Alignement temporel musique-sur-partition par modèles graphiques discriminatifs

Cyril Joder

▶ **To cite this version:**

Cyril Joder. Alignement temporel musique-sur-partition par modèles graphiques discriminatifs. Traitement du signal et de l'image [eess.SP]. Télécom ParisTech, 2011. Français. NNT : . pastel-00664260

HAL Id: pastel-00664260
https://pastel.hal.science/pastel-00664260

Submitted on 30 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École Doctorale
d'Informatique,
Télécommunications
et Électronique de Paris

Thèse

présentée pour obtenir le grade de docteur

de TELECOM ParisTech

Spécialité : Signal et Images

Cyril JODER

Alignement temporel musique-sur-partition par modèles graphiques discriminatifs

Soutenu le 29 Septembre 2011 devant le jury composé de

Régine ANDRÉ-OBRECHT	Présidente
Laurent DAUDET	Rapporteurs
François YVON	
Roger DANNENBERG	Examinateurs
Meinard MÜLLER	
Arshia CONT	Invité
Slim ESSID	Directeurs de thèse
Gaël RICHARD	

Remerciements

Je souhaite ici, sans emphase mais néanmoins sincèrement, remercier tous ceux qui, par leur aide, leurs conseils, leur collaboration, leur compréhension, ou leur humour, m'ont permis de mener à bien cette thèse.

Les premiers de ces remerciements vont bien sûr à mes directeurs de thèse, Gaël Richard et Slim Essid. Ils m'ont accordé leur confiance dès le stage de master que j'ai effectué à TELECOM ParisTech, déjà sous leur direction. Leur expérience et leurs conseils m'ont été précieux tout au long de ces quatre années et demi de collaboration. J'ai ainsi pu profiter, entre bien d'autres choses, des nombreuses idées de Slim, de son talent de pédagogue et de son indéfectible optimisme. De Gaël, je soulignerai en particulier la vision globale de son domaine de recherche, la pertinence de ses remarques, ainsi que le soutien qu'il offre à ses doctorants, notamment grâce à son pléthorique carnet d'adresses. Mais au delà de l'aide qu'ils m'ont accordée pour la réalisation de mes travaux scientifiques, je tiens aussi à saluer leurs qualités humaines, leur disponibilité, leur ouverture d'esprit, leur simplicité et leur passion pour la musique.

Je remercie également l'ensemble des membres du jury pour l'attention porté à ces travaux et pour l'honneur qu'ils m'ont fait de leur présence à la soutenance. Merci tout particulièrement à Laurent Daudet et François Yvon, rapporteurs de cette thèse, qui ont accepté d'étudier le manuscrit à une période où ils auraient probablement pu occuper leur temps de manière encore plus divertissante. Je remercie aussi spécialement à Arshia Cont pour m'avoir communiqué la base de données de MIREX 2006, ainsi qu'à Meinard Müller pour m'avoir accueilli pendant à Sarrebruck.

Merci encore à toute l'équipe Audiosig du département TSI (Traitement du Signal et des Images) de TELECOM ParisTech où j'ai effectué cette thèse. La qualité de la recherche produite dans cette équipe, mais aussi de l'ambience qui y règne, doivent beaucoup aux cadres que sont les enseignants-chercheurs Yves Grenier, Nicolas Moreau, Jacques Prado, Bertrand David et Roland Badeau. Les doctorants, post-docts et stagiaires ne sont pas en reste, ce qui contribue à faire de TSI un cadre exceptionnel, que je quitte à regret. Merci donc à tous ceux que j'y ai cotoyé et qui m'ont fait la grâce de leur amitié, sans ordre particulier Jean-Louis, Nancy, Romain, Mounira, Valentin, Rémi, Alexey, Laurent, Félicien, Benoît F, Manuel, Adrien, Antoine, Laurence, Pierre, François, Benoît M, Sébastien G., Fabrice, Sylvain, Thomas, Cédric, Sébastien F., Nicolas L, Gaël L., Kristoffer et bien d'autres qui, je l'espère, se reconnaitront.

D'autres remerciements sont adressés à ceux qui, au dehors du laboratoire, m'ont permis de garder ma raison pendant ces quatres annés, et même d'en tirer des moments que je n'oublierai pas. Même s'il m'est impossible de les nommer tous, je tiens à citer pêle-mêle Pierre-Yves, Antoine, Simon, Fred, Jérémy, Hadrien P. et Rémi D. du Cosmic Red Ape Orchestra, Dante des Dililies, Hadrien H., Jean, Benjamin, Bérenger, Rosa, Minh-Tâm et Olivier, cornistes ou dignes de l'être, les musiciens du COGE et des autres orchestres avec qui j'ai eu le bonheur de jouer, Manu, Axel, Eddy, Perrine, Bordo, Donatien, Stéphane Guillaume. Et pardon à ceux que j'ai oubliés.

Je ne peux terminer cette liste de remerciements sans exprimer ma gratitude à mes parents et à ma famille, nombreuse mais unie. La certitude de leur soutien et de leur

bienveillance m'est précieuse. Et une pensée toute particulière pour Divine, sans qui mon titre de docteur n'aurait pas la même saveur.

Enfin merci à toi, lecteur. En espérant que cette lecture te soit profitable...

Avertissement

Pour la rédaction de cette thèse, j'ai choisi d'appliquer les *rectifications orthographiques du français de 1990*[1]. Cependant, conscient du caractère encore polémique de ces recommandations, j'ai cru bon d'avertir le lecteur de ce choix, afin qu'il ne s'émeuve pas de l'accent grave du mot « évènement », ni de l'absence de l'accent circonflexe sur le mot « cout », pour ne citer que deux exemples fréquents dans ce documents. J'espère que les différentes relecturesont été suffisantes pour assurer la bonne tenue et la cohérence de cette thèse, du point de vue linguistique comme scientifique.

1. J.O. du 6 décembre 1990.

Résumé

Cette thèse étudie le problème de l'alignement temporel d'un enregistrement musical et de la partition correspondante. Cette tâche peut trouver de nombreuses applications dans le domaine de l'indexation automatique de documents musicaux. Nous adoptons une approche probabiliste et nous proposons l'utilisation de modèles graphiques discriminatifs de type *champs aléatoires conditionnels* pour l'alignement, en l'exprimant comme un problème d'étiquetage de séquence. Cette classe de modèles permet d'exprimer des modèles plus flexibles que les modèles de Markov cachés ou les modèles semi-markoviens cachés, couramment utilisés dans ce domaine. En particulier, elle rend possible l'utilisation d'*attributs* (ou descripteurs acoustiques) extraits de séquences de trames audio qui se recouvrent, au lieu d'observations disjointes. Nous tirons parti de cette propriété pour introduire des attributs qui réalisent une modélisation implicite du tempo au plus bas niveau du modèle.

Nous proposons trois structures de modèles différentes de complexité croissant, correspondant à différents niveaux de précision dans la modélisation de la durées des évènements musicaux. Trois types de descripteurs acoustiques sont utilisés, pour caractériser localement l'harmonie, les attaques de notes et le tempo de l'enregistrement.

Une série d'expériences réalisées sur une base de données de piano classique et de musique pop permet de valider la grande précision de nos modèles. En effet, avec le meilleur des systèmes proposés, plus de 95 % des attaques de notes sont détectées à moins de 100 ms de leur position réelle.

Plusieurs attributs acoustiques classiques, calculés à partir de différentes représentation de l'audio, sont utiliser pour mesurer la correspondance instantanée entre un point de la partition et une trame de l'enregistrement. Une comparaison de ces descripteurs est alors menée sur la base de leurs performances d'alignement. Nous abordons ensuite la conception de nouveaux attributs, grâce à l'apprentissage d'une transformation linéaire de la représentation symbolique vers une représentation temps-fréquence quelconque de l'audio. Nous explorons deux stratégies différentes, par *minimum de divergence* et *maximum de vraisemblance*, pour l'apprentissage de la transformation optimale. Les expériences effectuées montrent qu'une telle approche peut améliorer la précision des alignements, quelle que soit la représentation de l'audio utilisée.

Puis, nous étudions différents ajustements à effectuer afin de confronter les systèmes à des cas d'utilisation réalistes. En particulier, une réduction de la complexité est obtenue grâce à une stratégie originale d'élagage hiérarchique. Cette méthode tire parti de la structure hiérarchique de la musique en vue d'un décodage approché en plusieurs passes. Une diminution de complexité plus importante que celle de la méthode classique de recherche par faisceaux est observée dans nos expériences. Nous examinons en outre une modification des modèles proposés afin de les rendre robustes à d'éventuelles différences structurelles entre la partition et l'enregistrement. Enfin, les propriétés de scalabilité des modèles utilisés sont étudiées.

Abstract

This thesis focuses on the problem of aligning a musical recording to the corresponding score, which can find numerous applications in the field of music information retrieval. We choose a probabilistic approach and introduce the use of discriminative graphical models called *conditional random fields* (CRF) for this task, by expressing it as a sequence labeling problem. Indeed, the CRF framework is aimed at sequence segmentation or labeling, and it allows for the design of more flexible models than hidden Markov and hidden semi-Markov models which are commonly used in the alignment literature. In particular, CRFs allow for the use of a acoustic features extracted from a whole sequence of audio frames, instead of a single observation. We take advantage of this property to design features which perform an implicit modeling of the notion of tempo, at the lowest level of the model.

Furthermore, we propose three different dependency structures for the modeling of the musical event durations, corresponding to different degrees of precision in the modeling of musical event durations. Three types of features are used, characterizing the local harmony, note attacks and tempo.

Experiments run on a large database of classical piano and popular music exhibit very accurate alignments. Indeed, with the best performing system, more than 95 % of the note onsets are detected with a precision finer than 100 ms.

Several traditional features, extracted from different representations of the audio, are considered for the characterization of the local match between the score and the recording. A comparison of these descriptors is conducted on the basis of their efficiency on the alignment task. Furthermore, we address the design of novel features, by learning a linear transformation from the symbolic to any time-frequency audio representation. We explore a best fit strategy (*minimum divergence*) as well as a discriminative criterion (*maximum likelihood*) for the estimation of the optimal mapping and show that such a learning has the potential to increase the alignment accuracy, for all the tested audio representations.

Finally, we explore several strategies to take into account constraints relating to real use cases. In particular, complexity reduction is obtained thanks to a novel dedicated hierarchical pruning strategy. This method takes advantage of the hierarchical structure of music for a multi-pass decoding approach, yielding a better overall efficiency than the beam search method traditionally used in HMM-based models. We additionally show how the proposed framework can be modified in order to be robust to possible structural differences between the score and the musical performance, and we study the scalability properties of the models used.

Table des matières

Notations

Polices de caractères :

majuscule X	pour une variable aléatoire
minuscule x	pour une réalisation de cette variable
caligraphique $\mathcal{X}$	pour un ensemble
$\mathbf{x}_{1:N}$ ou $\mathbf{x}$	pour $x_1, \ldots, x_N$, une séquence de longueur N (les indices peuvent être omis si cela n'occasionne pas d'ambiguïté)

Noms des variables et section de définition :

A	étiquette d'attaque	4.1.1
C	étiquette d'agrégat	3.2.2
D	étiquette d'occupation	3.4.2
G	observation de tempogramme	4.2.3
L	durée d'un agrégat (en nombre de trames)	3.4.1
M	étiquette de mesure	6.2.2
S	observation de flux spectral	4.2.2
T	étiquette de tempo	3.4.3
V	observation de chroma ou d'une autre représentation temps-fréquence	4.2.1 5.1.1
X	étiquette globale : $X = (C, A, D, T)$	3.2.1
Y	observation globale : $Y = (V, S, G)$	3.2.1

Autres symboles

ϕ	fonction d'observation d'un CRF	3.3.1
ψ	fonction de transition d'un CRF	3.3.1
Z	facteur de normalisation d'un CRF	3.3.1
$\mathbf{1}_{\mathcal{E}}$	fonction indicatrice pour un événement $\mathcal{E}$: 1 si $\mathcal{E}$ est vrai, 0 sinon.	
on utilisera aussi :	$\mathbf{1}_{\mathcal{X}}(x) = \mathbf{1}_{\{x \in \mathcal{X}\}}$	

Acronymes

BLCRF	*Beat-Level* CRF	6.3.1
CCMP	Cout de Classification Moyen Pondéré	2.4.1
CGM	ChromaGramme de Müller	5.2.1
CGZ	ChromaGramme de Zhu	5.2.1
CQT	transformée à Q constant (*Constant Q Transform*)	5.2.1
CRF	champ aléatoire conditionnel (*Conditional Random Field*)	3.3
HTCRF	CRF à tempo caché (*Hidden Tempo* CRF)	4.1.3
IMP	Imprécision Moyenne Pondérée	2.4.2
MAP	Maximum *A Posteriori*	3.2.2
MCRF	CRF markovien (Markovian CRF)	4.1.1
MD	Minimum de Divergence	5.3
MV	Maximum de Vraisemblance	5.4
RBD	Réseau Bayésien Dynamique	3.2
RF	Recherche par Faisceaux	6.2.5
SGF	SemiGramme calculé par banc de Filtres	5.2.2
SGQ	SemiGramme calculé par CQT	5.2.2
SMCRF	CRF semi-markovien (*Semi-Markovian CRF*)	4.1.2
SP	spectrogramme (Spectre de Puissance)	5.2.3
TAMP	Taux d'Alignement Moyen Pondéré	2.4.2

Chapitre 1

Introduction générale

1.1 L'Indexation automatique de fichiers musicaux

1.1.1 Contexte : grandes bibliothèques de documents multimédia

Depuis l'avènement d'Internet, n'importe quel utilisateur peut avoir accès, instantanément et sans effort, à une base de données fantastique de documents de toutes sortes (textes, images, sons, vidéos...). Par exemple, le site d'écoute de musique sur Internet www.deezer.com annonce un catalogue de 10 millions de titres en 2011. D'après le site www.youtube.com, plus de 24 heures de vidéos sont mises en ligne chaque minute ! Devant des contenus d'une telle ampleur, il devient alors indispensable d'utiliser des outils automatiques aidant à la navigation ou à la recherche de documents. C'est pourquoi de nombreux travaux dans le domaine du traitement d'informations multimédia se concentrent sur les problématiques d'*indexation automatique*. Ces thématiques visent à créer de nouveaux moyens d'accès aux bases de données, de navigation ou de recommandation de documents.

Actuellement, les moteurs de recherche acceptent des requêtes sous forme textuelle, et recherchent les documents sur la base des noms de fichiers et de l'éventuel contenu textuel des documents, ainsi que de *tags* (étiquettes) textuels associés aux documents. Cette stratégie est fiable, mais elle limite les possibilités de recherches à des requêtes textuelles, correspondant aux mots-clés explicitement présents dans les documents ou dans les *tags* associés. De plus, la création des *tags* par des opérateurs humains est une tâche fastidieuse, ce qui limite en pratique le nombre d'étiquettes utilisées.

On peut alors dégager deux approches dominantes pour la création de nouveaux moyens d'accès aux bases de données. La première vise à caractériser de façon automatique le contenu des documents. Une telle analyse rend possible l'utilisation de n'importe quel mot-clé pour une recherche, sans être limité aux termes explicitement présents dans les documents. De plus, les index ainsi créés peuvent décrire des parties de documents seulement. Cela permet non seulement une identification des documents recherchés, mais aussi une localisation des parties d'intérêt dans ces documents. La seconde approche tire parti de l'analyse des relations entre les documents, notamment la similarité. Cela permet alors d'utiliser des requêtes non uniquement textuelles, mais de format quelconque (image, vidéo, son...). Une

des applications les plus populaires est certainement celle proposée par le service *Shazam*[1], qui permet d'identifier un morceau de musique à partir d'un très court extrait sonore. Il est à noter que ces deux approches peuvent profiter l'une de l'autre. En effet il est possible de comparer deux documents en confrontant les informations issues d'une extraction de contenu. De même, on peut déterminer automatiquement certains *tags* d'un document si l'on découvre qu'il est extrait d'un autre.

1.1.2 Exploitation de la partition pour l'analyse des contenus musicaux

Nous nous intéressons plus particulièrement au cas des contenus musicaux. Un fichier de musique, sous sa forme numérique « brute », est une suite de valeurs décrivant l'onde acoustique produite par les musiciens. L'indexation d'un tel fichier suppose alors d'extraire de ce signal des informations décrivant les caractéristiques de la musique. Ces caractéristiques peuvent être diverses et de différents « niveaux d'abstraction », comme le volume sonore, les notes, l'instrumentation, l'artiste, le rythme, la gamme et les accords utilisés, les conditions d'enregistrement (concert ou studio), le style ou encore l'« humeur » du morceau (joyeux ou triste). Parmi les problèmes les plus étudiées, on peut encore citer la tâche de transcription automatique, qui consiste à retrouver la partition musicale à partir d'un enregistrement sonore.

L'extraction de ces informations caractérisant un morceau de musique est à ce jour sujette à des erreurs. Par exemple, les meilleurs systèmes de reconnaissance d'accords de l'évaluation MIREX 2010 [MIR, 2010] ont des performances de l'ordre de 80 % (ce score mesure la durée d'un morceau sur laquelle les accords sont correctement identifiés). En revanche, un grand nombre de ces informations peuvent être trouvées dans la partition du morceau étudié. Les notes, les rythmes et quelquefois l'instrumentation sont en effet indiqués de façon explicite dans la partition. Or, il existe un grand nombre de partitions dans différentes bibliothèques. En particulier, Internet permet un accès à de nombreuses bases de données de partitions sous format électronique, même si ces documents peuvent être plus ou moins fiables. L'exploitation de ces partitions disponibles librement a donc le potentiel de rendre plus aisée et plus robuste l'analyse des contenus musicaux. Par exemple, l'analyse de l'harmonie et de la tonalité est facilitée par la prise en compte de la représentation symbolique de la musique, de même que la reconnaissance de la mélodie principale.

1.2 L'Alignement musique-sur-partition

1.2.1 Qu'est-ce que l'alignement musique-sur-partition ?

Certaines caractéristiques globales (par exemple la tonalité générale ou le chiffre indicateur de mesure) d'un morceau de musique peuvent être extraites directement de la partition. En revanche, pour une description locale, comme une transcription en accord, le lien doit être trouvé entre les positions dans l'enregistrement et dans la partition. La construction de ce lien constitue l'alignement de la musique sur la partition.

1. http://www.shazam.com

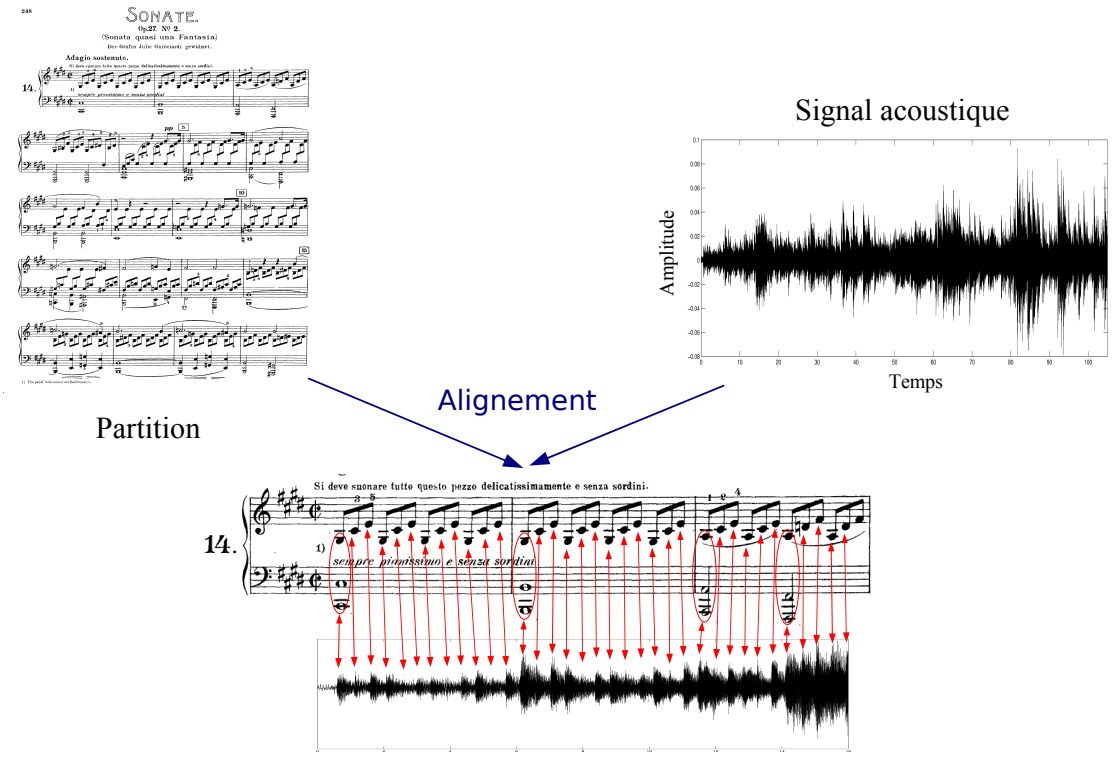


FIGURE 1.1 – Illustration de la tâche d'alignement : à chaque instant de l'enregistrement est associé une position dans la partition, et inversement.

Ce processus est illustré figure 1.1 : à partir d'une partition et d'une interprétation du même morceau de musique, l'alignement fait correspondre à chaque position de la partition la position correspondante dans l'interprétation et inversement. Si la partition est exacte, c'est-à-dire qu'elle décrit parfaitement l'interprétation, cette correspondance est une bijection : une unique position dans l'interprétation correspond à chaque position dans la partition.

Notons qu'en pratique, une partition ne décrit jamais de façon rigoureusement exacte l'interprétation. En effet, une légère désynchronisation entre les différents musiciens (ou les doigts d'un pianiste) est inévitable, même si elle est la plupart du temps inaudible. De ce fait, il est rare que les notes devant être jouées en même temps soient parfaitement simultanées. De plus, dans certains types de musique, notamment pour clavier seul, de telles variations peuvent être délibérément accentuées afin de créer un effet d'expression (accords arpégés). Les travaux d'alignement de Niedermayer et Widmer [2010b] visent à analyser ces recours expressifs dans le cas de musique pour piano. Cependant, en présence de variations entre la partition et l'interprétation (par exemple la répétition d'une note), la relation entre les deux modalités peut perdre son unicité et le problème initial devient alors mal posé. C'est pourquoi la partition est souvent supposée parfaite, afin d'éviter ce type d'ambiguïtés.

1.2.2 Musique sous forme symbolique ou enregistrement sonore

Une interprétation musicale peut être enregistrée sous deux formes différentes. Les premiers systèmes d'alignement, proposés concurremment par Dannenberg [1984] et Vercoe [1984] s'intéressaient à une interprétation sous forme symbolique, en l'occurence un flux MIDI (pour Musical Instrument Digital Interface) [MID]. Le standard MIDI est un protocole de communication numérique pour la commande d'instruments de musique (des synthétiseurs par exemple). Un flux MIDI est alors une séquence de commandes *note-on* (début de note) et *note-off* (fin de note). D'autres commandes existent, comme les changements de volume ou les variations de hauteur de note, mais en général, les seules évènements pris en compte par les systèmes d'alignement sont les débuts de notes. En effet, ce sont ces mêmes évènements que la partition indique de façon fidèle (les instants de fin de notes étant dépendants de l'articulation choisie par l'interprète). La comparaison d'un instant de l'interprétation et d'une position dans la partition est donc immédiate et les alignements attendus sont donc très fiables.

Cependant, la forme la plus courante de capture d'une interprétation musicale est bien sûr l'enregistrement sonore. Dans ce cas, seule l'onde acoustique est décrite et la comparaison avec la partition nécessite des traitements plus complexes. C'est dans ce cadre d'alignement d'un enregistrement sonore sur la partition que se place cette thèse.

1.2.3 Alignement hors ligne, en ligne ou temps réel

Une autre distinction peut être relevée dans les stratégies employées pour l'alignement-il peut être effectué *en-ligne* ou *hors-ligne*. Un alignement en ligne associe une position dans la partition à chaque instant de l'enregistrement, en considérant uniquement le « passé » de l'interprétation. Ces décisions sont donc prises « en continu », à mesure que l'interprétation est parcourue. Une stratégie en ligne rend possible ce que l'on appelle le *suivi de partition*, c'est-à-dire le suivi en temps réel d'une interprétation musicale [Orio *et al.*, 2003]. Néanmoins, il est à noter que pour fonctionner en temps réel, un système doit non seulement effectuer un alignement en ligne, mais aussi vérifier des contraintes supplémentaires, liées aux problèmes de latence.

Dans une stratégie hors ligne, l'interprétation entière est utilisable à toutes les étapes du processus d'alignement, et celui-ci est déterminé globalement. Cette stratégie permet d'atteindre une plus grande précision qu'un alignement en ligne puisqu'elle utilise davantage d'information.

1.2.4 Applications possibles

De nombreuses applications peuvent être rendues possibles grâce à l'alignement de musique sur partition. Un suivi d'interprétation en temps réel permet par exemple l'accompagnement automatique d'un musicien soliste, comme le système *Music Plus One* de Raphael [2001] [2]. Un tel suivi peut aussi être utilisé pour synchroniser des sons électroniques avec des musiciens en conditions de concert. Le système *Antescofo* de Cont [2008a]

2. http_//music-plus-one.com

a été exploité pour un certain nombre de compositions [3]. De façon générale, le suivi de partition peut permettre l'automatisation de nombreux processus nécessitant d'être synchronisés avec une interprétation musicale. Cela comprend par exemple un tourneur de pages automatique [Arzt *et al.*, 2008], des transformations sonores comme l'*auto-tuning* ou le contrôle d'effets visuels comme des surtitres d'opéra ou les éclairages.

D'autres applications, qui ne nécessitent pas un alignement en temps réel, peuvent également être développées. Parmi les plus immédiates, on peut citer la navigation dans un morceau à partir de la partition. Cela peut ainsi permettre de visualiser la partition en même temps que l'on joue un enregistrement, ou encore de sélectionner certaines parties de l'enregistrement grâce à la partition [Müller *et al.*, 2010]. Cela peut aussi constituer une aide pour le montage sonore, en synchronisant automatiquement toutes les prises d'un même morceau sur le plan de montage. L'analyse d'une interprétation musicale peut de même être facilitée par un alignement sur la partition. De telles analyses peuvent être utiles pour les études musicologiques, mais aussi pour l'apprentissage de la musique, en décelant automatiquement les erreurs d'interprétation de l'élève. Réciproquement, l'alignement peut être employé pour détecter de potentielles erreurs de transcription dans une partition, voire pour les corriger.

D'autres problèmes d'indexation peuvent tirer parti des deux modalités (audio et symbolique) pour une analyse du contenu musical (structure, tonalité, mélodie principale...). La partition peut aussi être utilisée pour fournir d'importantes informations dans la tâche de séparation de source [Hennequin *et al.*, 2011]. Un autre intérêt est l'annotation de signaux audio, pour l'apprentissage et/ou l'évaluation de systèmes de transcription automatiques. Enfin, l'alignement peut être utilisé dans une application de recherche d'enregistrements à partir d'une partition-requête, ou à l'inverse, de l'identification d'un enregistrement parmi une base de données de partitions [Orio, 2007].

1.3 Problématique et contributions de cette thèse

1.3.1 Propriétés souhaitées d'un système d'alignement

Les propriétés d'un système d'alignement musique-sur-partition idéal sont de plusieurs ordres. Bien entendu, la précision des alignements obtenus est un caractère majeur. Néanmoins, on peut aussi retenir d'autres critères importants, en particulier la *généralité*, la *scalabilité* et la *robustesse*.

Par *généralité*, nous entendons la capacité à traiter une grande variété de musiques. Les performances d'un système général devraient donc être indépendantes du style de musique, de l'instrumentation, de la polyphonie (c'est-à-dire le nombre de notes différentes jouées simultanément) ou encore du tempo. Cependant, de nombreux travaux sur cette tâche se sont limités, par exemple à de la musique monophonique [Cano *et al.*, 1999 ; Raphael, 1999] ou mono-instrumentale [Grubb et Dannenberg, 1998 ; Cont, 2006]. De plus, la plupart des campagnes d'évaluation internationales utilisent uniquement de la musique classique, où la polyphonie est limitée. La base de données de la campagne MIREX 2006 [MIR, 2006]

3. Répertoire d'Antescofo : http://repmus.ircam.fr/antescofo/repertoire

comprenait en effet uniquement de la musique pour instrument solo. C'est pour la campagne 2010 [MIR, 2010] que des morceaux multi-instrumentaux ont été introduits. Néanmoins, la musique utilisée est encore exclusivement classique. Il est en revanche inévitable de se limiter à des musiques pouvant être écrites sous forme symbolique et dont la partition existe.

La *scalabilité* d'un système désigne la faculté à opérer pour différents positionnements du compromis précision/complexité. Cette propriété n'a, à notre connaissance, pas été prise en compte dans la littérature. Pourtant, elle peut être considérée comme importante car le compromis entre la précision et la complexité d'un alignement peut dépendre de l'application visée. Par exemple, une faible complexité sera favorisée dans une application de recherche de partition d'après un enregistrement-requête, car un temps de réponse court doit être maintenu. En revanche, pour une application de séparation de sources informée, la finesse de la caractérisation du signal sera préférée à la vitesse d'exécution.

La propriété de *robustesse* que nous considérons désigne la capacité à prendre en compte des partitions caractérisant l'enregistrement de façon imparfaite. En d'autres termes, il s'agit de traiter certaines divergences entre l'enregistrement et la partition. Nous distinguons plusieurs types de différences : les « erreurs », qui concernent des notes isolées ou des passages de très courtes durées (par exemple une fausse note d'un interprète ou une erreur de transcription), et les « modifications structurelles », qui correspondent à des variations d'une grande partie du morceau, comme la suppression d'une reprise ou l'ajout d'un solo. Cette propriété de robustesse est importante dans le but de pouvoir aligner plusieurs versions d'un même morceau (par exemple une version studio et un enregistrement de concert) et de tirer parti des partitions de qualité variable que l'on peut trouver sur Internet. On peut en outre considérer comme un problème de robustesse la prise en compte d'une partition ne définissant qu'en partie la musique. De telles partitions incomplètes sont courantes notamment dans le jazz, où la plupart des standards sont diffusés sous forme de *lead sheets* décrivant uniquement la mélodie principale et les accords.

L'objectif de cette thèse est alors de tenter de répondre à ces attentes en proposant de nouveaux systèmes pour un alignement musique sur partition précis et robuste dans des conditions réalistes. Nous présentons pour cela un cadre unifié de modèles graphiques discriminatifs permettant de traiter des enregistrements polyphoniques, où le nombre et la nature des instruments est quelconque et dont la structure peut être différente de celle de la partition. À l'intérieur de ce cadre, plusieurs structures de modèles sont possibles, donnant lieu à différentes valeurs du compromis précision-complexité.

1.3.2 Contributions

Les contributions de cette thèse peuvent être résumées comme suit :

- L'utilisation des Champs Aléatoires Conditionnels (CRF pour *Conditional Random Fields*) est proposée dans le chapitre 3 pour l'alignement musique sur partition. Ce formalisme fournit un cadre probabiliste plus large que la classe des Réseaux Bayésiens Dynamiques, auxquels appartiennent les modèles statistiques de la littérature. Ces modèles peuvent donc être présentés de manière unifiée dans le formalisme CRF.

- Plusieurs architectures de modèles graphiques sont présentées chapitre 4, modélisant la dimension temporelle de la musique avec différents degrés de complexité.
- Nous définissons une « fonction d'observation » tenant compte d'un voisinage autour de chaque instant de l'enregistrement pour estimer sa similarité avec les positions possibles dans la partition. Une telle fonction de similarité peut être intégrée dans un modèle probabiliste grâce au cadre CRF. En outre, nous proposons l'utilisation de descripteurs acoustiques caractérisant explicitement le tempo dans la fonction d'observation.
- La fonction mesurant la correspondance locale entre le contenu instantané de l'enregistrement et les notes indiquées dans la partition est formalisée au chapitre 5 grâce à l'exploitation d'une transformation linéaire, d'après une représentation vectorielle de la partition. De cette manière, un même formalisme peut être appliqué à différents descripteurs « spectraux » du signal. Cinq descripteurs sont testés et comparés.
- Différentes stratégies d'optimisation de cette transformation du domaine symbolique vers le domaine des descripteurs acoustiques sont proposées et étudiées, d'après plusieurs critères. Nous montrons que cette optimisation peut améliorer la précision des alignements de manière significative.
- Une modification des systèmes visant à prendre en compte des différences structurelles entre la partition et l'enregistrement est implémentée au chapitre 6. Nous montrons que cette modification permet de répondre à ce problème, en affectant peu la précision sur des partitions exactes.
- Nous proposons une méthode d'élagage hiérarchique pour la réduction de la complexité de l'alignement. Cette stratégie hors ligne tire parti de la structure de la musique en pulsations et mesures afin d'effectuer un décodage approché des modèles probabilistes. Cette méthode améliore de façon significative le temps d'exécution et les besoins en mémoire des systèmes d'alignement, sans nuire à la précision des résultats.
- Une importante validation expérimentale des systèmes d'alignement est conduite, sur une grande base de données contenant plusieurs styles de musiques (classique et pop) et des instrumentations variées.
- Nous menons une étude de la complexité des modèles présentés, exploitable en vue de la construction d'un système scalable d'alignement temporel.

Plan du document

À la suite de cette introduction, le document est organisé comme suit.

Nous commençons dans le chapitre 2 par présenter plus précisément la problème de l'alignement musique sur partition. Nous détaillons tout d'abord la structure d'un système d'alignement, avant de présenter un état de l'art sur le sujet. Nous introduisons alors la base de données utilisées dans ce travail et nous discutons comment évaluer la performance des systèmes d'alignement.

Le chapitre 3 décrit les outils statistiques utilisés dans cette thèse. Après avoir défini de manière générale les modèles graphiques, nous illustrons la façon dont ce formalisme est exploité dans les systèmes d'alignement de la littérature, grâce à des modèles graphique génératifs de type réseau bayésien dynamique (RBD). Nous présentons alors les champs

aléatoires conditionnels (CRF) et montrons comment ces modèles constituent une généralisation des RBD dans notre tâche d'alignement. Trois structures de CRF permettant une modélisation temporelle de plus en plus précise des données sont étudiées.

Nous détaillons ensuite dans le chapitre 4 les différents modèles CRF proposés pour l'alignement. Nous nous intéressons tout d'abord aux contraintes de transitions existant entre les variables aléatoires considérées. Puis nous présentons la fonction d'observation, faisant le lien entre les observations extraites de l'enregistrement et la position dans la partition, avant d'exposer la stratégie de décodage des modèles obtenus. Nous menons alors une étude expérimentale des performances de ces modèles sur la tâche d'alignement.

Le chapitre 5 examine plusieurs approches d'optimisation de la fonction d'observation. Grâce à une formulation unifiée de la mesure de similarité locale entre enregistrement et partition, nous comparons de multiples versions de cette fonction d'observation, issues de différentes représentations de l'audio. Deux stratégies d'apprentissage sont alors expérimentées et nous évaluons leurs influences sur les résultats d'alignement des modèles CRF précédents.

Dans le chapitre 6, sont étudiées différentes méthodes pour prendre en compte les contraintes d'utilisation dans des conditions réalistes. Nous proposons une approche originale d'élagage hiérarchique pour une réduction de la complexité de décodage. Puis nous examinons l'influence de certaines modifications de structure entre la partition et l'enregistrement, avant d'étudier les propriétés de scalabilité des modèles utilisés.

Enfin, nous concluons par un résumé des principales contributions de cette thèse et nous proposons des pistes de recherche pour une amélioration des systèmes d'alignement.

Chapitre 2

Alignement musique-sur-partition : introduction et état de l'art

Dans ce chapitre, nous présentons de manière plus approfondie le domaine de l'alignement de musique sur partition.

Nous détaillons tout d'abord la structure d'un système d'alignement. S'appuyant sur une représentation linéaire de la partition en une séquence d'objets appelés *agrégats*, il peut se décomposer en deux « couches ». La première couche, dite de bas niveau, calcule une mesure de similarité entre chaque agrégat de la partition et chaque instant de l'enregistrement. La couche de haut niveau est alors chargée de relier ces scores locaux afin de rechercher la meilleure correspondance entre les deux séquences. Un panorama des approches constituant l'état de l'art est alors présenté, pour chacune de ces deux parties. Nous discutons comment évaluer la performance des systèmes d'alignement et nous exposons ensuite les données utilisées dans ce travail.

2.1 Structure d'un système d'alignement

2.1.1 Prise en compte de la partition

Avant de s'atteler à la tâche d'alignement audio-sur-partition, il est bon de définir ce que nous désignons par une partition. Dans le sens le plus général, une partition est une représentation symbolique des sons qui composent une pièce de musique. Cependant, cette définition est trop vague pour être véritablement exploitée. Des restrictions sont donc apportées, inspirées par la notation de la musique classique occidentale, pour pouvoir mettre en œuvre un système d'alignement automatique.

En premier lieu, nous considérons que la partition est une liste (discrète) d'« évènements sonores ». Les évènements sonores peuvent être des notes ou des frappes de percussions. Cette représentation est une simplification car les frontières entre ces évènements peuvent être floues, voire impossible à définir, comme dans le cas d'un *glissando* entre deux notes.

En second lieu, chaque évènement doit pouvoir être caractérisé par un certain nombre (petit) de valeurs numériques, représentant des paramètres musicaux (comme la hauteur

des notes, l'instrument, l'intensité, etc. . .). Cela implique de ne pas prendre en compte certaines informations qui ne peuvent être converties en nombre (par exemple des indications de caractère d'interprétation comme « passionné » ou « majestueux »). De plus, les valeurs numériques caractérisant les évènements sont supposées constantes sur la durée de la note. Il en résulte que certains phénomènes comme le *crescendo* doivent être simplifiés (dans ce cas, séparer en plusieurs évènements d'intensité croissante) ou simplement supprimés.

Enfin, afin de pouvoir mettre en correspondance une partition musicale et son interprétation, il est souhaitable de supprimer les ambiguïtés dans la séquence d'évènements sonores correspondant à une interprétation. Cette contrainte concerne la structure temporelle des évènements. Nous faisons alors l'hypothèse que les évènements sont totalement ordonnés, c'est-à-dire qu'ils doivent être joués dans un ordre défini par la partition. De plus, nous supposons que les instants d'attaque et d'extinction de chaque évènement sont spécifiés dans une échelle de durée musicale. Ces durées s'expriment en *pulsation*, aussi appelés *temps*. La correspondance entre les durées musicales (en pulsations) et les durées réelles (en secondes) est donnée par le *tempo*.

Notre spécification d'une partition pour une tâche d'alignement temporel peut donc être résumée comme une séquence discrète d'évènements sonores, caractérisés par un vecteur de paramètres musicaux ainsi que deux index temporels de début (attaque) et de fin (extinction), exprimés en pulsations. Dans cet ouvrage, les évènements considérés sont les notes de musique, dont les seuls paramètres de hauteur sont utilisés. Cette décision est motivée par les applications potentielles visées, qui exploitent des partitions informatiques extraites de différentes sources (export d'un éditeur de partition graphique, enregistrement à partir d'une interface numérique ou résultat d'une analyse optique de partition musicale). Ces partitions peuvent être de qualité variable et de multiples indications peuvent être non fiables ou manquantes, en particulier dans les styles autres que classique, où les partitions sont en général transcrites « à l'oreille ». Les parties de percussions, en particulier, sont souvent annotées de façon imprécise. De plus, dans le cas de fichiers issus de partitions graphiques, les informations de dynamiques (intensité sonore) sont absentes.

D'autre part, il est nécessaire de faire appel à une approche d'apprentissage des timbres des instruments de musique, si l'on veut exploiter les indications d'instruments qui peuvent être présentes dans la partition. C'est pourquoi nous choisissons de ne pas utiliser ces informations, afin de n'avoir pas recours à de telles stratégies couteuses.

Pour nos expériences, nous utilisons le format *standard MIDI file*, qui condense un flux MIDI [MID] dans un fichier informatique. Ce format permet en effet de représenter les informations que nous exploitons. De plus, c'est un des formats les plus répandus pour la représentation symbolique de la musique et de nombreuses partitions sous ce format peuvent être trouvées facilement sur Internet. Un fichier MIDI peut contenir plusieurs pistes, associées généralement aux différents instruments présents. Chaque piste est alors constitué d'une suite de commandes, dont les plus principales sont les *note-on* (début de note) et *note-off* (fin de note). Ces commandes indiquent la hauteur de la note, codée selon la gamme chromatique tempérée, ainsi qu'une information d'intensité (appelée *vélocité*). D'autres commandes existent, comme les changements d'intensité ou de hauteur de note, mais nous avons vu qu'elles n'étaient pas exploitées par le sytème d'alignement. À chaque commande est associé un index temporel exprimé en pulsations, dont la correspondance

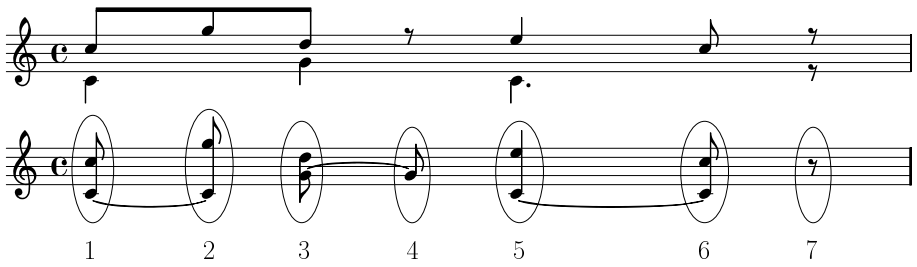


FIGURE 2.1 – Segmentation de la partition en agrégats. Haut : partition graphique originale. Bas : représentation en séquence d'agrégats (numérotés ici pour plus de clarté). Dans cet exemple, plusieurs notes *liées* apparaissent, aux agrégats 2, 4 et 6. L'agrégat 4 est un agrégat *lié*, puisque la seule note qu'il contient est *liée*. Tous les autres agrégats sont *attaqués* car ils comportent au moins une note *attaquée* (non liée).

avec le temps effectif est contrôlée par des commandes spécifiques[1]. Comme indiqué plus haut, nous extrayons de chaque fichier toutes les notes présentes dans le morceau, ainsi que leurs instants de début et de fin.

2.1.2 Représentation d'une partition polyphonique

La tâche d'alignement telle que nous la définissons consiste à déterminer, pour chaque position dans l'enregistrement, l'évènement musical qui est joué. Cependant, dans le cas de musique polyphonique, où plusieurs notes peuvent être jouées en même temps, des « évènements complexes » doivent pouvoir être pris en compte. Afin de préserver la relation d'ordre total des évènements (autrement dit la structure séquentielle de la partition), tous les systèmes d'alignement de musique polyphonique adoptent la même approche (par exemple Raphael [2006]) : la partition polyphonique est découpée en segments homogènes, c'est-à-dire dont le contenu (en terme d'évènements sonores) est constant. Ces unités sont appelées agrégats. La segmentation est donc une segmentation temporelle, regroupant « verticalement » les notes : à chaque début ou fin de note, un nouvel agrégat est créé, contenant tous les évènements présents à cet instant. Nous appelons *agrégats* les éléments ainsi séparés. Un agrégat est donc un « évènement composé », défini comme un ensemble d'un ou plusieurs évènements (notes) joués en même temps. Un silence est représenté par un agrégat vide. La figure 2.1 illustre la création de cette représentation de la partition en une séquence d'agrégats.

Cette représentation soulève deux considérations. Premièrement, elle implique l'hypothèse que l'interprétation de la partition suivra le même ordre temporel que celui indiqué dans la partition. Cette hypothèse est une simplification car elle suppose en particulier une synchronisation parfaite entre les différents musiciens (ou les doigts du pianiste). En effet, si deux notes censées être attaquées simultanément ne sont pas jouées rigoureusement en même temps, l'enregistrement comportera deux agrégats (le premier contenant la première

1. En toute rigueur, un fichier MIDI utilise une échelle temporelle appelée le « tick », mais le nombre de ticks par pulsations est fixé à une valeur constante sur chaque morceau

note seule, le second contenant les deux notes) au lieu d'un seul. De la même façon, les accords « arpégés » ne sont pas pris en compte par cette représentation.

De plus, lors de la segmentation de la partition, une note peut être séparée en plusieurs agrégats successifs, si une autre note apparait ou disparait pendant que la première sonne. Cela est représenté par une *liaison* dans l'écriture musicale classique. Dans le cadre de notre représentation, les notes contenues dans chaque agrégat sont séparées en deux catégories : les notes *attaquées*, non présentes dans l'agrégat précédent, et les notes *liées*, qui correspondent aux notes liées dans la notation classique. Cette distinction est nécessaire afin de différencier une note liée d'une note répétée (arrêtée, puis jouée à nouveau). Par généralisation, nous appellerons *agrégat attaqué* un agrégat contenant au moins une note attaquée et un *agrégat lié* sera un agrégat comportant uniquement des notes liées. Cette distinction est illustrée figure 2.1.

Une partition polyphonique est donc représentée par l'ensemble $\{(\tau_c, \dot{\mathcal{J}}_c, \breve{\mathcal{J}}_k)\}_{c=1...Q_C}$, où τ_c est l'index temporel de début de l'agrégat c, exprimé en pulsations (l'index de fin n'est pas nécessaire puisqu'il correspond au début de l'agrégat suivant), $\dot{\mathcal{J}}_c$ et $\breve{\mathcal{J}}_c$ sont des ensembles contenant respectivement les hauteurs des notes attaquées et celles des notes liées. Q_C est le nombre d'agrégats de la partition. Cet ensemble peut être assimilé à une séquence, puisque les éléments sont totalement ordonnés par leurs index temporels. Par convention, le premier et le dernier accord sont vides et on a $\tau_1 = 0$.

2.1.3 Principe général d'un système d'alignement temporel

Après avoir défini la représentation d'une partition musicale, nous pouvons déterminer comment réaliser l'alignement temporel d'une partition sur un enregistrement. Pour cela, l'enregistrement est tout d'abord segmenté en fenêtres temporelles (ou trames), assez courtes pour pouvoir considérer que le signal est stationnaire, c'est-à-dire que les « caractéristiques acoustiques » de l'enregistrement sont constantes, sur la durée de chacune de ces trames. En particulier, on suppose qu'une trame ne contient qu'un seul agrégat. Cette hypothèse est bien sûr une approximation, mais elle ne porte pas à conséquence car la précision temporelle recherchée, qui correspond au pouvoir séparateur de l'oreille, est du même ordre de grandeur que la longueur des trames (quelques dizaines de millisecondes). La tâche d'alignement peut alors être réduite à la détermination de l'agrégat correspondant à chaque fenêtre temporelle, parmi ceux indiqués par la partition.

La structure de tous les systèmes de l'état de l'art peut être divisée en deux « couches ». Une première couche, dite couche de bas niveau ou encore modèle d'observation, calcule un *score d'appariement local* entre une trame et un agrégat de la partition. Pour cela, on extrait des *descripteurs acoustiques*, aussi appelés *caractéristiques*, qui sont des valeurs numériques décrivant le contenu instantané du son à l'intérieur de chaque trame. Pour l'alignement temporel, on utilise des descripteurs qui peuvent permettre de déduire les notes jouées, mais aussi parfois des caractéristiques décrivant d'autres types d'information, comme la détection des attaques de notes. On utilise le terme « matrice de similarité » pour désigner le tableau contenant les scores d'appariement locaux entre toutes les paires trame-agrégat.

La couche de haut niveau, ou modèle temporel, opère alors l'alignement proprement dit, en tenant compte de ces scores d'appariement local et de certains critères sur l'évolution

temporelle des agrégats joués. Il est par exemple très courant de considérer les séquences d'agrégats jouées dans le même ordre que celui de la partition. Un autre exemple consiste à appliquer une pénalité liée à la longueur de chaque agrégat détecté. Nous reviendrons plus en détail sur les modèles classique d'évolution temporelle dans la section 2.3. Un schéma illustrant la structure d'un système d'alignement est présenté figure 2.2.

2.2 Paramétrisations de l'audio (couche de bas niveau)

Dans cette section, nous décrivons les descripteurs acoustiques usuellement utilisés pour l'alignement musique-sur-partition, ainsi que les scores d'appariement calculés entre chaque instant (trame) de l'interprétation et les positions possibles dans la partition. Il est à noter qu'un même système peut exploiter plusieurs descripteurs, en combinant les scores d'appariement correspondants.

2.2.1 Information de hauteur des notes

La couche de bas niveau d'un système d'alignement calcule un score de similarité entre chaque agrégat de la partition et chaque trame de l'enregistrement. Or, les agrégats sont caractérisés par les notes qu'ils contiennent. C'est pourquoi tous les travaux d'alignement se fondent sur une représentation de l'audio décrivant le contenu fréquentiel de chaque trame. Nous présentons ci-dessous les principales représentations utilisées dans le contexte de l'alignement.

Représentation symbolique (transcription)

Un certain nombre de systèmes de suivi de partition s'intéressent à une interprétation de la musique sous forme symbolique [Vercoe, 1984 ; Dannenberg, 1984 ; Vercoe et Puckette, 1985 ; Bloch et Dannenberg, 1985 ; Dannenberg et Mukaino, 1988 ; Baird *et al.*, 1990 ; Large, 1993 ; Grubb et Dannenberg, 1994 ; Vantomme, 1995 ; Heijink *et al.*, 2000 ; Schwarz *et al.*, 2004 ; Tekin *et al.*, 2005 ; Jordanous et Smaill, 2009 ; Raphael et Gu, 2009], en l'occurence un flux MIDI. Cela permet une comparaison immédiate entre les notes de l'interprétation et celles de la partition. Dans le cas d'une mélodie monophonique, on a une mesure de similarité binaire (égal ou différent). Pour tenir compte de la polyphonie, le score d'appariement peut être construit comme la proportion (ou le nombre) de notes égales, ou mesurer directement la similarité de deux agrégats d'après un modèle estimé *a priori*[Pardo et Birmingham, 2001]. Le travail de Baird *et al.* [1993] constitue un cas particuler, où la couche bas-niveau mesure l'appariement d'un motif musical entier (une séquence de longueur fixée) de l'interprétation avec une position dans la partition. En revanche, ce travail est limité à de la musique monophonique.

