IDIAP RESEARCH REPORT

Chord Representations for Probabilistic Models

Jean-François Paiement [a] Douglas Eck [b]
Samy Bengio [a]

IDIAP–RR 05-58

September 2005

soumis à publication

[a] IDIAP Research Institute
[b] Université de Montréal

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny − Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Rapport de recherche de l'IDIAP 05-58

Chord Representations for Probabilistic Models

Jean-François Paiement   Douglas Eck   Samy Bengio

September 2005

soumis à publication

Résumé. Chord progressions are the building blocks from which tonal music is constructed. Inferring chord progressions is thus an essential step towards modeling long term dependencies in music. In this paper, three different representations for chords are designed. In a first representation, Euclidean distances roughly correspond to psychoacoustic dissimilarities between chords. Estimated probabilities of chord substitutions are then derived from these distances and are used to introduce smoothing in graphical models observing another chord representation. Finally, a third representation where we model directly each chord components leads to a probabilistic model considering the interaction between melodies and chord progressions. Parameters in the graphical models are learnt with the EM algorithm and the classical Junction Tree algorithm is used for inference. Various model architectures are compared in terms of conditional out-of-sample likelihood. Both perceptual and statistical evidence show that binary trees related to meter are well suited to capture chord dependencies.

1 Introduction

Probabilistic models for analysis and generation of polyphonic music would be useful in a broad range of applications, from contextual music generation to on-line music recommendation and retrieval. However, modeling music involves capturing long term dependencies in time series. This has proved very difficult to achieve with traditional statistical methods. Note that the problem of long-term dependencies is not limited to music, nor to one particular probabilistic model Bengio et al. (1994). This difficulty motivates our exploration of chord progressions and their interaction with melodies. Chord progressions constitute a fixed, non-dynamic structure in time and thus can be used to aid in describing long-term musical structure.

One of the main features of tonal music is its organization around *chord progressions*. A chord is a group of three or more notes (generally six or less). A chord progression is simply a sequence of chords. In general, the chord progression itself is not played directly in a given musical composition. Instead, notes comprising the current chord act as central polarities for the choice of notes at a given moment in a musical piece. Given that a particular temporal region in a musical piece is associated with a certain chord, notes comprising that chord or sharing some harmonics with notes of that chord are more likely to be present. In typical tonal music, most chord progressions are repeated in a cyclic fashion as the piece unfolds, with each chord having in general a length equal to integer multiples of the shortest chord length.

Chord changes tend to align with metrical boundaries in a piece of music. Meter is the sense of strong and weak beats that arises from the interaction among a hierarchy of nested periodicities. Such a hierarchy is implied in Western music notation, where different levels are indicated by kinds of notes (whole notes, half notes, quarter notes, etc.) and where bars establish measures of an equal number of beats Handel (1993). For instance, most contemporary pop songs are built on four-beat meters. In such songs, chord changes tend to occur on the first beat, with the first and third beats (or second and fourth beats in syncopated music) being emphasized rhythmically. Chord progressions strongly influence melodic structure in a way correlated with meter. For example, in jazz improvisation notes perceptually closer to the chord progression are more likely to be played on metrically-accented beats with more "dissonant" notes played on weaker beats. See Cooper and Meyer (1960) for a complete treatment of the role of meter in musical structure.

This strong link between chord structure and overall musical structure motivates our attempt to model chord sequencing directly. With an appropriate chord representation, it is then possible to learn the interaction of chords with melodies. The space of sensible chord progressions is much more constrained than the space of sensible melodies, suggesting that a low-capacity model of chord progressions could form an important part of a system that analyzes or generates polyphonic music. As an example, consider blues music. Most blues compositions are variations of a basic *same* 12 bar chord progression[1]. Identification of that chord progression in a sequence would greatly contribute to genre recognition.

In this paper we present chord representations designed to be embedded in graphical models. These probabilistic models can capture the chord structures and their interaction with melodies in a given musical style using as evidence a limited amount of symbolic MIDI[2] data. One advantage of graphical models is their flexibility, suggesting that our models could be used either as analytical or generative tools to model chord progressions. Moreover, model like ours could be integrated into more complex probabilistic transcription models Cemgil (2004), genre classifiers, or automatic composition systems Eck and Schmidhuber (2002).

Cemgil (2004) uses a somewhat complex graphical model that generates a mapping from audio to a piano-roll using a simple model for representing note transitions based on Markovian assumptions. This model takes as input audio data, without any form of preprocessing. While being very costly, this approach has the advantage of being completely data-dependent. However, strong Markovian

[1]In this paper, chord progression are considered relative to the key of each song. Thus, transposition of a whole piece has no effect on our analysis.

[2]In our present work, we only consider notes onsets and offsets in the MIDI signal.

assumptions are necessary in order to model the temporal dependencies between notes. Hence, a proper chord transition model could be appended to such a transcription model in order to improve polyphonic transcription performance.

Raphael and Stoddard (2003) use graphical models for labeling MIDI data with traditional Western chord symbols. In this work, a Markovian assumption is made such that each chord symbol depends only on the preceding one. This assumption seems sufficient to infer chord symbols, but we show in this paper (see Section 2.3.1) that longer term dependencies are necessary to model chord progressions by themselves in a generative context, without regard to any form of analysis.

Lavrenko and Pickens (2003) propose a generative model of polyphonic music that employs Markov random fields. Though the model is not restricted to chord progressions, the dependencies it considers are much shorter than in the present work. Also, octave information is discarded, making the model unsuitable for modeling realistic chord voicings. For instance, low notes tend to have more salience in chords than high notes Levine (1990).

Allan and Williams (2004) designed a harmonization model for Bach chorales using Hidden Markov Models (HMMs). A harmonization is a particular choice of notes given a sequence of chord labels. While generating excellent musical results, this model has to be provided sequences of chords as input, restricting its applicability in more general settings. Our work goes a step further by modeling directly chord progressions in an unsupervised manner. This allows our proposed models to be directly appended to any supervised model without the need for additional data labeling.

The generalization performance of a generative model depends strongly on how observed data is represented. If we had an infinite amount of data, we could simply represent each chord as the state of a discrete random variable with a number of possible states equal to the total number of possible chords. Unfortunately, typical symbolic music databases are very small compared to the complexity of the polyphonic music signal. To solve this problem, we explore three different ways of including musical knowledge in models for chord progressions. In Section 2, we build a continuous space embedding chords where the Euclidean distance between two chords corresponds to psychoacoustical similarity. In Section 3, we go a step further and convert these Euclidean distances into probabilities of substitution between chords in order to include the chord similarity measure in the graphical model framework. Finally, we present in Section 4 a chord representation that is closer to the data in the sense that we model directly each component of the chords. In each section, we also describe and evaluate a probabilistic model for chord sequences observing these representations. We evaluate these models in terms of prediction ability. Note that it is also possible to sample these models in order to generate chord progressions.

2 Continuous Chord Space

A useful approach for building a statistical model for chord progressions is to include notions of psychoacoustic similarity between chords. This allows the model to redistribute efficiently a certain amount of probability mass to unseen events during training according to musical similarity. To achieve this, we found it more convenient to build a general representation directly tied to the acoustic properties of chords rather than considering some attributes of Western chord notation such as "minor" and "major". A possibility for describing chord similarities is set-class theory, a method that has been compared to perceived closeness Kuusi (2001) with some success. In this section, we consider a simpler approach where each group of observed notes forming a chord is seen as a single timbre Vassilakis (1999). From this timbre information, we derive a continuous distributed representation where perceptually similar chords tend also to be close in Euclidean distance. We propose in Section 2.2 a graphical model that directly observes these continuous representations of chords.

2.1 Chord Representation

More specifically, the frequency content of an idealized musical note i is composed of a fundamental frequency $f_{0,i}$ and integer multiples of that frequency. The amplitude of the h-th harmonic $f_{h,i} = hf_{1,i}$

of note i can be modeled with geometric decaying ρ^h, with $0 < \rho < 1$ Valimaki et al. (1996).

