons déjà évalué l'efficacité de cet apriori sur le modèle HALCA dans une tâche d'estimation de hauteur simple (section 6.6.1 page 102), et qu'on ne peut pas l'utiliser pour traiter des signaux polyphoniques, nous ne testerons pas ici son application au modèle BHAD.

**Apriori de Parcimonie.** Dans le modèle BHAD, l'enveloppe spectrale d'un spectre harmonique dépend de la fréquence fondamentale et du temps. Si le modèle est, grâce à cela, très expressif, il est également trop peu contraint, et l'utilisation de l'apriori de parcimonie sur les activations temps-fréquence $P_h(i,t)$ s'avère appropriée pour y remédier. En appliquant cet apriori, on suppose qu'il est préférable d'expliquer un signal d'entrée avec le moins de notes possibles.

**Apriori de temporalité.** Nous pourrions, comme nous l'avons fait au chapitre précédent, appliquer l'apriori de continuité temporelle aux coefficients d'enveloppe $P_h(z|i,t)$ pour chaque valeur de $i$. Cependant, nous avons vu que cet apriori ralentissait cruellement chaque itération d'algorithme, et comme on a ici beaucoup plus de paramètres de coefficients d'enveloppe que pour le modèle HALCA, il ne serait pas raisonnable d'appliquer cet apriori au modèle BHAD. De plus nous souhaitons plutôt tester l'apriori de ressemblance sur ces paramètres.

**Apriori de ressemblance.** Plutôt que l'apriori de continuité, on peut tenter d'appliquer l'apriori de ressemblance sur les coefficients d'enveloppe $P_h(z|i,t)$ pour chaque valeur de $i$, supposant ainsi que l'enveloppe spectrale du spectre d'une note varie peu dans le temps (pour un $i$ donné et pour chaque $z \in [\![1, Z]\!]$, on va encourager les ressemblances suivantes : $P_h(z|i,t=1) \approx P_h(z|i,t=2) \approx \cdots P_h(z|i,t=T)$). Si la force de l'apriori n'est pas trop grande, cela peut permettre de contraindre un peu plus le modèle BHAD (en plus de l'apriori de parcimonie), en autorisant tout de même une certaine variété d'enveloppes. On peut facilement justifier l'ajout de cet apriori si les instruments ayant produit le signal ont un timbre assez stable pour une note donnée. En revanche l'apriori ne sera pas approprié pour des instruments pouvant produire de fortes variations d'enveloppe spectrale par note : voix humaine, guimbarde ou guitare associée à une pédale wah-wah par exemple.

**Frein.** Enfin, nous pouvons, comme pour le modèle HALCA, freiner la convergence des coefficients d'enveloppe. Dans le cas du modèle BHAD, cela permet de supposer que la vraie valeur de ces paramètres n'est pas très éloignée de leur initialisation. Nous verrons lors de l'application du modèle BHAD à l'estimation *multipitch* dans la section suivante, que cela rend les activations temps-fréquence $P_h(i,t)$ beaucoup plus parcimonieuses, de la même manière que nous pouvions le constater sur une application à la PLCA (*cf.* figure 4.1 page 65).

**Résumé de l'algorithme.** Nous résumons dans l'Algorithme 8 les différentes étapes de l'algorithme BHAD. On y a fusionné les étapes E et M pour avoir des règles de mises à jour

multiplicatives des paramètres, et on a également pris en compte les différents modules que l'on peut ajouter.

## 7.5 Tests préliminaires : estimation de hauteurs multiples

Nous évaluons dans cette section l'algorithme BHAD et ses multiples instanciations (ajout des différents modules) dans une tâche d'estimation de hauteurs multiples. Cela va nous permettre de mieux comprendre l'influence des modules et de sélectionner un nombre réduit de systèmes pour l'application à la transcription au chapitre suivant.

**Description du système complet.** Commençons par détailler le système d'estimation *multipitch* :

- Pour un signal d'entrée donné, la CQT est calculée puis analysée grâce à l'algorithme BHAD (30 itérations sont suffisantes pour la convergence).
- À la fin de l'algorithme on détecte les pics de chaque colonne des activations temps-fréquences $P_h(i,t)$.
- Au temps $t$, un pic $i_0$ donné est associé à la note MIDI $n_0$ la plus proche.
- On peut alors obtenir la vélocité de la note MIDI $n_0$ au temps $t$ en sommant les coefficients de $P_h(i,t)$ pour les $i$ qui sont éloignés de moins d'un quart de ton de $i_0$, et l'on obtient ainsi une matrice $P(n,t)$ de puissance des notes : puisque la CQT est calculée avec trois points fréquentiels par demi-ton, $P(n_0,t)) = P_h(i_0,t) + P_h(i_0-1,t) + P_h(i_0+1,t)$.
- On normalise ensuite $P(n,t)$ de la manière suivante : $P(n,t) \leftarrow P(n,t)/\max_{n,t} P(n,t)$.
- Pour finir, on binarise la matrice $P(n,t)$ en appliquant un seuil $P_{\min}$ pour obtenir une estimation de la matrice d'activation des notes : $\hat{A}(n,t) = 1$ si $P(n,t) > P_{\min}$, et à $\hat{A}(n,t) = 0$ sinon.

**Base de données et métriques.** Ce sont les mêmes que celles utilisées pour l'algorithme HALCA : les systèmes seront testés sur la base $BD_{\text{app}}$ (*cf.* annexe C) et les performances seront quantifiées grâce aux trois métriques de précision $\mathcal{P}$, rappel $\mathcal{R}$ et F-mesure $\mathcal{F}$.

**Évaluation et étude des résultats.** Dans la Table 7.2, on liste l'ensemble des instanciations de l'algorithme BHAD avec leurs valeurs correspondantes d'hyperparamètres (fixées manuellement après observation de l'algorithme sur quelques exemples) et leur temps de calcul approximatif. Les symboles qui les représentent sont construits de la manière suivante : $B - option$ correspond à l'algorithme BHAD où *option* désigne les modules dont les hyperparamètres sont non-nuls ($p$ pour l'apriori de parcimonie, $r$ pour l'apriori de ressemblance et $f$ pour le frein sur les coefficients d'enveloppe).

Sur la figure 7.3, on illustre la valeur moyenne des trois métriques en fonction du seuil $P_{\min}$ pour chacun des algorithmes. La colonne (a) permet de visualiser l'effet des modules sur les

**Algorithme 8:** Algorithme BHAD

**Initialisation**

· $P_h(i,t)$ et $P_b(i,t)$ sont initialisés uniformément ;
· $P_h(z|t,i)$ est initialisé de manière décroissante en $z$ pour chaque $t$ et $i$ ;
· $P(c)$ est initialisé de sorte que $P(c=h) < P(c=b)$ ;

**Algorithme EM**

**répéter**

· calcul du modèle :

$$P(f,t) = P(c=h)\sum_{i,z} P_h(i,t)P_h(z|i,t)P_h(f-i|z) + P(c=b)\sum_{i} P_b(i,t)P_b(f-i); \tag{7.17}$$

· mises à jours multiplicatives :

$$\tilde{P}(c=h) = P(c=h)\sum_{f,t,i,z}\frac{V_{ft}}{P(f,t)}P_h(i,t)P_h(z|i,t)P_h(f-i|z), \tag{7.18}$$

$$\tilde{P}_h(i,t) = P_h(i,t)P(c=h)\sum_{f,z}\frac{V_{ft}}{P(f,t)}P_h(z|i,t)P_h(f-i|z), \tag{7.19}$$

$$\tilde{P}_h(z|t,i) = P_h(z|t,i)\left(P(c=h)\sum_{f}\frac{V_{ft}}{P(f,t)}P_h(i,t)P_h(f-i|z) + \beta_{\text{frein}}\right) \tag{7.20}$$

$\beta_{\text{frein}}$ étant le coefficient de freinage pour $P_h(z|t,i)$,

$$\tilde{P}(c=b) = P(c=b)\sum_{f,t,i}\frac{V_{ft}}{P(f,t)}P_b(i,t)P_b(f-i), \tag{7.21}$$

$$\tilde{P}_b(i,t) = P_b(i,t)P(c=b)\sum_{f}\frac{V_{ft}}{P(f,t)}P_b(f-i); \tag{7.22}$$

· normalisation et application des aprioris :

$$P(c) \propto \tilde{P}(c),$$
$$P_b(i,t) \propto \tilde{P}_b(i,t);$$
$$\forall i \in [\![0, I-1]\!],\ P_h(z|t,i) = \text{Res}\left(\tilde{P}_h(z|t,i), \beta_{\text{temp}}\right) \quad \text{(Algorithme 4 page 53)},$$
$$P_h(i,t) = \text{Parci}\left(\tilde{P}_h(i,t), \beta_{\text{parci}}\right) \quad \text{(Algorithme 1 page 46)}.$$

**jusqu'à** *convergence*;

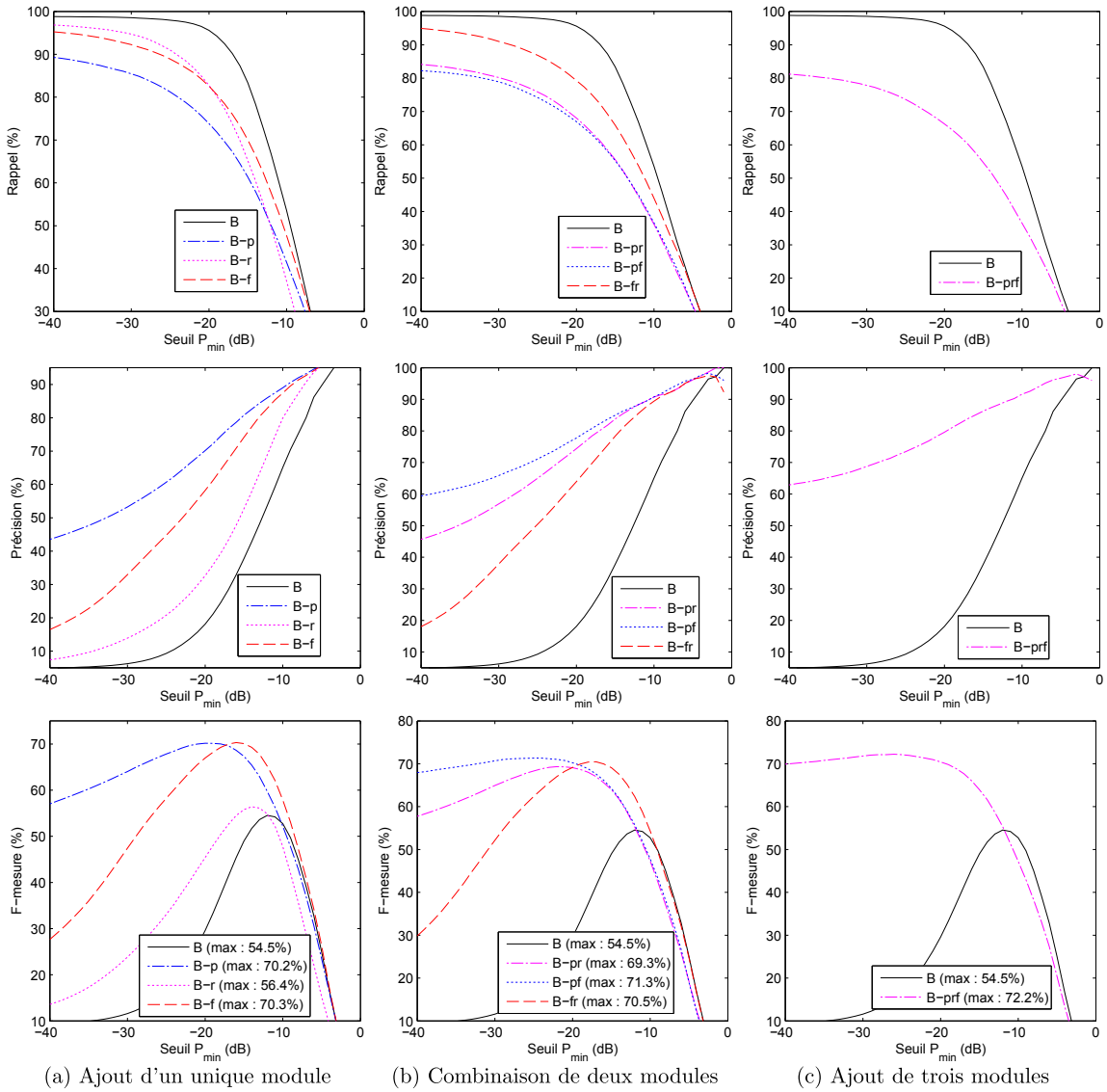
| Symbole | $\beta_{\text{parci}}$ | $\beta_{\text{res}}$ | $\beta_{\text{frein}}$ | Temps de calcul ($\times$ temps réel) |
|---|---|---|---|---|
| $B$ | 0 | 0 | 0 | 2 |
| $B-p$ | 0.3 | 0 | 0 | 3.5 |
| $B-r$ | 0 | 0.5 | 0 | 12 |
| $B-f$ | 0 | 0 | 10 | 2 |
| $B-pr$ | 0.3 | 0.5 | 0 | 13 |
| $B-pf$ | 0.3 | 0 | 10 | 3.5 |
| $B-fr$ | 0 | 0.5 | 10 | 12 |
| $B-prf$ | 0.3 | 0.5 | 10 | 13 |

**Table 7.2** – Les différentes instanciations du modèle BHAD pour l'estimation de hauteurs multiples, ainsi que le temps de calcul approximatif correspondant (mesuré en utilisant une version 64 bits de Matlab, avec un processeur à deux cœurs de 3.1GHz)..

résultats quand ils sont ajoutés séparément. La colonne (b) présente les résultats quand on combine deux modules. Enfin la colonne (c) montre les performances avec l'ensemble des trois modules. Plusieurs remarques intéressantes peuvent être déduites de ces résultats :

- Un premier constat est que l'algorithme $B$, sans l'ajout d'un quelconque module (c'est-à-dire sans ajout de contraintes) présente une F-mesure assez médiocre (figure 7.3 (a), à comparer avec les performances de l'algorithme $H_4$, figure 6.12 page 107). Cela est dû essentiellement aux très faibles valeurs de précision. En fait, le modèle BHAD est tellement peu contraint qu'une note de musique de fréquence fondamentale $i_0$ présente dans le signal se trouve modélisée par l'ensemble de tous les spectres harmoniques de hauteurs multiples ou sous-multiples de $i_0$. L'ajout de modules semble donc être indispensable pour ce modèle.
- Comme prévu, l'apriori de parcimonie permet de rendre les activations temps-fréquence beaucoup plus parcimonieuses, et ainsi d'obtenir une bien meilleure précision et F-mesure.
- Le freinage des coefficients d'enveloppe a sur le modèle BHAD une influence très bénéfique, même quand il est utilisé tout seul. Alors qu'il s'agit juste d'un coefficient à ajouter lors des mises à jours des paramètres, il permet d'obtenir une meilleure précision et de surélever ainsi la courbe de F-mesure. En fait son rôle est assez similaire à celui de l'apriori de parcimonie : comme on l'avait constaté quand on l'avait appliqué à la PLCA (figure 4.2 page 67), il a pour effet de rendre les activations plus parcimonieuses.
- Concernant l'apriori de ressemblance, son effet est un peu moins probant, qu'il soit utilisé seul, ou en combinaison avec un autre module. En revanche, c'est avec la combinaison des trois modules qu'on obtient la meilleure F-mesure pour une valeur de seuil $P_{\min}$ optimale.

D'après ces observations, et en considérant à la fois la qualité de l'estimation de hauteurs simples et le temps de calcul, on retiendra pour l'application du modèle BHAD à la transcription auto-




(a) Ajout d'un unique module (b) Combinaison de deux modules (c) Ajout de trois modules

**Figure 7.3** – Étude de l'influence des modules sur le modèle BHAD pour une tâche d'estimation de hauteurs multiples : résultats moyens en fonction du seuil de détection $P_{\text{min}}$.

matique (chapitre 8 page 125) trois systèmes :

- $B$, qui servira d'algorithme de référence et permettra d'évaluer les bénéfices de l'ajout de modules pour la transcription,
- $B - pf$ qui présente un excellent rapport performance/rapidité,
- $B - prf$ qui dans cette expérience présente la meilleure valeur maximale de F-mesure.

## 7.6 Conclusion

Nous avons présenté dans ce chapitre un nouveau modèle permettant de décomposer chaque colonne de CQT comme une somme de spectres harmoniques et de bruit. Chaque spectre harmonique possède sa propre enveloppe spectrale, dépendant du temps, ce qui rend le modèle BHAD très expressif, lui permettant ainsi de modéliser n'importe quelle note possédant des variations de hauteur et d'enveloppe spectrale. Nous avons également vu qu'il pouvait décrire des spectres légèrement inharmoniques, et ce même si les noyaux fixes sur lesquels le modèle repose sont strictement harmoniques. Nous avons ensuite expliqué comment appliquer les différents aprioris ou le frein sur les paramètres. Les tests préliminaires, à savoir l'estimation de hauteurs multiples, nous a permis de constater que si l'algorithme BHAD ne donnait pas de bons résultats tel quel, de par sa trop forte expressivité, l'ajout des différents modules permettait de le rendre bien plus performant. Trois instanciations de l'algorithme ont été sélectionnées pour l'application à la transcription automatique du prochain chapitre.