Certains travaux traitant d'enregistrements sonores mettent en œuvre les mêmes types de techniques d'alignement que pour la musique symbolique, en exploitant la sortie d'un algorithme de transcription automatique. C'est le cas des systèmes de Puckette et Lippe [1992] ; Puckette [1995], et İzmirli *et al.* [2002] destinés au suivi en temps réel de signaux monophoniques (de voix chantée pour les deux premiers et de clarinette pour le dernier),

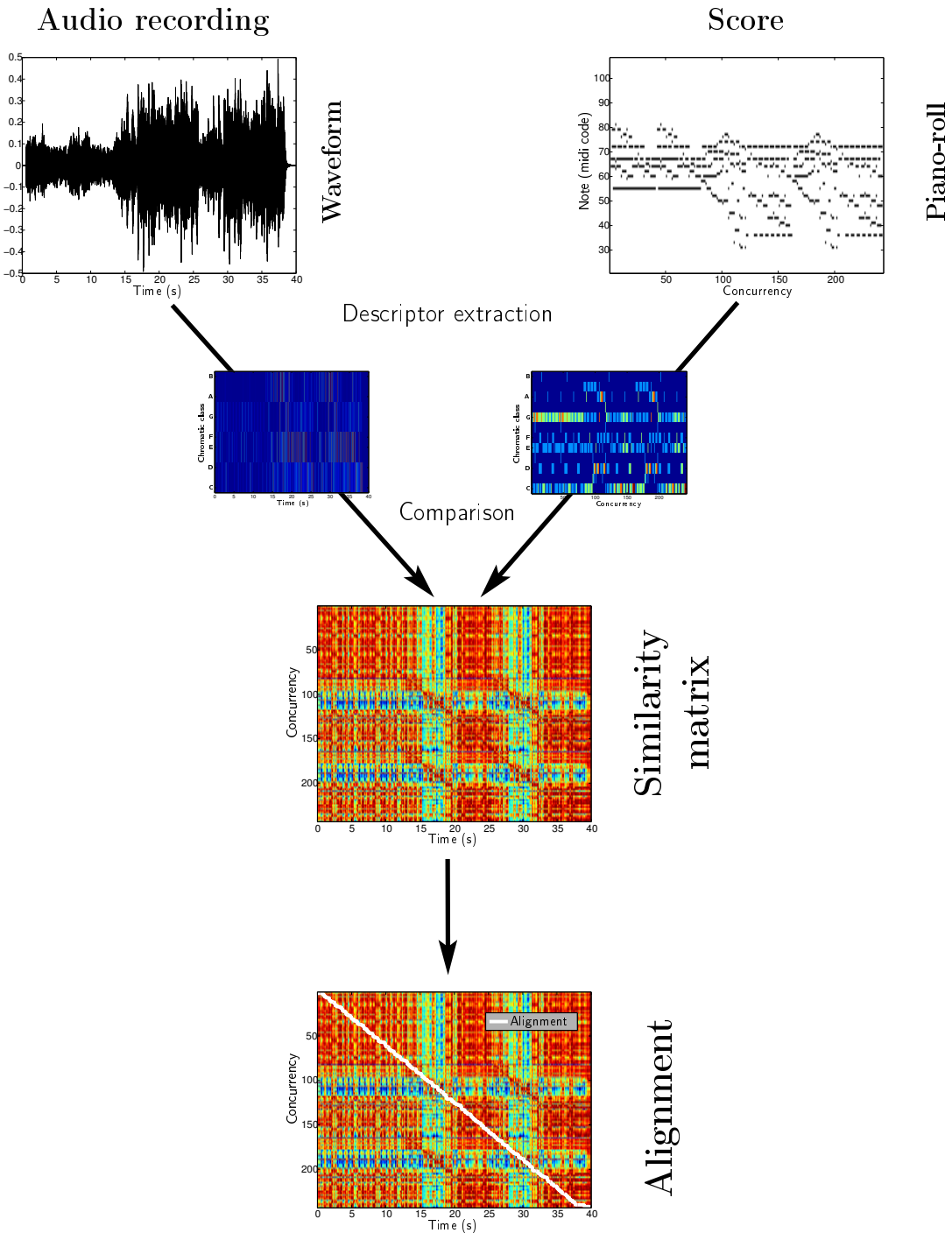


Figure 2.2 – Structure d'un système d'alignement musique sur partition.

ainsi que du travail de Arifi *et al.* [2005] pour l'alignement hors-ligne de musique polyphonique. Néanmoins, ce dernier système a été testé uniquement sur des morceaux joués au piano et les auteurs rapportent que la phase de transcription requiert un choix minutieux des paramètres de l'algorithme.

D'autres systèmes pour l'alignement d'enregistrements audio se fondent encore sur une estimation des fréquences fondamentales, en autorisant des fréquences qui ne correspondent pas exactement à une note de la gamme. C'est le cas des systèmes de Grubb et Dannenberg [1997, 1998] (repris dans [Grubb, 1998]), Cano *et al.* [1999], Pardo et Birmingham [2002, 2005] et Devaney *et al.* [2009], qui traitent des enregistrements monophoniques. Il est à noter que ces systèmes peuvent aussi utiliser d'autres descripteurs acoustiques de bas niveau comme l'amplitude du signal ou sa dérivée. Les mesures de similarité sont alors données par un modèle probabiliste.

Cependant, dans un cadre polyphonique, les estimateurs automatiques de fréquences fondamentales peuvent être complexes et sont sujets à des erreurs, notamment des *erreurs d'octave* ou *de quinte* lorsque la musique est très consonante. C'est pourquoi dans ce contexte, des descripteurs de plus bas niveau sont extraits afin de conserver le plus d'information possible, même si cette information n'est pas toujours directement interprétable.

Spectrogramme

Le spectre de puissance, issu d'une transformée de Fourier à court terme est couramment employé, en particulier pour le suivi de partition en temps-réel. Ainsi, Meron et Hirose [2001] calculent le score de similarité entre un agrégat et un spectre de puissance comme la moyenne des énergies correspondant aux premières harmoniques des notes de l'agrégat. Les systèmes décrits par Orio et Déchelle [2001] ; Orio et Schwarz [2001] ; Orio [2002] ; Soulez *et al.* [2003] ; Cont *et al.* [2005] ; Montecchio et Orio [2008] mesurent la proportion de l'énergie spectrale située autour de ces fréquences.

Dans les travaux de Cont [2008a,b, 2010], ce score est calculé d'après une mesure de divergence entre le spectre de puissance et un gabarit théorique correspondant à l'agrégat, afin de mieux modéliser la distribution d'énergie entre les différents partiels de chaque note. Des stratégies similaires sont utilisées dans la plupart des autres systèmes d'alignement, où les gabarits peuvent être heuristiques [Devaney et Ellis, 2009], déduits d'un modèle probabiliste [Raphael, 2004, 2006 ; Peeling *et al.*, 2007] ou encore extraits d'une synthèse sonore de la partition MIDI [Turetsky et Ellis, 2003].

Le spectrogramme peut être calculé très rapidement par l'algorithme de transformée de Fourier rapide (FFT pour *Fast Fourier Transform*). Cependant, cette faible complexité s'accompagne d'une dimension relativement élevée (par exemple 640 pour une taille de fenêtre d'analyse de 40 ms). De plus, le fait de représenter fidèlement le timbre des sons peut être un désavantage dans les méthodes précédentes, où les modèles d'observations ne tiennent pas compte de cette variabilité.

Le travail de Maezawa *et al.* [2011] constitue alors un cas particulier intéressant. En effet, une approche bayésienne leur permet d'estimer les paramètres des modèles d'observations (qui sont alors spécifiques à chaque instrument) sur chaque morceau traité, conjointement au calcul de l'alignement.

Le score d'appariement de Otsuka *et al.* [2011] est fondé sur l'utilisation de gabarits théoriques, mais il exploite aussi la structure temporelle de la musique. Pour cela, la séquence des observations issues du passé immédiat de la trame courante est comparée aux séquences de gabarits correspondant aux positions dans la partition. Plusieurs hypothèses de tempo sont considérées dans la création des séquences de gabarits. Cela permet d'une part de rendre la matrice de similarité plus fiable et d'autre part de donner un score de confiance aux différentes valeurs de tempo.

« Semigramme »

Une autre représentation est formée par les énergies à la sortie d'un banc de filtres espacés logarithmiquement, correspondant aux demi-tons de la gamme musicale tempérée. Cette paramétrisation, appelée *semigramme* par İzmirli et Dannenberg [2010], fournit une interprétation plus intuitive du contenu spectral de l'audio, en rapport avec l'échelle utilisée pour représenter les hauteurs de notes de chaque agrégat. Par rapport au spectrogramme, elle présente l'avantage d'une plus grande précision dans les basses fréquences, tout en conservant une dimension plus petite. Montecchio et Orio [2009] proposent un système dans lequel les descripteurs sont comparés à des vecteurs-gabarits construits d'après les agrégats. Néanmoins, cette représentation présente encore une dépendance aux timbres des instruments utilisés.

Chromagramme

Le *chromagramme* (aussi appelé représentation en *vecteurs de chroma* ou *pitch class profile* en anglais) est une paramétrisation fréquemment employée dans l'analyse automatique de l'harmonie d'un morceau [Peeters, 2004 ; Zhu et Kankanhalli, 2006]. Cette représentation consiste en un « repliement » du semigramme sur une seule octave. Plus précisément, on sépare la gamme musicale en 12 *classes chromatiques*, correspondant aux noms des notes (do, do#, ré,..., si). Un *vecteur de chroma* est alors un vecteur à 12 composante, dont chaque valeur est l'énergie cumulée des bandes de fréquences correspondant à une classe chromatique. Un exemple de chromagramme est visible figure 2.3.

Cette représentation présente l'avantage d'être robuste aux différences d'octaves (par construction) et, dans une certaine mesure, aux différences de timbre. C'est pourquoi elle est couramment utilisée pour l'alignement audio/audio, mettant en jeux deux enregistrements sonores [Müller et Appelt, 2008 ; Dixon et Widmer, 2005] ou un enregistrement et un fichier audio issu de la synthèse d'une partition MIDI [Dannenberg et Hu, 2003]. Le score d'appariement est alors donné par une simple mesure de similarité entre les deux vecteurs de chroma.

Il est possible d'exploiter le même type de score d'appariement sans passer par une synthèse audio. Il suffit en effet d'associer un vecteur de chroma-gabarit à chaque agrégat. Cette stratégie est employée dans de nombreux systèmes d'alignement hors-ligne [Hu *et al.*, 2003 ; Müller *et al.*, 2005a,b, 2006 ; Müller et Ewert, 2008 ; Ewert *et al.*, 2009 ; Fremerey *et al.*, 2009, 2010 ; Niedermayer, 2009a ; Niedermayer et Widmer, 2010a,b] ou même en-ligne [Fox et Quinn, 2007 ; Otsuka *et al.*, 2009, 2011]. Un exemple de gabarit de chroma

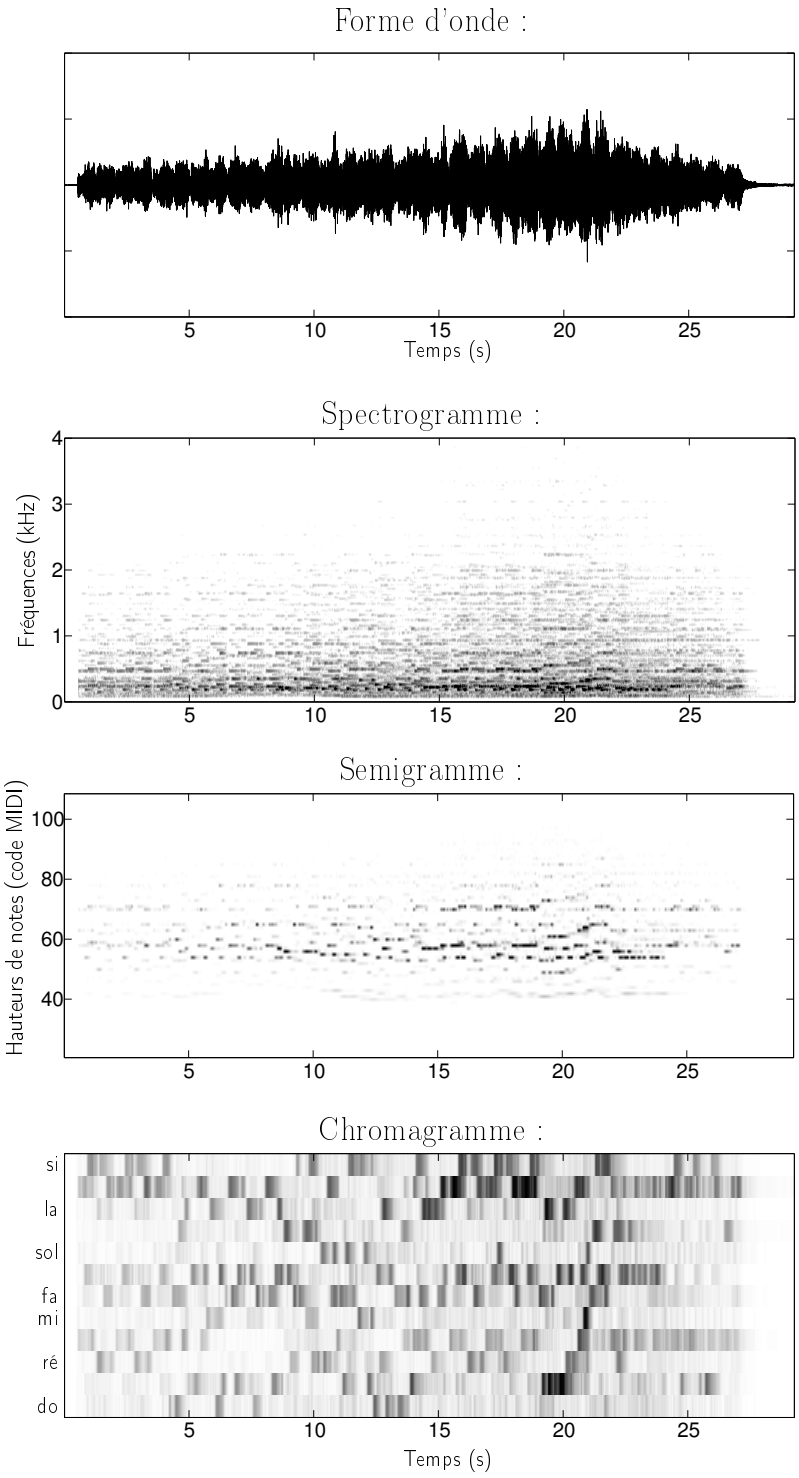


FIGURE 2.3 – Exemple de descripteurs de hauteurs de notes extraits d'un morceau de piano.

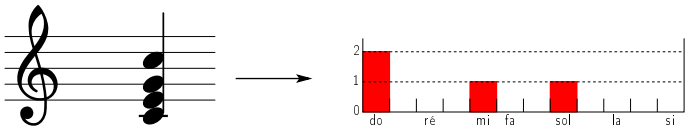


Figure 2.4 – Exemple de construction du gabarit théorique de vecteur de chroma à partir d'un agrégat de 4 notes de la partition.

est représenté figure 2.4.

Décomposition en matrices non négatives

Dans les travaux de Niedermayer [2009b,a] ; Niedermayer et Widmer [2010a,b] et Cont [2006], des descripteurs de niveau intermédiaire sont extraits par une décomposition en matrices non-négatives des spectres de puissance, d'après un dictionnaire de gabarits correspondant aux notes possibles. Une telle décomposition fournit alors des « coefficients d'activation » positifs ou nuls, associés aux notes de la gamme. Les coefficients d'activation de chaque trame —ou leurs dérivées temporelles— peuvent ensuite être directement comparés à la composition d'un agrégat. Cette méthode permet alors une précision temporelle accrue. En revanche, elle nécessite l'apprentissage d'un dictionnaire de spectres qui sont spécifiques à chaque instrument, ce qui la rend peu générale.

Autres descripteurs spectraux

D'autres paramétrisations sont aussi trouvées dans la littérature. Celle utilisée par Dixon [2005] ; Dixon et Widmer [2005] ; Arzt *et al.* [2008] ; Arzt et Widmer [2010a,b] ; Macrae et Dixon [2010] est en quelque sorte une hybridation entre le spectrogramme et le semigramme, puisqu'il s'agit d'un spectre de puissance distribué selon une échelle linéaire en basses fréquences et logarithmique en hautes fréquences. Camarena-Ibarrola et Chávez [2010] emploient une représentation inspirée des techniques de tatouage audio [Cano *et al.*, 2005]. L'entropie du spectre à court terme d'une trame audio est calculée sur 24 bandes de fréquences d'une échelle psycho-acoustique. Chaque observation est alors un vecteur binaire indiquant si l'entropie sur chaque bande est supérieure à celle de la trame précédente. Ces deux représentations sont conçues pour l'alignement de deux enregistrements audio. Leur utilisation pour la tâche d'alignement audio sur partition nécessite donc une synthèse de cette partition. La couche de bas niveau obtenue est alors dépendante du synthétiseur utilisé.

Raphael [1999] extrait les maximums locaux du spectrogramme dans un certain nombre de sous-bandes. À cela s'ajoute l'énergie totale du signal et un estimateur de l'accélération de cette énergie. Les valeurs sont alors discrétisées (quantifiées) et les probabilités conditionnelles de chaque vecteur d'observation sachant les agrégats sont calculées d'après un modèle probabiliste. Cette représentation est de plus faible dimension que le spectrogramme, mais son utilisation est limitée au suivi d'un instrument monophonique. Peeling *et al.* [2007] utilisent eux aussi un modèle probabiliste (en l'occurrence un mélange de gaussiennes) liant les agrégats aux paramètres d'une décomposition du signal en une

somme de sinusoïdes amorties. Ils rapportent une précision temporelle meilleure que celle du spectrogramme, au prix d'une complexité significativement plus élevée, dûe à l'analyse sinusoïdale.

2.2.2 Détection d'attaques

Afin d'associer de façon précise les débuts de notes, certains systèmes exploitent des *fonctions de détection de transitoires*, dont le but est de détecter les attaques de notes et les sons percussifs. Dans le système de Turetsky et Ellis [2003] les dérivées temporelles des valeurs du spectrogramme étendent les vecteurs d'observation de l'enregistrement et de la partition synthétisée.

D'autres modèles ne passant pas par une synthèse de la partition tirent aussi parti de fonctions de détection d'attaques. Afin de discriminer les attaques des différentes notes, une fonction différente peut être calculée pour chaque composante du semigramme [Müller *et al.*, 2004 ; Shalev-Shwartz *et al.*, 2004 ; Keshet *et al.*, 2007] ou du chromagramme [Ewert *et al.*, 2009 ; Otsuka *et al.*, 2009]. Une mesure de similarité entre un de ces vecteurs et le gabarit correspondant à un agrégat constitue alors le score associé à l'attaque de cet agrégat.

Rodet *et al.* [2004] proposent d'utiliser des descripteurs détectant la présence spécifique de sons de grosse caisse et de caisse claire. Pour chacune de ces deux classes de sons, une « fonction de détection » est calculée comme la corrélation entre le signal et un son-type représentatif de cette classe. Le score d'appariement est alors donné par la combinaison des fonctions de détection associées aux potentielles percussions de l'agrégat, combiné au score calculé d'après le spectrogramme.

Certains travaux [Müller *et al.*, 2004 ; Otsuka *et al.*, 2009] exploitent de plus la détection des transitoires dans le but de réduire le nombre de trames considérées par la couche haut niveau et ainsi limiter la complexité de l'alignement. Cela est effectué par une extraction des pics dans la fonction de détection de transitoires, qui constituent alors les positions potentielles des attaques d'agrégats.

2.2.3 Descripteur de tempo : le *tempogramme*

Les systèmes de Otsuka *et al.* [2009, 2011] extraient, en plus des descripteurs de hauteur de notes et de détection d'attaques, une autre représentation caractérisant le tempo courant de l'enregistrement. Cette représentation, appelée *tempogramme* est formée d'un vecteur par trame, dont chaque composante est associée à une valeur de tempo. Ce vecteur contient une mesure de « prédominance » du tempo concerné autour de la trame courante. Ces valeurs sont directement utilisées comme score d'appariement entre une trame et une valeur de tempo.

2.3 Modèles temporels (couche de haut niveau)

À partir des scores d'appariement locaux entre les couples d'éléments de la partition et de l'interprétation, la couche de haut niveau détermine l'alignement optimal de l'inter-

prétation dans son ensemble sur la partition. Cette section passe en revue les différents modèles temporels utilisés dans la littérature. Elles peuvent être séparées en deux classes : les méthodes d'alignement de séquences et les modèles probabilistes à états cachés.

2.3.1 Méthodes d'alignement de séquences

Séquences symboliques

Comme indiqué plus haut, de nombreux systèmes de suivi de partition en temps-réel exploitent des interprétations musicales sous forme symbolique. Ces travaux utilisent une stratégie similaire aux algorithmes d'alignement de chaines de caractères. Le premier système de Dannenberg [1984] cherche la plus longue sous-séquence de notes communes à la partition et à l'interprétation. Ce problème correspond à la maximisation du score d'alignement global, défini comme la somme des scores d'appariement locaux entre interprétation et partition. Pour éviter au système d'« avancer trop vite » dans la partition, une fonction de pénalité est ajoutée, sanctionnant les sauts de note. Cette méthode est utilisée dans la plupart des travaux d'alignement de musique symbolique, dans des versions modifiées, afin notamment de prendre en compte une synchronisation imparfaite entre les différentes voix (au sens où les notes censées être attaquées simultanément n'apparaissent pas rigoureusement en même temps) dans une musique polyphonique [Bloch et Dannenberg, 1985 ; Dannenberg et Mukaino, 1988 ; Puckette et Lippe, 1992 ; Hoshishiba *et al.*, 1996 ; Grubb et Dannenberg, 1994].

Pour tous ces modèles, l'alignement optimal peut être calculé grâce à des techniques de programmation dynamique [Bellman, 2003]. Les systèmes de suivi en temps réel ont de plus recours à des stratégies de seuillage qui limitent le nombre d'hypothèses considérées, afin de réduire la complexité de l'alignement. Ainsi, l'évènement de l'interprétation correspondant à une note future de la partition est souvent recherchée uniquement aux alentours d'une position extrapolée d'après l'alignement courant [Desain *et al.*, 1997 ; Heijink *et al.*, 2000].

La prise en compte des durées de notes constitue une autre variation par rapport aux algorithmes dédiés aux chaines de caractères. Vercoe et Puckette [1985] ; Arifi *et al.* [2005] utilisent des fonctions de cout explicitement associées à la dispersion des instants d'attaques par rapport à un tempo constant. Pour la comparaison de *séquences* (et non d'*instants*) de l'interprétation et de la partition, Baird *et al.* [1993] ; İzmirli *et al.* [2002] testent plusieurs hypothèses de tempo. L'alignement est alors effectué par un algorithme « glouton » : à chaque instant, le tempo conduisant à la plus grande similarité est sélectionné et la position courante dans la partition est augmentée de la valeur correspondante.

Lorsque l'interprétation n'est pas sous forme symbolique, il est tout de même possible d'employer un modèle temporel similaire, au moyen d'une segmentation effectuée par la couche bas-niveau [Müller *et al.*, 2004]. Dans ce système, l'extraction des pics d'une fonction de détection d'attaque sépare l'enregistrement en segments de durées de mêmes ordres que les agrégats, et dont le contenu (en terme de notes) est homogène. L'alignement est alors opéré en maximisant la somme des scores locaux associés aux attaques d'agrégats, de la même façon que précédemment. Néanmoins, le processus de segmentation peut lui-même être sujet à des erreurs, c'est pourquoi peu de travaux exploitent une telle stratégie.

Parmi les systèmes d'alignement de musique symbolique, celui proposé par Vantomme [1995] constitue un cas particulier. En effet, le système proposé se base principalement sur les instants des attaques de notes, confrontés aux prévisions données par un estimateur du tempo. L'information de hauteur des notes est exploitée uniquement si les motifs temporels ne correspondent pas.

Séquences de descripteurs

Dans le cas de l'alignement d'un enregistrement audio, un grand nombre de travaux font appel à l'algorithme *Dynamic Time Warping* (DTW) [Sakoe et Chiba, 1978], originellement dédié à au traitement de la parole. Cet algorithme est une méthode de programmation dynamique pour rechercher l'alignement optimal entre deux séquences de descripteurs, si le score global est défini comme la somme des mesures de similarité locale entre les couples de points appariés. Les scores locaux peuvent être pondérés en fonction de la déformation locale, afin de favoriser plus ou moins un alignement « diagonal », c'est-à-dire sans déformation temporelle.

Pour les systèmes ayant recours à une synthèse de la partition [Turetsky et Ellis, 2003 ; Dannenberg et Hu, 2003], la mise en œuvre de l'algorithme est immédiate. Dans les autres modèles, une séquence de descripteurs simulant une synthèse est construite. Pour cela, un descripteur-gabarit est associé à chaque agrégat. Les gabarits sont alors répliqués un nombre de fois correspondant à la durée de l'agrégat dans la partition. Les systèmes de Meron et Hirose [2001] ; Orio et Schwarz [2001] ; Hu *et al.* [2003] ; Soulez *et al.* [2003] ; Rodet *et al.* [2004] par exemple, utilisent cette stratégie.

De nombreuses variantes ont été apportées pour réduire la complexité de l'algorithme [Salvador et Chan, 2004 ; Kaprykowsky et Rodet, 2006], effectuer l'alignement en ligne [Dixon, 2005 ; Macrae et Dixon, 2010], chercher des alignements partiels au lieu d'un alignement global [Ewert *et al.*, 2011] ou autoriser des différences structurelles entre la partition et l'interprétation [Arzt *et al.*, 2008 ; Fremerey *et al.*, 2010]. Une autre modification est introduite par Arzt et Widmer [2010a] afin d'intégrer une dimension de tempo dans l'alignement temps-réel. Une estimation du tempo, calculée d'après l'alignement courant, est utilisée pour « étirer » la séquence représentant la partition en ajoutant ou supprimant des vecteurs.

2.3.2 Modèles probabilistes à états cachés

L'autre classe de couches de haut niveau couramment utilisées est formée de modèles statistiques à états cachés (qu'on appelle aussi variables latentes). Une variable aléatoire d'état cachée (dont la valeur n'est pas observée) est associée à chaque instant de l'interprétation, caractérisant la position dans la partition. Dans ces modèles, la mesure de similarité locale calculée par la couche de bas niveau est considérée comme la probabilité conditionnelle d'observer le descripteur, sachant l'état caché. Le modèle définit alors les lois de probabilités reliant les variables cachées entre elles. Par rapport aux approches d'alignement de séquences, l'utilisation d'un modèle probabiliste permet une plus grande souplesse dans la forme des contraintes temporelles introduites. De nombreux *a priori* peuvent être

exprimés de manière simple et même être estimés par des stratégies d'apprentissage. En revanche, cela se traduit souvent par un plus grand nombre de paramètres et une complexité supérieure à l'algorithme DTW.

Modèles de Markov cachés

La forme de modèle statistique la plus utilisée pour l'alignement est sans doute le Modèle de Markov Caché (MMC), qui sera présenté avec plus de détails au Chapitre 3. Dans ces modèles, chaque agrégat de note est représenté par un nombre d'états fixe [Cano *et al.*, 1999] ou dépendant de sa durée dans la partition [Raphael, 1999 ; Orio, 2002 ; Cont, 2006]. Ces modèles autorisent en général uniquement des transitions entre un agrégat et le suivant dans la partition. Néanmoins, d'autres transitions sont rendues possibles dans les modèles de Orio et Déchelle [2001] ; Schwarz *et al.* [2004] ; Montecchio et Orio [2008, 2009], vers des états « fantômes » représentant des erreurs dans l'interprétation.

Les systèmes de suivi de partition en temps réel associent à chaque instant de l'interprétation l'agrégat le plus probable, sachant les observations passées [Orio et Déchelle, 2001 ; Schwarz *et al.*, 2004 ; Pardo et Birmingham, 2005 ; Montecchio et Orio, 2008, 2009]. Raphael [1999] définit un autre critère pour le décodage du modèle, qu'il appelle *minimisation de l'erreur de segmentation*. Ces alignements peuvent être calculés efficacement par l'algorithme *forward* décrit par Rabiner [1989]. Afin de réduire davantage la complexité de l'alignement, Cont [2006] fait appel à une méthode de filtrage particulaire [Arulampalam *et al.*, 2002].

Le système de Cano *et al.* [1999] est présenté comme un MMC, qui effectue l'alignement en recherchant la séquence d'états cachés la plus probable. Cependant, ils modifient l'algorithme de Viterbi [Viterbi, 1967] et combinent aux probabilités de transitions une pénalité liée à la durée passée dans chaque état. De ce fait, ce système peut être vu comme un modèle semi-markovien caché [Yu, 2010], décodé par une méthode approchée.

Modèles à variable de tempo cachée

D'autres modèles statistiques ont été proposés afin de mieux rendre compte du rythme de la musique. Ainsi, une variable aléatoire cachée représentant le tempo courant est ajoutée. Cette variable de tempo peut être discrète [Grubb et Dannenberg, 1997 ; Otsuka *et al.*, 2011] ou continue [Raphael, 2006 ; Fox et Quinn, 2007 ; Cont, 2010].

Les options choisies pour le décodage du modèle sont diverses. Raphael [2006] utilise une méthode de programmation dynamique pour un décodage hors ligne de la séquence de variables cachées de plus grande probabilité. Fox et Quinn [2007] ; Otsuka *et al.* [2011] ; Montecchio et Cont [2011] ; Duan et Pardo [2011] font appel à des méthodes de filtrage particulaire pour estimer en temps réel le couple position/tempo le plus probable. Enfin, dans les modèles de Grubb et Dannenberg [1997] et Cont [2010], le décodage est adaptatif : un estimateur de tempo à l'instant précédent est utilisé pour mettre à jour la loi de probabilité instantanée de la position dans la partition. En retour, l'évolution de cette position est exploitée pour une estimation du tempo courant.

D'autres travaux ne sont pas présentés comme des modèles à état cachés, mais sont

équivalents en pratique. En effet, la fonction-objectif maximisée par Keshet *et al.* [2007] a la même forme que la probabilité *a posteriori* d'une séquence d'états cachés dans les modèles présentés plus haut. De même, le système présenté par Otsuka *et al.* [2009] peut être vu comme un modèle adaptatif, puisqu'à chaque instant, deux estimateurs de position et de tempo se mettent à jour mutuellement.

2.3.3 Points d'ancrages et passes multiples

Certains travaux proposent d'exploiter des *points d'ancrage*, c'est-à-dire des points dont l'alignement est considéré comme très fiable, afin d'améliorer la complexité ou la précision de l'alignement global. Dans le système de İzmirli et Zahler [2005] pour le suivi d'un soliste en temps réel, la couche de bas niveau compare le passé immédiat de l'interprétation à des séquences de la partition préalablement sélectionnées comme points d'ancrage. Lorsque la similarité dépasse un certain seuil, le système considère automatiquement que le point d'ancrage est détecté et supprime les autres hypothèses. La même méthode est proposée par Müller *et al.* [2004] pour un alignement hors ligne, afin de limiter la programmation dynamique aux intervalles entre points d'ancrage.

Dans la stratégie de Camarena-Ibarrola et Chávez [2010] chaque position peut être considérée comme un point d'ancrage. En effet, ils calculent pour chaque observation extraite de l'interprétation, les K positions de la partition les plus semblables. L'alignement est alors simplement effectué en associant la position la plus proche du point d'alignement précédent.

Dans le système de Niedermayer et Widmer [2010a,b], des points d'ancrages sont déduits d'un premier alignement par DTW, correspondant à des notes isolées dans la partition. Puis une seconde passe, pouvant exploiter une autre paramétrisation, est effectuée pour situer précisément chaque attaque de note, autour des points d'ancrage. Devaney *et al.* [2009] effectuent un deuxième alignement complet par MMC, après un premier alignement par DTW. Néanmoins, dans la seconde passe, la recherche est réduite aux alentours du premier chemin d'alignement.

2.4 Évaluation de l'alignement

Dans l'idéal, l'évaluation d'un système d'alignement musique-sur-partition doit prendre en compte l'application visée. En effet, les caractéristiques les plus importantes de l'alignement peuvent varier selon l'utilisation finale. Prenons l'exemple de la tâche de séparation de sources informée, dont le but est de reconstituer les signaux provenant des différents instruments présents sur un enregistrement sonore, avec l'aide de la partition. Dans ce cas, il est important que les agrégats associés aux trames du signal contiennent les notes effectivement présentes et jouées par les bons instruments. Il n'est par contre pas forcément nécessaire que la position détectée dans la partition soit la bonne. En effet, si la partition comporte plusieurs répétitions du même agrégat, la détection d'une mauvaise instance de cet agrégat ne nuit pas en général à la séparation de sources.

D'autre part, pour une application de visualisation de la partition en phase avec l'enregistrement, il est important que la position détectée dans la partition ne soit pas trop

éloignée de la position réelle. Notons qu'une grande précision n'est pas forcément indispensable, puisque le fragment de partition affiché doit rester assez large pour permettre un certain confort de lecture.

Les exemples précédents mettent en évidence deux visions possibles de l'alignement musique sur partition. Le premier point de vue consiste à le traiter comme une tâche de *classification* des trames audio en agrégats. Dans ce cadre, l'évaluation est effectuée trame à trame, en considérant le cout de chaque mauvaise classification. Dans la seconde perspective, on cherche à localiser dans l'enregistrement les évènements de la partition. Or, ces évènements recherchés sont les frontières entre les agrégats. La tâche d'alignement est alors vue comme un problème de *segmentation* de l'enregistrement en différents agrégats et l'évaluation mesure la précision temporelle de cette segmentation.

Les évaluations MIREX [MIR, 2010] (pour Music Information Retrieval Evaluation eXchange) constituent à notre connaissance la seule campagne d'évaluation objective d'alignement musique sur partition. La tâche évaluée est l'alignement *en ligne*, car l'application visée est le suivi d'une interprétation en temps réel en vue de l'interaction d'une machine avec les musiciens. La vision adoptée est alors le point de vue *segmentation*. Un certain nombre de mesures sont proposées par Cont *et al.* [2007] pour cette évaluation, dont certaines sont spécifiques aux systèmes temps-réel (comme la latence). Nous reprenons néanmoins les autres métriques, détaillées en section 2.4.2.

2.4.1 Métrique de classification

Pour mesurer la qualité d'un alignement du point de vue *classification*, nous calculons le cout moyen de la classification obtenue. Pour un enregistrement $\mathfrak{e}$ de longueur $N_\mathfrak{e}$ (en nombre de trames), nous notons $\mathbf{C}^\mathfrak{e}_{1:N_\mathfrak{e}} = C^\mathfrak{e}_1, \ldots, C^\mathfrak{e}_{N_e}$ la séquence des agrégats annotés (qui constitue la vérité-terrain). Soit alors $\hat{\mathbf{C}}^\mathfrak{e}_{1:N_\mathfrak{e}} = \hat{C}^\mathfrak{e}_1, \ldots, \hat{C}^\mathfrak{e}_{N_\mathfrak{e}}$ la séquence d'agrégats donnée par un alignement. Le *Cout de Classification* $\mathrm{CC}(\mathfrak{e})$ de cet alignement est défini par

$$\mathrm{CC}(\mathfrak{e}) = \frac{1}{N_\mathfrak{e}} \sum_{n=1}^{N_\mathfrak{e}} D(\hat{C}^\mathfrak{e}_n, C^\mathfrak{e}_n), \tag{2.1}$$

où D est une fonction de cout. Dans le cas général, il est possible d'utiliser une fonction non binaire pour mesurer les différences entre agrégat détecté et agrégat réel, comme proposé par Daniel *et al.* [2008] dans le cadre d'une tâche de transcription musicale. Néanmoins, dans le présent travail, nous nous contentons de la distance de Hamming, qui est égale à 1 si et seulement si les deux agrégats comparés sont différents. De ce fait, le cout de classification utilisé mesure simplement la proportion de trames qui sont incorrectement classées.

Le *Cout de Classification Moyen Pondéré* (CCMP) est la moyenne de cette mesure sur tout le corpus, pondérée par les longueurs des enregistrements :

$$\mathrm{CCMP} = \frac{\sum_\mathfrak{e} N_\mathfrak{e} \times \mathrm{CC}(\mathfrak{e})}{\sum_\mathfrak{e} N_\mathfrak{e}}. \tag{2.2}$$

Remarque

Pour mesurer le cout de classification d'un alignement, une annotation trame à trame de l'enregistrement est nécessaire. Or, une telle annotation est forcément imparfaite, en particulier à cause des ambiguïtés présentes aux extinctions de notes. En effet, contrairement aux attaques qui sont localisables avec une relative fiabilité, il est souvent très difficile de définir l'instant où une note disparait. Pour le corpus MAPS, les fichiers d'annotation sont les fichiers MIDI lus par le piano Disklavier. De ce fait, l'extinction annotée correspond à l'instant où la touche du piano cesse d'être enfoncée. La vibration des cordes est donc censée se terminer au même moment. En revanche, il est possible que la note puisse encore être entendue dans l'enregistrement, à cause de la réverbération de la salle. Dans le corpus RWC-pop, les annotations ont la forme d'une partition graphique, où les longueurs de notes indiquées ne correspondent pas toujours précisément à ce qui est effectivement joué. Par exemple, une note dont la valeur rythmique est une *noire* peut, si elle est jouée dans un caractère « piqué », durer moins d'une pulsation. De plus, il est fréquent dans la musique pop d'appliquer des effets de réverbération qui peuvent être différents selon les instruments. Pour ces raisons, les couts de classification calculés ne sont qu'indicatifs et leurs valeurs absolues ne sont pas totalement fiables.

2.4.2 Métriques de segmentation

Le second point de vue évalue la précision temporelle de la localisation des évènements de la partition, c'est-à-dire les frontières entre agrégats. Or, comme indiqué dans la remarque précédente, les frontières correspondant aux fins de notes sont très difficiles à définir. Les évènements à localiser sont donc les attaques de notes. Cette vision est en général favorisée dans les évaluations objectives, entre autre à cause des ambiguïtés inhérentes à une annotation trame à trame. Une autre raison est historique : en effet, les premiers systèmes d'alignements visaient le suivi de partition en temps réel. Comme indiqué plus haut, les mesures que nous utilisons dans cette thèse pour mesurer la précision de la segmentation sont issues du travail de Cont *et al.* [2007] pour les évaluations MIREX.

Taux d'alignement

Soient $\mathcal{A}(\mathfrak{e})$ l'ensemble des agrégats attaqués dans un enregistrement $\mathfrak{e}$. On rappelle que τ_c est l'index temporel du début d'un agrégat c. Soit $\hat{\tau}_c$ l'index temporel estimé par un système d'alignement. Le *taux d'alignement* $\mathrm{TA}(\mathfrak{e})$ est défini comme la proportion d'attaques d'agrégats localisés à l'intérieur d'une fenêtre de tolérance autour de la vérité-terrain :

$$\mathrm{TA}(\mathfrak{e}) = \frac{1}{\mathrm{Card}\big(\mathcal{A}(\mathfrak{e})\big)} \sum_{c \in \mathcal{A}(\mathfrak{e})} \mathbf{1}\big(|\hat{\tau}_c - \tau_c| \leq \theta\big) \tag{2.3}$$

où $\mathbf{1}$ représente la fonction indicatrice et θ est le seuil de tolérance choisi. Dans nos expériences, nous utiliserons trois valeurs de ce seuil (300 ms, 100 ms et 50 ms), pour des évaluations à différents niveaux de précision.

De même que précédemment, on définit le taux d'alignement moyen pondéré (TAMP) par

$$\mathrm{TAMP} = \frac{\sum_{\mathfrak{e}} \mathrm{Card}\big(\mathcal{A}(\mathfrak{e})\big) \times \mathrm{TA}(\mathfrak{e})}{\sum_{\mathfrak{e}} \mathrm{Card}\big(\mathcal{A}(\mathfrak{e})\big)}. \tag{2.4}$$

Imprécision

L'imprécision mesure la différence temporelle moyenne entre les attaques détectées et la vérité-terrain. Cependant, un biais peut intervenir lorsque l'alignement est « perdu », c'est-à-dire lorsque toute une série d'évènements d'un morceau est détectée « loin » de la vérité-terrain (par exemple à une répétition du même motif musical). Afin de limiter ce biais, l'imprécision est calculée uniquement pour les attaques détectées à moins de $\theta_d = 1$ s de la position réelle. On définit donc l'imprécision $\mathrm{I}(\mathfrak{e})$ d'un enregistrement $\mathfrak{e}$ par

$$\mathrm{I}(\mathfrak{e}) = \frac{\sum_{c \in \mathcal{A}(\mathfrak{e})} \mathbf{1}\big(|\hat{\tau}_c - \tau_c| \leq \theta_d\big) \times |\hat{\tau}_c - \tau_c|}{\sum_{c \in \mathcal{A}(\mathfrak{e})} \mathbf{1}\big(|\hat{\tau}_c - \tau_c| \leq \theta_d\big)}. \tag{2.5}$$

L'imprécision moyenne pondérée (IMP) est alors calculée de la même façon que précédemment.

2.4.3 Évaluation subjective

Une dernière forme d'évaluation subjective peut être menée, par l'écoute simultanée de l'enregistrement et d'une synthèse de la partition alignée. C'est pourquoi quelques exemples sonores ont été créés à partir de résultats d'alignement. Ils sont consultables sur la page internet http://perso.telecom-paristech.fr/~joder/CRF_alignment_examples/alignment_examples.html.

Dans ces exemples, un des deux canaux stéréophonique contient l'enregistrement original et le second contient la synthèse de la partition synchronisée. Pour un rendu indépendant du timbre des instruments présents sur l'enregistrement, la synthèse est effectuée en créant, à la position de chaque note détectée, une sinusoïde pure à la fréquence théorique de cette note (le diapason utilisé pour cette synthèse est à 440 Hz). Afin de mettre en valeur la précision des attaques et en raison de la faible fiabilité des longueurs de notes annotées, la durée des notes synthétisées est fixée à $\frac{1}{3}$ des longueurs reconnues.

Il est à noter que l'oreille est un évaluateur très sévère, concernant la précision d'un alignement. En effet d'après nos expériences personnelles, une différence temporelle de quelques dizaines de millisecondes entre une attaque réelle et sa détection peut laisser une impression très désagréable à l'écoute, surtout si la structure rythmique est modifiée. Notre « tolérance » aux imprécisions parait donc plus faible que celle nécessaire à certaines applications possibles, par exemple une visualisation de la partition graphique synchronisée avec l'enregistrement.

2.5 Bases de données

S'il devient de plus en plus facile de trouver des enregistrements musicaux et des partitions, notamment grâce aux bibliothèques consultables sur l'Internet, les bases de données annotées, c'est-à-dire comprenant les indications de synchronisation entre la partition et l'enregistrement, sont très rares. En effet, il est extrêmement fastidieux pour un humain de réaliser l'alignement audio-sur-partition, ou même de contrôler la validité d'un alignement automatique à un niveau de précision fin.

Or, pour évaluer les performances des systèmes automatiques, il est nécessaire de comparer les alignements avec une vérité-terrain fiable. Nous utilisons pour nos expériences deux corpus différents.

2.5.1 Corpus MAPS

La base de données MAPS (pour *MIDI Aligned Piano Sounds*), créée par Emiya *et al.* [2010], est un ensemble de sons de piano dédié à la transcription automatique du piano et l'estimation de fréquences fondamentales. Cette base de données contient, entre autres choses (notes isolées, agrégats aléatoires, accords usuels), des morceaux du répertoire classique enregistrés via un piano Disklavier Yamaha. Ce modèle possède un dispositif mécanique qui actionne les marteaux et les pédales du piano, permettant de « jouer » un fichier MIDI. Cela assure une synchronisation très précise entre l'enregistrement et la partition.

Nous utilisons 59 morceaux de cette base de données (environ 4h15 de musique), correspondant à deux conditions d'enregistrement différentes du Disklavier (micros rapprochés et éloignés). Cela constitue ce que nous appelons le corpus MAPS. Les fichiers MIDI joués par le dispositif comportent une piste de tempo correspondant à une interprétation expressive des pièces. Dans nos expériences, la vérité-terrain est fournie par ces fichiers MIDI. Les partitions à aligner proviennent aussi de ces mêmes fichiers. Cependant, le tempo est alors fixé à une valeur constante, de telle sorte que la longueur (en secondes) du morceau à ce tempo est la même que la durée de l'enregistrement. Cela correspond à une hypothèse raisonnable, dans le cas où les informations de tempo ne sont pas indiquées, ou ne sont pas accessibles de manière fiable dans la partition (par exemple si cette dernière est issue d'une reconnaissance optique de partition graphique). Dans la plupart des morceaux, les variations de tempo sont occasionnées uniquement par la libre interprétation des musiciens et le tempo s'éloigne peu de sa valeur moyenne. En revanche, certaines pièces contiennent plusieurs parties, où les tempos peuvent être radicalement différents, comme le premier mouvement de la sonate pour piano n°8 « pathétique » de Beethoven qui comporte une introduction très lente suivie d'une seconde partie *allegro*. Dans de tels cas, l'utilisation du tempo moyen peut mener à des *a priori* de durée très imprécis.

Cependant, nous verrons au chapitre 4 que parmi les trois modèles temporels proposés pour l'alignement, un seul (le modèle semi-markovien) exploite l'information de tempo de la partition, à travers la modélisation de la durée absolue de chaque agrégat. En effet, le second modèle exploite uniquement les informations de longueur en pulsations des agrégats, grâce à une modélisation d'un processus de tempo et le troisième modèle ne prend en compte

aucune indication de durée.

2.5.2 Corpus RWC-pop

Un autre corpus est utilisé, tiré de la sous-base de musique pop de la base de données RWC [Goto *et al.*, 2002]. Dans cet ensemble, 90 chansons (environ 6h de musique) sont intégrées à ce que nous appelons le corpus RWC-pop. Ces morceaux sont des enregistrements polyphoniques, multi-instrumentaux qui contiennent pour la plupart des percussions. L'annotation est constituée de fichiers MIDI fournis avec les enregistrements. Ces annotations sont le résultat d'une détection automatique des pulsations, qui a ensuite été corrigée à la main. Cependant, des inexactitudes peuvent subsister, notamment à des niveaux de précision très fins.

Dans ce corpus, comme dans presque toute la musique pop, le tempo des chansons est constant. Par conséquent, des changements de tempo sont introduits dans la partition MIDI à aligner, afin de simuler un tempo fluctuant de l'interprétation. Chaque fichier est séparé en segments de longueur égale en pulsations (environ 16 pulsations) et pour chaque segment, un unique tempo est tiré aléatoirement d'après une distribution uniforme entre 40 et 240 pulsations/s. Ces modifications représentent des changements de tempos extrêmes, qui peuvent occasionner des alignements relativement imprécis pour notre modèle exploitant les informations de durées absolues. Néanmoins, elles correspondent à un cas limite, et nous considérons les scores obtenus comme une borne inférieure des performances de ce modèle.

Dans notre scénario applicatif, les partitions proviennent de l'Internet. Or, les transcriptions en fichiers MIDI de morceaux pop que l'on peut trouver en ligne peuvent contenir des erreurs dans les parties de percussions, ou bien souvent ne contiennent pas de percussion du tout. De la même façon, les annotations des percussions du corpus RWC-pop sont de qualité variable. Nous choisissons donc de ne pas tenir compte des pistes de percussions dans les partitions à aligner.

2.5.3 Base d'apprentissage et base de test.

Nos expériences requièrent pour la plupart une estimation des paramètres des systèmes d'alignement. Pour effectuer celle-ci, nous utilisons une *base d'apprentissage*, constituée de 30 morceaux du corpus RWC-pop et de 20 morceaux de MAPS (environ un tiers de la base totale), choisis aléatoirement. Les évaluations des systèmes sont alors menées sur le reste des corpus.

Le tableau 2.1 récapitule un certain nombre d'informations concernant les bases de données utilisées dans cette thèse. Pour l'analyse des résultats, nous supposons que les détections des attaques et les classifications des trames sont effectuées de façon indépendantes. Cette approximation grossière nous permet d'estimer les intervalles de confiance théoriques à 95% pour les scores typiques obtenus. Cependant, les valeurs calculées pour le Cout de Classification Moyen Pondéré ne sont pas vraiment fiables, puisque les annotations comportent des imprécisions.

Base de données	Apprentissage		test	
	MAPS	RWC-pop	MAPS	RWC-pop
Nombre de morceaux	20	30	39	60
Durée totale	1h39	2h02	2h38	4h03
Nombre moyen de trames	14808	12222	12118	12143
Écart-type par morceau	9840	2283	7979	2043
Intervalle de confiance CCMP	—	—	0,14%	0,11%
Nombre d'évènements à détecter	26553	36299	48009	71905
Moyenne	1328	1210	1231	1198
Écart-type par morceau	964	373	1083	405
Intervalle de confiance TAMP	—	—	0,4%	0,3%

TABLE 2.1 – Statistiques des bases de données utilisées. Les intervalles de confiance sont les intervalles théoriques à 95% pour les scores typiques. Les mesures TAMP (Taux d'Alignement Moyen Pondéré) et CCMP (Cout de Classification Moyen Pondéré) sont définies en section 2.4.

2.6 Conclusion

Dans ce chapitre, nous avons étudié la composition et l'évaluation des systèmes d'alignement audio-sur-partition.

Après avoir détaillé la représentation de la partition nécessaire à cette tâche, nous avons vu qu'un système d'alignement pouvait être séparé en deux couches : la couche de bas niveau, calculant une mesure de similarité entre chaque position de l'enregistrement et de la partition, et la couche de haut niveau, qui incorpore certaines contraintes temporelles à ces scores locaux afin de réaliser l'alignement global de toute la séquence musicale. Un état de l'art du domaine a alors été réalisé sur la base de ces deux parties. Nous avons ensuite discuté des stratégies possibles d'évaluation d'un alignement, avant de présenter les bases de données utilisées dans ce travail.

Chapitre 3

Modèles graphiques pour l'alignement

Ce chapitre présente les outils théoriques qui permettent de réaliser l'alignement temporel d'un enregistrement musical et de sa partition par des modèles graphiques. Dans un premier temps, une courte introduction générale aux modèles graphiques est proposée. Puis nous présentons comment le problème d'alignement peut être traité comme un problème de décodage d'un *réseau bayésien dynamique*. Tous les modèles statistiques utilisés pour l'alignement musique-sur-partition font en effet partie de cette classe. À chaque instant de l'enregistrement, une variable aléatoire dite *cachée* représente l'agrégat joué. Les modèles de type réseaux bayésiens dynamiques supposent alors une loi régissant l'évolution de ce processus aléatoire ainsi que les distributions conditionnelles des observations, sachant l'agrégat joué. L'alignement est alors donné par le calcul de la séquence d'agrégats la plus probable, d'après le modèle.

Ces modèles sont dits *génératifs*, car ils supposent un processus particulier de génération des observations en fonction des agrégats. Or, puisque les observations sont données, il n'est pas nécessaire de considérer la façon dont elles ont été produites. Il est donc possible d'utiliser des modèles de *champs aléatoires conditionnels* (CRF), qui ne font aucune hypothèse concernant les observations, mais modélisent uniquement les probabilités conditionnelles des agrégats sachant la séquence d'observations (un tel modèle est dit *discriminatif*). Le cadre CRF permet de représenter des modèles équivalents (dans le sens où le décodage des variables cachées est identique) aux réseaux bayésiens dynamiques. Il présente de plus certains avantages sur ces derniers, en particulier la suppression d'une hypothèse d'indépendance conditionnelle des observations. L'exploitation de dépendances supplémentaires entre les variables est alors possible, sans augmentation notable de la complexité du décodage.

Enfin, nous nous intéressons aux lois régissant les durées des agrégats dans un modèle CRF. Nous présentons alors différentes structures utilisées pour une modélisation explicite de ces durées.

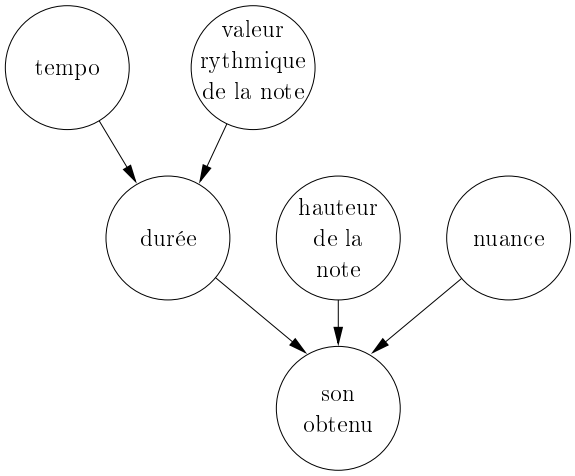


Figure 3.1 – Exemple d'un modèle graphique orienté, ou réseau bayésien, modélisant des paramètres de jeu d'une note de musique. On peut interpréter les liaisons entre variables (correspondant aux sommets du graphe) comme des relations de cause à effet. Une variable est en quelque sorte « créée » par ses parents. Formellement, cela se traduit par la définition de la probabilité conditionnelle de la variable sachant ses parents.

3.1 Modèles graphiques

3.1.1 Définition

Les *modèles graphiques* sont des modèles probabilistes dont la structure de dépendance entre les différentes variables aléatoires est représentée par un graphe. Ces représentations permettent une visualisation rapide et intuitive des dépendances, ce qui peut être précieux pour des modèles faisant appel à un grand nombre de variables. Ces modèles sont utilisés lorsque la distribution de probabilité jointe de toutes les variables du modèle peut être factorisée en un produit de termes faisant apparaitre chacun un petit nombre de variables.

Un modèle graphique est donc formé à partir d'un graphe $\mathcal{G}$. Chaque sommet du graphe est identifié à une variable aléatoire du modèle probabiliste et une arête représente une "dépendance directe" entre les deux variables qu'elle relie. La nature de cette dépendance dépend du type de modèle graphique considéré. En effet, on sépare les modèles graphiques en deux classes : les *réseaux bayésiens*, ou modèles graphiques orientés, qui sont formés à partir d'un graphe orienté acyclique, et les *champs de Markov*, ou modèles graphiques non orientés, qui utilisent un graphe non orienté. Dans la suite de cette section, nous définissons de façon rapide ces deux types de modèles graphiques. Pour une introduction aux modèles graphiques plus complète, voir par exemple Koller et Friedman [2009].

