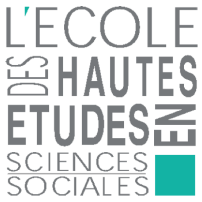
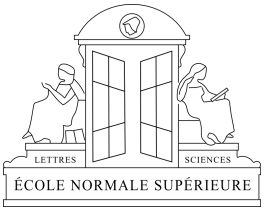
Master 2 ATIAM
Mémoire de recherche

Modèles harmoniques

Auteur :
Floriane Dardard

Encadrant :
Marc Chemillier





1er mars 2012 - 31 juillet 2012

Résumé

Ce travail s'inscrit dans la lignée des systèmes d'improvisation musicale avec imitation stylistique, se plaçant plus particulèrement dans le contexte ImproteK, un dérivé d'Omax 2.0 développé à l'IRCAM, dans l'équipe RepMus. Il s'agit ici d'ajouter une fonction de substitution ou enrichissement harmonique, qui stimulerait la créativité d'un musicien dans un contexte d'interaction homme-machine. L'objectif est de construire automatiquement cette fonction selon les modulations harmoniques inhérentes au style du musicien. Une phase de reconnaissance d'accords est nécessaire pour repérer les substitutions et les apprendre automatiquement, précédant la phase d'application des substitutions, l'ensemble réalisé grâce à un Modèle de Markov Caché.

Table des matières

Remerciements . . . 3
Introduction . . . 4

1 État de l'art **5**
1.1 Reconnaissance d'accords . . . 5
1.2 Grammaires harmoniques . . . 6
1.3 OMAX et IMPROTEK . . . 7

2 Reconnaissance d'accords **8**
2.1 Modèles de Markov Cachés . . . 8
2.1.1 Définition . . . 8
2.1.2 Application à la situation . . . 9
2.2 Phase d'apprentissage . . . 10
2.2.1 Représentation des données . . . 10
2.2.2 Construction du modèle . . . 11
2.3 Module de reconnaissance . . . 11
2.3.1 Identification d'observations . . . 11
2.3.2 Algorithme de Viterbi . . . 13
2.3.3 Avantages et limitations du modèle de reconnaissance . . . 14

3 Substitutions Harmoniques **15**
3.1 Détection par le Modèle . . . 15
3.2 Première idée : Application dans une séquence . . . 15
3.2.1 Réutilisation de Viterbi . . . 16
3.3 Formalisation et généralisation des substitutions . . . 17
3.4 Et le contexte, alors ? . . . 17
3.5 Deuxième idée : Modèle enrichi et généralisé . . . 18
3.5.1 Appliquer en profondeur les substitutions . . . 18
3.5.2 Cas particuliers . . . 19
3.5.3 Parcours du modèle enrichi . . . 19
3.6 Problèmes à résoudre . . . 20

4 Expériences **21**
4.1 Bernard Lubat 21
4.2 Blues et Steedman 22
4.2.1 Transcription des règles 22
4.2.2 Extraction des règles 22
4.3 Génération audio 23

5 Conclusion et avancées **24**
Références 25

Remerciements

Je tiens à remercier d'abord mon maître de stage, Marc Chemillier, qui a pris la peine de concocter un sujet personnalisé, avec lequel j'ai pu découvrir un style de musique, le jazz, d'une manière incroyable. Beaucoup de voyages, en séminaire ou en voiture, très intructifs. Merci pour cette si grande patience, ce support volontaire et cette participation exemplaire à n'importe quel moment.
Merci à Jérôme Nika aussi, avec qui nous avons formé une vraie équipe de recherche, même sans bureau attribué. Merci pour les coups de main qui sauvent, et pour l'ébullition des idées en milieu rural.
Merci à toute l'équipe d'Uzeste avec qui nous avons partagé plus que de simples sessions professionnelles, pour leur accueil et leur hospitalité inégalées, en particulier à Fabrice Viera et Bernard Lubat.
Merci à Sylvie de m'avoir accueillie dans son bureau animé, toujours débordante d'énergie communicative.
Merci à tous les gens de l'IRCAM avec qui j'ai eu l'occasion de parler de mon stage, et surtout à l'équipe RepMus, Laurent Bonasse, Gérard Assayag et Benjamin Lévy.
Merci à la promo ATIAM pour l'ambiance des pauses cafés rieuses et culturelles.
Merci à Sarah, Pablo et Laure pour leur soutien, accueil, générosité sans bornes et relecture de ce rapport.

Introduction

Comment se fait-il que tous les musiciens de jazz jouent les mêmes classiques, et qu'ils soient si différents à chaque interprétation ? Ce n'est pas seulement une question d'arrangement, ou d'ensemble d'instruments différents, mais de *caractère différent* : En effet, même si les musiciens jouent sur la même grille d'accords, l'esprit du jazz est de favoriser les enrichissements, les substitutions, prônant une liberté harmonique incroyable ! Chaque musicien part dans la direction harmonique qui l'inspire. On pourrait citer les nombreuses versions de "Round Midnight", composé en 1944 par Thelonious Monk, et interprété, voire ré-écrit harmoniquement, à de nombreuses reprises : par Miles Davis, Bill Evans, Claude Nougaro, ou Hermeto Pascoal, pour ne citer qu'eux.

Chaque version a un style harmonique propre au musicien qui l'a interprétée, reconnaissable par les auditeurs habitués, qui discernent, souvent inconsciemment, les particularités des richesses de l'accompagnement. Peut-on rendre ce processus conscient, par l'extraction automatique de règles harmoniques ? Serait-il possible de comprendre intrinsèquement les spécificités harmoniques des musiciens ?

Le travail ci-dessous est le fruit d'un stage encadré par Marc Chemillier, dans le cadre de la formation de Master Acoustique, Traitement du signal et Informatique Aplliqués à la Musique (ATIAM) dispensée par l'IRCAM et validée par l'UPMC.
L'objectif était de développer une nouvelle fonctionalité du logiciel ImproteK, un logiciel d'improvisation avec imitation stylistique par oracle dans la lignée d'Omax. Il s'agit de représenter formellement les enchaînements harmoniques caractéristiques d'un musicien, et plus particulièrement les substitutions effectuées par rapport à la grille, dans le but de générer des phrases harmoniques enrichies par rapport à celles produites d'après l'oracle.
En effet, l'oracle produit déjà des substitutions naturellement, cependant l'objectif ici est de formaliser ces substitutions apprises afin de les généraliser et produire des enchaînements originaux qui se rapprochent du style du musicien.

Dans l'optique de préserver autant que possible le caractère automatisé d'Omax, la théorie musicale des substitutions ne devait pas être codée en dur, mais *apprise* par le programme à partir de la base de données disponible. Néanmoins, il a été nécessaire d'apprendre à ImproteK ce que signifiait le chiffrage des accords, pour pouvoir détecter les substitutions et les enrichissements. Le reste est laissé à l'intelligence artificielle.

Un préambule nécessaire à la détection de substitution était la reconnaissance des accords joués par le musicien, que l'on appellera *voicings*, pour pouvoir comparer avec l'accord théorique de la grille et constater s'il y a eu substitution ou non. Cette phase est accomplie par un Modèle de Markov Caché, avec apprentissage supervisé, c'est-à-dire que l'algorithme se base sur un fichier midi chiffré à la main au préalable, pour pouvoir reconnaître les accords d'autres fichiers.

Cette étape accomplie, il apparut naturel de *réutiliser ce modèle* pour appliquer des substitutions à une phrase, avec la notion de chemin harmonique alternatif dans le modèle. Le modèle a été aussi utilisé pour extraire *a posteriori* des règles harmoniques, alimentant, on l'espère, la théorie musicale d'une façon totalement expérimentale.

Chapitre 1

État de l'art

1.1 Reconnaissance d'accords

L'analyse harmonique automatique a fait l'objet de nombreux travaux, adoptant des démarches très différentes. Notons que le problème comporte une difficulté non négligeable, qui est celle de la segmentation du flux de notes, que nous ne traiterons pas à sa juste valeur ici, car l'architecture de beats d'ImproteK permet de s'en débarasser.

Règles explicites d'harmonie Des tentatives d'expliciter les règles qui permettent de chiffrer un ensemble de notes ont été effectuées. Souvent, elles passent par une étape de détermination de la fondamentale de l'accord, puis de sa nature. Ces modèles, pour être efficaces, doivent inclure un filtrage dans les notes, pour déterminer celles faisant partie de l'harmonie ou non. Ainsi, dans le système de Maxwell de [M92], une cinquantaine de règles modélisant la consonance et la dissonance permettent d'ignorer les notes extérieures à l'accord, ce qui restreint le modèle à l'harmonie classique. Temperley, dans [T97] introduit des règles de préférence en se basant sur la ligne des quintes, et en construisant un enchaînement dans cette ligne. L'idée a été poussée jusqu'à la correspondance entre séquences harmoniques et chemin de poids minimal dans un graphe dans le système HarmAn dans [BP02] et repris dans [IIR07]. Pour effectuer ce parcours, il est nécessaire de partitionner d'abord le flux de notes en entités cohérentes, puis de leur assigner un ensemble de labels possibles, avec un poids entre les labels pour mettre en valeur les cadences.

