Belief Propagation algorithm for Automatic Chord Estimation

Vincent P. Martin, Sylvain Reynal, Dogac Basaran, Hélène C. Crayencour

Belief Propagation algorithm for Automatic Chord Estimation

Vincent P. Martin
LaBRI, UMR 5800
Univ. Bordeaux, FRANCE
vincent.martin@labri.fr

Sylvain Reynal
ETIS, UMR 8051 Univ. Paris Seine
Univ. Cergy-Pontoise, ENSEA, CNRS, FRANCE
reynal@ensea.fr

Dogac Basaran
Ircam Lab
Sorbonne Univ., FRANCE
dogac.basaran@ircam.fr

Hélène-Camille Crayencour
L2S, UMR 8506
Univ. Paris-Sud, CNRS, FRANCE
helene.camille.crayencour@gmail.com

ABSTRACT

This work aims at bridging the gap between two completely distinct research fields: digital communications and Music Information Retrieval. While works in the MIR community have long used algorithms borrowed from speech signal processing, text recognition or image processing, to our knowledge very scarce work based on digital communications algorithms has been produced. This paper specifically targets the use of the Belief Propagation algorithm for the task of Automatic Chord Estimation. This algorithm is of widespread use in iterative decoders for error correcting codes and we show that it offers improved performances in ACE by genuinely incorporating the ability to take constraints between distant parts of the song into account. It certainly represents a promising alternative to traditional MIR graphical models approaches, in particular Hidden Markov Models.

1. INTRODUCTION

This paper focuses on Automatic Chord Estimation (ACE), that is estimating a series of chords from an audio file. Among the oldest tasks ever tackled by MIR, it is essentially an inference problem which consists in estimating the chords of a song (hidden variable) given the audio file from which observations are computed (chromas). While initially relying on hand-crafted features, ACE algorithms have progressively incorporated language models and deep learning in recent times [1]. But even with these advances, the performances of existing approaches stagnate, differing only slightly from one another [2–4].

An important limitation of existing work is that in most models, analysis is typically limited to short timescale. This overlooks long-term structural dependency between music events and does not reflect the rich underlying relational structure. For instance, the chord progression is both related to the high-level semantic structure organization (e.g. in a song, all choruses are likely to have a similar chord progression [5]), and to a lower-level metrical structure organization (e.g. chord changes are likely to happen on downbeats [6]). A fundamental question that remains open is how to model this complex hierarchical relational structure.

The MIR community has explored a handful of approaches to encode long-term structure with short-term analysis. To our knowledge, one of the first to mention harmonic modeling using graphical model is [7], which uses HMM. Another possible scheme is to rely on recurrences in a song to label all music segments of the same type with the exact same chord progression, replacing all identically labeled tatums by their mean chroma [5]. This approach lacks flexibility however since it ignores possible variations between several occurrences of the same structural segment in a piece of music. A more flexible strategy uses Markov Logic Networks [8] in order to model long-term dependencies between chords, but it is limited by a slow inference process, making it difficult to process long pieces and model complex dependencies. More recently Recursive Neural Networks [9] have been considered seeing they can, in principle, model arbitrarily complex long-term temporal dependencies. However, they have exhibited difficulties to make the model learn long-term dependencies from data [10] and do not explicitly use the structure, yet fuzzy information specified by the network. Finally in [11] the strategy elaborated bears some resemblance to ours, namely a graph is designed so that each chord has a short and a long term context. However, the graph construction and the estimation of the chord sequence is not carried out in the same way: where the author use Expectation-Maximisation, we propose a novel approach, based on Belief Propagation algorithm.

These previously mentioned limitations represent an incentive to explore new approaches inspired by other communities, e.g., statistical physics or digital communication, where information is also represented by complex graph models and marginalization represents a difficult challenge [12]. Indeed when computing marginals of probability distributions with a huge number of degrees of freedom, brute force search has an exponential complexity. Numerous algorithms have thus been devised in the last twenty years or so to tackle this issue. Amid those, Belief Propagation (BP) algorithms (also called cluster mean-field algorithms in statistical physics) have emerged as an efficient way to compute marginal probabilities by i) performing iterative updates of local probability distribution (the so-called "beliefs") based on a sort of local survey of opinions — or gossip — and ii) travelling along the Bayesian graph of constraints to update all beliefs in turns.