3.1.2 Réseaux bayésiens

Comme expliqué plus haut, le graphe associé à un modèle du type réseau bayésien est un graphe orienté acyclique qui exprime la donnée d'une distribution de probabilité

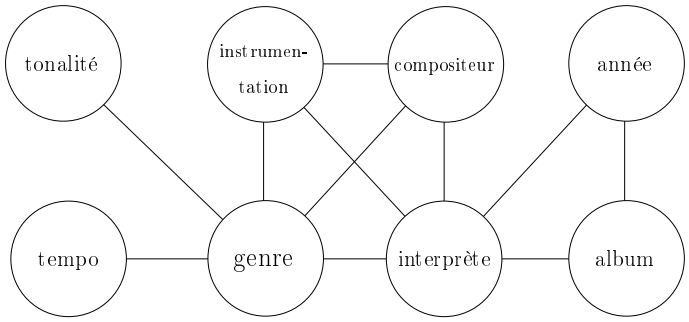


FIGURE 3.2 – Exemple d'un modèle graphique non orienté, ou champ de Markov modélisant les relations entre certaines méta-données d'un enregistrement musical. Dans un tel modèle, l'interprétation des arêtes du graphes est moins directe. Elles expriment une relation de dépendance entre deux variables, qui s'influencent mutuellement.

conditionnelle de chaque variable aléatoire (identifiée au sommet correspondant) sachant ses parents. Intuitivement, on peut considérer qu'une variable est « causée » par ses parents. La figure 3.1 montre un exemple de réseau bayésien qui modélise la production d'une note de musique. Plus formellement, pour une variable aléatoire X du modèle, soit $\mathrm{Pa}(X)$ l'ensemble (potentiellement vide) des parents de cette variable (au sens du graphe orienté). Un réseau bayésien permet de factoriser la probabilité jointe de l'ensemble $\mathcal{X}$ des variables du modèle sous la forme

$$P(\mathcal{X}) = \prod_{X \in \mathcal{X}} p\left(X | \mathrm{Pa}(X)\right). \tag{3.1}$$

Un tel modèle est donc défini par toutes les probabilités conditionnelles du type $p\left(X | \mathrm{Pa}(X)\right)$.

3.1.3 Champs de Markov

Un champs de Markov est un modèle graphique formé d'après un graphe non orienté. Un exemple est représenté sur la figure 3.2, où sont modélisés des paramètres ayant trait au genre musical d'un morceau. Les arêtes représentent ici l'existence d'une dépendance conditionnelle. En effet dans un tel modèle, deux variables aléatoires non voisines sont conditionnellement indépendantes sachant toutes les autres variables. Ou de façon équivalente, une variable aléatoire est conditionnellement indépendante de toutes les autres variables sachant ses voisins.

Sous certaines conditions (qui dans notre cas seront vérifiées), la distribution de probabilité jointe sur l'ensemble des variables peut être factorisée selon les *cliques* du graphe. Une clique d'un graphe est un sous-graphe complet de ce graphe, c'est-à-dire un ensemble de sommets deux à deux adjacents. Soit Ξ l'ensemble des cliques du graphe. La factorisation est de la forme :

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{\xi \in \Xi} \phi_\xi(\xi). \tag{3.2}$$

Les ϕ_ξ sont des fonctions à valeurs positives appelées *potentiels*. Ces potentiels ne sont pas toujours interprétables comme des probabilités conditionnelles, car ils ne satisfont pas

forcément l'axiome des probabilités. En revanche, cette possible hétérogénéité des potentiels locaux permet aux champs de Markov de modéliser les dépendances entre variables de manière plus souple que les réseaux bayésiens. Le facteur de normalisation Z assure alors que les valeurs calculées satisfont l'axiome des probabilités.

L'intérêt des modèles graphiques est la possibilité de « transmettre des informations » localement entre les différentes variables aléatoires, en suivant les liens du graphe. En effet la donnée d'une information sur une des variables permet de déduire certaines contraintes concernant les variables voisines. Prenons l'exemple représenté figure 3.2. Dans ce cas, si on observe un tempo lent, cela influence notre connaissance du genre musical (par exemple le genre *ballade* devient beaucoup plus probable que le genre *techno*). L'information se propage alors aux nœuds voisins (par exemple, les instrumentations comprenant du chant sont favorisées). Un modèle graphique permet donc, à partir de l'observation de certaines variables, de déduire des connaissances sur les autres variables, avec une complexité limitée puisque la transmission de l'information s'effectue de proche en proche en considérant uniquement les voisins de chaque variable.

3.2 Alignement temporel par réseaux bayésiens dynamiques (RBD)

Des deux classes de modèles graphiques présentées précédemment, la plus employée pour le traitement du signal audio est la classe des réseaux bayésiens. Cela s'explique par l'utilisation de probabilités conditionnelles, qui permettent une interprétation intuitive des modèles. Dans le domaine de l'alignement musique-sur-partition, on fait appel à une catégorie particulière de réseaux bayésiens : les réseaux bayésiens dynamiques (RBD).

3.2.1 Réseaux bayésiens dynamiques : définition

Un réseau bayésien dynamique [Murphy, 2002] est un réseau bayésien représentant les différentes variables aléatoires d'un processus temporel discret. Le modèle peut alors être découpé en « tranches temporelles », dont chacune est associée à un instant du processus (dans notre cas, une trame de l'enregistrement musical). Ces « tranches » présentent souvent une structure répétitive, ce qui traduit l'hypothèse que les lois régissant l'évolution du processus ne dépendent pas du temps. C'est le cas par exemple dans les systèmes dynamiques.

Les RBD les plus largement utilisés sont probablement les Modèles de Markov Cachés (MMC) [Rabiner, 1989], qui sont très employés notamment pour le traitement de la parole. Dans un tel modèle, la séquence de variables observées (par exemple une suite de descripteurs extraits du signal audio) est supposée être produite par un processus markovien caché, pouvant prendre un nombre fini de valeurs. Chaque observation est supposée dépendre uniquement de la variable correspondant à la valeur du processus non observé au même instant.

La structure répétitive d'un MMC est donc composée de deux variables aléatoires représentant respectivement la variable cachée et la variable observée à l'instant considéré.

Le processus caché étant markovien, le seul parent d'une variable cachée est la variable cachée précédente.

Un exemple de MMC est représenté sur la figure 3.3, pour la modélisation d'un enregistrement monophonique. L'enregistrement est transformé en une séquence de valeurs de descripteurs acoustiques (appelés observations), caractérisant chacune le contenu du signal sur une fenêtre temporelle, en général courte. Dans cet exemple, le descripteur utilisé est la *puissance spectrale*, obtenue par une transformée de Fourier à court terme. Soit N la longueur de cette séquence et $\mathbf{Y}_{1:N} = Y_1, \dots Y_N$ la suite des variables aléatoires associées à ces observations. Soit $\mathbf{X}_{1:N} = X_1, \dots X_N$ la séquence de variables aléatoires représentant les notes jouées aux instants correspondants. La modélisation par un MMC repose sur deux hypothèses :

1. Chaque observation dépend uniquement de la note jouée à cet instant ;
2. À chaque instant, la note jouée est indépendante de tout le passé, sachant la note précédente (le processus est Markovien d'ordre 1).

De ce fait, on peut écrire pour tout n :

$$P(Y_n|\mathbf{X}_{1:N}, \mathbf{Y}_{1:N}) = P(Y_n|X_n) \; ; \tag{3.3}$$

$$P(X_n|\mathbf{X}_{1:n}, \mathbf{Y}_{1:n}) = P(X_n|X_{n-1}), \tag{3.4}$$

ce qui correspond au graphe orienté représenté figure 3.3. La distribution de probabilité se factorise alors comme suit :

$$P(\mathbf{X}_{1:N}, \mathbf{Y}_{1:N}) = P(X_1)P(Y_1|X_1) \prod_{n=2}^{N} P(X_n|X_{n-1})P(Y_n|X_n). \tag{3.5}$$

Une telle factorisation permet l'utilisation de méthodes de programmation dynamique [Rabiner, 1989] pour réaliser plusieurs opérations de calcul de probabilités ou d'estimation de paramètres avec une complexité limitée. En particulier le *décodage* du modèle, c'est-à-dire l'estimation des variables cachées correspondant le mieux (au sens des probabilités conditionnelles) à une séquence d'observations, peut être effectué grâce à l'algorithme dit de Viterbi.

3.2.2 Alignement par réseau bayésien dynamique

Construction du modèle d'après la partition

Les modèles statistiques utilisés dans les systèmes d'alignement existants sont tous des réseaux bayésiens dynamiques. Comme dans l'exemple de la figure 3.3, ils modélisent les observations extraites de l'enregistrement comme les réalisations de variables aléatoires « engendrées » par une séquence de variables cachées (non observées dans le signal). Ces variables cachées correspondent à des informations interprétables, desquelles peuvent être déduites les positions dans la partition correspondant aux observations.

Par la suite, nous noterons X_n la variable aléatoire correspondant à l'*étiquette* assignée à la trame n de l'enregistrement, c'est-à-dire l'ensemble des variables cachées associées à cette trame.

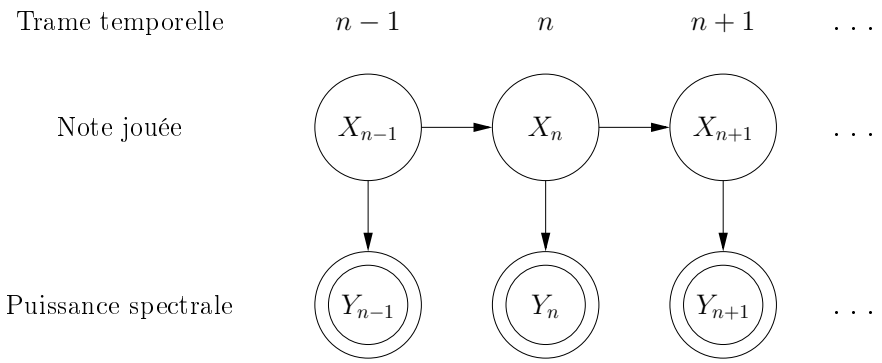


Figure 3.3 – Représentation graphique d'un modèle de Markov caché modélisant une mélodie monophonique. Les variables observées sont représentées en traits doubles.

Un modèle simple pour l'alignement musique-sur-partition est obtenu grâce à un MMC semblable à celui représenté sur la figure 3.3 mais dont les variables cachées correspondent aux agrégats joués. Soit $\mathbf{C}_{1:N} = C_1, \ldots, C_N$ la séquence de variables aléatoires représentant les agrégats joués aux trames 1 à N de l'enregistrement. Dans ce cas simple, on a pour tout n, $X_n = C_n$.

On retrouve la structure en deux couches exposée en 2.1.3. En effet, la couche bas niveau correspond au modèle de production des observations, c'est-à-dire à la donnée des probabilités conditionnelles $P(Y_n|X_n)$. Le modèle temporel est alors défini par la distribution de probabilité de l'agrégat initial $P(X_1)$ et par les probabilités de transition $P(X_{n+1}|X_n)$.

On peut représenter ces possibilités de transitions entre agrégats grâce à un automate, c'est-à-dire un graphe orienté, dont les états (*i.e.* les sommets) correspondent aux agrégats de la partition. Les arêtes représentent alors les transitions possibles. Un tel automate est très facilement construit d'après la partition. Un exemple correspondant à la partition de la figure 2.1 est représenté sur la figure 3.4, dans le cas où l'enregistrement est supposé sans erreur. Dans cet exemple simple, les probabilités de transitions sont supposées constantes (c'est-à-dire indépendante de la trame n). On les note alors $p_{i,j} = P(C_{n+1} = j|C_n = i)$.

Critère de décodage

Un réseau bayésien dynamique permet donc de relier par un modèle probabiliste la partition et l'enregistrement. Il est à noter que ce modèle est spécifique à une partition, puisque les variables cachées sont directement déduites de cette partition. Une fois le modèle donné, le chemin d'alignement est défini comme la séquence de variables cachées optimale, expliquant le mieux les observations extraites de l'enregistrement. Mais pour cela, il faut encore définir ce qu'on entend par la séquence « optimale ». En effet, plusieurs critères d'optimalité peuvent être imaginés. On peut par exemple choisir le critère du *maximum de vraisemblance*, qui définit la séquence optimale $\hat{\mathbf{x}}^{\mathrm{MV}}$ par :

$$\hat{\mathbf{x}}^{\mathrm{MV}}_{1:N} = \arg\max_{\mathbf{x}_{1:N}} P(\mathbf{Y}_{1:N}|\mathbf{X}_{1:N} = \mathbf{x}_{1:N}). \tag{3.6}$$

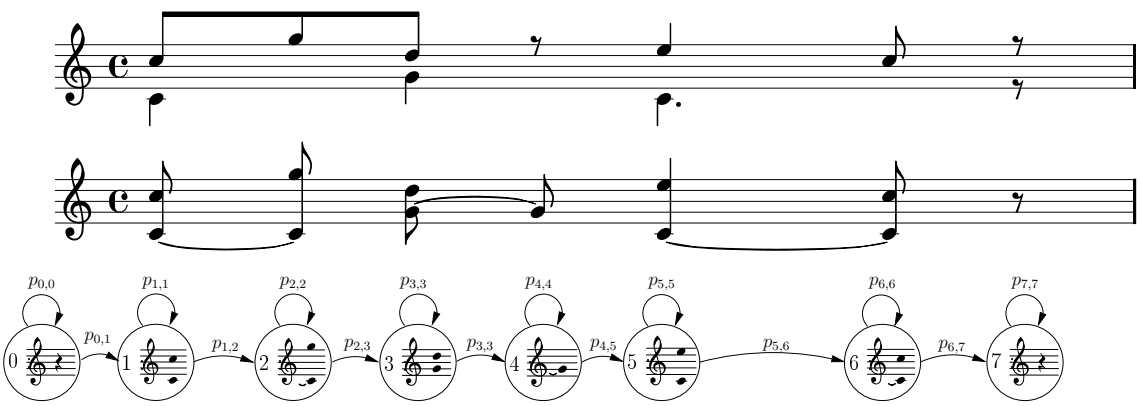


FIGURE 3.4 – Exemple d'automate des agrégats associé à une partition. Haut : partition graphique originale. Milieu : représentation en séquence d'agrégats. Bas : automate des agrégats. Les flèches représentent les transitions possibles entre états de l'automate. Le paramètre $p_{i,j}$ correspond à la probabilité de la transition de l'état i à l'état j.

D'après l'équation (3.3), l'étiquette optimale $\hat{x}_n^{\mathrm{MV}}$ pour chaque trame n est alors donnée par

$$\hat{x}_n^{\mathrm{MV}} = \arg\max_{x_n} P(Y_n | X_n = x_n). \tag{3.7}$$

Le critère du maximum de vraisemblance cherche donc à maximiser les probabilités conditionnelles des observations. Ainsi, il tient compte uniquement de ce que nous nommons la couche de bas niveau, puisque les probabilités des transitions ne sont pas considérées.

Il est alors souvent plus pertinent d'utiliser le critère du *maximum a posteriori* local, qui associe à chaque trame l'état caché le plus probable, sachant la séquence d'observations $\mathbf{Y}$. On définit l'état caché optimal $\hat{x}_n^{\gamma}$ à la trame n comme

$$\hat{x}_n^{\gamma} = \arg\max_{x} P(X_n = x | \mathbf{Y}_{1:N}). \tag{3.8}$$

Pour les systèmes en ligne, notamment les systèmes conçus pour le suivi de partition en temps réel, il n'est pas possible d'exploiter ce critère car l'état caché à une trame n doit être estimé alors que la séquence d'observations n'est pas connue entièrement. Une variante de ce critère est alors employée, utilisant uniquement les observations des trames précédentes. L'estimateur $\hat{x}_n^{\alpha}$ de l'état caché à la trame n est alors :

$$\hat{x}_n^{\alpha} = \arg\max_{x} P(X_n = x | \mathbf{Y}_{1:n}). \tag{3.9}$$

Il est à noter que ce critère n'est pas équivalent à celui de l'équation (3.8). L'alignement obtenu n'est alors pas le résultat optimal que l'on aurait une fois toute la séquence connue. Les lettres α et γ sont employées en référence aux notations classiques des probabilités *a posteriori* des états cachés dans un MMC. Ces probabilités, ainsi que le chemin d'alignement optimal, peuvent être calculés grâce à des techniques de programmation dynamique, comme décrit par Rabiner [1989].

Les deux critères des équations (3.8) et (3.9) prennent en compte les probabilités des variables cachées sur chaque trame. Cela peut rendre le décodage plus robuste, car la marginalisation effectuée sur les autres variables cachées amoindrit les effets de potentielles erreurs commises sur des trames éloignées. En revanche, il n'est pas garanti que le chemin d'alignement obtenu soit « possible » (c'est-à-dire de probabilité non nulle). En effet, il est possible que les états cachés sélectionnés en deux trames consécutives soient associés à une probabilité de transition nulle.

Dans le cas hors ligne, un autre critère possible est le *maximum a posteriori* (MAP), qui considère la probabilité de la séquence entière d'état cachés. Par construction, la séquence décodée est alors forcément de probabilité non nulle. La séquence optimale $\hat{\mathbf{x}}^{\text{MAP}}$ est donc définie par :

$$\hat{\mathbf{x}}_{1:N}^{\text{MAP}} = \arg\max_{\mathbf{x}_{1:N}} P(\mathbf{X}_{1:N} = \mathbf{x}_{1:N} | \mathbf{Y}_{1:N}). \tag{3.10}$$

C'est cette définition du chemin d'alignement optimal que nous utiliserons.

Il est à noter que d'autre critères de décodage peuvent être envisagés, en rapport avec l'application visée. Par exemple, le décodage utilisé par Raphael [1999] cherche le chemin qui minimise l'espérance mathématique du nombre d'erreurs de segmentation, c'est-à-dire le nombre de transitions entre notes détectées à plus d'une durée θ de sa position réelle. Cependant, comme avec les critères de *maximum a posteriori* local, un tel décodage ne garantit pas que le chemin d'alignement obtenu sera « possible ».

3.3 Champs aléatoires conditionnels (CRF)

Les réseaux bayésiens couramment utilisés, en particulier les MMC, sont des modèles dits *génératifs*, qui modélisent les distributions des observations conditionnellement aux variables cachées $P(Y_n|X_n)$. Or cette modélisation n'est pas forcément nécessaire pour l'application d'alignement. En effet, les critères de décodage utilisés ont la forme d'une probabilité (ou d'une espérance) conditionnée par les observations, de la forme $P(\mathbf{X}|\mathbf{Y})$. C'est pourquoi on peut aussi envisager d'autres modèles, dits *discriminatifs*, qui modélisent directement la probabilité des variables cachées sachant les observations. Nous nous intéressons donc aux champs aléatoires conditionnels (CRF pour l'anglais *Conditional Random Fields*), qui sont une classe de modèles graphiques discriminatifs introduits par Lafferty *et al.* [2001] pour la segmentation et l'étiquetage de données séquentielles.

3.3.1 Définition

Un CRF est un cas particulier de modèle graphique non orienté, contenant deux types de variables, dites *cachées* et *observées*. Contrairement aux champs de Markov classiques, qui modélisent la probabilité conjointe de toutes les variables du modèle, les CRF modélisent uniquement la probabilité conditionnelle des variables cachées, conditionnellement aux variables observées. Un exemple de représentation graphique d'un CRF simple est présenté figure 3.5 (droite).

Un CRF peut être défini sur un graphe non orienté quelconque. Cependant, nous nous limitons ici au cas qui nous intéresse, où les variables peuvent être organisées en « tranches

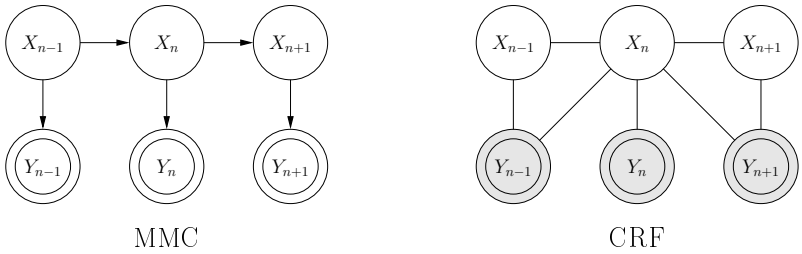


FIGURE 3.5 – Comparaison des représentations graphiques d'un modèle de Markov caché (MMC) et d'un champs aléatoire conditionnel (CRF). Les sommets doubles représentent les variables observées et les sommets grisés correspondent aux variables de conditionnement du modèle CRF.

temporelles », de la même façon que dans un réseau bayésien dynamique. Les modèles de ce type sont appelés champs aléatoires conditionnels dynamiques par Sutton *et al.* [2004]. Pour des raisons de complexité, les modèles utilisés en pratique se restreignent au cas où les variables cachées forment une chaine de Markov, conditionnellement aux observations. Sur la représentation graphique, cela se traduit par le fait qu'aucune arête ne relie deux variables cachées associées à des trames temporelles non successives.

Dans ce cas, la probabilité d'une séquence d'étiquettes $\mathbf{X}$ sachant la séquence d'observations $\mathbf{Y}$ est donnée par

$$P(\mathbf{X}|\mathbf{Y}) = \frac{1}{Z(\mathbf{Y})}\phi(X_1, \mathbf{Y}, 1)\prod_{n=2}^{N}\psi(X_n, X_{n-1}, \mathbf{Y}, n)\phi(X_n, \mathbf{Y}, n), \tag{3.11}$$

où $Z(\mathbf{Y})$ est un facteur de normalisation, qui assure que la valeur calculée est bien une probabilité. La fonction-potentiel ψ, que nous appellerons *fonction de transition*, contrôle les transitions entre les étiquettes (dépendant éventuellement des observations) et la *fonction d'observation* ϕ fait le lien entre les observations et les étiquettes.

La dépendance de la fonction d'observation ϕ à la trame n est nécessaire pour « sélectionner » la position correspondante dans la séquence d'observation $\mathbf{Y}$. Mais elle permet en outre d'exprimer des contraintes générales sur la relation entre les positions dans l'enregistrement et dans la partition. Par exemple, si l'on suppose que l'enregistrement commence forcément au début de la partition (avec une étiquette 0), la fonction d'observation utilisée pour la première trame peut être définie comme : $\phi(X_1, \mathbf{Y}, 1) = \mathbf{1}_{\{0\}}(X_1)$, où $\mathbf{1}_{\{0\}}$ représente la fonction indicatrice du singleton $\{0\}$. Une dépendance de la fonction de transition ψ à la trame, autre que pour la localisation de l'observation courante dans la séquence, est par contre plus difficilement interprétable. C'est pourquoi dans les modèles courants, la forme de cette fonction est uniforme sur toute les trames temporelles. Dans la suite, pour des raisons de clarté, les notations ne feront pas intervenir la trame temporelle dans les fonctions-potentiels, lorsque cela n'engendre pas de confusion. Nous écrirons donc par exemple $\phi(X_1, \mathbf{Y})$ pour $\phi(X_1, \mathbf{Y}, 1)$.

Le facteur de normalisation $Z(\mathbf{Y})$ de l'équation (3.11) peut se factoriser sous la forme

suivante :

$$\begin{aligned} Z(\mathbf{Y}) &= \sum_{\mathbf{X}\in\boldsymbol{\mathcal{X}}} \phi(X_1,\mathbf{Y}) \prod_{n=2}^{N} \psi(X_n, X_{n-1}, \mathbf{Y})\phi(X_n,\mathbf{Y}) \qquad (3.12)\\ &= \sum_{X_1} \phi(X_1,\mathbf{Y}) \sum_{X_2} \psi(X_2,X_1,\mathbf{Y})\phi(X_2,\mathbf{Y}) \sum_{X_3} \dots \sum_{X_N} \psi(X_N,X_{N-1},\mathbf{Y})\phi(X_N,\mathbf{Y}) \end{aligned}$$

où $\boldsymbol{\mathcal{X}}$ représente l'ensemble des séquences d'étiquettes possibles. On peut introduire des matrices $\Psi_n(\mathbf{Y})$ dont chaque élément d'indices (i,j) s'écrit :

$$\forall n \in \{2,\dots,N\}, \quad [\Psi_n(\mathbf{Y})]_{i,j} = \{\psi(x_n, x_{n-1}, \mathbf{Y})\phi(x_n,\mathbf{Y})\}\,\big|_{(x_{n-1},x_n)=(i,j)}. \qquad (3.13)$$

En utilisant ces notations, on peut écrire :

$$Z(\mathbf{Y}) = \Phi_1^{\mathrm{T}} \left(\prod_{n=2}^{N} \Psi_n(\mathbf{Y}) \right) \mathbb{1} \qquad (3.14)$$

où Φ_1 est le vecteur-colonne contenant les valeurs de $\phi(X_1|\mathbf{Y})$ et $\mathbb{1}$ est un vecteur dont toutes les composantes sont égales à 1. Le calcul de $Z(\mathbf{Y})$ peut donc s'effectuer comme un produit matriciel.

Malgré les différences en terme de modélisation par rapport aux réseaux bayésiens dynamiques, les mêmes techniques de programmation dynamique peuvent être utilisées pour le décodage ou l'inférence, en particulier l'algorithme de Viterbi [Rabiner, 1989], comme nous le verrons à la section 4.3. Un alignement employant un CRF n'est donc pas plus couteux qu'avec un modèle RBD, si la structure de dépendance des variables cachées est la même.

Pour de plus amples informations sur les CRF, le lecteur est invité à consulter Lafferty *et al.* [2001], Wallach [2004] ou encore Sutton et McCallum [2011].

3.3.2 Les CRF comme généralisation des RBD pour l'alignement

Il est important de noter que pour notre application d'alignement, un réseaux bayésien dynamique peut toujours être considéré comme un cas particulier de CRF. Illustrons cette propriété grâce à l'exemple d'un modèle de Markov caché. Pour un MMC $\mathcal{H}$ donné, on considère le graphe non orienté possédant les mêmes arêtes que le graphe du MMC[1]. On définit alors les potentiels comme suit :

$$\begin{aligned} \psi(X_n, X_{n-1}, \mathbf{Y}) &= \psi(X_n, X_{n-1}) &&= p_{\mathcal{H}}(X_n|X_{n-1}) \qquad (3.15)\\ \phi(X_n,\mathbf{Y}) &= \phi(X_n, Y_n) &&= p_{\mathcal{H}}(Y_n|X_n) \qquad (3.16) \end{aligned}$$

où $p_{\mathcal{H}}$ représente la (densité de) probabilité correspondant au MMC. La probabilité (au sens du modèle CRF ainsi obtenu) d'une séquence d'étiquettes sachant les observations

1. Pour un réseau bayésien quelconque, le graphe non orienté correspondant est obtenu par *moralisation* du graphe orienté. La moralisation consiste à relier tous les parents d'un même nœud, puis de désorienter toutes les arêtes.

s'écrit alors, d'après l'équation (3.11),

$$p(\mathbf{X}|\mathbf{Y}) = \frac{1}{Z(\mathbf{Y})}\phi(X_1, Y_1)\prod_{n=2}^{N}\psi(X_n, X_{n-1})\phi(X_n, Y_n) \tag{3.17}$$

$$= \frac{1}{Z(\mathbf{Y})}p_{\mathcal{H}}(Y_1, X_1)\prod_{n=2}^{N}p_{\mathcal{H}}(X_n|X_{n-1})p_{\mathcal{H}}(Y_n|X_n) \tag{3.18}$$

$$= \frac{1}{Z(\mathbf{Y})}p_{\mathcal{H}}(\mathbf{Y}, \mathbf{X}). \tag{3.19}$$

On rappelle que $\boldsymbol{\mathcal{X}}$ désigne l'ensemble des séquences d'étiquettes possibles. On a alors

$$Z(\mathbf{Y}) = \sum_{\mathbf{X}\in\boldsymbol{\mathcal{X}}}\phi(X_1, Y_1)\prod_{n=2}^{N}\psi(X_n, X_{n-1}), \phi(X_n, Y_n) \tag{3.20}$$

$$= \sum_{\mathbf{X}\in\boldsymbol{\mathcal{X}}}p_{\mathcal{H}}(\mathbf{Y}, \mathbf{X}) \tag{3.21}$$

$$= p_{\mathcal{H}}(\mathbf{Y}). \tag{3.22}$$

Il s'ensuit

$$p(\mathbf{X}|\mathbf{Y}) = \frac{p_{\mathcal{H}}(\mathbf{Y}, \mathbf{X})}{p_{\mathcal{H}}(\mathbf{Y})} \tag{3.23}$$

$$= p_{\mathcal{H}}(\mathbf{X}|\mathbf{Y}). \tag{3.24}$$

Les deux modèles sont alors équivalents du point de vue du décodage. Les CRF forment donc une classe de modèles plus large que les réseaux bayésiens dynamiques pour l'alignement.

3.3.3 Propriétés des CRF et avantages sur les RBD pour l'alignement

Les CRF présentent deux avantages principaux sur les RBD classiquement utilisés pour l'alignement. Nous nous concentrons ici sur le *décodage* du modèle et laissons de côté les questions relatives à l'*apprentissage* des paramètres des modèles. De fait, dans la plupart des systèmes exploités pour l'alignement de musique sur partition, ces paramètres sont déterminés de façon heuristique, faute de bases d'apprentissage suffisantes.

Modèle discriminatif

Le premier, dû au caractère *discriminatif* des CRF, est la suppression de l'hypothèse d'indépendance conditionnelle des observations. En effet, dans de nombreux modèles génératifs (en particuliers les MMC), une observation Y_n est supposée indépendante de toutes les autres variables du modèle, conditionnellement à son étiquette associée X_n. Dans ce cas, aucun lien ne peut être modélisé entre cette observation et une autre étiquette $X_{n'}$. Une certaine forme de dépendance peut être prise en compte grâce à des modèles de type *modèle de Markov caché autorégressifs* [Murphy, 2002], où l'observation Y_n peut dépendre des

variables observées précédentes $Y_{n-1}, Y_{n-2}, \ldots$. De ce fait, une dépendance implicite est présente entre l'étiquette X_n et ces observations passées. Néanmoins, ces dépendances sont toujours orientées « vers le futur », puisque le modèle représente le processus de génération des observations.

Dans un CRF, au contraire, aucune hypothèse n'est faite quant à la façon dont les observations sont produites, puisqu'il modélise uniquement les probabilités conditionnellement à ces variables. Un tel modèle permet donc la prise en compte de dépendances explicites entre n'importe quelle étiquette et n'importe quelle(s) observation(s) située dans le passé ou le futur (et potentiellement la séquence entière).

Modèle non orienté

Le deuxième avantage principal des CRF vient de l'utilisation d'un modèle graphique non orienté. Dans un tel modèle, les potentiels définis en (3.11) ne correspondent pas forcément à des probabilités conditionnelles. Cela permet donc l'utilisation de fonctions-potentiels plus générales, dont la somme sur chaque clique n'est pas égale à 1.

L'intérêt d'utiliser des potentiels non normalisés est d'éviter le problème appelé *label bias* par Lafferty *et al.* [2001]. Ce problème est causé par la normalisation locale des probabilités de transition des modèles discriminatifs lorsque les probabilités des transitions « de sortie » des différentes étiquettes sont hétérogènes. Une illustration de ce problème est représentée figure 3.6, où sont représentés un modèle CRF (non orienté) et le réseau bayésien discriminatif obtenu en normalisant les potentiels. Un tel modèle est appelé *modèle de Markov d'entropie maximale* McCallum *et al.* [2000]. Il est important de noter que dans ce dernier modèle, les deux potentiels de transition et d'observation sont normalisés conjointement pour former des probabilités conditionnelles de transition, sachant les observations. Ces probabilités sont alors définies par :

$$P_{\mathcal{M}}(X_n|X_{n-1}, Y_n) = \frac{\psi(X_n, X_{n-1})\phi(X_n, Y_n)}{\sum_{x_n} \psi(x_n, X_{n-1})\phi(x_n, Y_n)}. \tag{3.25}$$

Dans notre exemple, cette normalisation locale des deux potentiels fait apparaitre un biais en faveur de l'étiquette 1. En effet, entre les trames 2 et 3, la probabilité conditionnelle de la transition $1 \to 1$ ne tient pas compte du potentiel d'observation puisque cette transition est la seule possible (sa probabilité est donc toujours égale à 1). De plus, cette probabilité devient alors plus élevée que celles des transitions sortant de l'état 2, ce qui est en contradiction avec les valeurs des potentiels non normalisés. Dans le modèle CRF, le chemin le plus probable est $(1, 2, 3)$, qui visite les étiquettes de plus hauts potentiels en passant par les transitions les plus fortes. En revanche, le chemin $(1, 1, 1)$ est favorisé par la normalisation locale dans le RBD. Sa probabilité est alors $4/9$, alors que la probabilité du chemin $(1, 2, 3)$ est $5/12$.

Des potentiels non normalisés pour les transitions peuvent quelquefois permettre une modélisation plus interprétable des contraintes qui s'exercent sur les séquences d'étiquettes. Considérons par exemple un MMC construit à partir de l'automate représenté figure 3.7. C'est un automate « gauche-droite » sans saut, c'est-à-dire qui ne comporte pas de cycle de longueur strictement supérieure à 1, et dont le nombre de transitions sortant de chaque

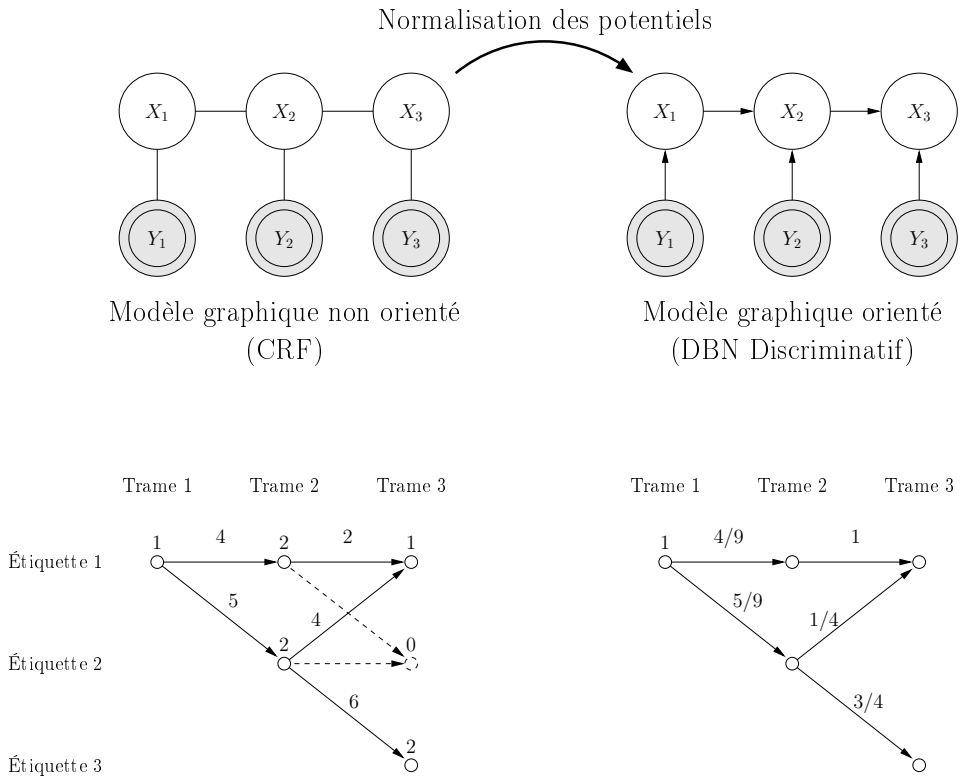


FIGURE 3.6 – Illustration du problème du *label bias* à travers un exemple simple : à gauche, un modèle CRF à 3 étiquettes ; à droite, le modèle graphique orienté obtenu en normalisant les potentiels. Notons que cette normalisation doit prendre en compte conjointement les potentiels de transition ainsi que les potentiels d'observation des états d'arrivée de ces transitions. En haut, les représentations graphiques des modèles. En bas, les treillis de décodage avec les potentiels ou probabilités conditionnelles (les états de potentiels nuls sont en pointillés ou ne sont pas représentés).

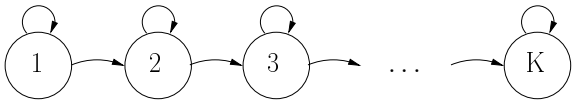


Figure 3.7 – Automate « gauche-droite » sans saut.

état est inférieur ou égal à 2. On sait que la séquence de variables cachées commence par l'étiquette 1. On pose donc la probabilité initiale $P(X_1 = 1) = 1$. Calculons par exemple les probabilités de transitions à fixer pour que toutes les séquences $\mathbf{X}_{1:N}$ de longueur N admissibles (c'est-à-dire de probabilité non nulle) aient la même probabilité *a priori*. La probabilité d'auto-transition (c'est-à-dire de rester dans un même état) d'un état k à la trame n est alors le rapport du nombre de séquences admissibles visitant cet état à la trame n et du nombre de séquences admissibles visitant le même état à la trame $n+1$. Cette valeur est définies par :

$$\forall n \leq N, \forall k \leq K, \quad P(X_{n+1} = k | X_n = k) = \frac{\sum_{i=0}^{K-k} \binom{N-n-1}{i}}{\sum_{i=0}^{K-k} \binom{N-n}{i}} \tag{3.26}$$

où la notation $\binom{j}{i} = \frac{j!}{i!(j-i)!}$ représente un coefficient binomial. La chaine de Markov ainsi formée est donc non homogène, c'est-à-dire que les probabilités de transitions dépendent de la trame temporelle n. De plus, les probabilités de transition dépendent de la longueur N de la séquence décodée. Par contre, cet *a priori* non informatif peut être exprimé de façon très simple dans un CRF de même structure. Il suffit en effet de fixer une même valeur pour toutes les transitions possibles, par exemple :

$$\forall k \leq K, \ \psi(k+1, k) = \psi(k, k) = 1. \tag{3.27}$$

Cela donne des potentiels de transitions plus facilement interprétables.

Problème d'un décodage partiel

Un inconvénient possible des CRF est qu'ils ne permettent pas de décodage en ligne. En effet, ils modélisent les probabilités conditionnelles des étiquettes, sachant la séquence entière d'observation. Il n'est donc pas possible en toute rigueur d'inférer des informations sur les variables cachées si l'on dispose seulement d'une partie des observations. Seul un décodage *global* est alors possible.

En pratique, une hypothèse supplémentaire permet tout de même d'utiliser un modèle CRF pour effectuer un décodage partiel. En effet, on peut supposer que les probabilités des séquences partielles sont proportionnelles aux produits partiels des potentiels. Cela implique en particulier que les fonctions d'observation ϕ ne font pas apparaitre d'observation postérieure à la trame courante. Cette hypothèse s'écrit :

$$\forall N_1 \leq N, \ P(\mathbf{X}_{1:N_1} | \mathbf{Y}_{1:N_1}) = \frac{1}{Z(\mathbf{Y}_{1:N_1})} \phi(X_1, \mathbf{Y}_{1:N_1}) \prod_{n=2}^{N_1} \psi(X_n, X_{n-1}, \mathbf{Y}_{1:N_1}) \phi(X_n, \mathbf{Y}_{1:N_1}). \tag{3.28}$$

Cela revient à considérer à chaque nouvelle trame un modèle différent, modélisant les probabilités conditionnelles des séquences partielles d'étiquettes jusqu'à la trame courante, sachant la séquence partielle d'observations. Moyennant ces suppositions, les mêmes stratégies que dans les DBN peuvent être envisagées pour un décodage *causal*, c'est-à-dire utilisant uniquement les informations issues de trames passées. Mais de la même façon, l'alignement obtenu ne sera pas forcément le résultat optimal, connaissant la séquence entière.

3.4 Modélisation des durées dans les modèles CRF

La modélisation de la durée des agrégats constitue une partie très importante du problème d'alignement temporel. L'alignement est même totalement déterminé par ces durées si l'on suppose que l'enregistrement est une interprétation sans erreur de la partition, dans le sens où tous les agrégats de la partition sont joués dans l'ordre indiqué par celle-ci. Un état de l'art important de ce domaine existe dans le cas des MMC et des réseaux bayésiens en général (voir par exemple les articles de Rabiner [1989] ; Johnson [2005] ; Yu [2010] et leurs références). En revanche, la question a, à notre connaissance, peu été traitée dans le cas des CRF. C'est pourquoi nous nous intéressons ici aux modèles temporels associés aux CRF, c'est-à-dire aux lois régissant les durées des agrégats.

3.4.1 Transitions markoviennes

Nous étudions dans un premier temps les modèles CRF pour lesquels la fonction de transition dépend uniquement des agrégats. Cela correspond à la transposition dans le cadre CRF d'un modèle de Markov caché comme celui représenté figure 3.3. L'hypothèse markovienne (conditionnellement aux observations) porte sur la séquence $\mathbf{C}_{1:N}$ des agrégats et la fonction de transition ne dépend pas ici des observations. De plus, comme indiqué en section 3.3 nous supposons que cette fonction est indépendante de la trame temporelle.

En pratique, dans les modèles utilisés pour l'alignement audio-sur-partition, les transitions sont très contraintes. En effet, si l'on suppose que l'enregistrement est une interprétation sans erreur de la partition, alors l'automate des agrégats est un automate « gauche-droite » sans saut, comme celui de la figure 3.4. Seules deux transitions sont alors possibles depuis chaque agrégat. La première possibilité est de continuer l'agrégat courant, ce qui se traduit par l'égalité $C_{n+1} = C_n$ et la seconde est de passer à l'agrégat suivant ($C_{n+1} = C_n + 1$). La fonction de transition est alors de la forme :

$$\psi(C_n, C_{n-1}) = \begin{cases} \lambda_0(C_{n-1}) & \text{si } C_n = C_{n-1} \\ \lambda_1(C_{n-1}) & \text{si } C_n = C_{n-1} + 1 \\ 0 & \text{sinon.} \end{cases} \tag{3.29}$$

Probabilité de durée

Un modèle CRF étant discriminatif, il ne définit pas en toute rigueur de distribution *a priori* des durées (sans considérer les observations). En revanche, on peut étudier les probabilités dans le cas où les observations sont non-informatives (c'est-à-dire que pour chaque trame, les valeurs des potentiels d'observation sont uniformes), ce qui a la même

interprétation intuitive. Dans cette section, nous considérerons toutes les probabilités conditionnellement à des observations non informatives. Notons $\phi(\mathbf{Y}_{1:N}, n) = \phi(C_n, \mathbf{Y}_{1:N})$ la valeur commune de la fonction d'observation à la trame n. La probabilité d'une séquence d'agrégat $\mathbf{C}_{1:N}$ s'écrit alors, d'après l'équation (3.11) :

$$\begin{aligned} P(\mathbf{C}_{1:N}|\mathbf{Y}_{1:N}) &= \frac{1}{Z(\mathbf{Y}_{1:N})}\phi(\mathbf{Y}_{1:N},1)\prod_{n=2}^{N}\psi(C_n,C_{n-1})\,\phi(\mathbf{Y}_{1:N},n) \\ &= \frac{\boldsymbol{\phi}(\mathbf{Y}_{1:N})}{Z(\mathbf{Y}_{1:N})}\prod_{n=2}^{N}\psi(C_n,C_{n-1}) \end{aligned} \tag{3.30}$$

avec $\boldsymbol{\phi}(\mathbf{Y}_{1:N}) = \prod_{n=1}^{N}\phi(\mathbf{Y}_{1:N}, n)$. Les probabilités de transition, sachant les observations non informatives, sont alors données par :

$$\begin{aligned} P(C_n = j|C_{n-1} = i, \mathbf{Y}_{1:N}) &= \frac{\sum_{\mathbf{C}_{1:N}} P(\mathbf{C}_{1:N}|\mathbf{Y}_{1:N})\mathbf{1}_{\{C_{n-1}=i\}}\mathbf{1}_{\{C_n=j\}}}{\sum_{\mathbf{C}_{1:N}} P(\mathbf{C}_{1:N}|\mathbf{Y}_{1:N})\mathbf{1}_{\{C_{n-1}=i\}}} \\ &= \frac{\sum_{\mathbf{C}_{1:N}} \frac{\boldsymbol{\phi}(\mathbf{Y}_{1:N})}{Z(\mathbf{Y}_{1:N})}\prod_{k=2}^{N}\psi_k(C_k,C_{k-1})\mathbf{1}_{\{C_{n-1}=i\}}\mathbf{1}_{\{C_n=j\}}}{\sum_{\mathbf{C}_{1:N}} \frac{\boldsymbol{\phi}(\mathbf{Y}_{1:N})}{Z(\mathbf{Y}_{1:N})}\prod_{k=2}^{N}\psi_k(C_k,C_{k-1})\mathbf{1}_{\{C_{n-1}=i\}}} \\ &= \frac{\sum_{\mathbf{C}_{1:n-1}}\prod_{k=2}^{n-1}\psi_k(C_k,C_{k-1})\psi_{n-1}(i,C_{n-2})\;\psi_n(j,i)\sum_{\mathbf{C}_{n+1:N}}\psi_{n+1}(C_{n+1},j)\prod_{k=n+2}^{N}\psi(C_k,C_{k-1})}{\sum_{\mathbf{C}_{1:n-1}}\prod_{k=2}^{n-1}\psi_k(C_k,C_{k-1})\psi_{n-1}(i,C_{n-2})\sum_{\mathbf{C}_{n:N}}\psi_n(C_n,C_{n-1})\prod_{k=n+2}^{N}\psi_k(C_k,C_{k-1})} \\ &= \frac{\psi_n(j,i)\sum_{C_{n+1}}\psi_{n+1}(C_{n+1},j)\prod_{k=n+2}^{N}\sum_{C_k}\psi_k(C_k,C_{k-1})}{\sum_{C_n}\psi_n(C_n,c_{n-1})\prod_{k=n+1}^{N}\sum_{C_k}\psi_k(C_k,C_{k-1})} \end{aligned} \tag{3.31}$$

où la notation $\psi_n(j,i)$ remplace ici l'expression $\psi(j,i,n)$ introduite en (3.11), pour plus de concision. Cette expression indique que les probabilités de transitions ne sont pas forcément indépendantes entre elles, même dans le cas d'observations non informatives. De plus, elles dépendent de la position temporelle dans l'enregistrement, ce qui les rend difficiles à interpréter.

Fonction de transition normalisée

En revanche, si l'on fixe la contrainte de normalisation de la fonction de transition $\forall c, \sum_{c'}\psi(c',c) = \lambda_0(c) + \lambda_1(c) = 1$, on obtient un modèle équivalent à un MMC. En effet, l'équation (3.31) devient :

$$P(c_n|c_{n-1}, \mathbf{Y}_{1:N}) = \psi(c_n, c_{n-1}). \tag{3.32}$$

Les valeurs de la fonction de transition sont alors égales aux probabilités de transitions. Ces probabilités sont donc indépendantes et la loi de la longueur L^c d'un agrégat c est calculable :

$$P(L^c = l|\mathbf{Y}) = \lambda_0(c)^{l-1}\lambda_1(c). \tag{3.33}$$

Il s'agit d'une loi géométrique, donc décroissante, qui ne permet pas de favoriser une longueur d'agrégat particulière (sauf s'il s'agit de la longueur 1). En revanche, si une durée l^c est attendue, on peut fixer les valeurs des paramètres afin de maximiser la probabilité de cette durée. Les valeurs obtenues sont $\lambda_0 = \frac{l^c-1}{l^c}$ et $\lambda_1(c) = \frac{1}{l^c}$.

Contraintes temporelles sur les séquences d'étiquettes individuelles

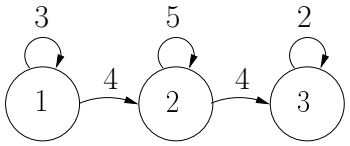
Bien que les paramètres λ_0 et λ_1 aient une interprétation intuitive simple, en tant que « pénalités » affectées aux transitions, les probabilités de transitions associées, exprimées en (3.31), sont difficilement exploitables dans le cas général. On peut alors se poser la question de la pertinence de l'étude des probabilités des transitions ou des durées, pour caractériser les contraintes temporelles utiles à l'alignement. En effet, notre stratégie d'alignement, utilisant le critère MAP défini dans l'équation (3.10), cherche à déterminer une séquence de probabilité maximale. Or la probabilité $P(L^c = l)$ est la somme des probabilités de toutes les séquences d'étiquettes vérifiant $L^c = l$. Elle peut donc dépendre du nombre de séquences vérifiant cette propriété et la valeur de cette probabilité n'exprime alors pas forcément les contraintes temporelles s'exerçant sur chaque séquence considérée individuellement.

Prenons l'exemple de la figure 3.8, représentant le décodage d'un modèle très simple à trois étiquettes, pour une séquence d'observations non informatives. On rappelle que la probabilité d'une séquence d'étiquettes (conditionnellement à des observations non informatives) est proportionnelle au produit des potentiels de transitions (le « score »). Le facteur de normalisation est ici égal à 114. Dans cet exemple, la probabilité que la durée de l'étiquette 2 soit égale à 1 est $P(L^2 = 1) = \frac{12+16+8}{114} = \frac{36}{114}$. La durée 1 est donc plus probable que la durée 3, puisqu'on a $P(L^2 = 3) = \frac{25}{114}$. Or, chaque séquence d'étiquettes vérifiant $L^2 = 1$ est moins probable que la séquence $2 \to 2 \to 2$. Et de fait, c'est cette dernière séquence qui est décodée.

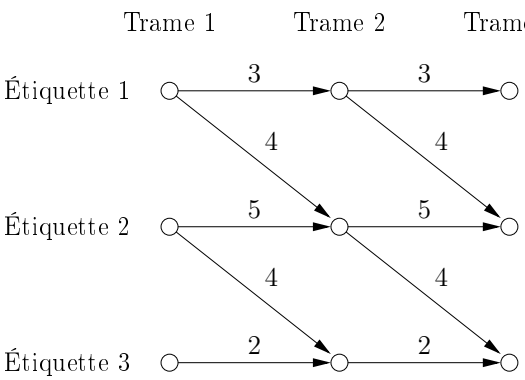
Cet exemple illustre donc le fait que les probabilités marginales des durées, de la forme $P(L^c = l)$, ne correspondent pas aux contraintes temporelles s'appliquant aux séquences d'étiquettes dans un décodage selon le critère MAP. En effet, un tel décodage considère la probabilité de chaque séquence individuellement. Il est alors plus pertinent de considérer les « scores de transitions » associés aux durées, c'est-à-dire appliqués à chaque séquence vérifiant les durées considérées. Dans notre modèle markovien, le score associé à la durée l d'un agrégat c est égal à $\lambda_0(c)^{l-1}\lambda_1(c)$.

Comme précédemment, quelles que soient les valeurs des paramètres λ_0 et λ_1, le score d'une durée est une fonction monotone de l. Il n'est donc toujours pas possible de favoriser une longueur particulière. Si des durées $l_1, l_2, \ldots$ sont attendues pour les agrégats, on peut vouloir faire en sorte que les scores associés à ces durées soient tous égaux, par exemple à 1. Cela donne alors la relation $\lambda_1(c) = \frac{1}{\lambda_0(c)^{l^c}}$. Il n'y a en revanche pas vraiment de stratégie permettant de fixer *a priori* les paramètres λ_0 de façon satisfaisante. En

Automate des étiquettes :



Treillis de décodage :



Séquence d'étiquettes	« Score »
$1 \to 1 \to 1$	9
$1 \to 1 \to 2$	12
$1 \to 2 \to 2$	20
$1 \to 2 \to 3$	16
$2 \to 2 \to 2$	25
$2 \to 2 \to 3$	20
$2 \to 3 \to 3$	8
$3 \to 3 \to 3$	4
facteur de normalisation :	114

Figure 3.8 – Exemple simple de modèle CRF markovien à trois étiquettes, pour le décodage d'une séquence de longueur trois où les observations sont non informatives. Le « score » d'une séquence d'étiquette est ici défini comme le produit des potentiels de transition associés à cette séquence.

effet, le score de transition associé à une séquence d'étiquettes est toujours de la forme $\prod_{c=1}^{K} \lambda_0(c)^{l^c-1} \prod_{c=1}^{K-1} \lambda_1(c)$ où K est la dernière étiquette de la série et $\sum_{c=1}^{K} l^c = N$. On peut alors montrer que le modèle aura toujours tendance à favoriser les durées les plus longues possibles des étiquettes de paramètre λ_0 maximal.

