

VERS UN SYSTEME D'IDENTIFICATION AUTOMATIQUE
D'AUTEURS DE MUSIQUE

Martin PERNOLLET

Introduction

Music Information Retrieval (MIR), ou recherche d'information musicale, est un sujet largement étudié. On trouvera ainsi dans la littérature de nombreuses voies explorées, parmi celles-ci : classification de styles musicaux par réseaux de neurones [NEU], classification de mélodies par la mesure de leur complexité [KOL], classifications basées sur la reconnaissance de schémas approximativement répétés [REP], ou encore identification de profils harmoniques [HAR].

Les différentes méthodes de classification musicale peuvent être globalement regroupées en deux catégories :

1. Apprentissage de séquence, généralement mis en œuvre au travers de chaînes de Markov. Le principe est de déterminer des probabilités de changement d'états harmoniques dans l'évolution d'un morceau et ainsi d'élaborer un modèle propre à la catégorie des musiques ayant servi de support à l'apprentissage. [BUZ] propose ainsi un modèle permettant de générer des chorales à *la manière* de Bach.
2. Apprentissage intemporel, ne prenant pas en compte l'ordre des informations. La méthode est basée sur une mesure de similarité entre deux ensembles de vecteurs décrivant les morceaux. Plusieurs déclinaisons ont été proposées : entropie relative de la concaténation des vecteurs décrivant les objets analysés [CUT], complexité de Kolmogorov moyenne des deux ensembles [KOL], etc.

Notre défi est d'identifier un compositeur de pièces de piano classique en apprenant un ensemble de ses partitions, donc de reconnaître des styles de composition (et non plus des styles de musique). On souhaite ainsi pouvoir déterminer quelle est la probabilité pour qu'une partition nouvellement présentée au système ait été écrite par ce même auteur.

Cette recherche sera en partie guidée par la lecture de [MDT], documentation d'un Toolkit musical pour Matlab présentant de nombreuses notions pour le traitement statistique de musiques. Ce toolkit a par ailleurs été utilisé car il permet de travailler avec le format MIDI.

Nous avons choisi ce format car il évite l'étape de traitement audio, un fichier MIDI ne stockant aucune donnée sonore, mais uniquement des informations concernant les notes du morceau :

- hauteur (*pitch*)
- vitesse (force du jeu)
- instant de déclenchement
- durée

N'ayant pas d'a priori sur la méthode la plus adaptée parmi celles citées ci-dessus (1 ou 2), des travaux préliminaires ont été effectués dans les deux directions pour déterminer celle susceptible de mener le plus efficacement au résultat. Une présentation de ces travaux fera l'objet d'une première partie.

Ces essais, et la lecture d'une partie des articles cités ci-dessus, ont permis de réfléchir à une approche qui sera présentée dans une deuxième partie. Les résultats seront discutés, et des pistes pour poursuivre la recherche seront proposées.

I. Travaux d'approche

I.1. Approche séquentielle

Les travaux de Buzzanca & Fernandez [BUZ], respectivement musicologue et statisticien, montrent qu'une modélisation des séquences d'accords par Hidden Markov Model est une méthode efficace pour générer des chorales à la manière de Bach.

Notons cependant que les travaux de [BUZ] s'affranchissent de l'étape d'inférence automatisée en transformant manuellement la partition en grille harmonique¹. Or ceci n'est pas possible en ce qui concerne notre projet pour deux principales raisons :

- Leur projet faisait intervenir une vingtaine de personne pour effectuer la conversion.
- Les chorales de Bach sont construites sur 4 voies construisant des accords de 4 notes. Le piano offre un potentiel de 10 notes simultanées et rend l'analyse et la conversion beaucoup plus longues.

Néanmoins, il semblait intéressant de suivre cette piste, en tentant d'inférer, de façon automatisée, la séquence d'accord correspondant à une pièce de piano à partir de sa partition.

Inférence automatisée de grille harmonique

Supposant que les évolutions harmoniques peuvent être l'un des critères de discrimination d'un compositeur, j'ai orienté la première étape de mon projet sur la reconnaissance de grille harmonique.

Différentes approches existent concernant l'inférence des séquences d'accords d'un morceau. L'outil [MDT] implémentant les idées de Krumhansl (1990), j'ai choisi d'appliquer cette méthode. Celle-ci consiste à associer la distribution des notes d'une séquence à l'une des tonalités de référence la plus probable parmi les 24 accords de base (les 12 accords C, C#, ... B² déclinés en majeurs ou mineurs). Nous constatons, grâce à ce projet, que cette approche demeure efficace pour des partitions très simples, voire monophoniques³.

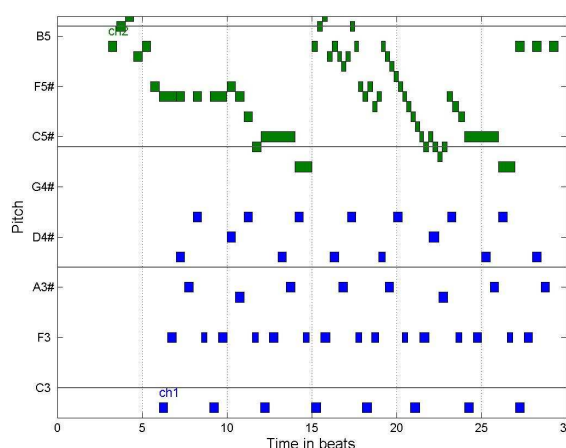
Le cas étudié était le premier nocturne de Chopin (opus 9 n°1), qui crée d'entrée une ambiguïté avec sa mélodie, en répétant des quintes qui perturbent la distribution et faussent l'inférence. D'autres morceaux, tels que la Sonate au Clair de Lune de Beethoven, ont révélé cette même faiblesse⁴.

