Étude d'algorithmes pour la détection de la tonalité de fichiers musicaux et implémentation en Clojure

Antoine Passemiers

Université Libre de Bruxelles
apassemi@ulb.ac.be

Résumé

Le projet consiste en la discussion de différents algorithmes relatifs à la détection automatisée de tonalité de fichiers musicaux, l'implémentation de ceux-ci en Python, ainsi qu'une réflexion sur la comptabilité de ces derniers avec le paradigme fonctionnel. Le choix de l'algorithme a concevoir selon l'approche fonctionnelle sera basé sur des critères de rapidité d'exécution et de précision de la détection. L'algorithme final sera alors implémenté en Clojure.

1. Introduction

La tonalité d'une oeuvre musicale se caractérise par l'ensemble des sons formant une même gamme diatonique. A la différence de la gamme, où les sons se succèdent de façon contigüe, la tonalité (ou ton) regroupe des sons qui peuvent être disjoints et/ou superposés (Danhauser, 1929). En conséquence, nous nous intéressons à l'analyse de mélodies polyphoniques, où plusieurs notes peuvent être jouées en même temps.

En particulier, nous allons nous pencher sur deux catégories d'algorithmes : ceux basés sur des modèles cognitifs, et ceux incluant des notions d'apprentissage automatique. Les premiers tentent d'intégrer au mieux les connaissances de la théorie musicale et reposent sur la façon dont les personnes reconnaissent les différentes tonalités, alors que les autres utilisent l'inférence statistique pour déterminer celles-ci.

L'approche cognitive utilisée pour la détection de la tonalité repose en partie sur la solution proposée par Ibrahim Sha'ath lors de la conception du logiciel KeyFinder (Sha' ath, 2011). La précision de la détection est évaluée à l'aide d'une base de données, constituée de 250 fichiers musicaux au format wav, dont les tonalités sont connues et inscrites dans un fichier csv. Ces fichiers font partie de ceux utilisés par Ibrahim Sha'ath dans le cadre de sa recherche.

Pour ce qui est de la partie apprentissage automatique, les méthodes présentées seront principalement en lien avec les modèles de Markov cachés. Leur évaluation se fera en divisant la base de données en un jeu de données d'apprentissage et un jeu de données de validation (respectivement 60 % et 40 % du jeu de données d'origine). Ce mini-mémoire se voulant concis et centré sur les objectifs décrits, le lecteur est supposé déjà disposer de connaissances suffisantes en apprentissage automatique, en théorie musicale et en traitement logiciel de signaux.

TODO : (Pauws, 2004) (Takeuchi, 1994) -> Partie 2.1.1.

2. Considérations théoriques

2.1. Pré-traitement et estimation spectrale

2.1.1. Constant-Q Transform (CQT) Le signal audio est premièrement extrait du fichier wav, puis la moyenne entre les deux canaux est effectuée si le fichier a été enregistré en stéréo. En effet il n'est pas nécessaire de prendre en compte le panoramique puisque celui-ci n'a que peu d'influence sur la mélodie dans le domaine spectral. Étant donné que les notes jouées sont uniquement caractérisées par leur fréquence fondamentale, il n'est pas nécessaire de considérer l'entièreté du spectre du fichier musical. De fait, la fréquence d'échantillonnage est abaissée à un dixième de la fréquence standard (4410 Hz), mais ce sous-échantillonnage est susceptible de provoquer des phénomènes d'aliasing. Contrairement à la solution de Sh'ath, qui gère les problèmes d'aliasing sonore par l'application d'un filtre passe-bas, une approche plus simpliste et plus rapide se limiterait à l'application d'une moyenne mobile sur une fenêtre glissante de taille arbitraire. L'avantage de la méthode est de bénéficier d'effets passe-bas sans devoir effectuer de calculs dans le domaine fréquentiel. Pour ce qui est de la taille de la fenêtre temporelle, il s'agit d'un hyper-paramètre fixé durant l'évaluation de l'algorithme. (Sha' ath, 2011)

