# Belief Propagation algorithms for Automatic Chord Estimation

*End-of-course internship ENSEA (SYM)*

Vincent MARTIN [1]

March 1$^{st}$, 2018 - July 31$^{st}$, 2018

*Supervised by:* Sylvain Reynal (ETIS - UMR 8051) and Hélene Papadopoulos (L2S - UMR 8506)

*Accepted :* September 17$^{th}$, 2018

[1] vincent.martin@mail.com

**Remerciements**

Je remercie M.Matthias Quoy et l'ensemble des équipes d'ETIS pour m'avoir permis de réaliser ce Projet de Fin d'Études et pour leur accueil chaleureux.

Je remercie mes deux encadrants de choc, Hélène Camille Crayencour et Sylvain Reynal pour leur excellence académique et leur ouverture d'esprit permettant à des projets aussi fous de voir le jour. Plus spécifiquement, je remercie Hélène pour m'avoir encadré sur les aspects MIR de ce stage et pour ses questions et remarques qui ont toujours été très enrichissantes. Je remercie Sylvain pour avoir partagé avec moi son enthousiasme concernant ce projet ainsi que pour la grande confiance et l'immense liberté scientifique qu'il m'a accordé.

Ce projet m'a amené à rencontrer d'autres chercheurs en MIR qui m'ont soutenu et que je voudrais remercier : Dogac Basaran (Telecom Paris) for his relevant questions and his enthousiasm, Meinard Mueller for his time and the reflexions that we shared around the PhD Thesis, and Magdalena Fuentes Lujambio for her help with *madmom* and her interest in the project.

Je tiens particulièrement à remercier mes amis (Stan, Maxime, Hugo, Quentin, Karél, Ju' et tous les autres) et ma famille pour leur soutien. Remerciements particuliers à Luca pour sa relecture attentive de ce document et sa bonne humeur communicative.

***Abstract :*** This work aims at making the coexistence of two completely different fields : telecomunications and Music Information Retrieval. If the researcher of MIR already use algorithms borrowed from voice signal processing, text recognition or image processing, no work using telecommunications algorithms have been produced. The Belief Propagation Algorithm, used in telecomunications for LDPC decoding, shows great abilities to process the inference in Automatic Chord Estimation and replace the Hidden Markov Models.

**Keywords :** Music Information Retrieval, Automatic Chord Detection, Hidden Markov Model, Belief Propagation Algorithm.

***Résumé :*** Ce projet a pour but de faire coexister deux domaines de recherche différents : les télécoms et la Music Information Retrieval. Si les chercheurs de ce dernier domaine utilisent des algorithmes empruntés au traitement de la voix, la reconnaissance de texte ou le traitement d'image, aucun travail à ce jour n'a utilisé les télécoms en MIR. L'algorithme de Propagation de Croyance, utilisé pour le décodage des LDPC en télécoms, montre de bonnes dispositions pour effectuer la phase d'inférence en détection automatique d'accords et remplacer les Chaînes de Markov Cachées.

**Mots clés :** Music Information Retrieval, Reconnaissance Automatique d'Accords, Chaîne de Markov Cachée, Algorithme de Propagation de Croyances.

# Contents

**Introduction** **5**

**1 Presentation of the context** **7**
1.1 Presentation of ETIS . . . 7
1.1.1 The different teams . . . 7
1.1.2 Seminaries . . . 7
1.2 MIR and Community . . . 8
1.2.1 Skype Calls . . . 8
1.3 Automatic Chord Estimation . . . 9
1.3.1 What is the aim of Automatic Chord Estimation ? . . . 9
1.3.2 Overview of the classical systems . . . 9
1.4 State of the Art on Deep Learning for ACE . . . 9

**2 Real world dataset** **10**
2.1 Method . . . 10
2.1.1 Evaluation of the performances . . . 10
2.1.2 Database . . . 11
2.1.3 Estimated vs ground truth information . . . 11

**3 Belief Propagation for Automatic Chord Estimation** **12**
3.1 Observation probabilities . . . 12
3.1.1 Chromas extraction . . . 12
3.1.2 Construction of the template of the chords . . . 13
3.1.3 Observation probability . . . 14
3.2 Transitions matrix . . . 14
3.3 HMM and BP . . . 14
3.3.1 Description of the Viterbi HMM . . . 14
3.3.2 Description of the BP algorithm . . . 15
3.3.3 Example . . . 16
3.3.4 HMM viewed as a BP algorithm . . . 16
3.3.5 Advantages and flaws of the Belief Propagation Algorithm . . . 17
3.3.6 Cycles and convergence criteria . . . 17
3.3.7 Sum-Product and Max-Product Algorithm . . . 18

**4 Feeding the graph** **22**
4.1 Downbeats and Structure: two ways to enhance ACE performances . . . 22
4.1.1 Taking into account the downbeats positions . . . 22
4.1.2 Taking into account the structure of the song . . . 23
4.1.3 Taking into account both structural information . . . 24
4.2 Taking into account similarities . . . 25

4.2.1 Taking into account self-similarity . . . 25
4.2.2 Taking into account similarity and anti-similarity . . . 26

**5 Noise robustness** **32**
5.1 Audio . . . 32
5.2 Features extraction . . . 32
5.3 Noise addition . . . 32
5.4 Method . . . 32
5.5 Results and discussion . . . 35

**6 Results and discussion** **36**
6.1 Results of the system and discussion . . . 36
6.2 Methodology discussion . . . 37
6.2.1 Reliability of the ground truth . . . 37
6.2.2 Methodology flaw . . . 38
6.3 Future work . . . 38
6.4 Used and learnt skills . . . 38
6.4.1 Used skills . . . 38
6.4.2 Learnt skills . . . 38

**Conclusion** **38**

**Appendices** **40**

**A Basic Music Theory for ACE** **40**
A.1 Rythm . . . 40
A.1.1 Tempo . . . 40
A.1.2 Tactus . . . 40
A.2 Harmony . . . 40
A.2.1 The 12 semi tones scale . . . 40
A.2.2 What is a chord ? . . . 40
A.2.3 Notation with indexes . . . 41

**B Chromas and common audio signal processing tools in MIR** **42**
B.1 FFT and MFCCs . . . 42
B.2 Chromas vectors . . . 43

**C Basic Graphs Theory** **45**
C.1 Definition of a Graph . . . 45
C.2 Properties of the graphs . . . 45
C.2.1 Directed graphs . . . 45
C.2.2 Adjacency matrix . . . 45
C.2.3 Tanner graphs . . . 46
C.3 Full connected graph . . . 46

**D State of the Art on Deep Learning for Chord Detection** **47**

**Abbreviations used in this document**

**ACE**: Automatic Chord Estimation
**BP**: Belief Propagation Algorithm
**CRF**: Conditional Random Field
**DBN**: Deep Bayesian Network
**HMM**: Hidden Markov Model
**MIR**: Music Information Retrieval
**MLN**: Markov Logic Network
**RNN**: Recurrent Neural Network

# Introduction

Music Information Retrieval is a recent field of Audio Signal Processing that aims at estimating musical information from audio files. From genre and style estimation to audio-to-score alignment systems, a high diversity of tasks exists. Taking advantage of the diversity of its researchers, the MIR uses algorithms from Statistics, Algebra, Physics, Music Theory, Psycho-Acoustics, Acoustics, and more recently Deep Learning: almost any signal processing field is represented in MIR.

Tools developed thanks to MIR are used by musicians (audio to score estimation, composition helper, rhythm tracker, ...) but also by video signal scientists, customized music providers or home automation engineers. Sound is used in numerous domains and progresses made in MIR serve all of them.

Automatic Chord Estimation from an audio file is among the oldest tasks of MIR([35]). This task aims at estimating the chords of a song from an audio file. Firstly relying on signed features features, it has evolved little by little, taking into account language models and more recently using deep learning. But even with these progresses, the performances of the proposed approaches stagnate, differing only slightly from one another.

In addition, in [45], a lack of cutting-edge inter-disciplinary approaches is shown: with the exception of the use of deep learning, few approaches bring real novelty in MIR research. Moreover, the usual fields crossing over with musicology are almost always the same: speech processing, text retrieval and vision.

This work has the following objectives. Firstly, attempting to break the glass ceiling of Automatic Chord Recognition with a new system. Secondly, bringing the advantages of telecommunication algorithms to MIR and showing the great concordance between the two fields.

We highly recommend to the reader to read the Appendix A to have the basis of the simple musical theory and the Appendix C to understand the basis of the graph theory that will be used later in this document. .

# Part 1

# Presentation of the context

## 1.1 Presentation of ETIS

ETIS is a Mixed Research Unit headed by Mathias Quoy and Emanuelle Bourdel. The lab is located in two sites: at the ENSEA, Cergy (where this internship has be done) and in the University of Cergy-Pontoise, Pontoise. ETIS is under the supervision of the Computer Sciences and their Interactions Institute of the CNRS (Institut des Sciences Informatiques et leurs Interactions (INS2I)). The lab employs 120 people, distribued between 50 researchers, approximately 50 PhD students and 8 administrative and technical staff.

### 1.1.1 The different teams

The lab is divided in four teams:

- Indexation Multimédia et Intégration de Données (MIDI) :this team works around two research axis. The first is about Big Data, i.e. the integration of Web data at a big scale, and the search of patterns in data storage centre and in the social networks' graphs.

  The second axis is about multimedia systems, and the extraction of visual content from multimedia files such as images, videos and 3D objects. It is also about indexation of databases and statistical learning for the searching in these bases.

- Information, Communications, Imagerie (ICI) :this team works about numerical communication and imagery. The first division of the team conceive and analysis systems of numerical communication using optimisation, signal processing and error correcting codes tools. The second part of the team is dedicated to image processing and analysis, mainly with medical applications.

- Architecture, Systèmes, Technologies pour les unités Reconfigurables Embarquées (ASTRE) :this team works on reconfigurationality of systems on heterogeneous chip with the originality to explore their adaptability on different scales, circuit, system and software, for numerous embed applications (video, telecommunications, embed system for health).

- Neurocybernetique :this team propose to model the neural mechanisms and the cerebral structures that are important in the infantile development in order to conceive autonomous robots that can learn things by themselves, perceive the outside world, and interact with it.

### 1.1.2 Seminaries

Working in a big laboratory such as ETIS allowed me to attend three seminaries of the lab. More than just technical presentations, these seminaries offer the opportunity to discuss subjects not necessarily connected to the specialities of all the members of the lab, and to promote inter-disciplinarity.

The first one, leaded by Nicolas Priniotakis (University of Cergy-Pontoise, France), was about facial reconstruction. In this presentation, he emphasised the benefits brought by modern technology to his field, while recognising some aspects of the work lack scientific rigour. In his opinion, a collaboration between his team and some member of ETIS could lead to the development of more rigorous methods and more effective models that could help in the difficult task of estimating a face from a skull.

The second one, presented by Yann Soullard (LITIS, Rouen, France), was divided in two parts. The first one was about credal classification with prudent Hidden Markov Models. Even if it was in the field of technical gesture recognition with video, this work was very inspiring in the way that like the BP, it is another way to decode processes from observations and transitions information (see 3.3 for more information about the HMM and the BP). The second part was about a new type of Deep Learning architecture: deep neural networks based on dilated convolutions. The main advantage of this architecture is that it allows using convolutional neural networks (CNN) without losing resolution as it is the case in classical CNN. In this case too, even if the original task (line identification in a text) is not at first glance related to MIR, CNN are used on a non-negligible part of MIR systems and could benefit from these methods.

The third one, presented by Jean-Luc Gaudiot (University of California, Irvine), was about an overview of Autonomous Vehicle Systems. This very interesting talk showed the different tasks and interests in autonomous vehicles research, from captors to decision making. Unfortunately, no topic developed in this talk could have any application to our project and MIR more generally.

## 1.2 MIR and Community

The MIR community is a recent and small community gathering researchers from audio signal processing and musicology but also psychology, mathematics or speach recognition, making it a highly diversified community with multi-skilled people. The present work at ETIS shows that this community can benefit from all the fields more or less linked to signal processing, even telecommunications ! One of the special features of MIR is that, contrary to most signal processing fields, most researchers are musician and love the data they manipulate (subject discussed with Meinard Mueller in a Skype call, see below).

### 1.2.1 Skype Calls

This internship allowed me to meet MIR people and to exchange about my work with specialists.

Firstly, I had the great opportunity to talk with Meinard Mueller, Semantic Audio Processing Professor and notorious MIR researcher from AulioLabs (Erlangen, Germany). The discussion was mainly about the MIR community and all the factors around a PhD Thesis, which is my project after this work. It was a very inspiring discussion that made me realise that getting a PhD Thesis is something more complicated than just a financial and team research issue.

Secondly, following an exchange about how to estimate downbeats with the Python library *madmom* (see subsection 2.1.3), I had the opportunity to speak with Magdalena FUENTES LUJAMBIO, a former PhD student of Hélène, currently working in rythm estimation. During this conversation, I explained the Belief Propagation algorithm and its benefits over the HMM, and faced for the first time the difficulties of explaining to full fledged MIR researcher a telecommunications algorithm that is totally new to the community. This illustrates pretty well the fact that MIR people face difficulties to accept approaches that differ too much from the over-used classical ones. In the end, Belief Propagation will be used instead of Conditional Random Fields in her current work.

Research is a working environment that fits me well: the possibility to study interesting topics at a high level and to encounter top level in their fields researchers is very enriching.

## 1.3 Automatic Chord Estimation

### 1.3.1 What is the aim of Automatic Chord Estimation ?

Automatic Chord Estimation is one of the oldest tasks of MIR ([35]). Its aim is to estimate,with the help of different information sources, the chords played in audio files. This is a sub-task of multiple MIR tasks, like automatic score alignment or lead sheet transcription([30, 12]). It also has applications in cover song detection or song mood estimation ([12]).

### 1.3.2 Overview of the classical systems

**Steps of ACE**

The classical approaches of ACE have the following steps [7] :

1. Feature extraction. Hand-designed or more recent deep learning systems are used to extract features from the sprectrum. In a large majority, these features are *chroma* vectors (see SectionB.2).

2. Pre-filtering. Some works, when the features are computed, apply a pre-filtering step aiming at helping the pattern matching step. In our case, this step is explained in Section 3.1.