Consider the function

$$m(f) = 12(\log_2(f) - \log_2(8.1758))$$

that maps frequency f to MIDI note $m(f)$. Let $\mathbb{X} = \{\mathcal{X}_1 \ldots \mathcal{X}_s\}$ be the set of the s chords present in a given corpus of chord progressions. Then, for a given chord $\mathcal{X}_j = \{i_1, \ldots, i_{t_j}\}$ with t_j the number of notes in chord $\mathcal{X}_j$, we associate to each MIDI note n a perceived loudness

$$l_j(n) = \max_{h \in \mathbb{N}, i \in \mathcal{X}_j} (\{\rho^h | \texttt{round}(m(f_{h,i})) = n\} \cup \{0\}) \tag{1}$$

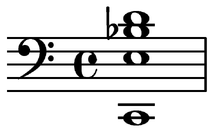
where the function `round` maps a real number to the nearest integer. The max function is used instead of a sum in order to account for the masking effect Moore (1982). The quantization given by the rounding function corresponds to the fact that most of the tonal music is composed using the *well-tempered tuning*. For instance, the 3rd harmonic $f_{3,i}$ corresponds to a note $i+7$ which is located one perfect fifth (i.e. 7 semi-tones) over the note i corresponding to the fundamental frequency. Building the whole set of possible notes from that principle leads to a system where flat and sharp notes are not the same, which was found to be impractical by musical instrument designers in the baroque era. Since then, most Western musicians used a compromise called the well-tempered scale, where semi-tones are separated by an equal ratio of frequencies. Hence, the rounding function in Equation (1) provides a frequency quantization that corresponds to what an average contemporary music listener experiences on a regular basis.

For each chord $\mathcal{X}_j$, we then have a distributed representation $\mathbf{l}_j = \{l_j(n_1), \ldots, l_j(n_d)\}$ corresponding to the perceived strength of the harmonics related to every note n_k of the well-tempered scale, where we consider the d first notes of this scale to be relevant. For instance, one can set the range of the notes n_1 to n_d to correspond to audible frequencies. Using octave invariance, we can go further and define a chord representation $\mathbf{v}_j = \{v_j(0), \ldots, v_j(11)\}$ where

$$v_j(i) = \sum_{n_k : 1 \le k \le d,\ (n_k \bmod 12) = i} l(n_k). \tag{2}$$

This representation gives a measure of the relative strength of each pitch class[3] in a given chord. For instance, value $v_j(0)$ is associated with pitch class `c`, value $v_j(1)$ to pitch class `c` sharp, and so on.

Throughout this paper, we define chords by giving the pitch class letter, sometimes followed by symbol `#` (sharp) to raise a given pitch class by one semi-tone. Finally, each pitch class is followed by a digit representing the actual octave where the note is played. For instance, the symbol `c1e2a#2d3` stands for the 4-note chord



with a `c` on the first octave, an `e` and an `a` sharp (`b` flat) on the second octave, and finally a `d` on the third octave.

Figure 1 show the normalized values given by Equation (2) for 2 voicings of the C major chord, as defined in Levine (1990). We see that perceptual emphasis is higher for pitch classes present in the chord. These two chord representations have similar values for pitch classes that are not present in either chords, which makes their Euclidean distance small. We have also computed Euclidean distances between chords induced by this representation and found that they roughly correspond to perceptual closeness, as the trained musician should see in Table 1. Each column gives Euclidean distances

[3]All notes with the same note name (e.g. C#) are said to be part of the same *pitch class*.

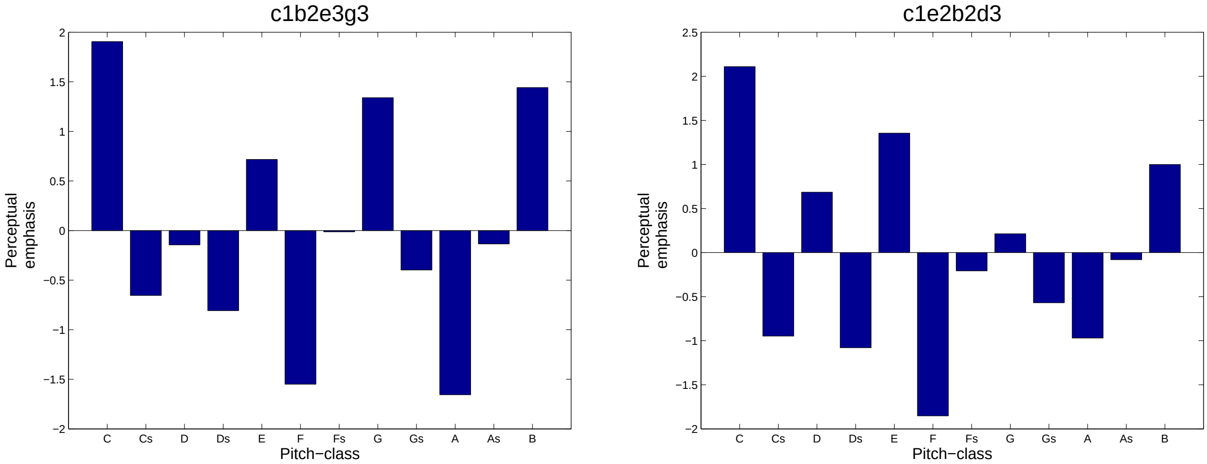


FIG. 1 – Normalized values given by Equation (2) for 2 voicings of the C major chord. We see that perceptual emphasis is higher for pitch classes present in the chord. These two chord representations have similar values for pitch classes that are not present in either chords, which makes their Euclidean distance small.

between the chord in the first row and some other chords that are represented as described here. For instance, the second column is related to a particular inversion of the C minor chord (`c1d#2a#2d3`). We see that the closest chord in the dataset (`c1a#2d#3g3`) is the second inversion of the same chord, as described in Levine (1990). Hence, we raise the note d#3 by one octave and replace the note d3 by g3 (separated by a perfect fourth). These two notes share some harmonics, leading to a close vectorial representation.

This distance measure could have considerable interest in a broad range of computational generative models in music as well as for music composition.

2.2 Graphical Model in the Continuous Space

Graphical models Lauritzen (1996) are a useful framework to describe probability distributions where graphs are used as representations for a particular factorization of joint probabilities. Vertices are associated with random variables. If two vertices are not linked by an edge, their associated random variables are considered to be unconditionally independent. A directed edge going from the vertex associated with variable A to the one corresponding to variable B accounts for the presence of the term $P(B|A)$ in the factorization of the joint distribution for all the variables in the model. The process of calculating probability distributions for a subset of the variables of the model given the joint distribution of all the variables is called *marginalization* (e.g. deriving $P(A,B)$ from $P(A,B,C)$). The graphical model framework provides efficient algorithms for marginalization and various learning algorithms can be used to learn the parameters of a model, given an appropriate dataset.

We now propose a graphical model for chord sequences using the input representation described in Section 2.1. The main assumption behind the proposed model is that conditional dependencies between chords in a typical chord progression are strongly tied to the metrical structure associated to it. Another important aspect of this model is that it is not restricted to local dependencies, like a simpler Hidden Markov Model (HMM) would be. This choice of structure reflects the fact that a chord progression is seen in this model as a two dimensional architecture. Every chord in a chord progression depends both on its position in the chord structure (global dependencies) and on the surrounding chords (local dependencies.) We show in Section 2.3 that considering both aspects leads

TAB. 1 – Euclidean distances between the chord in the first row and other chords when chord representation is given by Equation (2), choosing $\rho = 0.97$.

c1a2e3g3	0.000	c1d#2a#2d3	0.000
c1a2c3e3	1.230	c1a#2d#3g3	1.814
c1a2d3g3	1.436	c1e2a#2d#3	2.725
c1a1d2g2	2.259	c1a#2e3g#3	3.442
c1a#2e3a3	2.491	c1e2a#2d3	3.691
a0c3g3b3	2.920	a#0d#2g#2c3	3.923
c1e2b2d3	3.162	a#0d2g#2c3	4.155
c1g2c3e3	3.398	g#1g2c3d#3	4.363
a0g#2c3e3	3.643	c1e2a#2c#3	4.612
c1f2c3e3	3.914	a#1g#2d3g3	4.820
c1d#2a#2d3	4.295	f1a2d#3g3	5.030
e1e2g2c3	4.548	d1f#2c3f3	5.267
g1a#2f3a3	4.758	a0c3g3b3	5.473
e0g2d3f#3	4.969	g1f2a#2c#3	5.698
f#0e2a2c3	5.181	b0d2a2c3	5.902
g#0g2c3d#3	5.393	e1d3g3b3	6.103
f#1d#2a2c3	5.601	f#1e2a#2d#3	6.329
g0f2b2d#3	5.818	d#1c#2f#2a#2	6.530
g1f2a#2c#3	6.035	g#0b2f3g#3	6.746
g1f2b2d#3	6.242	b0a2d#3g3	6.947

to better generalization performance as well as better generated results than by only considering local dependencies.

