Probabilistic Melodic Harmonization

Jean-François Paiement[1], Douglas Eck[2], and Samy Bengio[1]

[1] IDIAP Research Institute, Rue du Simplon 4, Case Postale 592, CH-1920, Martigny, Switzerland.
paiement@idiap.ch
bengio@idiap.ch
[2] Département d'Informatique et de recherche opérationnelle, Université de Montréal, Pavillon André-Aisenstadt, CP 6128, succ Centre-Ville, Montréal, QC, H3C 3J7, Canada.
eckdoug@iro.umontreal.ca

Abstract. We propose a representation for musical chords that allows us to include domain knowledge in probabilistic models. We then introduce a graphical model for harmonization of melodies that considers every structural components in chord notation. We show empirically that root notes 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. However, a local model seems to be sufficient for generating proper harmonizations when root notes progressions are provided. The trained probabilistic models can be sampled to generate very interesting chord progressions given other polyphonic music components such as melody or root note progressions.

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 [1]. This difficulty motivates our exploration of chord progressions and their interaction with melodies. In its simplest definition, a chord is a group of note names. Chord progressions constitute a fixed, non-dynamical structure in time and thus can be used to aid in describing long-term musical structure in tonal music. A harmonization is a particular choice of chord progression given other components of tonal music (e.g. melodies or bass lines). In this paper, we propose a graphical model to generate harmonizations given melodies based on training data. In general, the notes comprising a chord progression are not played directly. Instead, given that a particular temporal region in a musical piece is associated with a chord, notes comprising that chord or sharing some harmonics with notes of that chord are more likely to be present.

Graphical 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[3] data. One advantage of graphical models is their flexibility, suggesting that our models could be used either as an analytical or a generative tool to model chord progressions. Moreover, models like ours could be integrated into more complex probabilistic transcription models [2], genre classifiers, or automatic composition systems [3].

Cemgil [2] 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 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 [4] use graphical models for labeling MIDI data with traditional Western chord symbols. Lavrenko and Pickens [5] propose a generative model of polyphonic music that employs Markov random fields. While being very general, this model would benefit from having access to more specific musical knowledge. For instance, we go a step further in this paper by including abstract chord representation in the model[4] as a smoothing technique towards better generalization. Allan and Williams [8] designed a harmonization model for Bach chorales using Hidden Markov Models (HMMs). While generating excellent musical results, this model has to be provided polyphonic music with specific 4 voice structure as input, restricting its applicability in more general settings. Our proposed model is more general in the sense that it is possible to extract the appropriate chord representation from any polyphonic music, without regard to specific labeling or harmonic structure. One can then use it to generate harmonization given any melody without regard to the musical style of the corpus of data at hand.

2 Graphical Models

Graphical models [9] 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. 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

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

[4] The proposed model is defined using standard jazz chord notation as described in [6, 7].

for marginalization and various learning algorithms can be used to learn the parameters of a model, given an appropriate dataset.

The Expectation-Maximization (EM) algorithm [10] 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) [9]. 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 that some of 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 4 or more. Then, we can form clusters with the maximal cliques of the triangulated graph. The Junction Tree representation is formed by joining these clusters together. We finally apply a message passing scheme between the potential functions associated to each cluster of the Junction Tree. These potential function 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. Exact marginalization techniques are tractable in the proposed model given its limited complexity.

3 Interactions Between Chords and Melodies

Each note in a chord has 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 in a model dealing directly with the notes comprising a chord. While such a model is somewhat tied to a particular musical style, it is also able to achieve complex tasks like melodic accompaniment.

3.1 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 is a quarter note in 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-beat 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 1 a way of constructing a weight vector assessing the relative importance of each time-step in a 4-beat measure divided into 12 time-steps with swing eight notes, relying on the theory of meter [11]. 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.

Table 1. This table illustrates a way to construct a vector assessing the relative importance of each time-step in a 4-beat measure divided into 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	.	.
	.											
	.						.					
	.			.			.			.		
	.		.	.		.	.		.	.		.
	.	.	.	.	.	.	.	.	.	.	.	.
	5	1	2	3	1	2	4	1	2	3	1	2

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

3.2 Modeling Root Note Progressions

One of the most important notes in a chord with regard to its interaction with the melody may be the root note[5]. For example, bass players play the root note

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

of the current chord very often when accompanying other musicians in a jazz context. Figure 1 shows a model that learns interactions between root notes (or chord names) and the melody.