Par contre, dans le cas où l'on n'a pas de connaissance *a priori* sur les durées des agrégats (ou si l'on ne souhaite pas injecter d'information de durée dans le modèle), on peut fixer tous les λ à la valeur 1. De cette façon, tous les chemins auront la même probabilité.

3.4.2 Transitions semi-markoviennes

Nous avons vu dans la section précédente qu'une fonction de transition markovienne conduit à un « score de transition » qui est une fonction exponentielle de la durée de chaque agrégat. Cette forme ne permet donc pas de favoriser une durée particulière pour un agrégat (à moins que cette durée ne soit égale à 1 trame). Pour pouvoir modéliser de façon plus souple les durées des étiquettes, d'autres types de structures ont été introduites, comme généralisations des MMC.

Pour une modélisation explicite des durées, ces structures introduisent une variable cachée supplémentaire, représentant la « position » à l'intérieur de l'état (l'agrégat dans notre cas) courant. Cette nouvelle variable d'*occupation* d'agrégat est notée D. Comme elle rend compte uniquement de la dimension temporelle, cette variable n'a pas d'influence sur la fonction d'observation, mais seulement sur la fonction de transition. Différents noms ont été donnés à ces types de modèles (MMC à durée explicite [Ferguson, 1980], MMC inhomogène [Ramesh et Wilpon, 1992], MMC à états développés [Cook et Russell, 1986]...), suivant les transitions possibles entre les valeurs de la variables d'occupation. Nous les regroupons cependant sous l'appellation de modèles semi-markoviens [Yu, 2010]. Une comparaison des différentes structures couramment utilisées est menée par Johnson [2005].

Pour une interprétation plus intuitive de la variable d'occupation, on peut représenter les transitions entre les différentes valeurs possibles de cette variable à l'intérieur d'un même agrégat par un automate, comme illustré figure 3.9. La structure de l'automate des occupations de chaque agrégat détermine alors la modélisation temporelle. Cependant, les contraintes temporelles résultant d'une structure arbitraire peuvent avoir une formulation très compliquée [Bonafonte *et al.*, 1996] et être difficilement interprétables. Les deux topologies présentées figure 3.9, déjà étudiées par Russell et Cook [1987], permettent néanmoins d'exprimer des modèles de durée explicitement calculables.

Structure de type A : modèle temporel

La topologie de type A est exploitée dans la plupart des modèles probabilistes pour l'alignement audio-sur-partition, par exemple dans les travaux de Raphael [1999], Orio [2002], Cont [2006] et Montecchio et Orio [2009]. Cette structure a en effet, l'avantage de pouvoir représenter des durées arbitrairement grandes avec relativement peu de « sous-étiquettes » d'occupation, grâce aux auto-transitions. De plus, dans le cadre des MMC, où la condition $\lambda_0 + \lambda_1 = 1$ est vérifiée, seuls deux paramètres (la probabilité λ_0 et le nombre

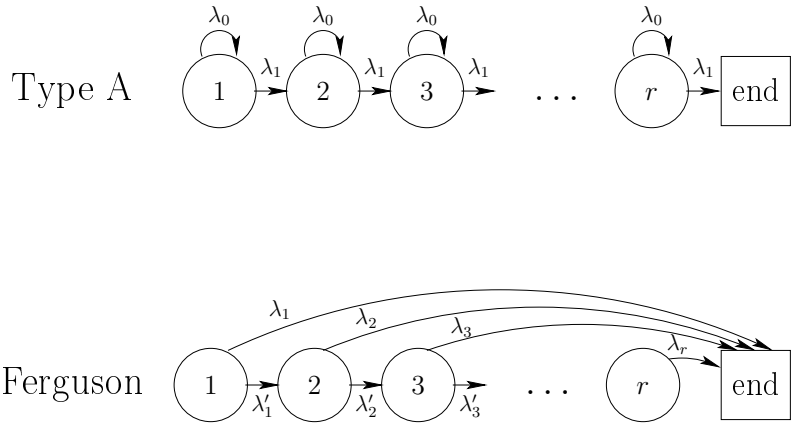


Figure 3.9 – Deux sous-structures courantes pour la modélisation des durées d'étiquettes.

de sous-étiquettes r) contrôlent le modèle temporel associé.

Pour étudier les contraintes s'appliquant aux durées d'étiquettes dans un modèle de type semi-markovien, il est nécessaire de prendre en compte le critère de décodage choisi. En effet, si l'on décode conjointement toutes les variables du modèle, c'est-à-dire si l'on cherche à déterminer la séquence $(\hat{\mathbf{c}}_{1:N}, \hat{\mathbf{d}}_{1:N})$ définie par :

$$(\hat{\mathbf{c}}_{1:N}, \hat{\mathbf{d}}_{1:N}) = \underset{(\mathbf{C}_{1:N}, \mathbf{D}_{1:N})}{\arg\max}\ P(\mathbf{C}_{1:N}, \mathbf{D}_{1:N} | \mathbf{Y}_{1:N}), \tag{3.34}$$

alors le score de transition correspondant à la durée l —c'est-à-dire affecté a toute séquence telle que la longueur de l'agrégat considéré soit l— est égal à $\lambda_0^{l-r}\lambda_1^r$, pour $l \geq r$. Ce score est une fonction dépendant de façon exponentielle de la durée. L'utilisation de la structure de type A avec ce critère de décodage ne présente donc pas vraiment d'intérêt puisque la seule différence avec un modèle markovien est la contrainte que la durée de l'agrégat soit supérieure à r trames.

En revanche, si l'on marginalise les scores de transition associés à une durée l d'agrégat, on obtient alors $\binom{l-1}{r-1}\lambda_0^{l-r}\lambda_1^r$, ce qui correspond à une loi de probabilité binomiale négative dans le cas $\lambda_0 + \lambda_1 = 1$. Si le paramètre λ_0 est inférieur à 1, ce score considéré comme une fonction de l présente un maximum à la valeur $l = \lfloor \frac{r-1}{1-\lambda_0} \rfloor$.

Il est donc possible de favoriser cette durée grâce à la structure étudiée, en décodant uniquement les étiquettes de haut niveau (les agrégats), c'est-à-dire en marginalisant sur les variables d'occupation. La séquence optimale $\hat{\mathbf{c}}_{1:N}$ cherchée est donc :

$$\begin{aligned} \hat{\mathbf{c}}_{1:N} &= \underset{\mathbf{C}_{1:N}}{\arg\max}\, P(\mathbf{C}_{1:N} | \mathbf{Y}_{1:N}) \\ &= \underset{\mathbf{C}_{1:N}}{\arg\max} \sum_{\mathbf{D}_{1:N} \in \boldsymbol{\mathcal{D}}} P(\mathbf{C}_{1:N}, \mathbf{D}_{1:N} | \mathbf{Y}_{1:N}) \end{aligned} \tag{3.35}$$

où $\boldsymbol{\mathcal{D}}$ est l'ensemble des séquences de variables d'observations possibles. Une stratégie de programmation dynamique permet de calculer efficacement cette séquence optimale [Cook et Russell, 1986].

Structure de Ferguson

La structure précédente permet donc d'introduire des contraintes temporelles favorisant une durée précise. Cependant, cette contrainte est limitée à une forme binomiale négative. Pour palier à cette limitation, la topologie de Ferguson représentée figure 3.9 peut être utilisée, car elle permet l'expression d'un score arbitraire associé à la durée d'un agrégat. D'après la construction de l'automate correspondant, la valeur de la variable d'occupation est égale au nombre de trames écoulées depuis le début de l'agrégat courant.

Ferguson [1980] exploite cette propriété pour construire un MMC permettant la modélisation d'une forme quelconque de la loi de probabilité *a priori* de la durée L^c d'un état c (sous réserve que le support de cette loi soit borné). On fixe donc r à la plus grande valeur admissible, et les λ_l sont définis par :

$$\forall l \in \{1, \ldots, r\}, \quad \lambda_l(c) = P(L^c = l | L^c \geq l). \tag{3.36}$$

Les valeurs des λ'_l découlent alors de la contrainte de normalisation des probabilités de transition.

Il est à noter que dans une sous-structure de ce type, il existe au plus un chemin de longueur l reliant l'état initial à l'état final, quel que soit l. Il est donc équivalent de décoder conjointement toutes les variables ou de marginaliser les probabilités selon les variables d'occupation.

En pratique, au lieu d'utiliser les paramètres définis en (3.36), on peut fixer des valeurs dont l'interprétation est encore plus intuitive. En effet, comme le cadre CRF ne nécessite pas de normalisation des fonctions de transition, on peut poser :

$$\forall l \in \{1, \ldots, r\}, \quad \lambda'_l(c) = 1, \tag{3.37}$$

$$\lambda_l(c) = \rho(l, c) \tag{3.38}$$

où ρ est la fonction de score que l'on souhaite appliquer aux durées de l'agrégat. Cette forme peut être équivalente à la précédente en posant $\rho(l, c) = P(L^c = l)$ où P désigne la probabilité *a priori* du MMC. On a alors une formulation simple et intuitive des contraintes temporelles introduites.

3.4.3 Extension : prise en compte du tempo

Les modèles semi-markoviens permettent de représenter n'importe quelle contrainte de durées s'exerçant sur les séquences d'étiquettes. En revanche, ils présentent une limitation pour la modélisation temporelle de la musique. En effet, dans les modèles précédents, les scores de transitions associés aux durées des agrégats sont fixés *a priori*. Ces modèles ne permettent donc pas de modéliser les corrélations entre les durées d'agrégats (la seule source de corrélation est la contrainte reliant la somme des durées et la longueur de la séquence). Or le temps musical est en général très structuré et les durées sont très fortement corrélées.

Ces relations entre les durées de notes, qui peuvent être très contraintes, sont contenues dans les notions de *rythme* et de *tempo*. Dans la majorité des musiques, le *rythme*, c'est-à-dire les rapports entre les durées de notes, est caractéristique du morceau et doit donc

être invariant. La valeur rythmique d'un agrégat (blanche, noire, croche,...) correspond à la durée en pulsations.

Le *tempo* définit la durée de la pulsation, qui est l'unité rythmique de référence. On peut donc déduire la durée (en secondes) d'un agrégat d'après sa valeur rythmique et la valeur du tempo. Les durées (en secondes) des agrégats sont donc reliées entre elles par le tempo. Or, si les valeurs rythmiques sont connues, puisqu'indiquées par la partition, le tempo est à la fois inconnu et variable. La seule hypothèse faite est que les variations du tempo sont en général lentes par rapport à la durée de la pulsation, afin d'assurer une certaine régularité locale des pulsations. Ainsi, le tempo peut être considéré comme constant durant plusieurs pulsations consécutives.

Suivant ces considérations, les modèles utilisés par Raphael [2006] et Cont [2010] sont des réseaux bayésiens qui exploitent une variable aléatoire supplémentaire représentant le tempo courant. Afin de maintenir une interprétation intuitive de cette variable, elle est supposée constante pendant toute la durée d'un agrégat. Nous notons T^c la variable de tempo associée à l'agrégat c. L'intérêt de cette variable est la possibilité de faire dépendre les scores de transition associés aux durées d'agrégats de la valeur courante du tempo. On ajoute alors cette dépendance à la définition de l'équation (3.38) pour obtenir une nouvelle fonction de pénalité $\rho_{\mathrm{d}}(l, c, t)$. Dans les réseaux bayésiens cités plus haut, cette pénalité est définie comme la probabilité conditionnelle de la durée de l'agrégat sachant la valeur du tempo $P(L^c = l \big| T^c = t)$.

L'évolution de la variable de tempo est alors régie par une fonction de pénalité $\rho_{\mathrm{t}}(t^c, t^{c-1})$, qui correspond à la probabilité de variation de tempo $P(T^c \big| T^{c-1})$ d'un réseau bayésien. Il est à noter qu'un tel modèle peut en fait être considéré comme un modèle semi-markovien, modélisant les durées d'étiquettes de la forme (C, T).

Diverses fonctions de pénalité ρ peuvent être utilisées, conduisant à des contraintes temporelles différentes. Nous détaillerons dans le chapitre suivant la forme que nous proposons pour la modélisation des durées musicales.

3.5 Conclusion

Dans ce chapitre, nous avons présenté l'utilisation de modèles graphiques pour l'alignement temporel audio sur partition. Nous avons défini le problème d'alignement musique-sur-partition comme un problème d'étiquetage de séquence, consistant à associer à chaque trame temporellle de l'enregistrement un agrégat de la partition. Ce problème peut alors être traité avec un modèle graphique probabiliste, en introduisant pour chaque trame une variable aléatoire cachée représentant l'agrégat joué. Nous avons présenté, à travers l'exemple d'un modèle de Markov caché, comment des modèles génératifs utilisant le cadre des réseaux bayésiens dynamiques sont employés dans la littérature.

Nous avons ensuite vu que la classe des modèles discriminatifs de type champs aléatoires conditionnels (CRF) pouvait être considérée comme une généralisation des modèles précédents pour la tâche d'alignement. Ce cadre présentant un certain nombre d'avantages sur les réseaux bayésiens dynamiques, nous avons choisi d'employer le formalisme CRF dans nos travaux.

Ce formalisme permet alors d'exprimer diverses formes de dépendances entre les variables aléatoires, qui mènent à différentes contraintes s'appliquant aux durées des agrégats. Trois structures particulières ont été étudiées, ainsi que les modèles temporels qu'elles occasionnent. Ces structures peuvent alors être utilisées pour la conception de systèmes d'alignement temporel audio-sur-partition, qui est l'objet du chapitre suivant.

Chapitre 4

Présentation de nos modèles d'alignement par CRF

Nous avons vu, dans le chapitre précédent, que les modèles graphiques utilisés pour l'alignement temporel peuvent tous être exprimés dans le cadre des CRF. Nous utilisons donc ce formalisme pour présenter les modèles envisagés dans cette thèse. Nous proposons ici trois formes différentes de la fonction de transition, correspondant aux trois modèles temporels exposés dans le chapitre précédent.

Nous proposons en outre une fonction d'observation exploitant plusieurs types de descripteurs acoustiques représentant respectivement le contenu spectral, l'« impulsivité » (à travers la détection des transitoires) et le tempo.

Nous exposons ensuite la stratégie adoptée pour le décodage de ces modèles, avant d'évaluer les systèmes obtenus dans la tâche d'alignement, sur les deux bases de données MAPS et RWC-pop.

4.1 Fonctions de transition utilisées

Nous nous intéressons tout d'abord aux fonctions de transition de nos trois modèles, qui définissent les contraintes s'appliquant aux durées des agrégats.

4.1.1 Modèle markovien (MCRF)

Notre premier modèle, appelé modèle CRF markovien (MCRF), est fondé sur la fonction de transition markovienne présentée en section 3.4.1. Cependant par rapport à cette dernière structure, une variable aléatoire supplémentaire est introduite afin de modéliser les deux principales *phases* d'un agrégat (la phase d'attaque et la phase stationnaire). Cela permet alors de prendre en compte certaines variations des observations, pouvant intervenir à un niveau plus fin que l'agrégat. Nous notons A_n la variable aléatoire représentant la phase de l'agrégat courant à la trame n. La valeur de A_n sera égale à la fonction indicatrice de la phase d'attaque. On aura donc $A_n = 1$ si et seulement si la trame n correspond à la phase d'attaque de l'agrégat C_n. L'étiquette du modèle obtenu s'écrit ainsi $X_n = (C_n, A_n)$.

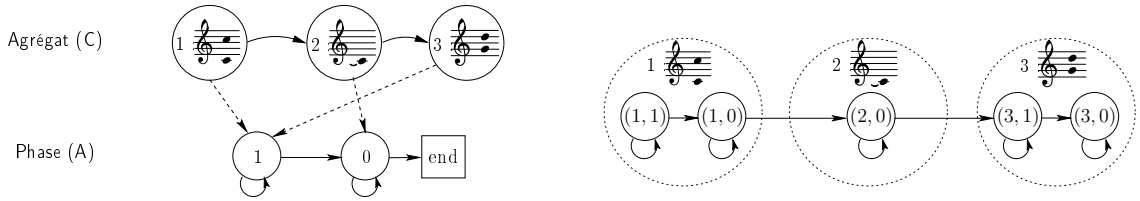


Figure 4.1 – **(a)** : Automate des étiquettes du modèle hiérarchique d'une partition simple. **(b)** : L'automate "aplati" équivalent. Ici, les agrégats 1 et 3 sont *attaqués* (contiennent une note attaquée), alors que l'agrégat 2 est *lié*. Ce dernier ne comporte donc pas de phase d'attaque.

Pour représenter ce modèle, on peut utiliser un formalisme équivalent à celui d'un modèle de Markov caché hiérarchique [Fine et Singer, 1998]. Un exemple de modèle hiérarchique simple est représenté figure 4.1 (a). Les étiquettes sont structurées en deux niveaux hiérarchiques. Le niveau supérieur correspond à l'agrégat et le niveau inférieur représente la phase de l'agrégat courant. Les transitions possibles dans un automate hiérarchique sont les suivantes. Lorsqu'un nouvel agrégat est visité, l'étiquette de niveau inférieur est atteinte par une « transition verticale », représentée en pointillés sur la figure 4.1 (a). Durant toute la durée de l'agrégat, le système suit les transitions « horizontales » (lignes continues) jusqu'à atteindre l'état de fin. Alors, le système remonte au niveau supérieur et un nouvel agrégat est visité en suivant une transition horizontale à ce niveau.

Il est à noter qu'une structure hiérarchique d'étiquettes peut toujours être convertie en un automate "aplati", à un seul niveau [Murphy, 2002], comme illustré sur la figure 4.1 (b). L'espace des états de l'automate aplati est alors le produit cartésien des espaces des variables aléatoires contenues dans une étiquette. La structure hiérarchique permet une représentation et une interprétation plus intuitive des transitions entre étiquettes.

Dans notre utilisation de la phase des agrégats, nous tirons parti de la distinction faite entre les agrégats *attaqués* et les agrégats *liés* (section 2.1.2). Une phase d'attaque sera présente au début de chaque agrégat attaqué (comme les agrégats 1 et 3 de la figure 4.1), suivie par la phase stationnaire. Les agrégats liés ne contiennent qu'une phase stationnaire.

On désigne par $\dot{\mathcal{C}}$ l'ensemble des agrégats attaqués et $\mathbf{1}_{\dot{\mathcal{C}}}$ la fonction indicatrice de cet ensemble. La fonction de l'équation (3.29) est alors modifiée, afin de prendre en compte la nouvelle variable aléatoire. La fonction de transition du modèle MCRF, notée ψ^{M}, s'écrit alors

$$\psi^{\mathrm{M}}(X_n, X_{n-1}) = \begin{cases} \lambda_0(C_{n-1}) & \text{si } C_n = C_{n-1} \text{ et } A_n \leq A_{n-1} \\ \lambda_1(C_{n-1}) & \text{si } C_n = C_{n-1} + 1 \text{ et } A_n = \mathbf{1}_{\dot{\mathcal{C}}}(C_n) \\ 0 & \text{sinon.} \end{cases} \tag{4.1}$$

Comme les variables A_{n-1} et A_n sont binaires, le terme $A_n \leq A_{n-1}$ comprend tous les évènements sauf la combinaison $A_{n-1} = 0$ et $A_n = 1$. Cette notation compacte exprime le fait qu'une phase d'attaque ($A_n = 1$) ne peut pas intervenir après une phase stationnaire ($A_{n-1} = 0$) à l'intérieur d'un même agrégat. Le terme $A_n = \mathbf{1}_{\dot{\mathcal{C}}}(C_n)$ force les agrégats attaqués à commencer par une phase d'attaque et les agrégats liés à commencer par une

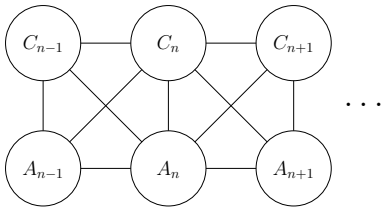


FIGURE 4.2 – Modèle graphique représentant les dépendances entre les variables cachées du modèle MCRF (conditionnellement à des observations non informatives, qui ne sont pas représentées).

phase stationnaire. Le modèle graphique correspondant à cette fonction de transition est représenté figure 4.2

Malgré l'introduction de la variable de phase, le modèle temporel associé à cette structure est en pratique équivalent à celui présenté en section 3.4.1. En effet, le score de transition associé à la durée l de l'agrégat c est toujours égal à $\lambda_0(c)^{l-1}\lambda_1(c)$. La seule modification est la contrainte qui force les agrégats attaqués à durer au moins deux trames. En effet, les deux phases (attaque et phase stationnaire) doivent être visitées. En revanche, l'introduction de la variable de phase rend possible l'utilisation d'une fonction d'observation qui rend compte de la différence entre les deux phases.

Dans notre utilisation pratique de ce modèle, nous choisissons de ne pas exploiter les informations de durées d'agrégats indiquées par la partition. En effet, certaines expériences préliminaires indiquaient que l'emploi de ces durées pour fixer les valeurs des paramètres λ, comme exposé section 3.4.1, n'améliorait pas toujours la qualité des résultats. De plus, nous avons constaté qu'une variation (modérée) de ces valeurs avait peu d'influence sur les alignements obtenus. Les paramètres λ_0 et λ_1 sont alors fixés à la valeur 1. Ce choix présente en outre l'avantage d'une invariance aux cas où les durées théoriques des agrégats sont non accessibles dans la partition, ou peu fiables (correspondant par exemple à l'utilisation d'une partition issue d'une reconnaissance optique de partition graphique).

4.1.2 Modèle semi-markovien (SMCRF)

La seconde structure utilisée correspond à l'adaptation du modèle temporel semi-markovien présenté à la section 3.4.2 afin de prendre en compte la variable de phase. La fonction de transition doit alors prendre en compte les deux variables aléatoires « de niveau inférieur » A_n et D_n pour chaque agrégat C_n, représentant respectivement la phase et l'occupation. Or ces deux dernières sont corrélées, puisque la phase d'attaque doit être présente au début d'un agrégat attaqué et durer peu de temps. Plus précisément, nous supposons que cette phase doit durer au plus 2 trames[1]. Ces contraintes peuvent être représentées par les automates des étiquettes de la figure 4.3, dont la structure est directement tirée de celle de Ferguson. De façon similaire au modèle markovien, les sous-structures

1. Nous verrons en section 4.2 que le pas d'avancement entre deux trames est fixé à 20 ms. De ce fait, la phase d'attaque est limitée à 40 ms, ce qui est un choix raisonnable puisque la plupart des instruments ont des temps d'attaque inférieurs à quelques dizaines de millisecondes [Fletcher et Rossing, 1998].

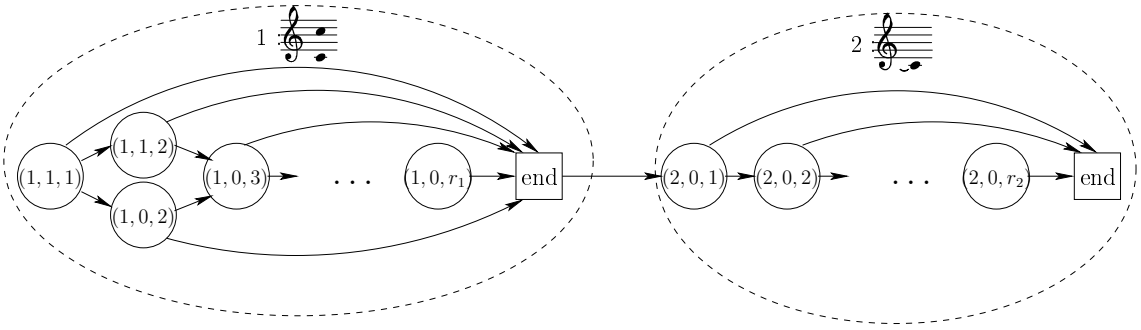


Figure 4.3 – Sous-structure utilisée dans notre modèle CRF semi-markovien. Le premier agrégat est attaqué et le second est lié. Les variables représentent respectivement l'agrégat C, la phase A et l'occupation D.

des agrégats attaqués et des agrégats liés sont différentes.

On définit alors le potentiel $\psi_{\text{a}}^{\text{SM}}$, qui exprime cette contrainte d'« admissibilité d'une étiquette » :

$$\psi_{\text{a}}^{\text{SM}}(A_n, D_n, C_n) = \begin{cases} \mathbf{1}_{\{A_n=1\}} & \text{si } C_n \in \dot{\mathcal{C}} \text{ et } D_n = 1 \\ 1 & \text{si } C_n \in \dot{\mathcal{C}} \text{ et } D_n = 2 \\ \mathbf{1}_{\{A_n=0\}} & \text{sinon.} \end{cases} \tag{4.2}$$

Cette formule exprime donc le fait que pour un agrégat attaqué ($C_n \in \dot{\mathcal{C}}$), la première trame doit correspondre à une phase d'attaque. La deuxième peut être indifféremment en phase d'attaque ou de *sustain* et toutes les autres seront en phase de *sustain*. Dans un agrégat continué, seule cette dernière phase est autorisée.

Comme pour le modèle markovien présenté précédemment, les transitions entre agrégats sont très contraintes : après un agrégat c on passe forcément à l'agrégat $c+1$. De plus, le début d'un agrégat est signalé par la valeur $D = 1$ de la variable d'occupation. Pour toute autre valeur de cette variable, l'agrégat courant est forcément le même qu'à la trame précédente. Le potentiel $\psi_{\text{c}}^{\text{SM}}$ régissant les transitions entre agrégats s'écrit donc :

$$\psi_{\text{c}}^{\text{SM}}(C_n, C_{n-1}, D_n) = \begin{cases} \mathbf{1}_{\{C_n=C_{n-1}+1\}} & \text{si } D_n = 1 \\ \mathbf{1}_{\{C_n=C_{n-1}\}} & \text{sinon.} \end{cases} \tag{4.3}$$

Enfin, les équations (3.37) et (3.38), donnent le potentiel $\psi_{\text{d}}^{\text{SM}}$ exprimant les pénalités de durée de note :

$$\psi_{\text{d}}^{\text{SM}}(D_n, D_{n-1}, C_{n-1}) = \begin{cases} \rho(D_{n-1}, C_{n-1}) & \text{si } D_n = 1 \\ 1 & \text{si } D_n = D_{n-1} \\ 0 & \text{sinon.} \end{cases} \tag{4.4}$$

Pour les scores associés aux durées d'agrégats, nous choisissons une pénalité de type gaussien, dont la moyenne est la durée attendue, déduite de la partition. Soit l^c cette valeur attendue de l'agrégat c, on a alors :

$$\rho(l, c) = \exp\left(-\gamma_{\text{l}} \left(l - l^c\right)^2\right). \tag{4.5}$$

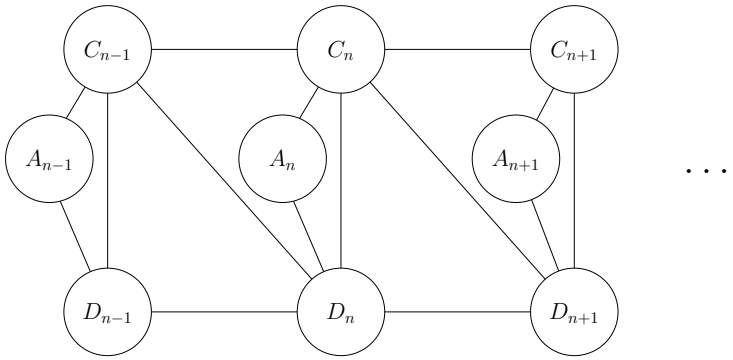


FIGURE 4.4 – Représentation graphique des dépendances entre les variables cachées du modèle SMCRF (conditionnellement à des observations non informatives, qui ne sont pas représentées).

Le paramètre γ_{l} contrôle la contrainte temporelle. Il est homogène à l'inverse de la variance d'une gaussienne.

On rappelle que X_n désigne l'étiquette de la trame n, représentant toutes les variables cachées associées à cette trame. La fonction de transition du modèle SMCRF est alors donnée par :

$$\psi^{\mathrm{SM}}(X_n, X_{n-1}) = \psi_{\mathrm{d}}^{\mathrm{SM}}(D_n, D_{n-1}, C_{n-1})\psi_{\mathrm{c}}^{\mathrm{SM}}(C_n, C_{n-1}, D_n)\psi_{\mathrm{a}}^{\mathrm{SM}}(A_n, D_n, C_n) \tag{4.6}$$

et le modèle graphique correspondant est représenté figure 4.4.

4.1.3 Modèle à tempo caché (HTCRF)

Le dernier modèle que nous proposons, appelé modèle à tempo caché (HTCRF pour *Hidden Tempo* CRF) exploite une variable de tempo, comme exposé en section 3.4.3. Le modèle temporel est alors déterminé par la donnée des deux fonctions de pénalités ρ_{d} et ρ_{t}, contrôlant respectivement les durées d'agrégats et les variations de tempo.

Pénalité de durée d'agrégat

Pour la première de ces pénalités, Raphael [2006] modélise la loi conditionnelle de la durée d'un agrégat c sachant le tempo t (exprimé en nombre de trames par pulsation) par une gaussienne dont la moyenne est $\ell_c t$, où ℓ_c est la durée en pulsations de l'agrégat. La variance est fixée comme paramètre. Dans le travail de Raphael, cette forme de la pénalité de durée est nécessaire pour le décodage optimal du modèle. En effet, son modèle représente le tempo par une variable aléatoire à valeurs réelles (continues), ce qui empêche l'utilisation des algorithmes classiques de décodage par programmation dynamique. Un algorithme spécifique est donc nécessaire, tirant parti de la propriété de stabilité des gaussiennes par conjugaison.

À l'opposé, nous choisissons de discrétiser l'ensemble des tempos possibles, afin de modéliser le tempo par une variable aléatoire discrète. Cela permet l'emploi de pénal-

ités de durées quelconques, tout en conservant la possibilité d'un décodage optimal par l'algorithme de Viterbi.

Nous adoptons une pénalité de forme gaussienne centrée sur la durée attendue d'après le tempo $\ell_c t$, mais à la différence de Raphael, nous choisissons un écart-type proportionnel à cette durée. Ce choix est motivé par notre intuition musicale, mais correspond aussi à des résultats de psychoacoustique. En effet, les expériences de Friberg et Sundberg [1995] indiquent que le seuil différentiel de durée entre deux notes répétées, c'est-à-dire le seuil au-delà duquel la différence est perceptible, est proportionnel à cette durée, lorsque cette dernière se situe entre 240 ms et 1 s. La fonction de pénalité ρ_d est donc définie comme suit :

$$\rho_\mathrm{d}(l, c, t) = \exp\left(- \gamma_\mathrm{d} \left| \frac{l - \ell_c t}{\ell_c t} \right|^2 \right). \tag{4.7}$$

Cette fonction est contrôlée par le paramètre $\gamma_\mathrm{d} \geq 0$.

Pénalité de variation de tempo

Comme Cemgil *et al.* [2001], nous construisons la pénalité de variation de tempo ρ_t comme une fonction du rapport entre les deux tempos. Ce choix est fondé sur l'intuition selon laquelle les changements de tempo sont relatifs plus qu'absolus. Par exemple, nous considérons qu'il est aussi probable de doubler le tempo que de le diminuer de moitié, quelle que soit sa valeur initiale. La fonction de pénalité ρ_t est choisie de type log-normale. Cependant, nous considérons que le cas d'un rapport de tempos supérieur à 2 (ou inférieur à $\frac{1}{2}$) correspond à la prise d'un nouveau tempo non lié au précédent. Ce cas est relativement fréquent dans la musique classique, où différentes parties d'un même morceau peuvent être jouées à des tempos différents. En revanche, du point de vue de la modélisation, ces ruptures de tempo rompent le cadre de l'hypothèse de variation lente. La pénalité appliquée est alors la même que celle associée à un doublement de tempo. On définit donc :

$$\rho_\mathrm{t}(t^c, t^{c-1}) = \max\left\{ \exp\left(- \gamma_\mathrm{t} \left| \log \frac{t^c}{t^{c-1}} \right|^2 \right),\ \exp\left(- \gamma_\mathrm{t} \big(\log 2 \big)^2 \right) \right\} \tag{4.8}$$

La dynamique de cette pénalité est contrôlée par le paramètre $\gamma_\mathrm{t} \geq 0$.

Fonction de transition

La figure 4.5 représente le modèle graphique correspondant au modèle HTCRF proposé. Notons que la variable T_n, représentant le tempo associé à la trame n, est égale à la variable T^{C_n} liée à l'agrégat courant. Comme dans le cas du SMCRF, le graphe reliant deux « tranches » temporelles successives peut être décomposé en plusieurs cliques et la fonction de transition peut alors être décomposée en un produit de plusieurs potentiels.

Les potentiels exprimant respectivement les contraintes d'« admissibilité des étiquettes » et de transitions entre agrégats sont les mêmes que pour le modèle SMCRF. On a donc

$$\psi_\mathrm{a}^\mathrm{HT} = \psi_\mathrm{a}^\mathrm{SM} \tag{4.9}$$

$$\psi_\mathrm{c}^\mathrm{HT} = \psi_\mathrm{c}^\mathrm{SM} \tag{4.10}$$

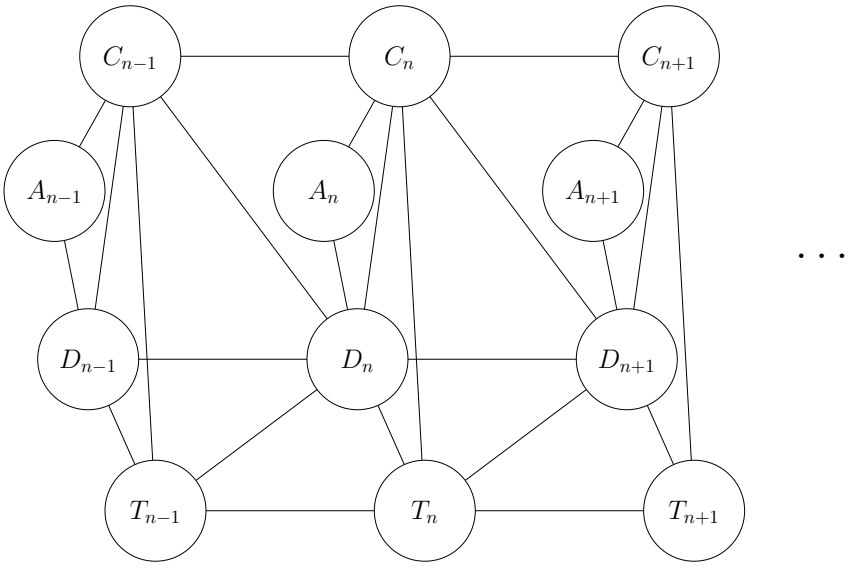


FIGURE 4.5 – Représentation graphique des dépendances entre les variables cachées du modèle HTCRF (conditionnellement à des observations non informatives, qui ne sont pas représentées).

avec les fonctions définies en (4.2) et en (4.3).

En revanche, le potentiel $\psi_{\mathrm{d}}^{\mathrm{HT}}$ représentant les contraintes de durées d'agrégats fait intervenir la variable de tempo. On a alors

$$\psi_{\mathrm{d}}^{\mathrm{HT}}(D_n, D_{n-1}, C_{n-1}, T_{n-1}) = \begin{cases} \rho_{\mathrm{d}}(D_{n-1}, C_{n-1}, T_{n-1}) & \text{si } D_n = 1 \\ 1 & \text{si } D_n = D_{n-1} \\ 0 & \text{sinon.} \end{cases} \tag{4.11}$$

avec la fonction définie en (4.7).

Enfin, le potentiel $\psi_{\mathrm{t}}^{\mathrm{HT}}$, contrôlant les transitions entre tempos, est déduit de (4.8) comme suit :

$$\psi_{\mathrm{t}}^{\mathrm{HT}}(T_n, T_{n-1}, D_n) = \begin{cases} \rho_{\mathrm{t}}(T_n, T_{n-1}) & \text{si } D_n = 1 \\ \mathbf{1}_{\{T_n = T_{n-1}\}} & \text{sinon.} \end{cases} \tag{4.12}$$

La fonction de transition du modèle HTCRF est alors donnée par :

$$\begin{aligned} \psi^{\mathrm{HT}}(X_n, X_{n-1}) = \psi_{\mathrm{t}}^{\mathrm{HT}}(T_n, T_{n-1}, D_n)\psi_{\mathrm{d}}^{\mathrm{HT}}(D_n, D_{n-1}, C_{n-1}, T_{n-1}) \\ \psi_{\mathrm{c}}^{\mathrm{HT}}(C_n, C_{n-1}, D_n)\psi_{\mathrm{a}}^{\mathrm{HT}}(A_n, D_n, C_n) \end{aligned} \tag{4.13}$$

et le modèle graphique correspondant est représenté figure 4.5.

4.2 Modèle d'observation

Comme indiqué dans la section 3.3, le potentiel donné par la fonction d'observation relie les étiquettes aux observations. Il est donc avantageux d'extraire de l'enregistrement

des observations qui caractérisent le mieux possible les étiquettes. Ces observations devraient donc refléter les informations correspondant aux variables aléatoires des étiquettes, à savoir l'agrégat joué (c'est-à-dire les notes présentes), la phase de l'agrégat, l'occupation et le tempo. Malheureusement, il est difficile d'imaginer une manière de caractériser la variable d'occupation à partir du signal enregistré. Nous utilisons donc trois vecteurs de caractéristiques, présentés plus bas, qui dépeignent les trois autres variables.

À chaque trame n sont associés un *vecteur de chroma* v_n, une valeur de *flux spectral* s_n et un vecteur de *tempogramme cyclique* g_n. Ces descripteurs sont donc calculés en utilisant le même pas d'avancement entre deux fenêtres d'analyse successives. Ce pas d'avancement, fixé à 20 ms, définit la résolution temporelle des alignements obtenus. Le vecteur d'observations complet s'écrit $y_n = (v_n, s_n, g_n)$.

Nous supposons alors que la fonction d'observation, présentée en section 3.3, est séparable en autant de facteurs. Pour une valeur x_n de la variable X_N, on écrit alors :

$$\phi(x_n, \mathbf{y}_{1:N}) = \phi_1\left(x_n, \mathbf{v}_{1:N}\right) \phi_2\left(x_n, \mathbf{s}_{1:N}\right) \phi_3\left(x_n, \mathbf{g}_{1:N}\right). \tag{4.14}$$

Cette hypothèse recouvre en particulier le cas où les différentes observations sont conditionnellement indépendantes sachant les étiquettes. Nous étudierons donc successivement les potentiels ϕ_i utilisés dans nos modèles.

4.2.1 Attributs d'agrégat

Afin de caractériser l'agrégat présent à chaque trame de l'enregistrement musical, nous utilisons une représentation en *vecteurs de chroma* (présenté en section 2.2.1). On rappelle qu'un vecteur de chroma est un vecteur à 12 composantes, dont chacune représente l'énergie des bandes de fréquences correspondant à une classe chromatique de la gamme musicale. Un exemple de chromagramme est représenté figure 4.6 Parmi les nombreuses méthodes existante pour le calcul de vecteurs de chroma, nous utilisons celle de Zhu et Kankanhalli [2006]. Ce choix est motivé par notre étude [Joder *et al.*, 2010b], qui montre que l'utilisation de cette représentation donne de bons résultats d'alignement, avec un modèle simple. Afin de rendre la fonction d'observation indépendante des dynamiques de la musique, les vecteurs de chroma obtenus sont normalisés de façon que leur somme soit égale à 1.

Comparaison de l'observation à des gabarits théoriques

Ces observations sont comparées à des *gabarits*, qui représentent des vecteurs de chroma-types correspondant aux agrégats. Chaque gabarit est construit de façon très simple, d'après la composition de l'agrégat associé. En effet, on crée un vecteur de chroma dont chacune des composantes est proportionnelle au nombre de notes de l'agrégat appartenant à la classe chromatique correspondante. Le gabarit est alors obtenu en normalisant ce vecteur et en le superposant à un vecteur « de bruit », dont toutes les composantes sont égales.

Plus formellement, soient c un agrégat et $\mathcal{J}_c = \dot{\mathcal{J}}_c \cup \breve{\mathcal{J}}_c$ l'ensemble des notes (attaquées et liées) de cet agrégat, comme défini section 2.1.2. Pour chaque note j de cet agrégat, $\mathrm{pc}(j)$

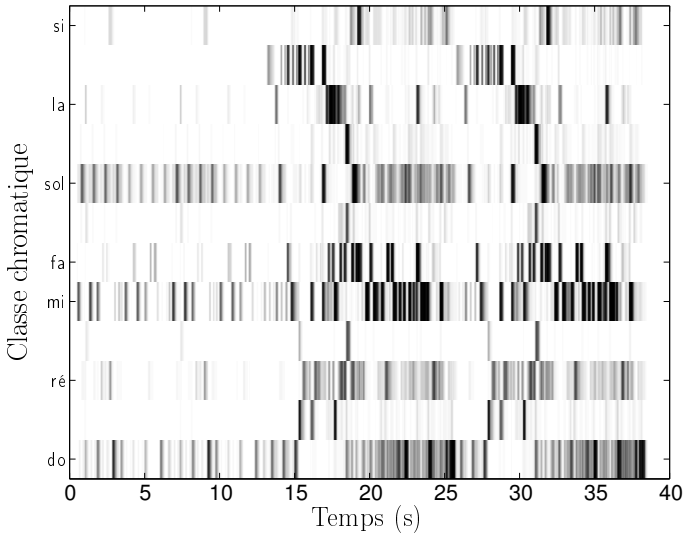


FIGURE 4.6 – Exemple de chromagramme extrait d'un enregistrement musical.

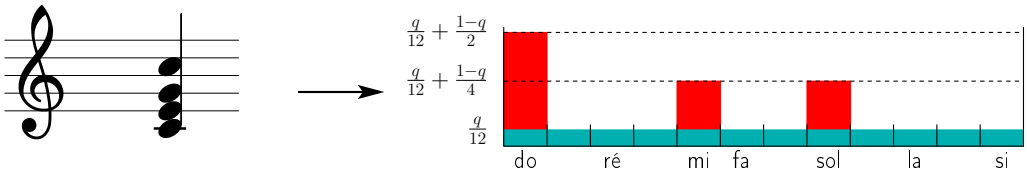


FIGURE 4.7 – Exemple de construction du gabarit théorique de chroma à partir d'un agrégat de 4 notes. Ici, on a $q = \frac{1}{2}$.

désigne l'index de la classe chromatique de cette note. Le gabarit théorique u_c associé à l'agrégat est défini par :

$$\forall i = \{1, \ldots, 12\}, \quad u_c(i) = \sum_{j \in \mathcal{J}} \frac{1-q}{\mathrm{card}(\mathcal{J})} \mathbf{1}_{\{i=\mathrm{pc}(j)\}} + \frac{q}{12}. \tag{4.15}$$

Le paramètre $q \in [0, 1[$ contrôle l'importance donnée au terme de bruit. La construction du gabarit est illustrée figure 4.7.

Notons que les gabarits sont normalisés de la même manière que les observations, de façon que leur somme soit égale à 1. On peut alors considérer ces vecteurs comme des distributions de probabilités sur les classes chromatiques et utiliser une distance probabiliste pour comparer une observation et un gabarit. Nous choisissons la divergence de Kullback-Leibler. L'attribut d'agrégat $f_1(c, v)$ est alors défini par :

$$f_1(c, v) = \sum_{i=1}^{I} v(i) \log \left(\frac{v(i)}{u_c(i)} \right). \tag{4.16}$$

Le terme de bruit de l'équation (4.15) peut aussi être interprété comme un terme de

« lissage ». Il permet en effet d'éviter les valeurs nulles du gabarits, qui entraineraient une valeur infinie de la divergence.

Intégration du voisinage

Comme indiqué en section 3.3.3, la fonction ϕ_1, reliant les observations de chroma aux étiquettes, peut faire intervenir des vecteurs d'observations issues de plusieurs trames de l'enregistrement, sans hypothèse d'indépendance. Le potentiel $\phi_1\left(x_n, \mathbf{v}_{1:N}\right)$ est alors formé en comparant l'étiquette x_n aux observations de chroma extraites d'un voisinage de la trame n.

Nous faisons tout d'abord l'hypothèse que l'interprétation est fidèle à la partition, dans le sens où les notes et le rythme sont conformes aux indications. Nous supposons de plus que le tempo peut être considéré comme constant sur une fenêtre temporelle courte (durant plusieurs trames). Sous ces deux hypothèses, la donnée d'une étiquette associée à une trame de l'enregistrement est suffisante pour définir les agrégats joués autour de la trame courante. En effet, les variables c_n et d_n définissent la position dans la partition et le tempo t_n permet d'extrapoler les positions (dans la partition) correspondant au voisinage de la trame n. On peut alors comparer terme à terme la séquence d'observations de ce voisinage et la séquence déduite de l'étiquette x_n.

Plus formellement, soit ν un entier tel que le tempo peut être considéré comme constant sur une fenêtre de $2\nu+1$ trames. Pour une étiquette $x_n = (c_n, d_n, a_n, t_n)$ correspondant à la trame n, on construit la séquence des $2\nu+1$ agrégats correspondant à une interprétation « exacte » de la partition au tempo t_n, autour de la position (c_n, d_n) —telle que la position dans la partition à la trame n soit (c_n, d_n). Une telle séquence est représentée figure 4.8. Soit $\bar{c}_{n-\nu}, \ldots, \bar{c}_{n+\nu}$ les gabarits correspondant à ces agrégats. Le potentiel d'agrégat ϕ_{c} est alors défini par :

$$\phi_1\left(x_n, \mathbf{v}\right) = \exp\left(-\sum_{k=-\nu}^{\nu} \mu_1^{(k)} f_1\Big(\bar{c}_{n+k}, v_{n+k}\Big)\right) \tag{4.17}$$

où les $\mu_1^{(k)}$ sont des paramètres contrôlant le poids donné aux différentes observations autour de la trame courante.

Intuitivement, on souhaite favoriser les observations temporellement proches de la trame courante, tout en restant symétrique entre le passé et le futur. La valeur de $\mu_1^{(k)}$ est donc une fonction décroissante de $|k|$. On choisit une fenêtre exponentielle de paramètre 25, ce qui correspond à une division du poids par 2 en environ 350 ms. Afin de donner encore plus d'importance à l'observation courante, on choisit de lui affecter un poids supplémentaire égal à la somme des paramètres de la fenêtre exponentielle. On note alors μ_1 le poids total donné aux observations de chroma. Les paramètres sont donc définis par :

$$\mu_1^{(k)} = \mu_1 \left(\frac{1}{2\sum\limits_{\kappa=-\nu}^{\nu} e^{-25|\kappa|}} e^{-25|k|} + \frac{1}{2}\mathbf{1}_{\{k=0\}}\right) \tag{4.18}$$

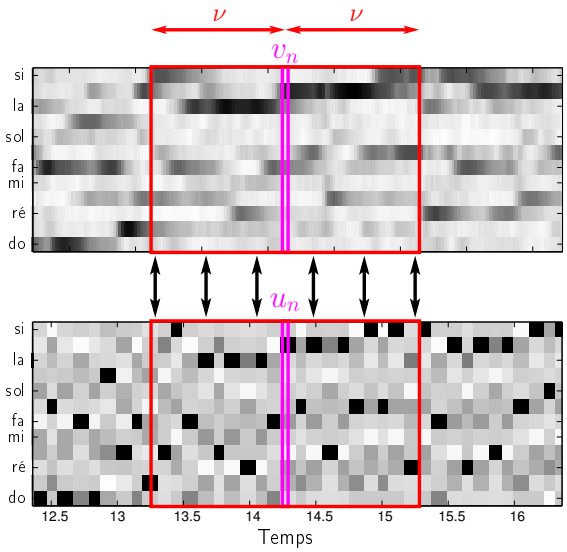


FIGURE 4.8 – Illustration du calcul du potentiel de chroma. Haut : la séquence d'observations autour de la trame n ; bas : les gabarits correspondant à une interprétation « mécanique » de la partition, au tempo constant t_n autour de la position (c_n, d_n). Sur cet exemple, t_n est un tempo plus rapide que l'enregistrement.

Notons que dans le cas général, la fonction d'observation proposée fait apparaitre des dépendances supplémentaires entre les variables C_n, D_n et T_n lorsque ces variables ne sont pas reliées. En revanche, dans le cas particulier où $\nu = 0$, c'est-à-dire où l'on prend en compte uniquement l'observation de chroma issue de la trame courante, la valeur du potentiel de chroma ne dépend ni de D_n, ni de T_n. Dans ce cas, aucune dépendance supplémentaire n'est impliqué par la fonction d'observation.

4.2.2 Attribut d'attaque

Afin de discriminer la phase d'attaque de la phase stationnaire de chaque agrégat, nous exploitons ce que l'on appelle une fonction de détection d'attaque. Cet attribut est calculé de manière simple à partir du *flux spectral* extrait par la méthode d'Alonso *et al.* [2005]. Le flux spectral est défini comme la somme des dérivées temporelles d'un spectre de puissance sur toutes les bandes de fréquences. La longueur des fenêtres d'analyse utilisées pour le calcul du spectre est ici 40 ms. Comme on le voit sur la figure 4.9 (haut), ce descripteur présente des pics, qui correspondent aux position d'attaques de notes. Afin de mettre en valeur ces pics, on calcule alors un seuil local en appliquant un filtre d'ordre de 67% et de longueur 200 ms aux valeurs du flux spectral (ces valeurs sont choisies de manière heuristique). Notre fonction de détection d'attaque, représentée figure 4.9 (bas), est alors obtenue en retranchant ce seuil au flux spectral. Nous notons s_n la valeur de cette fonction de détection issue de la fenêtre n. L'attribut d'attaque $f_2(a, s_n)$ est alors défini par :

$$f_2(a, s_n) = a s_n. \tag{4.19}$$

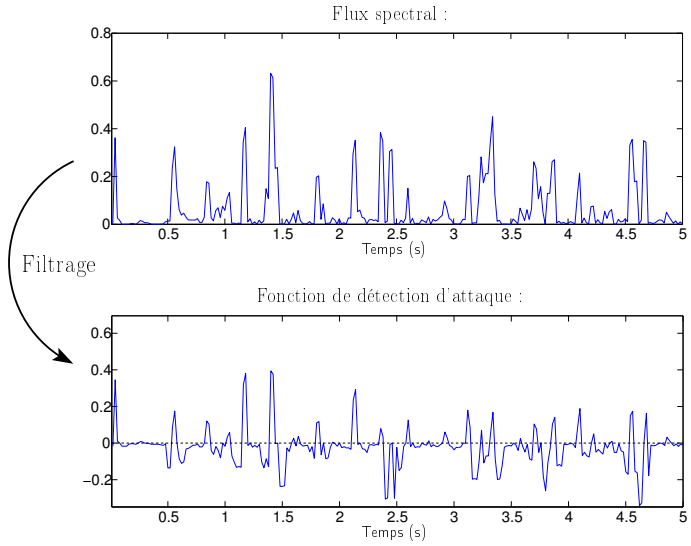


Figure 4.9 – Illustration de la construction de l'attribut d'attaque.

La variable de phase A étant binaire, cet attribut caractérise uniquement les phases d'attaques. Le potentiel ϕ_2 de l'équation (4.14) est alors défini par :

$$\phi_2(x_n, \mathbf{s}) = \exp\big(\mu_2 f_2(a, s_n)\big). \tag{4.20}$$

Le paramètre $\mu_2 \geq 0$ contrôle l'importance accordé à cet attribut. Notons que ce potentiel ne fait pas apparaitre de dépendance entre deux variables cachées, car il fait uniquement intervenir la variable de phase A_n.

4.2.3 Attribut de tempo

Le troisième attribut utilisé est le *tempogramme cyclique*. Cet attribut, introduit par Grosche *et al.* [2010] pour une tâche de structuration, fournit une représentation de niveau intermédiaire du tempo. Comme le *tempogramme* [Cemgil *et al.*, 2001], cette représentation calcule des « puissances » associées à certaines fréquences considérées comme des tempos potentiels. Cependant, de façon similaire au chromagramme, le tempogramme cyclique regroupe les tempos par classes d'octaves et intègre les puissances sur chaque classe, de façon à « replier » ces valeurs sur une seule octave. Ainsi les valeurs calculées sont égales pour deux tempos dont l'un est le double de l'autre, comme on le voit sur la figure 4.10. La représentation obtenue est donc robuste à certaines variations des motifs rythmiques (par exemple, une accentuation de toutes les doubles-croches où seulement des noires). Par construction, elle occasionne forcément des ambiguïtés d'octave. Néanmoins, on suppose que ces ambiguïtés sont levées grâce aux autres sources d'information (chroma et détection d'attaque).