The design of our model is motivated by theories of musical rhythm Cooper and Meyer (1960) and music structure Lerdahl and Jackendoff (1983). A given musical note does not itself have a certain meaning. Its meaning, if any, is defined by the role it plays in longer musical elaborations such as melodies. To make an analogy to language, musical notes are perhaps more similar to *letters* than to *words*. However, the analogy is not entirely correct because even musical phrases do not have meaning in isolation in the same way that words do. A principal source of music structure is the meter of a piece. Almost all Western music is metered, indicating a fixed hierarchical temporal structure with small integer relationships between levels. We used meter to guide the construction of probabilistic trees, employing a binary tree structure suggested by the meter of the jazz standards in our database. Though this tree structure differs from that of other forms of music (thus representing a built-in stylistic prior motivated by music theory) the difference is not as great as it might seem. Most meters yield binary trees similar to the one we employ. Furthermore, if a tree is non-binary, then it is usually so only on a single level. For example, in a typical 3/4 piece of waltz music, the quarter-note level is indeed ternary (3 :1). However, the higher-level relationships remain binary, with musical phrases being formed out of 2, 4 or 8 measures.

Figure 2 shows a graphical model constructed as described above. Discrete nodes in levels 1 and 2 are not observed. The purpose of the nodes in level 1 is to capture global chord dependencies related to the meter. Nodes in level 2 are modeling local chord dependencies conditionally to the global dependencies captured in level 1. For instance, the fact that the algorithm is accurately generating proper endings is constrained by the upper tree structure. On the other hand, the smoothness of the voice leadings (e.g. small distances between generated notes in two successive chords) is modeled by the horizontal links in level 2.

The bottom nodes of the model are continuous observations conditioned by discrete hidden variables. Hence, Gaussian distributions can be used to model each observation given by the distributed representation described in Section 2.1. Suppose a Gaussian node G has a discrete parent D, then the

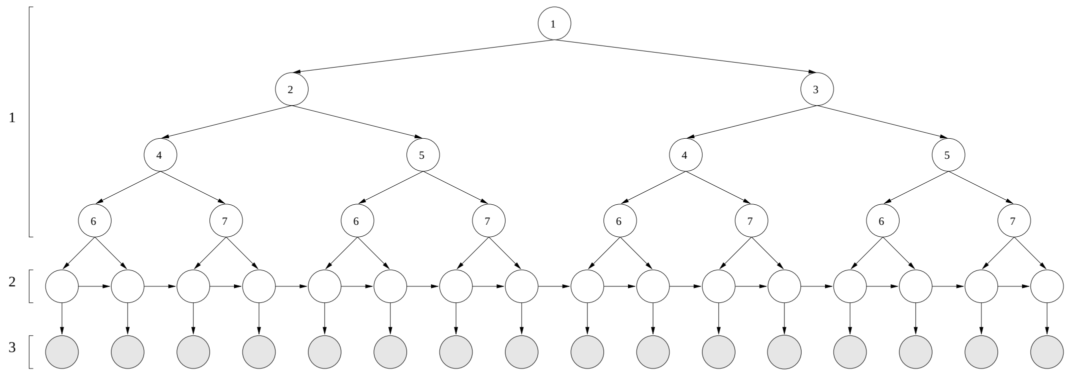


Fig. 2 – A probabilistic graphical model for chord progressions. White nodes correspond to discrete hidden variables while gray nodes correspond to observed multivariate Gaussian nodes. Nodes in level 1 directly model the contextual dependencies related to the meter. Nodes in level 2 combine this information with local dependencies in order to model smooth chord progressions. Finally, continuous nodes in level 3 are observing chords embedded in the continuous space defined by Equation (2). Numbers in level 1 nodes indicate a particular form of parameter sharing that is evaluated in Section 2.3.1.

conditional density $p(G|D)$ is given by

$$p(G|D = i) \sim \mathcal{N}(\mu_i, \sigma_i) \tag{3}$$

where $\mathcal{N}(\mu, \sigma)$ is a k-dimensional Gaussian distribution with mean $\mu \in \mathbb{R}^k$ and diagonal covariance matrix $\Sigma \in \mathbb{R}^k \times \mathbb{R}^k$ determined by its diagonal elements $\sigma \in \mathbb{R}^k$.

The Expectation-Maximization (EM) algorithm Dempster et al. (1977) can be used to estimate the conditional probabilities of the hidden variables in a graphical model. This algorithm proceeds in two steps applied iteratively over a dataset until convergence of the parameters. First, the E step computes the expectation of the hidden variables, given the current parameters of the model and the observations of the dataset. Secondly, the M step updates the values of the parameters in order to maximize the joint likelihood of the observations and the expected values of the hidden variables.

Marginalization must be carried out in the proposed model both for learning (during the expectation step of the EM algorithm) and for evaluation. The inference in a graphical model can be achieved using the Junction Tree Algorithm (JTA) Lauritzen (1996). In order to build the junction tree representation of the joint distribution of all the variables of the model, we start by moralizing the original graph (i.e. connecting the non-connected parents of a common child and then removing the directionality of all edges) so the independence properties in the original graph are preserved. In the next step (called triangulation), we add edges to remove all chord-less cycles of length greater than 4. Finally, we can form clusters with the maximal cliques of the triangulated graph. The junction tree representation is formed by joining these clusters together. To each cluster, we associate a potential function which can be normalized to give the marginalized probabilities of the variables in that cluster. Given evidence, the properties of the junction tree allow these potential functions to be updated by local message passing. Exact marginalization techniques are tractable in the proposed model given its limited complexity.

Many variations of the proposed graphical structure are possible, some of which are compared in Section 2.3. For instance, conditional probability tables can be tied in various ways. Also, more horizontal links in the model can be added to reinforce the dependencies between higher level hidden variables. The chord progressions are intimately tied to the metrical structure, which has obviously binary structure in the corpus of data. However, other tree structures may be more suitable for music having different meters (e.g. ternary structures for waltzes). Using a tree structure has the advantage of reducing the complexity of the considered dependencies from the order m to the order $\log m$, where m is the length of a given chord sequence. It should be pointed out that in this paper we only consider musical productions with fixed length. Fortunately, the current model could be easily extended to chords sequences with variable length by adding conditional dependencies arrows between many normalized subtrees.

Considering global dependencies to model time series is a general issue also present in other domains. For instance, tree models with structures derived from common syntactical patterns could be used to learn global dependencies in natural language processing applications. However, it should be noted that dependencies are much more complex in natural language than in chord progressions.

2.3 Experiments in the Continuous Space

52 jazz standard excerpts from Sher (1988) were interpreted and recorded by the first author in MIDI format on a Yamaha Disklavier piano. Standard 4-note jazz piano voicings as described in Levine (1990) were used to convert the chord symbols into musical notes. Thus, this particular model is considering chord progressions as they might be expressed by a trained jazz musician in a realistic musical context. The complexity of the chord sequences found in the corpus is representative of the complexity of common chord progressions in most jazz and pop music.

Every jazz standard excerpt was 8 bars long, with a 4 beats meter, and with one chord change every 2 beats (yielding observed sequences of length 16.) Longer chords were repeated multiple times (e.g. a 6 beats chord is represented as 3 distinct 2-beat observations.) This simplification has a limited impact on the quality of the model since generating a chord progression is simply a first (but very important) step toward generating complete polyphonic music, where modeling actual event lengths would be more crucial. The jazz standards were carefully chosen to exhibit a 16 bar global structure. We used the last 8 bars of each standard to train the model. Since every standard ends with a cadenza (i.e. a musical ending), the chosen excerpts exhibit strong regularities.