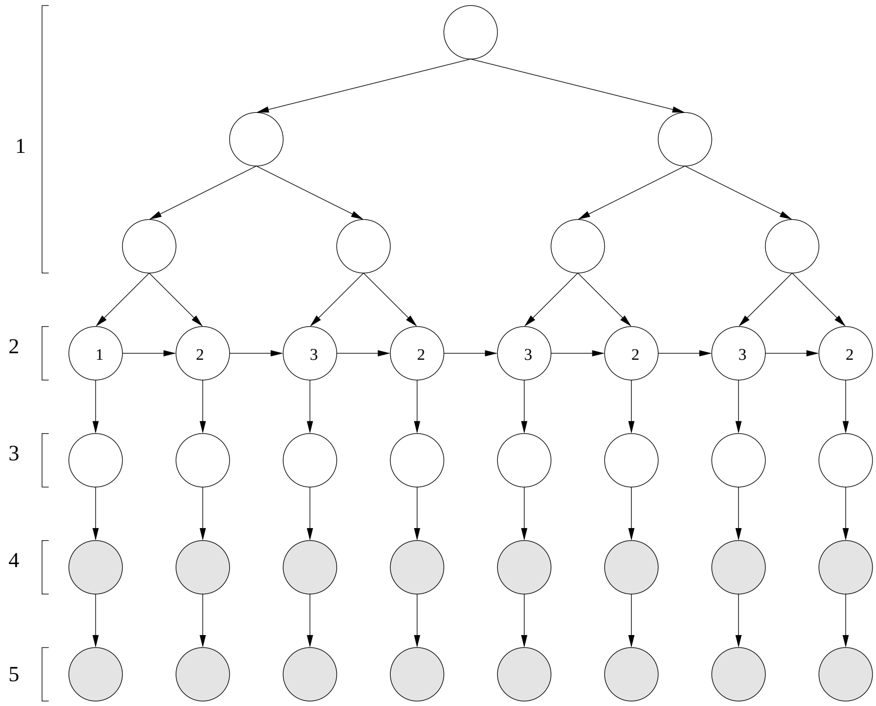

Fig. 1. A graphical model to predict root note progressions given melodies. White nodes are hidden random variables while gray nodes are observed

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 [11, 12]. 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.

Such a model is able to predict sequences of root notes 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 with probabilities of substitution related to psychoacoustic similarities between notes [13]. These random variables have 12 possible states corresponding to each possible root note. We thus model the probability of substituting one root note for one another. Nodes in level 3 are hidden while nodes in level 4 are observed. Discarding level 4 and directly observing nodes in level 3 would assign extremely low probabilities to unseen root notes in the training set. Instead, when observing a given chord on level 4 during learning, the probabilities of *every* root notes are updated with respect to the fixed probabilities of substitution. Nodes in level 5 are continuous 12-dimensional Gaussian distributions that are also observed during training where we model each melodic observation using the technique presented in Section 3.1.

Evaluation of Root Notes Prediction Given Melody In order to evaluate the model presented in Figure 1, a database consisting of 47 standard jazz melodies in MIDI format and their corresponding root note progressions taken in [6] has been compiled by the authors. Every sequence was 8 bar long, with a 4-beat meter, and with one chord 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 3.1 was used over a time span t of 2 beats corresponding to the chords lengths.

The proposed tree model was compared to an HMM (built by removing nodes in level 1) in terms of prediction ability *given* the melody. 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

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

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 and the corresponding melody.

Double cross-validation is a recursive application of cross-validation [14] where both the optimization of the parameters of the model and the evaluation of the generalization of the model are carried out simultaneously. We let the number of possible states for random variables in levels 1 and 2 go independently from 2 to 15. This technique has been used to optimize the number of possible values of hidden variables 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.

The fact that results are better for the tree model than for the HMM tells us that non-local dependencies are present in root notes progressions [12]. Generated root notes sequences given out-of-sample melodies are presented in Section 3.4 together with generated chord structures.

3.3 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

Table 2. Average conditional negative out-of-sample log-likelihoods of sub-sequences of root notes 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. The results are better (lower negative likelihood) for the tree model than for the HMM

Model	Negative log-likelihood
Tree	6.6707
HMM	8.4587

allows us to model dependencies between each chord component and the proper pitch-class components in the melodic representation presented in Section 3.1.