Ensuite, une fenêtre de Blackman est appliquée sur les données observées afin d'éviter les distortions spec-

trales dues à la taille limitée de la fenêtre (et au principe d'incertitude d'Heisenberg). Le spectre est alors approximé à l'aide de la transformée de Fourier rapide (FFT). L'intuition derrière la Constant-Q Transform (CQT) est de penser que les coefficients de la FFT dont les fréquences ne correspondent pas à des notes de musique prennent plus de poids que les autres coefficients. Ceci doit être réajusté en appliquant des fenêtres spectrales centrées sur les notes de musiques. Pour chaque fenêtre, les coefficients résultants de cette opération sont alors sommés pour ne former qu'un seul coefficient spectral. L'ensemble des coefficients globaux constitue alors la CQT.

2.1.2. Méthodes basées sur l'auto-corrélation Les algorithmes reposant sur l'utilisation de fonctions d'auto-corrélation sont plus souples car l'auto-corrélation peut être calculé de diverses manières, et seulement pour les fréquences voulues. En effet, choisir le lag permet de fixer la fréquence voulue et, par extension, il est possible d'analyser l'auto-corrélation pour toutes les notes musicales situées entre 65 Hz et 1109 Hz en évaluant la valeur de la fonction d'auto-corrélation pour tous les lags correspondants.Le lag est alors simplement égal à la période voulue. Un autre avantage des fonctions d'auto-corrélation est que celles-ci sont indépendantes des phases du spectre.

2.1.3. Estimation spectrale par moindres carrés Il est possible d'estimer le spectre par régression, au travers d'algorithmes n'exigeant pas de grandes quantités de calcul. Parmi les algorithmes les plus importants résident celui de Vaníek et celui de Lomb-Scargle. Leur particularité est de pouvoir traiter des données qui ne sont pas reçues à intervalles réguliers (la majorité des observations se font la nuit). En outre, contrairement à la transformée de Fourier rapide qui possède une résolution aussi précise que la fenêtre d'observation est grande, ces algorithmes calculent un périodogramme dont la taille est fixée par le nombre de fréquences qui nous intéressent. Au plus le nombre de fréquences analysées est élevé, au plus le temps d'exécution sera conséquent.

$$\hat{\theta}_k = (A_k^T A_k)^{-1} A_k^T y \tag{1}$$

Selon la méthode de Vanícek, le signal est supposé être centré sur zéro et représenter une combinaison linéaire de sinusoïdes et d'un bruit blanc. Les coefficients de cette combinaison linéaire sont réunis dans un vecteur θ_k, de telle manière que le signal équivaut à $y \approx A\theta_k$. La matrice A est telle que chaque ligne de celle-ci constitue une sinusoïde de fréquence d'intérêt. Les coefficients sont alors finalement donnés par l'équation 1. (Petr Stoica, 2009)

Le défaut de la méthode est de ne pas considérer les phases des fréquences concernées. En effet chaque sinusoïde contenue dans la matrice A est supposée posséder une phase nulle. La méthode de Lomb-Scargle permet de résoudre ce problème en pré-calculant les phases des sinusoïdes utilisées et en tenant compte au mieux de celles-ci lors du calcul du spectre. Le déphasage τ correspondant à la fréquence f est donné par l'équation 2. (Lomb, 1975)

$$\tan 2\pi f\tau = \frac{\sum_{j=1} \sin 2\pi f t_j}{\sum_{j=1} \cos 2\pi f t_j} \tag{2}$$

Tout comme dans la méthode de Vanícek, la régression consiste en la recherche de coefficients qui expliquent au mieux le signal sous la forme d'une pondération de sinusoïdes. Le coefficient associé à la fréquence f est donné par l'équation 3 :

$$\Delta R(f) = \frac{(YC)^2}{CC} + \frac{(YS)^2}{SS} \tag{3}$$

Les valeurs CC et SS peuvent être pré-calculées, car les fréquences d'intérêt ne changent pas d'un fichier à l'autre.