2.3.1 Generalization

The chosen discrete chord sequences were converted into sequences of 12-dimensional continuous vectors as described in Section 2.1. Frequencies ranging from 20Hz to 20kHz (MIDI notes going from the lowest note in the corpus to note number 135) were considered in order to build the representation given by Equation (1). A value of ρ of 0.96 was arbitrarily chosen for the experiments. It should be pointed out that since the generative models have been trained in an unsupervised setting, it is irrelevant to compare different chord representations (including the choice of ρ) in terms of likelihood. This problem will be addressed in Section 3. However, it is possible to measure how well a given architecture is modeling conditional dependencies between sub-sequences of chords. In order to do so, average negative conditional out-of-sample likelihoods of sub-sequences of length 4 on positions 1, 5, 9 and 13 have been computed. For each sequence of chords $\mathbf{x} = \{x_1, \ldots x_{16}\}$ in the appropriate validation set, we average the values

$$-\log P(x_i, \ldots, x_{i+3} | x_1, \ldots, x_{i-1}, x_{i+4}, \ldots, x_{16}). \tag{4}$$

with $i \in \{1, 5, 9, 13\}$. Hence, the likelihood of each subsequence is conditional on the rest of the sequence (taken in the validation set) from which it originates.

Double cross-validation is a recursive application of cross-validation Hastie et al. (2001) where both the optimization of the parameters of the model and the evaluation of the generalization of the model are carried out simultaneously. This technique has been used to optimize the number of possible values of hidden variables for various architectures and results are given in Table 2 in terms of average conditional negative out-of-sample log-likelihoods of sub-sequences.

This measure is similar to perplexity or prediction ability. We chose this particular measure of generalization in order to account for the binary metrical structure of chord progressions, which is not present in natural language processing, for instance.

TAB. 2 – Average conditional negative out-of-sample log-likelihoods of sub-sequences of length 4 on positions 1, 5, 9 and 13. These results are computed using double cross-validation in order to optimize the number of possible values for hidden variables. The numbers in parentheses indicate which levels of the tree are tied, as described in Figure 2. Since smaller values yield better prediction ability, we see that some combinations of parameter tying in the trees perform better than the standard HMM.

Model (tying)	Negative log-likelihood
Tree (2, 3)	93.8910
Tree (1, 3)	94.0037
Tree (1, 2, 3)	94.9309
Tree (3)	98.2446
HMM	98.2611

Different forms of parameter tying for the tree model shown in Figure 2 have been tested. All nodes in level 3 share the same parameters for all tested models. Hence, we used only one 12-dimensional Gaussian distributions (as in Equation (3)) independently of time, in order to constrain the capacity of the model. Moreover, a diagonal covariance matrix Σ has been used, thus reducing the number of free parameters to 24 in level 3 (12 for μ and 12 for Σ). Hidden variables in level 1 and 2 can be tied or not. Tying for level 1 is done as illustrated in Figure 2 by the numbers inside the nodes.

The fact that the contextual out-of-sample likelihoods presented in Table 2 are better for the different trees than for the HMM indicates that time-dependent regularities are present in the data. Sharing parameters in levels 1 or 2 of the tree increases the out-of-sample likelihood. This indicates that regularities are repeated over time in the signal. Further investigations would be necessary in order to assess to what extent chord structures are hierarchically related to the meter. On the other hand, the relatively high values obtained in terms of conditional out-of-sample negative log-likelihood indicates that the number of training sequences may not be sufficient to efficiently represent the variability of the data with *this* representation. The model is allowed to consider regions in the continuous space that could not be associated to any realistic chord, thus increasing perplexity. Hence, we propose in Sections 3 and 4 alternative chord representations where the variability of the data is more constrained with respect to musical knowledge.

2.3.2 Generation

One can sample the proposed model in order to generate novel chord progressions. Fortunately, Euclidean distances are relevant in the observation space created in Section 2.1. Thus, a simple approach to generate chord progressions is to take the nearest neighbors (nearest chords in the training set) of each sampled values obtained by sampling the observation nodes.

Chord progressions generated by the models presented in this paper are available at `http ://www.idiap.ch/∼paie` For instance, Figure 2.3.2 shows a chord progression that has been generated by the graphical model shown in Figure 2. This chord progression has all the characteristics of a standard jazz chord progression. For instance, the trained musician can observe that the last 8 bars of the sequence is a II-V-I[4] chord progression Levine (1990), which is very common. Figure 4 shows a chord progression generated by the HMM model. While the chords are following each other in a smooth fashion, there is no global relation between chords. For instance, one can see that the lowest note of the last chord is not a `c`, which was the case for all the chord sequences in the training set. The fundamental qualitative difference between both methods should be obvious even for the non-musician when listening to the generated chord sequences.

[4]The lowest notes are `d`, `g` and `c`.

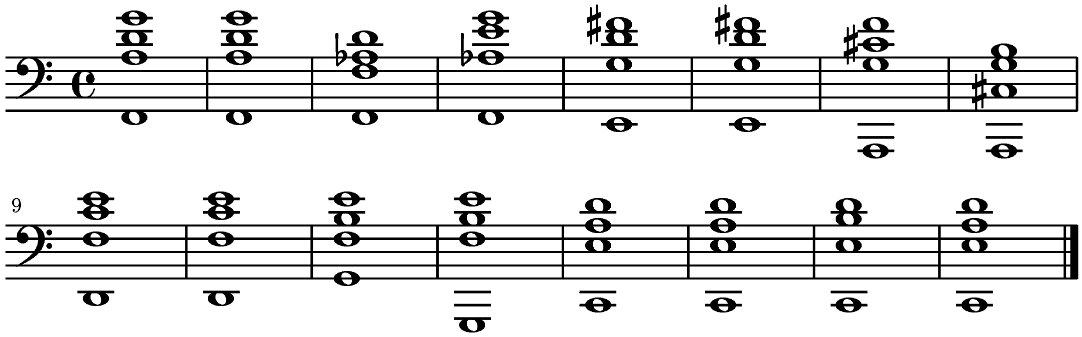


FIG. 3 – A chord progression generated by the proposed model. This chord progression is very similar to a standard jazz chord progression.

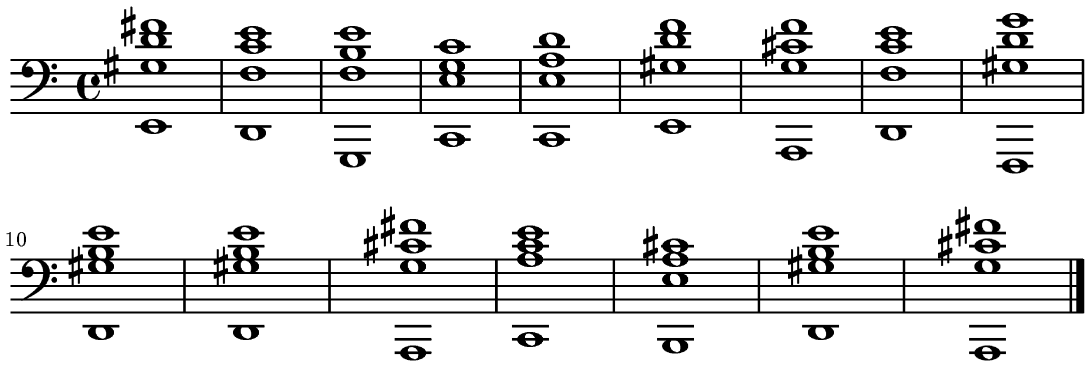


FIG. 4 – A chord progression generated by the HMM model. While the individual chord transitions are smooth and likely, there is no global chord structure.

3 Probabilities of Substitution

Although it provides a very intuitive and appealing representation for chords, the representation for chords introduced in the previous section suffer from two major drawbacks. As already pointed out in section 2.3.1, this representation allows the model to consider regions where no realistic chord is present. In fact, it is unnatural to compress discrete information in a continuous space ; one could easily think of a one-dimensional continuous representation that would overfit any discrete dataset. Second, there is no direct way to represent Euclidean distances between discrete objects in the graphical model framework. Since the set of likely chords is finite, one may prefer to observe directly discrete variables with a finite number of possible states.