The model that we present in this section is observing chord symbols as they appear in [6] instead of actual *instantiated* chords (i.e. observing directly musical notes derived from the chord notation by a real musician). 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 note component (which can have 12 possible values given by the pitch-class of the root note of the chord) and 6 structural components detailed in Table 3. While it is out of the scope of this paper to describe jazz chord notation in detail [7], we just note that there exists a one-to-one relation between the chord representation introduced in Table 3 and chord symbols as they appear in [6].

We show in Table 4 the mappings of some chord symbols to structural vectors according to this representation. 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 not look 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 us to directly model the dependencies between chord components and melodic components.

3.4 Chord Model given Root Note Progression and Melody

Figure 2 shows a probabilistic model designed to predict chord progressions *given* root note progressions and melodies. The nodes in level 1 are discrete hidden nodes as in the root notes progressions model. The gray boxes are subgraphs that are detailed in Figure 3.

Table 3. 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 (b) 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	-	-

Table 4. Mappings from some chord symbols to structural vectors according to notation described in Table 3. For instance, the chord with symbol 7#5 has a major third (M), an augmented fifth (#), a minor seventh (m), no ninth, no eleventh and no thirteenth

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

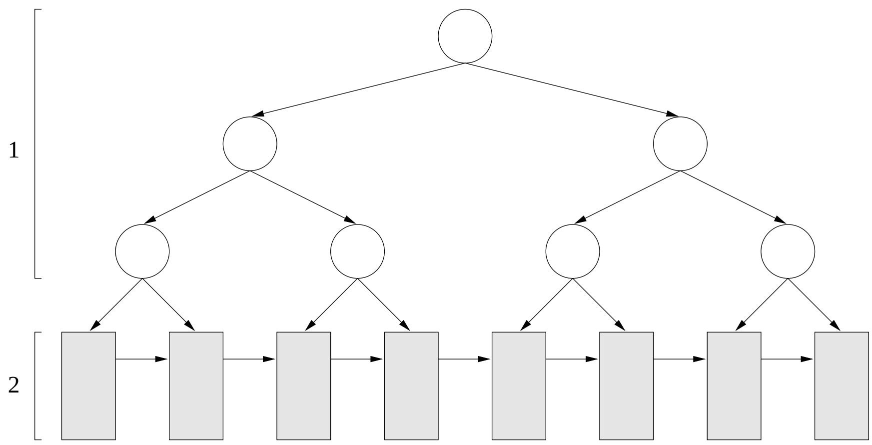

Fig. 2. A graphical model to predict chord progressions given root notes progressions and melodies. The gray boxes correspond to subgraphs presented in Figure 3

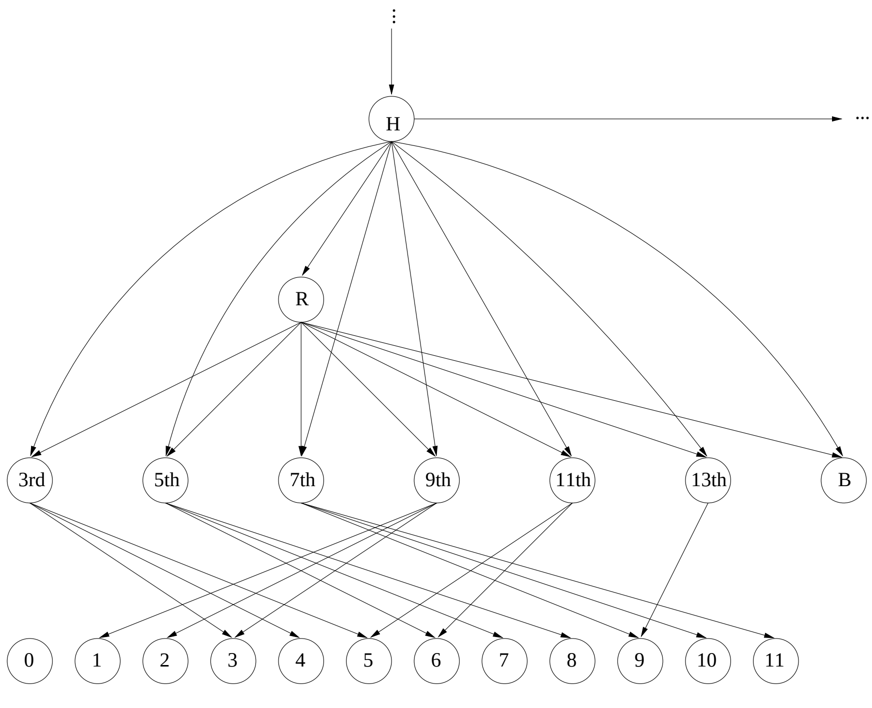

Fig. 3. Subgraph of the graph presented in Figure 2. Each chord component is linked with the proper melodic components on the bottom