# Quatrième partie

# Applications

# Chapitre 8

# Application à la transcription automatique de musique

Ce chapitre est consacré à l'évaluation des systèmes de décomposition de CQT que nous avons élaborés dans les chapitres 6 et 7, dans une tâche de transcription automatique de musique polyphonique. Pour rappel, nous avons retenu trois instanciations pour chacun des modèles HALCA et BHAD : $H_4$, $H_4 - pf$, $H_4 - ptf$, $B$, $B - pf$ et $B - prf$ (*cf.* Tables 6.3 page 106 et 7.2 page 120). Nous commencerons par décrire l'étape de post-traitement permettant d'estimer l'ensemble des notes, par leur hauteur et leurs temps de début et de fin, à partir des valeurs des paramètres que nous fournissent nos algorithmes. Ensuite nous présenterons les métriques utilisées pour quantifier la qualité de l'estimation de transcription. L'étape de post-traitement dépend de paramètres que l'on doit fixer, et nous entrainerons donc dans la section suivante nos systèmes de transcription sur une base d'apprentissage, afin de les apprendre. Nous évaluerons ensuite les performances de nos algorithmes sur plusieurs bases d'évaluation, et nous les comparerons avec des algorithmes de l'état de l'art. Pour finir, nous présenterons certains des résultats obtenus lors de l'évaluation internationale MIREX 2012, pour laquelle nous avons soumis l'algorithme $B - pf$.

## 8.1 Post-traitement : détection des débuts et fins de notes

Les modèles HALCA et BHAD nous permettent d'obtenir des activations temps-fréquence $P_h(i,t)$ (dans le cas du modèle HALCA, on les obtient en sommant sur l'ensemble des sources $P_h(i,t) = \sum_s P_h(i,t,s)$) représentant la puissance des spectres harmoniques présents dans un signal pour chaque temps $t$ et chaque fréquence fondamentale $i$. C'est à partir de ces paramètres que nous allons pouvoir déduire une transcription :

- À partir des activations $P_h(i,t)$, nous souhaitons d'abord obtenir la puissance de chaque note MIDI en fonction du temps. Pour cela, on commence par détecter les pics de chaque colonne de $P_h(i,t)$. Au temps $t_0$, chaque pic $i_0$ sera ensuite associé à la note MIDI $n_0$

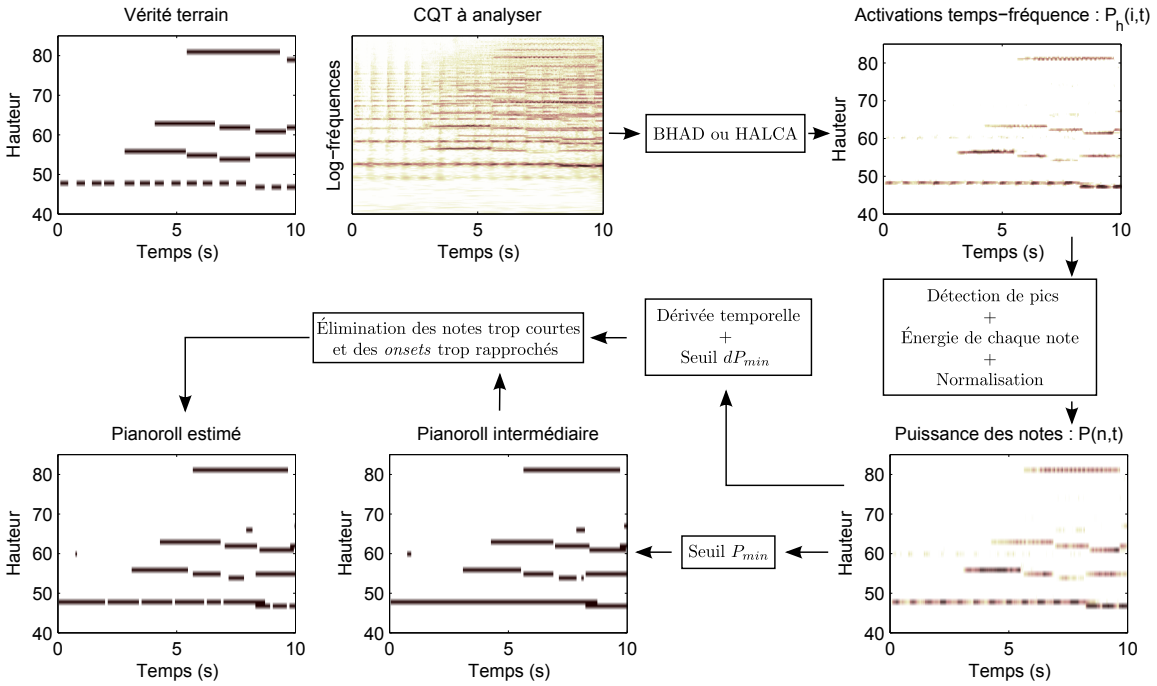



**Figure 8.1** – Système complet de transcription.

la plus proche. La puissance $P(n_0, t_0)$ de cette note sera alors déduite en sommant les coefficients de $P_h(i, t_0)$ pour les $i$ éloignés de moins d'un quart de ton de $i_0$. Puisque les CQTs avec lesquels nous travaillons ont trois points fréquentiels par demi-ton (*cf.* section 2.3 page 36), dans notre cas nous avons $P(n_0, t_0) = P_h(i_0, t) + P_h(i_0 - 1, t) + P_h(i_0 + 1, t)$.

- On normalise ensuite $P(n, t)$ de la manière suivante : $P(n, t) \leftarrow P(n, t)/\max_{n,t} P(n, t)$.
- Pour détecter les onsets et les offsets des notes, on procède en deux temps. Premièrement, un onset (resp. offset) est détecté pour chaque note $n$ dès que $P(n, t)$ devient plus grand (resp. plus petit) qu'un premier seuil $P_{\min}$ pendant plus de 70ms.
- Deuxièmement, afin de considérer le cas où une nouvelle note est jouée alors qu'une autre note de même hauteur est toujours active, ou le cas où l'on a des notes répétées liées, on propose une détection d'onset fondée sur la dérivée : pour chaque note active, un onset est détecté dès que la dérivée temporelle de $P(n, t)$ dépasse un certain seuil $dP_{\min}$ (pour éviter une surestimation des onsets, si deux onsets pour une même hauteur MIDI se retrouvent rapprochés de moins de 100ms, seul le premier est conservé).

Sur la figure 8.1, on a illustré les différentes étapes de ce post-traitement.

## 8.2 Évaluation d'une transcription estimée

Afin de quantifier la qualité d'une transcription on procède de la manière suivante. On suppose qu'une note de la transcription estimée est correcte s'il existe dans la vérité terrain une note

de même hauteur dont le temps d'attaque n'est pas éloigné de plus de 50 ms. Nous ne prenons pas en compte les temps de fin, car comme nous l'avions fait remarquer dans l'introduction (page 4), ils ne sont pas toujours identifiables de manière incontestable. Après avoir fait correspondre quand c'est possible chaque note de l'estimation de transcription avec une note de la vérité terrain, on peut alors dénombrer l'ensemble $TP$ (*True Positive*) des notes correctement estimées, l'ensemble $FP$ (*False Positive*) des notes estimées à tort (qui ne sont pas dans la vérité terrain) et enfin l'ensemble $FN$ (*False Negative*) des notes oubliées. Les trois métriques de rappel $\mathcal{R}$, précision $\mathcal{P}$ et F-mesure $\mathcal{F}$ [vR79] que nous avions introduites pour la tâche d'estimation *multipitch* du modèle HALCA (section 6.6.2 p.103), sont ici définies comme (la notation $|.|$ signifie *nombre cardinal*) :

$$\mathcal{R} = \frac{|TP|}{|TP| + |FN|},$$
$$\mathcal{P} = \frac{|TP|}{|TP| + |FP|},$$
$$\mathcal{F} = \frac{2\mathcal{R}\mathcal{P}}{\mathcal{R} + \mathcal{P}}.$$

$\mathcal{R}$ mesure alors le taux de notes correctement estimées sur le nombre de notes de la vérité terrain, $\mathcal{P}$ le taux de notes correctement estimées sur le nombre de notes de la transcription trouvée, et $\mathcal{F}$ est un score global permettant de synthétiser les deux précédentes mesures.

## 8.3 Apprentissage des seuils

Comme nous l'avons vu, le post-traitement proposé pour obtenir une transcription à partir des activations temps-fréquence dépend de deux seuils $P_{\min}$ et $dP_{\min}$. Comme nous souhaitons proposer des systèmes de transcription génériques, ces seuils doivent être prédéfinis. Pour cela nous entrainons nos algorithmes sur une base de données d'apprentissage. La base est la même que celle utilisée dans les sections 6.6.2 page 103 et 7.5 page 118 : $BD_{\text{app}}$ (annexe C page 171). Le critère de performance retenu est la F-mesure moyenne. Dans la Table 8.1, on liste pour chacun des algorithmes les valeurs des seuils optimaux trouvés, ainsi que les valeurs d'hyperparamètres.

Pour chacun de nos systèmes de transcription, il peut être intéressant de comparer sur cette même base d'apprentissage les résultats obtenus si l'on fixe le seuil $P_{\min}$ de sorte que la F-mesure soit maximale en moyenne, ou si $P_{\min}$ est choisi de manière optimale pour chaque fichier de $BD_{\text{app}}$. Les résultats sont consultables dans la Table 8.2. On observe clairement un gain significatif des performances quand le seuil est réglé au mieux pour chaque morceau. On peut en déduire qu'il n'existe pas un unique meilleur seuil pour tous les signaux de musique : une bonne perspective de recherche future sera donc d'imaginer un moyen d'estimer le seuil lors de l'analyse plutôt que de l'apprendre *a priori*.

| Algorithme | Modèle | $S$ | $\beta_{\text{parci}}$ | $\beta_{\text{frein}}$ | $\beta_{\text{temp}}$ | $\beta_{\text{res}}$ | $P_{\min}$ (dB) | $dP_{\min}$ |
|---|---|---|---|---|---|---|---|---|
| $H_4$ | HALCA | 4 | 0 | 0 | 0 | – | −23 | 0.018 |
| $H_4 - pf$ | HALCA | 4 | 0.02 | 5 | 0 | – | −20 | 0.018 |
| $H_4 - ptf$ | HALCA | 4 | 0.02 | 5 | $10^4$ | – | −19 | 0.018 |
| $B$ | BHAD | – | 0 | 0 | – | 0 | −13 | 0.018 |
| $B - pf$ | BHAD | – | 0.3 | 10 | – | 0 | −25 | 0.018 |
| $B - prf$ | BHAD | – | 0.3 | 10 | – | 0.5 | −21 | 0.018 |

**Table 8.1** – Valeur des hyperparamètres et des seuils pour chacun des algorithmes de transcription.

| | $H_4$ | $H_4 - pf$ | $H_4 - ptf$ | $B$ | $B - pf$ | $B - prf$ |
|---|---|---|---|---|---|---|
| $P_{\min}$ fixé (optimal en moyenne) | 50.8 | 54.3 | 54.1 | 36.4 | 51.7 | 53.0 |
| $P_{min}$ optimal pour chaque fichier | 57.3 | 60.4 | 60.4 | 45.0 | 58.7 | 58.7 |

**Table 8.2** – F-mesure (%) moyenne obtenue avec un seuil fixe optimal ou des seuils optimaux pour chaque fichier de $BD_{\text{app}}$.

## 8.4 Évaluation et comparaison avec des algorithmes de référence

Dans cette section, nous évaluons les performances de nos six systèmes de transcription, définis dans la Table 8.1, et les comparons avec deux algorithmes de référence : Vincent'10 [VBB10] et Dessein'12 [DCL12]. Nous avions déjà parlé de ces deux algorithmes dans l'état de l'art (sections 1.3.3 page 22 et 1.3.4 page 22). Dessein'12 est une méthode de décomposition de RTF$^+$ supervisée. En premier lieu un dictionnaire de spectres de notes de musique sont appris (ici ce sont des notes de piano). Ensuite, un spectrogramme d'entrée est décomposé sur ce dictionnaire, grâce à la minimisation d'une $\beta$-divergence entre les données et le spectrogramme reconstruit. Cet algorithme a été soumis en 2010 à la compétition internationale MIREX [Webc] et a obtenu de bons classements : troisième place pour la tâche de transcription sur l'ensemble des bases de données d'évaluation par exemple. Vincent'10 est un algorithme semi-supervisé de type NMF où les spectres de base ont une contrainte d'harmonicité (à l'instar de nos modèles) et de régularité d'enveloppe spectrale. De plus le signal est supposé redondant et une seule enveloppe spectrale par note est autorisée. La divergence utilisée pour l'estimation des paramètres du modèle est la $\beta$-divergence avec $\beta = 0,5$. Pour ces deux algorithmes de référence, le post-traitement des données de la décomposition pour obtenir une transcription est fondé sur un unique seuil de détection. Leur implémentation nous a aimablement été fournie par leurs auteurs respectifs.

Pour évaluer nos propres algorithmes et ceux de référence, nous avons constitué trois bases de données différentes : $BD_{\text{maps}}$, $BD_{\text{mirex}}$ et $BD_{\text{quasi}}$, toutes décrites en détails dans l'annexe C page 171. $BD_{\text{maps}}$ est un ensemble de 10 pièces de piano tirées de la base MAPS [EBD10]. Cinq d'entre elles sont synthétisées par un logiciel et les cinq autres enregistrées sur un vrai piano

| Algorithme | $\mathcal{R}$ (%) | $\mathcal{P}$ (%) | $\mathcal{F}$ (%) | Temps de calcul ($\times$ temps réel) |
|---|---|---|---|---|
| $H_4$ | 55.6 | 61.5 | 57.8 | 4.5 |
| $H_4 - pf$ | 52.3 | 70.9 | 59.4 | 9 |
| $H_4 - ptf$ | 54.9 | 73.6 | **61.8** | 27 |
| | | | | |
| $B$ | 56.1 | 41.9 | 47.5 | 2 |
| $B - pf$ | 52.8 | 71.7 | 60.0 | 3.5 |
| $B - prf$ | 51.6 | 76.7 | **60.6** | 13 |
| | | | | |
| Vincent'10 | 67.0 | 35.8 | **45.3** | 0.9 |
| Dessein'12 | 43.3 | 48.5 | 45.1 | 0.8 |

**Table 8.3** – Résultats moyens pour la base $BD_{\text{maps}}$.

| Algorithme | $\mathcal{R}$ (%) | $\mathcal{P}$ (%) | $\mathcal{F}$ (%) | Temps de calcul ($\times$ temps réel) |
|---|---|---|---|---|
| $H_4$ | 51.4 | 79.4 | 62.4 | 4.5 |
| $H_4 - pf$ | 45.7 | 84.6 | 59.3 | 9 |
| $H_4 - ptf$ | 51.7 | 84.6 | **64.2** | 27 |
| | | | | |
| $B$ | 55.5 | 69.0 | 61.5 | 2 |
| $B - pf$ | 51.3 | 83.7 | **63.6** | 3.5 |
| $B - prf$ | 47.4 | **88.2** | 61.6 | 13 |
| | | | | |
| Vincent'10 | 81.1 | 45.0 | **57.9** | 0.9 |
| Dessein'12 | 48.6 | 55.9 | 52.0 | 0.8 |

**Table 8.4** – Résultats moyens pour la base $BD_{\text{mirex}}$.

acoustique. Seules les 30 premières secondes de chaque pièce sont retenues. $BD_{\text{mirex}}$ n'est pas à proprement parler une base de données puisqu'elle ne contient qu'un unique morceau de 54 secondes : il s'agit d'une transcription pour quintette à vent d'un quatuor à corde de Beethoven, issue de la base de développement de MIREX [Webc]. Enfin $BD_{\text{quasi}}$ est un sous-ensemble de cinq fichiers de la base QUASI Transcription [Webf], une nouvelle base composée dans le cadre de cette thèse et du projet QUAERO [1]. Elle est constituée de morceaux de musique de différents genres (rock, reggae, chanson,...). Les scores moyens pour chaque base de données et pour chaque système de transcription, ainsi que les temps de calculs approximatifs respectifs [2], sont présentés dans les Tables 8.3, 8.4 et 8.5.

On peut d'abord noter qu'il n'y a pas un algorithme qui soit plus performant que les autres sur toutes les bases de données considérées. Pour $BD_{\text{maps}}$ les deux meilleurs algorithmes en terme de F-mesure moyenne sont $H_4 - ptf$ et $B - prf$. Ce sont les algorithmes qui sont les

1. `http://www.quaero.org`
2. On rappelle que les temps de calcul donnés ont été mesurés en utilisant une version 64 bits de Matlab, avec un processeur à deux cœurs de 3.1GHz.

| Algorithme | $\mathcal{R}$ (%) | $\mathcal{P}$ (%) | $\mathcal{F}$ (%) | Temps de calcul ($\times$ temps réel) |
|---|---|---|---|---|
| $H_4$ | 38.1 | 41.9 | 38.8 | 4.5 |
| $H_4 - pf$ | 37.9 | 50.3 | **41.5** | 9 |
| $H_4 - ptf$ | 36.8 | 49.6 | 40.7 | 27 |
| | | | | |
| $B$ | 39.7 | 32.9 | 32.9 | 2 |
| $B - pf$ | 40.0 | 52.0 | **43.1** | 3.5 |
| $B - prf$ | 34.3 | 46 | 37.3 | 13 |
| | | | | |
| Vincent'10 | 63.8 | 12.3 | 20.3 | 0.9 |
| Dessein'12 | 33.4 | 17.0 | **20.9** | 0.8 |

**Table 8.5** – Résultats moyens pour la base $BD_{\text{quasi}}$.

plus contraints et il semble donc que pour le piano, en plus de la parcimonie et du coefficient de freinage, les aprioris de continuité temporelle ou de ressemblance permettent d'obtenir de meilleurs résultats. On comprend facilement cela puisque les signaux de $BD_{\text{maps}}$ sont les plus « simples » : les spectres qui composent une note donnée ne varient pas beaucoup sur l'ensemble d'un morceau puisque les signaux sont constitués d'un unique instrument et que pour le piano, les notes ont un contenu spectral assez stable au cours du temps (pas de vibrato par exemple). On remarque quand même que l'apriori de continuité temporelle apporte bien plus au modèle HALCA que l'apriori de ressemblance n'apporte au modèle BHAD.