In this work we propose to go beyond the current limita-

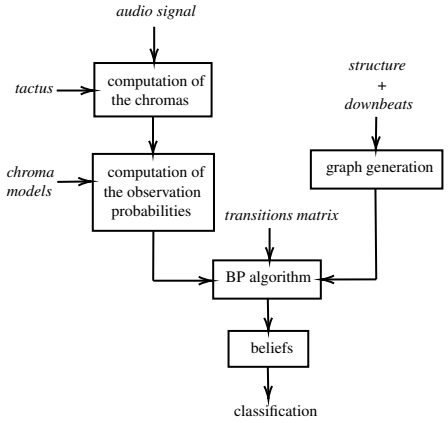

Figure 1. Flowchart of our system. Italic represents the the input of the system, and each box corresponds to a part of the signal processing that is detailed in the corresponding section.

tions of ACE by relying on an approach inspired by iterative decoders for error-correcting codes. We take advantage of the BP algorithm to model and incorporate song structure information in the chord estimation. One of the main benefits of the BP algorithm indeed is that it can embody constraints between any state to be inferred, whatever their proximity on the timeline, so that various long-term correlations can be incorporated in the inference process and make it more robust. In this respect, BP has already been considered as a mean to compute the marginal probability of the state variables in each analysis frame in the case of beat tracking [13] but only correlations between consecutive events were considered. In the present work, we aim at exploring how to encode long-term structure using BP algorithm.

The paper is structured as follows. In Section 2, we provide a brief description of the proposed system and the Belief Propagation algorithm. In Section 3, we present the advantages of using downbeats and the structure to enhance performance. Section 4 presents the dataset, Section 5 proposes a brief study on noise robustness, Section 6 shows the results of the experiment and a discussion. Section 7 discuss some methodology biases. Finally, conclusions and future work are presented in Section 8.

2. BELIEF PROPAGATION FOR AUTOMATIC CHORD ESTIMATION

As in the majority of computational models for ACE [14], our system has a two-step architecture that consists of a feature extraction step (in our case, handcrafted chroma features) followed by a classification step. The latter means inferring hidden states, i.e., putting labels on chords from a chord dictionary by using information from the observations.

The flowchart of our system is presented in Figure 1. Elements in italic are the inputs of the system. They are either obtained from the ground truth or estimated by other means. Given spectral information (chroma) and a model we estimate the probability of a given chord from a set of observation vectors detailed in 2.1. We further feed the conditional probability of a chord given other chords into the estimation process. Based on these inputs a decoding algorithm, e.g., BP or Hidden Markov Models (HMM), computes the most probable sequence of chords, during the pattern matching step.

2.1 Observation and transition probabilities

The observation probabilities are computed from the chroma vectors in the same way as in [6]: each element of the observation vector is the cosine similarity between the chroma and a theoretical template.We compute tatum-synchronous observations, where tatum (smallest time interval between two successive notes [15]) are in our case quarter notes. First, they are computed with ground truth beats, and , in a second time, they are estimated (see Section 4.3).

Transition matrices are an important ingredient of the pattern matching step: they allow us to take into account information from other chords, which in turns improves the inference process. As proposed in [16], we use the perceptual transition matrix elaborated in [17]. Results with the "cycle of fifths" transition matrix proposed in [16] are also provided for completeness in Section 6.

2.2 HMM vs BP

As the main objective of this work concentrates on the pattern matching step, we will devote this section to highlighting the main differences between the HMM with Viterbi inference and the BP algorithms. We show in particular that we can rewrite the HMM algorithm as a particular case of the BP algorithm. Then we will see how we can incorporate structural information to take full advantages of the BP algorithm.

2.2.1 Viterbi with HMM

A HMM is a statistical model that relates the probability vector of a hidden state to observations and transitions probabilities. Let x_i be the chord to be inferred at position i. The algorithm is described by the following parameters: π_i, the probability that x_i is the initial state, a_{ij}, the transition probability from x_i to x_j and $b_i(O)$, the probability that observation O is emitted for chord x_i. In our case, the hidden states x_i are the chords that we want to infer ($x_i \in [[1, N_D]]$, where N_D is the size of the chords dictionary), the observations are the chromas and a_{ij} and $b_i(O)$ are the transition matrix and the model to compute the observations.