3. Pattern Matching. This step allow to introduce temporal dependencies such as transition probabilities or structural information. Chords being temporally dependent in a song, it is essential to include it in the model.

4. Post-Filtering. Depending what is the result of the previous step, a post-processing step can be needed. In our case, this step will only be a max-likelihood filtering.

Until then, the pattern matching step was done either with simple templates matching or temporal models such as Hidden Markov Models.
Our work aims at surpassing the performances of the HMM during the pattern-matching step using the BP, taking into account the structure of the song (see Section 3.3). An overview of our system is proposed fig 3.1.

## 1.4 State of the Art on Deep Learning for ACE

To compare our algorithm to the state of the art performances, it was been necessary to search for these performances. We observed that no overview of the state of the art in deep learning for ACE exists so to be as accurate as possible in our comparisons, we did this overview. A copy of this document is in Appendix D.

# Part 2

# Real world dataset

The algorithms presented in the following will be tested following the procedure and on the dataset described in this part.

## 2.1 Method

### 2.1.1 Evaluation of the performances

To evaluate the performances of a retrieval task, different method exist. Most of them use annotated data base, specifically designed to test a specific task. Automatic Chord Estimation is a task that can be assimilated to classification. It requires to choose a criteria to measure if the method has well reproduced the ground truth, but also the different classes on which the different observation can be classified (the dictionary). We invite the reader to read the paper of Bob L. Sturm [52] for more information on the formalisation of Music Information Retrieval.

#### Dictionary

As in most studies about chord estimation ([35]), we will limit ourselves to the 24 major and minor chords. As described in [44], the chords of the ground truth that are not in this dictionary are associated to major or minor chords, so that the recall can be computed.

#### Weighted Chord Symbol Recall

We evaluate the performances of the system with the python library mir_eval. [44]. The performance is measured by the Weighted Chord Symbol Recall (WCSR) defined in [42] by:

$$SR = \frac{\sum\limits_{segment_i} s_i o_i}{\sum\limits_{segment_i} s_i} \tag{2.1}$$

$$WCSR = \frac{\sum\limits_{song_i} l_i SR_i}{\sum\limits_{song_i} l_i} \tag{2.2}$$

with:

- $s_i$: length of the segment $i$
- $l_i$: length of the song $i$

- $o_i$: 1 if the chord estimated on the segment $s_i$ is equal to the ground truth, else 0.

### 2.1.2 Database

The different inference algorithms are tested on the Beatles subset of the Isophonics data set. Following [39], some songs are not considered, due to the uncertainty of their structure or the errors in the ground truth provided by Isophonics. These songs are listed in the table 2.1.
We so consider 157 songs in our data set. For each song, the *wav* audio file, the annotated chords and their respective beginning and ending time are available.

| Name of the song | Reason of eviction |
|---|---|
| Get Back | Lack of downbeats file |
| Glass Onion | Lack of downbeats file |
| Revolution 9 | Lack of downbeats file |
| Lovely Rita | Bad annotations |
| Baby's In Black | Complicated metric |
| You've Got To Hide Your Love Away | Complicated metric |
| Norwegian Wood | Complicated metric |
| She's leaving Home | Complicated metric |
| Long, Long, Long | Complicated metric |
| Oh! Darling | Complicated metric |
| Dig A Pony | Complicated metric |
| Dig It | Complicated metric |
| A taste Of Honey | Complicated metric |
| Lucy In The Sky With Diamonds | Complicated metric |
| Being For The Benefit Of Mr. Kite | Complicated metric |
| Strawberry Fields Forever | Complicated metric |
| All You Need Is Love | Complicated metric |
| Happiness Is A Warm Guy | Complicated metric |
| I Want You (She's So Heavy) | Complicated metric |
| Two Of Us | Complicated metric |
| I Me Mine | Complicated metric |

Table 2.1: Song removed from the data base

### 2.1.3 Estimated vs ground truth information

To test the robustness of our system to variations in the beats and the downbeats, we chose to compute the performances of the system with ground truth beats and downbeats but also with estimated beats and downbeats.
This estimation is made with the state of the art Python library *madmom* ([5]). The algorithms contained in this library use Recursive Neural Networks and Deep Bayesian Network.
The only drawback of this library is that the downbeats can be estimated only on some rhythmic signatures (only rhythmic signatures that are over 4), that are a parameter of the system. The 3/4 and 4/4 signatures seem to work well on our database, but the use of this library for more "exotic" musical content is prohibited (for example, irish traditional music contains a lot of *jig* in 6/8).

## Part 3

# Belief Propagation for Automatic Chord Estimation

The flowchart of our system is presented in fig 3.1. The elements in italic are the input of the system. They are either obtained from the ground truth or estimated by other systems. The different parts are explained in the following section of this document:

- *computation of the chromas*: Appendix B.2
- *computation of the observations*: Section 3.1
- *transitions matrix*: Section 3.2
- *BP algorithm, beliefs*: Section 3.3
- *graph generation*: Section 4.1

The chromas and the observations are computed with Matlab (code supplied by Hélène) and the other steps are implemented in Julia [22], a new high-level language that aims at becoming a reference in science.

The computation of the chromas and the observation allow to estimate the probability of a chord given spectral information and a model. The transition matrix allow to estimate the probability of a chord from the previous one. BP algorithm and HMM allow to combine them to compute a more exact probability of the chord. These diferent parts are explained below.

## 3.1 Observation probabilities

The observation probabilities (prior probability) from the chroma vectors are computed in the same way than in [15]. Briefly, the observation probability vector is constructed as follows.

### 3.1.1 Chromas extraction

**Working on frames vs working on tactus**

To study the best feature extraction approach is not the aim of this work. Nonetheless, an interesting question is whether it is best to work on the frame level (one state equals the state of a frame) or at the tactus[1] level (one state equals the state of all the tactus). As shown by [28], the frame level approaches

[1] see Appendix B.2

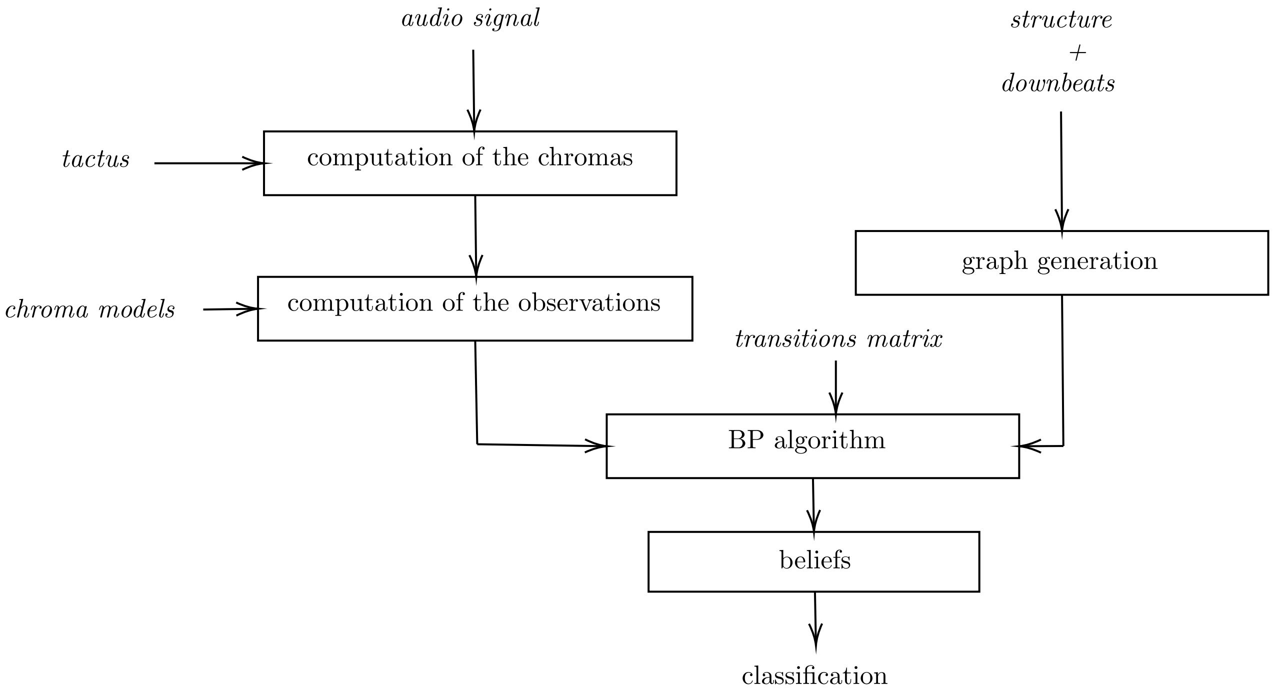


Figure 3.1: Flowchart of our system

seem to have reached their limit: we choose to work on the tatum level. First, the tatum will be computed with ground truth tactus, then they will be estimated (see Section 2.1.3)

**Chroma vectors extraction**

The chroma vectors are extracted as described in the Appendix B.2.
To compute the observation probabilities, we compute the distance between the chroma and a template, constructed as explained in the next section.

### 3.1.2 Construction of the template of the chords

- We consider first each note of the chord separately. Each one of these notes has harmonics.
- For the note N, the $h^{th}$ harmonic has an amplitude of $0.6^{h-1}$, where $h$ is the rank of the harmonic. We consider models with 5 harmonics and compute the corresponding chroma vector. This is illustrated in fig 3.2 .
- We sum, for each chord, the notes and their harmonics that are comprised in the chord. For example, for a C-Major chord, we sum the model of the C note, the E note and the G note (see Appendix A for more information on how to build a major or minor chord). Fig 3.2 shows the model for a C Major chord.

We obtain in this way the different models for the 24 chords of the dictionary, presented in fig 3.3.

### 3.1.3 Observation probability

The observation probability is then computed by the cosine distance between the observed chroma and the different models. The cosine distance is defined by:

$$d(C_i, C_j) = \frac{C_i \cdot C_j}{||C_i||_2 \cdot ||C_j||_2}$$

where $||.||_2$ is the second order norm. This way, we obtain so the observation probabilities of the different nodes. These observation probabilities are the features of our system.

## 3.2 Transitions matrix

In music, chords don't follow a random order and most of the time, the following chord of a given one is partly determined by itself.
Transitions matrices are very useful to ACE. Indeed, they are the most basic element of pattern matching: they allow to take into account information from the other chords to enhance the detection of a given one.
Multiple transitions matrices exist to process transitions between chords. The three main approaches are

1. theoretical music information approaches [1]
2. perceptual approach [29]
3. learning approach [39]

Each approach has is advantages and its flaws. The approach based on musical theory has the advantage to be "universal": all western music based on the 12 semi tones scale respects this rule. Its flaw is its lack of specificity: this matrix gives good results but others give better results on some specific musical genres or specific tasks.
The approach based on perception gives goods results by taking into account the subjectivity of the auditors but its flaw resides in that it depends too much on the auditors and their musical habits.
The approach based on learning seems the best but faces a dangerous problem: some transitions, whatever the size of the data base, are very rarely encountered in it. That is why this approach is rarely used.
As proposed in [39], we chose the perceptual transitions matrix elaborated by C.Krumhansl in [29], represented in fig 3.4.

## 3.3 HMM and BP

The main objective of this work is to work on the pattern matching step. After explaining what is the HMM and the BP, it has been necessary to translate the HMM into a particular BP, before adding structural information to take full advantages of this algorithm.

### 3.3.1 Description of the Viterbi HMM

A Hidden Markov Model is a statistical model that computes the probability vector of a hidden state taking into account the observations probabilities and transitions probabilities . It is fully described by three parameters, given that the states are represented by $S_i$:

- $\pi_i$, the initial probability that $S_i$ is the initial state
- $a_{ij}$, the transition probability between states $S_i$ and $S_j$
- $b_i(O)$, the probability to emit the observation $O$ in the state $S_i$

These parameters have the following constraints:

- $\sum_i \pi_i = 1$
- $\forall i, \sum_j a_{ij} = 1$
- $\forall i, \sum_{observationO} b_i(O) = 1$

In our case, the hidden states $S_i$ are the chords that we want to infere ($S_i \in [[1, 24]]$) , the observations are as described in Appendix B.2. $a_{ij}$ and $b_i(O)$ are given by musical studies.
The initial state $S_0$ is initialised as the column vector $(\frac{1}{N_D})_{N_D,1}$ where $N_D$ is the size of the chords dictionary. The Viterbi inference is processed as follow:

$$\forall i, S_i = \arg\max_k \{b_i(O_k) \times a_{i-1,k}\} \tag{3.1}$$

### 3.3.2 Description of the BP algorithm

The BP algorithm, as the HMM, is designed to infere hidden states given observations and transitions between hidden states([23]). The main difference is the topology of the problem: where HMMs are linear, BP can use any graph topology.

#### Difference with the HMM

One of the main differences between BP and HMM is that HMM are intricately linked to temporal dependencies. Where the BP uses any graph, HMM always infer a state with information from the previous states. One important thing here is that BP not only allows more flexibility on information that can be used during the pattern matching step, but it is also independent from time. Instead of temporal dependencies, one can think about the BP in term of spatial dependencies: given the environment of one node, what information one can obtain to enhance the estimation of it ?

#### Parameters of the BP

BP algorithm is fully described by three parameters, given that the states are represented by $y_i$:

- the adjacency matrix of the graph
- $\phi_i(x_i)$, the observations vectors
- $\psi_{i,j}(x_i, x_j)$, the constraints between node i an j

A graphical view of the Belief Propagation Algorithm is shown in fig 3.5.

#### Sum-Product BP processing

Two version of the algorithm: the Sum-Product and the the Max-Product algorithm (see Subsection 3.3.7). We present here the Sum-Product algorithm.

For all the nodes of the graph, we compute the message from one node to one of his neighbours:

$$m_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{p\in N(i),p\neq j} m_{p\to i}(x_j) \tag{3.2}$$

These messages are like a pool: every node connected to a node $i$ will transfer to it the probabilities associated to the different states possible for it.