Cet attribut est extrait à partir de l'autocorrélation du flux spectral, calculée sur une fenêtre glissante de 5 s centrée sur la trame courante, pour des décalages entre 200 ms et $3,2$ s. Ces décalages correspondent à des tempos entre 18,75 et 300 bpm (pulsations

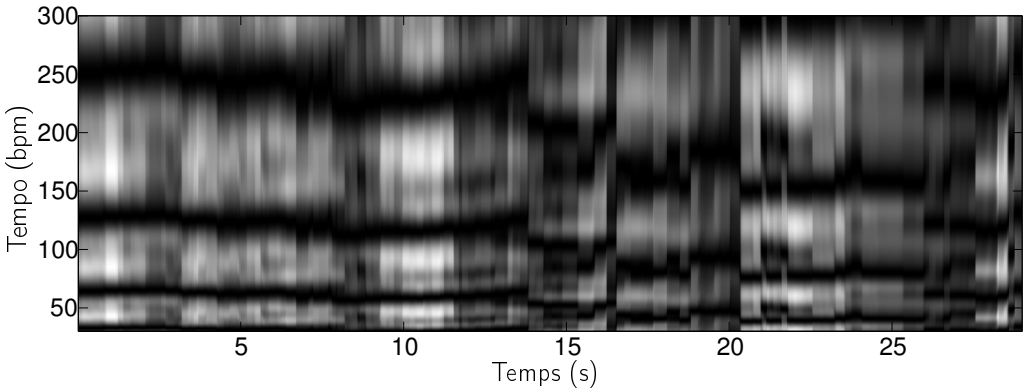


FIGURE 4.10 – Exemple d'attribut de tempo, calculé sur un court enregistrement de piano.

par seconde). Notons que les valeurs extrêmes ne sont pas utilisées comme tempos potentiels, mais peuvent apporter des informations supplémentaires correspondant aux périodes multiples ou sous-multiples de la pulsation.

Nous notons alors $g_n(\tau)$ la valeur de l'autocorrélation locale autour de la trame n correspondant au décalage τ. L'attribut de tempo $f_3(t, g_n)$ associé à la valeur t de la variable de tempo est la valeur du tempogramme cyclique à ce tempo, calculé en additionnant l'autocorrélation sur toute la classe d'octave :

$$f_3(t, g_n) = \sum_{k \in \mathbb{Z}} g_n(2^k t). \tag{4.21}$$

En pratique, cette somme est limitée aux décalages précisés plus haut. Le potentiel ϕ_3 est alors donné par la formule :

$$\phi_3(y_n, \mathbf{g}) = \exp\Big(\mu_3 f_3(t, g_n)\Big) \tag{4.22}$$

où le paramètre μ_3 contrôle le poids accordé à cet attribut. Remarquons que ce potentiel fait intervenir uniquement la variable de tempo, sans dépendance aux autres variables cachées.

4.3 Décodage au sens du maximum *a posteriori*

Les différentes variables introduites dans nos modèles sont récapitulées dans la table 4.1 et les structures des modèles sont représentés figures 4.11 et 4.12. Comme indiqué en section 4.2.1, notre fonction d'observation fait apparaitre des liaisons supplémentaires entre certaines variables lorsque le potentiel de chroma exploite le voisinage de la trame courante (c'est-à-dire si $\nu > 0$).

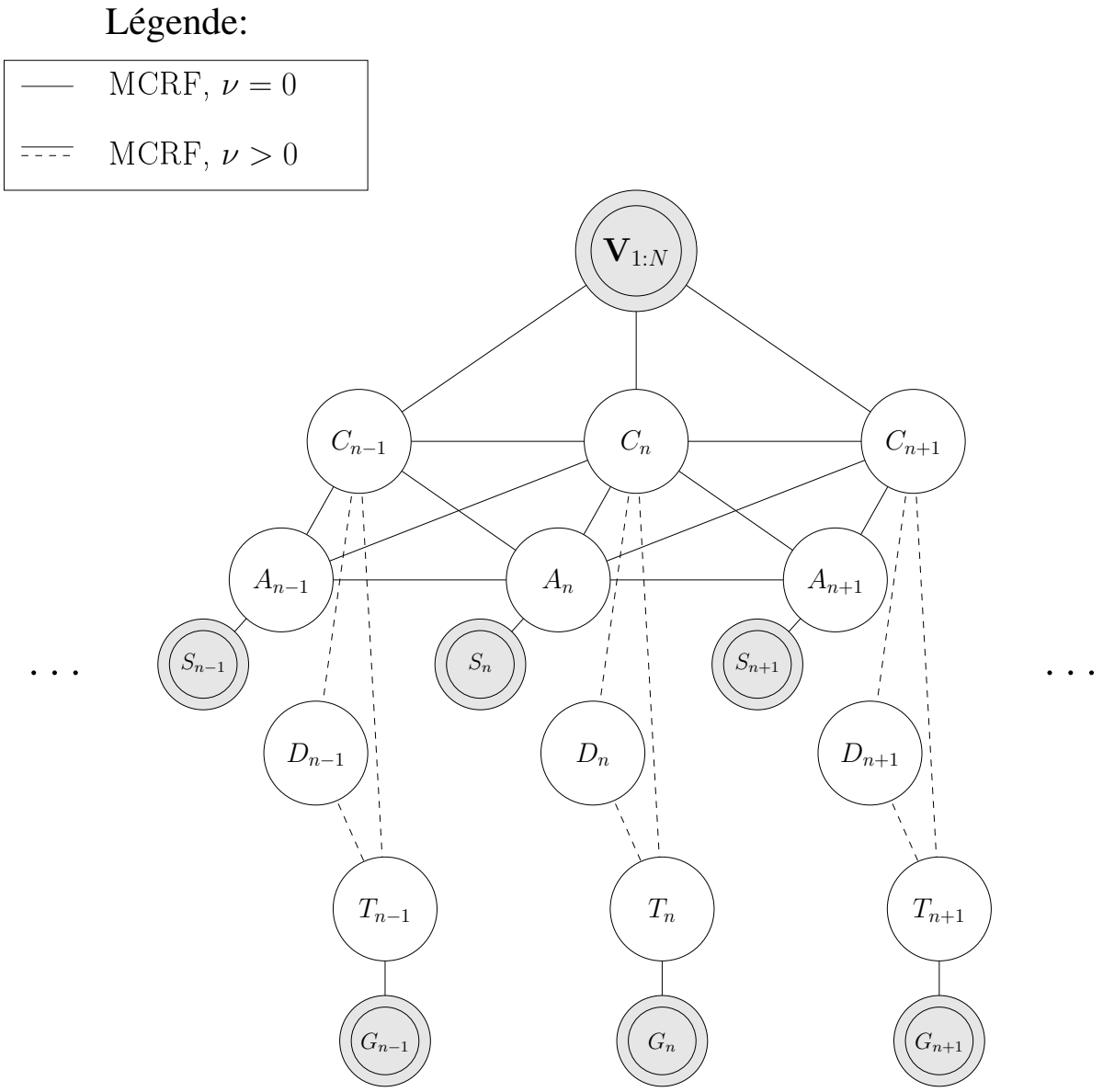


Figure 4.11 – Représentations graphiques du modèle MCRF. Les sommets doubles et grisés correspondent aux variables observées, qui conditionnent le modèle.

Variables cachées (étiquettes) :	
C_n	Agrégat
A_n	Phase
D_n	Occupation
T_n	Tempo
$X_n = (C_n, A_n, D_n, T_n)$	

Observations :	
V_n	Vecteur de chroma
S_n	Détecteur d'attaque
G_n	Tempogramme
$Y_n = (V_n, S_n, G_n)$	

Table 4.1 – Récapitulatif des variables utilisées dans les modèles CRF d'alignement.

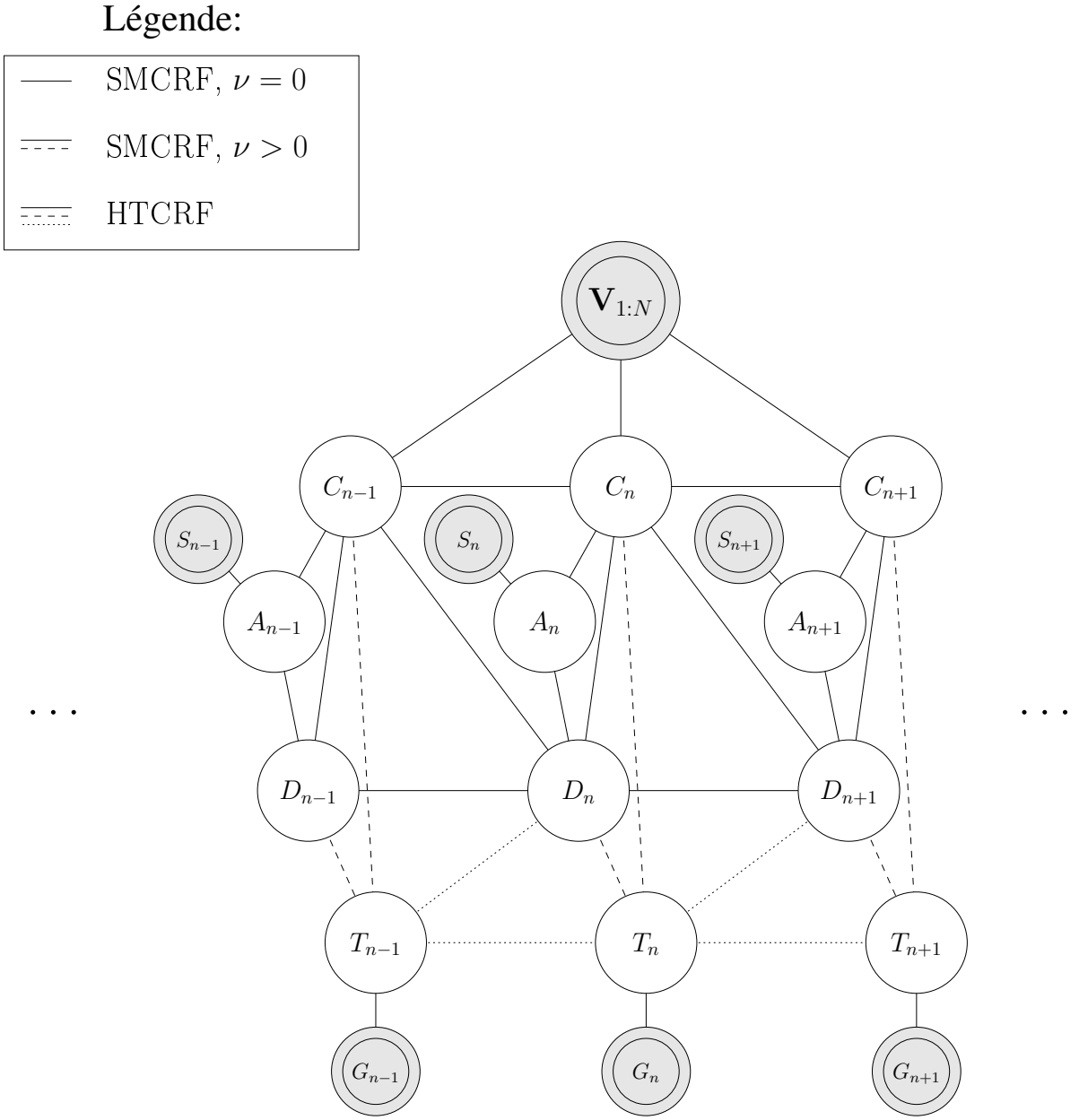


FIGURE 4.12 – Représentations graphiques des modèles SMCRF et HTCRF. Les sommets doubles et grisés correspondent aux variables observées, qui conditionnent le modèle.

4.3.1 Algorithme de Viterbi

Rappelons que le décodage de ces modèles est effectué selon le critère MAP défini en section 3.2.2. On recherche donc la séquence :

$$\hat{\mathbf{x}}_{1:N} = \arg\max_{\mathbf{x}_{1:N}} P(\mathbf{X}_{1:N} = \mathbf{x}_{1:N} | \mathbf{Y}_{1:N} = \mathbf{Y}_{1:N}) \tag{4.23}$$

$$= \arg\max_{\mathbf{x}_{1:N}} \phi(x_1, \mathbf{y}_{1:N}) \prod_{k=2}^{N} \psi(x_k, x_{k-1}) \phi(x_k, \mathbf{y}_{1:N}). \tag{4.24}$$

L'algorithme de Viterbi, permet de calculer cette séquence optimale. Cet algorithme s'appuie sur les valeurs

$$\hat{p}_n(x_n|\mathbf{y}_{1:N}) = \max_{\substack{\mathbf{X}_{1:n} \\ X_n = x_n}} \phi(X_1, \mathbf{y}_{1:N}) \prod_{k=2}^{n} \psi(X_k, X_{k-1}) \phi(X_k, \mathbf{y}_{1:N}). \tag{4.25}$$

En effet, ces valeurs peuvent être calculées itérativement, grâce à la relation :

$$\hat{p}_n(x_n|\mathbf{y}_{1:N}) = \max_{x_{n-1}} \left\{ \hat{p}_{n-1}(x_{n-1}|\mathbf{y}_{1:N}) \, \psi(x_n, x_{n-1}) \right\} \phi(x_n, \mathbf{y}_{1:N}). \tag{4.26}$$

On peut donc calculer la dernière étiquette de la séquence recherchée, par la formule

$$\hat{x}_N = \arg\max_{x_N} \hat{p}_N(x_N|\mathbf{y}_{1:N}). \tag{4.27}$$

On peut alors montrer que la séquence $\hat{\mathbf{x}}_{1:N}^{\text{MAP}}$ est celle obtenue par *retour en arrière* (*backtracking*), suivant à chaque trame l'étiquette réalisant le maximum de l'équation (4.26). En pratique, on peut garder en mémoire tous ces « précédents » pour accélérer le décodage.

L'algorithme nécessite donc des nombres de multiplications et de comparaisons tous deux de l'ordre de $\mathcal{O}(E_X Q_X N)$, où Q_X est le cardinal de l'ensemble des étiquettes possibles et E_X est le nombre moyen de « transitions entrantes » par étiquette, dans l'automate des étiquettes. Il est à noter que ces ordres de grandeur ne tiennent pas compte de la complexité du calcul des fonctions d'observation et de transition. La complexité en espace de l'algorithme est de $\mathcal{O}(Q_X N)$, car l'étape de retour en arrière nécessite le stockage des « précédents optimaux » de chaque couple étiquette-trame (ou alors de tous les scores partiels $\hat{p}_n(x_n|\mathbf{y}_{1:N})$).

Il est à noter que des stratégies d'« élagage » comme la *recherche par faisceaux* [Ortmanns *et al.*, 1996] peuvent être utilisées afin de réaliser un décodage approché avec une complexité moindre, en espace comme en nombre de comparaisons. Ces méthodes explorent seulement une (petite) partie des étiquettes à chaque trame. Nous proposons une stratégie originale d'élagage adaptée à l'alignement audio/partition, qui sera détaillée en section 6. L'élagage sera presque systématiquement utilisé, car il améliore la vitesse de l'alignement, sans dégrader les résultats en pratique.

Néanmoins, nous présentons dans cette section une étude de la complexité du décodage optimal des modèles présentés plus haut. En effet, les complexités relatives des différents systèmes restent inchangées avec l'élagage.

4.3.2 Complexité du décodage du modèle HTCRF

Pour le HTCRF, le nombre d'étiquettes possibles est inférieur à $Q_C Q_A Q_D Q_T$ où Q_C est le nombre d'agrégats de la partition, $Q_A = 2$ est le nombre de phases possibles, Q_D est le nombre de valeurs possibles de la variable d'occupation et Q_T est le nombre de tempos considérés.

On peut encore affiner cette borne. En effet, puisque les contraintes de l'équation (4.2), représentées figure 4.3 limitent les valeurs possibles du couple (A, D). Pour chaque agrégat c, le nombre de ces valeurs possibles est inférieur à $D_{\max}(c) + 1$, où $D_{\max}(c)$ désigne la durée maximale possible de l'agrégat. De plus, nous limitons la durée maximale de chaque agrégat à la valeur $\ell_c T_{\mathrm{M}}$, correspondant à la longueur théorique de l'agrégat au plus lent tempo considéré (rappelons que ℓ_c est la durée en pulsations de l'agrégat c). Le cardinal de l'ensemble des étiquettes possibles est alors $Q_X = (\pounds T_{\mathrm{M}} + Q_C) Q_T$. La complexité en espace du décodage du modèle HTCRF est donc $\mathcal{O}\big((\pounds T_{\mathrm{M}} + Q_C) Q_T N\big)$, où $\pounds$ est la longueur totale en pulsations du morceau de musique.

D'après la fonction de transition de ce modèle, définie en section 4.1.3, les transitions entre étiquettes sont très contraintes et chaque état de l'automate des étiquettes comporte très peu d'arêtes entrantes. En effet, comme représenté figure 4.3, à l'intérieur d'un agrégat (hors états initiaux) seuls les nœuds correspondant à la valeur $D = 3$ possèdent plusieurs transitions entrantes ; les autres en ont une unique. En revanche, les Q_T états de début d'un agrégat c sont reliés à chacune des étiquette de l'agrégat précédent, dont le nombre est de l'ordre de $\ell_{c-1} T_{\mathrm{M}} Q_T$. Comme l'agrégat c comporte $\ell_c T_{\mathrm{M}} Q_T$ étiquettes, le nombre moyen de transitions entrantes par étiquette peut être assimilé à $E_X = Q_T$. Les nombres de multiplications et de comparaisons nécessaires au décodage sont donc tous deux en $\mathcal{O}\big(\pounds T_{\mathrm{M}} Q_T^2 N\big)$.

4.3.3 Modèle SMCRF : Complexité

Modèle SMCRF

Pour le modèle SMCRF, une modification peut être apportée à cette forme de l'algorithme de Viterbi. Cela est rendu possible par le fait que la fonction de transition ne dépend pas de la variable de tempo T_n. Le problème de maximisation de l'équation (4.24) peut alors être factorisé, en traitant cette variable séparément. On rappelle que l'on a défini $x_n = (c_n, a_n, d_n, t_n)$. On pose ici :

$$
\begin{aligned}
x'_n &= (c_n, a_n, d_n) && (4.28)\\
\psi'(x'_n, x'_{n-1}) &= \psi(x_n, x_{n-1}) && (4.29)\\
\hat{\phi}'\left(x'_n, \mathbf{y}_{1:N}\right) &= \max_{t_n} \phi\left(x_n, \mathbf{y}_{1:N}\right). && (4.30)
\end{aligned}
$$

On a alors :

$$\begin{aligned} P(\mathbf{X}_{1:N} = \hat{\mathbf{x}}_{1:N} | \mathbf{Y}_{1:N} = \mathbf{y}_{1:N}) &= \max_{\mathbf{x}_{1:N}} \phi(x_1, \mathbf{y}_{1:N}) \prod_{k=2}^{N} \psi(x_k, x_{k-1}) \phi(x_k, \mathbf{y}_{1:N}) \\ &= \max_{\mathbf{x}'_{1:N}} \max_{t_1} \phi(x_1, \mathbf{y}_{1:N}) \prod_{k=2}^{N} \max_{t_k} \psi(x_k, x_{k-1}) \phi(x_k, \mathbf{y}_{1:N}) \\ &= \max_{\mathbf{x}'_{1:N}} \hat{\phi}'(x'_1, \mathbf{y}_{1:N}) \prod_{k=2}^{N} \psi'(x'_k, x'_{k-1}) \hat{\phi}'(x'_k, \mathbf{y}_{1:N}). \quad (4.31) \end{aligned}$$

On peut donc appliquer l'algorithme de Viterbi exposé plus haut au « sous-problème » du décodage de la séquence optimale $\hat{x}'_{1:N}$, représentant les variables d'agrégat, de phase et d'occupation. La séquence optimale des tempos peut ensuite être déduite en gardant en mémoire les valeurs qui réalisent les maxima calculés à l'équation (4.30), correspondant à la séquence des autres variables décodées. Mais en pratique, dans ce modèle on utilise les variables de tempo uniquement comme des contraintes implicites sur les variables d'agrégat et on ne s'intéresse pas aux valeurs décodées du tempo.

La complexité en espace du décodage du modèle SMCRF est alors réduite, puisque ce décodage stocke alors une unique valeur par triplet (C, A, D), au lieu de considérer toutes les possibilités de tempo. Elle est donc de l'ordre de $\mathcal{O}\big((\pounds T_{\mathrm{M}} + Q_C)N\big)$.

Le nombre de multiplications et de comparaisons diminue de même, puisqu'après les maximisations « locales » sur la variable de tempo, le décodage prend en compte uniquement les autres variables. Ces complexités deviennent donc chacune de l'ordre de $\mathcal{O}\big(\pounds T_{\mathrm{M}} Q_T N\big)$.

Cas $\nu = 0$

Lorsqu'on fixe le paramètre de voisinage ν, défini en section 4.2.1 à la valeur nulle, des liaisons disparaissent dans le modèle graphique (figure 4.12) car le potentiel de chroma ne dépend plus du tempo. Les variables de tempo deviennent alors rigoureusement indépendantes des autres variables cachées. On peut donc utiliser deux fois l'algorithme de Viterbi pour décoder indépendamment les variables C, A, D d'une part, et la variable T d'autre part. Cependant, comme on ne s'intéresse pas aux valeurs du tempo, ce deuxième décodage n'est tout simplement pas effectué.

Dans ce cas, la complexité en espace est toujours de l'ordre de $\mathcal{O}\big((\pounds T_{\mathrm{M}} + Q_C)N\big)$. Par contre, le nombre de comparaisons et de multiplications nécessaire pour le décodage qui nous intéresse diminue, puisqu'il ne nécessite plus aucune maximisation sur la variable de tempo. L'ordre de grandeur est donc ramené à $\mathcal{O}\big(\pounds T_{\mathrm{M}} N\big)$ pour chacune de ces deux complexités.

Modèle	Complexité du décodage		
	Espace	Comparaisons	Multiplications
HTCRF	$\pounds T_{\mathrm{M}} Q_T N$	$\pounds T_{\mathrm{M}} Q_T^2 N$	$\pounds T_{\mathrm{M}} Q_T^2 N$
SMCRF ($\nu>0$)	$\pounds T_{\mathrm{M}} N$	$\pounds T_{\mathrm{M}} Q_T N$	$\pounds T_{\mathrm{M}} Q_T N$
SMCRF ($\nu=0$)	$\pounds T_{\mathrm{M}} N$	$\pounds T_{\mathrm{M}} N$	$\pounds T_{\mathrm{M}} N$
MCRF ($\nu>0$)	$Q_C Q_A N$	$Q_C Q_A N$	$\pounds T_{\mathrm{M}} Q_T Q_A N$
MCRF ($\nu=0$)	$Q_C Q_A N$	$Q_C Q_A N$	$Q_C Q_A N$

TABLE 4.2 – Complexités théoriques de décodage des différents modèles proposés.
Définition des symboles :

Q_C	nombre d'agrégats du morceau
Q_A	nombre de phases possibles (en pratique $Q_A = 2$)
Q_T	nombre de valeurs de tempo considérées
T_{M}	valeur maximale du tempo
$\pounds$	longueur de la partition, en pulsations
N	longueur de l'enregistrement, en nombre de trames

4.3.4 Complexité du modèle MCRF

Cas général

Dans le modèle MCRF, la fonction de transition ne dépend ni de la variable de tempo, ni même de l'occupation. Le décodage peut donc être encore simplifié en séparant ces deux variables dans le décodage. Comme précédemment, on définit pour chaque trame n

$$x''_n = (c_n, a_n) \tag{4.32}$$
$$\psi''(x''_n, x''_{n-1}) = \psi(x_n, x_{n-1}) \tag{4.33}$$
$$\hat{\phi}''\left(x''_n, \mathbf{y}_{1:N}\right) = \max_{(d_n,t_n)} \phi\left(x_n, \mathbf{y}_{1:N}\right). \tag{4.34}$$

Il en résulte :

$$P(\mathbf{X}_{1:N} = \hat{\mathbf{x}}_{1:N}|\mathbf{Y}_{1:N}) = \max_{\mathbf{x}''_{1:N}} \hat{\phi}''(x''_1, \mathbf{Y}_{1:N}) \prod_{k=2}^{N} \psi''(x''_k, x''_{k-1}) \hat{\phi}''(x''_k, \mathbf{Y}_{1:N}). \tag{4.35}$$

L'algorithme de Viterbi permet alors de décoder $\hat{\mathbf{x}}''_{1:N}$, c'est-à-dire les variables d'agrégat et de phase correspondant à la séquence d'étiquette optimale, avec une complexité en mémoire de $\mathcal{O}(Q_A Q_C N)$. L'occupation et le tempo peuvent se déduire des deux variables décodées, en stockant les valeurs réalisant les maxima de l'équation (4.34). Les maximisations « locales » de l'équation (4.34) deviennent dominantes dans le nombre de comparaisons. Il est alors de l'ordre de $\mathcal{O}\big(\pounds T_{\mathrm{M}} Q_T Q_A N\big)$, ce qui n'est pas inférieur au modèle précédent. En revanche, le nombre de multiplication est moindre que celui du SMCRF puisqu'il est alors en $\mathcal{O}(Q_A Q_C N)$.

Ces complexités de décodages sont récapitulées dans le tableau 4.2.

Cas $\nu = 0$

Pour le cas $\nu = 0$, les variables d'occupation et de tempo deviennent indépendantes des deux autres variables cachées, comme représenté sur la figure 4.11. De ce fait, ces variables ne sont plus utiles au décodage de la séquence optimale d'agrégats. Dans ce cas, tous nos ordres de complexité sont réduit à $\mathcal{O}\big(Q_A Q_C N\big)$.

4.4 Expériences

Dans cette section, nous présentons les précisions d'alignement obtenues par les systèmes détaillés plus haut et représentés sur les figures 4.11 et 4.12. Ces expériences sont menées sur la base de données présentée en section 2.5.

4.4.1 Paramètres utilisés

Afin d'utiliser en pratique nos systèmes, il faut préciser un certain nombre de paramètres. Tout d'abord, l'ensemble $\mathcal{T}$ des tempos possibles est fixé à

$$\begin{aligned}\mathcal{T} = \{&28, 30, 34, 40, 48, 56, 64, 72, 80, 88, 96, 104,\\ &112, 120, 132, 146, 160, 176, 192, 208, 224, 240\}\end{aligned} \tag{4.36}$$

ou ces valeurs sont données en bpm (pulsations par minutes). Cet ensemble a été choisi en sélectionnant une valeur sur deux, d'après les tempos proposés par un métronome [2].

Nous testons deux versions de chacune des structures, utilisant deux différentes valeurs du paramètre de voisinage ν. La première version correspond au cas $\nu = 0$, où seule l'observation courante du chroma est prise en compte dans la fonction d'observation. Nous utilisons aussi la valeur $\nu = 50$, ce qui correspond à une fenêtre de voisinage de 1 s autour de la trame courante.

Enfin, les paramètres γ des fonctions de transition ainsi que les paramètres μ des fonctions d'observations doivent être fixés à des valeurs entrainant les meilleurs alignements possibles. Dans l'idéal, on pourrait imaginer une estimation de tous ces paramètres d'après un critère comme le maximum de vraisemblance sur notre ensemble d'apprentissage. Malheureusement, le coût en mémoire et en temps de calcul d'un apprentissage complet est prohibitif. Au chapitre 5 sera envisagée une estimation optimale de certains des paramètres. Pour les expériences présentes, les paramètres sont fixés par une méthode de recherche sur une grille de valeurs (appelée aussi *grid search*). Un petit nombre de valeurs est testé pour chaque paramètre et les alignements sont calculés avec toutes les combinaisons de paramètres obtenus. Enfin, la combinaison engendrant le meilleur alignement sur notre base d'apprentissage est sélectionnée.

Il est à noter qu'avec les modèles CRF présentés, le décodage est invariant si l'on multiplie tous les paramètres γ et μ par une même valeur. En effet, les fonctions de transition

2. En réalité, le tempo le plus lent du métronome est de 40 pulsations/s, mais nous avons ajouté des valeurs afin de pouvoir modéliser les morceaux très lents « à la croche » et les arrêts sur certains agrégats (points d'orgues)

Paramètre (section de définition)	Valeurs
μ_2 (4.2.2)	$\{0, \frac{1}{100}, \mathbf{\frac{1}{10}}, 1^*, 10, 100\}$
μ_3 (4.2.3)	$\{0, \frac{1}{100}, \frac{1}{10}, \mathbf{1}, 10^*, 100\}$
γ_l (4.1.2)	$\{\frac{1}{1000}, \frac{1}{500}, \mathbf{\frac{1}{200}}, \frac{1}{100}, \frac{1}{50}, \frac{1}{20}\}$
γ_d (4.1.3)	$\{5, \mathbf{10}, 20, 50, 100\}$
γ_t (4.1.3)	$\{10, 20, 50, \mathbf{100}, 200, 500, 1000\}$

TABLE 4.3 – Valeurs des paramètres considérées par la recherche sur grille. Les valeurs en gras sont sélectionnées pour tous les modèles, sauf en présence d'un astérisque, qui indique les paramètres retenus pour le modèle MCRF.

sont de la forme

$$\psi(X_n, X_{n-1}) = \exp\Big(\sum_k \gamma_k f_k^{(\mathrm{t})}(X_n, X_{n-1})\Big) \tag{4.37}$$

où les $f_k^{(\mathrm{t})}(X_n, X_{n-1})$ sont des *attributs* dépendant uniquement des transitions considérées (éventuellement égaux à $-\infty$). De même, la fonction d'observation est de la forme

$$\phi(X_n, \mathbf{Y}_{1:N}) = \exp\Big(\sum_{k=1}^{3} \mu_k f_k^{(\mathrm{o})}(X_n, \mathbf{Y}_{1:N})\Big) \tag{4.38}$$

où les $f_k^{(\mathrm{o})}(X_n, \mathbf{Y}_{1:N})$ ne dépendent pas des paramètres γ_k ou μ_k. De ce fait, la solution du problème de maximisation exprimé en (4.24) n'est pas modifiée si tous les paramètres considérés sont multipliés par la même constante.

Nous choisissons donc de fixer la contrainte $\mu_1 = 1$, afin de lever ce facteur d'ambiguïté. L'ensemble des valeurs considérées par la recherche sur grille de valeur (*grid search*) est compilé dans le tableau 4.3. Il est notable que l'importance accordée aux attributs de détection d'attaque et de tempo est plus grande dans le modèle markovien que dans les deux autres. Cela peut être expliqué par le fait que ces attributs sont utilisés pour compenser les contraintes temporelles plus lâches du premier système.

4.4.2 Résultats et discussion

Les résultats obtenus par les systèmes testés sont présentés dans le tableau 4.4. Les taux d'alignement mesurés sur le corpus RWC-pop avec le seuil $\theta = 50$ ms sont donnés seulement à titre informatif en raison de la potentielle imprécision des annotations de ce corpus. Pour tous les taux d'alignement, les rayons des intervalles de confiance à 95% sont inférieurs à $0,4$ %. Il est par contre difficile d'estimer les intervalles de confiance pour les autres mesures, car l'absence de seuil de tolérance rend ces métriques sensibles à l'imperfection des annotations.

Il est intéressant de noter que toutes les mesures suivent les mêmes tendances. Cela n'est pas étonnant, puisqu'en pratique la mesure de classification et les mesures de segmentation sont très corrélées. Cependant, cela indique que les différences entre les systèmes proposés ne sont pas dépendantes de l'application visée.

Corpus MAPS :

Modèle	MCRF		SMCRF		HTCRF	
ν	0	50	0	50	0	50
TAMP ($\theta = 300$ ms)	93.1%	94.9%	97.7%	97.9%	**99.3%**	**99.3%**
TAMP ($\theta = 100$ ms)	80.8%	85.7%	90.9%	93.5%	**97.8%**	97.7%
TAMP ($\theta = 50$ ms)	64.0%	69.2%	76.5%	83.3%	91.2%	**91.4%**
IMP	73 ms	62 ms	45 ms	38 ms	**25 ms**	**25 ms**
CCMP	31.3%	27.7%	21.4%	17.7%	13.4%	**13.2%**

Corpus RWC-pop :

Modèle	MCRF		SMCRF		HTCRF	
ν	0	50	0	50	0	50
TAMP ($\theta = 300$ ms)	85.2%	88.0%	94.2%	93.9%	**99.2%**	**99.2%**
TAMP ($\theta = 100$ ms)	58.3%	65.4%	75.3%	79.2%	**94.4%**	**94.4%**
TAMP ($\theta = 50$ ms)	39.9%	43.2%	51.7%	57.1%	**76.3%**	76.0%
IMP	132 ms	114 ms	83 ms	76 ms	**39 ms**	**39 ms**
CCMP	60.5%	53.7%	45.7%	42.4%	**32.2%**	32.3%

Table 4.4 – Résultats obtenus par les systèmes testés. On rappelle que TAMP désigne le Taux d'Alignement Moyen Pondéré, IMP est l'Imprécision Moyenne Pondérée et CCMP est le Cout de Classification Moyen Pondéré. Ces mesures sont définies en section 2.4.

Comparaison des corpus

Tous les systèmes testés obtiennent de meilleurs scores sur le corpus MAPS que sur le corpus RWC-pop. Trois raisons peuvent être avancées pour expliquer cette observation. Premièrement, le corpus RWC-pop, contrairement à MAPS, contient dans presque tous ses morceaux des sons percussifs (en particulier de la batterie) et d'autres sons non harmoniques (voix parlé ou chuchotée, applaudissements, bruits ou effets sonores...). La présence de ces sons affecte les observations de chroma, mais aussi de détection d'attaque.

Deuxièmement, les morceaux de RWC-pop contiennent souvent de nombreux instruments, dont les niveaux de mixage (c'est-à-dire les volumes sonores relatifs) peuvent être très divers. Ainsi, certains instruments d'« arrière-plan » sont quelquefois à peine audibles, ce qui peut rendre très difficile la détection des changements de notes. De plus, la partition ne fait pas forcément apparaitre les niveaux relatifs des instruments et les gabarits de chroma utilisés (voir section 4.2.1) ne reflètent alors pas toujours les observations réelles.

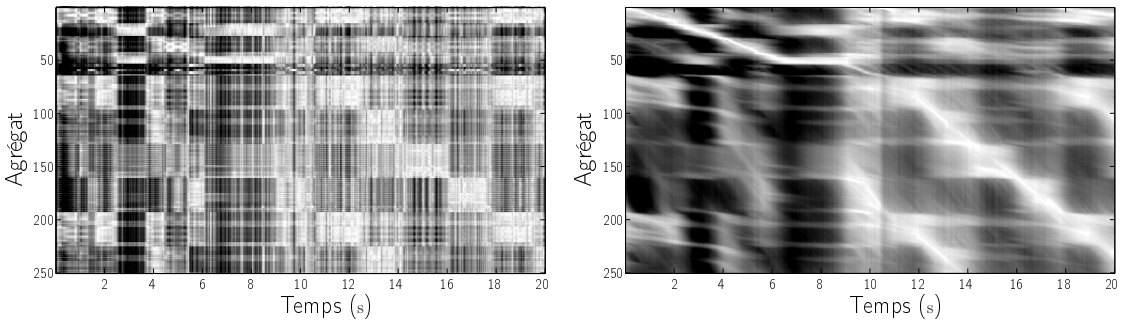
Enfin, la voix chantée, qui est prédominante dans pratiquement toutes les chansons pop, présente souvent des effets de vibrato et de glissando ou encore des imprécisions de justesse qui ne sont pas décrites par la partition. Toutes ces difficultés sont absentes du corpus MAPS puisqu'il contient exclusivement du piano solo.

Une autre observation peut interpeler le lecteur : en effet, même avec un taux d'alignement de plus de 99 % (mesuré avec le seuil de tolérance $\theta = 100$ ms), le meilleur système classe correctement seulement deux tiers des trames du corpus RWC-pop. La principale raison tient en la relative imprécision de l'annotation par rapport au contenu musical : en effet, les morceaux de ce corpus ont en général un tempo élevé et un grand nombre d'instruments. Les agrégats se succèdent donc à un rythme rapide et de ce fait, même un très léger décalage de l'annotation peut faire baisser notablement le cout de classification. Par exemple, si la durée moyenne d'un agrégat est de 200 ms (10 trames), un décalage de 40 ms de l'annotation résultera en un cout de classification de 20 % pour l'alignement parfait. Les valeurs absolues de cette métrique ne sont donc pas toujours représentatives de la qualité de l'alignement et il est préférable de s'attacher aux différences entre les scores obtenus.

Comparaisons des modèles de durée

D'après tous les indicateurs utilisés, la précision augmente avec la complexité du modèle de durée. En effet, les résultats du modèle semi-markovien sont systématiquement meilleurs que ceux obtenus par le système MCRF et le modèle à tempo caché produit les alignement les plus précis, avec ou sans prise en compte des observations issues du voisinage de la trame courante.

Par exemple, le MCRF atteint un taux d'alignement de $94,9$ % pour un seuil de tolérance de 300 ms sur le corpus MAPS. L'ajout du modèle de durée est efficace puisque le modèle SMCRF obtient un score maximal de $97,9$ %. Enfin, la prise en compte d'une variable de tempo dans les contraintes de durée permet au système HTCRF d'augmenter encore la précision de l'alignement, jusqu'à un taux d'alignement de $99,3$ %.



(a) **(b)**

Figure 4.13 – Contributions de l'observation de chroma courante **(a)** et du voisinage **(b)** dans la fonction d'observation $\hat{\phi}''$ (voir section 4.3.4). Le blanc indique une valeur élevée.

Intérêt de la prise en compte du voisinage

Une autre observation est que la considération des observations voisines dans la fonction d'observation permet d'accroitre la finesse des alignements fournis par les deux premiers systèmes (MCRF et SMCRF). Pour le modèle markovien, la prise en compte du voisinage correspond à un modèle implicite de tempo et de durée. Cela occasionne des augmentations absolues du taux d'alignement de respectivement $1,8$ % et $2,8$ % sur les corpus MAPS et RWC-pop. Le gain est encore plus important aux niveaux de précisions plus fins : en effet, des améliorations d'au moins $3,5$ points sont observées pour les seuils de tolérance de 100 ms et 50 ms. De fait, l'imprécision moyenne pondérée est réduite de respectivement 8 et 16 ms

La figure 4.13 compare la contribution des observations « instantanées » de chroma et celle des observations « intégrées » sur la fenêtre de voisinage complète, dans le cas d'un morceau de la base RWC-pop. Dans cet exemple, la fonction d'observation intégrée accentue clairement certains points, qui correspondent au chemin d'alignement cherché ainsi qu'à des répétitions des mêmes séquences d'agrégats.

Nous avons vu que pour le modèle SMCRF, fixer $\nu = 0$ équivaut à supprimer la variable de tempo dans le décodage de la séquence d'agrégats. L'ajout de la dépendance entre les agrégats et le tempo, par la considération des observations voisines, augmente la précision des alignements. Ainsi l'imprécision moyenne pondérée diminue de 7 ms pour les deux corpus. De même, presque tous les taux d'alignement sont améliorés, surtout aux niveaux de précision fins. Par exemple sur le corpus MAPS, les augmentations absolues mesurées avec des seuils de 100 et 50 ms sont respectivement de 3,9 et $5,4$ points.

Une exception est visible pour le taux d'alignement du corpus RWC-pop avec le seuil $\theta = 300$ ms. En effet, le score obtenu avec prise en compte du voisinage est inférieur (bien que de façon non significative) à celui du système « instantané » ($93,9$ % contre $94,2$ %). La principale cause de cette perte de précision est que dans certains morceaux, le système ne décode pas la séquence d'agrégats cherchée, mais une répétition, c'est-à-dire une séquence d'agrégats équivalents ou très proches au sens de la mesure de comparaison des chromas (ici la divergence de Kullback-Leibler).

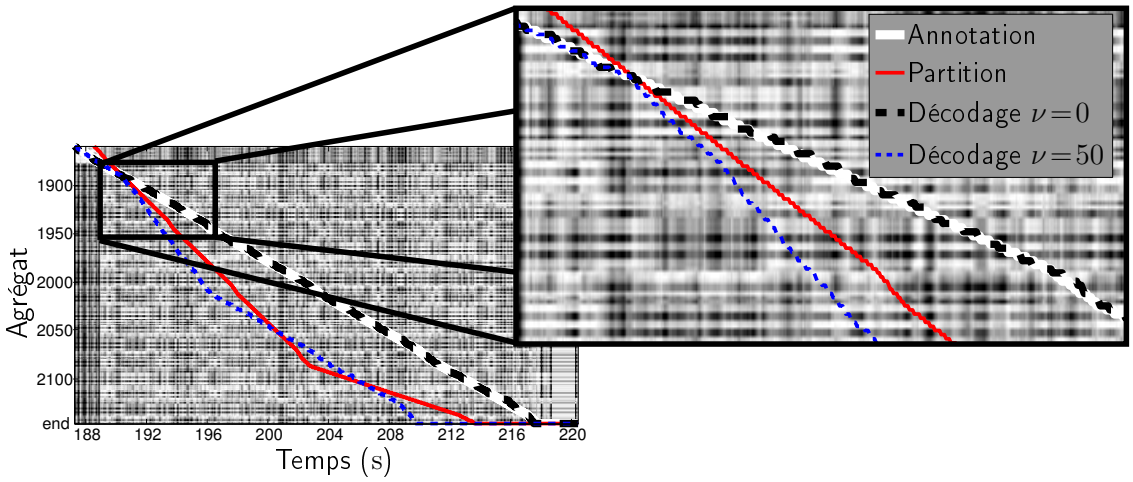


FIGURE 4.14 – Exemple du phénomène de « mauvaise répétition » . Les chemins d'alignement décodés avec les deux systèmes SMCRF sont comparés au chemin annoté et à celui indiqué par la partition. La fonction d'observation « instantanée » représentée ici est fortement bruitée par la présence de percussions. L'utilisation du voisinage mène alors à une dégradation du chemin décodé, à cause des indications temporelles erronées de la partition.

À titre d'exemple, la figure 4.14 représente la section finale d'un morceau de la base RWC-pop. Dans cet exemple, les percussions sont très présentes et affectent les vecteurs de chroma extraits, qui ne correspondent alors pas vraiment aux gabarits théoriques utilisés dans la fonction d'observation. Cette fonction d'observation est donc très bruitée et ne met pas en valeur le chemin d'alignement annoté. Dans le cas d'un bruit important, l'intégration de ces vecteurs de chroma sur un voisinage temporel ne rend pas forcément l'alignement plus précis, mais peut au contraire « lisser » les valeurs de la fonction d'observation. Le système exploitant le voisinage est alors plus fortement influencé par le modèle temporel, qui favorise ici des durées plus courtes que les durées réelles. On voit donc le chemin décodé suivre un tempo plus rapide, avant de retrouver une trajectoire parallèle à l'annotation, qui correspond à une autre instance du même motif musical dans la partition. Notons que dans cet exemple, les ambiguïtés entre les différentes répétitions ne sont pas levées par le retour en arrière de l'algorithme de Viterbi. En effet, l'état de *fin*, qui modélise les silences et les sons non harmoniques, présente une valeur élevée de la fonction d'observation en raison du fort niveau des percussions. Et de fait, la plupart de ces problèmes de « mauvaise répétition » se trouvent à la toute fin des morceaux.

Ce phénomène est assez limité car il se produit avec la combinaison d'observations de chroma bruitées et d'*a priori* de durées erronés. De plus, les agrégats détectés sont en pratique souvent équivalents aux agrégats réels, ce qui explique pourquoi le cout de classification est amélioré par rapport au cas sans intégration du voisinage (42,4% contre 45,7%).

Grâce au modèle de tempo explicite, le HTCRF obtient des résultats encore meilleurs

que les autres systèmes. Ce modèle est robuste au problème de « mauvaise répétition », car les *a priori* de durées sont alors beaucoup plus fiables. Cependant, dans la plupart des cas, l'exploitation du voisinage n'améliore pas significativement les performances d'alignement. Cela s'explique par le fait que l'information fournie par les observations du voisinage concerne les valeurs du tempo, qui est ici modélisé explicitement.

Remarque : retour sur les pénalités de durée

Le lecteur peut s'interroger sur la différence de forme de la pénalité de durée des modèles SMCRF et HTCRF. En effet, dans la première, définie par l'équation (4.5), l'écart-type de la gaussienne utilisée est constant, alors que dans la fonction de pénalité du modèle à tempo caché, définie en (4.7), l'écart-type est proportionnel à la longueur attendue de l'agrégat.

Nous avons mené une expérience comparant les deux méthodes sur l'ensemble d'apprentissage. Il s'est avéré que la stratégie « proportionnelle » entrainait un meilleur taux d'alignement ($96,6$ % pour le seuil $\theta = 100$ ms, contre $94,6$ % avec un écart-type constant). Cela tend à confirmer l'intuition selon laquelle les variations de durées sont proportionnelles aux longueurs de notes.

En revanche, les résultats du modèle SMCRF se sont révélés différents. Le taux d'alignement a augmenté sur le corpus MAPS (de $90,8$ % à $94,3$ %), où les durées indiquées par la partition sont proches de celles de l'enregistrement, mais il s'est effondré sur le corpus RWC-pop (de $71,2$ % à $64,7$ %), qui comporte d'importants changements de tempo. En effet, la pénalité de durée devient alors très rigide quand la durée attendue d'un agrégat est courte. De ce fait, les pénalités de déviations de durée sont comparativement plus importantes lorsque le tempo de l'interprétation est plus lent que celui de la partition que dans le cas inverse. Ce problème n'est pas présent dans le modèle à tempo caché, puisqu'on considère plusieurs hypothèses de tempo. En revanche, en présence de changements de tempo importants dans l'interprétation, un écart-type constant se révèle plus efficace pour le modèle semi-markovien.

4.5 Conclusion

Dans ce chapitre, nous avons détaillé la conception de modèles de champs aléatoires conditionnels pour l'alignement de musique sur partition. Nous nous sommes attachés en particulier à la modélisation temporelle du signal musical, à travers différentes structures de la fonction de transition. Ces trois structures, markovienne (MCRF), semi-markovienne (SMCRF) et à tempo caché (HTCRF) permettent une précision temporelle de plus en plus fine, qui s'accompagne d'une complexité croissante. La fonction d'observation utilisée ici exploite trois types d'information, liés au contenu harmonique, aux attaques de notes et au tempo local de l'enregistrement. Nous proposons en outre une prise en compte des informations temporelles dans la fonction d'observation, grâce à une intégration des observations extraites de tout un voisinage de chaque trame audio. Cela permet une modélisation implicite du tempo et des durées d'agrégat, qui se traduit par une amélioration de la précision d'alignement pour les modèles MCRF et SMCRF.

Malgré la précision globalement satisfaisante des systèmes proposées, nous avons vu que certains problèmes pouvaient subsister, en particulier pour le modèle semi-markovien, lorsque les *a priori* temporels étaient incorrects. Dans ce cas, une meilleure prise en compte des observations devrait limiter ces imprécisions. C'est là l'objet du prochain chapitre qui est dédié à l'étude et l'optimisation du modèle d'observation.

Chapitre 5

Optimisation du Modèle d'observation

Dans ce chapitre, nous revenons sur l'attribut d'agrégat utilisé dans le chapitre précédent, caractérisant la correspondance instantanée entre un point de la partition et une trame de l'enregistrement sur la base de l'harmonie, c'est-à-dire les notes jouées. Nous formulons le calcul de ces attributs à partir d'une transformation linéaire de la représentation symbolique vers le domaine des observations acoustiques. Cette formulation présente l'avantage d'être applicable à n'importe quelle représentation temps-fréquence de l'audio. De ce fait, nous pouvons comparer l'efficacité de plusieurs représentations usuelles ainsi que différentes mesures de distance entre observations et agrégats, pour un alignement par CRF.

La forme utilisée nous permet en outre d'effectuer un apprentissage de la transformation employée, afin d'élaborer de nouveaux attributs plus pertinents pour notre tâche. Cet apprentissage est supervisé, c'est-à-dire qu'il fait appel aux annotations disponibles sur la base d'apprentissage. Nous proposons tout d'abord un critère d'attache aux données, le *minimum de divergence*, pour une estimation de la matrice de transformation optimale. Puis, nous explorons une stratégie discriminative en adoptant le critère du *maximum de vraisemblance* pour l'apprentissage de cette transformation. Les expériences menées mettent en valeur les intérêts d'un tel apprentissage. En effet, bien qu'un phénomène de surapprentissage puisse être observé sur certains morceaux, la précision des alignements est bien souvent supérieure à ceux obtenus avec les approches heuristiques usuelles.

5.1 Formulation générale de l'attribut d'agrégat

L'attribut d'agrégat utilisé au chapitre précédent a été défini en 4.2.1 comme une divergence entre le vecteur de chroma observé et un gabarit théorique correspondant à l'agrégat considéré. Dans cette section, nous proposons une formulation plus générale du calcul de cet attribut, dans le but de réaliser un apprentissage des attributs optimaux.

5.1.1 Définition

L'attribut d'agrégat a pour but de quantifier la correspondance entre une observation extraite de l'audio et un agrégat de la partition. Afin de calculer cet attribut, nous définissons tout d'abord le *vecteur de notes*, qui constitue une représentation vectorielle de la partition. Dans l'hypothèse (peu réductrice) que l'ambitus d'un morceau ne dépasse pas le registre du piano moderne (du la^{-2} au do^{7}), nous numérotons les hauteurs de notes possibles de 1 à 88, suivant la gamme chromatique. Le vecteur de notes h_c d'un agrégat c est alors le vecteur dont chaque composante est égale au nombre de notes de la hauteur correspondante. Notons qu'une composante supplémentaire est ajoutée afin de rendre compte des agrégats ne contenant pas de note (correspondant au silence où à des sons sans hauteur définie). Cette dernière composante est égale à 1 si et seulement si toutes les autres sont nulles. La dimension de cette représentation en vecteur de notes est donc $J = 89$.

Pour une représentation temps-fréquence quelconque, v_n désigne maintenant le vecteur d'observation extrait de la trame n. Comme dans le cas du chapitre précédent, l'attribut d'agrégat $f_1(c, v_n)$ est alors obtenu en comparant l'observation à un gabarit correspondant à l'agrégat c. Nous supposons alors que le gabarit associé à un agrégat est la superposition des gabarits correspondants aux notes qui le composent. Cette approximation contient en fait deux hypothèses simplificatrice. La première est que les gabarits sont additifs et la seconde est que toutes les notes d'un agrégat contribuent de façon égale au gabarit de celui-ci. Ce gabarit peut alors être calculé par une application linéaire à partir du vecteur h_c. La forme générale de l'attribut est alors :

$$f_1(c, v_n) = -D\left(v_n, \mathbf{W}h_c\right), \tag{5.1}$$

où $D(\cdot, \cdot)$ est une fonction mesurant la dissemblance entre deux vecteurs (une distance, par exemple) et $\mathbf{W}$ est une matrice de dimension $I \times J$, où I est la dimension des vecteurs d'observation. Cette matrice peut être interprétée comme une transformation du domaine symbolique (les vecteurs de notes) vers le domaine des observations. Les colonnes de $\mathbf{W}$ constituent alors les gabarits des notes individuelles et le vecteur $u_c = \mathbf{W}h_c$ est le gabarit associé à l'agrégat c.

On peut observer que les gabarits utilisés dans le chapitre précédent, définis à l'équation (4.15), constituent un cas particulier de cette formulation générale. En effet, ils correspondent au choix de la divergence de Kullback-Leibler comme mesure de dissimilarité et à la matrice définie par :

$$\mathbf{W}_{i,j} = (1 - q)\mathbf{1}_{\{i=\text{pc}(j)\}} + \frac{q}{I}. \tag{5.2}$$

On rappelle que $\text{pc}(j)$ est la classe chromatique de la note j et que q est un paramètre contrôlant l'importance du terme de bruit dans le gabarit.