State x_0 is initialized as the column vector $(\frac{1}{N_D})_{N_D,1}$; there is no a priori distribution of the probabilities. Then Viterbi inference is carried out as follows:

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

2.2.2 Belief Propagation

BP, is designed to infer hidden states given observations and transition probabilities between them [18]. Yet the BP algorithm is above all an iterative, message-passing algorithm that leverages the topology of the underlying

Bayesian graph to improve estimates. While the viterbi inference with HMM is done linearly, BP can use any topology, including cycles. Modeling with an HMM is inspired by the chronology of events in the song and infers a given state using information from the previous state on the timeline. In this respect, HMM draws more upon directed Bayesian networks. On the contrary, BP can use context to constraint a given chord to any other part of the song.

Let x_i be the chord to be inferred at node i. The BP algorithm relies on the adjacency matrix of the Bayesian graph, the observation vectors $\phi_i(x_i)$ and a constraint $\psi_{i,j}(x_i, x_j)$ between nodes i and j that renders existing correlations. HMM transitions matrices are thus a particular case of transition matrices between neighbouring nodes.

2.2.3 Sum-Product vs Max-Sum versions

Two flavours exist of the BP algorithm, with specific benefits and drawbacks. The Sum-Product algorithm works as follows: for every node j associated with chord x_i to be inferred, we compute the incoming message $m_{i\to j}(x_j)$ from node i using the following "survey" equation,

$$m_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{p\in N(i), p\neq j} m_{p\to i}(x_j), \tag{2}$$

where $N(i)$ is the graph neighborhood of node i. A given message is thus the product of a local observation probability, a constraint and messages coming from the rest of the graph that in effect convey a poll on "what the best estimate of state x_j should be". As the process is iterative — since every node is considered in turn until convergence is reached — this equation can be envisioned as iteratively aggregating more and more of local beliefs as messages propagate through the graph. The messages are normalized at each iteration so that they sum to one.

When convergence is reached, we calculate each chord probability (the so-called *beliefs*) by using

$$b_i(x_i) = \phi_i(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i). \tag{3}$$

and infer the hidden states with

$$x_i = \arg\max_k \{b_i(x_k)\}. \tag{4}$$

Figure 2 shows an example of a message-passing step from node 3 to node 2:

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

$$= \sum_{x_i} \phi_3(x_i)\psi_{3,2}(x_i, x_j) m_{4\to 3}(x_j) m_{5\to 3}(x_j) \tag{6}$$

The Max-Sum version that computes messages according to

$$m_{i\to j}(x_j) = \max_{x_i} \phi_i(x_i)\psi_{i,j}(x_i, x_j) \prod_{p\in N(i), p\neq j} m_{p\to i}(x_j) \tag{7}$$

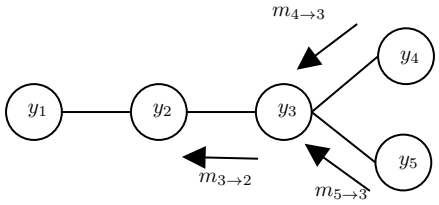

Figure 2. An example of message-passing iteration for the BP algorithm.

It is an interesting, less CPU intensive alternative when one is not interested in the exact marginal probabilities, but only in classification (see [19]), which is our case here. Our results on noise robustness also show that the max-product version provides lower error rates.

2.2.4 HMM viewed as a BP algorithm

HMM can be viewed as a very simple BP algorithm where the Bayesian graph is a simple-path, unweighted directed graph going from initial state $y_0 = S_0 = (\frac{1}{N_D})_{N_D,1}$ to the end of the song, and where messages that are propagated are simply the beliefs, that is,

$$\forall j > i \; m_{i\to j}(c) = \phi_j(c) \times \psi_{i,j}(y_{i-1}, c) \tag{8}$$

$$= O_c \times a_{i-1,c}. \tag{9}$$

with the most probable states y_j being computed at each step by

$$y_j = S_j = \arg\max_k \{m_{j-1,j}(k)\}. \tag{10}$$

2.3 Benefits and drawbacks of the BP algorithm

The BP algorithm can easily take into account non-local correlations by using any appropriate (i, j) edge with specifically tailored constraints.

The method we proposed above may suffer flaws, however. If the graph has has a small girth, which might depend on the song content and structure, the algorithm may converge to an incorrect solution, or even not converge at all.This in particular occurs if the set of constraints along a cycle creates conflicting constraints, so that message propagation may lead to beliefs oscillating between two or more chords at each node in the loop.

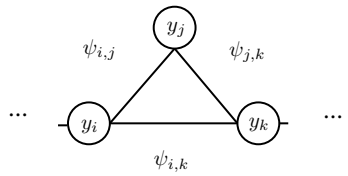

Figure 3. Example of a 3-cycle where the algorithm experiences difficulties converging or does not converge at all.