Utilisateur lui aussi de règles, mais cette fois-ci d'un genre différent, Ulrich s'est penché sur le jazz dans [U77] et a développé un système d'analyse d'accords en transcrivant d'abord les règles de renversement, d'omissions et de substitutions harmoniques sous forme d'une grammaire. Appliqué à la séquence de notes, cela est transformé en arbre, que son algorithme explore par backtracking.

Aggrégation spontanée de notes En utilisant la spirale des triades de Chew qui permet d'évaluer la tonalité d'une pièce dans [C00], un système de reconnaissance d'accords a été imaginé dans [CC07], en utilisant les distances minimales entre accords selon la théorie néo-Riemanienne et en établissant ainsi une base d'accords possibles. On attribue alors aux clusters de notes sélectionnés une fonction harmonique, probabilistiquement et par calculs de barycentres. Un système utilisant les fusions de clusters plutôt que des considérations algébriques a été développé dans [B01].

Cognition Chronologiquement, une des premières approches était basée sur la signification cognitive de consonance harmonique. Krumhansl et Schmuckler, à la suite d'expériences psychoacoustiques ont défini deux *profils de référence* (l'un majeur et l'autre mineur) en faisant écouter à des musiciens une gamme majeure ou mineure, suivie d'une note quelconque ([KK82], [K90]. Une fois la base de

profils constituée, on détermine la tonalité de la pièce, comme dans [T01] en comparant ces profils avec celui de la pièce. Il est aussi possible d'estimer les accords ([HRR09]) en considérant des profils sur une fenêtre d'analyse ponctuelle.

Modèles de Markov Cachés Cette approche basée sur l'apprentissage laisse à l'ordinateur le soin de construire les règles harmoniques qu'il jugera bon d'adopter. Le principe repose sur une hypothèse markovienne selon laquelle les enchaînements harmoniques dépendent du contexte (ce qui n'était en général pas pris en compte par les modèles précédents), mais les transitions entre accords sont de mémoire finie. En général, on prend des modèles d'ordre 1, les ordres supérieurs allongeant considérablement le temps de calcul pour peu d'améliorations au niveau des résultats. On peut classer les systèmes selon le type de données analysées : Midi ou Audio.

Pour les modèles traitant les données audio, les meilleurs résultats ont été obtenus à l'aide des *Pitch Class Profiles*, ou chromagrammes, qui représentent efficacement un ensemble de notes (modulo 12) par son énergie spectrale. Les systèmes diffèrent alors selon l'algorithme d'estimation des paramètres et l'initialisation du modèle. Tandis que certains l'initialisent au hasard ([ES03]), d'autres choisissent de biaiser avec des pré-requis sur la structure des accords et favoriser les cadences : ([BP05], [LS06])

Le modèle traitant les données Midi qui a servi d'inspiration est celui de Raphael et Stoddard [RS03], qui effectue des simplifications de l'ordre de l'harmonie classique, que nous ne reprendrons pas.

1.2 Grammaires harmoniques

Si les règles de l'harmonie classique sont l'objet de réglementations strictes et d'interdits clairs, l'harmonie du style "jazz" demeure l'objet de multiples études, pour tenter de comprendre a posteriori pourquoi certains enchaînements sont naturels.
Les accords de jazz peuvent faire l'objet d'enrichissements ou de substitutions. L'enrichissement d'un accord est l'ajout de notes à la structure classique de l'accord parfait, souvent des tierces, cela dans la limite des moyens instrumentaux à disposition. Qui écoute Hermeto Pascoal se doute que ses accords sont "originaux" dans le sens harmonique du terme, surprenants à la première écoute, utilisant des notes peu communes dans les chiffrages de son *Calendario do Som*. (Pour les curieux, se référer à [P96] et [CLN99]).
Les substitutions, en revanche, changent l'ensemble de l'accord en un accord avec une basse différente. Une grammaire générative des blues à 12 mesures a été introduite par Steedman en 1984 ([S84]). Il s'agit d'une grammaire contextuelle qui à partir d'une grille cadentielle très simple, permet d'obtenir un grand nombre de grilles de blues. L'intérêt harmonique de ces substitutions est discuté dans [JL91], qui soutient que les substitutions sont la source d'une richesse harmonique infinie car elles complexifient les enchaînements harmoniques que l'on pourrait juger trop communs.
Néanmoins ce modèle présente des inconvénients : Premièrement, il n'est pas implémentable dans sa totalité. Cela provient essentiellement du caractère contextuel des règles du modèle, qui rendent impossible son parsage automatique. D'autre part, ces règles comportent des règles de subdivision qui peuvent théoriquement être appliquées à l'infini, et donc une granularité harmonique très petite, ce qui n'est pas le cas dans la réalité où l'on observe rarement des transitions harmoniques au 32^e de croche près. Enfin, ces règles n'engendrent pas toutes les grilles de blues, comme montré par Pachet dans [P98] : en effet, les règles grammaticales permettant de construire pour chaque séquence reconnue un arbre de dérivation correspondant à l'appliquation successive de règles de substitutions, il n'est néanmoins pas possible de trouver cet arbre de dérivation pour *Solar*, de Miles Davis.

Nous allons essayer de répondre à ces limitations dans ce travail, en implémentant un modèle représentant les substitutions et enrichissements propres à un musicien, puis en réinjectant ces substitutions

dans une phrase harmonique générée par un logiciel d'improvisation. Ce problème avait déjà intéressé Marc Chemillier en 2004 [C04], et une première implémentation avait été tentée, en se basant sur la grammaire de Steedman. Néanmoins, il s'agit d'adopter une position moins théoricienne, en n'utilisant pas de pré-requis de grammaires harmoniques, mais en inférant des règles de substitution.

1.3 OMAX et IMPROTEK

OMAX et IMPROTEK sont deux logiciels d'improvisation avec interaction homme-machine et imitation stylistique par le parcours d'un Oracle des Facteurs ([ACR99]) inventé par G. Assayag et S. Dubnov en 2004 ([AD04]).

ImproteK est un dérivé d'Omax 2.0, codé en LISP et en Max/MSP, tandis que l'Omax de l'équipe RepMus a été complètement recodé en Max/MSP. Ces deux procédés poursuivent des objectifs différents dûs aux modes de jeu de prédilection de chacun : pour Omax, c'est plutôt le *free jazz*, tandis qu'ImproteK évolue dans un contexte de grille d'accords. Cependant, depuis quelques temps, les deux équipes semblent s'intéresser aux deux sujets communs que sont le « beat-tracking » et l'harmonisation automatique. A terme, il serait intéressant de « refaire la jonction » entre ces deux cousins germains.

Ce travail s'est appuyé plutôt sur le système ImproteK et plus particulièrement sur les améliorations de J. Nika ([N12]) qui a développé un système pour intégrer un cadre rythmique et une structure harmonique sous-jacente lors d'une session d'improvisation. La division du flux musical en pulsations, ou *beats* labelisés par les accords de la grille, est primordiale dans cet algorithme, et permet de simplifier le problème de la fragmentation des données ainsi que celui de la synchronisation lors de l'analyse et du mixage. Le beat est une unité de temps centrale de ce travail.

Chapitre 2

Reconnaissance d'accords

2.1 Modèles de Markov Cachés

2.1.1 Définition

Les Modèles de Markov Cachés (en anglais *Hidden Markov Models*, abbrégé en HMM) sont un outil statistique de l'intelligence artificielle, permettant de modéliser l'évolution d'un processus dont l'on ignore une caractéristique déterminante : c'est un automate markovien à états cachés. Inventés dans les années 60 par Leonard E. Baum ([BP66]), ils sont connus notamment pour leur usage en reconnaissance vocale, ([R89]) puis en analyse de séquences biologiques, en particulier l'ADN.

Les HMM tentent de modéliser des systèmes dont on ne connaît que des observations, et essaient d'en déduire un fonctionnement interne, notamment en fournissant des algorithmes efficaces pour résoudre trois types de problèmes :

1. inférer la suite d'états cachés la plus probable qui auraient conduit à une suite d'observations donnée,
2. calculer la probabilité d'occurence d'une suite d'observations donnée,
3. réestimer ses propres paramètres d'après une suite d'observations selon un maximum de vraisemblance.