Our proposed solution to these problems is to convert the Euclidean distances between chord representations into probabilities of substitution between chords. Chords can then be represented as individual discrete events. These probabilities can be included in a graphical model without relying on extra techniques such as finding the nearest neighbors during generation (see Section 2.3.2). It is interesting to note that the problem of considering similarities between discrete objects in statistical models is not restricted to music and encompasses a large span of applications, including natural language processing and biology.

One can define the probability $p_{i,j}$ of substituting chord $\mathcal{X}_i$ for chord $\mathcal{X}_j$ in a chord progression as

$$p_{i,j} = \frac{\phi_{i,j}}{\sum_{1 \leq j \leq s} \phi_{i,j}} \tag{5}$$

with

$$\phi_{i,j} = \exp\{-\lambda ||\mathbf{v}_i - \mathbf{v}_j||^2\} \tag{6}$$

with free parameter $0 \leq \lambda < \infty$ and $\mathbf{v}_k$ components being defined in Equation (2).

It is interesting to note that it was impossible in Section 2 to optimize the parameter ρ using cross-validation because this parameter was defining the observed representation over which likelihood was evaluated. On the contrary, the parameters λ and ρ can be optimized by validation on any chord progression dataset provided a suitable objective function, since the chord representation will be independent of their values. With possible values going from 0 to arbitrary high values, the parameter λ allows the substitution probability table to go from the uniform distribution with equal entries everywhere (such that every chord has the same probability of being played) to the identity matrix (which disallow any chord substitution). Table 3 shows substitution probabilities obtained from Equation (5) for chords in Table 1.

3.1 Graphical Model Using Probabilities of Substitution

We now propose a graphical model for chord sequences using the probabilities of substitution between chords described in the previous section. Again, the main assumption behind the proposed model is that conditional dependencies between chords in a typical chord progression are tied to the metrical structure associated with it. We show empirically in Section 3.2 that such tree structure leads again to better generalization performance as well as better generated results than by only considering local dependencies with an HMM model, like it was the case in Section 2.3.1.

Figure 5 shows a graphical model that can be used as a generative model for chord progressions in this fashion. All the random variables in the model are discrete. Nodes in level 1, 2 and 3 are hidden while nodes in level 4 are observed. Every chords are represented as distinct discrete events. Nodes in level 1 directly model the contextual dependencies related to the meter. Nodes in level 2 combine this information with local dependencies in order to model smooth chord progressions. Variables in level 1 and 2 have an arbitrary number of possible states optimized by cross-validation Hastie et al. (2001).

Variables in levels 3 and 4 have a number of possible states equal to the number of chords in the dataset. Hence, each state is associated with a particular chord. The probability table associated with the conditional dependencies going from level 3 to 4 is fixed during learning with the values given by

TAB. 3 – Subset of the substitution probability table constructed with Equation (5). For each column, the number in the first row corresponds to the probability of playing the associated chord with no substitution. The numbers in the following rows correspond to the probability of playing the associated chord instead of the chord in the first row of the same column.

c1a2e3g3	0.41395	c1d#2a#2d3	0.70621
c1a2c3e3	0.08366	c1a#2d#3g3	0.06677
c1a2d3g3	0.06401	c1e2a#2d#3	0.02044
c1a1d2g2	0.02195	c1a#2e3g#3	0.00805
c1a#2e3a3	0.01623	c1e2a#2d3	0.00582
a0c3g3b3	0.00929	a#0d#2g#2c3	0.00431
c1e2b2d3	0.00679	a#0d2g#2c3	0.00318
c1g2c3e3	0.00500	g#1g2c3d#3	0.00243
a0g#2c3e3	0.00363	c1e2a#2c#3	0.00176
c1f2c3e3	0.00255	a#1g#2d3g3	0.00134
c1d#2a#2d3	0.00156	f1a2d#3g3	0.00102
e1e2g2c3	0.00112	d1f#2c3f3	0.00075
g1a#2f3a3	0.00085	a0c3g3b3	0.00057
e0g2d3f#3	0.00065	g1f2a#2c#3	0.00043
f#0e2a2c3	0.00049	b0d2a2c3	0.00033
g#0g2c3d#3	0.00037	e1d3g3b3	0.00025
f#1d#2a2c3	0.00028	f#1e2a#2d#3	0.00019
g0f2b2d#3	0.00021	d#1c#2f#2a#2	0.00015
g1f2a#2c#3	0.00016	g#0b2f3g#3	0.00011
g1f2b2d#3	0.00012	b0a2d#3g3	0.00008

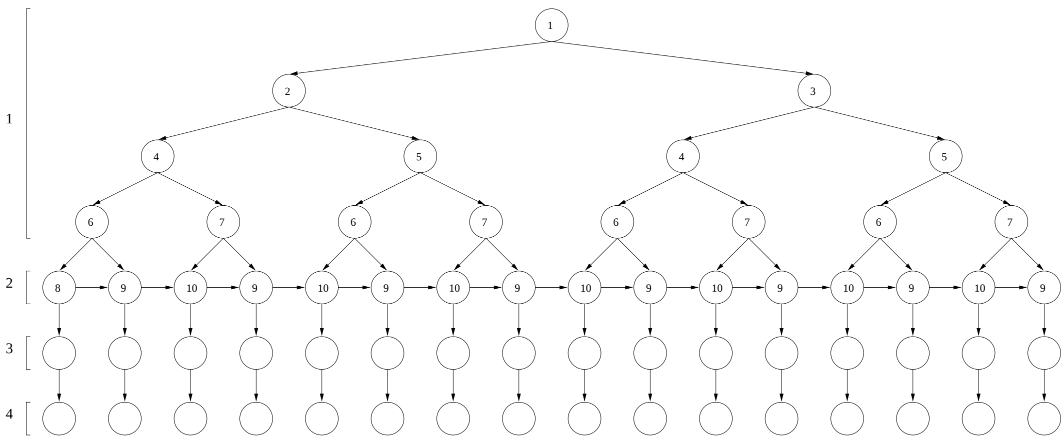


FIG. 5 – A probabilistic graphical model for chord progressions, as described in Section 3.1. Numbers in level 1 and 2 nodes indicate a particular form of parameter sharing that has been used in the experiments (see Section 2.3.1).

Tab. 4 – Average negative conditional out-of-sample log-likelihoods of sub-sequences of length 8 on positions 1, 9, 17 and 25, given the rest of the sequences. These results are computed using double cross-validation in order to optimize the number of possible values for hidden variables and the parameters λ and ρ. We see that the trees perform better than the HMM.

Model	(Tying in level 1)	Negative log-likelihood
Tree	No	32.3281
Tree	Yes	32.6364
HMM		33.2527

Equation (5). Values in level 3 are hidden and represent intuitively "initial" chords that could have been substituted by the actual observed chords in level 4.

The role of the fixed substitution matrix is to raise the probability of unseen events in a way that accounts for psychoacoustical similarities. Discarding level 4 and directly observing nodes in level 3 would assign extremely low probabilities to unseen chords in the training set. Instead, when observing a given chord on level 4 during learning, the probabilities of *every* chords of the dataset are updated with respect to the probabilities of substitution described in the previous section.

Again, the Junction Tree Algorithm (JTA) is used for marginalization and the EM algorithm for parameter learning. Many variations of this particular model are possible, some of which are compared in the following section.

3.2 Experiments with the Probabilities of Substitution

The same database as in Section 2.3 was used for the experiments. Every jazz standard excerpt was 16 bars long, with a 4 beat meter, and with one chord change every 2 beats (yielding observed sequences of length 32). The chosen discrete chord sequences were converted into sequences of 12-dimensional continuous vectors as described in Section 2.1. In order to measure how well a given architecture captures conditional dependencies between sub-sequences, average negative *conditional* out-of-sample likelihoods of sub-sequences of length 8 on positions 1, 9, 17 and 25 have been computed (see Equation (4)). Double cross-validation has been used to optimize the number of possible values of hidden variables and the parameters ρ and λ for various architectures. Results are given in Table 4.