Les résultats pour $BD_{\text{mirex}}$ sont un peu déroutants puisque les modèles BHAD et HALCA sans aucun module donnent des résultats comparables aux algorithmes plus sophistiqués. On remarque même que $H_4$ est meilleur que $H_4 - pf$ en terme de F-mesure. En observant les différences de valeurs entre rappel et précision on comprend que pour ce morceau particulier, le seuil $P_{\text{min}}$ est bien mieux réglé pour $H_4$ et $B$ que pour nos autres systèmes. Cela confirme le défaut d'un post-traitement basé sur un seuil de détection pré-appris : on se doute que les systèmes avec modules auraient présenté de meilleurs résultats avec un seuil mieux réglé. Pour le modèle HALCA, le système $H_4 - ptf$ obtient les meilleurs résultats, aussi bien en rappel qu'en précision : il semble que l'ensemble des modules a permis de bien considérer la nature du signal. L'ajout de l'apriori de ressemblance au modèle BHAD ne semble en revanche pas bénéfique : comme le fichier de $BD_{\text{mirex}}$ contient cinq instruments différents, la supposition selon laquelle l'enveloppe spectrale d'un spectre harmonique d'une hauteur donnée varie peu au cours du temps n'est plus forcément appropriée.

La base $BD_{\text{quasi}}$ est constituée de signaux que l'on pourrait qualifier de plus difficiles à analyser, puisqu'ils incluent des voix chantées, des effets sonores (distorsions, réverbération, flanger...), un plus grand nombre d'instruments, ou encore de la batterie. Les performances sont donc moins bonnes en générale pour cette base de données. Les résultats donnés par nos algorithmes permettent de dresser quelques conclusions. Il semble tout d'abord que l'apriori de parcimonie et le frein sur les coefficients d'enveloppes aient une influence bénéfique quel

que soit le modèle considéré. En revanche, aussi bien l'apriori de continuité temporelle que de ressemblance font baisser la précision et le rappel par rapport aux systèmes $H_4 - pf$ et $B - pf$. Il apparait qu'en raison de la richesse des fichiers à analyser, une trop forte contrainte sur la décomposition ait un effet néfaste sur les performances. Ici, c'est d'ailleurs le modèle BHAD (plus expressif que le modèle HALCA) avec son instanciation $B - pf$ qui obtient la meilleure F-mesure.

Analysons maintenant les résultats des algorithmes de référence. D'une manière générale, on peut remarquer qu'ils sont en deçà des résultats de nos algorithmes, quelle que soit la base de données. L'observation de la supériorité systématique de $\mathcal{R}$ sur $\mathcal{P}$ pour le système Vincent'10 conduit à penser que le seuil de détection est fixé trop bas, et qu'il pourrait avoir de bien meilleurs résultats s'il était réglé différemment. Cependant nous avons pris le parti de ne pas régler ce seuil de manière optimale pour chaque base d'évaluation. Aussi, de la même manière que nous avons fixé une valeur de $P_{\text{min}}$, apprise sur une base différente, nous n'avons pas souhaité modifier l'implémentation des auteurs. Ce qu'il est intéressant d'observer, c'est que les performances entre nos algorithmes et ceux de référence sur $BD_{\text{mirex}}$ sont assez similaires, alors que sur les autres bases, et surtout sur $BD_{\text{quasi}}$, les nôtres sont significativement meilleures. Cela souligne la robustesse de nos modèles à la diversité des signaux musicaux que l'on peut rencontrer. Nous pensons que cela est dû au caractère assez expressif des modèles BHAD et HALCA. En particulier, du fait que les coefficients d'enveloppe sont complètement dépendants du temps, ils ne supposent pas une redondance intrinsèque aux signaux de musique. Nous pouvons quand même signaler la rapidité avec laquelle les algorithmes Vincent'10 et Dessein'12 s'exécutent.

## 8.5 Résultats MIREX 2012

Une compétition internationale, MIREX (*Music Information Retrieval Evaluation eXchange*) [Webc], portant sur de multiples tâches d'analyse de signaux musicaux, est organisée tous les ans. En 2012, nous avons soumis l'algorithme $B - fp$ pour la campagne d'évaluation portant sur l'estimation de hauteurs multiples et la transcription automatique. Nous précisons que l'algorithme soumis n'est pas exactement le même que celui détaillé dans la Table 8.1, de par des valeurs différentes d'hyperparamètres ($\beta_{\text{parci}} = 0.2$, $\beta_{\text{frein}} = 20$, $P_{\text{min}} = -21$ dB et $dP_{\text{min}} = 0.06$). Nous reportons dans cette section les résultats globaux obtenus en transcription, en terme de précision, rappel et F-mesure, quand seulement les onsets sont pris en compte (c'est le protocole d'évaluation que nous avons retenu dans ce chapitre). En tout, 9 algorithmes ont été évalués, provenant de 6 groupes d'auteurs différents. Dans la Table 8.6, on liste l'ensemble des algorithmes soumis, avec pour chacun d'eux une brève description (quand c'est possible). Les résultats sont compilés dans la Table 8.7. Notre algorithme prend la deuxième place du classement, en terme de F-mesure moyenne.

| Algorithme | Référence | Description |
|---|---|---|
| BD2 | [BD11] | Extension de la SIPLCA, avec des atomes fixes pré-appris sur des instruments de l'orchestre et du piano. |
| BD3 | [BD11] | Extension de la SIPLCA, avec des atomes fixes pré-appris sur du piano uniquement. |
| CPG1 | [CGE12] | Extension de [GE11] avec modélisation de l'évolution temporelle des notes. |
| CPG2 | [CGE12] | *Idem.* |
| CPG3 | [CGE12] | *Idem.* |
| FBR2 | Chapitre 7 page 111 | Algorithme $B - fp$, incluant le post-traitement proposé dans la section 8.1. |
| FT1 | – | – |
| KD3 | [Dre12] | Approche déterministe, fondée sur une détection de pics. |
| SB5 | [BS12] | Approche par réseau de neurones (conçu pour transcrire uniquement le piano). |

**Table 8.6** – Algorithmes soumis à MIREX 2012.

| Algorithme | $\mathcal{R}$ (%) | $\mathcal{P}$ (%) | $\mathcal{F}$ (%) |
|---|---|---|---|
| BD2 | 52.4 | 38.1 | 43.0 |
| BD3 | 46.8 | 38.2 | 41.1 |
| CPG1 | 14.5 | 54.5 | 21.9 |
| CPG2 | 15.1 | 54.0 | 22.5 |
| CPG3 | 19.9 | 51.5 | 27.3 |
| FBR2 ($B - pf$) | 71.6 | 55.3 | **61.3** |
| FT1 | 3.3 | 21.8 | 5.5 |
| KD3 | 65.2 | 64.7 | **64.6** |
| SB5 | 63.5 | 42.3 | 49.8 |

**Table 8.7** – Résultats obtenus à MIREX 2012.

# Chapitre 9

# Application à la séparation de sources

## 9.1 Introduction

Toutes nos recherches précédentes ont été menées dans le but de traiter la transcription automatique, mais nous proposons ici de les appliquer à une tâche différente : la séparation de sources. Cette tâche consiste à extraire d'un mélange, ici un enregistrement monophonique, les signaux temporels individuels des différentes sources ou instruments. La plupart des techniques de séparation audio s'appliquent dans le domaine temps-fréquence [FCC06, Vir07, OF10] selon un schéma classique : pour chaque point $\boldsymbol{X}_{ft}$ d'une RTF complexe, on estime les contributions $M_s(f,t)$ d'énergie de chaque source $s$, définissant ainsi un ensemble de masques temps-fréquence. Pour estimer le signal temporel d'une source, il suffit alors de multiplier $\boldsymbol{X}$ par le masque correspondant, puis d'inverser la RTF résultante. Bien entendu, cela suppose de travailler avec des RTFs inversibles, d'où la prédominance de l'utilisation de la TFCT. Cette technique de masquage peut trouver une interprétation théorique, en terme de filtrage de Wiener [BBG06, CPDG07, LBR11], pour peu que l'on soit dans des cadres probabilistes appropriés permettant de modéliser une RTF complexe ou un signal temporel. Mais nous la considérerons ici plutôt comme une méthode *ad hoc* de filtrage temps-fréquence.

La principale difficulté de ces techniques réside dans l'estimation des masques, et dans le cas sous-déterminé [1], cela ne peut pas se faire sans l'ajout de connaissances sur la nature ou le contenu du signal de mélange. Typiquement, si un certain instrument dans un mélange est harmonique, une information couramment utilisée pour estimer son masque est la hauteur des notes qu'il joue. On sait en effet facilement mettre en relation la hauteur (information sémantique) avec la fréquence fondamentale, qui est une des caractéristiques physiques du spectre (comme expliqué dans la section 1.2 page 16). Aussi, de nombreux systèmes se servent d'une partition

1. Quand on a moins d'observations que le nombre de sources, ce qui est notre cas puisqu'on travaille avec des signaux monophoniques

correspondant au signal d'entrée pour séparer les différentes sources [HDB11, SC12, EM12]. D'autres estiment en amont, ou conjointement, la trajectoire de la hauteur de la ou des sources à séparer [VMR08, DDR11, FM12].

Aujourd'hui, on sait inverser la CQT [SK10, Pra11, DHGV11], et cela permet d'ouvrir de nouveaux horizons pour les méthodes traitant la séparation de sources à travers l'estimation du contenu sémantique d'un signal et en particulier l'estimation des hauteurs de notes. En effet, on a vu qu'il était aisé de considérer la hauteur des spectres harmoniques dans une CQT grâce à l'utilisation de modèles avec invariance par translation (*cf.* section 2.3 page 36). Les éléments constituants un modèle de CQT, une fois estimés, peuvent alors directement être utilisés pour la création des masques, et la CQT inverse nous permet alors de retrouver les signaux temporels correspondants.

Ce chapitre est donc consacré à l'utilisation des modèles qui ont été introduits dans cette thèse pour la séparation de sources. Nous allons voir en réalité deux applications particulières : la création d'une interface graphique pour traiter la séparation de notes assistée par l'utilisateur, publiée dans [FBR12a], ainsi que l'extraction aveugle de la mélodie principale du reste de l'accompagnement dans un morceau, publiée dans [FLBR12]. Mais avant cela, nous menons une étude expérimentale sur la qualité de séparation par masquage temps-fréquence avec l'utilisation de la CQT.

## 9.2 Séparation de sources par masquage temps-fréquence : TFCT vs. CQT

Nous commençons donc par étudier et comparer les performances de séparation par masquage temps-fréquence, selon qu'on utilise la CQT ou la STFT. Pour cela, considérons un signal temporel résultant de la somme de deux signaux sources (il est possible de généraliser à plus de deux sources) :

$$x = x_1 + x_2. \tag{9.1}$$

Si l'on applique une transformation linéaire T et inversible, du domaine temporel au domaine temps-fréquence (typiquement une TFCT ou une CQT complexe), on a alors :

$$\mathrm{T}(x) = \mathrm{T}(x_1 + x_2) \tag{9.2}$$

soit

$$\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{X}_2. \tag{9.3}$$

Le principe du masquage temps-fréquence est de trouver les meilleurs masques $\boldsymbol{M}_1$ et $\boldsymbol{M}_2$ tels que (l'opérateur (.) est le produit terme à terme) :

$$\forall(f,t)\begin{cases} M_1(f,t) & \in [0,1] \\ M_2(f,t) & \in [0,1] \\ M_1(f,t)+M_2(f,t) & = 1 \end{cases}$$

et

$$\hat{x}_1 = \mathrm{T}^{-1}(\boldsymbol{X}.\boldsymbol{M}_1) \approx x_1,$$
$$\hat{x}_2 = \mathrm{T}^{-1}(\boldsymbol{X}.\boldsymbol{M}_2) \approx x_2.$$

Dans le cas où l'opérateur T est la TFCT, et selon certaines hypothèses de gaussianité, on peut prouver [BBG06] que les meilleurs masques au sens des moindres carrées sont donnés par :

$$\boldsymbol{M}_1 = \frac{|\boldsymbol{X}_1|^2}{|\boldsymbol{X}_1|^2 + |\boldsymbol{X}_2|^2},$$
$$\boldsymbol{M}_2 = \frac{|\boldsymbol{X}_2|^2}{|\boldsymbol{X}_1|^2 + |\boldsymbol{X}_2|^2}.$$

Dès lors, pour tester le filtrage temps-fréquence dans le cas où T est la CQT, on peut comparer les vraies sources avec celles estimées en utilisant ces mêmes masques idéaux. Les résultats obtenus sont alors appelées performances oracles.

L'évaluation que nous avons menée consiste à évaluer ces performances dans un cadre d'extraction de mélodie principale. Pour cela, nous disposons de douze extraits monophoniques, échantillonnés à 44.1 kHz, d'une durée allant de 11 à 45 secondes (340 s au total) et issus de la base de données *QUASI Separation* [Webd]. Pour chaque fichier, nous disposons des sources séparées de mélodie (notée $x_m$) et d'accompagnement (notée $x_a$) ainsi que du mélange ($x = x_m + x_a$). Les morceaux de la base *QUASI Separation* appartiennent à différents genres musicaux, comme le rock, le reggae ou encore la bossa nova. Pour chacun des fichiers, on estime les sources grâce à l'utilisation des masques idéaux :

$$\hat{x}_m = \mathrm{T}^{-1}\left(\frac{|\mathrm{T}(x_m)|^2}{|\mathrm{T}(x_m)|^2 + |\mathrm{T}(x_a)|^2}\mathrm{T}(x_m + x_a)\right),$$
$$\hat{x}_a = \mathrm{T}^{-1}\left(\frac{|\mathrm{T}(x_a)|^2}{|\mathrm{T}(x_m)|^2 + |\mathrm{T}(x_a)|^2}\mathrm{T}(x_m + x_a)\right),$$

où T est soit la CQT ou la STFT. Ensuite, on quantifie la qualité des estimations, en les comparant aux sources originales, grâce à la boîte à outils BSSEval [VFG06]. Cette dernière fournit trois métriques :

—∘ le rapport source à distorsion (SDR pour *Source to Distortion Ratio*) qui donne une

| | SDR mél. | SDR ac. | SIR mél. | SIR ac. | SAR mél. | SAR ac. |
|---|---|---|---|---|---|---|
| STFT | $11.9_{2.1}$ | $\mathbf{14.4}_{2.3}$ | $20.5_{2.8}$ | $19.3_{3.1}$ | $\mathbf{12.62}_{2.1}$ | $\mathbf{15.3}_{2.2}$ |
| CQT | $\mathbf{12.0}_{2.1}$ | $11.8_{1.3}$ | $\mathbf{21.9}_{2.9}$ | $\mathbf{20.3}_{3.0}$ | $12.58_{2.0}$ | $12.6_{1.2}$ |

**Table 9.1** – Moyennes et écarts types (en indice) en dB des SDR, SAR et SIR après estimation des sources de mélodie principale (mél.) et d'accompagnement (ac.) par masquage temps-fréquence (performances oracles).

mesure globale de qualité de séparation ;

- le rapport source à interférences (SIR pour *Source to Interference Ratio*) qui mesure la quantité d'interférences, pour chaque source estimée, provenant des autres sources ;
- le rapport source à artéfacts (SAR pour *Source to Artifact Ratio*) qui mesure la quantité d'artéfacts dans les sources estimées.

Ces valeurs sont calculées en dB et sont d'autant plus élevées que la séparation est performante.

La STFT est calculée avec des fenêtres de 2048 échantillons (46.4ms), se recouvrant sur un quart de leur taille (512 échantillons), et avec un nombre de points fréquentiels égal à 2048. La CQT (complexe) est calculée différemment des algorithmes de transcription (*cf.* section 2.3 page 36), puisqu'il faut choisir un pas temporel plus petit si l'on veut garder la propriété d'inversibilité. Aussi elle est calculée avec 3 points fréquentiels par demi-ton, pour des fréquences allant de 61.7 Hz à 7902.1 Hz, et avec un pas temporel de 4 ms. Si la bande de fréquences d'analyse pour la CQT est réduite, c'est pour des raisons de taille de CQT : si nous voulions considérer toute la bande audible ($f \in [20\text{Hz}, 20\text{kHz}]$), il faudrait un pas temporel très petit, d'où des grandes tailles pour la CQT. Pour ne pas perdre d'énergie lors de la séparation, on suppose alors que pour les bandes de fréquences $f < 61.7Hz$ et $f > 7902.1Hz$, l'énergie du signal appartient uniquement à la source d'accompagnement. Aussi, avant le processus de séparation, on filtre le signal de mélange avec un filtre coupe-bande, puis on rajoute le signal filtré à la source estimée $\hat{x}_a$ en fin de processus.

Les moyennes et les variances des résultats sont présentées dans la Table 9.1. On peut difficilement déduire des propriétés générales de ces résultats, suivant que l'on utilise la CQT ou la STFT, mais la relative similarité des performances montre qu'il est tout aussi pertinent d'utiliser la CQT pour des techniques de séparation par masquage temps-fréquence. A l'écoute, on peut remarquer que les interférences sont moins (resp. plus) présentes dans les graves (resp. les aigus) quand on utilise la CQT. Il est facile de comprendre pourquoi : la CQT a une bien meilleure résolution fréquentielle dans les basses fréquences que la STFT. Inversement, la résolution fréquentielle est plus faible dans le haut du spectre.