Définition et notations

Un HMM (Figure 2.1) est décrit par un quintuplet $\lambda = (N, M, A, B, \pi)$ où
– N est le nombre d'états (cachés) du modèle : *On note* $(x_i)_{i\in\{1...N\}}$ *l'ensemble de ces états.*
– M est le nombre d'observations possibles : *On note* $(y_i)_{i\in\{1...M\}}$ *l'ensemble de ces états.*
Si on observe y_i *à l'instant* t*, on note :* $Y(t) = y_i$*. L'état de l'automate correspondant est noté* $X(t)$*.*

– A est la matrice $N \times N$ de probabilité de transition entre les états telle que :
$a_{ij} = P(X(t) = x_j \mid X(t-1) = x_i)$, pour tout t.
– B est la matrice $N \times M$ de probabilité d'observation selon les états telle que :
$b_{ij} = P(Y(t) = y_j \mid X(t) = x_i)$, pour tout t.
– π est le vecteur de distribution initiale tel que :
$\pi_i = P(X(0) = x_i) = P(X(t) = x_i)$, pour tout t..

Notons que le modèle est homogène, et que les probabilités ne dépendent pas du temps.

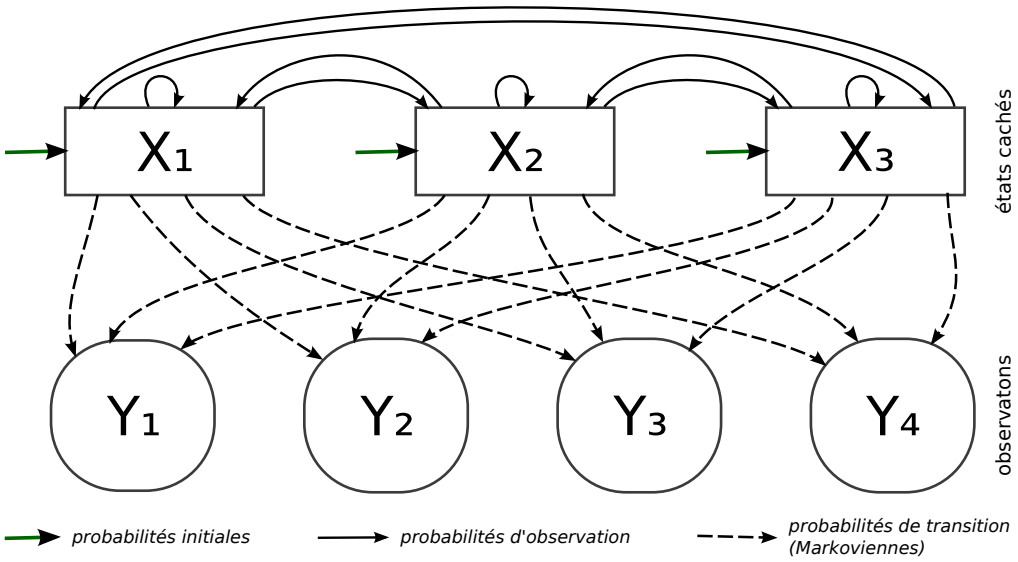


FIGURE 2.1 – Modèle de Markov Caché

2.1.2 Application à la situation

Problème A partir d'un fichier midi, ou d'un midibuff live, divisé en beats labelisés par la grille, il faut déterminer le chiffrage des notes que le musicien a jouées.

Dans notre situation de reconnaissance d'accords, nous avons à disposition les notes (en midi) jouées par le musicien, regroupées par *beats* labelisés par des accords de la grille, qui constitueront nos observations, à travers la classe *HarmonicObs* représentée sur la figure 2.2. Leur nombre dépend de la variété du jeu du musicien. On rétrécit ce nombre en remarquant que seul le pitch, modulo l'octave, est déterminant ici. On fait correspondre à chaque observation un *profil harmonique*, qui est l'ensemble des notes modulo 12 contenues dans le beat, accolé au label de la grille du beat, que l'on appelle *label harmonique théorique*.

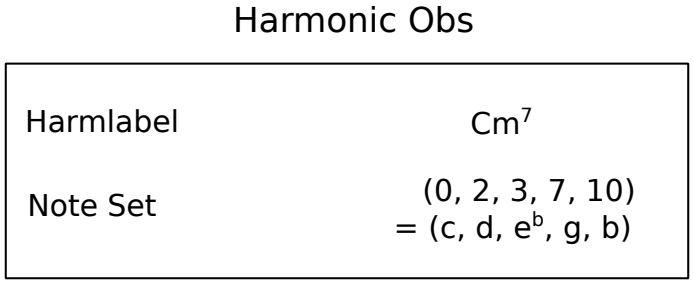


FIGURE 2.2 – Un objet de la classe HarmonicObs

Nous supposons que chacune de ces observations est une instance d'un accord particulier, et chiffrable, selon des règles implicites qu'il appartient au modèle de déterminer, et par conséquent ici les états cachés sont les chiffrages possibles des accords, ou *labels harmoniques réels*. La classe correspondante est *LabelingState* (figure 2.3).

Les transitions entre états sont les transitions harmoniques possibles (par exemple, la probabilité de passer d'un Cm7 à un F7). Les probabilités d'observations traduisent les façons de jouer cet accord par le musicien. Pour un musicien jouant classiquement, on aura :

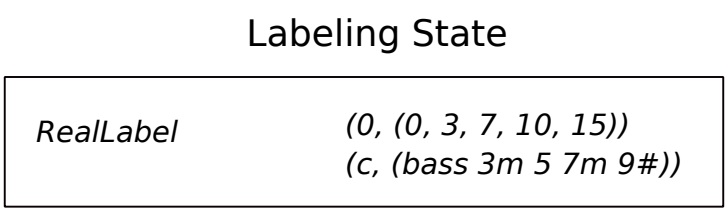


FIGURE 2.3 – La représentation LabelingState d'un chiffrage possible

$P(Y(t) = ((c, e, g), Cmaj) \mid X(t) = Cmaj) \approx 1.$

2.2 Phase d'apprentissage

2.2.1 Représentation des données

Avant d'être opérationnel, le HMM nécessite une phase d'apprentissage. Ici le choix effectué est celui d'un apprentissage supervisé : On chiffre à la main un fichier midi du musicien dont on veut modéliser le style harmonique. Le chiffrage est effectué en empilement de tierces sur la fondamentale de l'accord, avec possible absence de certaines tierces. Comme on est en présence d'accords pouvant aller jusqu'à la 13^e, on représente ce chiffrage modulo 24 : `(bass, contents)` où `bass` est la fondamentale de l'accord, et `contents` est un vecteur de contenu harmonique. On obtient ainsi une suite de pulsations chiffrées contenant des notes, comme en figure 2.4. On appelle chaque pulsation contenant ces informations un *analized beat*, qui comprend un champ *HarmonicObs* et un champ *LabelingState*.

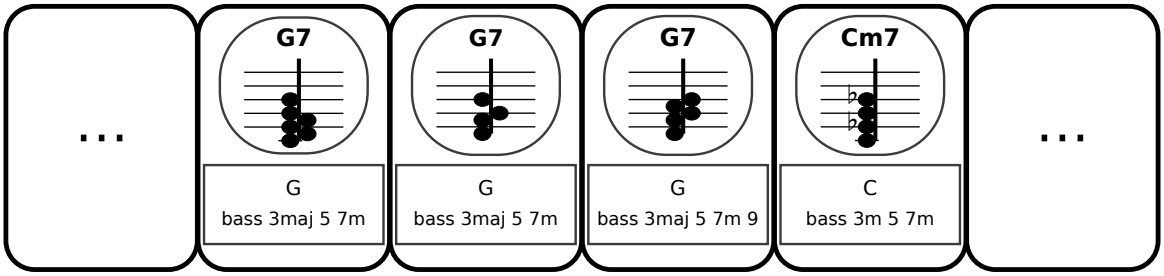


FIGURE 2.4 – Une séquence d'observations chiffrées à la main (fragment)

Le problème principal est ici qu'on a un nombre très important d'observations différentes, même en ne prenant en compte que la hauteur des notes ($2^{11} = 2048$). On veut absolument éviter de manipuler des matrices carrées de cette taille, pour des raisons de rapidité d'exécution nécessaire aux sessions ImproteK. Le choix de représentation des données se porte donc alors sur des *hash tables*, qui permettent de stocker seulement les données pertinentes, et d'y avoir accès efficacement.