Optimisation : séparation des mains

Il semblait ainsi judicieux de tenter de séparer la main droite - associée à la mélodie - de la main gauche qui fixe, souvent, sous forme d'accord ou d'arpège l'harmonie du passage.

Une première étape a consisté à séparer les notes en deux groupes de part et d'autre d'une note de référence. Cette approche simple s'est révélée efficace ponctuellement, mais perdait sa pertinence dès lors que les deux mains passaient d'un même côté de la frontière séparatrice.

Afin de remédier à ceci, un algorithme de clustering a été appliqué pour tenter de séparer les notes aiguës de la mélodie et les notes graves.



¹ Séquence de lettres caractérisant la tonalité de chaque sous-séquence du morceau.

² La notation anglo-saxonne utilise des lettres pour définir les noms de notes, et débute à La (A) pour terminer à Sol (G).

³ Monophonie : aucune note ne peut être jouée tant que la précédente est tenue.

⁴ Les auteurs de [MDT] reconnaissent ce problème mais ne propose pas à ce jour de solution générique.

Mais cette méthode n'a pas pu convenir de façon générale car les mains du pianiste sont parfois trop proches, ou les arpèges joués à la main gauche trop amples par rapport à une mélodie évoluant dans une zone mélodique trop étroite, et la séparation est alors mal placée.

I.2. Approche intemporelle

Suite à l'échec de cette approche, j'ai tenté d'apprendre les fragments de musique de façon brute : chaque morceau est découpé en fragments de 50 notes successives. Les fragments sont des fenêtres glissantes, où chaque fenêtre $f+1$ débute n notes après la fenêtre f .

L'opération a été appliquée sur 7 morceaux de Chopin et 9 morceaux de Beethoven. L'algorithme d'apprentissage était SVM, utilisant le noyau gaussien de largeur s :

$$e^{-|f_i - f_j|^2 / s}$$

Paramètres de l'expérience:

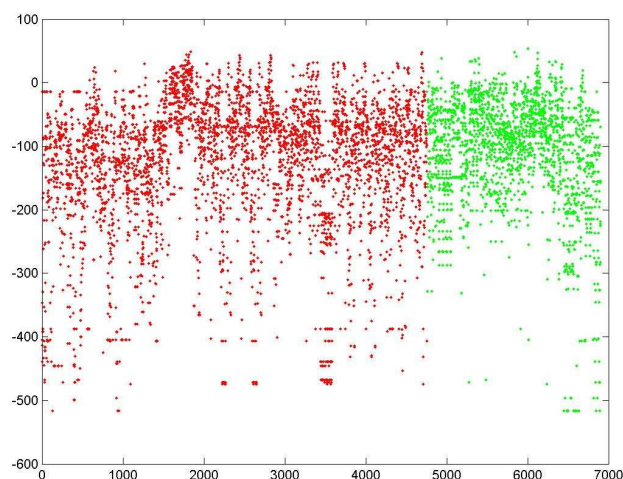
$s = 1$

90% des fragments ont servi à l'apprentissage, l'intégralité au test.

$n = 10$ offset entre deux fenêtres.

Légende:

Les points rouges représentent les fragments de morceaux de Beethoven, les points verts représentent les fragments des morceaux de Chopin. L'axe des ordonnées indique la distance à la ligne de séparation (axe $y=0$).



On peut voir dans cet exemple que l'essentiel des fragments a été mal classé : les fragments de morceaux de Chopin sont généralement classés dans la catégorie Beethoven.

Je pense que cette segmentation n'a pas de sens musicalement parlant : 50 notes peuvent être jouées sur 10 mesures comme sur 1 mesure, voire sur 13/5 de mesure. Le fragment ne représente ainsi pas une portion cohérente sur le plan musical.

Une analogie avec le texte consisterait à faire un apprentissage basé, non pas sur des mots, mais sur une séquence de 50 lettres représentées par leur position dans l'alphabet ($a=1$, $b=2$, etc) : on perdrait le sens de chaque mot en ne conservant pas le découpage permettant de donner du sens, à savoir considérer l'espace comme séparateur d'une séquence.

Il convient donc de proposer une méthode de découpage qui prenne en compte une dimension musicale prenant plus de sens : le temps. Considérer un fragment d'un temps ou deux temps permet de dégager une sous-séquence ayant une unité harmonique.

Plus intuitivement, on notera qu'un auditeur, pour se repérer dans le déroulement d'un morceau, bat le temps, mais ne compte pas les notes. De la même façon, un lecteur cherche les espaces entre les séquences de lettre pour extraire des mots.

Nous chercherons donc, dans la suite de nos travaux à découper les fragments selon un intervalle rythmique. Cela nous amènera à définir des fonctions afin de transformer des segments de longueur inégale en des vecteurs de dimension normalisée. La partie suivante proposera ainsi une série de descripteurs permettant de caractériser différents aspects de la musique (harmonie, mélodie, rythme, etc).

II - Mesure de similarité entre musiques

La conception des descripteurs a été influencée par la lecture de [MDT] qui propose de nombreux outils d'analyse musicale. Néanmoins, des améliorations ont dû être apportées, le toolbox cité ne fournissant pas systématiquement des fonctions adéquates à l'étude de morceaux polyphoniques⁵.