Two forms of parameter tying for the tree model have been tested. The conditional probability tables in level 1 of Figure 5 can be either tied as shown by the numbers inside the nodes in the figure or can be left untied. Tying for level 2 is always done as illustrated in Figure 5 by the numbers inside the nodes, to model local dependencies. All nodes in level 3 share the same parameters for all tested models. Also, recall that parameters for the conditional probabilities of variables in level 4 are fixed as described in Section 3.1.

As a benchmark, an HMM consisting of levels 2, 3 and 4 of Figure 5 has been trained and evaluated on the same dataset. The results presented in Table 4 are similar to perplexity or prediction ability. As in Section 2.3, the fact that these contextual out-of-sample likelihoods are better for the trees than for the HMM are an indication that time-dependent regularities are present in the data. Further investigations would be necessary in order to assess to what extent chord structures are hierarchically related to the meter.

It should be pointed out that the results obtained in Table 2 and in Table 4 can not be compared quantitatively to assess the generalization capabilities of one model compared to the other. These results can only be used to compare the prediction ability of one model versus one another over the *same* chord representation. In order to compare both chord representations quantitatively, a supervised task with an appropriate objective function (e.g. transcription, melody extraction, genre recognition) could be designed.

One can sample the joint distribution learned by the model presented in this section in order to generate novel chord progressions. Like in Section 2.3.2, we observe that chord progressions

Tab. 5 – This table illustrates a way to construct a vector assessing the relative importance of each time-step in a 4-beat measure divided in 12 time-steps. On each row, we add positions that have less perceptual importance than the previous added ones, ending with a weight vector covering all the possible time-steps.

Beat	1	.	.	2	.	.	3	.	.	4	.	.
	1											
	2						1					
	3			1			2			1		
	4		1	2		1	3		1	2		1
	5	1	2	3	1	2	4	1	2	3	1	2

generated by the tree model have all the characteristics of standard jazz chord progression (see `http ://www.idiap.ch/~paiement/ml`), which is not the case for chord progressions generated with an HMM.

4 Interactions Between Chords and Melodies

After having considered chord progressions by themselves, a further step towards full modelling of tonal polyphonic music is to model the interaction between chord progressions and melodies. A chord representation that tells directly which notes are present in a given chord appears to be well suited for this task. Every notes in a chord have a particular impact on the chosen notes of a melody and a proper polyphonic model should be able to capture these interactions. Also, including domain knowledge (e.g. A major third is not likely to be played when a diminished fifth is present) would be much easier to include in a model dealing directly with the notes comprising a chord. While such a model is inevitably much more tied to a particular music style, it is also able to achieve more complex tasks like melodic accompaniment.

4.1 Melodic Representation

A simple way to represent a melody is to convert it to a 12-dimensional continuous vector representing the relative importance of each pitch class over a given period of time t. We first observe that the lengths of the notes comprising a melody have an impact on their perceptual emphasis. Usually, the meter of a piece can be subdivided into small time-steps such that the beginning of any note in the whole piece will approximately occur on one of these time-steps. For instance, let t be the time required to play a whole measure. Given that a 4-beat piece (where each beat has a quarter note length) contains only eight notes or longer notes, we could divide every measure into 8 time-steps with length $t/8$ and every notes of the piece would occur approximately on the onset of one of these time-steps occurring at times $0, t/8, 2t/8, \ldots, 7t/8$. We can assign to each pitch-class a perceptual weight equal to the total number of such time-steps it covers during time t.

However, it turns out that the perceptual emphasis of a melody note depends also on its position related to the meter of the piece. For instance, in a 4 beats measure, the first beat (also called the downbeat) is the beat where the notes played have the greatest impact on harmony. The second most important one is the third beat. We illustrate in Table 5 a way of constructing a weight vector assessing the relative importance of each time-step in a 4 beats measure divided in 12, relying on the theory of meter Cooper and Meyer (1960), as described in Section 1. At each step represented by a row in the table, we consider one or more positions that have less perceptual emphasis than the previous added ones and increment all the values by one. The resulting vector on the last row accounts for the perceptual emphasis that we apply to each time-step in the measure.

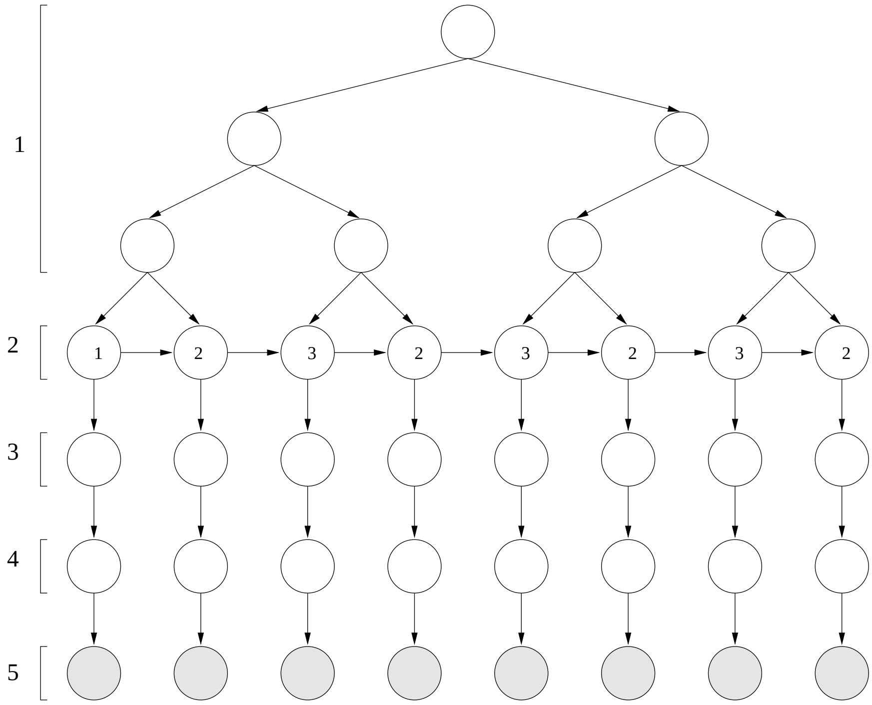


FIG. 6 – A graphical model to predict root progressions given melodies.

Although this method is based on widely accepted musicological concepts, more research would be needed to asses its statistical reliability and to find optimal weighting factors.

4.2 Modelling Root Progressions

One of the most important notes in a chord with regard to its interaction with the melody may be the root[5]. For example, bass players are playing the root note of the current chord very often when accompanying other musicians in a jazz context. Figure 6 shows a model that learns interactions between root notes (or chord names) and the melody. Such a model is able to predict sequences of root given a melody, which is a non-trivial task even for humans. Nodes in level 1 and 2 are discrete hidden variables and play the same role than in previous models. Nodes in level 2 are tied according to the numbers shown inside the vertices. Probabilities of transition between levels 3 and 4 are fixed by Equation (5) using single notes instead of chords and have 12 possible states corresponding to each possible root note. We thus model the probability of substituting one root for one another. Hence, nodes in level 3 are hidden while nodes in level 4 are observed. This part of the model is again necessary to redistribute efficiently probability mass to unseen events during training. Nodes in level 5 are continuous 12-dimensional Gaussian distributions as defined in Equation (3). Nodes in level 5 are also observed during training where we model each melodic observation using the technique presented in Section 4.1.

4.2.1 Evaluation of Root Prediction Given Melody

In order to evaluate the model presented in Figure 6, a database consisting of 47 standard jazz melodies in MIDI format and their corresponding root progressions taken in Sher (1988) has been compiled by the authors. Every sequence was 8 bar long, with a 4 beat meter, and with one chord

[5]The root note of a chord is the note that gives its name to the chord. For instance, the root of the chord Em7b5 is the note E.

Tab. 6 – Average conditional negative out-of-sample log-likelihoods of sub-sequences of roots of length 4 on positions 1, 5, 9 and 13 *given* melodies. These results are computed using double cross-validation in order to optimize the number of possible values for hidden variables. Again, the results are better for the tree model than for the HMM.

Model	Negative log-likelihood
Tree	6.6707
HMM	8.4587

Tab. 7 – Interpretation of the possible states of the structural random variables. For instance, the variable associated to the 5th of the chord can have 3 possible states. State 1 corresponds to the perfect fifth (P), state 2 to the diminished fifth and state 3 to the augmented fifth.