On va construire simultanément trois hash tables : `ini-dis` , `transitions` et `observations`. On peut voir cela comme des fonctions :

$$\begin{array}{rccc} \texttt{ini-dis} & : & \mathcal{L} & \rightarrow & [0;1] \\ & & x_i & \mapsto & P(X(0)) \end{array}$$

où $\mathcal{L}$ est l'ensemble des labels (réels possibles).

$$\begin{array}{rccc} \texttt{transitions} : & \mathcal{L}^2 & \to & [0;1] \\ & (x_i, x_j) & \mapsto & P(X(t) = x_j \mid X(t-1) = x_i) \end{array}$$

$$\begin{array}{rccc} \texttt{observations} : & \mathcal{O} & \to & (\mathcal{L} \times [0;1])_{\mathbb{N}} \\ & y_i & \mapsto & (x_n, P(y_i \mid x_n))_n \end{array}$$

où $\mathcal{O} = \mathcal{G} \times \mathcal{N}$ est l'ensemble des observations possibles, qui sont un couple constitué par un accord de la grille et un ensemble de notes jouées (respectivement $\mathcal{G}$ et $\mathcal{N}$ l'ensemble des accords de la grille, et l'ensemble des voicings possibles). La liste renvoyée est la liste des labels réels possibles, avec leurs probabilités respectives.

2.2.2 Construction du modèle

La première approche consiste à faire du dénombrement sur le fichier d'apprentissage pour calculer des probabilités. On obtient ainsi une bonne estimation des paramètres du modèle.

A partir du fragment de séquence en figure 2.4, on obtient le modèle suivant :

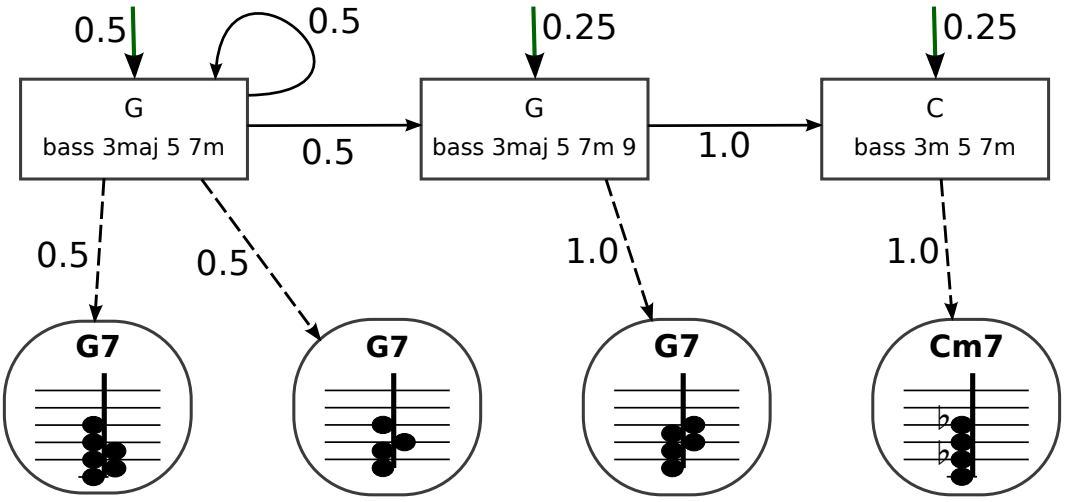


FIGURE 2.5 – Le modèle construit à partir du fragment précédent

2.3 Module de reconnaissance

Principe On veut chiffrer un certain voicing joué, pour cela on cherche d'abord à le rapprocher à une observation connue, puis on calcule le label harmonique le plus probable pour cette observation, en prenant en compte le contexte, grâce à l'algorithme de Viterbi.

Dans cette section, on considère le modèle d'exemple de la figure 2.6.

2.3.1 Identification d'observations

La difficulté ici provient du fait que toutes les observations combinatoirement possibles ne sont pas forcément présentes dans le modèle, donc le voicing joué n'est pas obligatoirement présent dans la base de données d'observations connues. Lorsqu'on est confronté à un voicing nouveau, comment

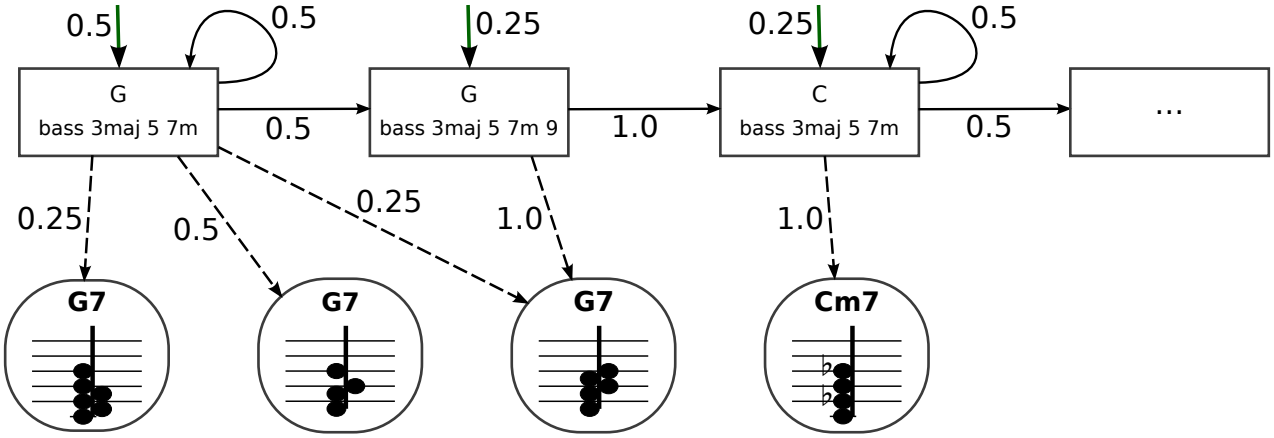


FIGURE 2.6 – Un modèle harmonique possible

déterminer l'observation connue la plus *proche* ?

On dispose des notes jouées par le musicien et de la grille théorique du morceau. On décide de choisir l'observation qui serait la plus proche mélodiquement des notes jouées, avec la même grille théorique, si possible.

On définit alors une *distance harmonique* entre deux ensembles de notes (représentées modulo 12) :

$$\begin{array}{rccc} \texttt{distance} & : & \mathcal{N} \times \mathcal{N} & \rightarrow & \mathbb{N} \\ & & (N_1, N_2) & \mapsto & card(N_1 \cup N_2) - card(N_1 \cap N_2) \end{array}$$

Cette distance permet de différencier un accord et son enrichissement, par exemple Dm^9 et Dm, plus efficacement que si l'on considérait seulement l'intersection des deux ensembles : un Dm joué pourrait être identifié à un $Dm9$ car l'intersection (d, f, a) est la même pour les deux.

En appliquant cet algorithme au modèle d'exemple, on peut faire correspondre au flux mélodique (avec grille) des observations connues par le modèle, comme dans la figure 2.7. Ici, le flux à reconnaître est sur la ligne du dessus, et chaque beat est associé à une observation connue.

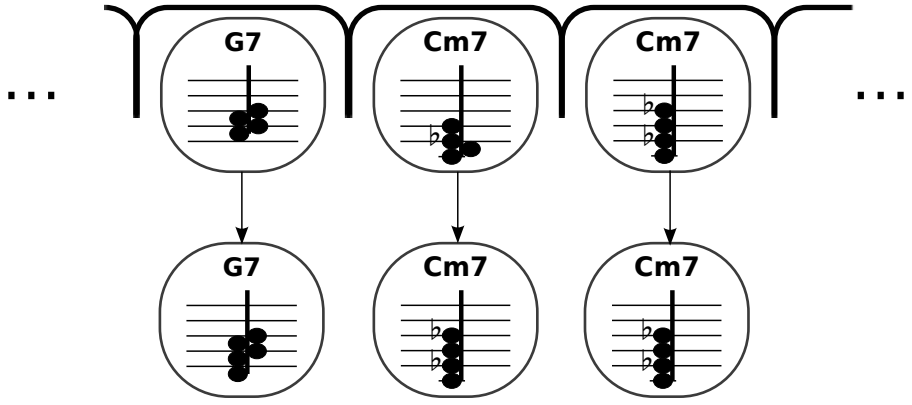


FIGURE 2.7 – Etablissement de la correspondance entre les données et les observations connues

2.3.2 Algorithme de Viterbi

L'algorithme de Viterbi est un algorithme de programmation dynamique qui permet à l'origine de corriger les erreurs de transmissions d'un canal bruité. Ici appliqué à un HMM, il permet de trouver la séquence d'états (c'est-à-dire de labels harmoniques réels) la plus probable ayant produit la séquence d'observations (c'est-à-dire de voicings joués).