Populating the Bayesian graph with $\psi_{i,j}$ constraints can lead to short cycles (at the bar scale) or very large ones (at the large-scale structure level). Short cycles undergo convergence issues (as explained above) but may also converge quite quickly. On the contrary, large cycles are stable

but consume a lot of iterations to feedback the information, hence they take time to converge.

A criteria has to be defined to stop the iteration:

$$\forall(i,j)\ \max_k |m_{i\to j}^{n+1}(k) - m_{i\to j}^{n}(k)| \leq \epsilon \qquad (11)$$

We arbitrarily set $\epsilon = 10^{-12}$ and a maximum number of updates of 200 beyond which the messages are considered not to have converged.

3. 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, namely the downbeats and the structure.

As shown in [8], information contained in the structure of a song improves the performance of ACE. It is incidentally quite intuitive that, when listening to a song it is not uncommon, after having roughly identified the structure of the song, to predict the next chords that will be played. BP can leverage this information, creating connections between parts of the song that are very similar, e.g., connecting the first beat of each chorus with the corresponding beat in every other chorus.

Utilizing downbeats proceeds along the same line, yet at the bar scale. As expected by the encouraging results obtained when using downbeats in tonality estimation in [20], using this information in this work has produced encouraging results as well.

3.1 Using downbeats

Inside a bar chords are not independent of each other: songs where chord change every beat are rare and it is not seldom that each chord is repeated twice in a bar. In practice, we connect all the beats of the same bar, with a given probability ψ' to be identical. This assumes that almost all chords in the same bar are identical, and yet the flexibility of the BP guarantees that the turn-over chords at the end of a bar will not be misinterpreted. From a Bayesian graph perspective, including the downbeats positions allows feeding each node with more mutual information: instead of receiving information from its neighbours only, it receives information from all the other nodes in the bar. The corresponding graph with both downbeats and structural information included is shown Figure 4.

Assuming we retain the aforementioned transition matrix ψ between subsequent bars (shown in red in Figure 4), there is still one parameter to be determined, i.e., the transition matrix ψ' between nodes of the same bar (shown in blue in Figure 4). We assume that ψ' is defined primarily by self transitions, i.e., the probability that the chords are identical, while other probabilities are uniformly distributed:

$$\psi'(i,j) = \begin{cases} \alpha & \text{if } i = j \\ \dfrac{(1-\alpha)}{N_D - 1} & \text{else} \end{cases} \qquad (12)$$

To set α, distinct values have been tested ranging from $\frac{1}{N_D}$ to 1. Values lower than $\frac{1}{N_D}$ were not tested: they would imply that self transition are disadvantaged, which would contradict the assumption that chords are mostly identical in a bar. This would then create much frustration in graph cycles and make convergence more difficult. Surprisingly the best results we have obtained are for a self transition of $\alpha = 0.05$: one could have expected indeed that higher values would give better results since they bind events in a bar in a stronger way, and chords of the same bar have higher probabilities to be identical.

All in all this adds a lot of messages to be computed but still the number of messages updates is lower than in the simple chain BP algorithm: short 3- or 4-cycles are created that converge quite quickly. The weakness of it is that it could also preclude convergence if there is contradiction between observations and incoming messages: we thus assume that a low value for α produces good results (see Section 2.3 for more details on short cycles).

3.2 Using the song structure and long-term correlations

Incorporating the structure allows for feeding far more information into each node: now they also receive information from all the nodes that share the same "position" in the song. For example, the first node of the first verse is connected to the first node of all other verses (see Figure 4). Determining the transition matrix ψ" between events that are connected by long-term correlations through the song structure follows the same guidelines as for downbeats: we take the same matrix defined by self transitions while other probabilities are uniformly distributed.

For downbeats we tried several values ranging from $\frac{1}{N_D}$ to 1 and we obtained the best results for $\alpha = 0.05$. This process adds a lot of edges to the graph and a lot more messages thus need to be calculated. As opposed to downbeats, incorporating the structure creates large cycles which also need a lot of updates to converge.

The global graph is represented in Figure 4. Populating the graph with both the structure and the downbeats gives the best results: the quick convergence due to short cycles (downbeats) makes up for large cycles that slow down converge. In only 3 songs of the database convergence was not reached.