L'utilisation des descripteurs proposés nécessite la fragmentation de chaque morceau étudié. On extrait un fragment de 4 temps tous les $\frac{1}{2}$ temps et obtient ainsi des listes de notes de longueur inégale. Afin de transformer ces fragments en vecteurs de dimension fixe, on applique diverses fonctions caractérisant un aspect musical du fragment :

- Distribution des hauteurs de notes du fragment (Pitch Class Distribution)
- Distribution des intervalles entre notes successives (Pitch Jump Distribution)
- Distribution des durées des notes du fragment (Duration Class Distribution)
- Nombre de notes du fragment (Number of Notes)

Un morceau est donc défini par 4 listes de vecteurs. La première étape consiste à étudier l'efficacité dans le regroupement de morceaux par auteur en utilisant chaque descripteur indépendamment.

La mesure de similarité entre groupes de vecteurs est issue des travaux de Cuturi & Vert [CUT]. Le principe est de définir la distance entre deux nuages de points (de taille inégale) en mesurant l'entropie relative des listes concaténées. Ainsi, deux nuages absolument identiques ont une entropie relative nulle, deux nuages fondamentalement différents ont une entropie relative forte.

Nous détaillons ci-après la définition des descripteurs. Dans un second temps nous montrerons comment est calculé le *entropy kernel* suggéré par [CUT], qui nécessite l'utilisation de l'estimateur de densité de Parzen. Pour apporter une compréhension concrète de la méthode, nous commenterons enfin l'implémentation d'une mesure de similarité pour l'un des descripteurs.

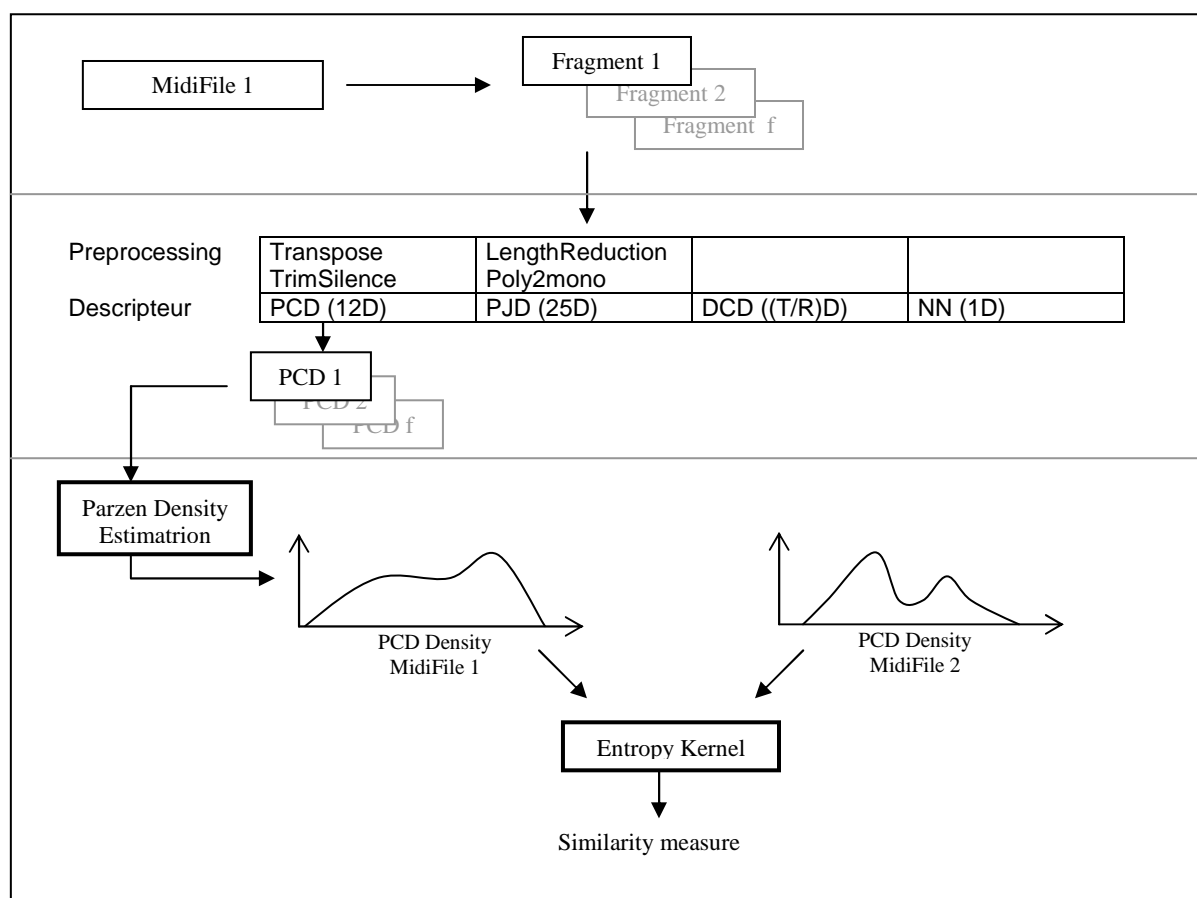


Schéma récapitulatif de la mesure de similarité entre morceaux

⁵ Polyphonie : plusieurs notes peuvent être superposées.

II.1. Descripteurs utilisés

PCD : descripteur harmonique

La *Pitch Class Distribution* [**MDT**] mesure la répartition des notes du fragment étudié. C'est une distribution discrète sur 12 valeurs (C, C#, D, D#, ... B), calculée en comptant le nombre de notes de chaque classe, et en pondérant chaque note par sa durée. Elle permet ainsi de définir l'*état harmonique* du fragment en supprimant l'information temporelle.

On considérera ensuite pour la mesure de similarité que chaque état harmonique (i.e. distribution) est un point de l'espace $[0,1]^{12}$.

La PCD ne donne cependant aucune information sur la nature du message mélodique. Ce descripteur est le même, que l'ensemble des notes jouées soit ordonnées en une progression ascendante, ou de façon désordonnée, alors que le message mélodique entendu est radicalement différent. Ainsi, nous devons introduire un second descripteur prenant en compte les intervalles entre notes successives.