Si la séquence est de durée T, on construit une matrice Vit de dimension $N \times T$ qui contient pour tous les indices $i, j : Vit(i,j) = P(X(j) = x_i \mid [Y(t)]_{t \leq j})$ dans le contexte de cette séquence particulière de notes. $Vit(i,j)$ contient donc la probabilité qu'un chemin d'états correspondants aux observations de longueur j arrive à l'état x_i, en prenant les transitions les plus probables et en considérant les probabilités d'observation de chaque état.

Cette probabilité est calculée itérativement sur l'instant j. A chaque instant j, on considère quelle est la meilleure façon d'arriver à l'état x_i depuis tous les autres états précédents possibles, et on multiplie par la probabilité que l'état x_i corresponde à l'observation $Y(j)$.

On a construit en figure 2.8 la matrice Vit correspondante à la séquence d'exemple.

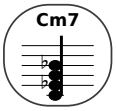
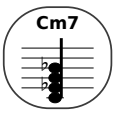
Etats / Observations	G bass 3maj 5 7m	G bass 3maj 5 7m 9	C bass 3m 5 7m
	0.125	0.25	0
	0	0	0.25
	0	0	0.125

FIGURE 2.8 – Exemple de calcul de la matrice de Viterbi

L'algorithme de reconnaissance extrait alors le parcours optimal, et le renvoie sous forme d'une liste d'analized beats, avec le label reconnu., comme montré dans la figure 2.9.

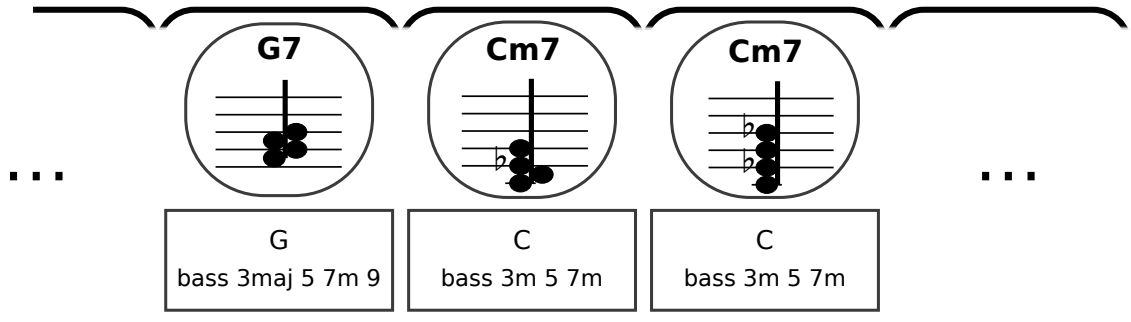


FIGURE 2.9 – Sortie de l'algorithme de reconnaissance

2.3.3 Avantages et limitations du modèle de reconnaissance

L'avantage du HMM est surtout l'influence déterminante du contexte, à travers sa mémoire à court terme et sa prise en compte de la grille théorique. Insistons sur le fait que ce n'est pas le cas des analyses accord-par-accord.

D'autre part, il n'est pas nécessaire d'avoir une connaissance théorique des enchaînements harmoniques, qui sont extraits automatiquement du fichier d'apprentissage. Le modèle est donc valable pour tout type de jeu, suivant une harmonie classique ou non. De fait, la sensibilité et la variété de reconnaissance du modèle sont déterminées totalement par le fichier d'apprentissage. S'il ne comporte pas un certain type d'accord, alors ce dernier ne sera pas pris en charge par le modèle.

Le chiffrage du fichier d'apprentissage est donc déterminant pour l'efficacité du modèle. De plus, le fait d'avoir un apprentissage supervisé aussi précis permettra d'extraire des règles de substitution formelles, et de vérifier a posteriori si les règles de substitution d'une grammaire connue, par exemple celle de Steedman, sont utilisées ou non par le musicien.

Chapitre 3

Substitutions Harmoniques

L'objectif ici est d'enrichir les improvisations créées par ImproteK par des substitutions qui ne se trouvent pas dans l'oracle des facteurs.

3.1 Détection par le Modèle

Qu'est-ce qu'une substitution dans notre modèle ? Il s'agit de faire correspondre à un label de la grille un autre accord. Or, la fonction d'`observations`, fait correspondre à une observation (qui est un label de la grille et un ensemble de notes jouées) une liste de labels réels possibles, avec leurs probabilités. Il suffit donc de réunir toutes les observations correspondant au même accord de la grille. On construit par commodité des champs intermédiaires dans le modèle, pour éviter de les calculer à chaque fois :

- `emissions`, qui fait correspondre à un label de la grille toutes les observations possibles :

$$\begin{array}{rccc} \texttt{emissions} : & \mathcal{G} & \rightarrow & (\mathcal{O} \times [0;1])_{\mathbb{N}} \\ & Acc & \mapsto & ((Acc, N_n), P((Acc, N_n) \mid Acc))_n \end{array}$$

- `substitutions`, qui extrait ainsi la liste de labels véritablement joués qui correspondent à un accord de la grille donné :

$$\begin{array}{rccc} \texttt{substitutions} : & \mathcal{G} & \rightarrow & (\mathcal{L} \times [0;1])_{\mathbb{N}} \\ & Acc & \mapsto & (Y_n, P(Y_n \mid Acc))_n \end{array}$$

On remarque que `substitutions` est une sorte de composée de la fonction `emissions` et de la fonction `observations`. C'est ainsi que l'on obtient une première représentation des substitutions effectuées par le musicien, sous la forme :

$$G7 \rightarrow (c\#, (bass\ 3m\ 5\ 7m))$$

Le modèle peut donc repérer les substitutions utilisées et les restituer sous forme de liste.

3.2 Première idée : Application dans une séquence

Il faut maintenant appliquer les substitutions apprises sur une phrase d'improvisation générée par ImproteK.

3.2.1 Réutilisation de Viterbi

On dispose pour cela d'un HMM que l'on va réutiliser pour la génération. En effet, l'algorithme de Viterbi fait correspondre à une séquence d'observations la suite de labels réels la plus probable de correspondre à ces observations. Si on l'applique sur des observations incomplètes, par exemple qui contiennent seulement les labels de la grille, on peut obtenir une suite de labels réels qui correspondent à la grille. Lors de l'appel de cet algorithme, il suffit de remplacer la fonction `observations` par la fonction `substitutions`, les fonctions `transitions` et `ini-dis` restant intactes. On peut aussi préciser les indices entre lesquels on veut opérer les substitutions, pour éviter de modifier l'ensemble de la phrase.

La sortie de cette étape est une suite de chiffrages alternatifs utilisant les substitutions que le musicien a jouées. Il s'agit maintenant de convertir ces chiffrages en voicings. On cherche donc probabilistiquement des instances de ces chiffrages en regardant dans le fichier d'apprentissage (On peut aussi construire un dictionnaire si on dispose de la mémoire suffisante).

L'ensemble du processus de génération est schématisé en figure 3.1.