	Values			
Component	1	2	3	4
3rd	M	m	sus	-
5th	P	b	#	-
7th	no	M	m	M6
9th	no	M	b	#
11th	no	#	P	-
13th	no	M	-	-

change every 2 beats (yielding observed sequences of length 16). It was required to divide each measure into 24 time-steps in order to fit each melody note to an onset. The technique presented in Section 4.1 was used over a time span t of 2 beats corresponding to the chords lengths.

The proposed tree model was compared to an HMM (builded by removing nodes in level 1) in terms of prediction ability *given* the melody. We always observe melody vectors in level 5 while we try to predict subsequences of roots in level 4. As in Section 2.3.1, average conditional negative out-of-sample likelihood of sub-sequences of roots of length 4 on positions 1, 5, 9 and 13 were computed and results are presented in Table 6.

Generated root sequences given out-of-sample melodies are presented in Section 4.4.1 together with generated chord structures.

4.3 Discrete Chord Model

Before describing a complete model to learn the interactions between complete chords and melodies, we introduce in this section a chord representation that allows to model dependencies between each chord component and the proper pitch-class components in the melodic representation presented in Section 4.1.

The model that we present in this section is observing chord symbols as they appear in Sher (1988) instead of actual *instantiated* chords (i.e. observing directly musical notes derived from the chord notation by a real musician) as in Sections 2 and 3. This simplification has the advantage of defining directly the chord components as they are conceptualized by a musician. This way, it will be easier in further developments of this model to experiment with more constraints (in the form of independence assumptions between random variables) derived from musical knowledge. However, it would also be possible to infer the chord symbols from the actual notes with a deterministic method, which is done by most of the MIDI sequencers today. Hence, a model observing chord symbols instead of actual notes could still be used over traditional MIDI data.

Each chord is represented by a root component (which can have 12 possible values given by the pitch-class of the root of the chord) and 6 structural components detailed in Table 7.

Tab. 8 – Mappings from some chord symbols to structural vectors according to notation described in Table 7.

Symbol	3rd	5th	7th	9th	11th	13th
6	1	1	4	1	1	1
M7	1	1	2	1	1	1
m7b5	2	2	3	1	1	1
7b9	1	1	3	3	1	1
m7	2	1	3	1	1	1
7	1	1	3	1	1	1
9#11	1	1	3	2	2	1
m9	2	1	3	2	1	1
13	1	1	3	2	1	2
m6	2	1	4	1	1	1
9	1	1	3	2	1	1
dim7	2	2	4	1	1	1
m	2	1	1	1	1	1
7#5	1	3	3	1	1	1
9#5	1	3	3	2	1	1

While it is out of the scope of this paper to describe jazz chord notation in detail Levine (1990), we just note that there exists a one-to-one relation between the chord representation introduced in Table 7 and chord symbols as they appear in Sher (1988). We show in Table 8 the mappings of some chord symbols to structural vectors according to this representation.

For instance, the chord with symbol 7#5 has a major third, an augmented fifth, a minor seventh, no ninth, no eleventh and no thirteenth. The fact that each structural random variable has a limited number of possible states will produce a model that is computationally tractable. While such a representation may look less general for a non-musician, we believe that it is applicable to most of tonal music by introducing proper chord symbol mappings. Moreover, it allows to directly model the dependencies between chord components and melodic components.

4.4 Chord Model given Root Progression and Melody

Figure 7 shows a probabilistic model designed to predict chord progressions *given* root progressions and melodies. The nodes in level 1 are discrete hidden nodes as in previous models. The gray boxes are subgraphs that are detailed in Figure 8. The H node is a discrete hidden node modelling local dependencies and corresponding to the nodes on level 2 in Figure 2. The R node corresponds to the current root. This node can have 12 different states corresponding to the pitch class of the root and it is always observed. Nodes labelled from 3rd to 13th correspond to the structural chord components presented in Section 4.3. Node B is another structural component corresponding to the bass notation (e.g. G7/D is a G seventh chord with a D on the bass). This random variable can have 12 possible states defining the bass note of the chord. All the structural components are observed during training to learn their interaction with root progressions and melodies. These are the random variables we try to predict when using the model on out-of-sample data. The nodes on the last row labelled from 0 to 11 correspond to the melodic representation introduced in Section 4.1.

It should be noted that the melodic components are observed *relative* to the current root. In Section 4.2, the model is observing melodies with absolute pitch, such that component 0 is associated to note C, component 1 to note C#, and so on. On the other hand, in the present model component 0 is associated to the root note defined by node R. For instance, if the current root is G, component 0 will be associated to G, component 1 to G#, component 2 to A, and so on. This approach is necessary

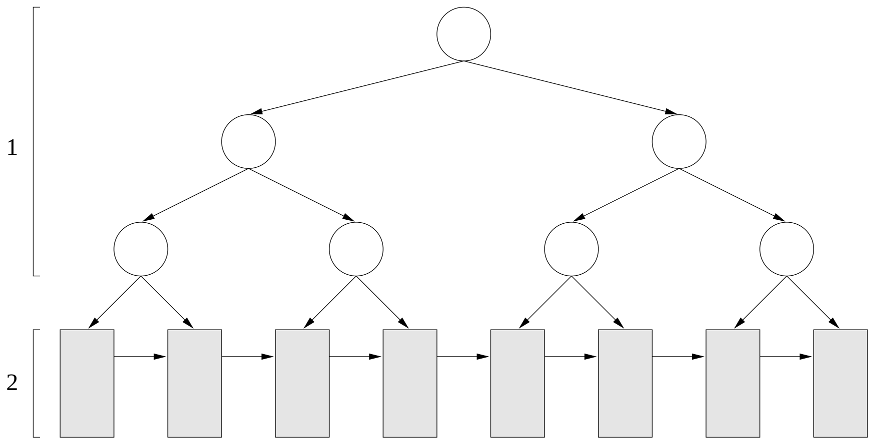


FIG. 7 – A graphical model to predict chord progressions given root progressions and melodies. The gray boxes correspond to subgraphs presented in Figure 8.

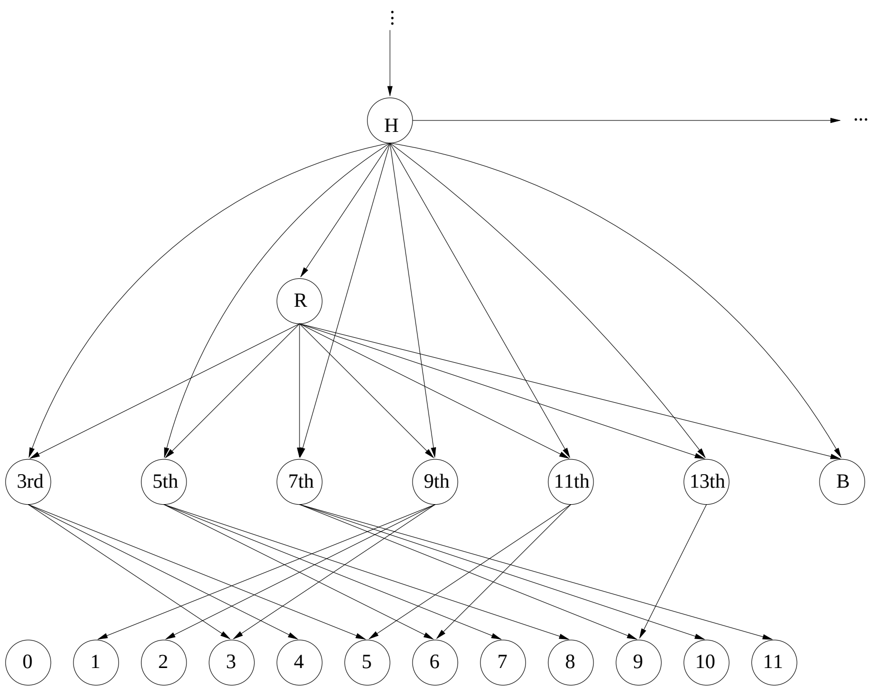


FIG. 8 – Subgraph of the graph presented in Figure 7. Each chord component is linked with the proper melodic components on the bottom.