$$CC = \sum_{j=1} \cos^2 2\pi f(t_j - \tau) \tag{4}$$

$$SS = \sum_{j=1} \sin^2 2\pi f(t_j - \tau) \tag{5}$$

Les valeurs YC et YS cherchent à mesurer respectivement les niveau d'orthogonalité du signal y avec une sinusoïde déphasée de $\pi/2$ et avec une sinusoïde de phase nulle. Ces sinusoïdes peuvent également être pré-calculées. Les formules permettant de calculer YC et YS sont :

$$YC = \sum_{j=1} y_j \cos 2\pi f(t_j - \tau) \tag{6}$$

$$YS = \sum_{j=1} y_j \sin 2\pi f(t_j - \tau) \tag{7}$$

Ainsi, les seules opérations restantes lors de l'exécution de l'algorithme sont les produits scalaires entre les sinusoïdes et le signal délimité par la fenêtre glissante, ce qui réduit drastiquement la quantité de calculs requis pour l'estimation du spectre.

2.1.4. Autres méthodes TODO : Cepstre, spectre de puissance, ...

2.2. Prédiction de la tonalité

2.2.1. Modèle cognitif

2.2.2. Modèles de Markov Cachés (HMM) Les modèles de Markov cachés sont des machines à états discrets cherchant à représenter des séries multivariées par leurs distributions, ainsi que par les probabilités de transition entre les états cachés de la machine. De plus, chaque état caché de cette dernière possède ses propres probabilités d'émission. Contrairement à des modèles d'apprentissage automatique plus populaires tels que les réseaux de neurones ou les machines à vecteurs de support, les HMM sont capables de traiter des séquences de longueur non fixée. Cette caractéristique est appréciable dans le cadre de l'analyse de morceaux de musique, qui ont des durées de nature très variables.

TODO : (Peeters, 2006) (Ramage, 2007)

2.2.3. Modèles de Markov Cachés de type entrée-sortie (IO-HMM) TODO : (Bengio and Frasconi, 1996)

Méthode	ACC	REL	PAR	OBF	TOT
COGN	0,307	0,148	0,095	0,042	0,593
EAA	–	–	–	–	–
HMM	–	–	–	–	–
IO-HMM	–	–	–	–	–

Table 1: Évaluation des méthodes présentées selon différents indices : ACC (accuracy), REL (relative keys), PAR (parallel keys) et OBF (out-by-a-fifth keys). Le tableau reprend les scores du modèle cognitif (COGN), du modèle d'autocorrélation, des modèles de Markov cachés (HMM), et du modèle de Markov caché d'entrée-sortie (IO-HMM).

2.2.4. Résultats

Implémentation en Clojure

TODO : (Kumar, 2015)

References

Bengio, Y. and Frasconi, P. (1996). Input-output hmm's for sequence processing.

Danhauser, A. (1929). *Théorie de la Musique*.

Kumar, S. (2015). Clojure high performance programming.

Lomb, N. R. (1975). Least-squares frequency analysis of unequally spaced data.

Pauws, S. (2004). Musical key extraction from audio. *Proceedings of the 9th International Conference on Digital Audio Effects, DAFx-06, Montreal, Canada*, pages 96–99.

Peeters, J. (2006). Musical key estimation of audio signal based on hidden markov modeling of chroma vectors. *Proceedings of the 9th International Conference on Digital Audio Effects, DAFx-06, Montreal, Canada*.

Petr Stoica, Jian Li, H. H. (2009). Spectral analysis of nonuniformly sampled data : A new approach versus the periodogram.

Ramage, D. (2007). Hidden markov models fundamentals.

Sha' ath, I. (2011). Estimation of key in digital music recordings. pages 1–64.

Takeuchi, A. H. (1994). Maximum key-profile correlation (mkc) as a measure of tonal structure in music. *Perception & Psychophysics*, pages 335–346.