5.1.2 Lien avec un modèle génératif

La formulation proposée dans l'équation (5.1) est est fortement liée à certains modèles génératifs utilisés pour une factorisation en matrice non négative [Lee et Seung, 1999]. Nous détaillons ici comme cette formulation peut être retrouvé à partir du modèle d'observation de Virtanen *et al.* [2008].

Ce modèle postule que chaque vecteur d'observation est la superposition de variables aléatoires indépendantes, correspondant chacune à une des notes jouées. De plus, chaque composante de ces variables est supposée distribuée suivant une loi de Poisson [1], indépendante des autres composantes. Si on pose $\mathbf{W}_{i,j}$ comme paramètre de la loi de probabilité conditionnelle de la i-ième composante d'un vecteur d'observation, sachant qu'une unique note j est jouée. Cette loi s'écrit alors :

$$P\Big(V(i) = v(i)|j\Big) = e^{-\mathbf{W}_{i,j}} \frac{(\mathbf{W}_{i,j})^{v(i)}}{\Gamma\Big(v(i)+1\Big)}, \tag{5.3}$$

où Γ désigne la fonction gamma, interpolant la fonction factorielle. $\mathbf{W}_{i,j}$ s'interprète donc comme le paramètre (qui est aussi l'espérance) de cette loi de Poisson.

On peut alors en déduire les probabilités des vecteurs d'observation lorsqu'un agrégat quelconque est joué. En effet, on suppose qu'ils sont formés par la somme de variables aléatoires indépendantes correspondant aux notes de cet agrégat. Or, on rappelle qu'une somme de variables de Poisson indépendantes est encore distribuée suivant une loi de Poisson, dont le paramètre est la somme des paramètres des lois individuelles. Chaque composante $v(i)$ d'une observation v, sachant l'agrégat joué c, suit donc une loi de Poisson dont le paramètre, noté $u_c(i)$, est la somme des paramètres $\mathbf{W}_{i,j}$ correspondant aux notes de l'agrégat. Ce paramètre peut s'exprimer à l'aide du vecteur de notes h_c, par la formule

$$u_c(i) = \sum_{j=1}^{J} \mathbf{W}_{i,j} h_c(j), \tag{5.4}$$

puisque l'agrégat c contient un nombre $h_c(j)$ de notes de hauteur j.

Par hypothèse d'indépendance, la probabilité conditionnelle globale d'un vecteur d'observation, sachant l'agrégat c, est le produit des probabilités de chaque composante $v(i)$. On peut donc écrire :

$$\begin{aligned} P(V = v|c) &= \prod_{i=1}^{I} e^{-u_c(i)} \frac{u_c(i)^{v(i)}}{\Gamma\big(v(i)+1\big)} \\ &= \exp\left\{ \sum_{i=1}^{I} v(i) \log\big(u_c(i)\big) - u_c(i) - \log\Gamma\big(v(i)+1\big) \right\} \\ &= \exp\left\{ -\left(\sum_{i=1}^{I} v(i) \log \frac{v(i)}{u_c(i)} - v(i) + u_c(i) \right) + Z(v) \right\} \\ &= \exp\big\{ -D_{\mathrm{KL}}\big(v \big\| u_c\big) + Z(v) \big\} \end{aligned} \tag{5.5}$$

1. Une loi de Poisson est une loi de probabilité discrète dépendant d'un paramètre λ, dont la fonction de masse s'écrit $P(k) = e^{-\lambda}\frac{\lambda^k}{k!}$. Il est probablement plus intuitif de considérer les valeurs de puissance spectrale comme des variables aléatoires continues. Néanmoins, il est tout de même possible d'utiliser cette loi en considérant que les observations sont quantifiées, en raison notamment de la précision finie des ordinateurs.

où

$$Z(v) = \sum_{i=1}^{I} v(i) \log\big(v(i)\big) - v(i) - \log\Gamma\big(v(i)+1\big) \tag{5.6}$$

est un facteur dépendant uniquement de l'observation v. D_{KL} est la divergence de Kullback-Leibler généralisée, définie par

$$D_{\mathrm{KL}}\left(v\|u\right) = \sum_{i=1}^{I} v(i) \log\left(\frac{v(i)}{u(i)}\right) - v(i) + u(i). \tag{5.7}$$

Avec ce choix de distance particulier, on a donc la relation :

$$P(V = v|c) \propto e^{f_1(c,v)}. \tag{5.8}$$

Comme on l'a vu en section 3.3.2, on peut alors construire un modèle CRF équivalent à ce modèle génératif en posant

$$\phi(v|c) = e^{f_1(c,v)}. \tag{5.9}$$

La formulation proposée dans l'équation (5.1) peut alors être considérée comme une généralisation d'un tel modèle génératif.

5.1.3 Distances utilisées

En théorie, une fonction de distance [2] quelconque peut être utilisée pour calculer l'attribut d'agrégat. Dans ce travail, nous nous focalisons sur différentes versions de la divergence de Kullback-Leibler. Cette divergence présente en effet l'avantage d'être interprétable comme dérivant d'un modèle génératif, comme on vient de le voir. D'autres types de divergences, comme la distance *cosinus* ou la divergence d'Itakura-Saito ont en effet été abordées dans des tests préliminaires, sans donner de meilleurs résultats. La première version a déjà été présentée à l'équation (5.7). Elle sera désignée par « KL1 ». Cela correspond à la forme utilisée dans le chapitre précédent [3]. Comme cette divergence n'est pas symétrique, nous testons aussi une deuxième version, appelée « KL2 » et dont l'expression est :

$$D_{\mathrm{KL}}\left(u\|v\right) = \sum_{i=1}^{I} u(i) \log\left(\frac{u(i)}{v(i)}\right) - u(i) + v(i). \tag{5.10}$$

Enfin, la dernière distance considérée est la divergence de Kullback-Leibler symétrisée, désignée par « KLs » :

$$D_{\mathrm{KLs}}\left(v,u\right) = D_{\mathrm{KL}}\left(v\|u\right) + D_{\mathrm{KL}}\left(u\|v\right). \tag{5.11}$$

Notons qu'afin de rendre cet attribut robuste aux dynamiques d'intensité globale, les vecteurs d'observations ainsi que les vecteurs de notes sont normalisés.

2. Comme indiqué plus haut, la fonction $D(\cdot,\cdot)$ est une *mesure de dissimilarité*, qui peut ne pas vérifier les propriétés d'une distance au sens mathématique.

3. L'équation (4.16) emploie la divergence de Kullback-Leibler originale (non généralisée). Cependant, les deux formes sont équivalentes dans le cas où les deux vecteurs comparés sont normalisés.

5.2 Représentations et attributs courants

Cette section présente les différentes représentations de l'audio considérées dans nos expériences et détaille le calcul des attributs usuels correspondant. Ces attributs peuvent s'exprimer dans le cadre présenté plus haut, grâce à des matrices $\mathbf{W}$ construites de manière heuristique. Un récapitulatif des représentations utilisées est présenté dans la table 5.1.

5.2.1 Chromagramme

La transformation canonique pour la représentation en vecteurs de chroma est celle définie dans l'équation (5.2). Dans nos expériences, nous comparons deux méthodes différentes de calculs du chromagramme. La première est celle de Zhu et Kankanhalli [2006], déjà utilisée au chapitre précédent. Elle sera désignée par CGZ (pour chromagramme de Zhu). La seconde représentation est le chromagramme proposé par Müller [2007] (appelé CGM).

La première méthode exploite le module d'une transformée à Q constant (CQT) [Brown, 1991]. Cependant, afin de maintenir une précision temporelle acceptable, les valeurs correspondant aux fréquences inférieures à 100 Hz ne sont pas calculées. La longueur de la plus grande fenêtre d'analyse est alors d'environ 170 ms. Nous limitons de même la plus haute fréquence à 4 kHz, considérant que les hautes fréquences sont principalement dominées par les percussions.

La représentation CGM intègre les énergies instantanées à la sortie d'un banc de filtres elliptiques dont les fréquences centrales correspondent aux notes de la gamme chromatique tempérée. Ici, les 88 bandes comprises entre le la^{-2} (27,5 Hz) et le do^{7} (4186 Hz) sont prises en compte.

5.2.2 Semigramme

On rappelle que la représentation en semigramme, définie section 2.2.1, est calculée comme un spectre de puissance dont l'échelle fréquentielle est logarithmique et dont les bandes de fréquences sont centrées sur les notes de la gamme musicale tempérée. Deux méthodes de calcul sont encore considérées pour cette représentation, exploitant respectivement une transformée à Q constant et le banc de filtres déjà évoqué. Comme précédemment, la première représentation (appelée SGQ) est limitée à la bande de fréquences 100 Hz–4 kHz.

Dans le cas des représentations en semigramme, les composantes non nulles du gabarit d'une note j correspondent aux harmoniques de cette note. Le nombre d'harmoniques considérées et les poids qui leurs sont affectés sont par contre des paramètres de la modélisation. Par exemple, Müller *et al.* [2004] prennent en compte les trois premières harmoniques qui sont pondérées de façon homogène, tandis que Montecchio et Orio [2009] considèrent toutes les harmoniques possibles mais leur affectent des poids décroissants.

Pour nos expériences, nous choisissons un gabarit binaire modélisant uniquement les deux premières harmoniques de chaque note. Ce gabarit est normalisé et superposé à un terme « de bruit » (ou de lissage). La matrice $\mathbf{W}$ s'écrit alors :

$$\mathbf{W}_{i,j} = \frac{(1-q)}{2}\left(\delta_{i,j} + \delta_{i,j+12}\right) + \frac{q}{I}. \tag{5.12}$$

Acronyme	Signification
SP	Spectre de puissance
SGF	Semigramme par banc de filtres
SGQ	Semigramme par CQT
CGM	Chromagramme de Müller
CGZ	Chromagramme de Zhu

Table 5.1 – Récapitulatif des représentations de l'audio considérées.

Notons que pour les notes de la dernière octave (la plus haute), une seule composante est active. La valeur affectée à cette composante est donc le double de celle des autres notes, afin de maintenir la normalisation des vecteurs.

5.2.3 Spectrogramme

La dernière représentation utilisée, appelée spectrogramme (SP) est le spectre de puissance issu d'une transformée de Fourier à court terme. Dans nos expériences, cette transformée de Fourier est calculée sur 2048 points à partir de fenêtres de 100 ms. Pour limiter les problèmes de résolution en basses fréquences, seules les bandes de fréquence supérieures à 100 Hz sont prises en compte. De même, nous limitons la plus haute fréquence considérée à 4 kHz, afin de réduire le bruit engendré par les percussions.

Pour cette représentation, nous considérons les gabarits utilisés par Raphael [2006] et Cont [2010]. Dans ces travaux, une note est représentée par un ensemble de gaussiennes centrées sur les différentes harmoniques de cette note. Plus formellement, soit $\mathrm{b}(j)$ la bande de fréquence du spectrogramme correspondant à la fréquence fondamentale de la note j. Nous considérons alors les K premières harmoniques de cette note, situées aux bandes de fréquences multiples de $\mathrm{b}(j)$ (dans notre cas, nous fixons $k = 5$). La matrice de transformation s'écrit alors

$$\mathbf{W}_{i,j} = (1-q)\sum_{k=1}^{K} w_k \mathcal{N}\left(i; k\,b(j), \sigma^2_{j,k}\right) + \frac{q}{I}, \tag{5.13}$$

où $\mathcal{N}(\,\cdot\,;\mu,\sigma^2)$ désigne la loi normale [4] de moyenne μ et de variance σ^2. Dans nos expériences, les poids w_k affectés aux harmoniques sont proportionnels à $1/k^2$ et sont normalisés de façon que leur somme soient égale à 1. Les paramètres $\sigma^2_{j,k}$ de « largeur de bande » des composantes harmoniques sont fixés à 30 cent (30 % d'un demi-ton), soit $\sigma^2_{j,k} = 2^{\frac{30}{1200}} \times k\,b(j)$.

4. L'expression de la loi normale est :

$$\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

5.2.4 Réglage du paramètre de bruit

Dans les transformations présentées plus haut, nous avons choisi de fixer tous les paramètres de construction des gabarits, sauf le paramètre q contrôlant l'importance du terme « de bruit ». Comme dans le chapitre précédent, ce paramètre est estimé par recherche sur une grille de valeurs. Des alignements sont calculés sur la base d'apprentissage avec différentes valeurs de ce paramètre et la valeur occasionnant les meilleurs taux d'alignement est sélectionnée. Pour cet « apprentissage », le système d'alignement le plus simple possible est utilisé, en l'occurrence le modèle MCRF sans intégration du voisinage et n'exploitant ni les attributs de phase, ni les attributs de tempo (les paramètres μ_2 et μ_3 sont égaux à 0). Nous nommerons ce système MCRF0. Ce modèle est choisi car il n'exploite aucune information de durée des agrégats. Les alignements sont donc principalement influencés par les observations, ce qui permet de mettre en valeur les différences entre les attributs. Les valeurs de q sont sélectionnées parmi l'ensemble :

$$\mathcal{Q} = \{0, \frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \frac{4}{10}, \frac{5}{10}, \frac{6}{10}, \frac{7}{10}, \frac{8}{10}, \frac{9}{10}\}. \tag{5.14}$$

5.2.5 Résultats d'alignement

Afin de comparer les attributs heuristiques issus des 5 différentes représentations et des 3 distances, nous calculons les alignements sur les deux corpus déjà présentés, avec le système MCRF0. Les taux d'alignement obtenus sont présentés figure 5.1.

Comparaison des distances

Il est intéressant de noter que, pour toutes les représentations testées, la divergence KL1 donne de meilleurs alignements que la version KL2 sur le corpus MAPS, alors que le contraire est observé sur le corpus RWC-pop. Cela s'explique par les différences de comportement de ces deux mesures de dissimilarité, déjà évoquées par Oudre [2010]. Rappelons l'expression de la distance KL1 entre une observation v et un gabarit théorique u :

$$D_{\mathrm{KL}}\left(v \| u\right) = \sum_{i=1}^{I} v(i) \log\left(\frac{v(i)}{u(i)}\right) - v(i) + u(i). \tag{5.15}$$

Dans cette version, le terme en logarithme de l'équation (5.15) est grand lorsqu'une composante $u(i)$ du gabarit est petite devant la valeur de l'observation $v(i)$. De ce fait, cette version pénalise les gabarits « ne contenant pas toutes les fréquences observées ». Réciproquement, la distance KL2 est grande lorsque le gabarit « contient des fréquences non observées ». Intuitivement (et de façon schématique), la distance KL1 pénalise donc les notes « en trop » des observations, tandis que la version KL2 pénalise les notes « manquantes » par rapport au gabarit.

Or, les enregistrements du corpus MAPS contiennent uniquement des morceaux joués au piano, dont la polyphonie (le nombre de notes jouées simultanément) est limitée, et enregistrés dans de bonnes conditions de bruit. Les observations extraites comportent donc de nombreuses valeurs faibles et l'attribut utilisant la distance KL2 présente alors un biais

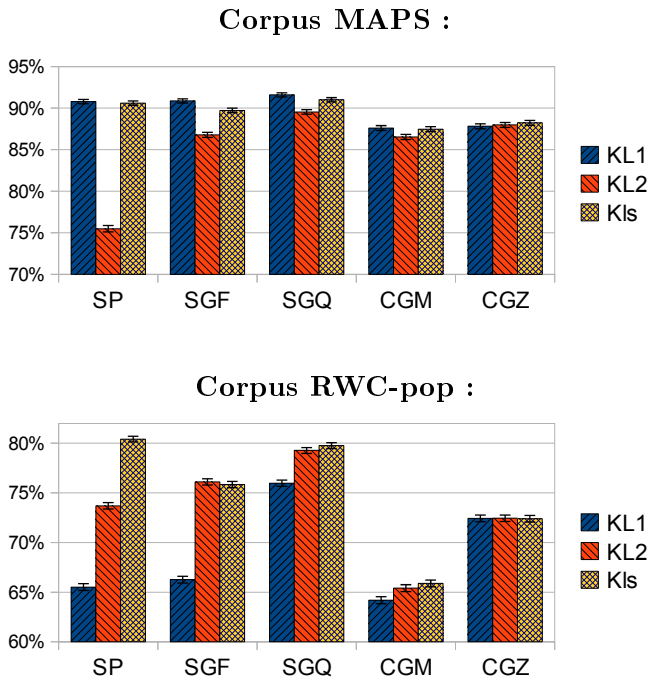


Figure 5.1 – Taux d'alignement moyens pondérés obtenus avec le modèle MCRF0 exploitant les différents attributs testés et leurs intervalles de confiance à 95%. Le seuil de tolérance est ici $\theta = 300$ ms.

en faveur des agrégats contenant peu de notes. Dans le corpus RWC-pop au contraire, on observe fréquemment des sons qui ne correspondent pas aux agrégats de la partition, en particulier à cause de la présence de percussions et des intonations quelquefois imprécises des chanteurs. La figure 5.2 illustre ces différences entre les contenus spectraux des deux corpus. La distance KL1 est alors biaisée en faveur des agrégats contenant les plus grands nombres de notes.

La distance de Kullback-Leibler symétrisée constitue alors un bon compromis, car elle ne présente pas ces biais (ou de façon moins marquée). De fait, les résultats indiquent que sur les deux corpus, les performances obtenues avec cette distance sont au même niveau (et quelquefois meilleures) que la meilleure des deux autres distances. Nous conserverons donc cette distance pour la suite de nos expériences.

Comparaison des représentations

Au vu des résultats, il est possible de mettre en valeur les différences entre les représentations considérées. Les deux représentations les plus efficaces sont ici le spectre de puissance (SP) et le semigramme SGQ, qui ont des performances similaires en moyenne sur les deux corpus.

Les deux chromagrammes (CGM et CGZ) sont moins performants que les autres représentations. Cela s'explique par la perte de l'information d'octave. Néanmoins, cette

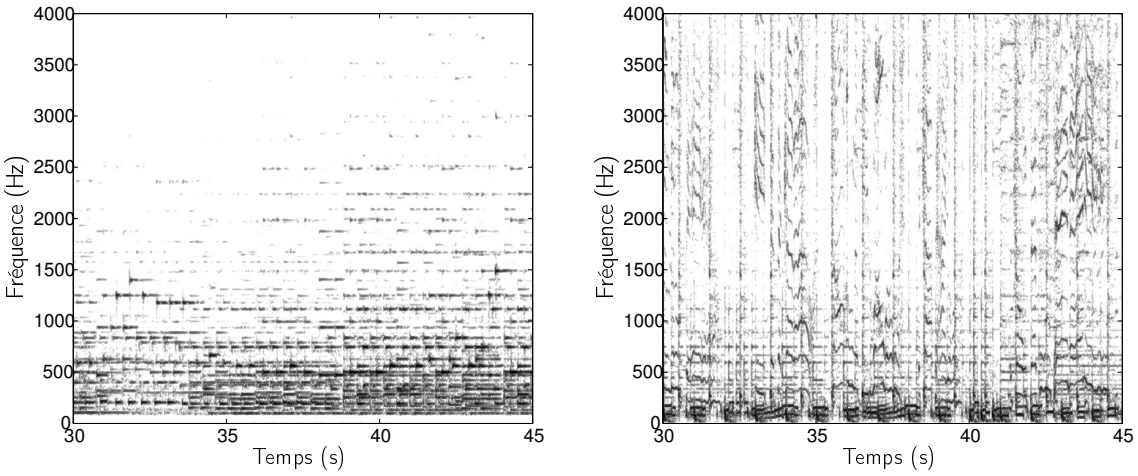


FIGURE 5.2 – Deux extraits de spectrogrammes représentatifs des enregistrements du corpus MAPS (à gauche) et du corpus RWC-pop (à droite).

invariance par rapport à l'octave peut être une caractéristique intéressante pour la prise en compte de partitions imparfaites.

On peut aussi observer que les représentations exploitant un banc de filtre (SGF et CGM) obtiennent des scores significativement plus faibles que leurs homologues calculés à partir d'une transformée à Q constant (SGQ et CGZ). Cela est dû principalement au niveau de bruit en très basses fréquences dans le corpus RWC-pop, causé par les percussions. En effet, la grosse caisse occasionne un niveau d'énergie à ces fréquences, ne correspondant pas aux notes des agrégats joués. Une solution simple est alors de ne pas prendre en compte les fréquences les plus basses, comme c'est le cas dans la transformée à Q constant utilisée.

5.3 Apprentissage automatique par minimum de divergence

La plupart des travaux portant sur l'alignement de musique polyphonique font appel à des transformations heuristiques, de forme similaire à celles présentées plus haut. Hu *et al.* [2003] rapportent d'ailleurs que dans leurs expériences exploitant une représentation en vecteurs de chroma, l'utilisation de la transformation canonique entraine des performances similaires à celles obtenues avec une synthèse audio de la partition. En revanche, la construction de cette transformation a rarement été remise en cause. À notre connaissance, seuls İzmirli et Dannenberg [2010] ont tenté d'estimer la valeur de la matrice en question à partir de données réelles. Cependant, ce travail se limite à la représentation en chromagramme et l'évaluation est effectuée sur une tâche de classification.

Dans la suite de ce chapitre, nous cherchons à optimiser les transformations utilisées pour toutes les représentations étudiées. Cela permet la construction de nouveaux attributs caractérisant les agrégats. Nous examinons alors l'influence de ces attributs sur les performances d'alignement.

5.3.1 Définition

Compte tenu de l'application visée, les critères les plus pertinents pour la construction de la matrice de transformation $\mathbf{W}$ sont ceux mesurant la qualité des alignements obtenus. Malheureusement, les liens entre les valeurs de $\mathbf{W}$ et ces critères ne sont pas directement calculables et une optimisation directe de ces mesures n'est pas réalisable. De même, la stratégie de recherche sur une grille n'est pas applicable ici en raison de la grande dimension du problème.

Il faut donc recourir à une fonction de cout plus simple pour déterminer la transformation optimale. Pour cela, considérons la construction de la matrice $\mathbf{W}$ comme un problème de régression linéaire. En effet, l'équation (5.1) définit l'attribut d'agrégat $f_1(c, v_n)$ comme la divergence entre l'observation v_n et une estimation linéaire à partir du vecteur de notes h_c. Nous proposons alors le critère du *minimum de divergence* (MD), qui vise à minimiser la distance cumulée sur la base d'apprentissage.

Soient $\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}} = v^{\mathfrak{e}}_1 \dots v^{\mathfrak{e}}_{N_{\mathfrak{e}}}$ et $\mathbf{c}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}} = c^{\mathfrak{e}}_1 \dots c^{\mathfrak{e}}_{N_{\mathfrak{e}}}$ respectivement la séquence d'observations et les agrégats de l'enregistrement $\mathfrak{e}$ de l'ensemble d'apprentissage (dont la longueur est $N_{\mathfrak{e}}$). Pour des raisons de clarté, nous noterons $h^{\mathfrak{e}}_n = h_{c^{\mathfrak{e}}_n}$ les vecteurs de notes correspondant aux agrégats annotés. Le critère utilisé définit la matrice de transformation optimale $\hat{\mathbf{W}}^{\mathrm{MD}}$ est comme

$$\hat{\mathbf{W}}^{\mathrm{MD}} = \arg\min_{\mathbf{W}} \sum_{\mathfrak{e}} \sum_{n=1}^{N_{\mathfrak{e}}} D\left(v^{\mathfrak{e}}_n, \mathbf{W} h^{\mathfrak{e}}_n\right), \tag{5.16}$$

Cette formulation peut aussi être dérivée du modèle génératif exposé en section 5.1.2 dans le cas de la distance KL1. En effet, comme indiqué par Virtanen *et al.* [2008], cela correspond alors au calcul des estimateurs du *maximum de vraisemblance* des paramètres des lois de Poisson intervenant dans ce modèle.

Afin de limiter les risques de surapprentissage à certaines notes ou tonalités, 12 versions de chaque morceau de la base d'apprentissage sont exploitées en transposant conjointement les vecteurs d'observations et les vecteurs de notes de -6 à $+5$ demi-tons. De cette façon, le nombre d'exemples d'apprentissage est homogène pour toutes les notes de chaque octave. L'opération de transposition effectuée est différente sur les trois types de représentations. Pour les chromagrammes, elle consiste en une simple permutation circulaire. Pour les semigrammes (ainsi que pour les vecteurs de notes), on procède en « décalant » les valeurs du bon nombre de composantes. Certains éléments sont alors supprimés et d'autres reçoivent une valeur nulle, dont il n'est pas tenu compte dans le calcul de la fonction de cout. Une transposition d'un spectrogramme correspond à une dilatation (ou une compression) de l'échelle fréquentielle. Elle est alors effectué à l'aide d'une interpolation linéaire des valeurs du spectrogramme.

5.3.2 Résolution

La fonction de cout définie dans l'équation (5.16) est convexe si la distance utilisée l'est. Il est alors aisé de montrer qu'avec la divergence de Kullback-Leibler symétrisée, l'estimation de $\mathbf{W}$ est un problème de minimisation convexe. De nombreuses méthodes existent pour la résolution de tels problèmes. L'approche itérative adoptée, fondée sur la

notion de *région de confiance*, est une variation de la méthode de Newton classique. Elle cherche à minimiser l'approximation quadratique donnée par le développement de Taylor à l'ordre 2 dans un voisinage du point courant. Cette limitation à un voisinage évite de considérer des pas trop grand lorsque la fonction-objectif est mal approximée par une fonction quadratique et permet donc de meilleures propriétés de convergence. L'inversion de la matrice hessienne est évitée en calculant une solution approchée du problème de minimisation locale par la méthode exposée par Branch *et al.* [1999].

L'implémentation employée est celle de l'*optimization toolbox* du logiciel MATLAB. Les conditions d'arrêt de l'algorithme sont des variations inférieures à la valeur 10^{-6} pour la fonction-objectif ou pour la norme de $\mathbf{W}$. Dans tous les cas, l'optimisation converge en une dizaine d'itération.

La figure 5.3 compare la matrice estimée à la matrice de gabarits heuristiques présentée section 5.2 pour la représentation SGQ. On peut tout d'abord remarquer que l'apprentissage fait apparaitre des harmoniques supplémentaires par rapport à la transformation heuristique. De plus, l'importance relative des différentes harmoniques varie suivant la note considérée. Ce comportement est observé pour toutes les représentations. Les colonnes de $\mathbf{W}$, ne sont donc pas toutes construites à partir d'un même gabarit, mais capturent des distributions d'énergie spectrale spécifiques à chaque note.

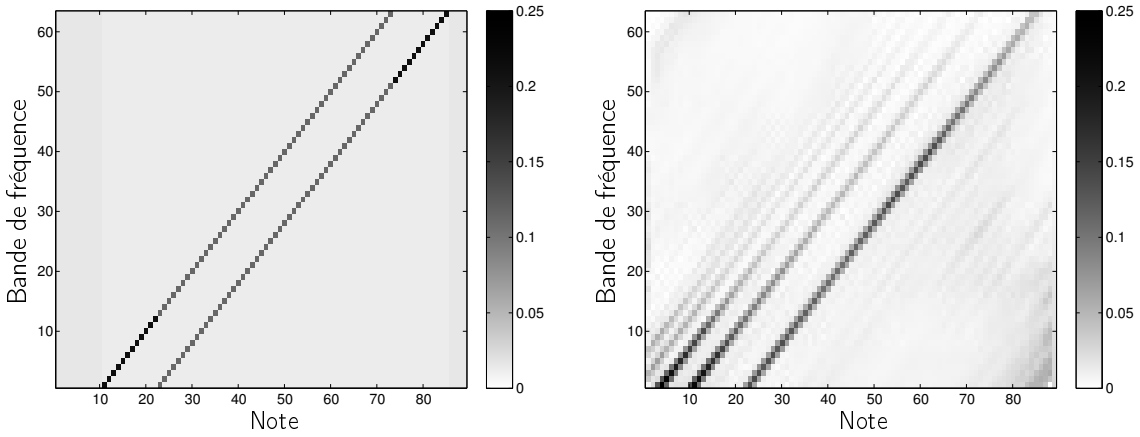


FIGURE 5.3 – Comparaison de la matrice de transformation heuristique (à gauche) et de la matrice estimée par le critère MD (à droite), pour la représentation en semigramme SGQ.

5.3.3 Influence sur l'alignement par un modèle simple

Le même système MCRF0 que présenté dans la section précédente est maintenant utilisé pour mesurer l'efficacité des transformations estimées sur une tâche d'alignement simple. Les taux d'alignement obtenus sont représentés figure 5.4.

Les résultats mettent en évidence l'intérêt de l'apprentissage de la transformation optimale. En effet, pour toutes les représentations testées, une amélioration significative de la qualité des alignements est observée sur les deux corpus. Le taux d'alignement à 300 ms

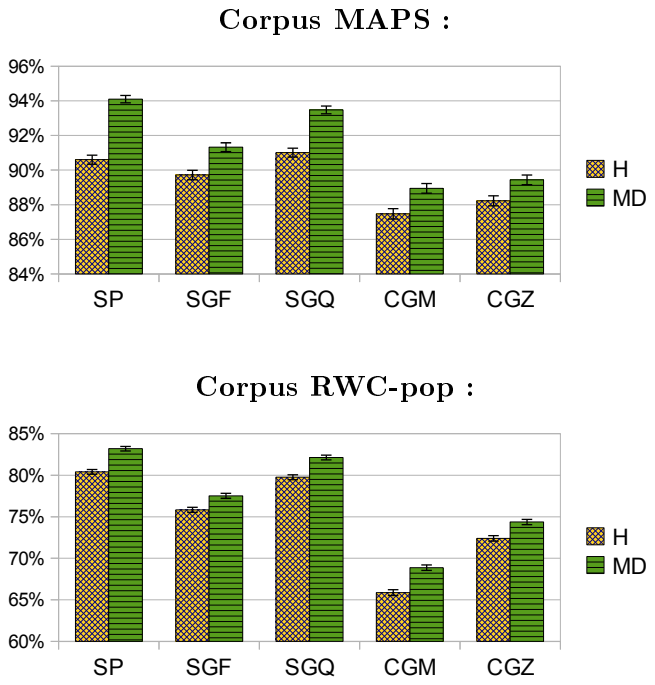


Figure 5.4 – Comparaisons des taux d'alignement à 300 ms obtenus avec les transformations heuristiques (H) et apprises (MD), pour un alignement par le modèle MCRF0.

du meilleur système passe ainsi de 80,4% à 83,2% pour la base RWC-pop et de 91,0% à 94,1% sur MAPS. De même, les couts de classification (CCMP) sont améliorés de 1% à 5% selon les représentations.

On peut aussi constater que la représentation en spectrogramme, qui obtenait des résultats similaires au semigramme SGQ avec les gabarits heuristiques, est ici plus efficace. Son taux d'alignement à 100 ms est 84,2% pour MAPS et 60,4% pour RWC-pop, contre 83,2% et 58,2% respectivement avec SGQ. Une explication tient à la plus grande dimension du spectrogramme, qui permet de représenter plus précisément le contenu de l'audio. Or, cette plus grande capacité de modélisation n'est apparemment pas exploitée de façon optimale par la transformation heuristique de l'équation (5.13), du fait du plus grand nombre de paramètres fixés manuellement. En revanche, les performances relatives des autres représentations ne sont pas modifiées par l'apprentissage. Cela indique que ces différences sont dûes aux représentations elles-mêmes et non à la transformation utilisée dans le calcul de l'attribut d'agrégat.

5.4 Apprentissage discriminatif par maximum de vraisemblance (MV)

5.4.1 Formulation

Nous proposons maintenant une autre stratégie pour un apprentissage discriminatif de la matrice $\mathbf{W}$. Cette méthode, fondée sur le critère de *maximum de vraisemblance* (MV), a pour avantage de tenir compte du modèle d'alignement par CRF pour la construction d'un attribut d'agrégat optimal. Comme on l'a vu dans les chapitres précédents, un modèle CRF permet de calculer la probabilité d'une séquence d'étiquettes, conditionnellement aux observations. La valeur optimale de la matrice $\mathbf{W}$, considérée ici comme un paramètre du modèle, est alors celle qui maximise la probabilité de la séquence d'agrégats réellement observée.

Pour des raisons de complexité, on utilise le modèle MCRF0 pour cet apprentissage. En effet, ce modèle tient compte uniquement des étiquettes d'agrégats et l'espace des variables cachées est donc assez petit pour être exploré exhaustivement. On peut alors exprimer, pour chaque enregistrement $\mathfrak{e}$ de l'ensemble d'apprentissage, la probabilité de la séquence d'agrégats $\mathbf{c}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}$ connaissant la représentation $\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}$ de l'audio par la formule :

$$P(\mathbf{c}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}|\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}};\Theta) = \frac{1}{Z(\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}})}\phi(c^{\mathfrak{e}}_1, v^{\mathfrak{e}}_1)\prod_{n=2}^{N_{\mathfrak{e}}}\psi^{\mathrm{M}}(c^{\mathfrak{e}}_n, c^{\mathfrak{e}}_{n-1})\phi(c^{\mathfrak{e}}_n, v^{\mathfrak{e}}_n) \tag{5.17}$$

avec une fonction de transition de la forme définie dans l'équation (3.29). Comme indiqué plus haut, nous choisissons de fixer à la valeur 1 tous les paramètres λ_0 et λ_1 des transitions. Les seules valeurs possibles de la fonction de transition sont alors 0 et 1. Cette fonction peut donc être interprétée comme constituant la fonction indicatrice de l'ensemble des séquences admissibles (de probabilité non nulle). Cet ensemble sera désigné par $\boldsymbol{\mathcal{C}}^{\mathfrak{e}}_{\mathrm{adm}}$. On peut alors écrire, pour une séquence d'étiquette $\mathbf{c}_{1:N_{\mathfrak{e}}}$ quelconque :

$$\prod_{n=2}^{N_{\mathfrak{e}}}\psi^{\mathrm{M}}(c^{\mathfrak{e}}_n, c^{\mathfrak{e}}_{n-1}) = \mathbf{1}_{\boldsymbol{\mathcal{C}}^{\mathfrak{e}}_{\mathrm{adm}}}(\mathbf{c}_{1:N_{\mathfrak{e}}})\,. \tag{5.18}$$

De plus, en notant $f_1(c^{\mathfrak{e}}_n, v^{\mathfrak{e}}_n; \mathbf{W})$ l'attribut d'observation, paramétré par la matrice $\mathbf{W}$, la fonction d'observation est ici :

$$\phi(c^{\mathfrak{e}}_n, v^{\mathfrak{e}}_n) = \exp\left\{\mu_1 f_1(c^{\mathfrak{e}}_n, v^{\mathfrak{e}}_n; \mathbf{W})\right\} \tag{5.19}$$

L'équation (5.17) peut donc s'écrire :

$$P(\mathbf{c}^{\mathfrak{e}}_{1:N}|\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}) = \frac{\mathbf{1}_{\boldsymbol{\mathcal{C}}^{\mathfrak{e}}_{\mathrm{adm}}}(\mathbf{c}^{\mathfrak{e}}_{1:N})}{Z(\mathbf{v}^{\mathfrak{e}}_{1:N})}\exp\left\{\mu_1\sum_{n=1}^{N} f_1(c^{\mathfrak{e}}_n, v^{\mathfrak{e}}_n; \mathbf{W})\right\}. \tag{5.20}$$

On note $\Theta = (\mu_1, \mathbf{W})$ les paramètres du modèle considéré. La log-vraisemblance de ces paramètres calculée sur toute la base d'apprentissage est donc donnée par :

$$\mathcal{L}(\Theta) = \sum_{\mathfrak{e}}\left\{\mu_1\sum_{n=1}^{N_{\mathfrak{e}}} f_1(c^{\mathfrak{e}}_n, v^{\mathfrak{e}}_n; \mathbf{W}) - \log Z(\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}})\right\} \tag{5.21}$$

et l'estimateur du *maximum de vraisemblance* $\hat{\Theta}^{\mathrm{MV}}$ est défini comme celui maximisant cette fonction.

Remarque : originalité de l'approche proposée

L'approche suivie ici se distingue de la plupart des stratégies pour l'apprentissage de modèles CRF. En effet, il est plus classique de considérer des attributs fixes, spécifiques à chaque étiquette, et d'estimer les poids μ correspondants, comme décrit par Lafferty *et al.* [2001]. Le choix des attributs pertinents est alors souvent laissé à l'utilisateur. Notre méthode peut ainsi être rapprochée des modèles proposés par Peng *et al.* [2009] et Do et Artières [2009], où les attributs sont issus de fonctions non linéaires des observations (en l'occurence des réseaux de neurones), dont les paramètres sont estimés conjointement à ceux du CRF. Cependant, l'application étudiée ici est différente des tâches de classifications usuelles visées par ces modèles. En effet pour l'alignement musique-sur-partition, la structure du modèle utilisé n'est pas fixe puisqu'elle dépend du morceau considéré, l'automate des agrégats étant construit d'après la partition. De plus, l'ensemble des étiquettes (c'est-à-dire des agrégats) que l'on souhaite modéliser est infini, puisqu'il correspond à toutes les combinaisons de notes possibles [5]. Il est donc impossible d'estimer des attributs spécifiques à chaque étiquette. En revanche, la représentation des agrégats en vecteurs de notes h_c, présentée en section 5.1.1, dote l'ensemble des étiquettes possibles d'une structure d'espace vectoriel. Cela permet alors faire appel à la transformation linéaire $\mathbf{W}$ pour définir un attribut général, applicable à un agrégat quelconque et ne dépendant pas de la structure de l'automate.

5.4.2 Calcul des paramètres optimaux

La log-vraisemblance définie dans l'équation (5.21) n'est pas concave, du fait de la non-linéarité de f_1 par rapport à $\mathbf{W}$. Néanmoins, au risque d'aboutir à un maximum local, il est possible d'utiliser des méthodes de type « descente de gradient » pour optimiser cette fonction.

Nous notons

$$F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}};\mathbf{W}) = \sum_{n=1}^{N_{\mathfrak{e}}} f_1(C_n, v_n^{\mathfrak{e}};\mathbf{W}) \tag{5.22}$$

Le facteur de normalisation $Z(\mathbf{v}_{1:N}^{\mathfrak{e}})$ s'écrit alors

$$Z(\mathbf{v}_{1:N_{\mathfrak{e}}}^{\mathfrak{e}}) = \sum_{\mathbf{C}_{1:N_{\mathfrak{e}}} \in \boldsymbol{\mathcal{C}}_{\mathrm{adm}}^{\mathfrak{e}}} e^{\mu_1 F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}};\mathbf{W}))}. \tag{5.23}$$

5. On peut imaginer de borner cet ensemble en limitant le nombre de notes dans un même agrégat. Cependant, la combinatoire d'un tel ensemble reste beaucoup trop importante. En effet, il n'est pas rare d'observer une quinzaine de notes simultanément.

La dérivée de la log-vraisemblance par rapport au paramètre μ_1 est donc

$$\begin{aligned}\frac{\partial \mathcal{L}(\Theta)}{\partial \mu_1} &= \sum_{\mathfrak{e}} \left\{ F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) - \frac{1}{Z(\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}})} \frac{\partial Z(\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}})}{\partial \mu_1} \right\} \qquad (5.24)\\ &= \sum_{\mathfrak{e}} \left\{ F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) - \frac{1}{Z(\mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}})} \sum_{\mathbf{C}_{1:N_{\mathfrak{e}}} \in \boldsymbol{\mathcal{C}}^{\mathfrak{e}}_{\text{adm}}} F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) e^{\mu_1 F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W})} \right\}.\end{aligned}$$

En utilisant l'équation (5.20), on a

$$\begin{aligned}\frac{\partial \mathcal{L}(\Theta)}{\partial \mu_1} &= \sum_{\mathfrak{e}} \left\{ F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) - \sum_{\mathbf{C}_{1:N_{\mathfrak{e}}} \in \boldsymbol{\mathcal{C}}^{\mathfrak{e}}_{\text{adm}}} F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) P(\mathbf{C}_{1:N_{\mathfrak{e}}} | \mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}; \Theta) \right\}\\ &= \sum_{\mathfrak{e}} \left\{ F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) - \mathbb{E}\left[F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W}) \middle| \mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}; \Theta \right] \right\} \qquad (5.25)\end{aligned}$$

où $\mathbb{E}\left[\cdot \middle| \mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}; \Theta \right]$ désigne l'espérance conditionnelle calculée par modèle CRF avec les paramètres Θ. Ce terme d'espérance peut en pratique être calculé grâce à une variante de l'algorithme *forward-backward* [Lafferty *et al.*, 2001].

Un calcul similaire donne la dérivée de la log-vraisemblance par rapport aux composantes de $\mathbf{W}$:

$$\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{W}_{i,j}} = \mu_1 \sum_{\mathfrak{e}} \left\{ \frac{\partial F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W})}{\partial \mathbf{W}_{i,j}} - \mathbb{E}\left[\frac{\partial F_1^{\mathfrak{e}}(\mathbf{C}_{1:N_{\mathfrak{e}}}; \mathbf{W})}{\partial \mathbf{W}_{i,j}} \middle| \mathbf{v}^{\mathfrak{e}}_{1:N_{\mathfrak{e}}}; \Theta \right] \right\}. \qquad (5.26)$$

Plusieurs difficultés se posent pour l'optimisation de la fonction-objectif. Tout d'abord, la matrice hessienne ne peut pas être utilisée car son calcul est très complexe. Nous choisissons alors une stratégie de descente dans la direction du gradient. Cependant, en raison du logarithme qui intervient dans le calcul de l'attribut, ce gradient diverge lorsque les composantes $\mathbf{W}_{i,j}$ tendent vers 0. La mise à jour doit donc être controlée, afin d'éviter une courbe d'optimisation trop chaotique. Enfin, les dérivées partielles de la fonction-objectif par rapport aux paramètres μ_1 et $\mathbf{W}$, détaillés respectivement dans les equations (5.25) et (5.26), ne sont pas forcément du même ordre de grandeur. L'approche d'optimisation adoptée est alors une méthode de recherche linéaire très simple, où les deux paramètres μ_1 et $\mathbf{W}$ sont alternativement mis à jour dans la direction de leur gradient, avec un pas adaptatif. L'algorithme 5.1 détaille cette stratégie d'optimisation.

En raison de la complexité du calcul du gradient, nous limitons le nombre d'itérations de l'algorithme de maximisation à 100. De plus, l'initialisation de $\mathbf{W}$ est effectuée avec la matrice issue de l'apprentissage par minimum de divergence. Ce choix d'initialisation est fondé sur l'intuition que cette valeur est proche de l'optimum et que l'algorithme convergera donc rapidement. Nous faisons alors l'hypothèse que la stratégie d'apprentissage MV n'est pas avantagée de façon significative par rapport à la précédente, car les critères optimisés ne sont pas les mêmes.

Un exemple de courbe d'optimisation est présenté figure 5.5, pour la représentation SGCQT. Des chutes brutales peuvent être observées dans l'évolution de la fonction-objectif,

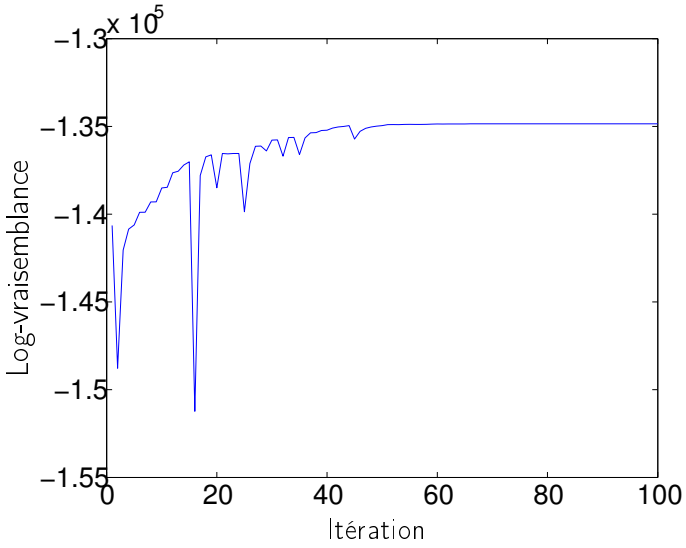


FIGURE 5.5 – Évolution de la log-vraisemblance dans l'apprentissage des paramètres de la réprésentation SGCQT par le critère du *maximum de vraisemblance* (MV).

correspondant à des pas de mise à jour trop important. Cela traduit la sensibilité de la log-vraisemblance aux valeurs de $\mathbf{W}$. Malgré la simplicité de l'algorithme mis en jeu, la procédure semble converger vers une valeur satisfaisante. Il pourrait néanmoins être intéressant d'envisager des stratégies d'optimisation plus efficaces, ainsi que de tester l'influence de l'initialisation, ce que nous n'avons pu mettre à exécution, faute de temps.

5.4.3 Matrices estimées

Les figures 5.6 et 5.7 représentent les matrices de transformation estimées par la stratégie MV pour les représentations SP et CGM. Ces matrices sont comparées à celles apprises par le critère MD. Dans le cas du spectrogramme (SP) figure 5.6, les différences sont peu visibles. Cela peut être expliqué par le nombre relativement faible d'itérations dans notre algorithme d'optimisation, comparativement à la très grande dimension de cette représentation (39872 composantes).

On peut néanmoins constater que le critère du maximum de vraisemblance mène à une plus grande « dispersion » de l'énergie sur les différentes bandes de fréquences. Cela peut s'expliquer par le principe d'entropie maximum, qui sous-tend la formulation des CRF Wallach [2004]. En effet, l'apprentissage effectué ici ne tend pas à modéliser les observations de façon précise, mais à discerner les différents agrégats. Intuitivement, cette stratégie estime la transformation la plus uniforme possible, permettant cependant une bonne discrimination des agrégats. Ce phénomène est surtout observé sur les chromagrammes, et dans une moindre mesure sur le semigramme SGQ. Une explication pourrait être que ces représentations sont plus grossières. De fait, les dynamiques des observations extraites sont moindres que celles des représentations SP et SGF et l'emploi de gabarits uniformes ne semble donc pas nuire aux capacités de discrimination de l'attribut résultant.

Données : $\Theta^{(0)} = \left(\mu_1^{(0)}, \mathbf{W}^{(0)}\right), \alpha_0 \in \mathbb{R}_+, \alpha_1 \in \mathbb{R}_+, p \in \{0,1\}$
pour $i \longleftarrow 1$ **à** *NbIter* **faire**
  cas où $p = 0$
    Calculer $\nabla_{\mu_i}\mathcal{L}\left(\Theta^{(i-1)}\right)$
    $\mu_1^{(i)} \longleftarrow \mu_1^{(i-1)} + \alpha_0 \frac{\nabla_{\mu_1}\mathcal{L}\left(\Theta^{(i-1)}\right)}{\|\nabla_{\mu_1}\mathcal{L}\left(\Theta^{(i-1)}\right)\|}$
    $\Theta^{(i)} \longleftarrow \left(\mu_1^{(i)}, \mathbf{W}^{(i-1)}\right)$
    si $\mathcal{L}(\Theta^{(i)}) > \mathcal{L}\left(\Theta^{(i-1)}\right)$ **alors**
      $p \longleftarrow 1$
      si α_0 *insuffisant* **alors** $\alpha_0 \longleftarrow \frac{3}{2}\alpha_0$
    sinon
      $\mu_1^{(i)} \longleftarrow \mu_1^{(i-1)}$
      $\alpha_0 \longleftarrow \frac{1}{2}\alpha_0$
  cas où $p = 1$
    Calculer $\nabla_{\mathbf{W}}\mathcal{L}\left(\Theta^{(i-1)}\right)$
    $\mathbf{W}^{(i)} \longleftarrow \mathbf{W}^{(i-1)} + \alpha_1 \frac{\nabla_{\mathbf{W}}\mathcal{L}\left(\Theta^{(i-1)}\right)}{\|\nabla_{\mathbf{W}}\mathcal{L}\left(\Theta^{(i-1)}\right)\|}$
    $\Theta^{(i)} \longleftarrow \left(\mu_1^{(i-1)}, \mathbf{W}^{(i)}\right)$
    si $\mathcal{L}(\Theta^{(i)}) > \mathcal{L}\left(\Theta^{(i-1)}\right)$ **alors**
      $p \longleftarrow 0$
      si α_1 *insuffisant* **alors** $\alpha_1 \longleftarrow \frac{3}{2}\alpha_1$
    sinon
      $\mathbf{W}^{(i)} \longleftarrow \mathbf{W}^{(i-1)}$
      $\alpha_0 \longleftarrow \frac{1}{2}\alpha_0$

Algorithme 5.1 : Algorithme d'optimisation utilisé pour l'apprentissage par maximum de vraisemblance. Le nombre d'itération est fixé à 100. La variable p utilisée dans la procédure désigne quel paramètre μ_1 ou $\mathbf{W}$ est mis à jour à l'itération courante. La condition d'*insuffisance* d'un pas α_p correspond au cas où les 3 itérations précédentes utilisant ce pas conduisent à une augmentation de la vraisemblance.

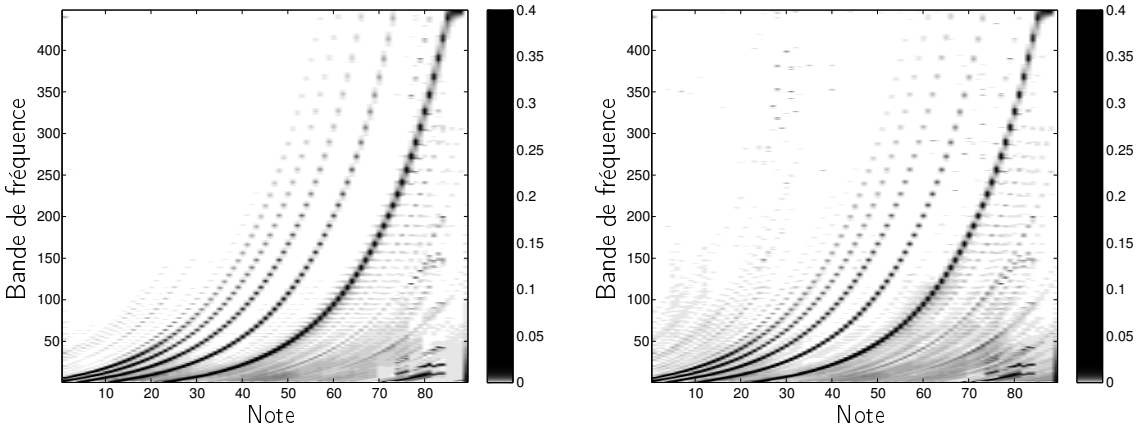


FIGURE 5.6 – Représentation SP : comparaison des matrices de transformation estimées par les critères MD (à gauche) et MV (à droite).

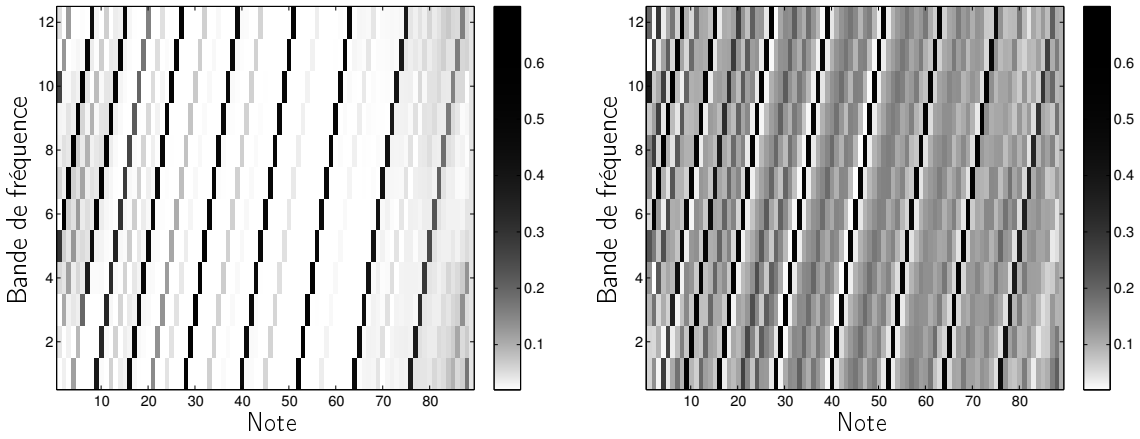


FIGURE 5.7 – Représentation CGM : comparaison des matrices de transformation estimées par les critères MD (à gauche) et MV (à droite).