Puisque les résultats montrent que l'utilisation de la CQT est efficace pour la séparation par masque temps-fréquence, nous pouvons maintenant proposer deux applications de nos modèles à la séparation de sources. Dans la suite, nous notons $\boldsymbol{X}$ une CQT complexe, de coefficients $X_{ft}$, et tous les algorithmes de décomposition dont nous parlerons seront appliqués à la CQT positive $\boldsymbol{V}$, dont les coefficients sont définis par $V_{ft} = |X_{ft}|$.

## 9.3 Interface graphique pour la séparation de sources supervisée

Dans cette section nous expliquons comment les modèles imaginés dans ce mémoire peuvent être utilisés pour traiter la séparation de sources supervisée. Ce travail est inspiré de [DT12], qui propose une interface graphique (GUI pour *Graphical User Interface*) où l'utilisateur peut sélectionner la trajectoire de fréquence fondamentale d'une ligne mélodique d'un instrument monodique à séparer. De manière similaire, on propose ici de fournir à l'utilisateur une interface graphique avec laquelle il peut sélectionner les notes qu'il souhaite extraire du reste d'un signal audio. Nous nous contenterons ici de proposer un système fondé sur le modèle BHAD, mais il pourrait tout aussi bien reposer sur le modèle HALCA. Nous suggérons au lecteur de se reporter au chapitre 7 p.111 pour la définition du modèle BHAD.

### 9.3.1 GUI et sélection de notes

Le modèle BHAD offre une représentation mi-niveau pertinente puisque les activations temps-fréquence $P_h(i,t)$ représentent la puissance des notes harmoniques présentes dans le signal, en fonction du temps et de la hauteur. Grâce à Matlab, nous avons développé un GUI, où l'on peut visualiser ces activations temps-fréquence, et surligner les notes que l'on souhaite extraire. On peut voir à quoi ce GUI ressemble sur la figure 9.1. Le GUI est constitué des éléments suivants :

1. la représentation des activations $P_h(i,t)$, sur laquelle l'utilisateur peut sélectionner des notes ou éditer les paramètres $P_h(i,t)$ (fonctions *effacer* et *dessiner*) s'il estime que l'algorithme BHAD s'est trompé dans son estimation ;
2. le panneau de configuration, où l'utilisateur peut régler les hyperparamètres de l'algorithme BHAD (la valeur de $\beta_{\text{parci}}$, ici simplement nommée *Sparseness*, par exemple), lancer l'algorithme, séparer les notes sélectionnées, réinitialiser tous les paramètres et enfin changer le contraste de la représentation des activations ;
3. la barre d'outils, composée d'outils de base permettant de charger un nouveau fichier .wav ou une session de travail antérieure, de sauvegarder la session courante, d'explorer l'image, de sélectionner ou désélectionner les notes à extraire, d'écouter une note de musique dont la hauteur est la même que la note sélectionnée, d'éditer les activations temps-fréquence (les fonctions *effacer* et *dessiner* dont nous parlions), et enfin d'écouter le signal.

### 9.3.2 Modèle de source et masquage temps-fréquence

L'utilisateur, en surlignant les notes qu'il souhaite extraire grâce à l'outil de « sélection de note », définit un masque binaire $B(i,t)$ sur les activations temps-fréquence, égal à 1 si le point temps-fréquence $(i,t)$ est sélectionné, et 0 sinon. $B(i,t)$ pourra alors être utilisé pour définir les masques $\boldsymbol{M}_1$ et $\boldsymbol{M}_2$, correspondant respectivement à la source 1 (les notes sélectionnées) et la source 2 (le reste) :

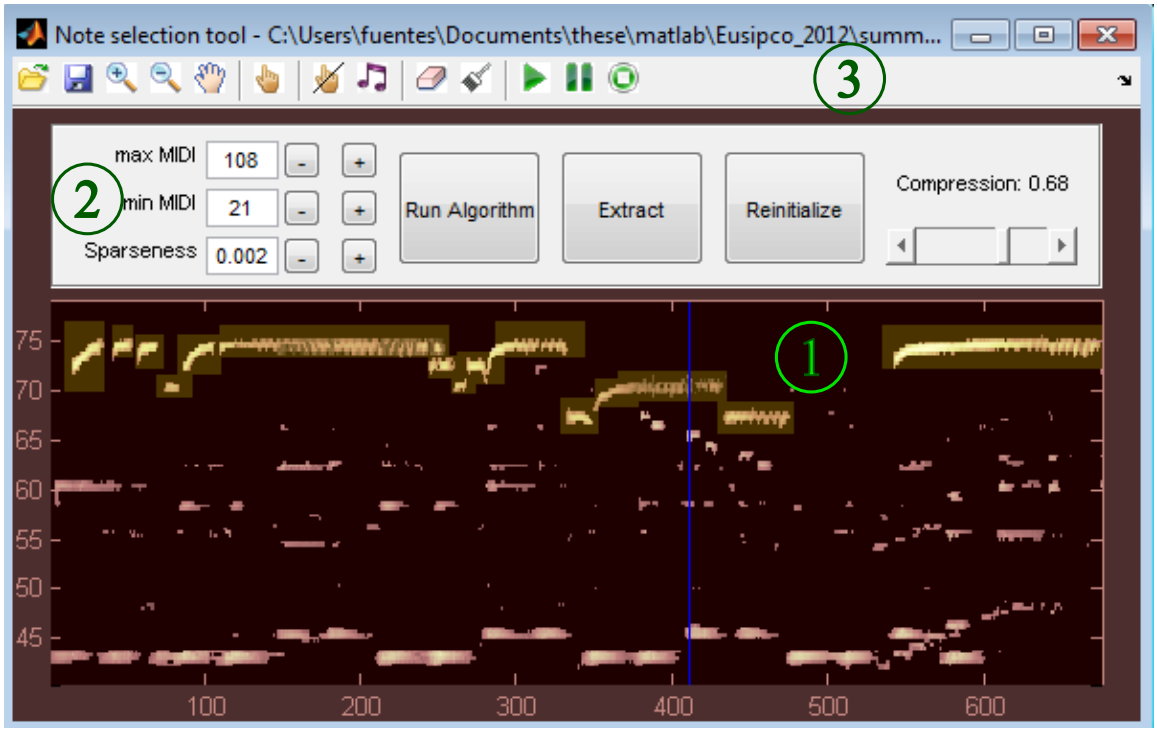



**Figure 9.1** – GUI pour la séparation supervisée de notes. Le fichier d'entrée est un extrait du standard de jazz *Summertime*, de George Gerswhin. Les zones surlignées correspondent aux notes sélectionnées par l'utilisateur.

$$M_1(f,t) = \frac{P(c=h)\sum_{i,z} B_1(i,t)P_h(z|i,t)P_h(f-i|z)}{P(f,t)}, \tag{9.4}$$

$$M_2(f,t) = \frac{P(c=n)P_n(f,t) + P(c=h)\sum_{i,z} B_2(i,t)P_h(z|i,t)P_h(f-i|z)}{P(f,t)}, \tag{9.5}$$

où $B_1(i,t)$ et $B_2(i,t)$ définissent respectivement les grandeurs $B(i,t)P_h(i,t)$ et $(1-B(i,t))\,P_h(i,t)$, $P(f,t)$ est donnée par l'équation (7.7) p.114, et $P_n(f,t)$ est donnée par l'équation (7.6) p.113. On peut vérifier que pour tout $(f,t)$, on a bien $M_1(f,t) + M_2(f,t) = 1$. Les estimations des signaux temporels des deux sources $\hat{x}_1$ et $\hat{x}_2$ sont alors données en appliquant les masques à la CQT complexe d'entrée $X_{ft}$ et en calculant la CQT inverse :

$$\hat{x}_1 = \mathrm{CQT}^{-1}\left(\boldsymbol{M}_1.\boldsymbol{X}\right), \tag{9.6}$$

$$\hat{x}_2 = \mathrm{CQT}^{-1}\left(\boldsymbol{M}_2.\boldsymbol{X}\right). \tag{9.7}$$

### 9.3.3 Évaluation

Afin d'évaluer les performances de séparation, notre système à été comparé à [DT12] dans une tâche d'extraction de mélodie principale (voix chantée). La base de données utilisée est composée

| | SDR | SIR | SAR |
|---|---|---|---|
| Système proposé | 4.0 | **16.6** | 4.5 |
| [DT12] | **5.2** | 16.2 | **6.0** |

**Table 9.2** – Moyennes en dB des SDR, SAR et SIR de l'extraction de mélodie principale assistée par l'utilisateur.

de 5 extraits de 15 s. issus du corpus de séparation de sources *QUASI Separation* [Webd] du projet QUAERO. Pour chaque fichier et chaque système, la trajectoire de hauteur de la mélodie principale a été localisée, sélectionnée grâce aux outils de sélection fournis par les deux GUIs, et extraite du reste de l'audio. Pour notre système, la CQT est calculée avec 3 points fréquentiels par demi-ton, pour des fréquences allant 27.5 Hz à 7040 Hz et avec un pas temporel de 4.4 ms. La qualité des estimations des signaux de mélodie est ensuite quantifiée selon les métriques fournies par BSSEval que nous avons présentées dans la section 9.2. Les résultats sont donnés dans la Table 9.2. Nous avons conscience que cette évaluation est un peu légère, puisque les qualités de séparation dépendent fortement de l'utilisateur, et que nous n'avons pas lancé une campagne d'évaluation incluant plusieurs évaluateurs. Elle nous permet cependant d'avoir une idée générale sur les performances de notre système. D'après les résultats, il semble que notre système soit légèrement moins efficace pour cette tâche d'extraction de mélodie. Nous pensons que cela est dû au modèle sous-jacent de [DT12], introduit dans [DDR11], qui est particulièrement conçu pour l'extraction de mélodie chantée. Par exemple, il permet de prendre en compte les sons non-voisés de la voix, contrairement au modèle BHAD. On peut noter cependant que notre système est plus générique puisqu'il permet de séparer des sources polyphoniques, là où l'autre système considère uniquement des sources monophoniques. L'implémentation Matlab du logiciel présenté dans cette section est disponible en ligne [2].

## 9.4 Extraction automatique de la mélodie principale

Cette section est consacrée à l'extraction automatique de la mélodie principale dans un enregistrement de musique. Pour cela, nous allons considérer une CQT d'entrée comme la somme d'un signal de mélodie et d'un signal d'accompagnement, chacune de ces composantes ayant un modèle propre : modèle HALCA pour la mélodie et modèle PLCA classique pour l'accompagnement. Des modèles hybrides similaires ont été également utilisés pour traiter la séparation parole/musique [RVCS10, DSC12]. Ici, nous commencerons par présenter le modèle complet de CQT. Nous décrirons ensuite l'algorithme permettant d'estimer ses paramètres pour enfin évaluer le système résultant.

2. `http://www.tsi.telecom-paristech.fr/aao/?p=756`

### 9.4.1 Modèle de CQT

Décrivons ici le modèle de $P(f,t)$, la distribution de probabilité représentant une CQT d'entrée. Puisque notre but est de séparer la mélodie principale de l'accompagnement, nous commençons par introduire une première variable cachée $c$ :

$$P(f,t) = P(c=a)P_a(f,t) + P(c=m)P_m(f,t). \tag{9.8}$$

$c$ représente alors les deux composantes ($c=a$ pour l'accompagnement et $c=m$ pour la mélodie), $P_a(f,t)_{(f,t)\in\mathbb{Z}\times[\![1,T]\!]}$ et $P_m(f,t)_{(f,t)\in\mathbb{Z}\times[\![1,T]\!]}$ représentent leur CQTs repsectives, et $P(c)_{c=a,m}$ leur énergie relative. Les notations $P_a(.)$ et $P_m(.)$ sont utilisées dans toute cette section pour $P(.|c=a)$ et $P(.|c=m)$. Nous présentons maintenant le modèle de chaque composante.

**Modèle d'accompagnement.** Pour l'accompagnement, le modèle utilisé est la PLCA classique (*cf.* section 2.1 page 29), où chaque colonne de CQT est modélisée comme une somme pondérée de spectres de base, ou noyaux :

$$P_a(f,t) = \sum_z P_a(z,t)P_a(f|z). \tag{9.9}$$

$P_a(z,t)$ représente les activations temporelles des noyaux, et $P_a(f|z)$ leur forme spectrale. Nous pouvons préciser qu'ici, le modèle PLCA n'est pas utilisé dans un but d'analyse, mais seulement de réduction de dimension.

**Modèle de mélodie.** Pour la mélodie principale, on utilise la partie harmonique du modèle HALCA (chapitre 6 page 87), avec une unique source. Ce modèle nous semble adapté puisqu'il permet de considérer les variations de fréquence fondamentale et d'enveloppe spectrale que l'on peut observer pour de nombreux instruments, dont la voix chantée. Au temps $t$, le spectre de la mélodie, représenté par $P_m(f,t)$, est décomposé comme une somme pondérée de $Z$ noyaux harmoniques à bande étroite fixés, notés $P_a(\mu|z)_{(\mu,z)\in[\![1,F]\!]\times[\![1,Z]\!]}$, convoluée par des activations temps-fréquence $P_m(i,t)_{(i,t)\in\mathbb{Z}\times[\![1,T]\!]}$. Les noyaux sont les mêmes que ceux utilisés dans le modèle HALCA. Leur coefficients de pondération, appelés coefficients d'enveloppe, sont notés $P_m(z|t)$. Comme la mélodie est portée par un instrument monodique, chaque colonne de $P_m(i,t)$ est un vecteur monomodal, la valeur du mode correspondant à la fréquence fondamentale de la mélodie. Le modèle s'écrit donc :

$$P_m(f,t) = \sum_{i,z} P_m(i,t)P_m(z|t)P_m(f-i|z). \tag{9.10}$$

### 9.4.2 Estimation des paramètres et algorithme

**Algorithme EM.** L'algorithme EM permet de trouver des mises à jour des paramètres faisant augmenter la vraisemblance des observations à chaque itération. Dans l'étape E, les distributions

*a posteriori* des variables latentes $i$, $z$, $c$ sont calculées grâce au théorème de Bayes :

$$P(z, c = a|f, t) = \frac{P(c = a)P_a(z, t)P_a(f|z)}{P(f, t)}, \tag{9.11}$$

$$P(i, z, c = m|f, t) = \frac{P(c = m)P_m(i, t)P_m(z|t)P_m(f - i|z)}{P(f, t)}, \tag{9.12}$$

où $P(f, t)$ est définie par les équations (9.8), (9.9) et (9.10). Lors de l'étape M, on maximise l'espérance conditionnelle de la log-vraisemblance des variables observées et cachées, ce qui permet d'obtenir les règles de mise à jour suivantes :

$$P(c = a) \propto \sum_{f,t,z} V_{ft} P(z, c = a|f, t), \tag{9.13}$$

$$P_a(z, t) \propto \sum_{f} V_{ft} P(z, c = a|f, t), \tag{9.14}$$

$$P_a(f|z) \propto \sum_{t} V_{ft} P(z, c = a|f, t), \tag{9.15}$$

$$P(c = m) \propto \sum_{f,t,i,z} V_{ft} P(i, z, c = m|f, t), \tag{9.16}$$

$$P_m(i, t) \propto \sum_{f,z} V_{ft} P(i, z, c = m|f, t), \tag{9.17}$$

$$P_m(z|t) \propto \sum_{f,i} V_{ft} P(i, z, c = m|f, t). \tag{9.18}$$

L'algorithme consiste à initialiser l'ensemble des paramètres, puis à itérer les équations (9.11), (9.12) et les différentes mises à jours (équations (9.13) à (9.18)).

**Algorithme de Viterbi et second tour d'algorithme EM.** Puisque nous souhaitons avoir des activations temps-fréquence qui soient monomodales pour chaque temps $t$, nous pourrions appliquer l'apriori de monomodalité (section 3.5 page 54) sur chaque colonne de $P_m(i, t)$. Cependant, nous voudrions prendre en compte le fait qu'on observe généralement une trajectoire plutôt régulière de la fréquence fondamentale de la mélodie, et comme l'apriori de monomodalité s'exécute de manière indépendante pour chaque $t$, il ne permet pas de considérer cette caractéristique. Nous optons ici pour une stratégie *ad hoc* mais efficace, similaire à celle proposée dans [DDR11]. D'abord un algorithme de Viterbi (strictement le même que celui proposé dans [DDR11, p. 570]) est appliqué aux activations temps-fréquence estimées : il permet de trouver la meilleure trajectoire de fréquence fondamentale, qui trouve un compromis entre forte énergie et chemin régulier. Ensuite, $P_m(i, t)$ est mis à zéro pour les couples $(i, t)$ se trouvant plus loin d'un quart de ton de cette trajectoire. Enfin, l'algorithme EM est relancé sur quelques itérations, afin de laisser les paramètres converger vers une nouvelle solution. La figure 9.2 illustre cette étape de lissage.