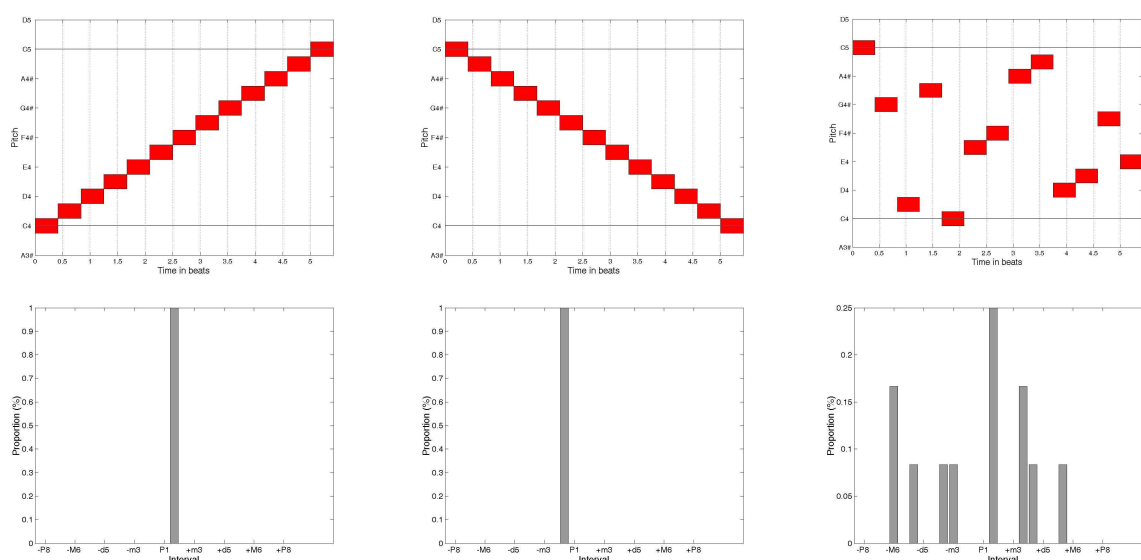
PJD : descripteur mélodique

La *Pitch Jump Distribution* répond à ce besoin en mesurant la répartition des écarts tonals entre notes. Une progression chromatique⁶ ascendante peut donc être distinguée d'une progression chromatique descendante.

Cette approche proposée par [**MDT**] ne convient cependant qu'à des morceaux monophoniques : on ne peut pas définir un intervalle entre une note et un accord. Nous avons ainsi appliqué le pré-traitement suivant pour transformer un morceau polyphonique en morceau monophonique⁷ :

- Réduire la durée de chaque note à une valeur minimale (1/100 de temps) pour limiter les recouvrement de notes.
- Sélectionner la note la plus aiguë lorsque plusieurs notes sont jouées simultanément (accords).

La PJD est ainsi un complément de la PCD qui fournissant une information sur le désordre mélodique du fragment.



Ces figures illustrent le cas de trois suites de notes pour lesquelles le descripteur PCD serait identique, mais qui peuvent être distinguées par le descripteur PJD.

⁶ Un intervalle *chromatique* est un écart d'un demi-ton entre deux notes.

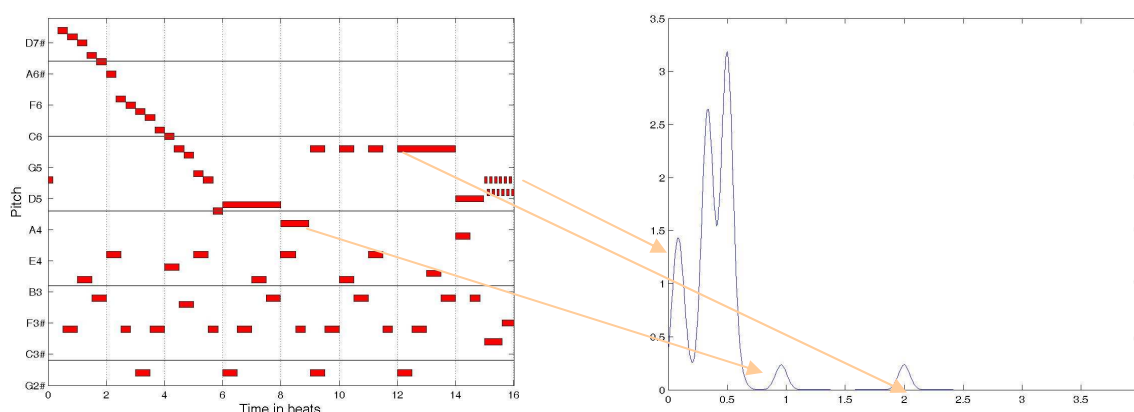
⁷ Des travaux ultérieurs permettraient d'approfondir ce point pour sélectionner la note la plus appropriée dans un accord, ou pour définir l'intervalle entre une note et un accord sur la base de la moyenne des distances. Plusieurs variantes sont possibles.

DCD : descripteur rythmique

La *Duration Class Distribution* mesure la répartition des durées de note du fragment étudié. Pour cet indicateur, deux paramètres doivent être fixés : la borne temporelle T (limite la durée des notes dépassant cette valeur), et la résolution R (durée rythmique minimale considérée).

La DCD est déterminée à l'aide d'un algorithme d'estimation de densité (Parzen⁸, que nous détaillons ultérieurement). Celui-ci permet d'entraîner un modèle à partir d'un ensemble de vecteurs, en l'occurrence des points dont la seule dimension représente la durée d'une note. Le modèle est ensuite utilisé pour obtenir la densité des points d'une grille de référence.