This can be re-written with more comfortable notations for chord recognition:

$$m_{i\to j}(\text{chord } c') = \sum_{\text{chords } c} Obs_i(c)\psi_{i,j}(c,c') \prod_{p\in N(i),p\neq j} m_{p\to i}(\text{chord } c') \tag{3.3}$$

When the convergence is reached, we calculate the belief of each state for each node:

$$b_i(x_i) = \phi_i(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i) \tag{3.4}$$

In other terms:

$$b_i(c) = Obs(c) \prod_{j\in N(i)} m_{j\to i}(c) \tag{3.5}$$

We infer then the hidden states by:

$$y_i = \arg\max_k \{b_i(x_k)\} \tag{3.6}$$

Important remark: from a computational point of view, it is vital to normalise the messages after each step, to avoid too little values that reach the machine zero. The messages conveying probabilities, they are normalised so that they sum to one ([23]).

### 3.3.3 Example

The graph for this example is presented in fig 3.6.
Let's compute $m_{3\to 2}$:

$$\begin{aligned} m_{3\to 2}(x_j) &= \sum_{x_i} \phi_3(x_i)\psi_{3,2}(x_i,x_j) \prod_{p\in\{4,5\}} m_{p\to 3}(x_j) \\ &= \sum_{x_i} \phi_3(x_i)\psi_{3,2}(x_i,x_j) m_{4\to 3}(x_j) m_{5\to 3}(x_j) \end{aligned}$$

with:

$$\begin{aligned} m_{4\to 3}(x_j) &= \sum_{x_i} \phi_4(x_i)\psi_{4,3}(x_i,x_j) \\ m_{5\to 3}(x_j) &= \sum_{x_i} \phi_5(x_i)\psi_{5,3}(x_i,x_j) \end{aligned}$$

(the node at the edges of the graph receive information only from one direction )

### 3.3.4 HMM viewed as a BP algorithm

A HMM can be viewed as a very simple BP algorithm where:

- The graph is a simple-path unweighted directed graph beginning with the initial state $y_0 = S_0 = (\frac{1}{N_D})_{N_D,1}$ to the end of the song.
- The messages propagated are directly the beliefs:

$$\forall j > i \ m_{i\to j}(c) = \phi_j(c) \times \psi_{i,j}(y_{i-1}, c) = O_c \times a_{i-1,c} \tag{3.7}$$

  The most probable states are computed gradually by

$$y_j = S_j = \arg\max_k \{m_{j-1,j}(k)\} \tag{3.8}$$

### 3.3.5 Advantages and flaws of the Belief Propagation Algorithm

The main advantage of the BP algorithm is the fact that it can easily take into account long distance information, only creating edges with the adapted constraints. Few other methods take into account long distance relationship, other examples including Markov Logic Network ([41]), meaning chromas that have the same position in the same structure ( replacing all the first tatum of the chorus by their mean chroma, doing the same with the second tatum of all the chorus, etc) ([33]) or the Recursive Neural Networks ([46]). So this advantage is a great advance for inference in MIR !

This method, however, has two main drawbacks. First, it tremendously relies on the graph :if the graph has not the appropriated structure, or the constraint are adapted to the observations, the algorithm can converge to a non exact solution, or not converge at all. Secondly, if the observations are in contradiction with the information given by the links and the transitions, the algorithm can't converge. This can happen in the following case:

- The transitions and the observations are strictly incompatible. For the sake of the example, we consider three nodes i,j and k, linked only by one edge each (see fig 3.7). Let's compute $m_{j\to k}$ in the case that $\phi_j = \begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}$ and $m_{i\to j} = \begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix}$. Following the steps described in the previous section, we obtain $m_{j\to k} = \sum_{x_j} \phi_j(x_j)\psi_{j,k}(x_j, x_k) m_{i\to j} = \begin{bmatrix}0\\0\\\vdots\\0\end{bmatrix}$. This message doesn't convey any information and annihilates all the other information.
- This case is more likely to happen on short cycles. To simplify, let's consider the second graph of fig 3.7. Let's also assume that this is a binary graph: there are only two states, state 0 or state 1. If the constraint between the nodes is a hard constraint compelling them to be the contrary of the previous one, the cycle oscillates. Indeed, if $S_i = \begin{bmatrix}0\\1\end{bmatrix}$, then $S_j = \begin{bmatrix}1\\0\end{bmatrix}$ and $S_k = \begin{bmatrix}0\\1\end{bmatrix}$. But $S_i = \bar{S_k} = \begin{bmatrix}1\\0\end{bmatrix}$, $S_j = \bar{S_i} = \begin{bmatrix}0\\1\end{bmatrix}$ and so on ... The cycle oscillates and never converges.

### 3.3.6 Cycles and convergence criteria

#### Cycles

Feeding the graph can lead to cycles, that can be very short (3 nodes long cycles) or very long cycles (with long distance edges). Both have their advantages and drawbacks. Short cycles have the drawback of having convergence issues ( as explained in the Section 3.3.5) but have the advantage of converging very quickly. On the contrary, large cycles are very stable but take a lot of iterations to feedback the information and hence take time to converge.

#### Convergence Criteria

A criteria has to be defined to stop the system. We choose the following :

$$\forall (i,j) \ \max_k |m^{n+1}_{i\to j}(k) - m^{n}_{i\to j}(k)| \leq \epsilon \tag{3.9}$$

The choice of the value of epsilon is crucial: if it is too small, the precision will be better but the calculus time will be huge; on the contrary, higher values of epsilon will compute quicker but with less precision. We arbitrarily choose $\epsilon = 10^{-12}$ and a maximum number of updates of 200: beyond 200 updates, the messages are considered as non convergent. In this case, the messages are not "false": they have just not reached the desired precision.

### 3.3.7 Sum-Product and Max-Product Algorithm

Another Belief Propagation algorithm exists: the Max-Sum BP algorithm. It is defined as follow:

$$m_{i \to j}(x_j) = \max_{x_i} \phi_i(x_i) \psi_{i,j}(x_i, x_j) \prod_{p \in N(i), p \neq j} m_{p \to i}(x_j)$$

The difference stands in the fact that the Sum-Product is aimed at computing the exact marginal probability while the max-product gives better results for classification (see [9]).
As the results on noise robustness have shown that the max-product algorithm gives better results than the sum-product one, the other algorithms are designed with the max-product algorithm.

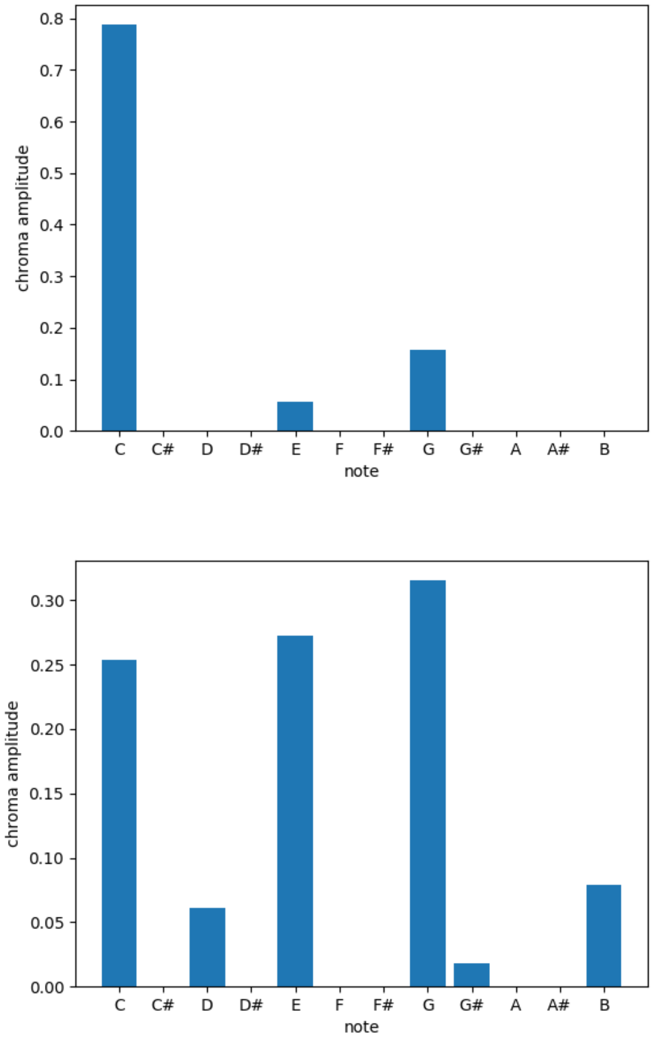


Figure 3.2: Template of a C solo note and a C Major chord chroma vector according to the model explained in Section 3.1

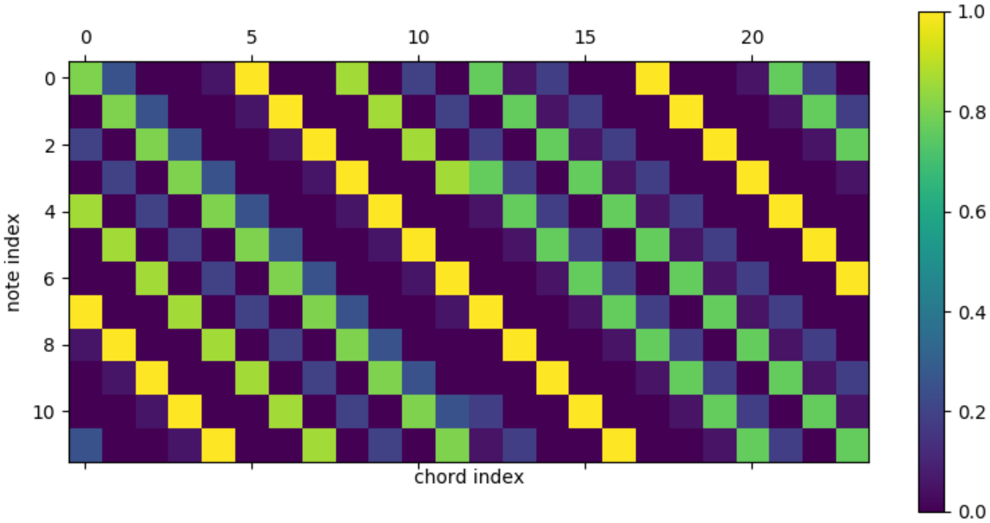


Figure 3.3: Templates of the 24 chords chroma vector according to the model. The note and chord indexes are explained in Appendix A.

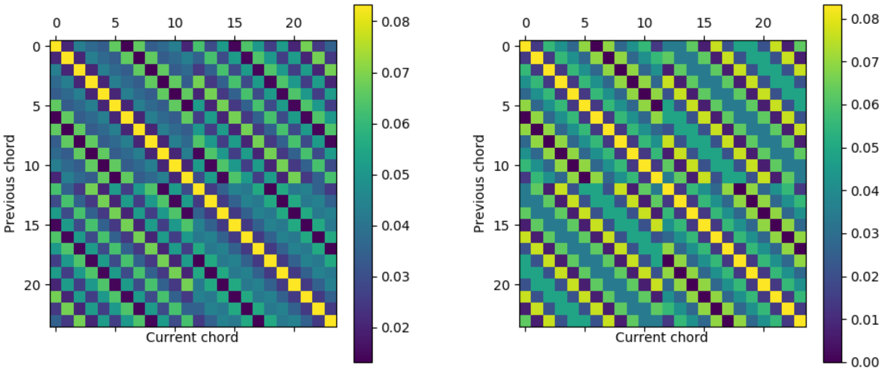


Figure 3.4: Perceptual transition matrix chosen for our study[29] and transition matrix based on the cycle of fifths [1]. Row and column are the 24 major and minor chords numerated as explained in Appendix A

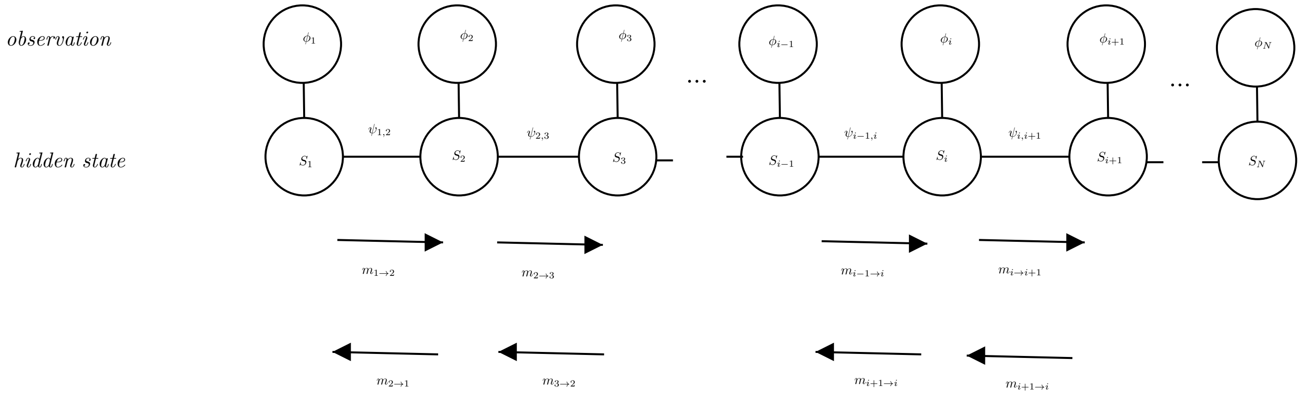


Figure 3.5: Belief Propagation and the different messages computed

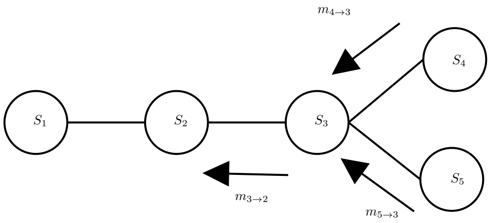


Figure 3.6: An example of graph for the Belief Propagation Algorithm

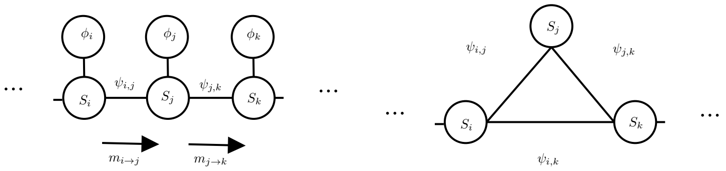


Figure 3.7: Graphs where the Belief Propagation doesn't converge or has difficulties to converge

# Part 4

# Feeding the graph

## 4.1 Downbeats and Structure: two ways to enhance ACE performances

To take full advantage of the BP algorithm, the next step is to feed the basic linear graph with structural information: the downbeats and the structure.