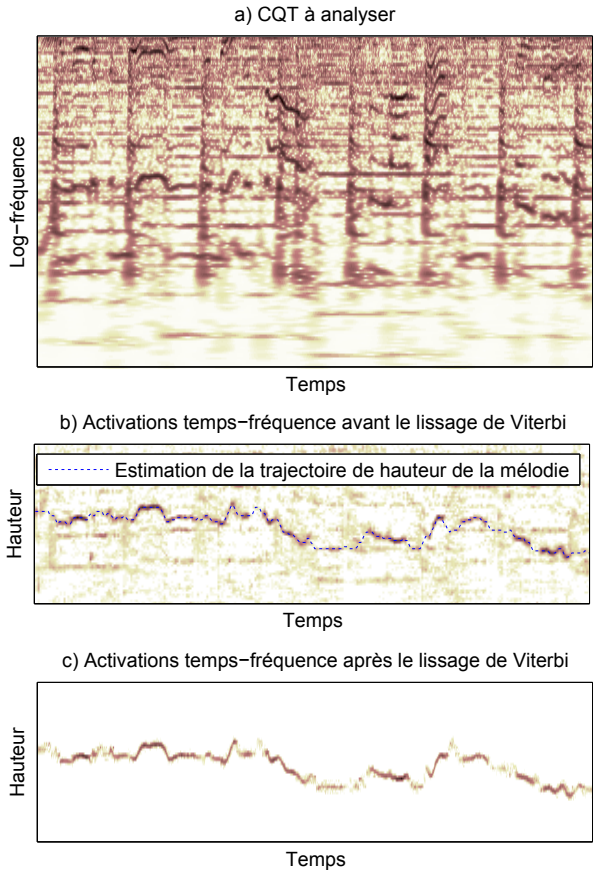



**Figure 9.2** – Illustration de l'algorithme de Viterbi.

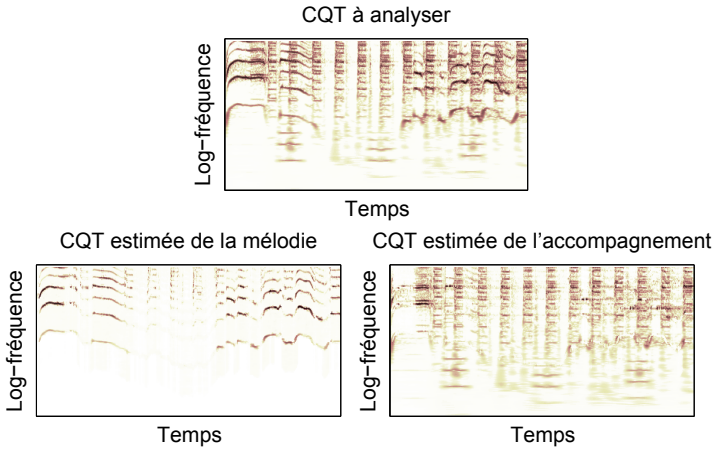



**Figure 9.3** – CQT d'entrée et estimation des CQT de chaque source.

**Détection de silence.** Le modèle présenté ici ne prend pas en compte la possible présence de silences dans la mélodie. Afin d'éviter la présence d'énergie dans l'estimation du signal de mélodie quand l'instrument principal ou le chant n'est pas actif, nous proposons un détecteur de silence. On applique d'abord un filtre passe bas de fréquence de coupure 1/10 Hz au signal de puissance de la mélodie estimée, définie par $P_m(t) = \sum_i P_m(i,t)$. Nous pouvons ensuite estimer la présence de silence quand le signal résultant passe en dessous d'un seuil, manuellement fixé à $-12$ dB. Nous appelons $\mathcal{T}_{silence}$ l'ensemble des temps où l'on a détecté un silence.

**Masquage temps-fréquence.** Une fois que tous les paramètres sont estimés, on peut définir des masques temps-fréquence $\boldsymbol{M}_m$ et $\boldsymbol{M}_a$ pour chaque source :

$$M_m(f,t) = \begin{cases} \frac{P(c=m)P_m(f,t)}{P(f,t)} & \text{si } t \notin \mathcal{T}_{silence} \\ 0 & \text{sinon} \end{cases},$$

$$M_a(f,t) = \begin{cases} \frac{P(c=a)P_a(f,t)}{P(f,t)} & \text{si } t \notin \mathcal{T}_{silence} \\ 1 & \text{sinon} \end{cases}.$$

Les signaux temporels des deux sources sont alors estimés en appliquant chaque masque à la CQT complexe $\boldsymbol{X}$ d'entrée, et en calculant la CQT inverse :

$$\hat{x}_m = \mathrm{CQT}^{-1}\left(\boldsymbol{M}_m.\boldsymbol{X}\right), \tag{9.19}$$

$$\hat{x}_a = \mathrm{CQT}^{-1}\left(\boldsymbol{M}_a.\boldsymbol{X}\right). \tag{9.20}$$

Sur la figure 9.3, on illustre l'estimation des CQTs des sources après application de l'algorithme, à savoir $|\boldsymbol{M}_m.\boldsymbol{X}|$ et $|\boldsymbol{M}_a.\boldsymbol{X}|$.

| | SDR mél. | SDR ac. | SIR mél. | SIR ac. | SAR mél. | SAR ac. |
|---|---|---|---|---|---|---|
| Système proposé | $\mathbf{4.6}_{2.2}$ | $\mathbf{7.2}_{2.4}$ | $\mathbf{12.6}_{4.0}$ | $\mathbf{11.1}_{3.3}$ | $\mathbf{5.9}_{1.4}$ | $10.1_{1.6}$ |
| [DDR11] | $4.4_{2.3}$ | $7.1_{2.9}$ | $12.2_{3.4}$ | $10.8_{3.5}$ | $5.7_{2.0}$ | $\mathbf{10.3}_{2.4}$ |

**Table 9.3** – Moyennes et écarts types (en indice) en dB des SDR, SAR et SIR après estimation automatique des sources de mélodie principale (mél.) et d'accompagnement (ac.).

### 9.4.3 Évaluation

Le système élaboré dans cette section a été testé sur la même base de données, et selon les mêmes paramètres pour le calcul de la CQT, que dans la section 9.2. Il a été de plus comparé à la méthode de l'état de l'art présentée dans [DDR11], dont l'implémentation Matlab est disponible sur le site de l'auteur [3]. On peut préciser qu'à l'instar de l'étude des performances oracles avec l'utilisation de la CQT (section 9.2 page 134), le signal de mélange est filtré en amont de notre algorithme par un filtre coupe-bande de fréquences de coupures 61.7 Hz et 7902.1 Hz. Le signal résultant, que nous supposons appartenir à la source d'accompagnement est ensuite ajouté à l'estimation $\hat{x}_a$. Les métriques fournies par BBSEval sont présentées dans la Table 9.3. Les deux algorithmes donnent des résultats assez proches, ce qui prouve que notre modèle est adapté à la tâche d'extraction de mélodie, et que l'utilisation de la CQT pour traiter la séparation de sources est pertinente. Des exemples sonores, ainsi que l'implémentation Matlab de cette méthode d'extraction sont disponibles en ligne [4].

## 9.5 Conclusion

Dans ce chapitre, nous avons montré que les modèles élaborés dans cette thèse pouvaient être appliqués à des tâches de séparation de sources. Nous avons en premier lieu mené une étude expérimentale pour savoir si l'utilisation de la CQT pour traiter la séparation de sources via masquage temps-fréquence était pertinente : les résultats ont été concluants. Nous avons ensuite présenté un logiciel permettant à l'utilisateur de sélectionner les notes qu'il voulait extraire. Ce logiciel est basé sur le modèle BHAD, mais il pourrait tout aussi bien reposer sur un autre algorithme de décomposition de CQT. Enfin, nous avons traité le problème d'extraction automatique de mélodie, en proposant un modèle hybride entre la PLCA et le modèle HALCA. Les résultats encourageants que nous avons obtenus permettent de faire valoir le lien existant entre l'analyse (BHAD et HALCA ont été conçus au départ pour répondre à la transcription automatique de musique) et la séparation.

3. `http://www.durrieu.ch/research/`
4. `http://www.tsi.telecom-paristech.fr/aao/?p=574`

# Conclusions et perspectives

Nous dressons ici le bilan de nos travaux ainsi que les conclusions que l'on peut tirer. Nous proposons ensuite quelques perspectives de recherches futures.

## Bilan et conclusions

L'analyse probabiliste en composantes latentes et sa version avec invariance par translation, présentées au chapitre 2, sont des outils pour l'analyse de données positives, comme les représentations temps-fréquences à coefficients positifs de signaux audio. Dans cette thèse, nous avons proposé des solutions pour mieux adapter, développer et enrichir ce cadre mathématique pour l'analyse de la transformé à Q constant (CQT) d'un signal musical.

En premier lieu nous avons proposé dans le chapitre 3 une batterie de nouveaux aprioris à appliquer sur les paramètres d'un modèle d'observation, quel qu'il soit. Ces aprioris permettent de mieux prendre en compte la nature des données que l'on analyse, et peuvent avoir deux effets sur l'algorithme d'estimation des paramètres : aider à converger vers une solution plus significative, et à ne pas rester bloquer dans des maxima locaux. Quatre types d'aprioris ont été introduits, suivant que l'on souhaite donner à un jeu de paramètres un caractère parcimonieux, une lente évolution temporelle, de faibles variations selon une certaine dimension ou un caractère monomodal. Nous avons proposé des exemples d'utilisation de ces aprioris avec l'analyse probabiliste en composantes latentes (PLCA) classique ou sa version avec invariance par translation (SIPLCA), et ils nous ont montré leur efficacité.

Dans le chapitre 4, une astuce simple a été introduite dans le but de freiner la vitesse de convergence d'un sous-ensemble de paramètres. Alors qu'il s'agit de simples coefficients à ajouter dans les mises à jour des paramètres, nous avons vu qu'ils pouvaient changer considérablement la manière dont l'algorithme converge. En particulier, nous avons montré que de tels coefficients de freinage pouvaient accentuer la parcimonie des paramètres non freinés, dans le cas où un modèle d'observation était sur-dimensionné. Nous avons aussi fait remarquer qu'ils pouvaient servir à encourager les paramètres freinés à être proches de leur initialisation à la fin de l'algorithme.

Le chapitre 5 a été consacré à l'introduction d'un nouveau méta-modèle de génération de données, appelé LCATS (*Latent Component Analysis with Temporal Structure*), permettant de considérer une structure temporelle pour un ensemble de variables cachées. Il a été décliné

selon deux cas particuliers. Le premier, le cas indépendant, nous a juste permis de vérifier si le modèle LCATS était une généralisation de la PLCA ou pas. Il s'est avéré qu'il ne l'était pas tout à fait en raison d'un coefficient de freinage qui restait dans une des mises à jour. Une deuxième déclinaison que nous avons proposée a été le cas markovien (MLCATS), où plutôt que de considérer directement la probabilité des variables cachées à chaque temps, nous avons modélisé leurs probabilités de transition entre deux instants successifs. Nous avons réussi à dériver l'algorithme Espérance-Maximisation de manière efficace (linéaire selon la taille de la dimension temporelle) et nous avons proposé un exemple simple d'utilisation dans lequel une initialisation intelligente des probabilités de transition permettait d'obtenir naturellement une évolution régulière des activations temporelles des atomes résultantes.

Après avoir présenté tous ces modules à appliquer à un sous-ensemble de paramètres dans un modèle de type PLCA, les chapitres 6 et 7 ont été consacrés à deux manières de décomposer la CQT d'un signal musical : les modèles HALCA (*Harmonic Adaptive Latent Component Analysis*) et BHAD (*Blind Harmonic Adaptive Decomposition*). Ils permettent d'analyser les structures harmoniques dans les signaux de musique. Leur caractéristique principale est de ne supposer aucune redondance dans le signal, contrairement à la SIPLCA par exemple. Aussi, chaque colonne de CQT est analysée indépendamment et cela nous permet facilement de pouvoir prendre en compte les notes possédant des variations de hauteur et d'enveloppe spectrale. En revanche, les aprioris, si l'on en fait l'utilisation, s'appliquent sur l'ensemble de tous les paramètres et prennent donc en compte la dimension temporelle du signal. Le modèle HALCA (chapitre 6) est fondé sur une notion de sources : les notes jouées à un temps donné par chacune d'entre elles possèdent une même forme spectrale. Après expérimentation, on s'est rendu compte que les sources n'incarnaient pas vraiment des instruments de musique spécifiques dans un mélange mais plutôt des méta-instruments, qui n'avaient pas vraiment de sens sémantique. On a alors remis en cause cette notion de sources pour proposer le modèle BHAD (chapitre 7), qui décompose chaque colonne de CQT comme une somme de spectres harmoniques de toutes les hauteurs possibles, chacun de ces spectres possédant une enveloppe spectrale propre. Afin de comprendre le comportement de ces deux modèles ainsi que l'influence des modules que l'on peut ajouter (aprioris et frein), nous les avons soumis à une tâche d'estimation de hauteurs multiples. Nous avons, grâce à ces tests préliminaires, retenu trois instanciations différentes par modèle pour les appliquer à la transcription automatique.

Cette dernière tâche a été le sujet du chapitre 8. Nous y avons d'abord expliqué comment traiter les données fournies par nos algorithmes pour obtenir des fichiers de transcription. Ce post-traitement dépend de deux seuils qui ont été appris sur une base de données d'évaluation. Ensuite l'ensemble des systèmes de transcription ainsi définis ont été évalués et comparés avec deux algorithmes de référence sur trois bases de données. L'ensemble des résultats obtenus ont permis de tirer deux conclusions principales. D'abord, l'hypothèse de redondance intrinsèque dans les signaux musicaux n'est pas nécessaire pour avoir de bons résultats, en témoignent les performances de nos algorithmes, même sur des enregistrements d'instrument solo comme la

base de piano $BD_{\text{maps}}$. Il semblerait même qu'elle soit néfaste quand les signaux à analyser se complexifient, avec le nombre ou le type d'instruments qui les composent. Aussi, l'apriori de ressemblance réduit les performances de l'algorithme BHAD sur les bases plus difficiles à analyser $BD_{\text{mirex}}$ et $BD_{\text{quasi}}$. La deuxième conclusion est le défaut des systèmes de transcription fondés sur des seuils pré-appris de détection d'onsets. Il semble en effet que les seuils optimaux soient différents pour chaque fichier à analyser, et on gagnerait beaucoup à imaginer un post-traitement adaptatif, ou alors des modèles incluant l'estimation des onsets.

Enfin, dans le chapitre 9, nous avons proposé des moyens d'utiliser nos algorithmes d'analyse pour traiter le problème de la séparation de sources. Nous avons tiré profit des récentes avancés sur l'inversibilité de la CQT pour montrer que la technique du masquage temps-fréquence pouvait être utilisée avec cette représentation temps-fréquence. Nous avons ainsi développé une interface graphique permettant à un utilisateur de sélectionner les notes qu'il souhaite extraire dans un signal audio, grâce à la décomposition proposée par l'algorithme BHAD. Nous avons également proposé une combinaison du modèle HALCA et de la PLCA afin d'extraire automatiquement la mélodie principale du reste de l'accompagnement dans un morceau de musique. Les résultats prometteurs que nous avons obtenus nous encouragent à explorer ce lien entre analyse et séparation.

## Perspectives

La recherche en analyse et traitement automatique des signaux musicaux a encore de belles années devant elle. Pour clôturer cette thèse, nous proposons ici quelques pistes à explorer pour poursuivre nos travaux.

### Concernant les aprioris

Il n'existe pas de solution analytique à l'étape du maximum *a posteriori* pour les aprioris de continuité temporelle, de ressemblance et de monomodalité (*cf.* chapitre 3 page 41). Aussi nous avons proposé pour ces aprioris des méthodes du point fixe qui, dans la pratique, semblent converger systématiquement vers une solution. Nous n'avons pas encore réussi à obtenir des preuves de convergence, mais c'est un problème que nous souhaiterions tenter de résoudre dans un travail futur.

### MLCATS : applications et améliorations

Le modèle MLCATS introduit dans le chapitre 5 n'a pas encore été utilisé pour une application précise, en raison de sa relative nouveauté. Nous proposons plusieurs pistes. D'abord pour la transcription ou l'estimation de hauteurs multiples, nous pourrions comparer les résultats obtenus avec ceux de la PLCA classique, et étudier si on obtient un gain de performance. Nous avons aussi annoncer dans la section 5.5 page 78 qu'il pouvait se substituer à n'importe quel

ensemble de paramètres bidimensionnel d'un modèle de type PLCA, et nous pourrions ainsi l'appliquer par exemple aux activations temps-fréquence du modèle BHAD. Enfin, le système d'extraction de mélodie de la section 9.4 page 139 inclue une étape *ad hoc* de lissage de Viterbi permettant de trouver la meilleure trajectoire de hauteur pour la mélodie, en termes d'énergie et de régularité. Puisque MLCATS permet de modéliser les transitions entre variables cachées au cours du temps, il semble que ce serait une bonne piste de le substituer aux activations temps-fréquence de la mélodie, en interdisant des transitions entres hauteurs trop éloignées.

Mis à part les applications, nous avons mis le doigt sur le caractère sur-paramétré de ce modèle MLCATS. Une piste de recherche serait donc de trouver une paramétrisation des matrices de transition dépendant de moins de paramètres, tout en conservant l'expressivité du modèle.

### Multiplier les niveaux sémantiques

Dans le modèle BHAD (chapitre 7 page 111), il faut plusieurs spectres harmoniques de hauteurs $i$ adjacentes pour modéliser le spectre d'une note réelle. Ces spectres représentent donc des « bouts » de notes et leur niveau sémantique s'en trouve réduit. On peut alors se demander s'il ne serait pas intéressant d'introduire une variable latente $n$ représentant vraiment une note sur l'échelle MIDI et de conditionner $i$ à cette nouvelle variable, à l'image du modèle proposé dans [BD11]. Cela permettrait au modèle d'être plus intelligemment manipulé : par exemple, si l'on souhaite exploiter le caractère parcimonieux généralement observé dans les pianorolls de morceaux de musique, il serait plus pertinent d'appliquer l'apriori de parcimonie sur la distribution des notes $n$ plutôt que sur celle des hauteurs $i$. Pour aller plus loin, on pourrait également introduire une variable cachée de niveau sémantique encore plus élevé, représentant un chroma, et conditionner $n$ à cette dernière. Il serait alors possible par exemple d'imaginer des modèles d'accords et ainsi prendre en considération des aspects théoriques d'harmonie dans la musique. Si nous ne donnons ici que des exemples spécifiques, cette idée de multiplier les niveaux sémantiques dans les décompositions de signaux audio, et d'imaginer donc des représentations « multi-niveaux » et non juste « mi-niveau » nous semble être une piste à explorer.