La DCD se distingue des propositions de [MDT] car elle prend en compte des durées courtes et peu fréquentes. En effet, la distribution rythmique de [MDT] est limitée aux durées de base : double-croche ($R=0.25$), croche ($R=0.5$), etc, alors que les morceaux que nous étudions dans nos tests contiennent des notes ayant des durées beaucoup plus complexes et inhabituelles ($3/22$, etc).



Extrait d'un nocturne de Chopin (opus 9 n°1)

Estimation de la distribution des durées de notes

L'exemple présenté ci-dessus a été obtenu avec les paramètres $T=4$ (seuil de durée) et $R=0.01$ (résolution rythmique). A titre d'exemple, mentionnons les valeurs rythmiques suivantes :

- noire 1 temps
- croche $\frac{1}{2}$ temps
- double-croche $\frac{1}{4}$ temps
- triple-croche $\frac{1}{8}$ temps

En pratique, les paramètres R et T proposés ici pour la visualisation ne sont pas envisageables car ils génèrent des vecteurs de dimension T/R (soit 400) et rendent les calculs trop lourds (et imprécis).

Afin de réduire le nombre de dimensions, tout en conservant une finesse pour les courtes durées, nous avons opté pour une échelle à pas variable : $R = 1/2$ entre 3 et 4 temps, $R = 1/4$ entre 2 et 3 temps, $R = 1/8$ entre 1 et 2 temps, $R = 1/16$ entre 0 et 1 temps. Ceci a permis d'avoir une précision suffisante sur les notes courtes, tout en réduisant la dimension des vecteurs à 31.

NN : descripteur de 'densité'

Le vecteur *Number of Notes* est doté d'une seule dimension mesurant le nombre de notes du fragment étudié. Il complète les descripteurs précédents en fournissant une information sur la densité du message musical.

⁸ Le paramètre sigma de l'estimateur était $R/2$

II.2. Similarité

Entropy kernel

Nous étudions la similarité (i.e. distance) entre morceaux selon chaque descripteur pris *indépendamment*. Nous considérons, pour chaque descripteur, que chaque morceau est un nuage de points (fragments) dans un espace de dimension égale à celle du descripteur analysé.

Basé sur les travaux de [CUT], nous définissons la similarité S entre deux ensembles de vecteur à partir de la distance de Jensen. Chaque ensemble de vecteurs V_i (résultat de l'application d'un descripteur sur les fragments du morceau i) est associé à une distribution d_i qui permettra de calculer la distance de Jensen :

$$J = h(d_1 \cdot d_2) - (h(d_1) + h(d_2)) / 2$$

$h(d_1)$ est l'entropie de la distribution des vecteurs de V_1 , $h(d_1 \cdot d_2)$ est l'entropie de la distribution du regroupement V_1 et V_2 . J mesure ainsi l'entropie relative des ensembles concaténés et permet de définir la similarité S :

$$S = e^{-\tau J}$$

τ est une variable d'ajustement permettant de calibrer la mesure (plus τ est grande, plus la mesure est sensible aux différences).

d_i , la distribution des vecteurs de V_i est obtenue en utilisant un estimateur de Parzen à noyaux, en considérant qu'un vecteur v_f (résultat de l'application d'un descripteur sur le fragment f) est un point de l'espace :

- $[0,1]^{12}$ pour les fragments PCD
- $[0,1]^{25}$ pour les fragments PJD
- $[0,1]^{T/R}$ pour les fragments DCD
- $[0,1]^1$ pour les fragments NN

Parzen : Estimation de densité

L'algorithme de Parzen [PAR] permet d'estimer la densité des valeurs prises par une variable, à partir de valeurs relevées empiriquement. Il permet donc d'estimer la distribution sous-jacente aux données en convoluant la distribution empirique avec un noyau de la forme :

$$k(x_i, x_j) = e^{-|x_i - x_j|^2 / 2s^2}$$

La densité d'un vecteur v est ensuite obtenu en déterminant la moyenne empirique suivante :

$$P(v) = 1/(N \cdot C) \cdot \sum (e^{-|v_i - v|^2 / 2s^2})$$

$$C = (\sqrt{2\pi}) \cdot s^D$$

C est une constante de normalisation qui assure que l'intégrale de $e^{-|x|^2 / 2s^2}$ sur \mathbb{R} vaut 1.

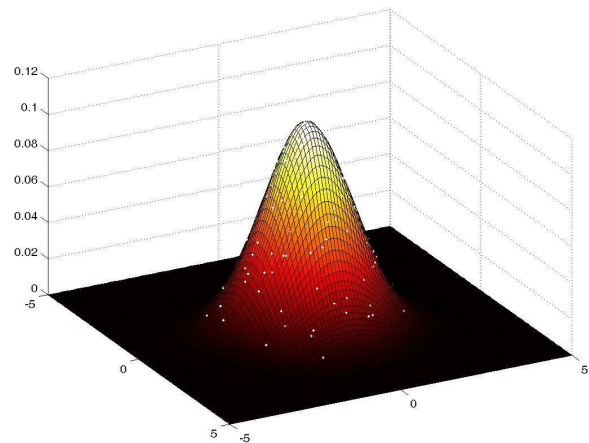
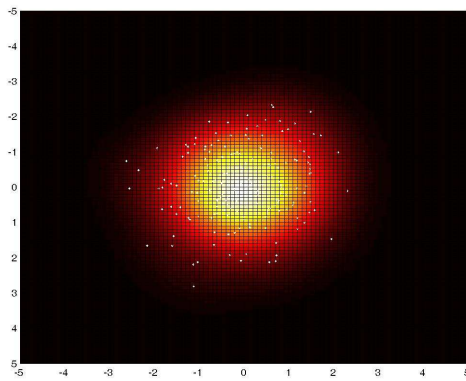
N est le nombre de vecteurs de référence (apprentissage)

D est le nombre de dimensions

L'algorithme considère une liste de N vecteurs de dimension D comme ensemble d'apprentissage. La complexité de la procédure d'entraînement est donc une constante négligeable limitée au temps de stockage des données.