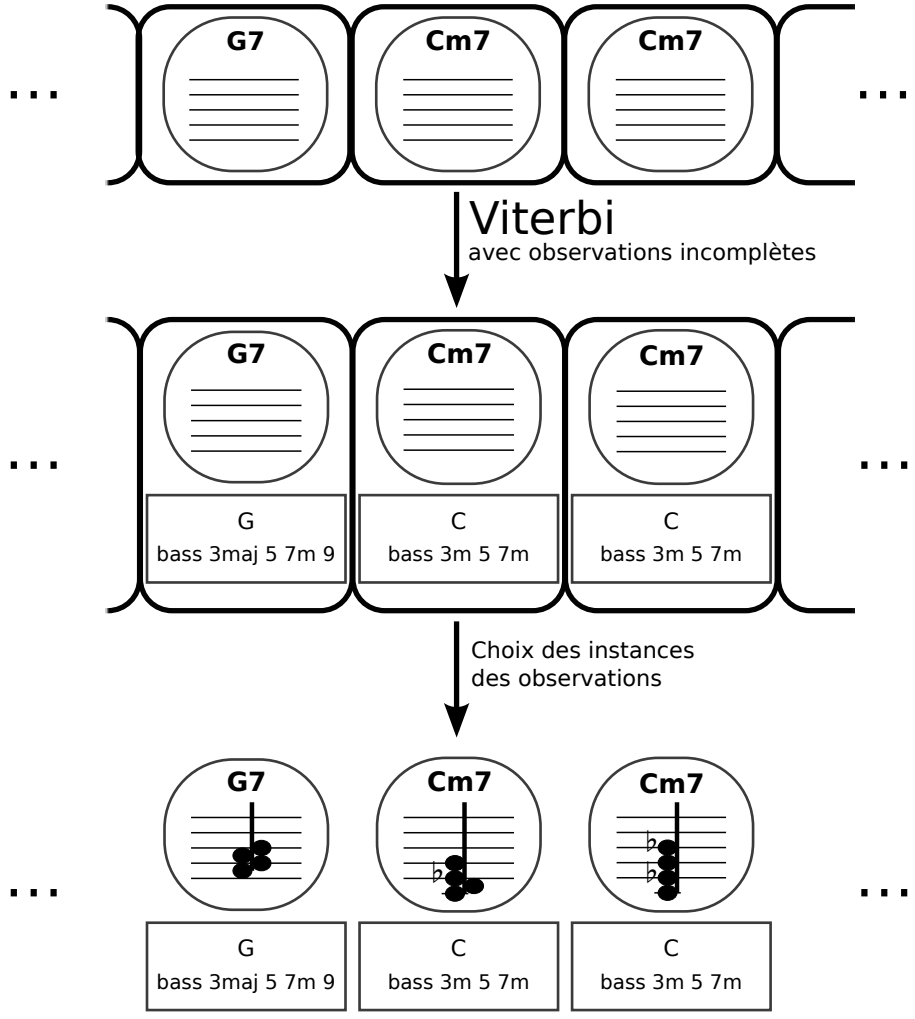


FIGURE 3.1 – Exemple de génération pour une suite de labels de la grille

Il faut noter que la cohérence entre ces instances n'est pas assurée. Il est possible d'effectuer un saut entre deux moments musicaux très différents. Un algorithme de conduite des voix a été envisagé, mais vite abandonné car demandant a priori trop d'informations théoriques à coder. D'autre part, la

richesse des substitutions n'était pas restituée à sa juste valeur : en effet, le programme n'ayant aucune connaissance théorique, il ne pouvait pas distinguer la substitution $(Cmaj^7 \mapsto (c, (bass\ 3maj\ 5\ 7m))$ qui est la "substitution identité" des autres substitutions, il était donc possible de n'appliquer aucune substitution. D'autre part, la substitution $(Cmaj^7 \mapsto (f\#, (bass\ 3maj\ 5\ 7m))$, qui est la substitution au triton, n'a aucun lien formel avec toutes ses transpositions possibles. Par conséquent, on ne trouvait dans le modèle aucune substitution n'ayant pas été jouée, ce qui rendait le modèle obsolète, n'apportant aucune avancée par rapport à l'oracle !

3.3 Formalisation et généralisation des substitutions

La formalisation des substitutions s'est donc imposée naturellement. Il suffit d'apprendre à l'algorithme ce que signifient les chiffrages théoriques des accords : que $Gmaj^7$ correspond à un contenu théorique $(g, (bass\ 3maj\ 5\ 7m))$.

On généralise alors le champ `substitutions` en `subs-generalized`, une fonction qui à un contenu harmonique donné, associe une transposition et un nouveau contenu harmonique, ce qui pourrait s'écrire, pour n'importe quel degré X, par exemple : $(X^7 \mapsto (X+t)^9_7)$. La transposition t marque le changement de fondamentale de l'accord. On remarque que si $t = 0$, alors la substitution est en fait un enrichissement de l'accord (au sens large).

3.4 Et le contexte, alors ?

Les règles extraites sont pour l'instant des règles de substitution ponctuelles, ce qui contredit en essence la nature même d'une substitution harmonique : le but recherché est de trouver un chemin alternatif dans l'espace des états harmoniques, en partant d'un état donné, et en arrivant à un état choisi avec soin. Les substitutions sont opérées pour créer de la surprise tout en ménageant l'auditeur, effectuant des transformations intégrées dans un contexte préparé petit à petit, pour rendre acceptable auditivement une modification harmonique.

La substitution d'un accord dépend donc des accords voisins à l'accord de départ. Selon Steedman ([S84]), l'application d'une substitution dans une grille de jazz dépend de l'accord suivant uniquement. En effet, toutes les règles de sa grammaire, exceptées celles de division des accords, qui n'opèrent pas de changement harmonique, peuvent s'écrire sous la forme

$$accord1\ accord2 \mapsto accord3\ accord2$$

où $accord2$ est déterminant dans la décision d'appliquer la substitution ou non.

Cependant, nous avons choisi d'adopter une position plus générale, et de supposer que l'application d'une substitution dépend de l'accord suivant, mais aussi de l'accord précédent, toujours en suivant l'idée de *direction musicale*.

Les substitutions sont ainsi classées par nature harmonique, qui peuvent être 7, $m7$, $maj7$, $m7b5$, dim. Cette nature harmonique détermine si une substitution est applicable ou non.

On extrait maintenant des substitutions de ce type :

$X7 \mapsto X+4$ avec $(bass\ 3maj\ 5\ 7m\ 9)$ avec une probabilité de substitution $p_{subs} = 0.3$.

1. avec l'accord précédent $X+6$ avec $(bass\ 3maj\ 5\ 7maj)$ avec une probabilité $p_{precedent1} = 0.5$.
2. avec l'accord précédent $X+0$ avec $(bass\ 3maj\ 5\ 7m\ 9)$ avec une probabilité $p_{precedent2} = 0.5$.

1. avec l'accord suivant $X + 5$ avec $(bass\ 3maj\ 5\ 7maj)$ avec une probabilité $p_{successeur1} = 1$.

C'est la substitution que nous allons prendre en exemple pour la suite.

3.5 Deuxième idée : Modèle enrichi et généralisé

A l'aide du formalisme exposé ci-dessus, les possibilités de substitutions sont multipliées : En effet, la généralisation à tous les degrés autorise maintenant les compositions de substitutions, ce qui potentiellement permettrait de générer une infinité de suites de chiffrages réels !

3.5.1 Appliquer en profondeur les substitutions

L'idée est simple : on va *appliquer la substitution extraite au modèle* pour obtenir un modèle enrichi.

Il s'agit de multiplier les chemins dans l'espace des accords, en rajoutant les accords substitués potentiels dans la liste des labels réels possibles. Cependant, lorsque l'accord substituant n'existe pas encore dans le modèle, il faut trouver une manière de l'intégrer.

Prenons le fragment de modèle en figure 3.2 en exemple. Nous allons lui appliquer la substitution citée en exemple dans la partie ci-dessus.

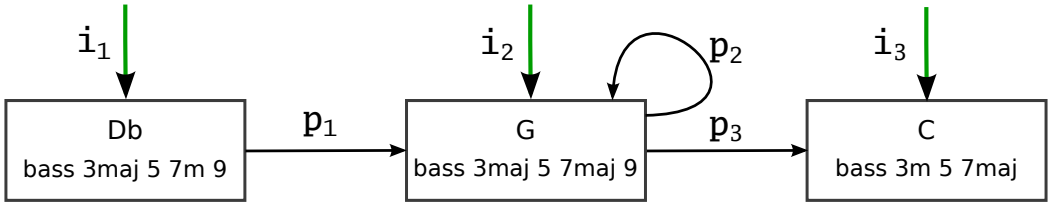


FIGURE 3.2 – Un fragment pertinent de modèle

Pour rajouter des chemins, on va parcourir état par état le modèle, vérifier si la substitution est applicable, et si c'est le cas, créer les états nécessaires et rajouter les transitions dictées par le contexte de la substitution. Nous allons ici traiter le cas où on applique la substitution à l'état du milieu, qui a un contenu harmoniquement compatible avec celui de l'accord $X7$ de la substitution exemple. Ici, harmoniquement compatible signifie inclusion de l'un dans l'autre ou vice-versa, ce qui correspond à des omissions ou des enrichissements harmoniques ne changeant pas la nature profonde de l'accord. Donc la substitution est applicable.

Pour chaque état du modèle, il faut calculer l'état substitué, c'est à dire la transposition et enrichissement de l'accord de base. Ici, il s'agit d'un accord de G de contenu 7 avec neuvième, qui devient un accord de B, de contenu $maj7$. Ici on considère que l'état n'existe pas, donc on le crée. S'il existe déjà, on remplacera le verbe "initialiser à" par le verbe "ajouter à la probabilité existante" dans la suite.

Il faut d'abord agir sur `ini-dis`. Si une substitution est susceptible de se produire, cela diminue la probabilité d'occurence de l'accord initial, et augmente la probabilité d'occurence de l'accord substitué.

On traite alors les transitions entrantes dans l'état initial. Notons qu'on ne change que celles qui sont susceptibles de conduire à une substitution, c'est à dire celles provenant des états potentiellement précédents dans la substitution. On retranscrit le fait qu'une transition sortant de l'état précédent peut aller dans l'état substitué en initialisant une transition de l'état précédent vers l'état substitué, avec une probabilité dépendant à la fois de la probabilité de substitution, de la probabilité de transition

de l'état précédent vers l'état de départ, et de la probabilité de l'état précédent de participer à la substitution. On traite ainsi la transition depuis Db vers G et de G dans lui-même.