### Détection d'onsets

Une des conclusions des résultats de la tâche de transcription (chapitre 8) est le défaut de robustesse des algorithmes proposés aux seuils de détection d'onsets lors du post-traitement. Une réponse possible pour pallier ce problème serait d'imaginer un post-traitement adaptif, ou les seuils sont estimés en fonction du signal à analyser. Par exemple, pour l'estimation du seuil $P_{\min}$ dans le cas du modèle BHAD, on pourrait imaginer un algorithme dans lequel on augmente petit à petit sa valeur, en mettant à zéro les paramètres d'activations temps-fréquence $P_h(i,t)$ qui passent sous le seuil. On s'arrêterait alors si la vraisemblance des observations baisse de manière trop évidente.

### Modèles plus réalistes pour la séparation

Pour finir, nous pouvons dire un mot sur les applications à la séparation de sources que nous avons faites de nos modèles. Même si les résultats sont prometteurs, nos modèles d'analyse n'ont pas été conçus originellement pour la séparation. En particulier, nous modélisons les spectres de notes harmoniques de manière un peu approximative (ce qui semble suffisant pour l'analyse) sans prendre en compte certaines caractéristiques, comme les lobes secondaires des spectres des sinusoïdes, ou la présence de bruit dans les signaux d'instruments harmoniques. Afin d'améliorer les performances de séparation, nous pensons qu'il serait bénéfique de proposer des modèles plus réalistes de spectres.

# Bibliographie

[AP04] S.A. Abdallah et M.D. Plumbley : Polyphonic music transcription by non-negative sparse coding of power spectra. *In Proc of ISMIR*, pages 318–325, Barcelone, Espagne, 2004.

[AP06] S.A. A Abdallah et M.D. D Plumbley : Unsupervised analysis of polyphonic music by sparse coding. *IEEE transactions on neural networks*, 17(1):179–196, janvier 2006.

[BBG06] L. Benaroya, F. Bimbot et R. Gribonval : Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):191–199, janvier 2006.

[BBV10a] R. Badeau, N. Bertin et E. Vincent : Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization. *IEEE transactions on neural network*, 21(12):1869–81, décembre 2010.

[BBV10b] N. Bertin, R. Badeau et E. Vincent : Enforcing Harmonicity and Smoothness in Bayesian Non-negative Matrix Factorization Applied to Polyphonic Music Transcription. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):538–549, 2010.

[BD11] E Benetos et S Dixon : Multiple-instrument polyphonic music transcription using a convolutive probabilistic model. *In Proc. of SMC*, pages 19–24, Padoue, Italie, 2011.

[Bea03] M. Beal : *Variational Algorithms for Approximate Bayesian Inference*. Thèse de doctorat, Univ. College of London, 2003.

[Ber09] N. Bertin : *Les factorisations en matrices non-négatives. Approches contraintes et probabilistes, application à la transcription automatique de musique polyphonique.* Thèse de doctorat, Institut Mines-Télécom, Télécom ParisTech, 2009.

[BEZ08a] A M Bruckstein, M Elad et M Zibulevsky : On the Uniqueness of Nonnegative Sparse Solutions to Underdetermined Systems of Equations. *IEEE Trans. IT*, 54(796):4813–4820, 2008.

[Bra99] M. Brand : Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11(5):1155–1182, 1999.

[Bro91] J. Brown : Calculation of a constant Q spectral transform. *JASA*, 89(1):425–434, 1991.

[BS12] S. Böck et M. Schedl : Polyphonic Piano Note Transcription with Recurrent Neural Networks. *In Proc. of ICASSP*, Kyoto, Japon, 2012.

[Cem04] A.T. Cemgil : *Bayesian Music Transcription*. Thèse de doctorat, Radboud University Nijmegen, 2004.

[CGE12] Z. Chen, G. Grindlay et D.P.W. Ellis : Transcribing multi-instrument polyphonic music with transformed eigeninstrument whole-note templates. *In MIREX*, 2012.

[CGH+96] R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey et D.E. Knuth : On the Lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[CJ09] M.G. Christensen et A. Jakobsson : *Multi-Pitch Estimation*. Synthesis Lectures on Speech and Audio Processing. Morgan and Claypool Publishers, 2009.

[CK08] A. Chaigne et J. Kergomard : *Acoustique des instruments de musique*. Echelles. BELIN, 2008.

[Con06] A Cont : Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical HMMs. *In Proc. of ICASSP*, Toulouse, France, 2006.

[CPDG07] A.T. Cemgil, P. Peeling, O. Dikmen et S. Godsill : Prior Structures for Time-Frequency Energy Distributions. *In Proc. of WASPAA*, pages 151–154, New Paltz, New York, USA, 2007.

[CZA06a] A. Cichocki, R. Zdunek et S. Amari : Csiszar's divergences for non-negative matrix factorization : Family of new algorithms. *In Proc. of LVA/ICA*, pages 32–39, Charleston, SC, USA, 2006.

[CZA06b] A. Cichocki, R. Zdunek et S. Amari : New algorithms for nonnegative matrix factorization in applications to blind source separation. *In Proc. of ICASSP*, volume 5, pages 32–39, Toulouse, France, 2006.

[dCK02] A. de Cheveigné et H. Kawahara : YIN, a fundamental frequency estimator for speech and music. *JASA*, 111(4):1917–1930, 2002.

[DCL12] A. Dessein, A. Cont et G. Lemaitre : Real-time detection of overlapping sound events with non-negative matrix factorization. *In* F Nielsen et R Bhatia, éditeurs : *Matrix Information Geometry*. Springer, 2012.

[DDR11] J.-L. Durrieu, B. David et G. Richard : A musically motivated mid-level representation for pitch estimation and musical audio source separation. *Selected Topics in Signal Processing, IEEE Journal of*, 5(6):1180–1191, 2011.

[Dem77] A.P. P Dempster : Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. ...*, 39(1):1–38, 1977.

[Dev90] D. Devie : *Le tempérament musical, philosophie, histoire, théorie et pratique*. Société de musicologie du Languedoc, 1990.

[DHGV11] M. Dörfler, N. Holighaus, T. Grill et G. Velasco : Constructing an Invertible Constant-Q Transform with Nonstationary Gabor Frames. *In Proc. of DAFX*, pages 93–99, Paris, France, 2011.

[Dre12] K. Dressler : Multiple fundamental frequency extraction for MIREX 2012. *In MIREX*, 2012.

[DSC12] C. Demir, M. Saraçlar et A.T. Cemgil : Catalog-Based Single-Channel Speech-Music Separation with the Itakura-Saito Divergence. *In Proc. of EUSIPCO*, Bucarest, Romanie, 2012.

[DT12] J.-L. Durrieu et J.-P. Thiran : Musical audio source separation based on user-selected F0 track. *In Proc. of LVA/ICA*, Tel-Aviv, Israël, 2012.

[DV05] R.A. Dobie et S.B. Van Hemem : *Hearing loss : determining eligibility for Social Security benefits*. National Academies Press, 2005.

[EBD10] V. Emiya, R. Badeau et B. David : Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. *IEEE Trans. on Audio, Speech, and Language Processing.*, 18(6):1643–1654, 2010.

[EK01] S. Eguchi et Y. Kano : Robustifying maximum likeli-hood estimation. Rapport technique, Institute of Statistical Mathematics, Tokyo, Japon, 2001.

[EM12] S. Ewert et M. Müller : Score Informed Source Separation. *In* M. Müller, G. Masataka et M. Schedl, éditeurs : *Multimodal Music Processing*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012.

[Emi08] V. Emiya : *Transcription automatique de la musique de piano*. Thèse de doctorat, Institut Mines-Télécom, Télécom ParisTech, 2008.

[FBD09] C. Févotte, N. Bertin et J.-L. Durrieu : Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis. *Neural Computation*, 21(3):793–830, 2009.

[FBR11a] B. Fuentes, R. Badeau et G. Richard : Adaptive harmonic time-frequency decomposition of audio using shift-invariant PLCA. *In Proc. of ICASSP*, pages 401–404, Prague, République Tchèque, 2011.

[FBR11b] B. Fuentes, R. Badeau et G. Richard : Analyse des structures harmoniques dans les signaux audio : modéliser les variations de hauteur et d'enveloppe spectrale. *In GRETSI*, Bordeaux, France, 2011.

[FBR12a] B Fuentes, R Badeau et G Richard : Blind Harmonic Adaptive Decomposition Applied to Supervised Source Separation. *In Proc. of EUSIPCO*, pages 93–99, Bucarest, Romanie, 2012.

[FBR13] B. Fuentes, R. Badeau et G. Richard : Harmonic Adaptive Latent Component Analysis of Audio and Application to Music Transcription. *IEEE Trans. on Audio, Speech and Language Processing*, 21(9):1854–1866, 2013.

[FCC06] D. Fitzgerald, M. Cranitch et E. Coyle : Sound Source Separation using Shifted Non-negative Tensor Factorisation. *In Proc. of ICASSP*, volume 5, pages 653–656, Toulouse, France, 2006.

[FH94] J.A. Fessler et A.O. Hero : Space-alternating generalized expectation maximization algorithm. *IEEE Transaction on Signal Processing*, 42(10):2664–2677, 1994.

[FI11] C. Févotte et J. Idier : Algorithms for nonnegative matrix factorization with the $\beta$-divergence. *Neural Computation*, 23(9):2421–2456, 2011.

[FLBR12] B Fuentes, A Liutkus, R Badeau et G Richard : Probabilistic Model for main melody extraction using constant-Q transform. *In Proc. of ICASSP*, pages 5357–5360, Kyoto, Japon, 2012.

[Fle64] H. Fletcher : Normal Vibration Frequencies of a Stiff Piano String. *JASA*, 36(1): 203–209, 1964.

[FM12] D Fourer et S Marchand : Informed Multiple-F0 Estimation Applied to Monaural Audio Source Separation. *In Proc. of EUSIPCO*, numéro 1, Bucarest, Romanie, 2012.

[FP12] T. Fillon et J. Prado : A Flexible Multi-Resolution Time-Frequency Analysis Framework. *In Proc. of ISSPA*, numéro 2, Montreal, Canada, 2012.

[Fue13] B. Fuentes : Code Matlab en ligne. http ://www.tsi.telecom-paristech.fr/aao/2012/12/19/Fuentes2013_PhD/, 2013.

[GE10] G. Grindlay et D.P.W. P W Ellis : A probabilistic subspace model for multi-instrument polyphonic transcription. *In Proc. of ISMIR*, pages 21–26, Utrecht, Pays Bas, 2010.

[GE11] G Grindlay et D P W Ellis : Transcribing Multi-Instrument Polyphonic Music With Hierarchical Eigeninstruments. *J. Sel. Topics Signal Processing*, 5(6):1159–1169, 2011.

[GN03] M. Goto et T. Nishimura : RWC Music Database : Music Genre Database and Musical Instrument Sound Database. *In Proc. of ISMIR*, Baltimore, Maryland, USA, 2003.

[HBD11a] R. Hennequin, R. Badeau et B. David : Beta-divergence as a subclass of Bregman divergence. *IEEE Signal Processing Letters*, 18(2):83–86, 2011.

[HBD11b] R. Hennequin, R. Badeau et B. David : NMF with Time-frequency activations to model non stationary audio events. *IEEE Trans. on Audio Speech and Language Processing*, 19(4):744–753, 2011.

[HBD11c] R. Hennequin, R. Badeau et B. David : Scale-invariant probabilistic latent component analysis. *In Proc. of WASPAA*, New Paltz, New York, USA, 2011.

[HDB11] R. Hennequin, B. David et R. Badeau : Score informed audio source separation using a parametric model of non-negative spectrogram. *In Proc. of ICASSP*, numéro 1, Prague, République Tchèque, 2011.

[Hen11] R. Hennequin : *Décomposition de spectrogrammes musicaux informée par des modèles de synthèse spectrale.* Thèse de doctorat, Institut Mines-Télécom, Télécom ParisTech, 2011.

[Hof01] T. Hofmann : Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.

[Hoy04] P.O. O Hoyer : Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[HR09] N. Hurley et S. Rickard : Comparing Measures of Sparsity. *Information Theory, IEEE Transactions on*, 55(10):4723–4741, 2009.

[IS68] F. Itakura et S. Saito : Analysis synthesis telephony based on the maximum likelihood method. *In In 6th International Congress on Acoustics*, pages C–17–C–20, 1968.

[Joh88] J.D. Johnston : Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 6(2):314–323, 1988.

[JS76] R.I. Jennrich et P.F. Sampson : Newton-Raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics*, 18(1):11–17, 1976.

[KD06] A. Klapuri et M. Davy : *Signal processing methods for music transcripion.* Springer, 2006.

[KL51] S. Kullback et R.A. Leibler : On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[KNS07] H. Kameoka, T. Nishimoto et S. Sagayama : A multipitch analyser based on Harmonic Temporal Structured Clustering. *IEEE Trans. on Audio, Speech and Language Processing*, 15(3):982–994, 2007.

[KT51] H.W. Kuhn et A.W. Tucker : Nonlinear programming. *In* J Neyman, éditeur : *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, Berkeley, California, 1951.

[LBR11] A. Liutkus, R. Badeau et G. Richard : Gaussian Processes for Underdetermined Source Separation. *Signal Processing, IEEE Transactions on*, 59(7):3155–3167, 2011.

[Lev07] P. Leveau : *Décompositions parcimonieuses structurées : applicationà la représentation objet de la musique.* Thèse de doctorat, Université Pierre et Marie Curie, 2007.

[Lin07] C.-J. Lin : Projected gradient methods for non-negative matrix factorization. Rapport technique, Rapport technique, Department of Computer Science, National Taiwan University, 2007.

[LS99] D.D. Lee et H.S. Seung : Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[LVRD08] P. Leveau, E. Vincent, G. Richard et L. Daudet : Instrument-Specific Harmonic Atoms for Mid-Level Music Representation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(1):116–128, 2008.

[Mek01] W.R. Mekwi : Iterative Methods for Roots of Polynomials. Mémoire de D.E.A., University of Oxford, 2001.

[MS09] G J Mysore et P Smaragdis : Relative pitch estimation of multiple instruments. *In Proc. of ICASSP*, pages 313–316, Taipei, Taiwan, 2009.

[MS12] G J Mysore et M Sahani : Variational Inference in Non-negative Factorial Hidden Markov Models for Efficient Audio Source Separation. *In Proc. of ICML*, Édimbourg, Écosse, 2012.

[Mys10] G J Mysore : *A Non-negative Framework for Joint Modeling of Spectral Structure and Temporal Dynamics in Sound Mixtures*. Thèse de doctorat, Stanford University, 2010.

[NLK+10] M Nakano, J Le Roux, H Kameoka, Y Kitano, N Ono et S Sagayama : Nonnegative Matrix Factorization with Markov-Chained Bases for Modeling Time-Varying Patterns in Music Spectrograms. *In Proc. of LVA/ICA*, pages 149–156, St. Malo, France, 2010.

[NLK+11] M. Nakano, J. Le Roux, H. Kameoka, N. Ono et S. Sagayama : Infinite-State Spectrum Model for Music Signal Analysis. *In Proc. of ICASSP*, pages 1972–1975, Prague, République Tchèque, 2011.

[OF10] A. Ozerov et C. Févotte : Multichannel Nonnegative Matrix Factorization in Convolutive Mixtures for Audio Source Separation. *IEEE Trans. on Audio Speech and Language Processing*, 18(3):550–563, 2010.

[OFC09] A. Ozerov, C. Févotte et M. Charbit : Factorial scaled hidden Markov model for polyphonic audio representation and source separation. *In Proc. of WASPAA*, pages 121–124, New Paltz, New York, USA, 2009.

[OKS12] K. Ochiai, H. Kameoka et S. Sagayama : Explicit Beat Structure Modeling for Non-negative Matrix Factorization-Based Multipitch Analysis. *In Proc. of ICASSP*, pages 133–136, Kyoto, Japon, 2012.

[ONP12] K O'Hanlon, H Nagano et M D Plumbley : Structured sparsity for automatic music transcription. *In Proc. of ICASSP*, volume 0, pages 441–444, Kyoto, Japon, 2012.

[POFP05] C.J. Plack, A.J. Oxenham, R.R. Fay et E.N. Popper : *Pitch : Neural Coding and Perception*. 2005.

[Pra11] J. Prado : Une inversion simple de la transformée à Q constant. Rapport technique, 2011.

[ROS07b] S.A. Raczynski, N. Ono et S. Sagayama : Multipitch analysis with harmonic nonnegative matrix approximation. *In Proc. of ISMIR*, pages 381–386, Vienne, Autriche, 2007.

[RVCS10] B. Raj, T. Virtanen, S. Chaudhuri et R. Singh : Non-negative matrix factorization based compensation of music for automatic speech recognition. *In Proc. of INTERSPEECH 2010*, pages 717–720, Makuhari, Japon, 2010.

[SB03] P. Smaragdis et J.C. Brown : Non-negative Matrix Factorization for Polyphonic Music Transcription. *In Proc. of WASPAA*, pages 177–180, New Paltz, New York, USA, 2003.

[SC12] U. Simsekli et A.T. Cemgil : Score Guided Musical Source Separation Using Generalized Coupled Tensor Factorization. *In Proc. of EUSIPCO*, Bucarest, Romanie, 2012.