TAB. 9 – Average negative conditional out-of-sample log-likelihoods of sub-sequences of chord structures of length 4 on positions 1, 5, 9 and 13, given the rest of the sequences and the complete root progressions and melodies using double cross-validation.

Model	Negative log-likelihood
Tree	9.9197
HMM	9.5889

TAB. 10 – Three different harmonizations of the last 8 measures of the jazz standard *Blame It On My Youth*. Rows beginning with OC correspond to the original chord progression. Rows beginning with OR correspond to the most likely chord structures given the original root progression and melody with respect to the model presented in Section 4.4.1. Finally, rows beginning with NH correspond to a new harmonization generated by the same model and the root progression model presented in Section 4.2 when observing original melody only.

OC (1-8)	AbM7	Bb7	Gm7	Cm7	Fm7	Fm7/Eb	Db9#11	C7
OR	AbM7	Bb7	Gm7	C7	Fm7	Fm7	Db7	Cm7
NH	C7	C7	Gm7	Gm7	Fm7	Fm7	Bb7	Bb7
OC (9-16)	Fm7	Edim7	Fm7	Bb7	Eb6	Eb6	Eb6	Eb6
OR	Fm7	E9	Fm7	Bb7	Eb6	Eb6	Eb6	Eb6
NH	Edim7	Gm7	Fm7	Bb7	Eb6	Eb6	Eb6	Eb6

to correctly link the structural components to the proper melodic components as shown by the arrows between the two last rows of nodes on Figure 8.

4.4.1 Generation of an harmonization

It is possible to evaluate the prediction ability of the model for chord structures. We present in Table 9 the average negative conditional out-of-sample log-likelihoods of chord structures of length 4 on positions 1, 5, 9 and 13, given the rest of the sequences, the complete root progressions and the melodies for the tree model and an HMM model builded by removing the nodes in level 1 in Figure 7.

Again, we used double cross-validation in order to optimize the number of hidden variables in the models. We observe that the HMM gives better results than the tree model in this case. This can be explained by the fact that the root progressions are given in these experiments. This would mean that most of the contextual information would be contained in the root progression, which make sense intuitively. Further statistical experiments could be done to investigate this behavior.

The process of generating a chord progression given a melody is called *harmonization*. The proposed model can be used to generate an harmonization over an unknown melody with respect to the dependencies learned during training. Table 10 shows three different harmonizations of the last 8 measures of the jazz standard *Blame It On My Youth*, by O. Levant.

When observing the predicted structures given the original root progression, we see that most of the predicted chords are the same as the originals. When the chord differs, the musician will observe that the predicted chords are still relevant and are not in conflict with the melody shown in Figure 9. It is more interesting to look at the sequence of chords generated by taking the sequence of root with the highest probability given by the root progression model presented in Section 4.2 and then finding the most likely chord structures given this predicted root progression and the original melody. While some chord change are debatable, most of the chords comply with the melody and we think that the final result is musically interesting. These results show that valid harmonization models for melodies that could learn different musical styles could be implemented in commercial software in the

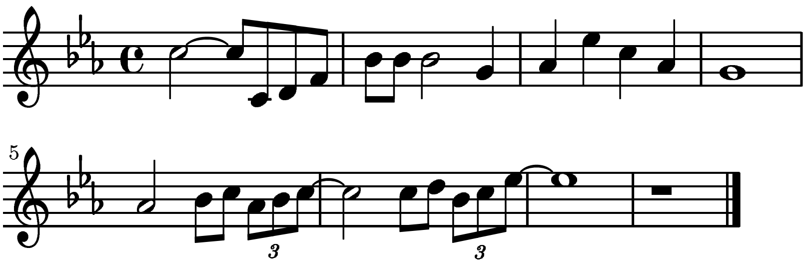


Fig. 9 – The last 8 measures of the melody of the jazz standard *Blame It On My Youth*.

short term. Again, more generated results from the models presented in this paper are available on `http ://www.idiap.ch/∼paiement/ml`.

5 Conclusion

In this paper, we introduced three different representations for chords that can be used as observations for probabilistic models. Each of these representations are adequate for different purposes. While the continuous representation presented in Section 2 is very intuitive and has a musicological interest, the derived probabilities of substitution presented in Section 3 seems more suited to the graphical model framework. Although these two representations are more general and computationally simple, we believe that a representation closer to the domain of application like the one presented in Section 4 is also very appealing. First, it is much easier to introduce domain knowledge (such as the relations between components of chords) in a model observing such a representation. Moreover, it is also easier to model statistical independence relationship (i.e. the absence of link between two random variables) between chord components and other polyphonic music components when modelling directly each chord component.

As already noted, the generalization capabilities of the different chord representations presented in this paper should not be compared in terms of prediction ability. Supervised tasks where these representations could play a major role such as chord recognition in audio signal, audio to symbolic transcription or genre recognition could be designed in future work to compare quantitatively these representations.

A second main contribution of our work is that we have shown empirically that chord progressions exhibit global dependencies that can be better captured with a tree structure related to the meter than with a simple dynamical HMM that concentrates on local dependencies. The importance of contextual information for modeling chord progressions is even more apparent when one compares sequences of chords sampled from both models. The time-dependent hidden variables enable the tree structure to generate coherent chord progressions both locally and globally. We have also shown that it is possible to design a probabilistic model that can generate proper chord progressions given other polyphonic music components such as melody or even chord root progressions.

Chord progressions are regular and simple structures that condition dramatically the actual choice of notes in polyphonic tonal music. Hence, we argue that chord models are crucial in the design of efficient algorithms that deal with such music data. Moreover, generating interesting chord progressions may be one of the most important aspects in generating realistic polyphonic music. Our proposed chord representations and their associated probabilistic models constitute a first step in that direction.

Acknowledgements

The first author would like to thank Yves Grandvalet and David Barber for helpful discussions. This work was supported in part by the IST Program of the European Community, under the PASCAL Network of Excellence, IST-2002-506778, funded in part by the Swiss Federal Office for Education and Science (OFES) and the Swiss NSF through the NCCR on IM2.

Références

Allan, M. and Williams, C. K. I. (2004). Harmonising chorales by probabilistic inference. In *Advances in Neural Information Processing Systems*, volume 17.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2) :157–166.

Cemgil, A. T. (2004). *Bayesian Music Transcription.* PhD thesis, Radboud University of Nijmegen.

Cooper, G. and Meyer, L. B. (1960). *The Rhythmic Structure of Music.* The Univ. of Chicago Press.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39 :1–38.

Eck, D. and Schmidhuber, J. (2002). Finding temporal structure in music : Blues improvisation with LSTM recurrent networks. In Bourlard, H., editor, *Neural Networks for Signal Processing XII, Proc. 2002 IEEE Workshop*, pages 747–756, New York. IEEE.

Handel, S. (1993). *Listening : An introduction to the perception of auditory events.* MIT Press, Cambridge, Mass.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning.* Springer series in statistics. Springer-Verlag.

Kuusi, T. (2001). *Set-Class and Chord : Examining Connection Between Theoretical Ressemblance and Perceived Closeness.* Number 12 in Studia Musica. Sibelius Academy.

Lauritzen, S. L. (1996). *Graphical Models.* Oxford University Press.

Lavrenko, V. and Pickens, J. (2003). Polyphonic music modeling with random fields. In *Proceedings of ACM Multimedia*, Berkeley, CA.

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music.* MIT Press, Cambridge, Mass.

Levine, M. (1990). *The Jazz Piano Book.* Sher Music Co./Advance Music.

Moore, B. (1982). *An Introduction to the Psychology of Hearing.* Academic Press.

Raphael, C. and Stoddard, J. (2003). Harmonic analysis with probabilistic graphical models. In *Proceedings of ISMIR 2003.*

Sher, C., editor (1988). *The New Real Book*, volume 1. Sher Music Co.

Valimaki, V., Huopaniemi, J., Karjaleinen, and Janosy, Z. (1996). Physical modeling of plucked string instruments with application to real-time sound synthesis. *J. Audio Eng. Society*, 44(5) :331–353.

Vassilakis, P. (1999). Chords as spectra, harmony as timbre. *J. Acoust. Soc. Am.*, 106(4/2) :2286. (paper presented at the 138th meeting of the Acoustical Society of America).