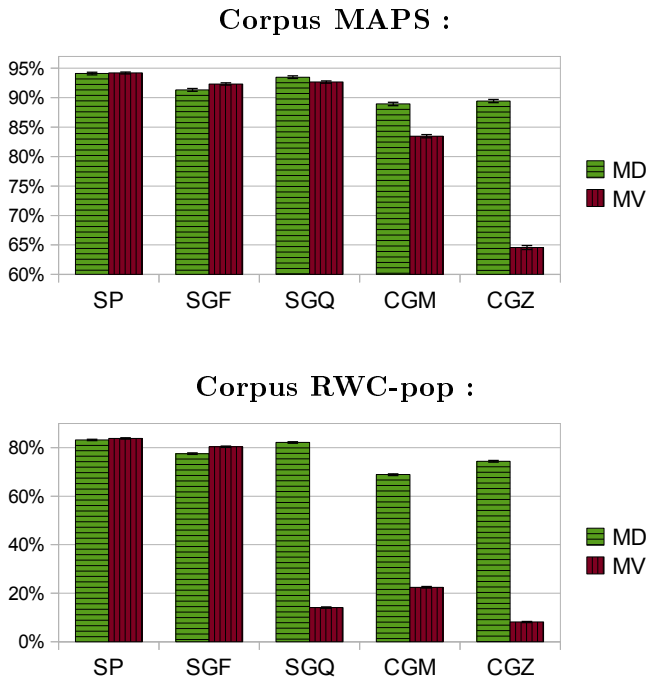


Figure 5.8 – Taux d'alignement moyens pondérés à 300 ms obtenus avec les transformations issues des deux stratégies d'apprentissage, pour un alignement par le modèle MCRF0.

5.4.4 Expérience d'alignement

Comme dans les sections précédentes, nous exploitons les matrices de transformations estimées pour un alignement avec le modèle MCRF0. Notons que la valeur de μ_1, bien que jouant un rôle dans le calcul de la vraisemblance et donc utilisée pour l'apprentissage de la matrice $\mathbf{W}$, n'a pas d'influence sur les séquences d'agrégats détectées ici, comme on l'a exposé dans la section 4.4. Les différences par rapport aux expériences précédentes sont donc dûes uniquement à la transformation. Les taux d'alignement obtenus sont présentés dans la table 5.2 et comparés à ceux exploitant l'apprentissage MD.

Puisque l'apprentissage par maximum de vraisemblance prend en compte le modèle CRF exploité pour l'alignement, on peut s'attendre à ce qu'il permette une amélioration des résultats. En effet, dans le cas des représentations SP et SGF, on observe une augmentation des taux d'alignement (même si cette augmentation n'est pas significative sur le corpus MAPS avec le spectrogramme). En revanche, les scores s'effondrent avec les autres représentations. Cette dégradation est spectaculaire sur le corpus RWC-pop, où le taux d'alignement à 300 ms passe par exemple de 74,4% à 8,1% pour le chromagramme CGZ.

La raison de ce phénomène est que le critère MV n'est pas lié à la maximisation de nos mesures d'évaluation. En effet, l'apprentissage maximise la probabilité des séquences d'agrégats annotées dans l'ensemble d'apprentissage, mais ne limite pas la probabilité des autres séquences. En particulier, rien n'assure que cette séquence annotée sera la plus

probable. En d'autres termes, même si la probabilité de la « bonne » séquence est optimisée, il est possible que d'autres séquences soient encore plus favorisées par cette optimisation. Il n'y a donc aucune garantie que les scores d'évaluation de l'alignement augmentent, même sur l'ensemble d'apprentissage. De fait, pour la même représentation, le cout de classification moyen pondéré sur cet ensemble est de 74,3%, alors qu'il est égal à 50,9% avec l'apprentissage MD.

En examinant plus précisément les résultats, on constate que les erreurs sont causées par de nombreux alignements aberrants, où presque tout le morceau est décodé comme étant un agrégat vide (représentant les parties où aucune note n'est jouée), représentant notamment le début ou la fin du morceau. L'attribut fait en effet apparaitre un biais très important en faveur de cet agrégat. Ce phénomène peut être expliqué par la forme de la vraisemblance. La dérivée partielle de l'équation (5.26) peut être développée en :

$$\begin{aligned}\frac{\partial \mathcal{L}(\Theta)}{\partial \mathbf{W}_{i,j}} &= \mu_1 \sum_{\mathfrak{e}} \sum_{n=1}^{N_\mathfrak{e}} \left\{ \frac{\partial f_1^\mathfrak{e}(c_n^\mathfrak{e}, v_n^\mathfrak{e}; \mathbf{W})}{\partial \mathbf{W}_{i,j}} - \sum_{c_n \in \mathcal{C}^\mathfrak{e}} \frac{\partial f_1^\mathfrak{e}(c_n, v_n^\mathfrak{e}; \mathbf{W})}{\partial \mathbf{W}_{i,j}} P\left(c_n \middle| \mathbf{v}_{1:N_\mathfrak{e}}^\mathfrak{e}; \Theta\right) \right\} \qquad (5.27) \\ &= \mu_1 \sum_{\mathfrak{e}} \sum_{n=1}^{N_\mathfrak{e}} \frac{\partial}{\partial \mathbf{W}_{i,j}} \left\{ f_1^\mathfrak{e}(c_n^\mathfrak{e}, v_n^\mathfrak{e}; \mathbf{W}) - \sum_{c_n \in \mathcal{C}^\mathfrak{e}} \omega_{c_n} f_1^\mathfrak{e}(c_n, v_n^\mathfrak{e}; \mathbf{W}) \right\} \Bigg|_{\omega_{c_n} = P\left(c_n \middle| \mathbf{v}_{1:N_\mathfrak{e}}^\mathfrak{e}; \Theta\right)}\end{aligned}$$

où $\mathcal{C}^\mathfrak{e}$ est l'ensemble des agrégats présents dans le morceau $\mathfrak{e}$. Cette équation traduit le fait que la stratégie MV vise à maximiser la différence entre la valeur de l'attribut des agrégats joués et la moyenne des attributs de tous les agrégats, pondérée par leurs probabilités. Cette idée, intuitivement valide, peut néanmoins conduire à un biais lorsque les occurences des étiquettes dans la base d'apprentissage sont déséquilibrées, puisque l'optimisation se concentre sur la sélection du bon agrégat parmi les plus probables.

De plus, les probabilités en jeu dans l'équation (5.27) sont les probabilités des *agrégats*, chacune égale à la somme des probabilités de toutes les séquences contenant l'étiquette considérée. De ce fait, une plus grande importance est donnée aux agrégats contenant un grand nombre de séquences de probabilités modérées par rapport aux agrégats contenant une seule séquence très probable. C'est pourquoi l'apprentissage accorde peu d'importance à l'agrégat vide, souvent isolé aux extrémités d'un morceau. Dans notre cas, cela conduit à une surestimation quasi-systématique de la probabilité de cet agrégat.

Modification de l'attribut de l'agrégat vide

Afin de limiter ce problème, nous choisissons de modifier la valeur de l'attribut f_1 de l'agrégat vide. Elle est alors calculée, pour chaque instance de cet agrégat, comme la moyenne des attributs des agrégats voisins, en considérant les 5 précédents et les 5 suivants. On évalue aussi l'influence de cette modification sur les alignements des systèmes précédents.

Les taux d'alignement de tous les systèmes considérés sont alors compilés table 5.2. On peut tout d'abord observer que la modification de l'attribut de l'agrégat vide améliore considérablement les scores des chromagrammes et du semigramme SGQ après l'apprentissage MV. Cette dernière représentation obtient alors le meilleur taux d'alignement sur le corpus

Corpus MAPS :

Représentation	SP		SGF		SGQ		CGM		CGZ	
Agrégat vide	(o)	(m)	(o)	(m)	(o)	(m)	(o)	(m)	(o)	(m)
W Heuristique	90.6	90.5	89.7	89.6	91.0	90.9	87.5	87.4	88.2	88.1
W Appr. MD	94.1	93.9	91.3	91.3	93.5	93.3	**88.9**	**88.9**	**89.4**	89.3
W Appr. MV	**94.2**	94.0	**92.3**	92.2	92.7	**93.6**	83.5	86.0	64.6	88.5

Corpus RWC-pop :

Représentation	SP		SGF		SGQ		CGM		CGZ	
Agrégat vide	(o)	(m)	(o)	(m)	(o)	(m)	(o)	(m)	(o)	(m)
W Heuristique	80.4	79.8	75.8	75.9	79.8	79.1	65.9	65.1	72.4	71.4
W Appr. MD	83.2	83.6	77.5	77.7	82.1	82.2	**68.9**	68.2	**74.4**	74.3
W Appr. MV	83.8	**84.0**	**80.4**	80.1	14.1	**84.4**	22.4	66.5	8.1	72.9

Table 5.2 – Taux d'alignement moyens pondérés à 300 ms obtenus par le modèle MCRF0 pour les différentes méthodes de calcul des attributs avec la distance KLs (en %). Les systèmes notés (o) et (m) utilisent respectivement les attributs originaux et modifiés des agrégats vides.

RWC-pop (84,4%). En revanche, la modification n'augmente pas de façon significative les résultats découlant des autres stratégies d'apprentissage.

D'après ces expériences, la stratégie d'apprentissage par maximum de vraisemblance permet donc d'améliorer les alignements par rapport au minimum de divergence, pour les représentations en spectrogramme et en semigramme. En revanche, elle obtient de moins bons résultats pour les représentations en chromagramme (CGM et CGZ). Cela peut être expliqué par le plus faible pouvoir discriminant de ces représentations du fait de leur plus petite dimension. Ainsi, l'apprentissage a tendance à se baser davantage sur la fonction de transition (qui apparait de façon implicite dans le calcul des probabilités) pour discerner les agrégats, privilégiant alors le critère d'entropie pour la mise à jour de la transformation. En comparaison, la stratégie MD ne tient pas compte de la fonction de transition et cherche alors une matrice **W** expliquant mieux les données.

5.5 Application aux modèles d'alignement par CRF

Dans cette section, nous intégrons les attributs d'agrégats calculés précédemment dans les systèmes d'alignement présentés au chapitre 4. Dans ces expériences, nous omettons les représentations SGF et CGM afin de conserver la représentation la plus performante de chaque classe (spectrogramme, semigramme et chromagramme). De plus, pour des raisons de temps, nous nous sommes limités aux systèmes ne prenant pas en compte l'intégration du voisinage pour le calcul de la fonction d'observation.

5.5.1 Modèle MCRF

Dans le modèle MCRF, l'attribut d'attaque défini au chapitre 4.2.2 est ajouté au système précédent. Pour ces expériences, la valeur optimale $\hat{\mu}_2^{\mathrm{MV}}$ du paramètre contrôlant l'importance donné à cet attribut est tout d'abord estimée pour chaque système par le critère du maximum de vraisemblance.

Cependant, les premiers résultats d'alignement obtenus avec ces valeurs ne montraient pas d'amélioration notable par rapport au système n'utilisant pas l'attribut d'attaque. Cela provient du fait que le critère MV n'est pas directement lié à la qualité de l'alignement obtenu, comme expliqué en section 5.3.3. Une nouvelle stratégie de recherche sur une grille est alors exploitée. Des alignements sont calculés sur la base d'apprentissage avec plusieurs valeurs pour le paramètre de l'attribut de phase, de la forme $\mu_2 = \beta\hat{\mu}_2^{\mathrm{MV}}$, avec $\beta \in \{1, 2, 5, 10, 20\}$. La valeur occasionnant les meilleures performances est sélectionnée. Les valeurs retenues sont toujours 5 ou 10.

Les résultats obtenus sur la base de test sont présentés en table 5.3. Rappelons que les expériences rapportées ici exploitent la divergence de Kullback-Leibler symétrique dans le calcul de l'attribut d'agrégat. Cela explique les différences entre scores obtenus ici avec la transformation heuristique pour la représentation CGZ et ceux issus des expériences du chapitre précédent. Par exemple, le taux d'alignement à 300 ms est ici de 95,0% sur le corpus MAPS, alors qu'il est égal à 93,1% avec la divergence KL1.

Comme attendu, les performances augmentent de façon significative par rapport au modèle MCRF0. On atteint ainsi une imprécision moyenne pondérée de 55 ms sur le corpus MAPS et de 89 ms sur RWC-pop, avec la représentation en semigramme. Nous voyons là encore l'avantage de l'apprentissage de la transformation $\mathbf{W}$. En effet, pour toutes les représentations, les alignements exploitant un apprentissage obtiennent de meilleurs scores, quelle que soit la mesure utilisée.

On peut aussi remarquer que l'ajout de l'agrégat d'attaque permet de résoudre une grande partie des problèmes liés au traitement de l'agrégat vide dans le cas de l'apprentissage MV. Cet agrégat, en détectant les phases d'attaque, limite fortement la probabilité de l'agrégat vide au cours d'un morceau. De ce fait, les performances des représentations en semigramme et en chromagramme avec l'attribut original issu de cet apprentissage sont ici meilleures que celle exploitant les transformations heuristiques.

5.5.2 Modèles SMCRF

Pour les alignements avec le modèle SMCRF, nous utilisons la valeur $\hat{\mu}_2^{\mathrm{MV}}$ calculée précédemment pour le paramètre de l'attribut d'attaque. Ce choix s'appuie sur les résultats de la recherche sur une grille de valeurs rapportés dans le chapitre précédent, qui indiquaient que dans le cas du modèle SMCRF, une valeur plus faible était préférable pour μ_2, par rapport au MCRF. La valeur $\gamma_1 = \frac{1}{200}$ est retenue pour le paramètre du modèle temporel, d'après les mêmes expériences.

La table 5.4 regroupe les résultats obtenus sur la base de test. Les comportements des différents systèmes sont globalement les mêmes que précédemment. On peut cependant remarquer que la représentation en spectrogramme est ici un peu moins précise que le

Représentation en spectrogramme (SP) :

	Corpus MAPS				Corpus RWC-pop			
Apprentissage	H	MD	MV(o)	MV(m)	H	MD	MV(o)	MV(m)
TAMP (θ=300 ms)	94.4	96.8	**96.9**	96.7	90.4	92.2	**92.4**	91.8
TAMP (θ=100 ms)	83.7	86.6	**86.9**	86.7	67.4	69.6	**69.9**	69.5
TAMP (θ=50 ms)	66.4	67.9	**68.4**	68.2	48.6	50.2	**50.6**	50.2
IMP (ms)	63	57	**56**	**56**	102	**93**	**93**	**93**
CCMP	30.2	26.4	**25.8**	26.7	53.4	51.5	**51.2**	52.8

Représentation en semigramme SGQ :

	Corpus MAPS				Corpus RWC-pop			
Apprentissage	H	MD	MV(o)	MV(m)	H	MD	MV(o)	MV(m)
TAMP (θ=300 ms)	95.3	96.7	**97.0**	96.9	90.1	91.5	90.7	**92.1**
TAMP (θ=100 ms)	83.1	85.1	85.7	**86.9**	67.3	69.2	69.4	**71.2**
TAMP (θ=50 ms)	63.6	65.3	66.1	**68.1**	49.0	50.8	51.6	**52.9**
IMP (ms)	65	58	58	**55**	100	94	93	**89**
CCMP	28.8	27.2	**25.4**	26.1	53.6	52.5	52.4	**52.2**

Représentation en chromagramme CGZ :

	Corpus MAPS				Corpus RWC-pop			
Apprentissage	H	MD	MV(o)	MV(m)	H	MD	MV(o)	MV(m)
TAMP (θ=300 ms)	95.0	94.9	**95.3**	94.7	86.8	**88.7**	**87.7**	87.6
TAMP (θ=100 ms)	82.0	**82.7**	82.1	82.1	59.3	**61.9**	60.9	61.5
TAMP (θ=50 ms)	62.7	**64.0**	62.7	63.8	40.9	**42.9**	42.1	42.6
IMP (ms)	68	**67**	**67**	68	122	**115**	117	117
CCMP	30.7	29.6	**29.1**	30.3	59.5	**57.6**	58.5	59.4

Table 5.3 – Résultats des alignements avec le modèle MCRF, pour les différents attributs d'agrégat. MV(o) et MV(m) désignent respectivement les attributs originaux et modifiés pour l'agrégat vide, après apprentissage par maximum de vraisemblance.

Représentation en spectrogramme (SP) :

	Corpus MAPS				Corpus RWC-pop			
Apprentissage	H	MD	MV(o)	MV(m)	H	MD	MV(o)	MV(m)
TAMP ($\theta = 300$ ms)	97.6	**98.7**	**98.7**	98.5	**94.4**	94.0	93.9	93.9
TAMP ($\theta = 100$ ms)	92.1	**93.4**	**93.4**	93.3	77.0	**77.4**	**77.4**	77.3
TAMP ($\theta = 50$ ms)	80.1	**80.5**	**80.5**	80.4	59.4	59.9	**60.0**	59.8
IMP (ms)	41	**37**	**37**	38	75	**74**	**74**	**74**
CCMP	21.0	**18.6**	**18.6**	19.1	42.1	**41.7**	**41.7**	42.0

Représentation en semigramme SGQ :

	Corpus MAPS				Corpus RWC-pop			
Apprentissage	H	MD	MV(o)	MV(m)	H	MD	MV(o)	MV(m)
TAMP ($\theta = 300$ ms)	97.9	98.6	**98.7**	98.6	95.9	95.8	89.2	**96.4**
TAMP ($\theta = 100$ ms)	91.9	93.4	**94.1**	94.0	79.5	80.2	75.9	**81.9**
TAMP ($\theta = 50$ ms)	78.1	79.4	**81.4**	81.3	56.4	57.6	55.1	**59.4**
IMP (ms)	42	38	**36**	**36**	73	70	67	**65**
CCMP	20.6	19.3	**18.3**	18.5	42.8	42.7	45.3	**42.2**

Représentation en chromagramme CGZ :

	Corpus MAPS				Corpus RWC-pop			
Apprentissage	H	MD	MV(o)	MV(m)	H	MD	MV(o)	MV(m)
TAMP ($\theta = 300$ ms)	97.4	97.9	**98.1**	**98.1**	94.1	**95.1**	83.8	93.6
TAMP ($\theta = 100$ ms)	89.9	**91.2**	90.8	90.7	74.0	**76.6**	66.9	74.6
TAMP ($\theta = 50$ ms)	74.4	**75.5**	75.1	74.9	50.0	**52.3**	45.7	50.8
IMP (ms)	48	**44**	45	46	86	**80**	84	83
CCMP	22.4	21.4	**21.2**	**21.2**	47.0	**45.1**	51.1	46.8

TABLE 5.4 – Résultats des alignements avec le modèle SMCRF, pour les différents attributs d'agrégat. MV(o) et MV(m) désignent respectivement les attributs originaux et modifiés pour l'agrégat vide, après apprentissage par maximum de vraisemblance.

semigramme, alors qu'elle obtenait des résultats similaires avec le modèle MCRF. Par exemple, le taux d'alignement à 50 ms est ici égal à 80,5% sur la base MAPS, contre 81,4% pour la représentation SGQ. De même, l'imprécision moyenne pondérée est de 74 ms contre 65 ms sur le corpus RWC-pop. Une explication tient en la plus grande sensibilité au bruit du spectrogramme. De ce fait, un poids moindre est accordé à l'attribut d'agrégat lors de l'apprentissage (effectué avec le modèle MCRF). En phase de test, le système exploitant cette représentation favorisera alors le modèle temporel. Or, les *a priori* de durées de notre base de données sont intentionnellement imprécis. Une solution à ce problème pourrait être d'effectuer un apprentissage spécifique des paramètres du modèle temporel, en fonction de la fonction d'observation utilisée. Cependant, un tel apprentissage serait d'une très grande complexité.

5.5.3 Modèles HTCRF

Dans une dernière série d'expériences, nous appliquons les fonctions d'observation estimées à un alignement par le modèle HTCRF. Comme précédemment, le paramètre de tempo est fixé à la valeur $\mu_3 = \frac{1}{10}$. Les paramètres de la fonction de transition sont ici encore estimés grâce à une recherche sur une grille de valeurs. Les valeurs retenues sont $\gamma_\mathrm{d} = 20$ et $\gamma_\mathrm{t} = 200$.

Nous avons constaté qu'avec ce modèle, l'utilisation des attributs modifiés pour l'agrégat vide permettait d'obtenir de meilleurs résultats, et cela pour toutes les fonctions d'observation. Cela s'explique par la grande diversité des observations que représente cet agrégat. De ce fait, l'exploitation de l'attribut original peut mener, selon les morceaux, à une sous-estimation ou une surestimation de la probabilité de cette étiquette. La modification de l'attribut permet de lui affecter une fonction d'observation peu informative. La probabilité de l'agrégat vide est alors déterminée principalement par les contraintes temporelles du modèle, qui sont très fiables dans le cas du HTCRF.

Les résultats des alignements avec ces attributs modifiés sont présentés en table 5.5. Le modèle temporel très précis utilisé diminue ici les différences entre les fonctions d'observation testées. On peut alors constater que les transformations $\mathbf{W}$ estimées par apprentissage ne donnent plus systématiquement les meilleurs résultats. Par exemple, pour le chromagramme CGZ, le taux d'alignement à 100 ms le plus haut sur le corpus MAPS est obtenu avec la transformation heuristique (97,8%). Cela peut être expliqué par la différence entre les deux corpus. En effet, l'apprentissage mène à une valeur de $\mathbf{W}$ qui opère un compromis entre les timbres des deux parties de l'ensemble d'apprentissage. Les gabarits théoriques semblent alors plus adaptés au corpus MAPS. Cependant, les deux stratégies d'apprentissage permettent une augmentation des scores sur le corpus RWC-pop (94,8% et 95,2% contre 94,6%). Une autre cause possible est la différence entre les longueurs des notes telles qu'indiquées par la partition et leurs durées effectives, comme expliqué en section 2.4.1. En raison de ce phénomène, causé notamment par la réverbération, des notes qui « devraient » être éteintes (d'après la partition) peuvent se superposer aux agrégats suivants. De ce fait, le contenu de certains agrégats de la partition peut ne pas correspondre à l'enregistrement. Dans ce cas, il est pénalisant d'utiliser un modèle d'observation trop discriminant.

Par ailleurs, contrairement au cas du SMCRF, la représentation en spectrogramme

Représentation en spectrogramme (SP) :

	Corpus MAPS			Corpus RWC-pop		
Apprentissage	H(m)	MD(m)	MV(m)	H(m)	MD(m)	MV(m)
TAMP (θ=300 ms)	99.4	99.5	**99.6**	**99.7**	99.6	**99.7**
TAMP (θ=100 ms)	98.0	98.0	**98.2**	95.9	95.9	**96.0**
TAMP (θ=50 ms)	**91.3**	90.1	90.9	**86.5**	86.0	86.1
IMP (ms)	**24**	25	**24**	**28**	**28**	**28**
CCMP	11.5	11.4	**11.2**	**22.0**	22.5	22.3

Représentation en semigramme SGQ :

	Corpus MAPS			Corpus RWC-pop		
Apprentissage	H(m)	MD(m)	MV(m)	H(m)	MD(m)	MV(m)
TAMP (θ=300 ms)	99.4	99.5	**99.6**	99.6	99.5	**99.7**
TAMP (θ=100 ms)	97.8	97.9	**98.3**	96.1	96.3	**97.4**
TAMP (θ=50 ms)	89.6	89.6	**91.2**	81.8	82.2	**84.3**
IMP (ms)	26	26	**24**	34	33	**31**
CCMP	12.1	12.2	**11.2**	28.5	29.0	**27.5**

Représentation en chromagramme CGZ :

	Corpus MAPS			Corpus RWC-pop		
Apprentissage	H(m)	MD(m)	MV(m)	H(m)	MD(m)	MV(m)
TAMP (θ=300 ms)	**99.5**	99.3	99.4	99.1	99.2	**99.5**
TAMP (θ=100 ms)	**97.8**	97.4	97.4	94.6	94.8	**95.2**
TAMP (θ=50 ms)	**88.4**	86.9	87.5	78.4	77.7	**78.7**
IMP (ms)	**27**	29	28	**37**	**37**	**37**
CCMP	**12.7**	13.2	13.1	30.3	30.9	**30.1**

TABLE 5.5 – Résultats des alignements avec le modèle HTCRF, pour les différents attributs d'agrégat. Dans ces expériences, tous les systèmes utilisent les attributs modifiés pour l'agrégat vide.

semble conduire ici à des alignements légèrement plus précis que le semigramme sur le corpus MAPS, avec une imprécision moyenne pondérée de 28 ms contre 31 ms. On peut pour cela avancer la même explication que précédemment : les systèmes exploitant le spectrogramme accordent plus d'importance aux contraintes temporelles, qui sont ici très fiables. De plus, l'exploitation d'une transformée à Q constant peut limiter la précision temporelle de la représentation SGQ, en raison de la taille importante des fenêtres d'analyse en basses fréquences.

Le poids plus important donné à la fonction de transition explique encore les meilleurs résultats de l'apprentissage MV sur la stratégie MD pour la représentation en chromagramme (13,1% de cout de classification moyen pondéré contre 13,2% sur MAPS et 30,1% contre 30,9% sur RWC-pop).

De façon générale, l'apprentissage par maximum de vraisemblance conduit donc à une amélioration de la qualité moyenne des alignements sur les deux bases de données par rapport à l'utilisation d'attributs heuristiques.

5.6 Conclusion

Une étude approfondie de l'attribut d'agrégat utilisé par nos systèmes d'alignement a été menée dans ce chapitre. Nous avons tout d'abord défini cet attribut à partir d'une transformation linéaire de la partition vers le domaine des observations acoustiques. Cela permet d'appliquer le même formalisme pour plusieurs fonctions de dissimilarité, ainsi que cinq différentes représentations temps-fréquence de l'audio. Parmi les paramètres testés, la divergence de Kullback-Leibler symétrisée apparait comme un bonne mesure de dissimilarité et deux représentations, en spectrogramme et en semigramme conduisent aux alignements les plus précis.

Deux stratégies d'apprentissage sont alors proposées pour l'estimation de la transformation linéaire optimale. La première utilise le critère du *minimum de divergence* afin de maximiser l'attache aux données. La seconde tire parti d'un modèle discriminatif d'alignement par CRF, en exploitant le critère du *maximum de vraisemblance*. Nos expériences mettent en valeur les améliorations induites par l'optimisation de la transformation. En effet, dans presque toutes nos expériences, les alignements obtenus sont plus précis que ceux qui utilisent une transformation heuristique. Les quelques exceptions observées concernent le modèle HTCRF, où le modèle temporel devient dominant par rapport à la fonction d'observation. Dans le cas des représentations en spectrogramme et en semigramme, la stratégie discriminative parait en outre la plus prometteuse, car elle mène à de meilleurs performances que l'apprentissage par *minimum de divergence*, alors que le modèle CRF exploité dans la phase d'apprentissage n'est pas le même que ceux utilisés pour le décodage.

Chapitre 6

L'Alignement dans le monde réel : améliorations pratiques

Dans les chapitres précédents, nous avons présenté la structure des modèles CRF ainsi que l'apprentissage de certains paramètres des attributs. Cependant, dans la perspective de tâches d'alignement audio-sur-partition réalistes, certains ajustements peuvent être apportés. Nous considérons dans ce chapitre trois questions qui peuvent se poser dans l'utilisation pratique de nos modèles. La première est celle de la prise en compte de changements de structure entre la partition et l'interprétation. La deuxième concerne la diminution de la complexité du décodage des modèles CRF, afin de réduire les besoins en mémoire et puissance de calcul de l'alignement. Enfin, nous nous intéressons à des considérations de *scalabilité* en comparant les différentes valeurs du compromis performance/complexité liés aux modèles proposés. Les travaux décrits dans ce chapitre ont été effectués avant ceux du chapitre précédent. C'est pourquoi toutes les expériences utilisent les descripteurs décrits dans la partie 4.2 (vecteurs de chroma CGZ).

6.1 Robustesse aux changements de structure musicale

6.1.1 Modification de la fonction de transitions

Dans les modèles présentés au chapitre 4, les seules transitions autorisées sortant d'un agrégat conduisent à l'agrégat suivant dans la partition. De ce fait, la structure de la séquence d'agrégats décodée est forcément la même que celle indiquée par la partition. Or, pour une prise en compte de variations possibles (omission d'une reprise, ajout d'un refrain supplémentaire...), cette contrainte n'est pas souhaitable. Nous proposons donc de modifier la fonction de transition afin d'autoriser les changements structurels entre la partition et l'interprétation.

Pour cela, nous supposons connu l'ensemble des positions dans la partition où des « sauts » peuvent intervenir. Dans le cas de la musique classique, cet ensemble est souvent indiqué dans la partition, puisqu'il correspond aux positions des barres de reprise et des autres symboles de répétition, comme représenté figure 6.1. Il peut aussi être donné par le

résultat d'une analyse structurelle de la partition (par exemple celle de Paulus et Klapuri [2009]), en prenant en compte les frontières des sections détectées. De ce fait, nous limitons les variations aux ajouts et omissions de sections, réduisant ainsi les chances d'obtenir des sauts erratiques dans le cas de fausses notes. Nous appelons ainsi les positions de sauts possibles « frontières de sections ».

Les frontières de sections sont caractérisées par l'ensemble noté $\mathcal{F}$, qui contient les agrégats de *début de section*. Les sauts entre sections sont alors autorisés, en modifiant le potentiel ψ_{c} (défini en 4.1.2) régissant les transitions entre agrégats des modèles SMCRF et HTCRF. Une valeur ϱ est associée aux sauts, de manière à les pénaliser par rapport aux transitions « classiques », qui reçoivent la valeur 1. Ce potentiel s'écrit donc :

$$\psi_{\mathrm{c}}(C_n, C_{n-1}, D_n) = \begin{cases} 1 & \text{si } D_n = 1 \text{ et } C_n = C_{n-1} + 1 \\ \varrho & \text{si } D_n = 1 \text{ et } C_n \neq C_{n-1} + 1 \text{ et } (C_{n-1}+1, C_n) \in \mathcal{F}^2 \\ \mathbf{1}_{\{C_n = C_{n-1}\}} & \text{sinon.} \end{cases} \tag{6.1}$$

De la même façon, la fonction de transition ψ^{M} du modèle markovien, initialement définie dans l'équation (4.1), devient :

$$\psi^{\mathrm{M}}(X_n, X_{n-1}) = \begin{cases} \lambda_0(C_{n-1}) & \text{si } C_n = C_{n-1} \text{ et } A_n \leq A_{n-1} \\ \lambda_1(C_{n-1}) & \text{si } C_n = C_{n-1} + 1 \text{ et } A_n = \mathbf{1}_{\dot{\mathcal{C}}}(C_n) \\ \varrho\lambda_1(C_{n-1}) & \text{si } C_n \neq C_{n-1} + 1 \text{ et } A_n = \mathbf{1}_{\dot{\mathcal{C}}}(C_n) \text{ et } (C_{n-1}+1, C_n) \in \mathcal{F}^2 \\ 0 & \text{sinon.} \end{cases} \tag{6.2}$$

Ce sont ici les seules modifications apportées aux différents modèles.

6.1.2 Influence sur la précision d'alignement

Une expérience a été menée afin de mesurer l'influence de cette prise en compte des changements de structure sur les alignements obtenus. Pour cela, la structure des partitions a été modifiée. Pour chaque morceau, une séquence arbitraire de 8 mesures a été dupliquée dans la partition, afin de simuler une indication de reprise non suivie par l'interprète. De plus, lorsque la partition fait apparaitre la répétition exacte d'une séquence d'au moins 4 mesures, la deuxième instance est supprimée, afin de simuler une reprise supplémentaire dans l'interprétation.

Des alignements sont alors calculés sur les partitions exactes et les partitions modifiées, avec les modèles initiaux (interdisant les sauts) et les versions autorisant les variations structurelles. Pour ces derniers systèmes, l'ensemble des frontières de sections exploité pour le décodage est le même que celui utilisé pour les modifications des partitions. Le paramètre de pénalisation des sauts est fixé de manière heuristique à la valeur $\varrho = \frac{1}{2}$. Dans ces expériences, la méthode d'évaluation doit être légèrement modifiée. En effet, à cause des répétitions introduites, certaines positions dans l'enregistrement peuvent correspondre à plusieurs endroits de la partition. On considèrera alors qu'un agrégat de l'enregistrement est correctement aligné s'il est décodé comme l'un des agrégats correspondants dans la partition. Les taux d'alignement obtenus sont regroupés dans le tableau 6.1. Notons que les chiffres sont calculés sur la base de test entière (MAPS et RWC-pop).

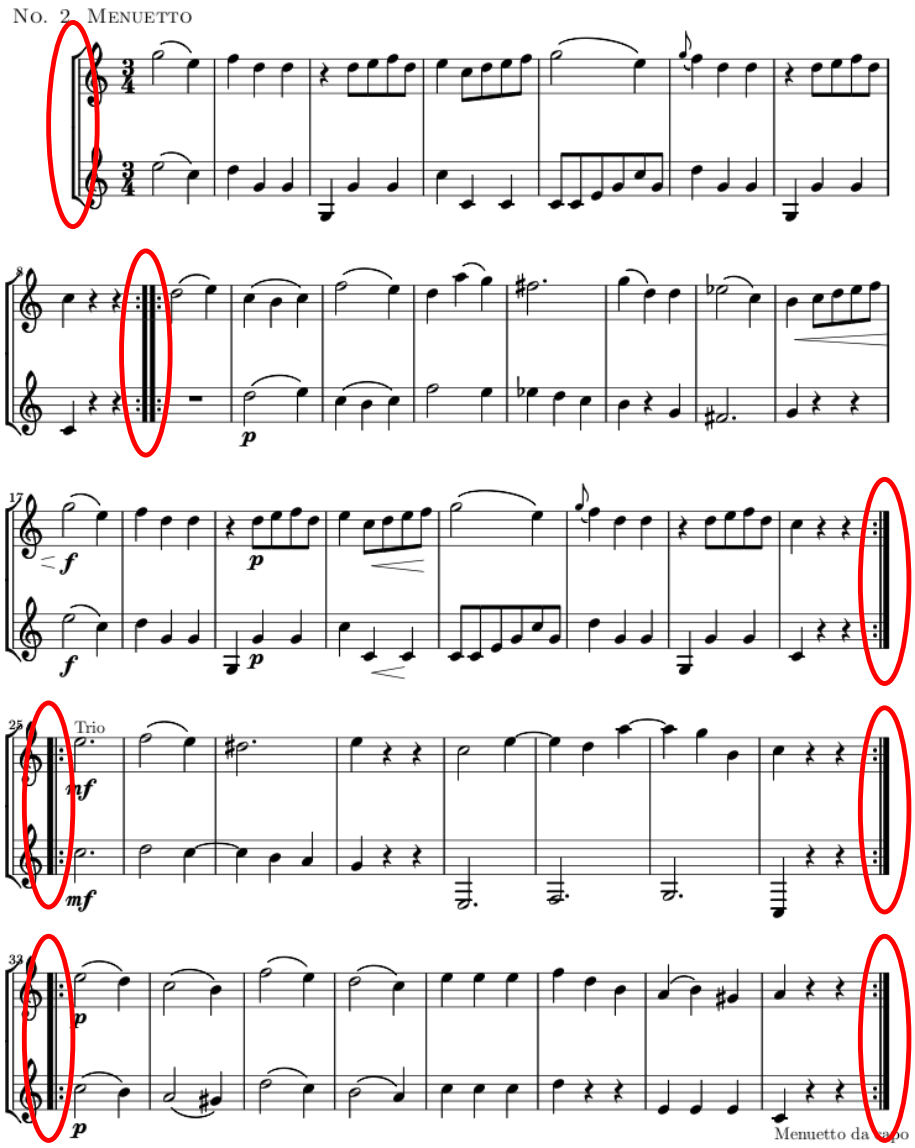


Figure 6.1 – Frontières de sections indiquées dans une partition (ici le duo pour cor KV 487 n°2 de Mozart). Ces frontières correspondent aux symboles de reprises, ainsi qu'un début du morceaux.

On observe ici deux tendances différentes sur les deux corpus utilisés. Sur RWC-pop, on peut remarquer une baisse des taux d'alignement d'environ 2 points pour les alignements sur partitions exactes, avec les systèmes autorisant les sauts. Comme pour le problème de « mauvaise répétition » exposé au chapitre 4.4.2, cela est dû aux quelques morceaux où les gabarits correspondent mal aux observations. Dans ces pièces, le chemin d'alignement a tendance à sauter une ou plusieurs section(s) et à rester une longue durée dans l'agrégat de fin. Cet agrégat représente le silence ou le bruit de fin d'enregistrement et associe un score d'appariement correct aux observations bruitées par des percussions ou d'autres sons non modélisés par la fonction d'observation. Avec le modèle HTCRF, cela concerne 9 pièces sur les 45 provenant du corpus RWC-pop.

Conformément à l'intuition, la précision de tous les systèmes diminue sur ce corpus lorsque les différences structurelles sont introduites entre la partition et l'interprétation. Néanmoins, alors que les taux d'alignement des systèmes initiaux s'effondrent avec des baisses absolues de l'ordre de 15 %, ces dégradations sont relativement faibles pour les systèmes autorisant les sauts. En effet, les scores diminuent d'environ 1 % à 2 % dans ces cas.

En outre, on peut observer que les taux d'alignement du système HTCRF modifié, même avec une partition imparfaite, sont supérieurs à ceux obtenus par le modèle SMCRF initial exploitant la partition exacte. De même le SMCRF modifié reste meilleur que le modèle markovien initial.

Sur le corpus MAPS, où les observations sont beaucoup moins bruitées, les performances des systèmes ne sont pas affectées par la modification du système autorisant les sauts dans la partition. En effet, seul le modèle MCRF sans prise en compte du voisinage voit son taux d'alignement diminuer faiblement (-0.2 % avec la partition exacte). De plus, aucun des systèmes autorisant ces sauts ne voit ses performances significativement réduite par l'introduction des variations structurelles dans la partition. Une telle prise en compte des modifications structurelle semble donc efficace.

6.2 Diminution de la complexité du décodage par élagage hiérarchique

Une deuxième préoccupation importante dans l'utilisation pratique de nos systèmes concerne la complexité de l'alignement. En effet, les modèles CRF utilisés comportent un grand nombre d'étiquettes. Or, un décodage exact par l'algorithme de Viterbi, détaillé au chapitre 4.3, nécessite l'exploration de toutes les combinaisons d'étiquettes à chaque trame de l'enregistrement, ce qui est trop couteux dans le cas des modèles SMCRF et HTCRF. Pour certains morceaux très longs (de l'ordre d'une dizaine de minutes), même le modèle markovien nécessite une capacité-mémoire supérieure à celle des ordinateurs dont nous disposons.

Pour une réduction de la complexité du décodage, on utilise donc en général des stratégies d'*élagage*. L'élagage consiste à réduire le nombre d'étiquettes explorées et à effectuer ainsi un décodage approché. Une des méthodes les plus classiques est la recherche par faisceaux, qui maintient uniquement les hypothèses les plus « prometteuses » à chaque

Corpus MAPS :

Sauts	Interdits		Autorisés	
Partition	Exacte	Modifiée	Exacte	Modifiée
MCRF $\nu = 0$	93.1	76.6	92.9	93.0
MCRF $\nu = 50$	94.9	78.2	94.9	94.9
SMCRF $\nu = 0$	97.7	77.1	97.9	97.8
SMCRF $\nu = 50$	97.9	77.4	98.0	98.0
HTCRF $\nu = 0$	99.3	80.8	99.4	99.3
HTCRF $\nu = 50$	99.3	80.9	99.4	99.4

Corpus RWC-pop :

Sauts	Interdits		Autorisés	
Partition	Exacte	Modifiée	Exacte	Modifiée
MCRF $\nu = 0$	85.2	75.6	83.9	83.4
MCRF $\nu = 50$	88.0	76.8	86.0	85.8
SMCRF $\nu = 0$	94.2	79.9	92.4	91.6
SMCRF $\nu = 50$	93.9	80.0	90.7	90.3
HTCRF $\nu = 0$	99.2	84.9	97.4	96.7
HTCRF $\nu = 50$	99.2	85.0	97.1	96.4

TABLE 6.1 – Robustesse aux changements de structures : taux d'alignement mesurés avec le seuil $\theta = 300$ ms.

itération. Les valeurs utilisées pour classer les hypothèses sont déduites des scores partiels accumulés par l'algorithme de décodage (dans notre cas, les valeurs $\hat{p}_n\left(x_n|\mathbf{Y}_{1:N}\right)$ définies au chapitre 4.3.1). Cette stratégie est, par exemple, utilisée par Raphael [2006]. Cette méthode est causale (elle exploite uniquement les observations passées) et peut donc être utilisée par des systèmes en temps réel. En revanche, cette propriété de causalité n'est pas forcément un avantage dans le cas d'un alignement hors ligne. En effet, en considérant uniquement les observations passées, la recherche par faisceaux présente le risque de supprimer certains chemins d'alignement dont les scores partiels sont trop faibles, même si ces chemins reçoivent une probabilité élevée, sachant les observations complètes.

Cela peut se produire par exemple dans le cas d'observations correspondant mal aux gabarits exploités. Alors, les chemins partiels passant par les agrégats « bruités » peuvent se voir supprimés, même si les observations suivantes permettent de rétablir une haute probabilité *a posteriori*.

6.2.1 Principe : utilisation d'une structure hiérarchique

Pour remédier à ce problème de la recherche par faisceaux, nous proposons un algorithme d'élagage tenant compte de la totalité de l'interprétation. Cet algorithme utilise une approche hiérarchique, inspirée de la méthode FastDTW présentée par Salvador et Chan [2004]. Le principe est de rechercher tout d'abord un alignement grossier peu cou-

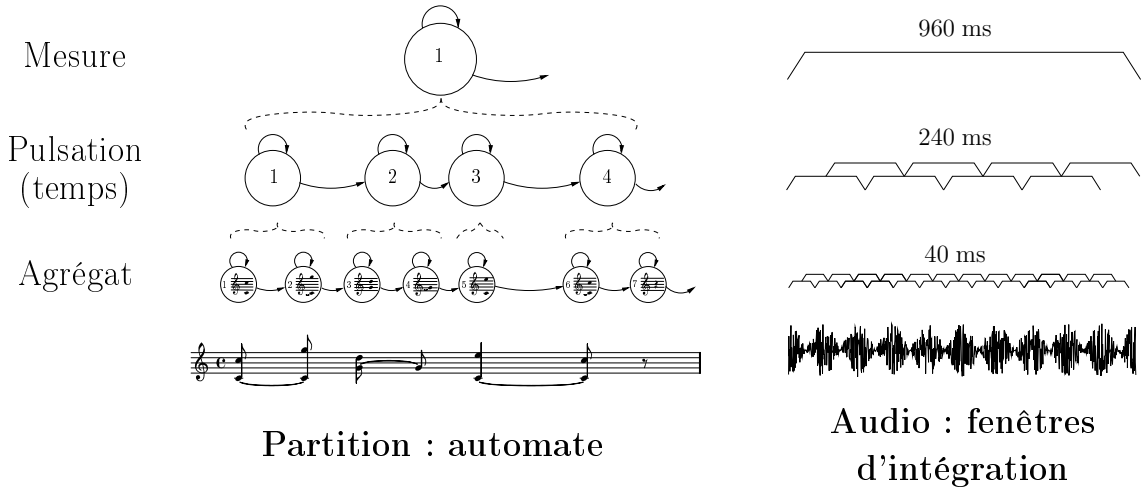


FIGURE 6.2 – Automate des étiquettes et fenêtres d'intégration de l'audio (sur lesquelles sont calculées les observations) aux trois niveaux de hiérarchie considérés. Les éventuels étiquettes de phase, occupation et tempo ne sont pas représentées.

teux et d'utiliser ensuite ce résultat pour réduire l'espace de recherche dans l'alignement à un niveau plus fin. Cette procédure nécessite donc la définition de plusieurs niveaux de précision et peut être répétée de façon hiérarchique autant de fois qu'il y a de niveaux.

Pour définir ces niveaux, nous tirons parti de structures musicales indiquées par la partition, en l'occurence les *pulsations* et les *mesures*. Moyennant la segmentation éventuelle d'agrégats durant plusieurs pulsations, les trois niveaux obtenus forment une structure temporelle hiérarchique de la partition. Trois alignements successifs sont donc calculés, pour lesquels les étiquettes représentent respectivement les mesures, pulsations et agrégats.

Les étiquettes de niveaux supérieurs désignent des unités temporelles plus longues. Le nombre d'étiquettes est donc plus petit, réduisant ainsi d'autant la complexité. Pour la même raison, les observations utilisées peuvent être extraites de fenêtres temporelles plus longues et leur fréquence d'échantillonage peut être réduite, conduisant à un nombre plus faible de trames. La figure 6.2 illustre les automates des étiquettes et l'extraction des observations pour les trois niveaux de précision considérés.

6.2.2 Déroulement de l'algorithme

L'algorithme débute au niveau *mesure*. Soit $\tilde{\mathbf{Y}}_{1:\tilde{N}}$ la séquence d'observations considérée à ce niveau, où $\tilde{N}$ est la longueur de cette séquence. Un CRF est construit afin de modéliser les relations entre observations et étiquettes, comme présenté au chapitre 4. Soient $\tilde{\psi}$ et $\tilde{\phi}$ les fonctions de transition et d'observation correspondantes. Ce modèle $\mathcal{M}$ permet d'estimer pour chaque mesure m et chaque trame $\tilde{n}$ la « probabilité de chemin maximale » au niveau mesure

$$\hat{p}_{\tilde{n}}(m) = \max_{\substack{\mathbf{M}_{1:\tilde{N}} \\ M_{\tilde{n}}=m}} \left\{ P(\mathbf{M}_{1:\tilde{N}} | \tilde{\mathbf{Y}}; \mathcal{M}) \right\} \tag{6.3}$$

où $\mathbf{M}_{1:\tilde{N}} = M_1, \ldots, M_{\tilde{N}}$ désigne la séquence d'étiquettes de mesures. Cette valeur peut être calculée par une extension de l'algorithme de Viterbi, auquel est ajoutée une phase de « retour en arrière » (*backtracking*). Cette procédure peut aussi être décrite comme une variante de l'algorithme *forward-backward* [Rabiner, 1989], où l'opération de somme des scores partiels est remplacée par une maximisation.

Si l'on suppose que ce modèle au niveau mesure est cohérent avec le modèle de bas niveau complet, et donc que les valeurs $\hat{p}_{\tilde{n}}(m)$ sont proches des probabilités « réelles », alors on peut utiliser ces valeurs pour supprimer les hypothèses de faible probabilité.

Pour chaque fenêtre n, on classe les étiquettes m d'après les valeurs de $\hat{p}_{\tilde{n}}(m)$. Puis, on conserve un nombre $\bar{K}$ fixé (et petit) de mesures qui seront considérées au niveau inférieur. Les autres hypothèses sont supprimées. La valeur de $\bar{K}$ est fixée de façon adaptative, en fonction des « probabilités de chemin maximales » : elle est définie comme le plus petit nombre tel que tous les chemins dont la probabilité est supérieure à un certain seuil sont conservés. Ce seuil est fixé à une fraction $\frac{1}{\eta}$ du maximum global de la probabilité des chemins. Le paramètre η contrôle donc le compromis entre la diminution de complexité et la perte d'optimalité. Ce processus d'élagage est illustré par la figure 6.3.

La même procédure est réitérée au niveau inférieur, où seuls les étiquettes sélectionnées sont explorées. D'autres hypothèses sont alors supprimées. Enfin, l'alignement au niveau le plus précis est recherché, parmi les étiquettes restantes.

Notons qu'il est possible d'exploiter les probabilités des étiquettes $P(M_{\tilde{n}} = m|\tilde{\mathbf{Y}})$ à la place des probabilités de séquences $\hat{p}_{\tilde{n}}(m)$ comme critère d'élagage. Cependant, nous pensons que ces derniers scores sont plus pertinents pour notre problème. En effet, la probabilité marginale d'une étiquette est la somme des probabilités de tous les chemins d'alignement passant par cette étiquette. Ainsi, une étiquette correspondant à un grand nombre de séquences, chacune de probabilité moyenne, peut être favorisée par rapport à une hypothèse contenant une séquence isolée, même si sa probabilité est élevée.

6.2.3 Variante pour partition parfaite

Une variante de cet algorithme est possible lorsque la partition est supposée exacte, sans différence structurelle avec l'interprétation. Dans ce cas, l'automate des étiquettes est « gauche-droite », c'est-à-dire sans cycle orienté, et les sauts sont interdits. On peut donc définir une relation d'ordre totale par :

$$x_1 \leq x_2 \text{ ssi il existe un chemin de } x_1 \text{ vers } x_2.$$

Une notion de proximité entre les étiquettes est donc exploitable. Le principe de l'élagage est alors de conserver les étiquettes autour du chemin d'alignement décodé à chaque niveau. On définit pour cela les « étiquettes admissibles les plus lointaines » $\hat{m}^-_{\tilde{n}}$ et $\hat{m}^+_{\tilde{n}}$ à chaque trame $\tilde{n}$:

$$\hat{m}^-_{\tilde{n}} = \min\left\{m \mid \hat{p}_{\tilde{n}}(m) \geq \frac{P(\hat{\mathbf{m}}|\mathbf{y})}{\eta}\right\} \tag{6.4}$$

$$\hat{m}^+_{\tilde{n}} = \max\left\{m \mid \hat{p}_{\tilde{n}}(m) \geq \frac{P(\hat{\mathbf{m}}|\mathbf{y})}{\eta}\right\}, \tag{6.5}$$

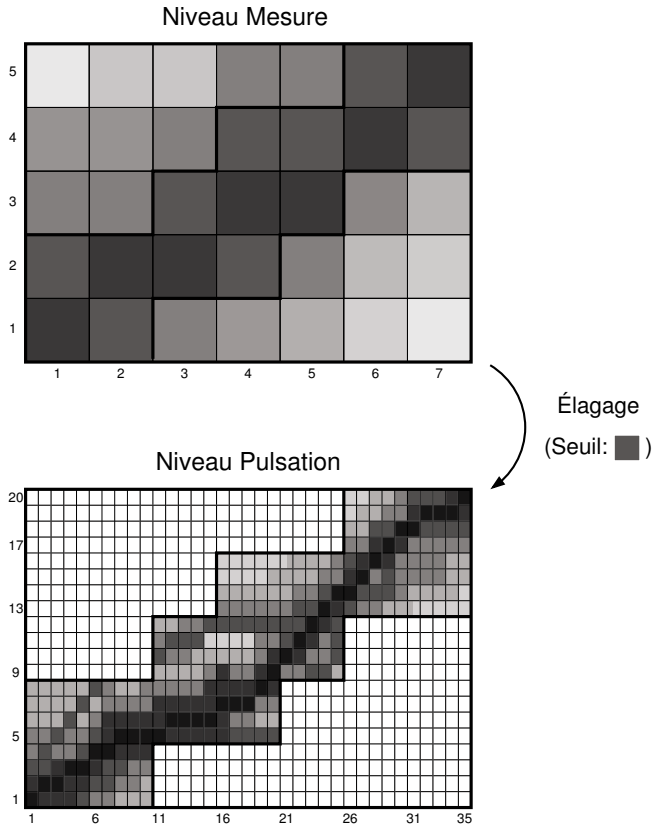


FIGURE 6.3 – Principe de l'algorithme d'élagage hiérarchique (première étape). Les niveaux de gris représentent les valeurs $\hat{p}_{\tilde{n}}(m)$ de l'équation (6.3). Au niveau inférieur (pulsation), seul le domaine délimité par les lignes est exploré.

où $\hat{\mathbf{m}}$ est la séquence d'étiquettes de mesures de plus grande probabilité. Comme précédemment, le « seuil d'élagage » est fixé à une fraction $\frac{1}{\eta}$ de la probabilité maximale. On définit alors les *rayons de tolérance* δ^- et δ^+ comme le nombre maximal d'étiquettes séparant $\hat{m}_{\tilde{n}}^-$ de $\hat{m}_{\tilde{n}}$ et $\hat{m}_{\tilde{n}}$ de $\hat{m}_{\tilde{n}}^+$, respectivement.

Les rayons de tolérance déterminent l'ensemble des étiquettes autour du chemin d'alignement décodé qui seront conservées dans les niveaux inférieurs.