3.3 Leveraging similarities

An alternative idea is to change the previous α in ψ' and ψ" according to the correlations between any pair of chroma. Indded, instead of having a tunable parameter that is the same for all the graph, the similarity constraint varies depending on the similarity between nodes. We compute the self-similarity matrix $M(i,j)$ (see [21]) and calculate the messages according to

$$\psi'(i,j) = \begin{cases} M(i,j) & \text{if } i = j \\ \dfrac{(1-M(i,j))}{N_D - 1} & \text{else} \end{cases} \qquad (13)$$

$$\psi"(i,j) = \psi'(i,j) \qquad (14)$$

This technique yields results with the same quality as with the previous model, but with longer computation time.

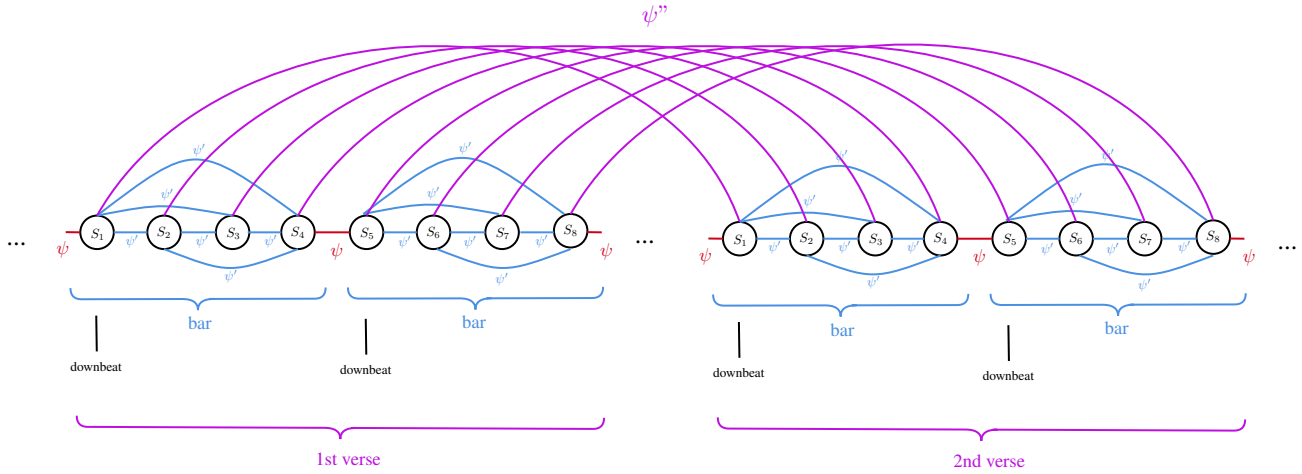

Figure 4. How the structure and the downbeats influence the topology of the Bayesian graph: constraints between chords of a bar are shown in blue ; those between chords of the same structural element are in magenta ; finally, transitions probability between bars are in red.

Similarity can also spark new ways to populate the graph. Instead of having the ground truth downbeats and structure, we introduce another method based uniquely on similarity. The point is to build a fully connected graph, with all edges weighted by the degree of similarity between the chromas which they connect. Each chord is thus to be estimated with information from all the other chords of the song, and not just the ones that share structural position. However, the computation time is quite large as for a graph of size N tatums, this method requires to compute $N(N-1)$ messages.

To keep computation time reasonable we introduce two parameters:

- α is a similarity threshold. If similarity is lower than α, we discard the edge between the corresponding nodes.
- β is the maximum number of edges connected to a givennode.

$\alpha \in 0.9, 0.95, 0.98$ and $\beta \in 5, 10, 20$ yielded very poor performances. Future works will include working on this issue.

4. REAL WORLD DATASET

4.1 Performance estimation

Various methods exist to evaluate the performances of a retrieval task. ACE can be seen as a classification task which requires i) a criterion to tell if the method has reproduced the ground truth to an acceptable degree and ii) classes onto which the different observation can be classified (the so-called dictionary). We invite the reader to refer to [22] for more information on the formalization of Music Information Retrieval.

As in most studies about chord estimation, we consider only the 24 major and minor chords [1]. Chords in the ground truth that are not in the dictionary are projected to major or minor chords (as in [23]), so that the recall can be computed.

We evaluate the performances of the system with the python library *mir_eval* [23]. The performance is measured by the Weighted Chord Symbol Recall (WCSR) defined in [24].

4.2 Database

The various inference algorithms are tested on the Beatles subset of the Isophonics data set. Following [16], some songs are not considered, due to the uncertainty on their structure or the errors in the ground truth provided by Isophonics. These songs are listed in the following list:

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

We considered 157 songs in our data set. For each song the *wav* audio file, the annotated chords and their respective starting and ending time are available.