On traite de même les transitions sortantes, ici de G vers C.

Les formules exactes sont données en figure 3.3. Le fragment du modèle a été choisi de sorte de ne faire apparaître que les transitions modifiées. Remarquons qu'il peut exister d'autres états qui ont une probabilité de transition non nulle vers (resp. depuis) l'état du milieu mais que ces transitions ne seront pas affectées.

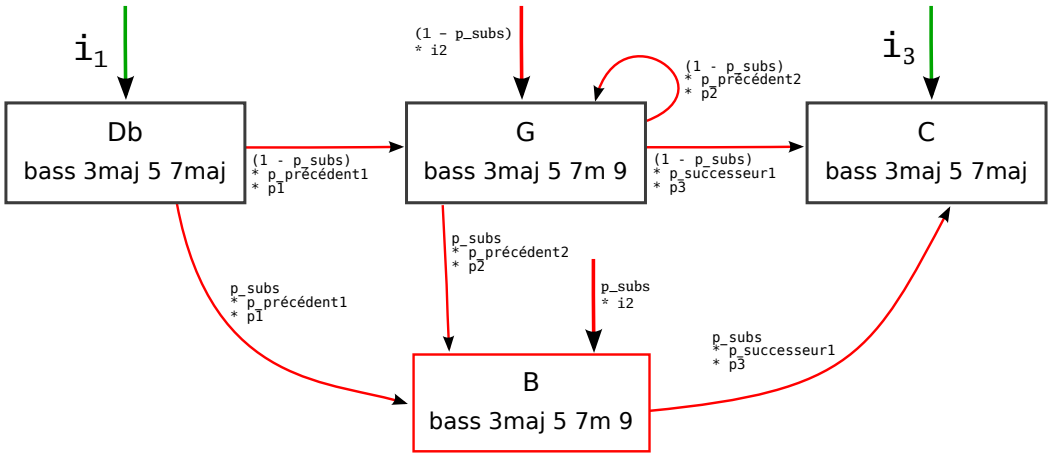


FIGURE 3.3 – Un fragment pertinent de modèle après application de la substitution à l'état du milieu

3.5.2 Cas particuliers

- Si l'accord précédent (resp. suivant) n'existe pas dans le modèle, on le crée et on met à jour les transitions comme prévu dans l'algorithme, mais on laisse à zéro la probabiblité d'occurence de ce nouvel état. Il sera accessible si et seulement si il est issu de l'application d'une autre substitution à un état accessible du modèle.
- Si tous les accords précédents (resp. suivant) possibles sont reliés par une transition de probabilité nulle vers (resp. depuis) l'état que l'on veut substituer, on n'appliquera pas cette substitution à l'état.

On remarque que l'enrichissement du modèle dépend de l'ordre dans lequel on traite les substitutions et les états du système. Il faut faire attention au nombre de calculs, et c'est une question ouverte de savoir si l'algorithme appliqué itérativement au modèle converge ou non. Il serait préférable dans le futur de pouvoir sélectionner les substitutions que l'on veut appliquer, avec éventuellement l'ordre dans lequel on souhaite le faire.

3.5.3 Parcours du modèle enrichi

En parcourant le nouveau modèle avec l'algorithme de Viterbi, la génération produit effectivement une suite de labels harmonique originale, contenant des substitutions non jouées par le musicien, et dont on a montré que l'application dépendait du contexte.

Pour jouer ces substitutions, on va d'abord trier les données midi d'apprentissage par fondamentale et contenu harmonique, ce qui est faisable car elles sont chiffrées, puis on code une fonction transposition primaire qui transpose le contenu midi des données. Il suffit maintenant de choisir selon le contenu harmonique et d'appliquer la transposition correspondante pour obtenir du contenu midi avec la nouvelle fondamentale désirée.

3.6 Problèmes à résoudre

Le principal problème de ce modèle est la totale ignorance de la conduite des voix lors de l'instanciation des labels harmoniques réels. Un travail de fond dans cette direction serait nécessaire, cependant le stage s'est concentré sur les aspects harmoniques, et donc sur une vision verticale de la musique plutôt qu'horizontale, qui correspondrait plus à la notion de chemin minimal entre les notes d'un accord.

Chapitre 4

Expériences

4.1 Bernard Lubat

Ce travail a été l'occasion de plusieurs sessions avec Bernard Lubat à Uzeste, dans le cadre du projet IMPROTEK. Des expériences en Live avec le modèle n'ont pas été possibles à temps, néanmoins beaucoup de matériel musical a pu être exploité.

Marc Chemillier a ainsi chiffré des improvisations de Bernard Lubat sur *D'ici d'en bas*, un morceau extrait de son album *Chansons enjazzées*.

Le module de reconnaissance d'accords a ainsi pu être testé sur toutes le matériel de *D'ici d'en bas* : dûe à la ressemblance entre les différents solos harmoniques de Bernard Lubat, la reconnaissance était efficace à plus de 95% !

Des modèles ont été construits pour modéliser les transitions harmoniques. Voici une retranscription des substitutions extraites pour les accords de nature 7 :

```
X7 --->X+0 with contents (bass 3maj 5 7m 9b) probability 0.33333334
with precedent X+10 with contents (bass 3maj 5 7m 9b) probability 0.375
with precedent X+2 with contents (bass 3maj 5 7m 9b) probability 0.375
with precedent X+5 with contents (bass 3m 5 7m 9) probability 0.125
with precedent X+8 with contents (bass 3maj 5 7maj) probability 0.125

with successor X+10 with contents (bass 3maj 5 7m 9b) probability 0.5
with successor X+2 with contents (bass 3maj 5 7m 9b) probability 0.5

X7 --->X+0 with contents (bass 3maj 5 7m 9#) probability 0.16666667
with precedent X+0 with contents (bass 3maj 5 7m 9#) probability 0.75
with precedent X+0 with contents (bass 3m 5) probability 0.25

with successor X+0 with contents (bass 3maj 5 7m 9#) probability 1.0

X7 --->X+0 with contents (bass 3maj 5 7m) probability 0.5
with precedent X+7 with contents (bass 3maj 5 7m) probability 0.75
with precedent X+5 with contents (bass 3m 5 7m 9 13) probability 0.125
with precedent X+2 with contents (bass 3maj 5 7m 9b) probability 0.083333336
with precedent X+5 with contents (bass 3m 5 7m 9) probability 0.041666668
```

```
with successor X+7 with contents (bass 3maj 5 7m) probability 1.0
```

On remarque que le musicien n'a joué ici que des enrichissements des accords 7 sur ce morceau, et aucune substitution. C'était le cas pour les autres natures d'accord. Il y a plusieurs explications possibles : l'incompatibilité possible des substitutions avec le morceau d'abord, à cause de la grille trop complexe, ou de sa nature de buguine, ou peut-être le style de jeu du musicien qui privilégie les enrichissements étonnants.

Néanmoins, il serait intéressant de faire des sessions en lançant le modèle ayant appris des substitutions par ailleurs sur des improvisations de Bernard Lubat, pour étudier la manière dont ses solos s'adaptent à un accompagnement avec substitutions.

4.2 Blues et Steedman

Des expériences ont été menées pour vérifier si l'on retrouve les règles de Steedman lorsque on analyse des grilles de blues qui sont issues de ces substitutions.

4.2.1 Transcription des règles

Les règles de Steedman peuvent en effet être écrites dans le format que mes règles produisent, en ignorant la liste des accords précédents, car elle est obsolète dans sa grammaire.

Par exemple, une des règles $3a$

$$w\ x7 \mapsto Dx7\ x7$$

devient :

$X7 \mapsto X+y$ pour un certain y avec $(bass\ 3maj\ 5\ 7m)$ avec une certaine probabilité de substitution p_{subs}.

1. avec l'accord précédent ...
2. avec l'accord précédent ...

1. avec l'accord suivant $X+5+y$ avec $(bass\ 3maj\ 5\ 7m)$ avec une probabilité $p_{successeur1} = 1$.

4.2.2 Extraction des règles

On utilise pour l'extraction des règles du matériel généré automatiquement par OpenMusic.