The H node is a discrete hidden node modeling local dependencies and corresponding to the nodes on level 2 in Figure 2. The R node corresponds to the current root note. This node can have 12 different states corresponding to the pitch class of the root note and it is always observed. Nodes labeled from 3rd to 13th correspond to the structural chord components presented in Section 3.3. 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 note 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 labeled from 0 to 11 correspond to the melodic representation introduced in Section 3.1.

It should be noted that the melodic components are observed *relative* to the current root note. In Section 3.2, 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 note is G, component 0 will be associated to G, component 1 to G#, component 2 to A, and so on. This approach is necessary 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 3.

Generation of harmonization It is possible to evaluate the prediction ability of the model for chord structures. We present in Table 5 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 note progressions and the melodies for the tree model and an HMM model built by removing the nodes in level 1 in Figure 2.

Table 5. 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 note progressions and melodies using double cross-validation

Model	Negative log-likelihood
Tree	9.9197
HMM	9.5889

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 note progressions are given in these experiments. This would mean that most of the contextual information would be contained in the root note progression, which make sense intuitively. Further statistical experiments could be done to investigate this behavior. Table 6 shows three different harmonizations of the last 8 measures of the jazz standard *Blame It On My Youth* [6] generated by the proposed model.

Table 6. 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 note progression and melody with respect to the model presented in Section 3.4. Finally, rows beginning with NH correspond to a new harmonization generated by the same model and the root note progression model presented in Section 3.2 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

When observing the predicted structures given the original root notes 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 original chords. It is more interesting to look at the sequence of chords generated by taking the sequence of root notes with the highest probability given by the root note progression model presented in Section 3.2 and then finding the most likely chord structures given this predicted root note 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 short term. More generated results from the models presented in this paper are available on `http://www.idiap.ch/~paiement/canai`.

4 Conclusion

In this paper, we introduced a representation for chords that allows us to easily introduce domain knowledge in a probabilistic model for harmonization by considering every structural components in chord notation.

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. However, the local (HMM) model seems to be sufficient when root notes progressions are provided. This behavior suggest that most of the time-dependent information may already be contained in root note progressions.

Finally, we designed a probabilistic model that can be sampled to generate very interesting chord progressions given other polyphonic music components such as melody or even root note progressions.

Acknowledgments

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.

References

[1] Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks **5** (1994) 157–166
[2] Cemgil, A.T.: Bayesian Music Transcription. PhD thesis, Radboud University of Nijmegen (2004)
[3] Eck, D., Schmidhuber, J.: Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In Bourlard, H., ed.: Neural Networks for Signal Processing XII, Proc. 2002 IEEE Workshop, New York, IEEE (2002) 747–756
[4] Raphael, C., Stoddard, J.: Harmonic analysis with probabilistic graphical models. In: Proceedings of ISMIR 2003. (2003)
[5] Lavrenko, V., Pickens, J.: Polyphonic music modeling with random fields. In: Proceedings of ACM Multimedia, Berkeley, CA (2003)
[6] Sher, C., ed.: The New Real Book. Volume 1. Sher Music Co. (1988)
[7] Levine, M.: The Jazz Piano Book. Sher Music Co./Advance Music (1990)
[8] Allan, M., Williams, C.K.I.: Harmonising chorales by probabilistic inference. In: Advances in Neural Information Processing Systems. Volume 17. (2004)
[9] Lauritzen, S.L.: Graphical Models. Oxford University Press (1996)
[10] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society **39** (1977) 1–38
[11] Cooper, G., Meyer, L.B.: The Rhythmic Structure of Music. The Univ. of Chicago Press (1960)
[12] Paiement, J.F., Eck, D., Bengio, S., Barber, D.: A graphical model for chord progressions embedded in a psychoacoustic space. In: Proceedings of the 22nd International Conference on Machine Learning. (2005)
[13] Paiement, J.F., Eck, D., Bengio, S.: A probabilistic model for chord progressions. In: Proceedings of the 6th International Conference on Music Information Retrieval. (2005)
[14] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer series in statistics. Springer-Verlag (2001)