4.3 Estimated vs ground truth information

To estimate the robustness of our algorithm against variations in the beats and the downbeats, we computed the performance of the system using ground truth beats and downbeats but also using estimated beats and downbeats. These estimated beats and downbeats are processed with state of the art Python library *madmom* [25]. The algorithms contained in this library use Recursive Neural Networks and Deep Bayesian Networks. The only drawback

of this library is that the downbeats can be estimated only with some rhythmic signatures (only those that are over 4). 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 difficult (for example, Irish traditional music contains a lot of *jig* in 6/8 or *slides* in 12/8).

5. NOISE ROBUSTNESS

Algorithms in digital communications are usually rated through their robustness to noise. Likewise here, the idea is to evaluate the efficiency of the various inference methods on "noise-corrupted" chromas. The flow-chart of the corresponding system is represented in Figure 5. We work with

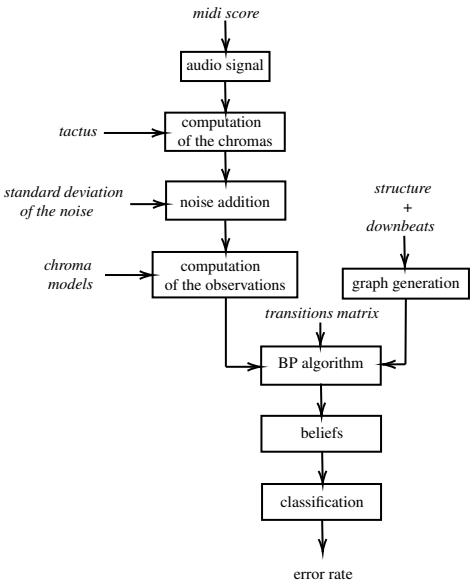

Figure 5. Flowchart of the test system for estimating the robustness to noise

a simple midi-track made up of 8 verses containing 4 bars in 4/4 of Em, C, G and D, repeated 4 times each (one distinct chord per bar), resulting in a total of 128 chords. The midi partition is then converted at 60 BPM to CD quality audio using the *grand piano* virtual instrument of Ableton Live. Chromas are then extracted using the Python Library Librosa [26]. First the harmonic part is extracted with the function *harmonic(y=y,margin=5)* (see [27]) and then CQT-chromas are computed with the function *feature.chroma_cqt*. Finally, an average chroma for each beat is computed.

Gaussian Noise with a standard deviation of σ is then added to the chroma vectors and the observations vectors are computed as in [28] from the corrupted chromas. For each algorithm and each σ, a set of 100 corrupted chroma vectors is generated and the average chord estimation error rate is recorded. The results are presented in Figure 6. We see that the max-product BP clearly beats the the sum-product flavour.

All in all, we argue that adding noise to the chromas allows for blending the whole complexity of music into the performance estimation process: chroma vectors are indeed sensitive to arrangements, e.g., percussive events that may randomize the distribution of chroma components. In addition, we have shown that adding long-term constraints improves the overall robustness of the inference process over such perturbations.

6. RESULTS AND DISCUSSION

Chromas and observation probabilities are computed with Matlab while subsequent steps are implemented in Julia [29].

The results over real world data set are presented in Table 1. Observations are the same for each row. HMM refers the simple Viterbi algorithm (see 2.2), while BP refers to Belief Propagation using the perceptual matrix. Rows 3, 4 and 5 take downbeats or structure information or both (see Figure 4) into account, respectively. Row 6 also includes the "cycle of fifths" transition matrix, while row 7 includes similarities as explained in Section 3.3.

Results with the perceptual transitions matrix and those with the cycle of fifth transitions matrix are very close. An unpaired t-test gives a probability p=0.98 for the null hypothesis at 95%: the groups are not statistically different. Moreover, the same occurs for "BP both" and "BP both (correlation)" (p=0.54229).