On a réussi à extraire, par exemple, une des règles 4 :

$$Dx7\ x7 \mapsto Stbx7\ x7$$

```
X7 --->X+6 with contents (bass 3maj 5 7m 9 11# 13) probability 0.22222223
with precedent X+0 with contents (bass 3m 5 7m 9) probability 0.5
```

```
with precedent X+5 with contents (bass 3maj 5 7m 9 11# 13) probability 0.5

with successor X+5 with contents (bass 3maj 5 7m 9 11# 13) probability 1.0
```

En effet, cette règle transforme bien un accord de dominante (du successeur $X+5$) en un accord de sus-tonique bémol, lorsque le successeur est la tonique avec probabilité 1, ce qui rend la condition de successeur nécessaire, comme indiqué par la grammaire de Steedman. On remarque que l'on a extrait un enrichissement à la 9^e, la $11^e\#$ et la 13^e.

On pourrait aussi écrire les règles de *Steedman inversée* correspondante à cet accord, c'est-à-dire les règles en ne considérant que les accords précédents (successeurs dans une progression temporelle inversée).

Les deux règles correspondantes seraient (sans enrichissements) :
Règle -4

$$xm7\ x7 \mapsto xm7\ Dbx7$$

$$Sdx7\ x7 \mapsto Sdx7\ Dbx7$$

où Sdx est l'accord de sous-dominante de x, et Dbx (ou $Sd\#x$) est l'accord de dominante bémol (ou sous-dominante dièse).

On peut ainsi écrire toute une grammaire différente. Il serait intéressant de chercher l'ensemble minimal de règles *inversées* capables de générer toutes les grilles de blues.

4.3 Génération audio

A partir des phrases générées par l'oracle, des substitutions ont été générées sur quelques morceaux.

Des exemples midi seront bientôt mis en ligne sur le site de Marc Chemillier.

Chapitre 5

Conclusion et avancées

Pendant ce stage, on a développé un modèle avec comme fonctions principales :
- La reconnaissance d'un accord à partir d'un beat midi
- L'extraction de règles de substitutions formelles
- La génération de phrases d'accompagnements avec imitation stylistique des substitutions originales

Le HMM utilisé permet de prender en compte le contexte, qui est déterminant lors du chiffrage automatique d'accords, et lors de l'extraction et l'application de règles de substitution qui ne sont pas ponctuelles mais insérées dans un chemin à travers les états harmoniques.

La formalisation de ces règles a permis de construire les nouveaux enchaînements harmoniques introduits par une substitution, qui à leur tour ont résolu l'impossibilité de jouer des substitutions non contenues dans l'oracle, en utilisant un modèle enrichi par ces enchaînements.

Le modèle enrichi dépend beaucoup de l'ordre dans lequel on appelle les substitutions et les états, mais il serait intéressant de contrôler manuellement. D'autre part, un algorithme de conduite des voix ferait beaucoup gagner en cohérence des générations, lorsqu'on choisit un beat correspondant au contenu harmonique réel recherché. On pourrait suggérer un parcours contraint sur les instances, en considérant que la beatlist est en puissance un oracle où les instances des beats sont reliées entre elles par des liens suffixiels et de transition.

La suite logique de ce travail serait de développer une version temps-réel de ce travail, qui reconnaisse selon ce que le soliste improvise les accords joués avec le module de reconnaissance et en même temps lancer un algorithme de génération pour modifier en temps réel les accords qu'il allait jouer, ce qui rendrait son accompagnement sensible aux substitutions jouées. Le solo d'un musicien dépendant aussi de l'accompagnement, l'interaction générerait de nouvelles séquences harmoniques, où le soliste et l'ordinateur effectuent à la fois l'analyse du présent et l'anticipation du futur de l'improvisation.

Ainsi, le module de reconnaissance d'accords permettrait de jouer à terme sans grille théorique, et laisser l'interaction entre le musicien et l'ordinateur prendre des directions plus libres. On reviendrait ainsi dans un contexte de *free jazz*, tout en gardant une pulsation, qui est à la base du module de reconnaissance d'accords ... jusqu'à ce que l'on trouve une autre manière efficace de segmenter le flux midi en unités harmoniques cohérentes !

Bibliographie

[U77] W. Ulrich. *The Analysis and Synthesis of Jazz by Computer*. Fifth International Joint Conference on Artificial Intelligence, MIT, Cambridge, Ma, pp. 865-872, 1977.

[M92] H.J. Maxwell. *An Expert System for Harmonic Analysis of Tonal Music*. In M. Balaban, K. Ebcioglu, and O. Laske, editors.*Understanding Music with AI : Perspectives on Music Cognition*. AAAI Press, 335-353.

[T97] D.Temperley. *An Algorithm for Harmonic Analysis*. MusicPerception, vol 15, No. 1, Fall 1997.

[BP02] B. Pardo, W. P. Birmingham. *Algorithms for chordal analysis*. Computer Music Journal, 2002.

[IIR07] P.R. Illescas, D. Rizo, and J.M. Inesta. *Harmonic, melodic, and functional automatic analysis*. Proceedings of the International Computer Music Conference, 2007.

[C00] E. Chew. *Towards a mathematical model of tonality*. Master's thesis, Operations Research Center, MIT, Cambridge, USA, 2000.

[CC07] C.H. Chuan and E. Chew.*A hybrid system for automatic generation of style-specific accompaniment*. 4th Intl Joint Workshop on Computational Creativity, 2007.

[B01] J. Barthelemy. *Figured bass and tonality recognition*. In ISMIR, 2001.

[KK82] C. Krumhansl and E. Kessler. *Tracing the Dynamic Changes in Perceived Tonal Organisation in a Spatial Representation of Musical Keys*. Psychological Review, volume 89, pages 334–368, 1982.

[K90] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, 1990.

[JL93] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, Cambridge, MA, 1993.

[T01] D. Temperley. *The cognition of basic musical structures*. The MIT Press, 2001.

[HRR09] T. Rocher, M. Robine, P. Hanna. *Amélioration des méthodes d'estimation d'accords et de tonalité depuis une représentation musicale symbolique*. Journées d'Informatique Musicale, France, 2009.

[R89] L. R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.

[ES03] A. Sheh, D. P. Ellis. *Chord segmentation and recognition using EM-trained hidden Markov models*. Proceedings of the International Symposium on Music Information Retrieval, Baltimore, MD, 2003.

[BP05] J. P. Bello, J. Pickens. *A robust mid-level representation for harmonic content in music signals*. Proceedings of the International Symposium on Music Information Retrieval, London, UK, 2005.

[LS06] K.Lee, M.Slaney. *Automatic chord recognition using an HMM with supervised learning*. Proceedings of the International Conference on Music Information Retrieval, Victoria, Canada, 2006.

[HS05] C. Harte, M. Sandler. *Automatic chord identification using a quantised chromagram.* Proceedings of the Audio Engineering Society, Madrid, 2005.

[RS03] C. Raphael, J. Stoddard. *Harmonic analysis with probabilistic graphical models.* Proceedings of ISMIR 2003, 2003.

[P96] H. Pascoal. *Calendário do Som.* Disponible en partie sur http ://www.hermetopascoal.com.br/, Senac, 2000 (Composé en 96-97).

[CLN99] L. Costa-Lima Neto. *The experimental music of Hermeto Pascoal and group (1981 – 1993) : Conception and language (Mestrado em muusica brasileira).* Universidade do Rio de Janeiro, Centro de letras e artes, 1999.

[S84] M.J. Steedman. *A Generative Grammar for Jazz Chord Sequences.* Music Perception, 2, pp. 52-77, 1984.

[JL91] P. N. Johnson-Laird. *Jazz Improvisation : A Theory at the Computational level.* In P. Howell, R. West and I. Cross. *Representing Musical Structures.* Academic Press, pp. 291-325, 1991.

[C04] M. Chemillier. *Grammaires, automates et musique* . In Fr. Pachet, Brio J.-P. *Informatique musicale. Du signal au signe musical.* Paris, Hermès, coll. IC2. Informatique et systèmes d'information, 2004.

[P98] F. Pachet. *Computer Analysis of Jazz Chord Sequences. Is Solar a Blues ?.* Readings in Music and Artificial Intelligence, Miranda E., Ed., Harwood Academic Publishers, 1998.

[ACR99] C.Allauzen, M.Crochemore, M.Raffinot. *Factor oracle : A new structure for pattern matching.* SOFSEM 99 : Theory and Practice of Informatics. Springer, 1999.

[AD04] G. Assayag, S. Dubnov. *Using Factor Oracles for machine Improvisation.* Soft Computing, vol. 8, n9, Septembre, 2004.

[N12] J. Nika. *ImproteK : Intégrer des contrôles harmoniques pour l'improvisation musicale dans la filiation d'Omax.* Journées d'Informatique Musicale, Belgique, 2012.

[BP66] L. E. Baum, T. Petrie. *Statistical Inference for Probabilistic Functions of Finite State Markov Chains.* The Annals of Mathematical Statistics 37, 1966.

[R89] L. R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.