La complexité de la procédure de test du modèle pour M points à tester est de $O(N \cdot M)$. Cette complexité peut devenir lourde sur de grosse liste de vecteurs dans des espaces à plusieurs dimensions.

Le calcul de la densité des vecteurs DCD est ainsi coûteuse, et une version ultérieure utilisera une procédure d'apprentissage optimisée ([PAR]), basée sur une étape de clustering des vecteurs de référence lors de l'apprentissage. Cette amélioration permet de réduire N et ainsi la complexité de la procédure de test.



Distribution de points générés selon une loi gaussienne dans un espace à deux dimensions, et modélisée par l'algorithme de Parzen (méthode classique).

Les points générés sont en blanc, la surface représente la modélisation de la distribution à partir d'une grille d'affichage.

Implémentation de la similarité

Mesurer la distance entre deux morceaux selon un descripteur (ici DCD pour l'exemple) consiste à définir la fonction suivante :

```
function s = similarity_dcd( features1, features2 )
% Return minimum similarity between two features' vectors' list

% Parzen estimations
params.trainer      = 'classic';
params.kernel.width = 10; % in n d space space

p1      = ml_parzen_train( features1, params );
p2      = ml_parzen_train( features2, params );
d1      = ml_parzen_apply( p1, features1 );
d2      = ml_parzen_apply( p2, features2 );
dmixed  = ( ml_parzen_apply( p1, [features1; features2] ) ...
            + ml_parzen_apply( p2, [features1; features2] ) ) ...
            / 2;

% Compute similarity
s = similarity( d1+(1e-120), d2+(1e-120), dmixed+(1e-120) );
```

features1 et features2 sont des listes de vecteur de dimension D
 params.kernel.width est la largeur du noyau utilisé par l'algorithme Parzen

La fonction similarity est implémentée ainsi :

```
function s = similarity( dist1, dist2, distmixed )
% Return the similarity between two distributions
t = 100; % base value : 1 | increase gaps : 1e+3
J = h(distmixed) - ( h(dist1) + h(dist2) ) / 2; % Jensen distance
s = exp( -t*J );

function entropy = h( densities )
% Return entropy of a distribution
entropy = - sum( log( densities ) ) / length(densities);
```

t est un paramètre qui accroît la sensibilité de la similarité en augmentant.

Notons enfin que la mesure de la similarité doit rester **invariante** selon plusieurs critères :

- Etre nulle pour un morceau et sa version transposée, quelle que soit sa transposition :
 - ce critère n'est à prendre en compte que pour le descripteur PCD (PJD est une mesure mélodique relative basée sur les intervalles entre notes, et DCD et NN sont, par définition des descripteurs insensibles à la tonalité).
 - Pour respecter ce critère, nous avons transposé chaque fragment de chaque morceau dans les 11 autres tonalités et avons ainsi calculé le descripteur PCD à partir d'une liste 11 fois plus grande de vecteurs.
- Etre nulle pour un morceau et sa version décalée ou ralentie dans le temps:
 - tout silence précédant le morceau est supprimé.
 - le format MIDI définit la durée et l'instant de déclenchement par rapport au *temps musical*, qui est indépendant du *tempo* (i.e. la vitesse d'exécution).
- Etre positive et symétrique :
 - ce critère est assuré par la définition du noyau entropique de [CUT] basé sur une fonction exponentielle.

III. Résultats

Les résultats présentés ici montrent l'inefficacité des descripteurs, lorsqu'ils sont *pris indépendamment* pour tenter d'identifier les auteurs (une combinaison des descripteurs serait certainement plus efficace, mais n'a pas pu être testée).

Le projet prévoyant de reconnaître des compositeurs de piano classique, les tests ont tout d'abord été fait sur des morceaux de Chopin et Mozart (téléchargés sur Internet). Face à l'échec dans le regroupement, ils ont été élargi de façon à comparer Chopin et Michael Jackson⁹ sans succès supplémentaire.

Nous limiterons donc la présentation *graphique* des résultats à la comparaison de 14 morceaux de Chopin à 14 morceaux de Mozart.

Paramètres des expériences

win_offset = 0.5 (nombre de temps séparant deux début de fragments)

win_size = 4 (nombre de temps d'un fragment)

t = 1000 (sensibilité de la similarité pour PCD)

t = 100 (sensibilité de la similarité pour PJD)

t = 1 (sensibilité de la similarité pour NN)

t = 1 (sensibilité de la similarité pour DCD)

Lecture des graphiques

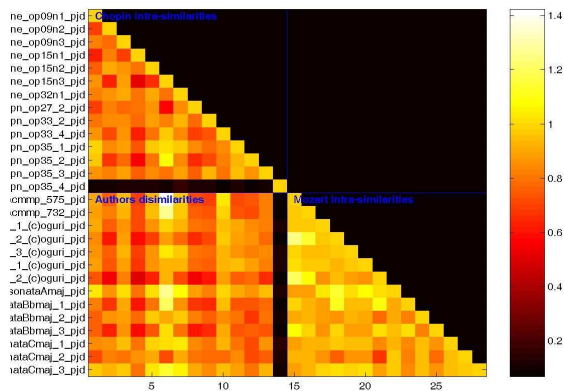
Les graphiques suivants représentent la matrice des distances entre morceaux selon chaque descripteur. La mesure de similarité étant symétrique, seule la moitié de la matrice a été calculée. La diagonale a été calculée pour déterminer la similarité maximale et normaliser les résultats.