Indeed, as shown in [41], information contained in the structure of a song allow to enhance the performance of ACE. It is in fact something very natural: when listening to a song, even heard for the first time and not knowing the score, it is not difficult, after having identifying the structure of the song, to anticipate the next chords that will be played. The idea of using BP is to use this information, creating links between parts of the song that are very similar (first tactus of the each chorus with the first tactus of the all the other chorus, second tactus of the each chorus with the second tactus of the all the other chorus, ...).

Using the downbeats is the same idea but with at the bar scale: instead of using the information contained in the structure, we use the information contained at the scale of a bar. As the usage of the downbeats has shown encouraging results in tonality estimation in [40], we are optimistic in the usage of these information with BP in ACE.

### 4.1.1 Taking into account the downbeats positions

In a bar, the chords are note independent of each other: song that change of chord every tactus are rare and most of the time, each chord is repeated at least two time in a bar. We assume that use the flexibility given by the BP to In practical terms, we link together all the tactus of the same bar, with a probability to be identical. It assumes that the chords in the same bar are almost all identical but thanks to the flexibility of the BP, the turn-over chords in the end of the bars will not be misinterpreted.
From a graph point of view, taking into account the downbeats positions allows adding more information to each node: instead of receiving only information from his neighbours, it receives information from all the other nodes of the bar.

#### Topology of the graph

The corresponding graph is represented in fig 4.1.

If we keep the previous transitions matrix $\psi$ between bars (in red), a parameter is still to be determined: the constraint matrix $\psi'$ between nodes of the same bar (in blue).

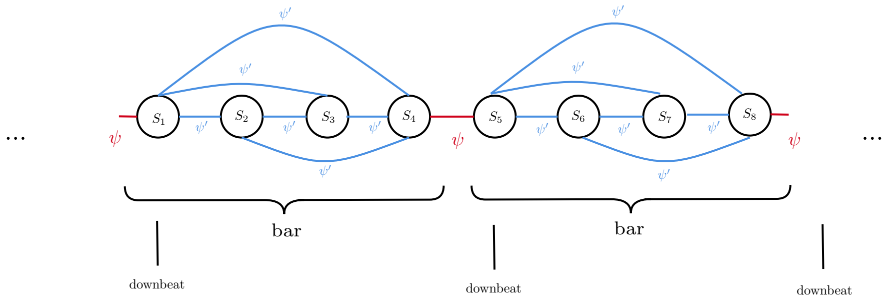


Figure 4.1: Taking into account the downbeats. The red transitions are between bars and use the previous transition matrix; the blue transitions are between elements of the same bar

We first think of a constraint matrix defined by self transitions (the probability that the chords are the same) and the other probabilities are uniformly distributed:

$$\psi'(i,j) = \begin{cases} \alpha & \text{if } i = j \\ \dfrac{(1-\alpha)}{N_D - 1} & \text{else} \end{cases} \tag{4.1}$$

**Choice of the constraint**

To determine $\alpha$, different values have been tested ranging from $\frac{1}{N_D}$ to 1. Values lower than $\frac{1}{N_D}$ are not tested: they imply that the self transition is disadvantaged, which is contrary to the assumption that chords are mainly identical in a bar. This creates a lot of contradiction between the observations and the transitions imposed by the graph and the BP doesn't converge.

Surprisingly, the best results have been obtained for a self transition of $\alpha = 0.05$: it could be expected that higher values give better values since they bind the elements of a bar in a stronger way, and chords of the same bar have high probabilities to be identical.

Another fact to mention is that this adds a lot of messages to compute but the number of messages updates is lower than in the simple BP algorithm: this creates short 3 or 4 nodes long cycles, that converge very quickly. The drawback is that it could also lead to the non-convergence of the BP, if there is a strong contradiction between observations and messages given by the neighbours: this is why we assume that a low value for $\alpha$ give good results (see 3.3.5 for more details about short cycles).

### 4.1.2 Taking into account the structure of the song

As described previously, taking into account the structure could enhance the performances of the BP inference.

**Topology of the graph**

As for the downbeats, taking into account the structure allows to add more information to each node: instead of receiving only information from its neighbours, it also receives information from all the nodes that share the same position in the same structure. For example, the first node of the first verse is connected to the first node of all the other verses. The same goes for all the other nodes of the verse and the other structures (see fig 4.3).

The choice for the constraint matrix between elements that are connected by the structure (long distance influence) is the same that for the downbeats: we take the same matrix defined by self transitions (the probability that the chords are the same) and the other probabilities are uniformly distributed.

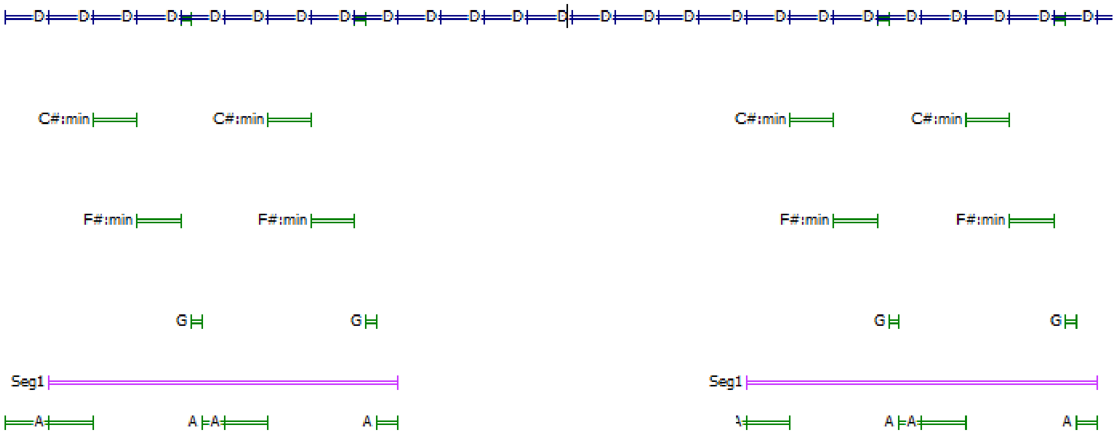


Figure 4.2: Two elements of the structure of *Help !* . The downbeats are represented in blue. The segment between these two part has been erased, to highlight the readability of the figure. It appears clearly that the same structure are composed by almost the same chords.

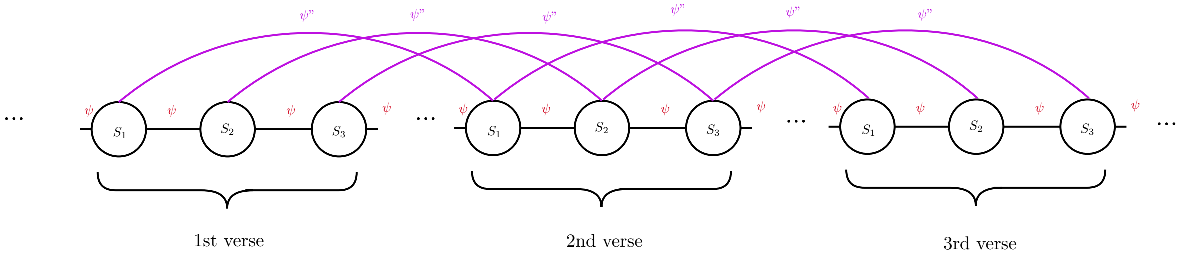


Figure 4.3: Taking into account the structure : transitions between nodes are assured by the previous transitions matrix (in red) and same position element of a same structure are linked with $\psi$" (in magenta)

$$\psi\text{"}(i,j) = \begin{cases} \alpha & \text{if } i = j \\ \dfrac{(1-\alpha)}{N_D - 1} & \text{else} \end{cases} \tag{4.2}$$

### Choice of the constraint

As for the downbeats, we test different values ranging from $\frac{1}{N_D}$ to 1 and as for downbeats, we obtain the best result for $\alpha = 0.05$.
This process adds a lot of links to the graph, and a lot more messages are to be calculated. Contrary to the downbeats, taking into account the structure creates large and big cycles, that take a lot of messages updates to converge: most of the time, they don't reach the convergence criteria in the desired time and the update is aborted (see 3.3.5 for more details about large cycles).

## 4.1.3 Taking into account both structural information

Taking into account both the structure and the downbeats allows to add, again, more information to the nodes. This structure gives the best results and takes advantage of both information: the speed of convergence given by the short cycles created when using information given by the downbeats compensates

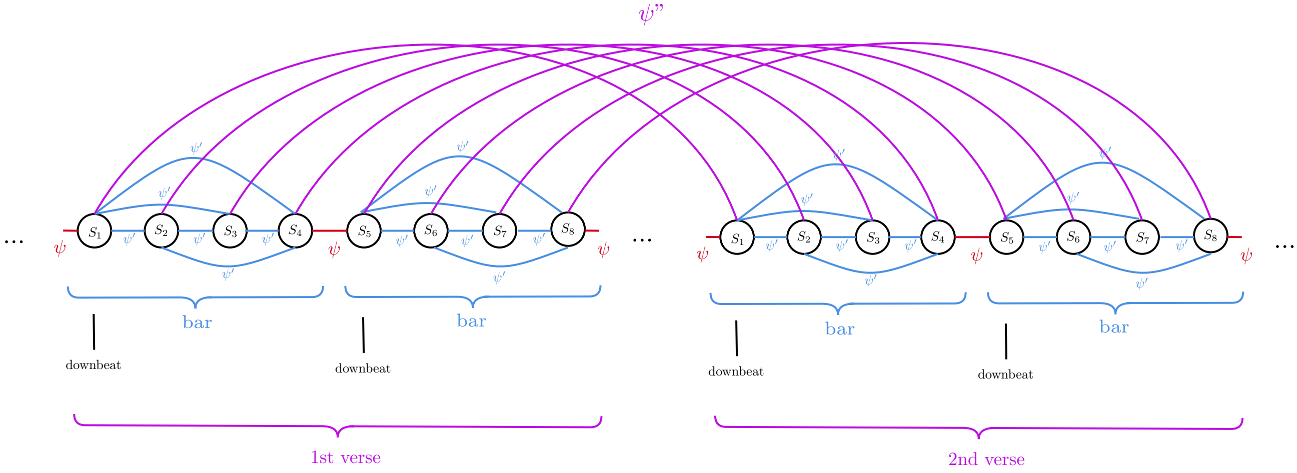


Figure 4.4: Taking into account the structure and the downbeats

long distance large cycles that didn't converge. The convergence was not reached in only 3 songs of the database.

## 4.2 Taking into account similarities

### 4.2.1 Taking into account self-similarity

#### Changing the constraints

An alternative idea could be to change the previous $\alpha$ in $\psi'$ and $\psi$" according to the correlation between the two chroma vectors considered.
To fulfil this idea, we compute the self-similarity matrix [13] defined by:

$$M(i,j) = \frac{Chroma_i \cdot Chroma_j}{||Chroma_i|| \cdot ||Chroma_j||}$$

This matrix is used in structure estimation, and is a reflection of the similarity between all the chroma vectors of the song. If the chromas are similar, $M(i,j)$ is close to 1, if not, $M(i,j)$ is close to 0. Such a matrix is represented in fig 4.5.
Then, when computing the messages:

$$\psi'(i,j) = \begin{cases} M(i,j) & \text{if } i = j \\ \dfrac{(1 - M(i,j))}{N_D - 1} & \text{else} \end{cases} \tag{4.3}$$

and

$$\psi"(i,j) = \psi'(i,j) \tag{4.4}$$

This technique leads to results in the same range than the previous model, but with higher calculus time: the calculus of the self-similarity matrix take time.

#### Feeding the graph with similarity

Similarity can also lead to a new way to feed the graph. Instead of having the ground truth downbeats and structure, we propose another method based uniquely on similarity.

The idea is to have a fully connected graph, with all the edges weighted by the similarity between the chromas that they link. The main of having a full connected graph is the fact each node receives information from all the other nodes of the graph. In ACE terms, each chord is determined with information from all the other chords of the song, and not only the one that share structural information. One of the principal drawback is the calculus time: this method requires to compute $N(N-1)$ messages (given that N is the number of nodes of the graph, i.e. the number of tatum of the song).
The constraint between the nodes connected in this manner is the similarity between the chromas.
To avoid too big calculus time, we fix two parameters:

- $\alpha$: a threshold for similarity. If similarity between two chromas is lower than $\alpha$, we don't keep the edge between the corresponding nodes.
- $\beta$: the maximal number of edges connected to a node.

With $\alpha \in 0.9, 0.95, 0.98$ and $\beta \in 5, 10, 20$, all the systems had very poor performances. After concerting with the team, we chose to persevere and to propose something a little more elaborated.

### 4.2.2 Taking into account similarity and anti-similarity

The idea is that it is possible to have more information than just similarity. The idea comes from the fact that with the previous method, if two chromas have "strong" information but do not represent the same chords, they are not taken into account.

For example, taking the theoretical case of two chromas, $Chroma_i$ and $Chroma_j$ defined by $Chroma_i = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$ and $Chroma_j = \begin{bmatrix}0\\0\\1\\0\\0\\0\\1\\0\\0\\1\\0\\0\end{bmatrix}$ $Chroma_i$ is the theoretical chroma for a C Major chord (C-E-G) and $Chroma_j$ is the theoretical chroma for a D Major chord (D-F#-A). Both have strong observation (it is easy to retrieve the chord from the chroma) but if we compute the similarity between $Chroma_i$ and $Chroma_j$, we found $M(i,j) = 0$. The aim of this part is to introduce new operators for "anti-similarity", i.e. chroma that have strong observation and low similarity, and to use this information to feed the graph.

**What is called ”similarity and anti-similarity” ?**

We define the following operators:

- The *Upper Chroma Translation* :

  $T_k(Chroma_i)$ = chroma vector $Chroma_i$ with a shift k.

  For example, if $Chroma_i = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$, $T_1(Chroma_i) = \begin{bmatrix}\uparrow 1\\0\\0\\0\\\uparrow 1\\0\\0\\\uparrow 1\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\\1\end{bmatrix}$, $T_2(Chroma_i) = \begin{bmatrix}\uparrow\uparrow 1\\0\\0\\0\\\uparrow\uparrow 1\\0\\0\\\uparrow\uparrow 1\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\\0\\1\\0\\0\\0\\0\\1\\0\end{bmatrix}$, ...