System	Ground truth	Estimated
HMM (Viterbi)	71.31 %	70.45 %
BP (perceptual)	71.36%	70.03 %
BP with downbeats	73.76%	71.9%
BP with structure	72.53%	-
BP both	75.32%	73.65%
BP both (cycle of fifths)	75.35%	-
BP both (correlation)	75.09%	-
State of the Art	86.80 %	

Table 1. Performances of the various algorithms using ground truth beats and downbeats. "Ground truth" column shows the results obtained with ground truth beats, downbeats and structures whereas the "Estimated" column shows those obtained with estimated information. The state of the art system is the system achieving the best performances for MIREX 2018 on the Isophonics dataset with the Maj/min dictionary (FK2 system, by Florian Krebs, Filip Korzeniowski, Sebastian Bck)

These interesting results have to be nuanced however. While in some songs the recognition rate may reach 95%, other yield result lower than 50%. Two main reasons have been identified. First, the restriction of the dictionary (non major/minor chords that have not been well mapped to the major-minor equivalent) leads to computational errors but the chords proposed by the BP are musically acceptable. Second, 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 crucial importance of the conditional probabilities between states is exemplified here: whenever the self-transition probability is increased, the previous issue disappear but then short chords transitions will not be detected. On the contrary, whenever the self transition is lowered, short chord transitions are very well detected but long time chords undergo poor detection.

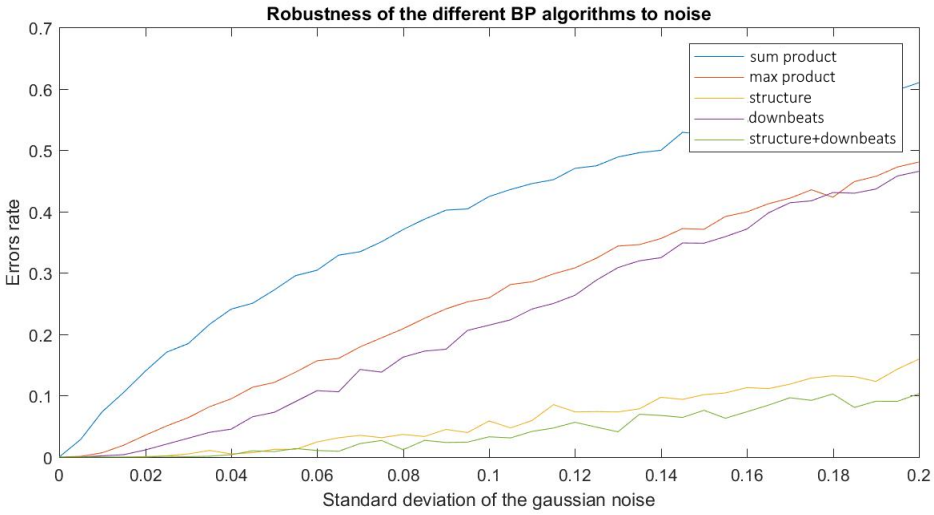

Figure 6. Chord estimation error rate of the various algorithms vs noise amplitude

It should also be noted that with estimated beats and downbeats, results are worse than those using ground truth but BP still stays ahead.

7. METHODOLOGY DISCUSSION

The measure of the performance of a system is usually done by the recall of the ground truth. But the ground truth itself depends on the people that elaborate it. In [30], it was attempted to elaborate a system that would take the subjectivity of the annotators into account. This practice should be given more attention in the following years. In [3] the authors attempted to compare the results of their systems with ground truth but also with two independent annotators. Their results show that WCSR is not the best criterion to measure the performance of a system and that above a certain threshold, an apparently acceptable WCSR does not make sense any more if it is larger than that of the annotator.

Moreover, as pointed out by [31], 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: HMM and the various BP algorithms are compared against the same dictionary and the same features. Only a few systems use large dictionaries.

8. CONCLUSION AND FUTURE WORK

We proposed a new approach to the inference step in ACE that outperforms the HMM one. While the current trend is to develop deep-learning based system, our method does not require training and so does not imply large annotated databases. The architecture of our system only uses structure information and downbeats from the ground truth — or estimated by other ways.

Although we do not make use of this feature in the present work, it is worth mentioning that the very general formulation of the BP algorithm makes it possible to feed any N-point correlation functioninto the iteration process, i.e., one that would describe higher-level correlations between tuples of chords, not just pairs. This opens up the way to taking complex musical context into account.

Future works about this project may include studying the influence of the graph girth on the stability of the inference process and on the computation time. Relying on the Generalized Belief Propagation algorithm [32] might be a way to improve the robustness of the system by suppressing oscillating behaviors. Finally, it would be promising to investigate further how using similarities between observed chromas could help improving the efficiency of the algorithm.

9. ACKNOWLEDGMENTS

This material is based upon work supported by Laboratoire ETIS, UMR 8051, CNRS/ENSEA/Université de Cergy-Pontoise and Université Paris-Seine.