Le carré supérieur gauche représente les distances entre des morceaux de Chopin. Le carré inférieur droit représente les distances entre les morceaux de Mozart. Le carré inférieur gauche représente la distance entre les morceaux de Chopin et les morceaux de Mozart.

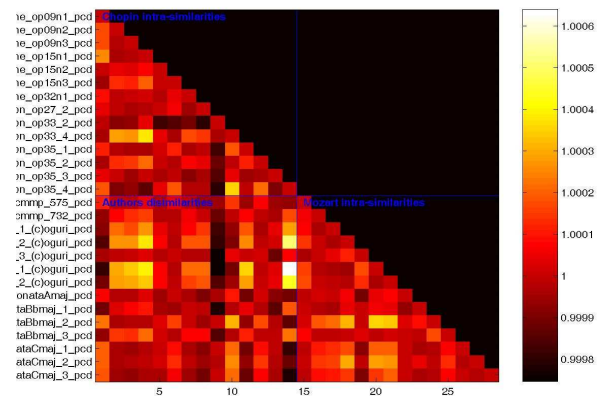
L'échelle de couleur indique le degré de similarité, une valeur proche de zéro indiquant une grande différence.

⁹ Dans les musiques d'orchestre, seuls les instruments mélodiques sont conservés, les batteries et percussions sont retirées.

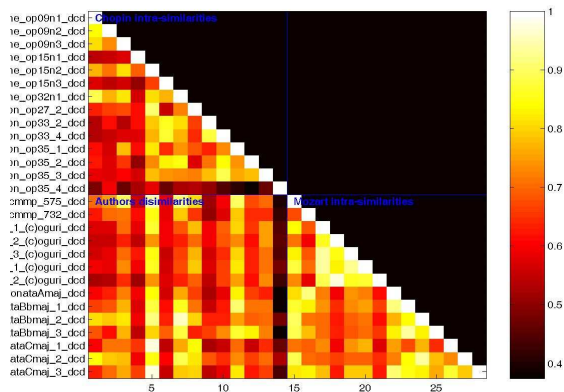
PJD similarity matrix



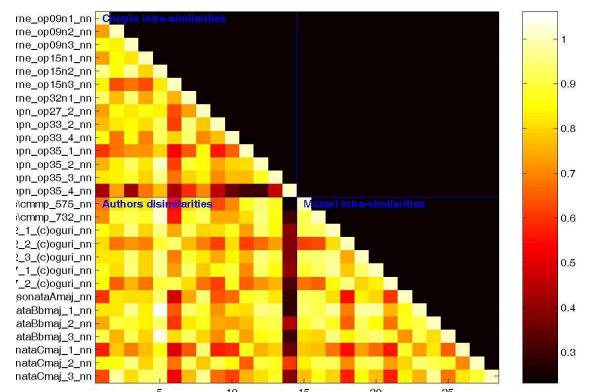
PCD similarity matrix



DCD similarity matrix



NN similarity matrix



Observations

L'échec de l'expérience se visualise en lisant des similarités trop faibles entre des morceaux d'un même auteur, et des similarités trop fortes entre des morceaux d'auteurs différents.

On note que la similarité sur le descripteur PJD doit être calibrée de façon à être plus sensible et avoir des distances moins homogènes.

Par ailleurs, la similarité maximale pour ce descripteur et pour NN n'est pas sur la diagonale. Les résultats numériques montrent que pour 1 à 5 comparaisons, la similarité est plus grande entre 2 morceaux différents qu'entre un morceau et lui-même. Ce paradoxe n'est pas élucidé à ce jour.

Enfin, le paramètre $t(=1)$ pour PCD ne permet pas de déterminer la similarité correctement. Il faut donc refaire le calcul en rendant la fonction plus sensible.

On note que la stratégie employée pour mesurer la similarité, souffre d'un gros manque de précision au regard du nombre de dimensions des vecteurs, et du nombre de vecteurs par morceau (entre 200 et 2000). En effet, une estimation de densité par Parzen nécessite un minimum de 50 exemples en dimension 1 (cette approximation reste bien entendue sujette au processus estimé). En dimension D , ce nombre d'exemples requis serait ainsi de 50^D , alors que nous ne disposons que de listes de 2000 vecteurs maximum.

IV. Directions envisageables

Les résultats précédents montrent qu'il n'est pas possible d'identifier un auteur à partir d'un descripteur *seul*. La simple concaténation des 4 descripteurs n'est pas envisageable car le nombre de vecteurs générés varie selon le descripteur (PCD implique de multiplier le nombre de vecteurs par 12 pour rester invariant selon la tonalité). Une première poursuite consisterait à prendre en compte les 4 descripteurs réunis et obtenir ainsi une distance « en 4 dimension » au lieu d'une.

De plus, la mesure de similarité entre des listes de vecteurs n'est pas adaptée à la dimension de ces vecteurs au regard du nombre de vecteurs contenu dans chaque liste. Une nouvelle mesure serait donc à envisager afin de ne pas souffrir d'un trop gros manque de précision.

Par ailleurs, il aurait été intéressant de tester l'application des descripteurs sur les morceaux complets plutôt que par fragment. En effet, outre le gain de temps de calcul, le descripteur rythmique DCD devrait révéler quelque-chose de plus net concernant des comparaisons entre Chopin, Mozart et Bach étant donné les intuitions que l'on peut avoir à l'oreille.

Bach a en effet une rythmique très ordonnée et régulière, parfois qualifiée de *mathématique*, tandis que Chopin écrit d'une façon beaucoup plus irrégulière et chaotique¹⁰. On constate aussi que Mozart tend à créer des sous-séquences courtes qui se répètent, alors que Chopin favorisera de longues évolutions harmoniques.