- The *Vector of Translations between two chromas* $(f_{i,j}^{obs})_{12,1}$: it is the vector of the *Upper Chroma Translations* of the observed chromas $Chroma_i$ and $Chroma_j$ for all the $k$.

$$\forall k \in [[0, 11]] \; f_{i,j}^{obs}[k] = Chroma_i \cdot T_k(Chroma_j)$$

$$f_{i,j}^{obs} = \begin{bmatrix} Chroma_i \cdot T_0(Chroma_j) \\ Chroma_i \cdot T_1(Chroma_j) \\ Chroma_i \cdot T_2(Chroma_j) \\ \vdots \\ Chroma_i \cdot T_{11}(Chroma_j) \end{bmatrix}$$

This vector reflects not only the similarity between two chromas (which is contained in $f_{i,j}^{obs}[0]$) but also the ”anti-similarities”.

For the sake of example, we consider the following observed chromas $Chroma_i$ and $Chroma_j$:

$Chroma_i = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix}$ and $Chroma_j = \begin{bmatrix}0\\0\\1\\0\\0\\0\\1\\0\\0\\1\\0\\0\end{bmatrix}$

We so have:

- $f_{i,j}^{obs}[0] = Chroma_i \cdot T_0(Chroma_j) = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} \cdot \begin{bmatrix}0\\0\\1\\0\\0\\0\\1\\0\\0\\1\\0\\0\end{bmatrix} = 0$

- $f_{i,j}^{obs}[1] = Chroma_i \cdot T_1(Chroma_j) = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} \cdot \begin{bmatrix}0\\0\\\uparrow 1\\0\\0\\0\\\uparrow 1\\0\\0\\\uparrow 1\\0\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} \cdot \begin{bmatrix}0\\1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\end{bmatrix} = 0$

- $f_{i,j}^{obs}[2] = Chroma_i \cdot T_2(Chroma_j) = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} \cdot \begin{bmatrix}0\\0\\\uparrow\uparrow 1\\0\\0\\0\\\uparrow\uparrow 1\\0\\0\\\uparrow\uparrow 1\\0\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} \cdot \begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} = 3$

- ...

- $f_{i,j}^{obs}[11] = Chroma_i \cdot T_11(Chroma_j) = 0$

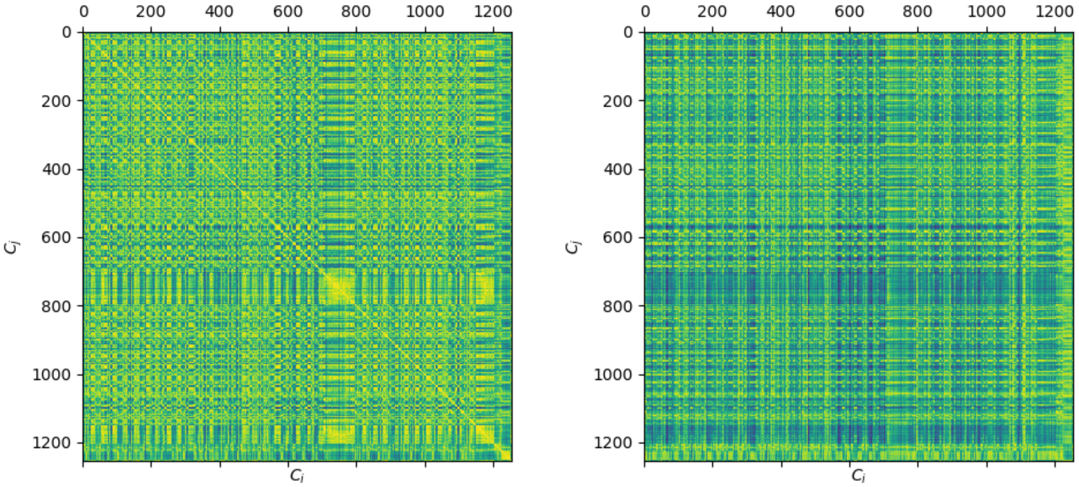


Figure 4.5: Left: Self Similarity matrix of the song *Twist and Shout*. Right: the same matrix but with a 9 steps circular permutation of all the $Chroma_j$

We obtain: $f_{i,j}^{obs} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

The idea in now to construct a full graph weigthed not only by similarity but also by anti-similarity and only keep the edges that are weighted above $\alpha$ and a maximum of $\beta$ edges by node. Before that, we need to introduce a new transitions matrix between the nodes, based on anti-similarity.

### Basis of theoretical *Upper Chromas Transitions*

Before computing this new transitions matrix, we need to compute a *basis of theoretical Upper Chromas Transitions*. The idea here is to decompose the previous Vector of Translations between two chromas ($f_{i,j}^{obs}$) in a basis of theoretical transitions, for all the possible transitions.
With the models of [15] ( taking into account the harmonics, see 3.1) the *basis of theoretical Upper Chromas Transitions* $(f^{th})_{12,24}$ is defined by:

$$f^{th} = \{Chroma^{th}(C), ..., Chroma^{th}(B), Chroma^{th}(Cm), ..., Chroma^{th}(Bm)\}$$

with:

$$f_C^{th} = \begin{bmatrix} Chroma^{th}(C) \cdot T_0(Chroma^{th}(C)) \\ Chroma^{th}(C) \cdot T_1(Chroma^{th}(C)) \\ Chroma^{th}(C) \cdot T_2(Chroma^{th}(C)) \\ \vdots \\ Chroma^{th}(C) \cdot T_{11}Chroma^{th}(C)) \end{bmatrix} f_{C\#}^{th} = \begin{bmatrix} Chroma^{th}(C) \cdot T_0(Chroma^{th}(C\#)) \\ Chroma^{th}(C) \cdot T_1(Chroma^{th}(C\#)) \\ Chroma^{th}(C) \cdot T_2(Chroma^{th}(C\#)) \\ \vdots \\ Chroma^{th}(C) \cdot T_{11}Chroma^{th}(C\#)) \end{bmatrix}, ...,$$

$$f_{Bm}^{th} = \begin{bmatrix} Chroma^{th}(C) \cdot T_0(Chroma^{th}(Bm)) \\ Chroma^{th}(C) \cdot T_1(Chroma^{th}(Bm)) \\ Chroma^{th}(C) \cdot T_2(Chroma^{th}(Bm)) \\ \vdots \\ Chroma^{th}(C) \cdot T_{11}(Chroma^{th}(Bm)) \end{bmatrix}$$

where $Chroma^{th}$ stands for the theoretical chromas according to the model explained in 3.1.

### Orthonormalisation of the basis

This family defines almost a basis in which we would like to project the $f_{i,j}^{obs}$. As it is not a base (the rank of the matrix is 12 for a 24×12 matrix), we choose to split it in two part:

- a Major to Major base $(f_i^{th})_{i \in 1,12}$
- a Major to Minor base $(f_i^{th})_{i \in 13,24}$

The minor to minor transitions are the same as the Major to Major ones and the Minor to Major transitions are the same than the Major to Minor ones.
As these bases are not orthogonal, we apply the process of Gramm-Schmidt to orthonormalise them:

- $f_{ortho}^{th}[0] = f^{th}[0]$
- $f_{ortho}^{th}[k] = f^{th}[k] - \sum_{i=1}^{k-1} \frac{f^{th}[k] \cdot f_{ortho}^{th}[i]}{||f_{ortho}^{th}[i]||^2} f_{ortho}^{th}[i]$

We decompose $f_{i,j}^{obs}$ in the two bases:

$$f_{i,j}^{obs}(MM) = \sum_{i=1}^{12} \alpha_i^{ortho} f_i^{ortho}$$

and

$$f_{i,j}^{obs}(Mm) = \sum_{i=13}^{24} \alpha_i^{ortho} f_i^{ortho}$$

We then compute these coordinates in the original base.
To come back to the base $f^{th}$ from $f_{ortho}$, we compute the transition matrix between base $f^{th}$ and $f_{ortho}^{th}$ (noted R) by a QR decomposition:

$$f^{th} = f_{ortho}^{th} R$$

We so have the transition matrix between the bases. To compute the coordinates in the base $f^{th}$, all we need is multiplying the vectors of coordinates by $R^{-1}$.
Given the $\alpha_i^{original}$, we can construct the transition matrix by blocks :

$$Mat = \left( \begin{array}{c|c} MM & Mm \\ \hline mM & mm \end{array} \right)$$

MM and mm are constructed by the circular permutation of the family $(\alpha_i^{original})_{i \in 1,12}$.
Mm and mM are constructed by the circular permutation of the family $(\alpha_i^{original})_{i \in 13,24}$.

Unfortunately, the orthonormalisation brings negative values of $\alpha$ out. Moreover, the decomposition turns out not to respect the transitions highlighted in $f_{i,j}^{obs}$: this work will have to be enhanced in the future.

One flaw of this method is that the shifted-similarities matrix evaluates if the chroma vectors are similar modulo a shift, but not if they convey information. In fact, two chroma vectors can be very similar and not convey any information at all. For example, in the extreme case where $Chroma_1 = (\frac{1}{24})_{24,1} = Chroma_2$, the self similarity is equal to one but neither vector conveys information.

# Part 5

# Noise robustness

One classical test to evaluate an algorithm in the field of telecoms is its noise robustness. The idea of the following test is to evaluate the efficiency of the different methods of inference on noised corrupted chromas. The flow-chart of the system designed to test noise robustness of the systems is presented in fig 5.1.

## 5.1 Audio

To be assured of the reliability of the test, we work with a simple midi-track, composed of 8 verses containing 4 bars in 4/4 of Em,C,G,D, repeated 4 times each (one different chord per time), resulting in a total of 128 chords (the midi partition in fig 5.2 is repeated 8 times).
The midi partition is then converted to audio (44100 Hz, 16 bits) with the *grand piano* virtual instrument of Ableton Live, at a 60 bpm tempo (one chord per second).

## 5.2 Features extraction

The chromas are extracted with the Python Library Librosa ([34]) following these steps:

1. Extraction of the harmonic part with the function *harmonic(y=y,margin=5)*. The margin set to 5 is the ratio between the harmonic spectrum and the percussive spectrum to be considered as harmonic (see [11]).

2. CQT-chromas computation with the function *feature.chroma_cqt(y=y_harm,sr=sr, bins_per_octave = 12*3)*.

3. Computing an average chroma for each tactus.

## 5.3 Noise addition

Centred Gaussian Noise with a standard deviation of $\sigma$ is then added to the different components of the chroma vectors.
The observations vectors are then computed as in Section 3.1 from the corrupted chromas.

## 5.4 Method

For each algorithm, for each $\sigma$, corrupted chroma vectors are computed on 100 identical songs and the average error rate over the chords recognition is saved. The results are presented in fig 5.3