10. REFERENCES

[1] M. McVicar, R. Santos-Rodrguez, Y. Ni, and T. De Bie, "Automatic Chord Estimation from Audio: A Review of the State of the Art," in *IEEE/ACM TASLP*, 2014.

[2] B. L. Sturm, "Revisiting Priorities: Improving MIR Evaluation Practices," in *17th ISMIR*, 2016.

[3] E. J. Humphrey, J. P. Bello, and T. Cho, "Four timely insights on automatic chord estimation," in *16th ISMIR*, 2015.

[4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, 2012.

[5] M. Mauch, "Automatic Chord Transcription from Audio Using Computational Models of Musical Context," Ph.D. dissertation, Queen Mary University of London, 2010.

[6] H. Papadopoulos and G. Peeters, "Joint Estimation of chords and downbeats from an audio signal," in *IEEE - TASLP*, 2009.

[7] R. Christopher and J. Stoddard, "Harmonic analysis with probabilistic graphical models," in *4th ISMIR*, 2003.

[8] H. Papadopoulos and G. Tzanetakis, "Models for Music Analysis From a Markov Logic Networks Perspective," in *IEEE/ACM TASLP*, Jan. 2017.

[9] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d'Avilar Garcez, and S. Dixon, "A Hybrid Recurrent Neural Network For Music Transcription," in *IEEE - ICASSP*, 2015.

[10] F. Korzeniowski and G. Widmer, "On the Futility of Learning Complex Frame-Level Language Models for Chord Recognition," in *AES Conf. on Semantic Audio*, 2017.

[11] J.-F. Paiment, D. Eck, and S. Bengio, "A Probabilistic Model for Chord Progressions," in *6th ISMIR*, 2005.

[12] M. Mezard and A. Montanari, *Information, Physics and Computation*, 2009.

[13] D. Lang and N. de Freitas, "Beat tracking the graphical model way." in *NIPS*, 2004.

[14] T. Cho and J. P. Bello, "On the Relative Importance of Individual Components of Chord Recognition Systems," in *IEEE/ ACM TASLP*, 2014.

[15] J. A. Bilmes, "Techniques to foster drum machine expressivity," in *Int. Comp. Music Conf.*, 1993.

[16] H. Papadopoulos, "Joint Estimation Of Musical Content Information From An Audio Signal," Ph.D. dissertation, 2010.

[17] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*, 1990.

[18] Y. W. Jonathan S. Yedidia, William T. Freeman, *Exploring artificial intelligence in the new millennium*, 2003, ch. Understanding belief propagation and its generalizations, pp. 239–269.

[19] J. Coughlan, "A Tutorial Introduction to Belief Propagation," 2009.

[20] H. Papadopoulos and G. Peeters, "Local Key Estimation from an Audio Signal Relying on Harmonic and Metrical Structures," in *IEEE - TASLP*, 2011.

[21] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *IEEE Int. Conf. on Multimedia and Expo*, 2000.

[22] B. L. Sturm, R. Bardeli, T. Langlois, and V. Emiya, "Formalizing The Problem Of Music Description," in *15th ISMIR*, 2014.

[23] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: a transparent implementation of common MIR metrics," in *15th ISMIR*, 2014.

[24] J. Pauwels and G. Peeters, "Evaluating automatically estimated chord sequences," in *IEEE - ICASSP*, 2013.

[25] S. Bock, F. Korzeniowski, J. Schlter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *24th ACM Int. Conf. on Multimedia*, 2016.

[26] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," in *SCIPY*, 2015.

[27] J. Driedger, M. Muller, and S. Dish, "Extending Harmonic-Percussive Separation of Audio Signals," in *15th ISMIR*, 2014.

[28] E. Gomez, "Tonal Description of Polyphonic Audio from Music Content Processing," *INFORMS Journal on Computing*, 2006.

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

[30] H. V. Koops, W. de Haas, J. Bransen, and A. Volk, "Chord Label Personalization through Deep Learning of Integrated Harmonic Interval-based Representations," in *Proc. of the First Int. Workshop on Deep Learning and Music joint with IJCNN*, May 2017.

[31] E. J. Humphrey, T. Cho, and J. P. Bello, "Learning a robust tonnetz-space transform for automatic chord recognition," in *IEEE - ICASSP*, 2012.

[32] S. Reynal, J.-C. Sibel, and D. Declercq, "An Application of Generalized Belief Propagation: Splitting Trapping Sets in LDPC Codes," in *IEEE Int. Symposium on Information Theory*, 2014.