6.2.4 Modèles de niveaux supérieurs

Puisque le but des alignements aux niveaux pulsation et mesure est de diminuer la complexité globale de l'alignement, l'utilisation de modèles peu complexes est préférable. Nous choisissons donc pour ces deux niveaux un modèle markovien ne prenant pas en compte l'information de voisinage (voir chapitre 4), où les étiquettes représentent uniquement l'index de la pulsation et de la mesure courante, respectivement. Les alignements aux différents niveaux sont effectués de manière indépendante. Ainsi, n'importe quel modèle

peut ensuite être utilisé pour l'alignement au niveau le plus fin.

Aux niveaux supérieurs, les observations exploitées sont des vecteurs de chroma, calculés comme la moyenne des vecteurs de chroma originaux sur les fenêtres d'intégration considérées (voir figure 6.2). Les observations de détection d'attaques ne peuvent pas être utilisées, car elles caractérisent des unités (les phases) trop courtes par rapport à l'échelle temporelle des niveaux supérieurs. L'utilisation d'observations moyennées est justifiée par des considérations musicales. En effet, l'harmonie (et donc l'information de chroma) est en général homogène sur la durée d'une pulsation et souvent ne change pas à l'intérieur d'une mesure.

Les paramètres des fenêtres d'intégration sont choisis en relation avec les tempos possibles. Si le tempo est considéré comme stable, l'utilisation d'un algorithme d'estimation du tempo peut être envisagé pour fixer les longueurs et les pas d'avancement. Cependant, une telle stratégie peut comporter un risque en présence de variations importantes et soudaines, car la fréquence d'échantillonnage des observations doit être au moins aussi grande que la fréquence des pulsations (ou mesures) « réelles ». En effet, si le tempo de l'interprétation est plus rapide que la fréquence d'observation des chromas intégrés, certaines étiquettes de pulsations ne pourront pas être atteintes par l'algorithme, résultant en un élagage systématique de ces chemins d'alignement.

Dans notre cas, nous souhaitons ne pas faire d'hypothèses limitantes sur les évolutions possibles du tempo. Ainsi, les paramètres des fenêtres d'intégration sont choisis de façon à admettre tous les tempos acceptables. En pratique, nous fixons la longueur des fenêtres à 240 ms pour le niveau pulsation, correspondant à un tempo de 250 bpm. Un recouvrement de 1/3 est utilisé, résultant en un pas d'avancement de 160 ms. Au niveau mesure, la longueur et le pas d'avancement des fenêtres d'intégration sont multipliées par le nombre de pulsations par mesure. Ce nombre est indiqué dans la partition et s'il change au cours du morceau, la plus petite valeur est choisie. Par exemple, dans le cas d'une mesure à 4 temps, la longueur d'une fenêtre sera 960 ms et le pas d'avancement 480 ms.

La fonction d'observation associée à ces chromas a la même forme que celle définie au chapitre 4.2.1. Néanmoins, comme le modèle comporte une seule étiquette par pulsation (ou mesure), les observations sont comparées à des « gabarits intégrés » correspondant à ces étiquettes. Un tel gabarit de niveau supérieur est calculé comme la moyenne des gabarits de tous les agrégats contenus dans l'unité représentée (pulsation ou mesure), pondérée par les durées en pulsations de ces agrégats.

6.2.5 Expériences

Nous évaluons l'impact de la stratégie d'élagage proposée sur la complexité de décodage. Dans ces expériences, nous mesurons certains indicateurs de complexité sur le modèle MCRF sans prise en compte du voisinage $\nu = 0$, pour l'alignement d'une base constituée de 119 morceaux de nos deux corpus. Les résultats sont compilés en table 6.2 pour la version classique, et en table 6.3 pour la variante dédiée aux partitions sans erreurs.

Plusieurs valeurs sont présentées. L'espace de recherche est la proportion moyenne d'étiquettes qui sont effectivement considérées dans le processus de décodage aux niveaux pulsation et agrégat. Le temps d'exécution comprend les trois phases d'alignement, mais ne

Élagage	Paramètre	Espace de recherche		Temps d'exécution en s (% temps réel)	Erreurs (nb)
		Pulsations	Agrégats		
Aucun	–	–	100%	3489 (12%)	0
RF	$\bar{K}=600$	–	27.24%	1330 (4.4%)	1
$\bar{K}$ constant	$\bar{K}=60$	0.85%	17.78%	872 (2.9%)	0
$\bar{K}$ adaptatif	$\eta=1000$	0.32%	13.38%	813 (2.7%)	0
	$\eta=100$	0.26%	10.62%	643 (2.1%)	0
	$\eta=50$	0.24%	9.74%	617 (2.0%)	0
	$\eta=20$	0.21%	8.52%	567 (1.9%)	1

Table 6.2 – Performance du système MCRF avec plusieurs stratégies d'élagage. L'espace de recherche est le rapport du nombre d'étiquettes explorées, sur le nombre d'étiquettes total du modèle au plus bas niveau (égal à $0,15$ % au niveau mesure pour toutes les versions de l'élagage hiérarchique). Le temps d'exécution est indiqué en secondes et en proportion du temps réel.

prend pas en compte l'extraction des descripteurs. L'implémentation des algorithmes a été réalisée en MATLAB et exécutée sur un processeur Intel Core2 cadencé à $2,66$ GHz avec 3 Go de RAM, sous linux. Le nombre d'*erreurs d'élagage* est aussi présenté. Une erreur est produite lorsque le chemin d'alignement « réel » est supprimé par le processus d'élagage.

Différentes versions de notre algorithme sont testées, utilisant différentes valeurs du paramètre η. À titre de comparaison, nous présentons aussi les résultats obtenus par un système exploitant la stratégie de recherche par faisceaux (RF), où à chaque étape les $\bar{K}$ hypothèses les « plus prometteuses » sont maintenues. Nous comparons de même une version de notre méthode hiérarchique où le nombre d'étiquettes maintenues est fixé comme paramètre. Le système présenté utilise la plus petite valeur de ce paramètre pour laquelle aucune erreur d'élagage n'est observée.

Les résultats montrent les avantages de la méthode proposée. En effet, dans les deux versions de notre stratégie de décodage, l'espace de recherche et le temps d'exécution sont tous deux inférieurs aux systèmes de référence. Pour la version originale, aucune erreur d'élagage n'intervient jusqu'à une valeur $\eta = 50$ du paramètre. Le temps d'exécution correspondant est alors d'environ $1/5$ du système sans élagage (617 s contre 3489 s). On observe que jusqu'à la valeur $\eta = 20$, les alignements obtenus sont équivalents à ceux obtenus par le système initial. Cela indique que l'élagage ne dégrade pas la précision des alignements.

En comparaison, la stratégie de recherche par faisceaux (RF) est moins fiable. En effet, pour la valeur du paramètre $\bar{K} = 600$ présentée, la complexité du décodage est supérieure à celle de notre méthode et une erreur d'élagage est observée. Dans le morceau concerné par cette erreur, les vecteurs de chroma observés correspondent mal aux gabarits théoriques et les attributs extraits présentent un biais en faveur de certains agrégats. Le système reste alors « bloqué » dans l'un de ces états dont le score est plus élevé que les autres. Le chemin d'alignement réel est donc supprimé à cause d'un score partiel trop faible dans l'algorithme de Viterbi, qui utilise uniquement les observations passées. En revanche, avec

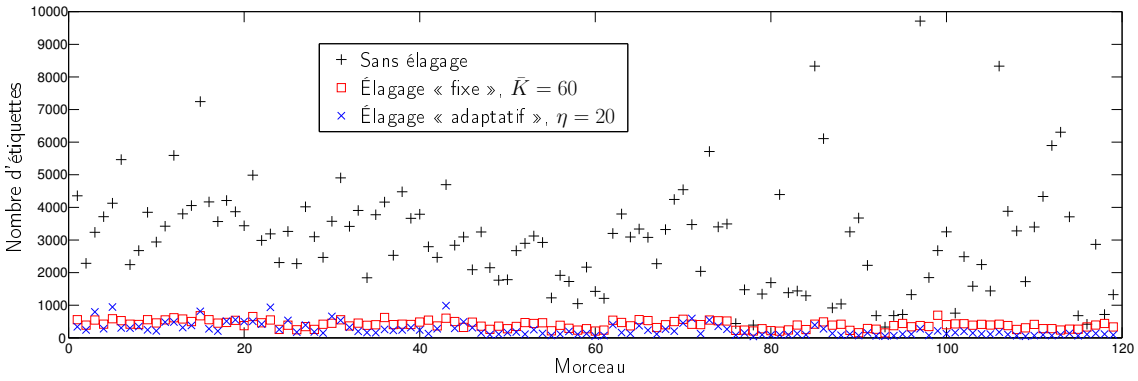


FIGURE 6.4 – Nombres moyen d'étiquettes explorées (au plus bas niveau) par morceau, pour plusieurs stratégies d'élagage.

notre méthode, la totalité du signal est considérée (bien qu'à un niveau plus grossier) et le risque de rester « bloqué » dans un état est réduit.

On peut penser que ce problème de la recherche par faisceaux serait moins important avec les autres modèles (SMCRF et HTCRF) grâce aux modèles temporels qui pénalisent les durées d'agrégats trop longues. Cependant, notre méthode présente un autre avantage comparé à la recherche par faisceaux. En effet, cette dernière stratégie nécessite, à chaque étape du décodage, la détermination des $\bar{K}$ plus grands scores parmi les chemins partiels considérés. Le cout de ce processus peut représenter une partie substantielle de la complexité du décodage, surtout lorsque la valeur du paramètre $\bar{K}$ est grande. Dans la méthode proposée, ce tri est moins couteux car il est exécuté aux niveaux supérieurs (mesure et pulsation). Quoi qu'il en soit, les deux stratégies d'élagage ne sont pas incompatibles et il est possible d'utiliser la recherche par faisceaux sur un espace qui a été préalablement réduit grâce à notre méthode hiérarchique.

On peut enfin observer l'intérêt de l'approche adaptative pour le choix du nombre d'étiquettes à maintenir dans l'élagage. La figure 6.4 compare les nombres moyens d'étiquettes après élagage avec cette approche et avec un paramètre fixe. Les deux stratégies réduisent de façon significative la complexité. On peut aussi remarquer que le nombre d'étiquettes maintenues par l'élagage ne croit pas avec le nombre d'étiquettes initial. Il est en revanche variable avec l'approche proposée, alors qu'il est globalement constant avec un paramètre $\bar{K}$ fixe. Cette stratégie adapte le nombre d'étiquettes élaguées aux données, ce qui permet d'explorer un plus grand nombre d'étiquettes dans certains morceaux « difficiles », tout en maintenant une moindre complexité dans la plupart des cas. De ce fait, l'espace de recherche et le temps d'exécution sont globalement plus faible qu'avec un paramètre fixe (10,62% et 617 s contre respectivement 17,78% et 872 s).

Les résultats donnés en table 6.3 permettent de confirmer que la variante de l'algorithme de décodage supposant la partition parfaite est plus rapide que la précédente, pour un espace de recherche équivalent. La durée du décodage sans élagage passe par exemple de 3489 s à 1517 s. En effet, la contrainte forte sur les transitions du modèle (seules deux transitions sont autorisées pour chaque étiquette) permet une implémentation plus simple

Élagage	Paramètre	Espace de recherche		Temps d'exécution en s (% temps réel)	Erreurs (nb)
		Pulsations	Agrégats		
Aucun	–	–	100%	1517 (5.0%)	0
RF	$\bar{K}=600$	–	27.24%	1330 (4.4%)	1
Rayons fixes	$\delta = 351$	4.49%	35.81%	515 (1.7%)	0
Rayons adaptatifs	$\eta = 1000$	1.07%	21.13%	567 (1.8%)	0
	$\eta = 100$	0.84%	16.97%	509 (1.7%)	0
	$\eta = 50$	0.76%	15.52%	490 (1.6%)	0
	$\eta = 20$	0.65%	13.70%	460 (1.5%)	0
	$\eta = 10$	0.56%	12.03%	435 (1.4%)	1

Table 6.3 – Performance du système MCRF avec plusieurs stratégies d'élagage, dans la variante pour partition parfaite.

du décodage, qui tire parti de la relation d'ordre totale existant entre les états.

Dans cette version, les espaces de recherche contiennent ceux de la version précédente et sont donc plus grands. En revanche, un segment « continu » de la partition est exploré ici à chaque étape du décodage, ce qui permet d'utiliser l'implémentation rapide du décodage. La recherche par faisceaux est la seule stratégie qui ne bénéficie pas de cette implémentation, car elle ne permet pas de maintenir un espace de recherche « continu ».

6.3 Considérations de *scalabilité*

Comme on l'a déjà évoqué, l'alignement musique-sur-partition peut avoir plusieurs applications, dont les exigences en terme de précision et de complexité sont différentes. Nous proposons donc dans cette section de comparer les compromis performances/complexité des différents systèmes CRF utilisés dans cette thèse.

6.3.1 Alignement rapide à un niveau grossier

Avant de présenter les résultats globaux, nous introduisons un système supplémentaire destiné à un alignement rapide. Ce système exploite la même stratégie hiérarchique que celle utilisée dans l'algorithme d'élagage proposé dans la section précédente. Le principe de ce système est alors de ne pas calculer l'alignement au niveau le plus fin, mais de le déduire de celui obtenu au niveau supérieur. De ce fait, seuls deux modèles CRF sont décodés, avec une complexité réduite.

À partir de l'alignement au niveau pulsation, les positions des frontières d'agrégats sont données par une simple interpolation entre les frontières de pulsations, sous l'hypothèse que le tempo est constant sur la durée de la pulsation. Nous appelons ce système BLCRF, pour Beat-Level CRF. La précision de ce système est médiocre, puisque les taux d'alignement avec un seuil de tolérance de 300 ms sont respectivement de $81,9$ % et $63,9$ % sur les corpus MAPS et RWC-pop. En revanche, ces alignements sont très rapides. En effet, le temps d'exécution mesuré est de seulement $0,23$ % de la durée des enregistrements. Ce

système peut donc être intéressant pour certaines applications de recherche de documents dans une grande base de données, comme la recherche de partitions correspondant à un enregistrement-requête. Dans ces tâches, il peut être avantageux d'effectuer une permière sélection rapide des documents les plus pertinents, avant d'affiner la recherche avec des systèmes plus complexes.

6.3.2 Compromis précision/complexité

Le tableau 6.4 compile des indicateurs de performance et de complexité des différents modèles présentés dans ce document. Les systèmes testés ici exploitent la méthode d'élagage proposée dans la section précédente et les résultats sont calculés sur la même base de données. Comme on le voit, les modèles les plus précis sont aussi les plus complexes. Pour une application donnée, le système le plus approprié peut alors être choisi dans ce tableau, selon la valeur voulue du compromis performances/complexité.

Il est à noter que les ordres de grandeurs de complexité en nombre d'opérations sont donnés pour le décodage par l'algorithme de Viterbi et ne prennent pas en compte le calcul de la fonction d'observation. Or, ce calcul est plus complexe dans les systèmes prenant en compte l'information de voisinage. Cela explique pourquoi les temps d'exécution des deux modèles HTCRF sont différents, alors que la complexité théorique de l'algorithme de Viterbi est identique.

On observe que parmi ces systèmes, le MCRF utilisant l'information de voisinage n'est pas intéressant, car son exécution est plus couteuse que celle du SMCRF « instantané », pour une précision moindre. De la même façon, l'utilisation du voisinage dans le modèle HTCRF n'apporte pas d'amélioration significative de performance, alors que la complexité augmente.

6.4 Conclusion

Nous avons étudié dans ce chapitre trois axes concernant la confrontation de nos systèmes d'alignement à certains problèmes d'utilisation pratique. Dans un premier temps, nous avons introduit une modification dans la fonction de transition, afin d'autoriser certaines différences structurelles entre la partition et l'enregistrement. Nous avons montré expérimentalement que cette méthode simple permet une meilleure robustesse à ces changements, avec un impact réduit sur les performances lorsqu'aucune différence n'est observée. En particulier, les alignements du corpus MAPS, où les observations sont peu bruités ne sont pas affectés par cette modification.

Nous avons ensuite proposé une méthode d'élagage hiérarchique pour la réduction de la complexité du décodage hors-ligne de nos modèles graphiques. Cette approche tire parti de la structure temporelle de la musique, qui peut se décomposer hiérarchiquement en mesures, pulsations et agrégats. Le décodage approché est alors effectué en plusieurs passes, à des niveaux de précision de plus en plus fins. Nos expériences menées avec le modèle MCRF indiquent que cette stratégie réduit la complexité en mémoire et en temps de calcul de façon plus importante que la méthode de recherche par faisceaux. En outre, une combinaison de ces deux approches est possible et pourrait encore accélérer l'alignement.

Système	Imprécision : centile		Complexité du Viterbi			Temps d'exécution
	90ème	95ème	Espace	Comparaisons	Multiplications	
BLCRF ($\nu=0$)	808 ms	1545 ms	$\tilde{Q}_B N$	$\tilde{Q}_B E_B N$	$\tilde{Q}_B E_B N$	0,23 % TR
MCRF ($\nu=0$)	303 ms	525 ms	$\tilde{Q}_C Q_A N$	$\tilde{Q}_C Q_A E_C N$	$\tilde{Q}_C Q_A E_C N$	1,5 % TR
MCRF ($\nu=50$)	255 ms	475 ms	$\tilde{Q}_C Q_A N$	$\tilde{Q}_C Q_A E_C N$	$\tilde{Q}_C(\bar{\ell} T_{\mathrm{M}} Q_T Q_A + E_C)N$	98 % TR
SMCRF ($\nu=0$)	152 ms	255 ms	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} E_C N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} E_C N$	9,4 % TR
SMCRF ($\nu=50$)	131 ms	250 ms	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}}(Q_T + E_C)N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}}(Q_T + E_C)N$	110 % TR
HTCRF ($\nu=0$)	62 ms	85 ms	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} Q_T N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} Q_T^2 E_C N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} Q_T^2 E_C N$	305 % TR
HTCRF ($\nu=50$)	62 ms	85 ms	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} Q_T N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} Q_T^2 E_C N$	$\tilde{Q}_C \bar{\ell} T_{\mathrm{M}} Q_T^2 E_C N$	394 % TR

TABLE 6.4 – Caractéristiques de performance et de complexité des systèmes proposés. La valeur du p-ième centile de l'imprécision indique que p% des attaques sont détectées avec une imprécision inférieure à cette valeur. $\tilde{Q}_C$ et E_C sont respectivement le nombre d'étiquettes d'agrégat du modèle (après élagage) et le nombre moyen de transitions entrantes dans chacune de ces étiquettes. $\bar{\ell}$ désigne la longueur moyenne en pulsations des agrégats. Les valeurs $\tilde{Q}_C$ et E_C correspondent aux pulsations.

Enfin, les compromis performance/complexité des différents modèles proposés sont confrontés, pour évaluer la scalabilité du cadre employé. Nous proposons un nouveau modèle similaire au MCRF, pour un décodage grossier mais très rapide. Nous observons alors que l'utilisation des observations de voisinage dans les modèles MCRF et HTCRF n'apparait pas intéressante, puisque dans ces deux cas, un autre modèle obtient des performances similaires à un moindre cout.

Conclusion

Dans cette thèse, le problème d'alignement temporelle musique-sur-partition est traité à l'aide de modèles graphiques discriminatifs. Ce choix d'une stratégie discriminative est motivé par l'idée de modéliser uniquement ce qui est nécessaire à la tâche, c'est-à-dire les distributions des variables cachées sachant les observations. Les modèles discriminatifs peuvent en effet être opposés aux modèles génératifs, qui modélisent le processus de création des données observées.

Ce parti pris de parcimonie a aussi été appliqué à l'estimation des paramètres de nos modèles. En effet, comme nous l'avons évoqué, une stratégie d'apprentissage automatique de tous les paramètres conjointement est inenvisageable pour des raisons de complexité. De ce fait, nous avons choisi de scinder nos systèmes en différentes parties et de considérer individuellement ces parties. En particulier, la fonction d'observation a été optimisée indépendamment du reste du modèle. D'autre part, nous nous sommes efforcés d'injecter dans nos modèles des connaissances *a priori* que l'on pouvait avoir sur le domaine. Ainsi la forme des modèles temporels est en grande partie déterminée par des considérations musicales, ce qui évite de recourir à un apprentissage automatique.

De plus, à travers notre modèle d'observation, nous avons montré qu'il était profitable d'exploiter de multiples sources d'information. Ainsi, la fonction d'observation proposée opère la fusion de plusieurs attributs issus de différentes trames (par la prise en compte des observations de voisinage) et caractérisant différents aspects du signal considéré (harmonie, attaques de notes et tempo).

Résumé des contributions

Nous avons développé dans cette thèse un cadre probabiliste discriminatif pour l'alignement temporel musique-sur-partition, grâce à des modèles graphiques de type champs aléatoires conditionnels (CRF). Nous avons vu que ce formalisme pouvait être vu comme une généralisation des modèles génératifs utilisés usuellement pour cette tâche. Il permet alors l'utilisation d'attributs plus souples que les réseaux bayésiens dynamiques, en particulier en prenant en compte des séquences recouvrantes d'observations.

Dans le chapitre 4, nous avons proposé trois structures de modèles correspondant à une modélisation de plus en plus fine (mais aussi de plus en plus complexe) des durées des étiquettes pour la couche haut niveau du modèle. Nous avons constaté une augmentation de la qualité des alignements avec la complexité du modèle temporel. La couche bas niveau présentée exploite trois types de descripteurs acoustiques, caractérisant respectivement

les agrégats (c'est-à-dire les notes jouées) de l'enregistrement, les attaques de notes et le tempo. Nous avons de plus mis en évidence le gain occasionné par l'intégration de tout un voisinage de chaque trame de l'enregistrement pour le calcul de l'attribut d'agrégat.

Nous avons alors proposé dans le chapitre 5 une formule générale pour l'attribut d'agrégat, à partir d'une transformation linéaire d'une représentation vectorielle de la partition. De cette façon, nous avons comparé l'efficacité de cinq différentes représentations temps-fréquence de l'audio en utilisant le même formalisme. D'autre part, nous avons mis en évidence deux représentations, en spectrogramme et en semigramme, qui conduisent à des alignements de grande précision. D'autre part, un apprentissage des matrices de transformation a été effectué, selon les critères du *minimum de divergence* et de *maximum de vraisemblance*. À travers nos expériences, nous avons montré que l'optimisation permet une amélioration significative des performances.

Enfin, au chapitre 6 nous avons exploré différentes stratégies pour la prise en compte de contraintes supplémentaires liées à des conditions réalistes d'alignement. Pour réduire la complexité du décodage, nous avons proposé une méthode d'élagage hiérarchique qui surpasse la stratégie de recherche par faisceaux pour le modèle markovien. Nous avons de plus étudié la robustesse du cadre proposé à certaines modifications structurelles de l'enregistrement, ainsi que les propriétés de scalabilité de nos modèles.

Perspectives

Étude approfondie de l'attribut d'agrégat

Parmi les nombreuses extensions qui pourraient être explorées dans une continuation de ce travail, un prolongement direct à notre étude de l'attribut d'agrégat serait d'examiner d'autres distances que la divergence de Kullback-Leibler. Il serait sans doute intéressant de généraliser encore la forme de la fonction d'observation. En effet, nous nous limitons dans ce travail à la considération d'un unique attribut caractérisant l'agrégat. Cependant, le cadre CRF autorise un nombre arbitrairement grand d'attributs. De fait, la divergence de Kullback-Leibler symétrique utilisée est elle-même la superposition de deux divergences. De même, la divergence est calculée comme la somme des contributions de chaque composante du vecteur d'observation. On pourrait de la même façon exploiter n'importe quelle combinaison linéaire d'attributs, où chaque attribut serait associé à une distance et à une (ou plusieurs) composante(s). Les poids relatifs pourraient alors être appris, par exemple par la stratégie du *maximum de vraisemblance*.

Nouveaux attributs

Il pourrait en outre être profitable d'incorporer des attributs supplémentaires afin de prendre en compte d'autres types d'informations dans la fonction d'observation et la fonction de transition. Un exemple simple est l'exploitation d'un attribut lié à l'énergie globale du signal, pour détecter les silences. De plus, le cadre CRF permet de concevoir des attributs exprimant des relations structurelles entre les trames, comme la similarité entre les observations. On peut par exemple imaginer un attribut de détection de « ruptures »

dans la séquence d'observation. L'exploitation d'un tel attribut dans la fonction de transition pourrait alors favoriser les changements d'agrégats lorsqu'une variation brusque est détectée dans les observations. Un tel attribut serait en effet robuste à certaines déviations entre les observations et la partition, comme les problèmes d'accord des instruments ou de justesse d'intonation. Dans le cas du modèle MCRF, on peut aussi imaginer une fonction de transition dépendant de la position du dernier pic de la fonction de détection d'attaque, pour une modélisation implicite des durées d'agrégats sans faire appel à la variable d'occupation.

Critères d'apprentissage et de décodage

Nous avons vu dans le chapitre 5 que le critère du *maximum de vraisemblance* pour l'apprentissage des paramètres du modèle n'occasionnait pas toujours les meilleurs performances. C'est pourquoi d'autres critères pourraient être examinés. On pourrait par exemple prendre en compte les potentielles imprécisions de l'annotation, en maximisant la probabilité de toutes les séquences d'agrégats situées autour de la séquence annotée (grâce à un seuil de tolérance). Le critère de *maximisation de la marge* proposé par Taskar *et al.* [2003] semble être une autre possibilité prometteuse. De la même façon, il pourrait être intéressant d'expérimenter d'autres critères de décodage, comme le critère de *minimum d'erreurs de segmentation* proposé par Raphael [1999].

Structure des modèles

Une modification de l'automate des agrégats pourrait être envisagée, afin de tenir compte des inévitables déviations existant entre les instants de fin de notes indiqués sur la partition et ceux mesurées dans le signal. Ces déviations sont notamment dûes aux différentes articulations utilisées par les musiciens et à la présence de réverbération. Les durées modélisées par la couche haut niveau seraient alors celles des intervalles entre deux attaques de notes et cette structure ferait alors apparaitre des « embranchements » entre certains de ces intervalles, correspondant aux différentes possibilités dans l'ordre des extinctions des notes.

Un autre raffinement possible des alignements concerne la détection et la correction des désynchronisations entre les différentes voix du morceaux, comme par exemple dans le cas des accords arpégés. Pour cela, une technique similaire à celle de Niedermayer et Widmer [2010a] pourrait être utilisée, consistant en une passe supplémentaire d'alignement, où la position de chaque note peut être modifiée indépendamment des autres.

Nos expériences exploitent une discrétisation plutôt grossière de l'espace des tempos possibles. Cette discrétisation donne des résultats très satisfaisants. Néanmoins, on peut imaginer une modélisation encore plus fine des valeurs du tempo en utilisant une variable continue. Un décodage exact par l'algorithme de Viterbi serait alors impossible, mais un décodage approché, par filtrage particulaire par exemple, pourrait être mis en œuvre. Une autre possibilité envisageable serait d'utiliser une quantification fine des valeurs du tempo et d'utiliser un algorithme de détection automatique du tempo pour supprimer les hypothèses les moins probables.

Adaptation de la fonction d'observation

Dans le chapitre 5, nous réalisons un apprentissage de la transformation linéaire W de la partition vers la représentation de l'audio. Or dans un modèle plus réaliste, cette transformation dépend du timbre de chaque instrument. Il est donc envisageable d'utiliser les informations d'instruments pour chaque note de la partition. Un apprentissage spécifique des instruments les plus courants pourrait permettre d'améliorer le modèle d'observation.

Une autre perspective qui semble prometteuse est d'effectuer une adaptation du modèle d'observation sur chaque morceau étudié, avec la possibilité d'estimer ainsi des transformations spécifiques aux instruments présents, dans les conditions d'enregistrement du morceau. Une telle approche, similaire à celle de Maezawa *et al.* [2011], pourrait non seulement améliorer la précision des alignements, mais aussi être exploitée directement pour une séparation des sources musicales.

On a vu que la présence de percussions (notamment d'une grosse caisse) pouvait dégrader la qualité des alignements obtenus. Il pourait donc être profitable d'effectuer une séparation des percussions afin de « nettoyer » les observations audio. Une telle séparation peut être l'objet d'un pré-traitement aveugle, mais on peut aussi imaginer qu'elle tire parti d'un premier alignement. En effet, un alignement peut fournir une décomposition des sons non percussifs sur la base des colonnes de W, qui peut alors être exploitée pour une séparation informée des autres sons.

Applications à d'autres domaines

Ce travail est axé spécifiquement sur la tâche d'alignement temporel musique-sur-partition. Cependant, il est envisageable d'adapter les approches présentées à d'autres domaines. Par exemple, un modèle temporel tiré de notre CRF à tempo caché (HTCRF) peut être exploité pour l'analyse rythmique d'un enregistrement musical. D'autres types d'alignements peuvent aussi être imaginés, parmi lesquels l'alignement de signaux de parole et du texte correspondant, ou la synchronisation de gestes issus d'une capture de mouvements.

Publications de l'auteur

C. Joder, S. Essid et G. Richard : Alignment kernels for audio classification with application to music instrument recognition. *In Actes de European Signal Processing Conf. (EUSIPCO)*, Lausanne, Suisse, 2008.

C. Joder, S. Essid et G. Richard : Temporal integration for audio classification with application to musical instrument classification. *IEEE Trans. Audio, Speech, Language Processing*, 17(1):174 –186, jan. 2009b.

C. Joder, S. Essid et G. Richard : Étude des descripteurs acoustiques pour l'alignement temporel audio-sur-partition musicale. *In Actes du Colloque GRETSI*, Dijon, France, 2009a.

C. Joder, S. Essid et G. Richard : Approche hiérarchique pour un alignement musique-sur-partition efficace. *In Actes de Compress. et Représ. des Signaux Audiovisuels (CORESA)*, p. 67–72, Lyon, France, oct. 2010a.

C. Joder, S. Essid et G. Richard : A comparative study of tonal acoustic features for a symbolic level music-to-score alignment. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP)*, Dallas, Texas, USA, p. 409–412, 2010b.

C. Joder, S. Essid et G. Richard : An improved hierarchical approach for music-to-symbolic score alignment. *In Actes de Inter. Soc. for Music Information Retrieval Conf. (ISMIR)*, p. 39–44, Utrecht, Pays-Bas, août 2010c.

C. Joder, S. Essid et G. Richard : A conditional random field viewpoint of symbolic Audio-to-Score matching. *In Actes de ACM Multimedia Conf.*, p. 871–874, Florence, Italie, oct. 2010d.

C. Joder, S. Essid et G. Richard : Hidden discrete tempo model : a tempo-aware timing model for audio-to-score alignment. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing (ICASSP)*, p. 397–400, mai 2011.

C. Joder, S. Essid et G. Richard : A conditional random field framework for robust and scalable audio-to-score matching. *IEEE Trans. Audio, Speech, Language Processing (à paraitre)*, PP, 2011.

Bibliographie

Musical instrument digital interface. www.midi.org.

Music information retrieval evaluation exchange 2006, score following task. http://www.music-ir.org/mirex/2006/index.php/Score_Following_Proposal, 2006.

Music information retrieval evaluation exchange 2010, score following task. http://www.music-ir.org/mirex/wiki/2010:Real-time_Audio_to_Score_Alignment_(a.k.a_Score_Following), 2010.

M. Alonso, G. Richard et B. David : Extracting note onsets from musical recordings. *In Actes de IEEE Inter. Conf. Multimedia & Expo*, p. 896–899, 2005.

V. Arifi, M. Clausen, F. Kurth et M. Müller : Score-pcm music synchronization based on extracted score parameters. *In* U. K. Wiil, éd. : *Computer Music Modeling and Retrieval*, vol. 3310 de *Lecture Notes in Comput. Science*, p. 193–210. Springer Berlin / Heidelberg, 2005.

M. S. Arulampalam, S. Maskell, N. Gordon et T. Clapp : A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174 –188, fév. 2002. ISSN 1053-587X.

A. Arzt et G. Widmer : Simple tempo models for real-time music tracking. *In Actes de Sound and Music Conf.*, Barcelona, Spain, 2010a.

A. Arzt et G. Widmer : Towards effective 'any-time' music tracking. *In Actes de Starting AI Researchers' Symposium (STAIRS)*, p. 24–36, Lisbon, Portugal, 2010b.

A. Arzt, G. Widmer et S. Dixon : Automatic page turning for musicians via real-time machine listening. p. 241–245, 2008.

B. Baird, D. Blevins et N. Zahler : The artificially intelligent computer performer : The second generation. *Journal of New Music Research*, 19:197–204, 1990.

B. Baird, D. Blevins et N. Zahler : Artificial intelligence and music : Implementing an interactive computer performer. *Computer Music Journal*, 17 :2:73–79, 1993.

R. Bellman : *Dynamic Programming*. Dover Publications, mars 2003. ISBN 0486428095.

J. J. Bloch et R. B. Dannenberg : Real-time accompaniment of polyphonic keyboard performance. *In Actes de Inter. Computer Music Conf.*, p. 279–289, 1985.

A. Bonafonte, J. Vidal et A. Nogueiras : Duration modeling with expanded hmm applied to speech recognition. *In Actes de Spoken Language (ICSLP)*, vol. 2, p. 1097–1100, oct. 1996.

M. A. Branch, T. F. Coleman et Y. Li : A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999.

J. C. Brown : Calculation of a constant q spectral transform. *Journal Acoust. Soc. Am.*, 89(1):425–434, 1991.

A. Camarena-Ibarrola et E. Chávez : Real time tracking of musical performances. *In* G. Sidorov, A. Hernández Aguirre et C. Reyes García, éds : *Advances in Soft Computing*, vol. 6438 de *Lecture Notes in Comput. Science*, p. 138–148. Springer Berlin / Heidelberg, 2010.

P. Cano, E. Batlle, T. Kalker et J. Haitsma : A review of audio fingerprinting. *Journal of VLSI Signal Processing*, 41:271–284, 2005. ISSN 0922-5773. 10.1007/s11265-005-4151-3.

P. Cano, A. Loscos et J. Bonada : Score-performance matching using hmms. *In Actes de Inter. Computer Music Conf.*, p. 441–444, 1999.

A. T. Cemgil, H. J. Kappen, P. Desain et H. Honing : On tempo tracking : Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 28 :4:259–273, 2001.

A. Cont : Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 245–248, 2006.

A. Cont : Antescofo : Anticipatory synchronization and control of interactive parameters in computer music. *In Actes de Inter. Computer Music Conf.*, août 2008a.

A. Cont : *Modeling Musical Anticipation : From the time of music to the music of time*. Thèse de doctorat, Université Paris VI – University of California San Diego, 2008b.

A. Cont : A coupled Duration-Focused architecture for Real-Time Music-to-Score alignment. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(6):974–987, juin 2010. ISSN 0162-8828.

A. Cont, D. Schwarz et N. Schnell : Training ircam's score follower. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, vol. 3, p. 253–256, 2005.

A. Cont, D. Schwarz, N. Schnell et C. Raphael : Evaluation of real-time audio-to-score alignment. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 315–316, 2007.

A. E. Cook et M. J. Russell : Improved duration modeling in hidden markov models using series-parallel configurations of states. *In Actes de Inst. Acoust. Conf.*, vol. 8, p. 299–306, 1986.

A. Daniel, V. Emiya et B. David : Perceptually-based evaluation of the errors usually mad when automatically transcribing music. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 550–555, 2008.

R. B. Dannenberg : An on-line algorithm for real-time accompaniment. *In Actes de Inter. Computer Music Conf.*, p. 193–198, 1984.

R. B. Dannenberg et N. Hu : Polyphonic audio matching for score following and intelligent audio editors. *In Actes de Inter. Computer Music Conf.*, p. 27–33, 2003.

R. B. Dannenberg et H. Mukaino : New techniques for enhanced quality of computer accompaniment. *In Actes de Inter. Computer Music Conf.*, p. 279–289, 1988.

P. Desain, H. Honing et H. Heijink : Robust score-performance matching : Taking advantage of structural information. *In Actes de Inter. Computer Music Conf.*, p. 337–340, 1997.

J. Devaney et D. P. Ellis : Handling asynchrony in audio-score alignment. *In Actes de Inter. Computer Music Conf.*, Montreal, Canado, août 2009.

J. Devaney, M. I. Mandel et D. P. W. Ellis : Improving MIDI-audio alignment with acoustic features. *In Actes de IEEE Workshop Applicat. of Signal Proccessing to Audio and Acoust*, p. 45–48, oct. 2009.

S. Dixon : Live tracking of musical performances using on-line time warping. *In Actes de DAFX Conf.*, Madrid, Spain, sept. 2005.

S. Dixon et G. Widmer : Match : a music alignment tool chest. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 492–497, 2005.

T.-M.-T. Do et T. Artières : Neural conditional random fields. *In Workshop on Deep Learning for Speech Recognition and Related Applications, NIPS*, 2009.

Z. Duan et B. Pardo : A state space model for online polyphonic audio-score alignment. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 197–200, mai 2011.

V. Emiya, R. Badeau et B. David : Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. Audio, Speech, Language Processing*, 18(6):1643–1654, août 2010. ISSN 1558-7916.

S. Ewert, M. Müller et P. Grosche : High resolution audio synchronization using chroma onset features. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 1869–1872, 2009.

S. Ewert, M. Müller et R. Dannenberg : Towards reliable partial music alignments using multiple synchronization strategies. *In* M. Detyniecki, A. García-Serrano et A. Nürnberger, éds : *Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User*, vol. 6535 de *Lecture Notes in Comput. Science*, p. 35–48. Springer Berlin / Heidelberg, 2011.

J. D. Ferguson : Variable duration models for speech. *In Actes de Symp. Applicat. Hidden Markov Models to Text and Speech*, p. 143–179, 1980.

S. Fine et Y. Singer : The hierarchical hidden Markov model : Analysis and applications. *In Actes de Machine Learning Conf.*, p. 41–62, 1998.

N. H. Fletcher et T. D. Rossing : *The Physics of Musical Instruments*. Springer-Verlag New York Inc., 1998.

C. Fox et J. Quinn : How to be lost : principled priming and pruning with particles in score following. *In Actes de Inter. Computer Music Conf.*, p. 81–84, Copenhagen, Denmark, août 2007.

C. Fremerey, M. Clausen, S. Ewert et M. Müller : Sheet music-to-audio identification. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 645–650, 2009.

C. Fremerey, M. Müller et M. Clausen : Handling repeats and jumps in score-performance synchronization. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 243–248, 2010.

A. Friberg et J. Sundberg : Time discrimination in a monotonic, isochronous sequence. *Journal Acoust. Soc. Am.*, 98:2525–2531, 1995.

M. Goto, H. Hashiguchi, T. Nishimura et R. Oka : RWC music database : Popular, classical, and jazz music databases. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 287–288, 2002.

P. Grosche, M. Müller et F. Kurth : Cyclic tempogram – a mid-level tempo representation for music signals. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 5522–5525, mars 2010.

L. Grubb et R. Dannenberg : A stochastic method of tracking a vocal performer. *In Actes de Inter. Computer Music Conf.*, p. 301–308, 1997.

L. Grubb et R. B. Dannenberg : Automatic ensemble performance. *In Actes de Inter. Computer Music Conf.*, p. 63–69, 1994.

L. Grubb et R. B. Dannenberg : Enhanced vocal performance tracking using multiple information sources. *In Actes de Inter. Computer Music Conf.*, p. 37–44, 1998.

L. V. Grubb : *A Probabilistic Method for Tracking a Vocalist.* Thèse de doctorat, Carnegie Mellon University, 1998.

H. Heijink, P. Desain, H. Honing et L. Windsor : Make me a match : An evaluation of different approaches to score-performance matching. *Computer Music Journal*, 24: 43–56, 2000.

R. Hennequin, B. David et R. Badeau : Score informed audio source separation using a parametric model of non-negative spectrogram. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, 2011.

T. Hoshishiba, S. Horiguchi et I. Fujinaga : Study of expression and individuality in music performance using normative data derived from midi recordings of piano music. *In Actes de Intern. Conf. Music Perception and Cognition*, p. 465–470, 1996.

N. Hu, R. B. Dannenberg et G. Tzanetakis : Polyphonic audio matching and alignment for music retrieval. *In Actes de IEEE Workshop Applicat. of Signal Proccessing to Audio and Acoust*, p. 185–188, 2003.

O. İzmirli et R. Dannenberg : Understanding features and distance functions for music sequence alignment. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 411–416, 2010.

O. İzmirli, R. Seward et N. Zahler : Compositional imperatives for implementing an audio alignment program in max/msp. *In Actes de Inter. Computer Music Conf.*, p. 266–269, 2002.

O. İzmirli et N. Zahler : Signatures and pattern anchoring for score following. *Journal of New Music Research*, 34(4):395–408, déc. 2005.

M. T. Johnson : Capacity and complexity of HMM duration modeling techniques. *IEEE Signal Processing Lett.*, 12(5):407–410, 2005.

A. Jordanous et A. Smaill : Investigating the role of score following in automatic musical accompaniment. *Journal of New Music Research*, 38:197–209, 2009.

H. Kaprykowsky et X. Rodet : Globally optimal short-time dynamic time warping : Application to score to audio alignment. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, vol. 5 de *249–252*, 2006.

J. Keshet, S. Shalev-Shwartz, Y. Singer et D. Chazan : A large margin algorithm for speech-to-phoneme and music-to-score alignment. *IEEE Trans. Audio, Speech, Language Processing*, 15(8):2373–2382, nov. 2007.

D. Koller et N. Friedman : *Probabilistic Graphical Models : Principles and Techniques.* MIT Press, 2009.

J. Lafferty, A. McCallum et F. Pereira : Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *In Actes de European Conf. Artificial Intelligence*, p. 282–289, 2001.

E. W. Large : Dynamic programming for the analysis of serial behaviors. *Behavior Research Methods, Instruments and Computers*, 25(2):238–241, 1993.

D. D. Lee et H. S. Seung : Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

R. Macrae et S. Dixon : Accurate real-time windowed time warping. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 423–428, 2010.

A. Maezawa, H. G. Okuno, T. Ogata et M. Goto : Polyphonic audio-to-score alignment based on bayesian latent harmonic allocation hidden markov model. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 185–188, mai 2011.

A. McCallum, D. Freitag et F. Pereira : Maximum entropy markov models for information extraction and segmentation. *In Actes de European Conf. Artificial Intelligence*, p. 591–598. Morgan Kaufmann, San Francisco, CA, 2000.

Y. Meron et K. Hirose : Automatic alignment of a musical score to performed music. *Acoustical Science and Technology*, 22(3):189–198, 2001.

N. Montecchio et A. Cont : A unified approach to real time autio-to-score and audio-to-audio alignment using sequential monte-carlo inference techniques. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 193–196, mai 2011.

N. Montecchio et N. Orio : Automatic alignment of music performances with scores aimed at educational applications. *In Actes de Inter. Conf. on Automated solutions for Cross Media Content and Multi-channel Distribution*, p. 17–24, 2008.

N. Montecchio et N. Orio : A discrete filterbank approach to audio to score matching for score following. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 495–500, 2009.

M. Müller : *Information Retrieval for Music and Motion*. Springer Verlag, 2007. ISBN 3540740473.

M. Müller et D. Appelt : Path-constrained partial music synchronization. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 65–68, 2008.

M. Müller, M. Clausen, V. Konz, S. Ewert et C. Fremerey : A multimodal way of experiencing and exploring music. *Interdisciplinary Science Reviews*, 35(2):138–153, 2010.

M. Müller et S. Ewert : Joint structure analysis with applications to music annotation and synchronization. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 389–394, 2008.

M. Müller, F. Kurth et M. Clausen : Audio matching via chroma-based statistical features. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 288–295, 2005a.

M. Müller, F. Kurth et M. Clausen : Chroma-based statistical audio features for audio matching. *In Actes de IEEE Workshop Applicat. of Signal Proccessing to Audio and Acoust*, p. 275–278, 2005b.

M. Müller, F. Kurth et T. Röder : Towards an efficient algorithm for automatic score-to-audio synchronization. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 365–372, 2004.

M. Müller, H. Mattes et F. Kurth : An efficient multiscale approach to audio synchronization. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 192–197, 2006.

K. P. Murphy : *Dynamic Bayesian Networks : Representation, Inference and Learning*. Computer science division, UC Berkeley, juil. 2002.

B. Niedermayer : Improving accuracy of polyphonic music to score alignment. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 585–590, 2009a.

B. Niedermayer : Towards audio to score alignment in the symbolic domain. *In Actes de Sound and Music Conf.*, p. 77–82, 2009b.

B. Niedermayer et G. Widmer : A multi-pass algorithm for accurate audio-to-score alignment. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 417–422, 2010a.

B. Niedermayer et G. Widmer : Strategies towards the automatic annotation of classical piano music. *In Actes de Sound and Music Conf.*, 2010b.

N. Orio : Alignment of performances with scores aimed at content-based music access and retrieval. *In* M. Agosti et C. Thanos, éds : *Research and Advanced Technology for Digital Libraries*, vol. 2458 de *Lecture Notes in Comput. Science*, p. 173–184. Springer Berlin / Heidelberg, 2002.

N. Orio : A system for the automatic identification of musicworks. *In Actes de IEEE Inter. Conf. on Image Analysis and Processing - Workshops*, p. 15–20, 2007.

N. Orio et F. o. Déchelle : Score following using spectral analysis and hidden markov models. *In Actes de Inter. Computer Music Conf.*, p. 129–129, 2001.

N. Orio, S. Lemouton et D. Schwarz : Score following : State of the art and new developments. *In Actes de New Interfaces for Musical Expression Conf.*, p. 36–41, 2003.

N. Orio et D. Schwarz : Alignment of monophonic and polyphonic music to a score. *In Actes de Inter. Computer Music Conf.*, p. 129–132, 2001.

S. Ortmanns, H. Ney et A. Eiden : Language-model look-ahead for large vocabulary speech recognition. *In Actes de Intern. Conf. Spoken Language*, vol. 4, p. 2095 –2098 vol.4, oct 1996.

T. Otsuka, K. Murata, K. Nakadai, T. Takahashi, K. Komatani, T. Ogata et H. G. Okuno : Incremental polyphonic audio to score alignment using beat tracking for singer robots. *In Actes de Intern. Conf. Intelligent Robots and Systems*, p. 2289–2296, 2009.

T. Otsuka, K. Nakadai, T. Takahashi, T. Ogata et H. G. Okuno : Real-time audio-to-score alignment using particle filter for coplayer music robots. *EURASIP J. Adv. Signal Process*, 2011:2 :1–2 :13, jan. 2011. ISSN 1110-8657.

L. Oudre : *Reconnaissance d'accords à partir de signaux audio par l'utilisation de gabarits théoriques / Template-based chord recognition from audio signals.* Thèse de doctorat, TELECOM ParisTech, nov. 2010.

B. Pardo et W. Birmingham : Following a musical performance from a partially specified score. *In Actes de IEEE Multimedia Technology Applications Conf.*, p. 202–207, 2001.

B. Pardo et W. Birmingham : Improved score following for acoustic performances. *In Actes de Inter. Computer Music Conf.*, p. 262–265, 2002.

B. Pardo et W. Birmingham : Modeling form for on-line following of musical performances. *In Actes de National Conf. on Artificial Intelligence*, p. 1018–1023, 2005.

J. Paulus et A. Klapuri : Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(6):1159–1170, août 2009. ISSN 1558-7916.

P. Peeling, A. T. Cemgil et S. Godsill : A probabilistic framework for matching music representations. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 267–272, Vienna, Austria, 2007.

G. Peeters : Deriving musical structures from signal analysis for music audio summary generation : "sequence" and "state" approach. *In* U. K. Wiil, éd. : *Computer Music Modeling and Retrieval*, vol. 2771 de *Lecture Notes in Comput. Science*, p. 169–185. Springer Berlin / Heidelberg, 2004.

J. Peng, L. Bo et J. Xu : Conditional neural fields. *In Actes de NIPS Conf.*, p. 1419–1427, 2009.

M. Puckette : Score following using the sung voice. *In Actes de Inter. Computer Music Conf.*, p. 175–178, 1995.

M. Puckette et C. Lippe : Score following in practice. *In Actes de Inter. Computer Music Conf.*, p. 182–185, 1992.

L. R. Rabiner : A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, fév. 1989.

P. Ramesh et J. G. Wilpon : Modeling state durations in hidden markov models for automatic speech recognition. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, vol. 1, p. 381–384, mars 1992.

C. Raphael : Automatic segmentation of acoustic musical signals using hidden Markov models. *IEEE Trans. Pattern Anal. Machine Intell.*, 21:360–370, 1999.

C. Raphael : Music plus one : A system for expressive and flexible musical accompaniment. *In Actes de Inter. Computer Music Conf.*, 2001.

C. Raphael : A hybrid graphical model for aligning polyphonic audio with musical scores. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 387–394, 2004.

C. Raphael : Aligning music audio with symbolic scores using a hybrid graphical model. *Machine Learning Journal*, 65:389–409, 2006.

C. Raphael et Y. Gu : Orchestral accompaniment for a reproducing piano. *In Actes de Inter. Computer Music Conf.*, 2009.

X. Rodet, J. Escribe et S. Durigon : Improving score to audio alignment : Percussion alignment and precise onset estimation. *In Actes de Inter. Computer Music Conf.*, 2004.

M. Russell et A. Cook : Experimental evaluation of duration modelling techniques for automatic speech recognition. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, vol. 12, p. 2376 – 2379, avr. 1987.

H. Sakoe et S. Chiba : Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Audio, Speech, Language Processing*, 26(1):43 – 49, fév. 1978. ISSN 0096-3518.

S. Salvador et P. Chan : Fastdtw : Toward accurate dynamic time warping in linear time and space. *In Actes de KDD Workshop on Mining Temporal and Sequential Data*, p. 70–80, 2004.

D. Schwarz, N. Orio et N. Schnell : Robust polyphonic midi score following with hidden markov models. *In Actes de Inter. Computer Music Conf.*, p. 442–445, 2004.

S. Shalev-Shwartz, J. Keshet et Y. Singer : Learning to align polyphonic music. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 381–386, 2004.

F. Soulez, X. Rodet et D. Schwarz : Improving polyphonic and poly-instrumental music to score alignment. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, p. 143–148, 2003.

C. Sutton et A. McCallum : An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 2011. To appear.

C. Sutton, K. Rohanimanesh et A. McCallum : Dynamic conditional random fields : factorized probabilistic models for labeling and segmenting sequence data. *In Actes de European Conf. Artificial Intelligence*, p. 99–106. ACM, 2004. ISBN 1-58113-838-5.

B. Taskar, C. Guestrin et D. Koller : Max-margin Markov networks. *In Actes de NIPS Conf.*, 2003.

M. E. Tekin, C. Anagnostopoulou et Y. Tomita : Towards an intelligent score following system : Handling of mistakes and jumps encountered during piano practicing. *In* U. K. Wiil, éd. : *Computer Music Modeling and Retrieval*, vol. 3310 de *Lecture Notes in Comput. Science*, p. 211–219. Springer Berlin / Heidelberg, 2005.

R. J. Turetsky et D. P. Ellis : Ground-truth transcriptions of real music from force-aligned midi syntheses. *In Actes de Inter. Soc. for Music Information Retrieval Conf.*, 2003.

J. D. Vantomme : Score following by temporal pattern. *Computer Music Journal*, 19(3): 50–59, 1995.

B. Vercoe : The synthetic performer in the context of live performance. *In Actes de Inter. Computer Music Conf.*, p. 199–200, 1984.

B. Vercoe et M. Puckette : Synthetic rehearsal : Training the synthetic performer. *In Actes de Inter. Computer Music Conf.*, p. 275–278, 1985.

T. Virtanen, A. Cemgil et S. Godsill : Bayesian extensions to non-negative matrix factorisation for audio signal modelling. *In Actes de IEEE Inter. Conf. Acoust. Speech, Signal Processing*, p. 1825 –1828, avr. 2008.

A. J. Viterbi : Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

H. M. Wallach : Conditional random fields : An introduction. Rap. tech. MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, 2004.

S.-Z. Yu : Hidden semi-Markov models. *Artificial Intelligence*, 174(2):215–243, 2010. ISSN 0004-3702. Special Review Issue.

Y. Zhu et M. Kankanhalli : Precise pitch profile feature extraction from musical audio for key detection. *IEEE Trans. Multimedia*, 8(3):575–584, juin 2006. ISSN 1520-9210.

Index