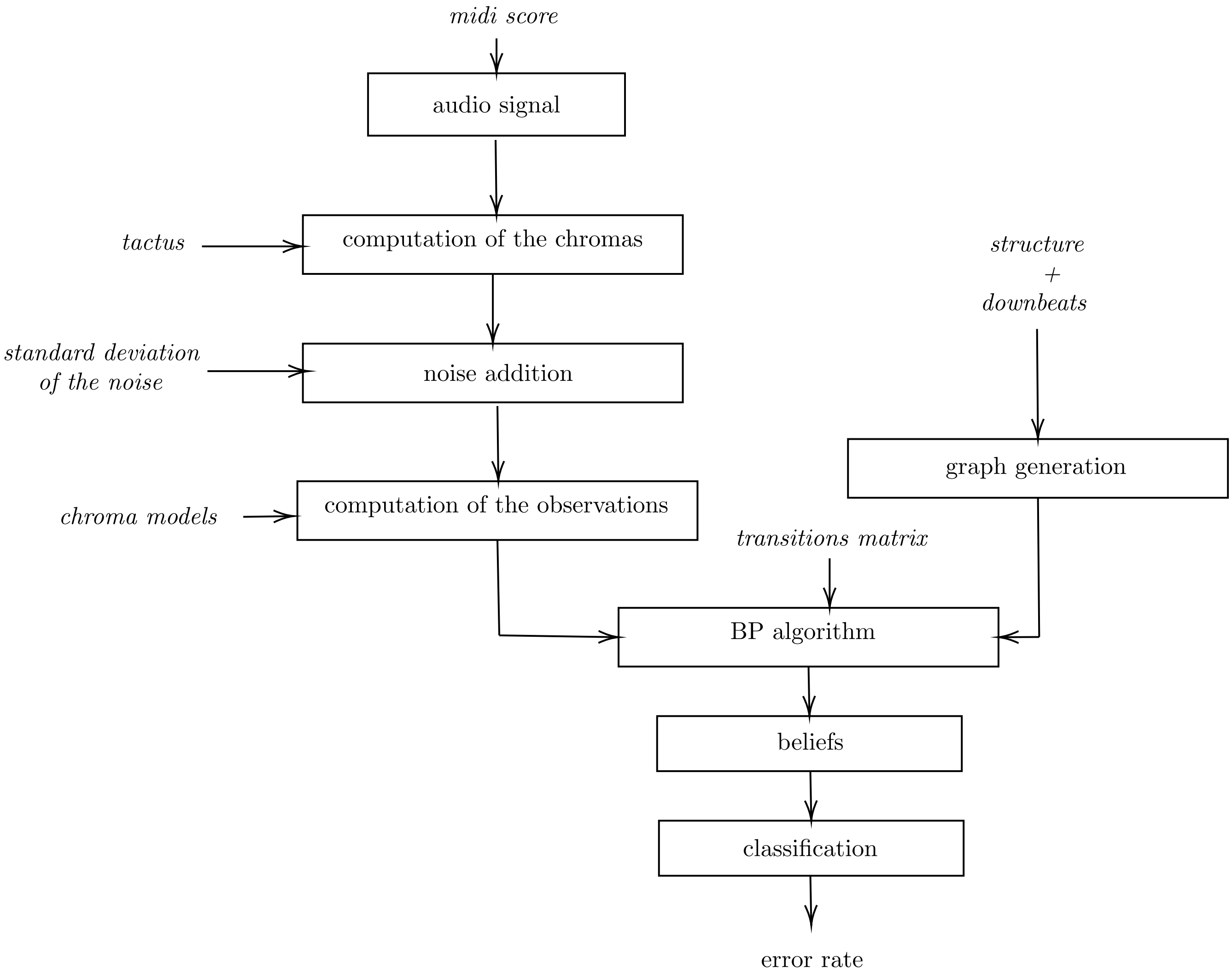


Figure 5.1: Flowchart of the noise robustness test system

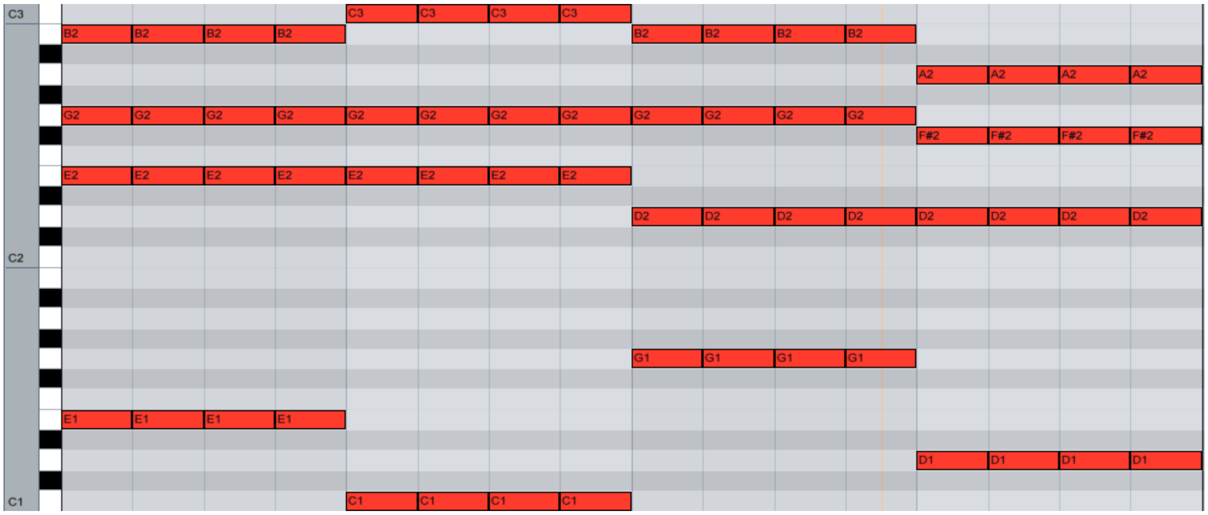


Figure 5.2: MIDI partition of the song used to compute the noise robustness of the different BP algorithm

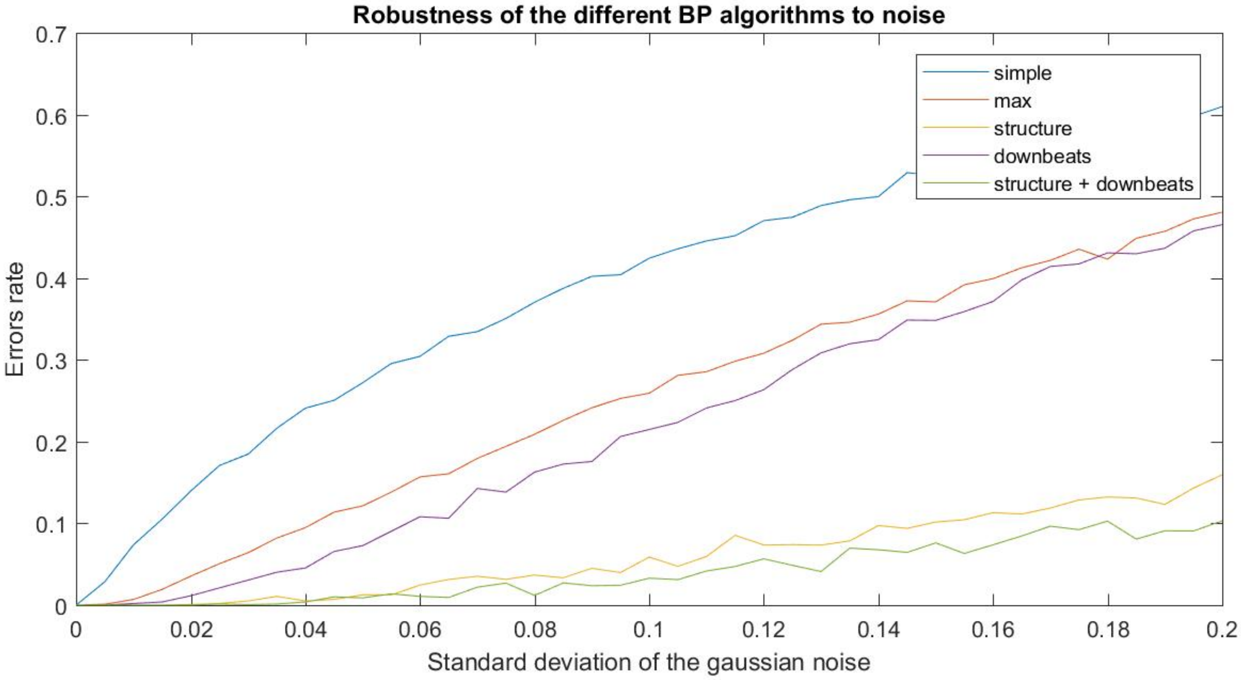


Figure 5.3: Error rate of the different algorithms in function of the SNR

## 5.5 Results and discussion

From this graph it could be assessed that the sum-product BP algorithm (called ”simple”) is more less robust than other algorithms. That comforts us in the choice of the max-product version of the BP.
The relevancy of this study on the musical side could be discussed. We defend that this is relevant because of the fact that adding noise to the chromas allows to take into account the complexity of the music: chroma vectors are sensible to the instrumentation, the presence of percussive elements that randomise the distribution of chroma components. By this study, we have shown that adding information to the BP algorithm allows it to be a lot more robust to these kind of perturbations.

# Part 6

# Results and discussion

## 6.1 Results of the system and discussion

The results of the real world data set are presented in table 6.1 for ground truth beats and downbeats, and 6.2 for estimated beats and downbeats.

| HMM | BP | BP with downbeats | BP with structure | BP both | BP both (cycle of fifths) | BP both (correlation) |
|---|---|---|---|---|---|---|
| 71.31 % | 71.36% | 73.76% | 72.53% | 75.32% | 75.35% | 75.09% |

Table 6.1: Performances of the different systems with ground truth beats and downbeats

| HMM | BP | BP with downbeats | BP both |
|---|---|---|---|
| 70.45% | 70.03% | 71.9% | 73.65% |

Table 6.2: Performances of the different systems with estimated beats and downbeats

The different notations are the following (the observations are the same for all the systems):

- HMM: the simple viterbi HMM as described in the section 3.3, with the perceptual transitions matrix presented in 3.2.
- BP: a simple chain max-product BP algorithm ( section 3.3.7), with the perceptual matrix.
- BP with downbeats: a max-product BP algorithm taking into account the downbeats information as described in 4.1, with the perceptual transitions matrix
- BP with structure: a max-product BP algorithm taking into account the structure information as described in 4.3, with the perceptual transitions matrix
- BP both: a max-product BP algorithm taking into account the downbeats and the structure information as described in 4.4, with the perceptual transitions matrix
- BP both (cycle of fifths): a max-product BP algorithm taking into account the downbeats and the structure information with the cycle of fifths transition matrix
- BP both (correlation): a max-product BP algorithm taking into account the downbeats and the structure information with the constraints explained in 4.2.1

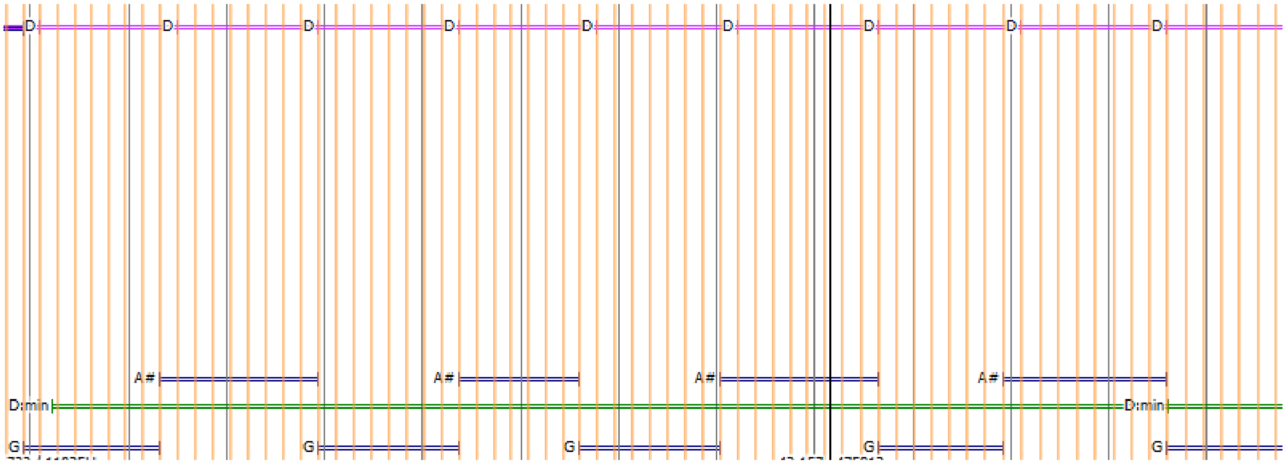


Figure 6.1: Errors made by the BP algorithm taking into account the structure and the bars on *Come together*. Instead of staying in the same chord, the decoding process (in blue) oscillate between two chords that are related to the ground truth (in green) instead of staying in the same state. Orange vertical line are the tatum and magenta D represent the position of the downbeats

The difference between the BP with the perceptual transitions matrix and the cycle of fifth transitions matrix is very slight. An unpaired t-test gives a probability p=0.98 for the null hypothesis at 95%: the difference is far from significance !
Moreover, the same occurs for ”BP both” and ”BP both with correlation” (p=0.54229).
These good results have to be nuanced. If on some songs, the recognition rate can be over 95%, some give result lower than 50%. The identified causes of these errors are:

- The restriction of the dictionary (non major/minor chords that haven't been well mapped to the major-minor equivalent). It is interesting to note the fact that in this case, these are computational errors and the chords proposed by the BP are musically acceptable.
- The transitions between states. As shown in fig 6.1, instead of identifying the chord on all its duration, the algorithm tends to oscillate between two states that are related to the ground truth chord. The importance of the transitions between states has already been shown previously and is illustrated here: if the self-transition probability is raised, the previous problem will fade but short chords transitions will not be detected. On the contrary, if the self transition is diminished, short chord transitions will be very well detected but long time chords will have a poor detection rate as in the previous example.

It should also be noted that with estimated beats and downbeats, the results are lower than with ground truth's ones but BP keeps it advantage over the HMM.

## 6.2 Methodology discussion

### 6.2.1 Reliability of the ground truth

The measure of the performance of a system is usually done by the measure of the recall of the ground truth. But the ground truth itself depends on the people that elaborate it. In [24], a team attempted to elaborate a system that would take into account the subjectivity of the annotators. This practice should be given more attention in the next years.[20] made the effort to compare the results of their systems with ground truth but also with two independent annotators. Their results show that the WSCR is not the best way to measure the performances of a system and that over a certain threshold, a good WSCR doesn't have sense if it is higher than the one of the annotator.

Moreover, as pointed out by [21], the fact that the dictionary is limited to the 24 major and minor chords can cause some further errors. The aim of this work is not to enhance the performances of the whole system but only the pattern matching part: the HMM and the different BP systems are compared with the same dictionary and the same features. Only a few systems use large dictionaries (cf Appendix D), that limit them.

### 6.2.2 Methodology flaw

In [51, 52, 50], Bob Sturm points a flaw in the habitual MIR methodologies. In a consequent number of systems proposed in MIR, the aim is really not to fulfil a given task to but to mimic the ground truth. With that method, aberrations emerge, like beat-tracking systems that are absolutely not robust to a 5% change of bpm or genre classification systems that are able to classify noise and non-musical signals. The questioning of the methodology is unfortunately too rare in the papers while it should be automatic: the glass ceiling faced in MIR is, in part, due to the disconnection between the systems and their intrinsic task.

Even if we assume that our system does estimate the chords, the application of the so-called "irrelevant transformations" elaborated in [50] could be applied to our project to be certain of its reliability .

## 6.3 Future work

Future works about this project may include:

- Studying the influence of the cycles on the stability and the computation time ;
- Using the results on BP to enhance the system ;
- Completing the study about the shifted similarities ;
- Applying irrelevant transformations to the system as discussed before. This could include transformation such as key shifting or the keeping only the maximum in the chromas.

## 6.4 Used and learnt skills

Working on this subject has required some skills and has been the opportunity to get new competences.

### 6.4.1 Used skills

The following skills have been used to complete successfully this project:

- Signal processing tools like DFT, MFCC and Chroma vectors ;
- Diverse competences on MIR acquired during the classes of Musical Acoustics and Audio Signal Processing ;
- High level computing languages (Matlab, Python).

### 6.4.2 Learnt skills

- Julia language
- Scientific practices and strictness
- How to do a state of the art
- Scientific communication

# Conclusion

This project has been for me the opportunity to propose a new inference method for Automatic Chord Estimation and more generally for MIR. This method gives better results than the classical Viterbi HMM. This has reinforced my desire to work in the research field: meeting passionate people and exploring and bringing to the community new systems that outperform the state-of-the-art approaches made this experience very satisfying and very rewarding. Working for an interdisciplinary project was very interesting because of the rigour and the teaching skills it requires to communicate with both of the fields, but also for the talks and the ideas that emerge from discussions with non specialists of the field. If the opportunity is given to me, I will continue this work in a PhD Thesis.

# Appendix A

# Basic Music Theory for ACE

Automatic Chord Estimation need some very basic theory about music.
A musical work can be described as an harmonic content organised in a given temporal structure (beats or tactus).
This part is divided in two sub-parts: rhythm, and harmony, that compose the essence of music.

## A.1 Rythm

The rhythm is the temporal aspect of music. It deals with the question: when is the harmonical content present or when does it change ?

### A.1.1 Tempo

The tempo is the main pulsation of a song, that almost everybody can clap one's hand. It is almost constant in the song but can locally change a little.

### A.1.2 Tactus

The tactus is the unit of the beat. Each time one clap in its hand during the tempo tracking, it is a tactus. Because sometimes the chords change between the tactus, we work at the *tatum* level, which is the half of a tactus.
These notions are subjective and in some cases, the tempo and the tactus can be subject to interpretation (it is not uncommon to have a 2 factor in the tempo between two beat-tracking systems).