[Sha07] M.V. Shashanka : *Latent variable framework for modeling and separating single-channel acoustic sources*. Thèse de doctorat, Boston University, Boston, MA, USA, 2007.

[SK10] C Schörkhuber et A Klapuri : Constant-Q transform toolbox for music processing. *In Proc. of the 7th Sound and Music Computing Conference*, Barcelone, Espagne, 2010.

[SL08] M.N. Schmidt et H. Laurberg : Non-negative matrix factorization with Gaussian process priors. *Computational Intelligence and Neuroscience*, 2008:3—-10, 2008.

[SRS08a] M V Shashanka, B Raj et P Smaragdis : Probabilistic Latent Variable Models as Nonnegative Factorizations. *Computational intelligence and neuroscience*, 2008(4): 947438, 2008.

[SRS08b] P. Smaragdis, B. Raj et M.V. Shashanka : Sparse and shift-invariant feature extraction from non-negative data. *In Proc. of ICASSP*, pages 2069–2072, Las Vegas, Nevada, USA, 2008.

[VBB08] E. Vincent, N. Bertin et R. Badeau : Harmonic and inharmonic Nonnegative Matrix Factorization for Polyphonic Pitch transcription. *In Proc. of ICASSP*, pages 109–112, Las Vegas, Nevada, USA, 2008.

[VBB10] E. Vincent, N. Bertin et R. Badeau : Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio Speech and Language Processing*, 18(3):528–537, 2010.

[VCG08] T. Virtanen, A.T. T Cemgil et S. Godsill : Bayesian extensions to non-negative matrix factorisation for audio signal modelling. *In Proc. of ICASSP*, pages 1825–1828, Las Vegas, NV, USA, 2008.

[VFG06] E. Vincent, C. Févotte et R. Gribonval : Performance measurement in blind audio source separation. *Audio, Speech and Language Processing, IEEE Trans. on*, 14(4):1462–1469, 2006.

[Vir06] T. Virtanen : *Sound Source Separation in Monaural Music Signals*. Thèse de doctorat, Tampere University of Technology, 2006.

[Vir07] T. Virtanen : Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. on Audio, Speech and Language Processing*, 15(3):1066–1074, 2007.

[VMR08] T. Virtanen, A. Mesaros et M. Ryynänen : Combining Pitch-Based Inference and Non-Negative Spectrogram Factorization in Separating Vocals from Polyphonic Music. *In Proc. of the ISCA Tutorial and Research Workshop on Statistical and Perceptual Audition*, Brisbane, Australie, 2008.

[VP05] E. Vincent et M.D. Plumbley : Predominant-F0 estimation using Bayesian harmonic waveform models. *In Proc. of MIREX*, Londres, UK, 2005.

[vR79] C.J. van Rijsbergen : *Information retrieval, 2nd Edition*. Butterworths, London, UK, 1979.

[Weba] Website : Algorithme YIN : code Matlab. http ://audition.ens.fr/adc/.

[Webb] Website : CQT inversible : code Matlab. http ://www.tsi.telecom-paristech.fr/aao/en/2011/06/06/inversible-cqt/.

[Webc] Website : Music Information Retrieval Evaluation eXchange (MIREX). http ://music-ir.org/mirexwiki/.

[Webd] Website : QUASI-Separation. http ://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/.

[Webf] Website : QUASI-Transcription, set 1, v1.1. http ://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/.

[Webg] Website : University of Iowa Musical Instrument Sample Database. http ://theremin.music.uiowa.edu/index.html.

[YCS11] K.Y. Yilmaz, A.T. Cemgil et U. Simsekli : Generalized Coupled Tensor Factorization. *In NIPS*, Granada, Spain, 2011.

[ZC07] R. Zdunek et A. Cichocki : Nonnegative matrix factorization with constrained second-order optimization. *Signal Processing, IEEE Transactions on*, 87(8):1904–1916, 2007.

[ZC08] R. Zdunek et A. Cichocki : Nonnegative Matrix Factorization with Quadratic Programming. *Neurocomputing*, 71(10-12):2309–2320, 2008.

[ZF07] Y Zhang et Y Fang : A NMF algorithm for blind separation of uncorrelated signals. *In Proc. of International Conference on Wavelet Analysis and Pattern Recognition*, pages 999–1003, Pékin, Chine, 2007.

# Cinquième partie

# Annexes

# Annexe A

# La gamme tempérée et l'échelle MIDI

La gamme tempérée est la gamme où l'octave est découpée en 12 intervalles égaux, appelés demi-tons, contrairement aux gammes naturelle, pythagoricienne, ou à tempérament inégal [Dev90]. Dans le standard MIDI, à chaque note de cette gamme est associé un entier (entre 0 et 127), le numéro 69 étant associé au $la_4$, dont la fréquence fondamentale est traditionnellement fixée à 440 Hz. Les formules de conversion entre une note MIDI $n$ et la fréquence fondamentale $f_0$ correspondante sont alors les suivantes :

$$n = \left[12 \log_2 \frac{f_0}{440}\right] + 69,$$
$$f_0 = 2^{(n-69)/12} \times 440.$$

La note la plus grave du piano correspond à la note MIDI 21 et la plus aigüe à la note 108, soit un total de 88 notes.

# Annexe B

# Mises à jour avec les aprioris de parcimonie

Dans cette annexe nous détaillons les calculs permettant de résoudre l'étape MAP avec l'utilisation des aprioris de parcimonie (section 3.2 page 42).

**Parcimonie et norme $l_{1/2}$.** Nous souhaitons trouver sur $\Omega$ l'argument $\hat{\boldsymbol{\theta}}$ vérifiant la contrainte $\varphi(\hat{\boldsymbol{\theta}}) = 1 - \sum_d \hat{\theta}_d = 0$ qui maximise la fonction $\mathcal{M}$ (équation (3.4) page 43), qui pour rappel est définie comme :

$$\begin{aligned} \mathcal{M} \ : \ \Omega = ]0,1[^D &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_d w_d \ln(\theta_d) - 2\beta_{\text{parci}} \sum_d \sqrt{\theta_d}. \end{aligned}$$

On sait que cet argument existe puisque $\mathcal{M}$ est continue et majorée par 0 sur $\Omega$. De plus, comme $\mathcal{M}$ et $\varphi$ sont deux fois différentiables, les conditions nécessaires du premier et du second ordre, propres aux maxima locaux sont vérifiées : il existe un unique réel $\rho$ tel que (l'opérateur $\langle .,. \rangle$ correspond au produit scalaire)

$$\nabla L_\rho(\hat{\boldsymbol{\theta}}) = 0 \quad \text{et} \tag{B.1}$$

$$\left\langle H_{L_\rho}\left(\hat{\boldsymbol{\theta}}\right) x, x \right\rangle \leq 0, \ \forall x \in \left\{ x \in \mathbb{R}^D / \left\langle \nabla\varphi(\hat{\boldsymbol{\theta}}), x \right\rangle = 0 \right\}, \tag{B.2}$$

où $L_\rho$ est la fonction de Lagrange, définie comme :

$$\begin{aligned} L_\rho \ : \ \Omega = ]0,1[^D &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \mathcal{M}(\boldsymbol{\theta}) + \rho\varphi(\boldsymbol{\theta}) \end{aligned} \tag{B.3}$$

et où $H_{L_\rho}\left(\hat{\boldsymbol{\theta}}\right)$ est la matrice Hessienne de $L_\rho$ au point $\hat{\boldsymbol{\theta}}$. On peut ajouter aussi des conditions suffisantes de maximalité : si $\boldsymbol{\theta}$ vérifie $\varphi(\boldsymbol{\theta}) = 0$ et s'il existe $\rho \in \mathbb{R}$ tel que

$$\begin{aligned} &\nabla L_\rho(\boldsymbol{\theta}) = 0 \quad \text{et} \\ &\langle H\left(L_\rho(\boldsymbol{\theta})\right) x, x\rangle < 0,\ \forall x \in \left\{x \in \mathbb{R}^D / \langle \nabla\varphi(\boldsymbol{\theta}), x\rangle = 0\right\}, \end{aligned} \tag{B.4}$$

alors $\boldsymbol{\theta}$ est un maximum local. Mais concentrons nous tout d'abord sur les conditions nécessaires. En notant $\hat{\boldsymbol{X}}$ le vecteur de coefficients $\hat{X}_d = \sqrt{\hat{\theta}_d}$, l'équation (B.1) revient à :

$$\forall d,\ \mathrm{g}_d(\hat{X}_d) = 0 \tag{B.5}$$

où $\mathrm{g}_d$ est la fonction suivante, définie sur $\mathbb{R}^+$ :

$$\mathrm{g}_d(X_d) = \rho X_d^2 + \beta_{\text{parci}} X_d - w_d. \tag{B.6}$$

En étudiant, sur $\mathbb{R}^+$, le tableau de variations ainsi que la dérivée seconde de $\mathrm{g}_d$ dans les deux cas $\rho \geq 0$ et $\rho < 0$, et en prenant en compte le fait que $\hat{X}_d \in ]0, 1[$, on peut distinguer deux cas de figure possibles.

*Le premier cas* a lieu si $\rho > \rho_0 = \max_d(w_d) - \beta$ ou si $\rho = \rho_0$ et $\rho < -\beta/2$, c'est-à-dire quand $\forall d$, on est sûr que la courbe de $\mathrm{g}_d$ ne rencontre qu'une unique fois l'axe des abscisses sur $]0, 1[$. Dans ce cas, $\forall d$, $\hat{X}_d$ vérifie nécessairement l'équation suivante :

$$\forall d,\ \hat{X}_d = \frac{-\beta_{\text{parci}} + \sqrt{\beta_{\text{parci}}^2 + 4\rho w_d}}{2\rho}, \tag{B.7}$$

et alors

$$\forall d,\ \hat{\theta}_d = \mathrm{h}_d^+(\rho) = \frac{2w_d^2}{\beta_{\text{parci}}^2 + 2\rho w_d + \beta_{\text{parci}}\sqrt{\beta_{\text{parci}}^2 + 4\rho w_d}}. \tag{B.8}$$

Pour finir avec ce cas, en étudiant la monotonie et les limites de $\mathrm{h}_d^+$ (strictement décroissante vers 0) sur son domaine de définition, on peut déduire que nécessairement $\sum_d \mathrm{h}_d^+(\rho_0) \geq 1$ et qu'il existe un unique $\rho \geq \rho_0$ tel que $\sum_d \mathrm{h}_d^+(\rho) = \sum_d \hat{\theta}_d = 1$.

*Le second cas de figure* possible a lieu si $\rho < -\beta/2$ , $\rho < \rho_0$ et $\rho \geq \rho_{min} = -\beta^2/4\max_d(w_d)$, c'est-à-dire quand on sait que $\forall d$, la courbe de $\mathrm{g}_d$ croise au moins une fois l'axe des abscisses entre 0 et 1. Dans ce cas, $\hat{\boldsymbol{X}}$ vérifie nécessairement l'équation suivante :

$$\forall d,\ \hat{X}_d = \frac{-\beta_{\text{parci}} \pm \sqrt{\beta_{\text{parci}}^2 + 4\rho w_d}}{2\rho}, \tag{B.9}$$

soit

$$\forall d,\ \hat{\theta}_d = \mathrm{h}_d^{+/-}(\rho) = \frac{2w_d^2}{\beta_{\text{parci}}^2 + 2\rho w_d \pm \beta_{\text{parci}}\sqrt{\beta_{\text{parci}}^2 + 4\rho w_d}}, \tag{B.10}$$

ce qui mène à $2^D$ solutions possibles ! Cependant, si l'on prend un vecteur $x$ tel que

$$\begin{cases} x_{d_1} = 1,\ d_1 \in [\![1, D]\!] \\ x_{d_2} = -1\ d_2 \in [\![1, D]\!] \backslash \{d_1\} \\ x_d = 0,\ \forall d \in [\![1, D]\!] \backslash \{d_1, d_2\} \end{cases}, \tag{B.11}$$

alors la condition (B.2) équivaut à :

$$-\frac{w_{d_1}}{\hat{\theta}_{d_1}^2} + \frac{\beta_{\text{parci}}}{2\hat{\theta}_{d_1}^{3/2}} - \frac{w_{d_2}}{\hat{\theta}_{d_2}^2} + \frac{\beta_{\text{parci}}}{2\hat{\theta}_{d_2}^{3/2}} \leq 0. \tag{B.12}$$

On peut alors déduire qu'il existe $d_0 \in [\![1, D]\!]$ tel que $\forall d \neq d_0$, $-\frac{w_d}{\hat{\theta}_d^2} + \frac{\beta_{\text{parci}}}{2\hat{\theta}_d^{3/2}} \leq 0$, soit $\forall d \neq d_0$, $\hat{\theta}_d \leq \frac{4w_d^2}{\beta_{\text{parci}}^2}$ et que donc :

$$\forall d \neq d_0, \hat{\theta}_d = \mathrm{h}_d^+(\rho). \tag{B.13}$$

$\hat{\theta}_{d_0}$ peut alors être égal à $\mathrm{h}_{d_0}^+(\rho)$ ou a $\mathrm{h}_{d_0}^-(\rho)$. Ainsi, on a réduit le nombre de possibilités à $D+1$. Avant de les étudier, on peut remarquer, en étudiant les fonctions $\mathrm{h}_d^+$ et $\mathrm{h}_d^-$, que l'on a nécessairement $\sum_d \mathrm{h}_d^+(\rho_0) < 1$ (il suffit de constater que pour $\rho$ vérifiant les hypothèses du cas de figure courant, on a $\forall d$, $\mathrm{h}_d^-(\rho) \geq \mathrm{h}_d^+(\rho) > \mathrm{h}_d^+(\rho_0)$). Dans le cas où $\hat{\theta}_{d_0} = \mathrm{h}_{d_0}^+(\rho)$, alors on sait que si $\sum_d h_d^+(\rho_{min}) \geq 1$, il existe un unique $\rho \in [\rho_{min}, \min(\rho_0, -\beta/2)[$ tel que $\sum_d \mathrm{h}_d^+(\rho) = 1$. Hélas, pour les autres solutions possibles, on ne peut pas s'assurer qu'il existe au maximum un unique $\rho$ tel que $\sum_d \hat{\theta}_d - 1 = 0$. Au mieux, en étudiant la régularité des fonctions $h_d^+(\rho)$ et $h_d^-(\rho)$ (dérivées première, seconde et même troisième) on peut trouver un nombre maximum de racines possibles.

Quoi qu'il en soit, dans la pratique, $\beta_{\text{parci}}$ est suffisamment faible pour que l'on ait toujours $\sum_d h_d^+(\rho_{min}) > 1$. Dans ce cas :

- si $\sum_d h_d^+(\rho_0) \geq 1$, on sait que nécessairement $\rho \geq \rho_0$, qu'il est unique, et que $\hat{\boldsymbol{\theta}}$ est donné par l'équation (B.8). $\hat{\boldsymbol{\theta}}$ correspond alors au maximum global puisque c'est l'unique argument qui vérifie la condition nécessaire (B.1),
- sinon, si $\hat{\boldsymbol{\theta}}$ vérifie l'équation (B.8), où $\rho$ est l'unique réel dans $]\rho_{min}, \min(\rho_0, -\beta/2)[$ tel que $\sum_d h_d^+(\rho) = 1$, alors il est facile de constater qu'il vérifie aussi les conditions (B.4) et qu'il correspond donc à un maximum local. On ne sait hélas pas s'il s'agit ou non du maximum global, mais l'on peut se contenter d'une telle solution : il suffit de vérifier qu'elle fait bien augmenter la valeur de $\mathcal{M}$. Si tel est le cas (ce que l'on observe systématiquement dans la pratique), en vertu de l'algorithme EM généralisé (GEM) [Dem77], cela suffit pour faire converger l'algorithme vers un maximum local de vraisemblance.

Quel que soit le cas, la valeur de $\rho$ peut être estimée par n'importe quel algorithme numérique de recherche de racine. On peut ajouter que pour toutes les expériences que nous avons faites avec l'utilisation de cet apriori, la condition la plus forte $\sum_d h_d^+(\rho_0) \geq 1$ a systématiquement été

vérifiée.

**Parcimonie et entropie.** Avec l'apriori entropique, la fonction $\mathcal{M}$ (fonction (3.7) page 44) à maximiser, sous la contrainte $\varphi(\boldsymbol{\theta}) = 1 - \sum_d \theta_d = 0$, est pour rappel définie par :

$$\begin{aligned} \mathcal{M} \ : \ \Omega = ]0,1[^D &\longrightarrow \mathbb{R} \\ \boldsymbol{\theta} &\longmapsto \sum_d w_d \ln(\theta_d) + \beta_{\text{parci}} \sum_d \theta_d \ln(\theta_d) \,. \end{aligned} \tag{B.14}$$

Pour résoudre ce problème, on propose la même démarche que pour l'apriori fondé sur la norme $l_{1/2}$. Si $\hat{\boldsymbol{\theta}}$ est l'argument du maximum global (on sait qu'il existe puisque $\mathcal{M}$ est majorée sur $\Omega$), alors il vérifie les conditions nécessaires du premier et du second ordre propres aux maxima locaux, formulées par les équations (B.1) à (B.3) ($\mathcal{M}$ et $\varphi$ sont ici aussi deux fois différentiables). La condition du premier ordre (B.1) revient à :

$$\forall d,\ \mathrm{g}_d(\hat{\theta}_d) = 0 \tag{B.15}$$

où $\mathrm{g}_d$ est la fonction définie sur $\mathbb{R}^+$ par :

$$\mathrm{g}_d(\theta_d) = (\rho - \beta_{\text{parci}})\theta_d - \beta_{\text{parci}}\theta_d \ln(\theta_d) - w_d. \tag{B.16}$$

On peut prouver (*cf.* [Bra99]) que si $\hat{\theta}_d$ vérifie $\mathrm{g}_d(\hat{\theta}_d) = 0$, alors forcément, il vérifie également

$$\hat{\theta}_d = \mathrm{h}_d^b(\rho) = \frac{w_d/\beta_{\text{parci}}}{-\mathcal{W}_b\left(-\frac{w_d}{\beta_{\text{parci}}} \exp\left(1 - \rho/\beta_{\text{parci}}\right)\right)} \tag{B.17}$$

où $\mathcal{W}$ est la fonction multivaluée de Lambert [CGH$^+$96], et $b = -1,\ 0$ le numéro de la branche de $\mathcal{W}$. Afin de déterminer, pour chaque valeur de $d$, quelle branche choisir, nous pouvons étudier la fonction $\mathrm{g}_d$. Là encore, on tombe sur deux cas de figure possibles.