La description rythmique devrait par ailleurs être approfondie par l'ajout du descripteur DJD (Duration Jump Distribution) qui, à l'image du PJD par rapport à PCD, permettrait de mesurer les intervalles de durées entre notes successives.

De plus, une lecture plus approfondie de [MDT] pourrait permettre de proposer de nouveaux descripteurs prenant en compte des notions telles que la *complexité mélodique*, l'*auto-similarité d'un contour mélodique* au sein d'un morceau (voir aussi [REP]), ou encore la structure des mouvements principaux par *segmentation de mélodie*.

Avec une liste élargie de descripteurs ([FMC]), il deviendra intéressant d'étudier l'influence mutuelle pour déterminer les descripteurs les plus pertinents en fonction des auteurs.

Une autre possibilité, à cheval entre l'approche par apprentissage des successions d'accords de [BUZ] et l'utilisation du descripteur PCD, consisterait à considérer ce vecteur (dans $[0,1]^{12}$) comme un état harmonique différent du référentiel discret habituellement utilisé dans l'écriture musicale (Cmin, F, Bbmin, G#, etc). La question serait alors de savoir exprimer une probabilité de transition entre des états non discret (Markov Chain ? Kohonen ?).

Conclusion

Après des lectures variées, différentes pistes ont été étudiées pour tenter de reconnaître automatiquement l'auteur d'un morceau à partir d'un ensemble d'apprentissage constitué de partitions MIDI.

La modélisation des séquences harmonique s'est avérée compliquée en l'absence d'intervention humaine préalable, car la reconnaissance automatique des accords à partir de la partition n'est pas encore maîtrisée (les méthodes existantes n'ont pas été exhaustivement testées, [HAR] permettrait par exemple de creuser la question).

Une approche non-temporelle a donc été choisie, et des descripteurs dérivés des travaux de [MDT] ont été proposés dans des versions plus adaptées à la polyphonie.

Les travaux de [KOL] sur la classification de styles de musique ainsi que les travaux de [CUT] sur la classification d'image montrent que la mesure de la complexité / du désordre est un critère de discrimination efficace pour des objets complexes.

Nous nous sommes donc inspiré de ces idées pour définir une mesure de similarité basée sur l'entropie de la distribution des descripteurs. Néanmoins, cette approche est convenable lorsque les listes comparées sont suffisamment grandes pour obtenir une bonne précision dans un espace ayant de nombreuses dimensions (entre 12 et 31 ici).

¹⁰ Ce type de jeu est appelé *rubato*

Notre approche reste infructueuse à ce jour mais le travail peut être approfondi. Tout d'abord il conviendrait de considérer tous les descripteurs *réunis* et non les descripteurs pris *indépendamment*. En effet, ceux-ci ont été conçu pour décrire des notions complémentaires : caractérisation de l'harmonie d'un fragment, de la mélodie, de la rythmique et enfin de la densité musicale. Par ailleurs nous n'avons défini que 4 descripteurs, alors que [MDT] et d'autres suggèrent d'autres critères. Néanmoins ces premiers pas ouvrent une voie pour une réflexion à poursuivre, notamment sur le choix d'une mesure de similarité adaptée à des listes de vecteurs courtes dans des espaces à grande dimension.

Remerciements

Je tiens à remercier chaleureusement Régis Vert pour l'intérêt qu'il a porté à ce projet, les conseils judicieux qu'il a pu me donner pour m'aider à avancer, et le temps qu'il m'a accordé pour m'expliquer des notions d'apprentissage statistique.

Je souhaite remercier mon manager Jérôme Kodjabachian de m'avoir donné l'opportunité de suivre cette formation.

Enfin je remercie Robert Azencott pour ses conseils concernant la mise en œuvre de la stratégie d'apprentissage proposée ici.

Je tiens par ailleurs à signaler que le travail technique sur ce projet a été largement facilité par l'utilisation de deux outils : *MLToolkit Matlab Toolbox* (MASA) pour l'utilisation d'algorithmes Machine Learning¹¹, et *MidiToolbox* pour le traitement de fichiers MIDI.

¹¹ SVM et Parzen dans ce projet

Bibliographie*Liste d'articles étudiés (partiellement) pour ce projet*

[BUZ] Buzzanca & Fernandez, R. (2005). A probabilistic approach to the composition of baroque chorales.

[MDT] Eerola, T. & Toivainen, P. (2004). MIDI Toolbox : Matlab Tools for Music Research.

[CUT] Cuturi, M. & Vert, J.P. (2004?). Semigroup Kernels on Finite Sets.

[PAR] Babich, G.A. & Camps, O.I. (1996). Weighted Parzen Windows for Pattern Classification.

[KOL] Li, M. & Sleep R. (?). Melody classification using a similarity metric based on Kolmogorov complexity.

Liste d'articles à lire pour poursuivre le projet

[NEU] Harford, S. (?) Automatic segmentation, learning and retrieval of melodies using a self-organizing neural network

[FMC] McKinney M.F., Breebaart, J. (2003). Features for Audio and Music Classification

[HAR] Raphael, C. & Stoddard, J. (2003). Harmonic Analysis with probabilistic graphical models

[REP] Lin, C.R., Liu N.H., Wu, Y.H., Chen, A. (?). Music classification using significant repeating patterns.

[VIW] Pampalk E., Dixon, S. Widmer, G. (2003). Exploring music collections by browsing different views.

[TSW] Downie, J.S. (2003). Toward the scientific evaluation of music information retrieval systems.

[EAP] Liu, N.H., L., Wu, Y.H. & Chen A. (?) An efficient approach to extracting approximate repeating patterns in music databases.