## A.2 Harmony

The harmony deals with the question: at a given time, what notes are being played ?

### A.2.1 The 12 semi tones scale

In the Western music system, the basic element of harmony is the scale, composed by 12 semi tones, as shown in fig A.1.

### A.2.2 What is a chord ?

Chords are conventionally created from the shown above scale. Our study is limited to major and minor chords, which are the simplest 3-notes chords. They are composed by:

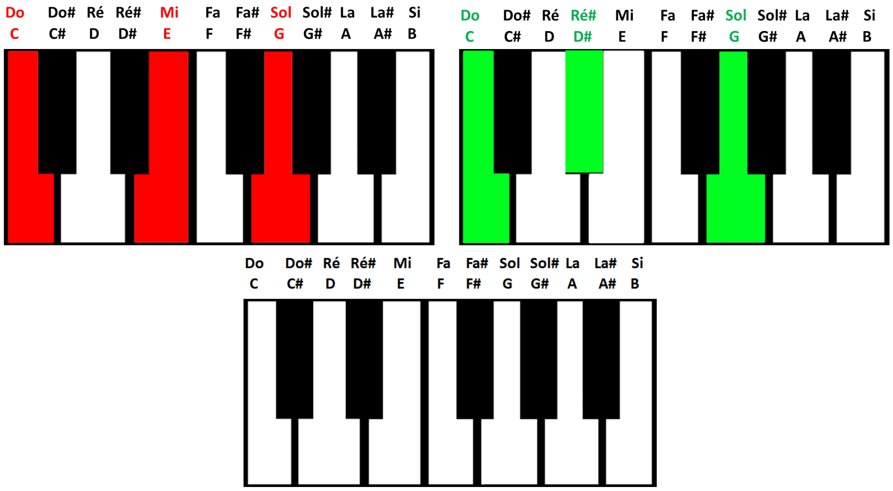

Do Do# Ré Ré# Mi Fa Fa# Sol Sol# La La# Si
C C# D D# E F F# G G# A A# B
Do Do# Ré Ré# Mi Fa Fa# Sol Sol# La La# Si
C C# D D# E F F# G G# A A# B
Do Do# Ré Ré# Mi Fa Fa# Sol Sol# La La# Si
C C# D D# E F F# G G# A A# B

Figure A.1: The 12 semi tones scale and the composition of C Major and C minor chords

- the fundamental, that gives its name to the chord.
- the fifth, that is the note 8 semi-tones higher than the fundamental.
- the third, that is 5 (major chord) or 4 (minor chord) semi-tones higher than the fundamental.

All these shifts are considered modulo 12. An example of chord is given in fig A.1.

### A.2.3 Notation with indexes

Chords are often not named after their label but after a number, comprised between 1 and 24 in our case (12 major chords and 12 minor chords). The correspondence is in Table A.1.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| C | C# | D | D# | E | F | F# | G | G# | A | A# | B |

| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Cm | C#m | Dm | Dm | Em | Fm | F#m | Gm | G#m | Am | A#m | Bm |

Table A.1: Correspondences between chords rank and their label

The same exists for notes, that are numerated from 1 to 12.

# Appendix B

# Chromas and common audio signal processing tools in MIR

The general flowchart to compute chroma vectors is presented in [39]: we present here only the crucial steps and refer the reader to the previous references for more information.

## B.1 FFT and MFCCs

The main idea when dealing with audio signals is to compute the DFT. But when dealing with musical audio signals, DFT is not enough: two signals can be the same note and have different spectral profiles (for example two different octaves: see figB.1).

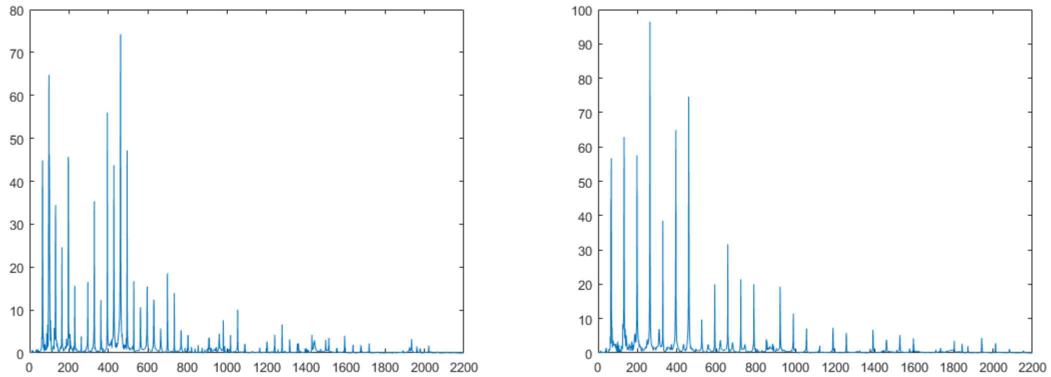


Figure B.1: DFT of a C0 and a C1. The profile of the DFT doesn't allow one to identify the note and is very sensitive to timber and noise.

Moreover, the DFT uses a linear frequency scale while human perception of pitch is logarithmic. To have a more relevant feature, we compute the Mel Frequency Cepstral Coefficients (MFCCs), obtained by filtering the DFT by a bank of filters defined by:

$$n(f_k) = 69 + 12\log_2 \frac{f}{f_{tuning}}$$
$$H_{n'}(f_k) = \frac{1}{2}\tanh(\pi(1 - 2(R|n' - n(f_k)|))$$

where:

- $n(f_k)$ is the midi note number corresponding to the frequency $f_k$ considering the tuning frequency of the song is $f_{tuning}$ (conversion from frequency $f_k$ to midi note $n$)
- $n'$ is the centre midi note of the filter

This bank of filters is represented in fig B.2
The output of each filter is a MFCC (as represented in fig B.2) and allows computing chromas vectors.

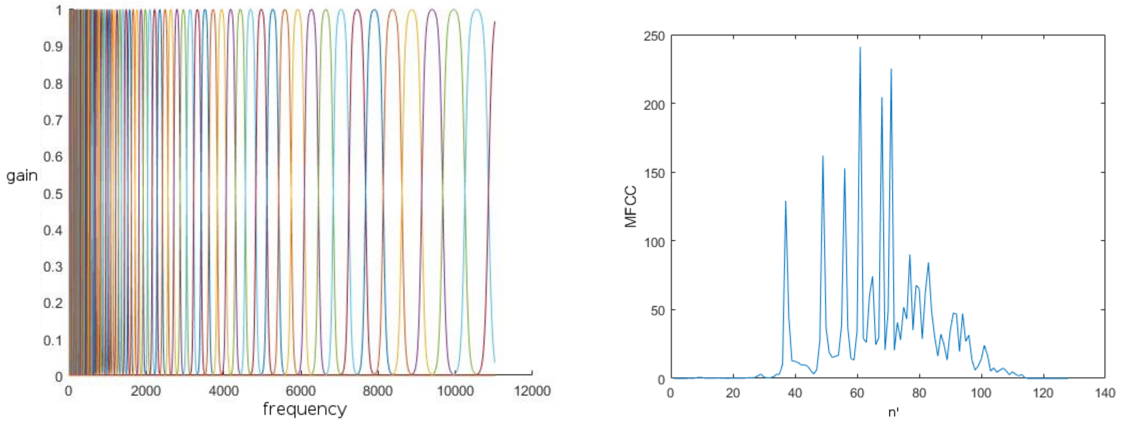


Figure B.2: Bank of filters to compute the MFCCs

## B.2 Chromas vectors

The last step to complete to have the chroma vectors is to gather them by label (C0 with C1, C2, C3, ...) to obtain a chroma vector, represented in fig B.3. The major peaks are at the positions of the notes composing the chord (C, E and A) but the other components are not null: the timber of the violin implies that each note has harmonics that appear on the chroma.
As shown by [53], chroma vectors are robust features, that resist noise and bad sample rates. It is very useful in musical signal processing to have features that are robust to noise because of the percussive parts: the influence of drums or instrumentation on chromas is far less important than on DFT or on MFCCs.
The very last step, when working on a tactus or tatum level, is to average the frame level chromas on the tactus or the tatum.

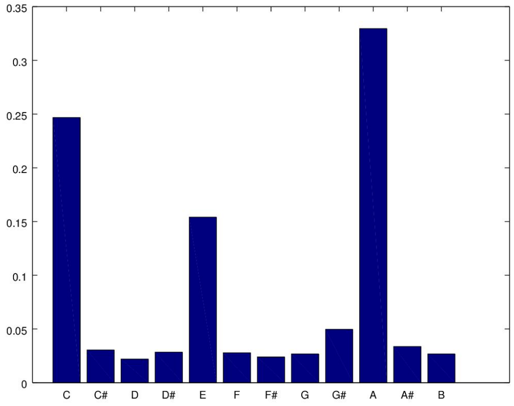


Figure B.3: Chroma vector of a A minor chord played on a violin.

# Appendix C

# Basic Graphs Theory

The Belief Propagation Algorithm uses graphs. But what are they ?

## C.1 Definition of a Graph

The intrinsic definition of graph is a collection of points (called 'nodes') and links between them (called 'edges'). If weights are assigned to the edges, it is called a *weighted graph*.

## C.2 Properties of the graphs

Depending on the different properties of these nodes and edges, some interesting properties can be shown.

### C.2.1 Directed graphs

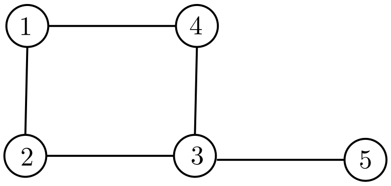
If the edges between nodes are one-way links, the graph is said to be *directed*. On the contrary, is edges between nodes allow a passage on the two way, the graph is *undirected*.

### C.2.2 Adjacency matrix

*The adjacency matrix* of a graph is matrix that summarises all the connection of the graph. It is defined by :

$$A(i,j) = \text{weight of the edge that links the node } i \text{ and the node } j$$

For non-weighted graphs, the adjacency matrix is binary. For undirected graphs, the adjacency matrix is symmetrical.



$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Figure C.1: Example of an undirected and unweighted graph and its adjacency matrix

### C.2.3 Tanner graphs

In telecommunications field, Tanner graphs are used to state constraint between nodes. Given how the code has been established at the emission, constraint nodes are constructed to enhance the recovering of the original message throughout a noisy channel. Using a different formalism, the idea is to state that the edges that link two or more nodes are in fact constraint between these nodes. Constraint can be the probability to be identical or different, or for more elaborated graphs, transitions matrix, given the different possible states for the nodes.

## C.3 Full connected graph

One particular graph that is used in the document is the full connected graph. It is a graph where each node is connected to all the other nodes. As a consequence, the adjacency matrix of full connected graphs is a $N \times N$ matrix of ones.

## Appendix D

# State of the Art on Deep Learning for Chord Detection

| Year | Ref | Acoustic Model | Language Model | Training set | Data Set | WCSR |
|---|---|---|---|---|---|---|
| 2005 | [38] | Chromas | **1 Layer ADALINE architecture** | | | |
| 2009 | [31] | **Convolutional Deep Belief Network**<br>Input: spectrogram with 80 components PCA<br>1 input layer and 1 hidden layer (300 units each), filter length of 6 and max-pool ratio of 3 | Audio classification | ISMIR for classification contest database % for testing) | ISMIR for classification contest database % for testing) | |
| 2010 | [16] | **Deep Belief Network**<br>Input: 513 bins DFT<br>3 layers with 50 units | Genre detection | Tzanetakis' dataset (50% for training, 20 % for validating and 30% for testing) | Majorminer dataset | |
| 2011 | [37] | **DBN**<br>Input: spectrogram<br>one hidden layer of 256 units | SVM and HMM | first 30-second exerpt of the MAPS dataset | Poliner and Ellis midi files<br>MAPS<br>Marolt | |
| 2011 | [32] | **Neural Network Based on Particle Swarm Optimization**<br>1 layer of different number of nodes (best: 40 hidden nodes)<br>Input: mid<br>Cadences are given | Neural Network (not precised) | Not precised (whole dataset ? ) | Midi Data of 84 children's ballads | "Recognition Rate" : 86.48% |
| 2012 | [21] | **CNN to learn Tonnetz-space**<br>Input: constant-Q coefficient of 3 frames<br>2 conv. layers (30 kernels(2×7) and 36 kernels (2×13)) and 3 fully connected layers (242,42,7 units) | GMM | 125 randomly selected tiles | 179 Beatles's dataset<br>20 Queen's dataset<br>100 RWC<br>194 from US pop dataset | M/m "Accuracy = 78.41%" |
| 2012 | [19] | **CNN to learn chromas**<br>Input: constant-Q coefficient processed with substractive-divise contrast normlization<br>Different set up | | | Beatles, RWC, Us Pop | |
| 2012 | [2] | same as [37] | **RNN-RBM** + HMM smoothing (multiple parameters)<br>Input: midi | | Piano-midi.de<br>Nottingham<br>MuseData<br>JSB chorales | improvement between 1.3% and 10% over the HMM |
| 2013 | [4] | **I/O RNN-RBM**<br>Input: PCA whitened spectrum<br>same architecture as [37] | variant of Beam search | Piano-midi.de, Nottingham, MuseData, JSB chroales, Pilner&Ellis | Piano-midi.de, Nottingham, MuseData, JSB chroales, Pilner&Ellis | |
| 2013 | [3] | **RBM**<br>Input: 1400 bins spectrogram after PCA-whitening<br>2 layers of 200 units | **RNN** + modified Viterbi<br>Input: last hidden layer of the RBM<br>100 hidden units | whole set | MIREX | M/m 93.6% (volontary overfitting) |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 2013 | [14] | **Auto-encoders with a recursive last layer**<br>Input: 60 bins Spectrogram after logarithmic transformation<br>trained with binary chord templates<br>3 layers of 300 units | Logistic regression layer with 12 outputs | 50% of the dataset (random) | Isophonics & RWC M/m | 77.03% |
| 2014 | [48] | **Rectifiers Unit vs Sigmoid units** fro genre classification<br>Input: 513 bins FFT frame<br>Diverse architectures and training methods<br>Influence of the dropout studied |  | GTZAN 50/25/25 train/validation/test | ISMIR 2004 Genre | ReLU + SGD + Dropout is the best |
| 2014 | [49] | **Stacked Denoising Encoders**<br>Input: multi-resolution multi-frames<br>2 layers with 800 hidden nodes<br>dropout = 0.5 | HMM | 71000 training samples | Beatles Isophonincs<br>Billboard Mm<br>Billboard extended | 70.05%<br>66.46%<br>50.81% |
| 2014 | [46] | **RNN and DNN**<br>Input: magnitude spectrogram<br>RNN: two stacked layers with 250 untis each<br>DNN: 3 layers of 100 units each<br>DNN+RNN | **RNN-NADES**: 150 hidden units for NADEs, 100 hidden units for RNN<br>HMM<br>learnt thresholds | MAPS dataset (270 songs) | 200 tracks for learning, 20 for validation and 50 for testing | DNN+RNN/RNN-NADE is the best: overfitting ? |
| 2015 | [20] | **CNN**<br>Input: constant-Q spetra<br>4 layers | Viterbi |  | multiple corpus | M/m: 75.9%<br>157: 64.9 % |
| 2015 | [47] | **DNN**<br>Input: CQT frames with context window of 7 frames<br>3 softmax layers of 100 units<br>dropout: 0.3 | **RNN**<br>LMST units<br>2 layers of 100 units | 80% of the data base | MIREX | M/m: 75% |
| 2015 | [55] | **RBM + CNN's way filtering**<br>Input :180 bons CQT with PCA and Z-Score normalisation<br>6 layers: 1024,512,256,256,512,1024 units each | HMM+viterbi | 80% of the data base | 180 Beatles<br>100 RWC<br>18 Zweieck<br>19 Queen | 91.9 %: overfitting ? ) |
| 2016 | [10] | 6 × 252 notegram | **DBN**<br>2 hidden layers 800 neurons<br>Dropout 0.5<br>Softmax | SeventhBass : JayChou dataset, CNPop20, KingQueen26, UsPop dataset<br>JazzACE: JazzGuitar99 | SeventhBass : Beatles<br>JazzACE: GaryBurton7 | M/m: 68.53 %<br>SeventhBass: 54.37 %<br>Jazz: 62.33% |
| 2016 | [10] | 6 × 252 notegram | **Multilayer perceptron**<br>2 hidden layers 800 LSTM each<br>Dropout 0.5 | Same as previous | Same as previous | M/m: 67.95%<br>SeventhBass: 54.09 %<br>Jazz: 61.81% |
| 2016 | [10] | 6 × 252 notegram | **Bidirectional-long-short-term-memory**<br>2 hidden layers 800 neurons<br>Dropout 0.5 | Same as previous | Same as previous | M/m: 72.62%<br>SeventhBass: 57.47%<br>Jazz: 66.41% |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 2016 | [43] | **AutoEncoder** to learn 12 dimensional features<br>Input: DFT | Simple Neural Network : features | 10 fold-cross validation | 50 midi of Beethoven solo piano | Accuracy :<br>80.56% vs 34.87% for 12-PCP method |
| 2016 | [27] | **CNN**<br>Input: spectrogram frames<br>see article for architecture<br>Dropout 0.5 | CRF | not precised | Isophonics<br>RWC<br>Robbie Williams | M/m:82.9%<br>82.5%<br>82.8% |
| 2016 | [26] | **DNN**<br>spectrogram 7 frames<br>3 layers with 512 rectifier units | logistic regression |  | Isophonics<br>RWC<br>Robbie Williams<br>total | 79.3%<br>77.3%<br>80.1%<br>78.8% |
| 2016 | [54] | **Mutitask learning** to use root notes<br>Input: early fused spectral and cepstral chroma serie ML<br>64 neurons | HMM | 5-fold cross validation 144 songs | Beatles Set of Isophonics (180 songs) | 78.3% |
| 2017 | [28] | **HMM vs RNN on frame-based chroma and downbeats-based chroma**<br>2 layers of 100 LSTM |  | 571 first songs of Billboard | Billboard 171 others | HMM: 78.9 %<br>RNN: 78.7% (frame based) |
| 2017 | [24] | **Rectifier units to learn new features: Shared Harmonic Interval Profile**<br>Input: 15 frames (7 left and 7 right) of 192 bins CQT<br>1024-512-256 |  | Isophonics and Ni et al. custom dataset | Isophonics and Ni et al. custom dataset |  |
| 2017 | [8] | Comparison between human expectation and learnt RNN expectation |  |  |  |  |
| 2017 | [6] | Tag estimation by CNN and LSTM |  |  |  |  |
| 2017 | [17] | Normalized chroma + specmurt analysis | **Stacked bidirectional LSTM networks** | 8 songs sliced into sequences | RWC | 85.8% |
| 2017 | [36] | **Residual DNN**<br>Input: Normalized and log-compressed chromas<br>15 layers of 1024 units<br>dropout = 0.5 | CRF | 343 random songs of the dataset | 180 Beatles<br>100 RWC<br>65 Robbie Williams<br>19 Queen<br>18 Zweieck | 0.803 Robbie Williams<br>0.771 Queen |
| 2017 | [18] | same as [2] | **SFM: LSTM on frequency domain with different resolutions** | same as [2] | same as [2] | Log-Likelihood :<br>MuseData: -4.80 (A-SFM)<br>JSB Chorales -5.45 (A-SFM)<br>Piano-midi.de : -6.76 (SFM) |
| 2018 | [25] | Comparison of the prediction task by N-grams, LSTM and GRU |  |  |  |  |
| 2018 | [12] | Review of the different usages of Deep learning in MIR |  |  |  |  |

Table D.1: Recap Chart of the different systems

# Bibliography

[1] Juan P. Bello and Jeremy Pickens. A Robust Mid-level Representation for Harmonic Content in Music Signals. *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 304–311, 2005.

[2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling Temporal Dependencies in High-Dimensional Sequences : Application to Polyphonic Music Generation and Transcription. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[3] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio Chord Recognition with Recurrent Neural Networks. In *ISMIR 2013*, 2013.

[4] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. High-Dimensional Sequence Transduction. In *ICASSP2013*, 2013.

[5] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. *ARXIV*, 2016.

[6] Ning Chen and Shijun Wang. High-level Music Descriptor extraction algorithm based on combination of multi-channel CNNs and LSTM. 2017.

[7] Taemin Cho and Juan P. Bello. On the Relative Importance of Individual Components of Chord Recognition Systems. *IEEE/ ACM Transaction on Audio, Speech, and Language Processing*, 2014.

[8] Carlos Concino-Chacón, Maarten Grachten, and Kat Agres. From Bach to the Beatles : the simulation of human tonal expectation using ecologically-trained preditcive models. 2017.

[9] James Coughlan. A Tutorial Introduction to Belief Propagation, 2009.

[10] Junqi Deng and Yu-Kwong Kwok. A Hybrid Gaussian-HMM-Deep-Leanring approach for automatic chord estimation with very large vocabulary. In *ISMIR 2016*, 2016.

[11] Jonathan Driedger, Meinard Müller, and Sascha Dish. Extending Harmonic-Percussive Sepration of Audio Signals. 2014.

[12] Anders Elowsson. Deep Layered Learning in MIR. *Audio and Speech Processing*, 2018.

[13] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. *IEEE International Conference on Multimedia and Expo*, 2000.

[14] Nikolai Glazyrin. Mid-level features for audio chord recognition using a deep neural network. In *Sriya Fiziko-Matematicheskie Nauki*, volume 155, pages 109–117. 2013.

[15] Emilia Gómez. Tonal Description of Polyphonic Audio from Music Content Processing. *INFORMS Journal on Computing*, 2006.

[16] Philippe Hamel and Douglas Eck. Learning Features From Music Audio With Deep Belief Networks. In *ISMIR 2010*, 2010.

[17] Takeshi Hori, Kazuyuki Nakamura, and Shigeki Sagayama. Music Chord Recognition Fraom Audio Data Using Bidirectional Encoder-decoder LSTMs. In *Proceedings of APSIPA Annual Summit and Conference 2017*, 2017.

[18] Hao Hu and Guo-Jun Qi. State-Frequency Memory Recurrent Neural Networks. In *Proceedings of the 34th Internation Conference on Machine Learning*, 2017.

[19] Eric J. Humphrey and Juan P. Bello. Rethinking Automatic Chord Recognition with Convolutional Neural Network. In *11th International Conference on Machine Learning and Applications*, 2012.

[20] Eric J. Humphrey and Juan P. Bello. Four timely insights on automatic chord estimation. 2015.

[21] Eric J. Humphrey, Taemin Cho, and Juan P. Bello. Learning a robust tonnetz-space transform for automatic chord recognition. 2012.

[22] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Reviews*, pages 65–98, 2017.

[23] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding Belief Propagation and its Generalizations. January.

[24] Hendrick Vincent Koops, W.Bas de Haas, Jeroen Bransen, and Anja Volk. Chord Label Personalization through Deep Learning of Integrated Harmonic Interval-based Representations. In *Proceedings of the First International Workshop on Deep Learning and Music joint with IJCNN*, May 2017.

[25] Filip Korzeniowski, David R. W. Sears, and Gerhard Widmer. A Large-scale Study of Language models for chord prediction. April 2018.

[26] Filip Korzeniowski and Gerhard Widmer. Feature Learning for Chord Recognition : the Deep Chroma Extractor. 2016.

[27] Filip Korzeniowski and Gerhard Widmer. A Fully convolutional deep auditory model for musical chord recognition. 2016.

[28] Filip Korzeniowski and Gerhard Widmer. On the Futility of Learning Complex Frame-Level Language Models for Chord Recognition. 2017.

[29] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. 1990.

[30] Antti Laaksonen. Automatic Melody Transcription Based On Chord Transcription. In *15th ISMIR*, 2014.

[31] Honglak Lee, Yan Largman, Peter Pham, and Andrew Y. Ng. Unsupervised feature leanring for audio classification using convolutional deep belief networks. 2009.

[32] Cheng-Jian Lin, Chin-Ling Lee, and Peng Chun-Cheng. Chord Recognition Using Neural Networks Based on Particle Swarn Optimization. In *Proceedings of Internation Joint Conference on Neural Networks*, 2011.

[33] Matthias Mauch. *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, Queen Mary, 2010.

[34] Brian McFee, Collin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa : Audio and Music Signal Analysis in Python. In *SCIPY 2015*, 2015.

[35] Matt McVicar, Raúl Santos-Rodríguez, Yizhao Ni, and Tijl De Bie. Automatic Chord Estimation from Audio : A Review of the State of the Art. In *IEEE*, February 2014.

[36] Shota Nakayama and Shuichi Arai. Residual DNN-CRF Model for Audio Chord Recognition. In *Proceedings of the 5th IIAE International Conference on Intelligent Systems and Image Processing 2017*, 2017.

[37] Juhan Nam, Jiquan Ngiam, Honglak Lee, and Malcolm Slaney. A Classification Based Plyphonic Piano Transcription Approach Using Learned Feature Representation. 2011.

[38] M. A. P. Neshadha Perera and S. R. Kodithuwakku. Music Chord Recognition Using Artificial Neural Networks. In *Proceedings of the International Conference on Information and Automation*, 2005.

[39] Hélène Papadopoulos. *Joint Estimation Of Musical Content Information From An Audio Signal.* PhD thesis, 2010.

[40] Hélène Papadopoulos and Geoffroy Peeters. Local Key Estimation from an Audio Signal Relying on Harmonic and Metrical Structures. *IEEE - Transactions on Audio, Speech and Language Processing*, 2011.

[41] Hélène Papadopoulos and George Tzanetakis. Models for Music Analysis From a Markov Logic Networks Perspective. *IEEE*, January 2017.

[42] Johan Pauwels and Geoffroy Peeters. Evaluating automatically estimated chord sequences. 2013.

[43] Vilailukkana Phongthongloa, Suwatchai Kamonsantiroj, and Luepol Pipanmaekaporn. Learning high-level features for chord recognition using Autoencoder. In *Proceedings of SPIE Vol.10011*, 2016.

[44] Collin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval : a transparent implementation of common MIR metrics. 2014.

[45] Xavier Serra, Michela Magas, Emmanouil Benetos, Magdalena Chudy, Simon Dixon, Arthur Flexer, Emilia Gómez, Fabien Gouyon, Perfecto Herrera, Sergi Jorda, Oscar Paytuvi, Geoffroy Peeters, Jan Schlüter, Hugues Vinet, and Gerhard Widmer. *Roadmap for Music Information ReSearch.* MIRES.CC. Geoffroy Peeters, 2013.

[46] Siddharth Sigtia, Emmanouil Benetos, Nicolas Boulanger-Lewandowski, Tillman Weyde, Artur S. d'Avilar Garcez, and Simon Dixon. A Hybrid Recurrent Neural Network For Music Transcription. 2014.

[47] Siddharth Sigtia, Nicolas Boulanger-Lewandowski, and Simon Dixon. Audio chord recognition with a hybrid recurrent neural network. 2015.

[48] Siddharth Sigtia and Simon Dixon. Improved Music Features With Deep Neural Networks. In *ICASSP 2014*, 2014.

[49] Nikolaas Steenbergen. *Chord Recognition with Stacked Denoising Autoencoders.* PhD thesis, July 2014.

[50] Bob L. Sturm. A Simple Method to Determine if a Music Information Retrieval System is a Horse. *IEEE Transactions on Multimedia*, 2014.

[51] Bob L. Sturm. Revisiting Priorities : Improving MIR Evaluation Practices. In *Proceedings of the 17th ISMIR*, 2016.

[52] Bob L. Sturm, Rolf Bardeli, Thibault Langlois, and Valentin Emiya. Formalizing The Problem Of Music Description. In *15th ISMIR*, 2014.

[53] Julián Urbano, Dmitry Bogdanov, Perfecto Herrera, Emilia Gómez, and Xavier Serra. What is the effect of audio quality on the robustness of MFCCs and chroma features. 2014.

[54] Mu-Heng Yang, Li Su, and Yi-Hsuan Yang. Highlighting root notes in chord recognition using cepstral features and multi-task learning. In *Proceedings of Signal and Information Processing Association Annual Summit and Conference*, 2016.

[55] Zinquan Zhou and Alexander Lerch. Chord Detection using Deep Learning. *ISMIR 2015*, 2015.