*D'abord*, si $\rho > \rho_0 = \max_d(w_d) + \beta$ ou si $\rho = \rho_0$ et $\rho < 2\beta$, alors, sur $]0,1[$, la courbe de $\mathrm{g}_d$ ne rencontre qu'une unique fois l'axe des abscisses, et le tableau de variation de $\mathrm{g}_d$ nous informe que $\hat{\theta}_d$ correspond forcément à la racine la plus petite, et que donc :

$$\forall d,\ \hat{\theta}_d = \mathrm{h}_d^{-1}(\rho) = \frac{w_d/\beta_{\text{parci}}}{-\mathcal{W}_{-1}\left(-\frac{w_d}{\beta_{\text{parci}}} \exp\left(1 - \frac{\rho}{\beta_{\text{parci}}}\right)\right)}. \tag{B.18}$$

En remarquant la nature strictement décroissante vers 0 de $\mathrm{h}_d^{-1}$ sur $[\rho_0, +\infty[$, on arrive à la conclusion que l'on a nécessairement $\sum_d \mathrm{h}_d^{-1}(\rho_0) \geq 1$, et qu'il existe un unique $\rho$ tel que $\sum_d \mathrm{h}_d^{-1}(\rho) = 1$.

*Le deuxième cas de figure* a lieu quand $\rho < \rho_0$, $\rho < 2\beta$ et $\rho \geq \rho_{min} = \beta\left(\ln\left(\max_d(w_d)/\beta\right) + 2\right)$. Alors on est assuré que la courbe de $\mathrm{g}_d$ croise au moins une fois l'axe des abscisses sur $]0,1[$ mais on ne peut *a priori* pas savoir quelle branche choisir pour $\hat{\theta}_d$. Il est possible alors d'invoquer la

condition du second ordre B.2, en utilisant le même vecteur $x$ que celui défini équation B.11. Cela mène à l'assertion

$$\forall (d_1, d_2)/d_1 \neq d_2,\ \frac{\beta_{\text{parci}}}{\hat{\theta}_d} - \frac{w_d}{\hat{\theta}_d^2} + \frac{\beta_{\text{parci}}}{\hat{\theta}_d} - \frac{w_d}{\hat{\theta}_d^2} \leq 0, \tag{B.19}$$

et l'on peut en déduire qu'il existe $d_0$ tel que $\forall d \neq d_0$, $\hat{\theta}_d \leq \frac{w_d}{\beta_{\text{parci}}}$, et donc que

$$\forall d \neq d_0, \hat{\theta}_d = \mathrm{h}_d^{-1}(\rho). \tag{B.20}$$

D'abord, on peut remarquer en étudiant les fonctions $\mathrm{h}_\mathrm{d}^0$ et $\mathrm{h}_\mathrm{d}^{-1}$ pour $\rho < \rho_0$, que l'on a nécessairement $\sum_d \mathrm{h}_d^{-1}(\rho_0) < 1$. De plus, si $\sum_d \mathrm{h}_d^{-1}(\rho_{min}) \geq 1$ et si $\hat{\theta}_{d_\grave{a}} = \mathrm{h}_{d_0}^{-1}(\rho)$, alors il existe un unique $\rho \in [\rho_{min}, \min(\rho_0, 2\beta)[$ tel que $\sum_d \mathrm{h}_d^{-1}(\rho) = 1$. Pour les autres solutions possibles, on ne peut hélas pas savoir s'il existe au maximum une seule valeur de $\rho$ telle que $\sum_d \mathrm{h}_d(\rho) = 1$ ($\mathrm{h_d}$ faisant référence à $\mathrm{h}_\mathrm{d}^{-1}$ ou $\mathrm{h}_\mathrm{d}^0$ suivant la possibilité traitée).

Dans la pratique, on a toujours $\sum_d \mathrm{h}_d^+(\rho_{min}) > 1$. Alors dans ce cas :

- si $\sum_d \mathrm{h}_d^+(\rho_0) \geq 1$, alors il n'existe qu'un unique point stationnaire, correspondant nécessairement au maximum global. Ce point $\hat{\boldsymbol{\theta}}$ vérifie l'équation (B.18) où $\rho$ est l'unique réel supérieur ou égal à $\rho_0$ tel que $\varphi(\hat{\boldsymbol{\theta}}) = 0$,
- sinon, le point $\hat{\boldsymbol{\theta}}$ vérifiant l'équation (B.18), où $\rho$ est l'unique réel de $]\rho_{min}, \min(\rho_0, 2\beta)[$ tel que $\varphi(\hat{\boldsymbol{\theta}}) = 0$, vérifie également les conditions suffisantes (B.4). Il s'agit donc d'un maximum local : s'il fait augmenter la valeur de $\mathcal{M}$ (toujours le cas en pratique), il peut être utilisé comme mise à jour des paramètres, en vertu de l'algorithme GEM.

La valeur de $\rho$ peut être estimée avec un algorithme de recherche de racine.

**Parcimonie et norme $l_2$.** Pour résoudre l'étape MAP avec cet apriori, on suit exactement le même cheminement qu'avec les deux précédents aprioris. Ainsi, on peut prouver que si $\sum_d \mathrm{h}_d^-(\rho_{min}) > 1$ (toujours le cas en pratique), avec $\rho_{min} = 2\sqrt{\beta \max_d(w_d)}$ et

$$\mathrm{h}_d^-(\rho) = \frac{\rho - \sqrt{\rho^2 - 4\beta w_d}}{2\beta} = \frac{2w_d}{\rho + \sqrt{\rho^2 - 4\beta w_d}}, \tag{B.21}$$

alors :

- si $\sum_d \mathrm{h}_d^-(\rho_0) \geq 1$ avec $\rho_0 = \max_d(w_d) + \beta$, il existe un unique point stationnaire (le maximum global) $\hat{\boldsymbol{\theta}}$, vérifiant $\forall d$, $\hat{\theta}_d = \mathrm{h}_d^-(\rho)$, où $\rho$ est l'unique réel supérieur à $\rho_0$ tel que $\varphi(\hat{\boldsymbol{\theta}}) = 0$,
- sinon, le point $\hat{\boldsymbol{\theta}}$ vérifiant $\forall d$, $\hat{\theta}_d = \mathrm{h}_d^-(\rho)$ avec $\rho$ l'unique réel dans $]\rho_{min}, \min(\rho_0, \beta/2)[$ tel que $\varphi(\hat{\boldsymbol{\theta}}) = 0$, correspond à un maximum local. S'il fait augmenter la valeur de $\mathcal{M}$ (ce que l'on observe systématiquement en pratique), l'algorithme GEM permet d'affirmer qu'il peut être utilisé comme nouvelle valeur des paramètres correspondants.

Enfin, $\rho$ peut être estimé avec un algorithme numérique de recherche de racine.

# Annexe C

# Les bases de données

Dans cette annexe, nous présentons en détails les trois bases de données qui ont été utilisées pour les tâches d'estimations de hauteurs multiples des chapitres 6 page 87 et 7 page 111, ainsi que de transcription (chapitre 8 page 125).

## $BD_{\text{app}}$

La base $BD_{\text{app}}$, que nous avons constituée pour entraîner nos algorithmes, est constituée de cinq fichiers audio monophoniques, échantillonnés à 44.1 kHz, d'une durée de 30 sec., et extraits de la base de musique classique de RWC [GN03]. La liste des fichiers de $BD_{\text{app}}$ est consultable dans la Table C.1. Les vérités terrains des transcriptions sont celles révisées par Meinart Müller et disponibles en ligne [1].

| Titre (Compositeur) | Numéro | Extrait (s) | # instr. / notes | Niveau de Pol. (Moy./Max.) |
|---|---|---|---|---|
| The Musical Offering (Bach) | 12 | $[0, 30]$ | 2 / 69 | 1.3 / 4 |
| String Quartet No. 19 (Mozart) | 13 | $[0, 30]$ | 4 / 93 | 2.7 / 6 |
| Clarinet Quintet Op.115. (Brahms) | 17 | $[0, 30]$ | 5 / 254 | 3.8 / 10 |
| Horn Trio Op.8 (Brahms) | 18 | $[6.4, 36.4]$ | 3 / 553 | 4.2 / 10 |
| The Anna Magdalena Bach Notebook (Bach) | 24a | $[0, 30]$ | 1 / 155 | 1.9 / 5 |

**Table C.1** – $BD_{\text{app}}$ : liste des extraits audio de la base de musique classique de RWC (catalogue RWC-MDB-C-2001), ainsi que leur nombre correspondant d'instruments et de notes (# instr. / notes) et leur niveau de polyphonie (Pol.).

1. `http://www.mpi-inf.mpg.de/resources/MIR/SyncRWC60/`

## $BD_{\mathbf{maps}}$

Au cours de son doctorat, Valentin Emiya a constitué une base de données de piano appelée MAPS [Emi08, EBD10]. Elle est constituée de plusieurs sous-bases, suivant que les fichiers audio ont été synthétisés par un logiciel ou enregistrés sur un piano acoustique Yamaha DisKlavier. La base $BD_{\text{maps}}$ que nous avons créée pour l'évaluation de nos algorithmes de transcription est constituée de cinq morceaux de piano virtuel et cinq morceaux de piano acoustique, où seulement les 30 premières secondes ont été retenues. On a reporté l'ensemble des morceaux sélectionnées dans la Table C.2.

| Nom du fichier | # notes | Niveau de Pol. (Moy./Max.) |
|---|---|---|
| MAPS_MUS-chpn_op25_e2_AkPnBcht | 436 | 2.4 / 10 |
| MAPS_MUS-chpn_op66_AkPnBcht | 458 | 6.4 / 18 |
| MAPS_MUS-chpn-p1_AkPnBcht | 327 | 5.5 / 11 |
| MAPS_MUS-chpn-p3_AkPnBcht | 384 | 5.2 / 13 |
| MAPS_MUS-chpn-p4_AkPnBcht | 195 | 3.8 / 6 |
| MAPS_MUS-chpn_op35_1_ENSTDkAm | 288 | 4.8 / 13 |
| MAPS_MUS-chpn_op66_ENSTDkAm | 457 | 7.0 / 18 |
| MAPS_MUS-chpn-p4_ENSTDkAm | 195 | 3.8 / 6 |
| MAPS_MUS-chpn-p14_ENSTDkAm | 434 | 5.6 / 12 |
| MAPS_MUS-chpn-p15_ENSTDkAm | 142 | 3.8 / 8 |

**Table C.2** – $BD_{\text{maps}}$ : liste des extraits audio de la base de piano MAPS, ainsi que leur nombre correspondant de notes (# notes) et leur niveau de polyphonie (Pol.). Les fichiers ont été coupés de sorte qu'ils ne durent pas plus de 30 s.

## $BD_{\mathbf{mirex}}$

La base $BD_{\text{mirex}}$ est constituée d'un unique morceau monophonique de 54 secondes, extrait de la base de développement de MIREX [Webc]. C'est une transcription pour quintette à vent (basson, cor, hautbois, clarinette et flute) d'un extrait du quatuor à cordes No.4 Op.18 de Beethoven. Les nombres d'instruments et de notes, ainsi que niveau de polyphonie sont reportés dans la Table C.3.

| Nom du fichier | Durée | # instr. / notes | Niveau de Pol. (Moy./Max.) |
|---|---|---|---|
| MIREX | 54 s. | 5 / 1011 | 3.1 / 6 |

**Table C.3** – $BD_{\text{mirex}}$ : durée, nombre de notes et d'instruments (# instr. / notes) ainsi que niveau de polyphonie (Pol.) moyen et maximum.

## $BD_{\text{quasi}}$

Dans le cadre du projet QUAERO [2] et de cette thèse, une nouvelle base de données pour la transcription automatique de musique, appelée QUASI-Transcription (QUASI pour *QUaero Audio SIgnals*), a été élaborée. Elle contient essentiellement des morceaux de musiques actuelles : rock, reggae, chanson, pop, etc. Pour le moment cette base contient deux sous-bases : *s1* et *s2*. L'ensemble *s1* est constitué de morceaux monophoniques sous licence Creative Commons, qui ont été transcrits à la main par un expert. Les morceaux (monophoniques) de l'ensemble *s2* ont été eux enregistrés pour l'occasion. Les instruments qui les composent sont un mélange d'instruments de synthèse logicielle (par exemple : piano, orgue, Rhodes, ou encore basse) et d'instruments acoustiques (par exemple : voix chantée, violon, guitare, guitare électrique). Pour les instruments virtuels, les transcriptions ont été obtenues grâce au fichier MIDI correspondant, alors que pour les instruments acoustiques, elles ont été élaborées à la main. Dans ce mémoire, nous avons retenu uniquement ce dernier sous-ensemble. Les caractéristiques de chaque morceau sont consultables dans la Table C.4.

| Nom du fichier | Durée | # instr. / notes | Niveau de Pol. (Moy./Max.) |
|---|---|---|---|
| RockSong | $01'14''$ | 9 / 1039 | 3.9 / 10 |
| Choir | $01'11''$ | 4 / 224 | 3.4 / 4 |
| Filter | $01'19''$ | 19 / 2418 | 5.9 / 11 |
| Unison | $01'17''$ | 6 / 561 | 5.6 / 9 |
| Accelerando | $01'49''$ | 3 / 1046 | 2.3 / 6 |

**Table C.4** – $BD_{\text{quasi}}$ : description des morceaux de la base QUASI-Transcription *s2* qui constitue notre base d'évaluation $BD_{\text{quasi}}$.

2. `http://www.quaero.org`

# L'analyse probabiliste en composantes latentes et ses adaptations aux signaux musicaux. Application à la transcription automatique de musique et à la séparation de sources.

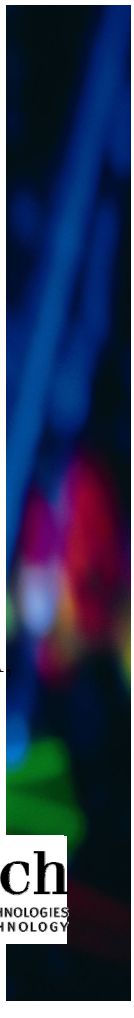
**Benoit FUENTES**

**RESUMÉ :** La transcription automatique de musique polyphonique consiste à estimer automatiquement les notes présentes dans un enregistrement, via trois de leurs attributs : temps d'attaque, durée et hauteur. Pour traiter ce problème, il existe une classe de méthodes dont le principe est de modéliser un signal comme une somme d'éléments de base, porteurs d'informations symboliques. Parmi ces techniques d'analyse, on trouve l'analyse probabiliste en composantes latentes (PLCA). L'objet de cette thèse est de proposer des variantes et des améliorations de la PLCA afin qu'elle puisse mieux s'adapter aux signaux musicaux et ainsi mieux traiter le problème de la transcription. Pour cela, un premier angle d'approche est de proposer de nouveaux modèles de signaux, en lieu et place du modèle inhérent à la PLCA, suffisamment expressifs pour pouvoir s'adapter aux notes de musique possédant simultanément des variations temporelles de fréquence fondamentale et d'enveloppe spectrale. Un deuxième aspect du travail effectué est de proposer des outils permettant d'aider l'algorithme d'estimation des paramètres à converger vers des solutions significatives via l'incorporation de connaissances *a priori* sur les signaux à analyser, ainsi que d'un nouveau modèle dynamique. Tous les algorithmes ainsi imaginés sont appliqués à la tâche de transcription automatique. Nous voyons également qu'ils peuvent être directement utilisés pour la séparation de sources, qui consiste à séparer plusieurs sources d'un mélange, et nous proposons deux applications dans ce sens.

**MOTS-CLEFS :** Transcription automatique de musique, estimation de hauteurs multiples, séparation de sources, PLCA, NMF

**ABSTRACT:** Automatic music transcription consists in automatically estimating the notes in a recording, through three attributes: onset time, duration and pitch. To address this problem, there is a class of methods which is based on the modeling of a signal as a sum of basic elements, carrying symbolic information. Among these analysis techniques, one can find the probabilistic latent component analysis (PLCA). The purpose of this thesis is to propose variants and improvements of the PLCA, so that it can better adapt to musical signals and thus better address the problem of transcription. To this aim, a first approach is to put forward new models of signals, instead of the inherent model of PLCA, expressive enough so they can adapt to musical notes having variations of both pitch and spectral envelope over time. A second aspect of this work is to provide tools to help the parameters estimation algorithm to converge towards meaningful solutions through the incorporation of prior knowledge about the signals to be analyzed, as well as a new dynamic model. All the devised algorithms are applied to the task of automatic transcription. They can also be directly used for source separation, which consists in separating several sources from a mixture, and two applications are put forward in this direction.

**KEY-WORDS:** Automatic music transcription, multipitch estimation, source separation, PLCA NMF.



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY