Probabilistic Models for Melodic Prediction

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

[a] *Idiap Research Institute, Centre du Parc, Rue Marconi 19, Case Postale 592, CH-1920 Martigny, Switzerland*

[b] *Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA*

[c] *University of Montreal, Department of Computer Science and Operations Research, Pavillon André-Aisenstadt, CP 6128, succ Centre-Ville, Montréal, QC, H3C 3J7, Canada*

Abstract

Chord progressions are the building blocks from which tonal music is constructed. The choice of a particular representation for chords has a strong impact on statistical modeling of the dependence between chord symbols and the actual sequences of notes in polyphonic music. Melodic prediction is used in this paper as a benchmark task to evaluate the quality of four chord representations using two probabilistic model architectures derived from Input/Output Hidden Markov Models (IOHMMs). Likelihoods and conditional and unconditional prediction error rates are used as complementary measures of the quality of each of the proposed chord representations. We observe empirically that different chord representations are optimal depending on the chosen evaluation metric. Also, representing chords only by their roots appears to be a good compromise in most of the reported experiments.

Key words: Music Models, Graphical Models, Probabilistic Algorithms, Machine Learning.

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

* Corresponding author.

Email addresses: paiement@gmail.com (Jean-François Paiement), bengio@google.com (Samy Bengio), douglas.eck@umontreal.ca (Douglas Eck).

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 [5]. This difficulty motivates our exploration of chord progressions and their interaction with melodies.

1.1 Music Background

A chord is a group of three or more notes. A chord progression is simply a sequence of chords. In probabilistic terms, the current chord can be seen as a latent variable (local in time) that conditions the probabilities of choosing particular notes in other music components, such as melodies or accompaniments. Chord changes occur at fixed time intervals in most of the musical genres, which makes them much simpler to detect than beginnings and endings of musical notes, which can happen almost everywhere in a music signal. Thus, knowing the relations between such chords and actual notes would certainly help to discover long-term musical structures in tonal music. For instance, an interesting challenge arising in the music information retrieval context is transcription, i.e. converting audio data into any kind of symbolic representation such as MIDI or traditional music notation. However, because of fundamental difficulties inherent to the nature of sound, state-of-the-art techniques are not able to accomplish this task with a sufficient level of precision for most practical applications. An intermediate goal is to try to infer chord symbols from audio data [3,19]. This task is simpler than complete transcription of polyphonic audio signal. Combining reliable chord transcription with a model of the conditional distribution of other music components (e.g. melodies) given chords could help to improve transcription error rates of existing algorithms. Following the same idea, such models could even be included in genre classifiers or automatic composition systems [9] to increase their performance.

In most tonal music theories, chord names are defined by a root note that can either be expressed by its absolute pitch-class [1] or by its relation with the current key. The key of a song is designated by a note name (the tonic), and is the base of a musical scale from which most of the notes of the piece are drawn. Most commonly, that scale can be either in major or minor mode.

[1] A pitch-class is a note name, like C or D. In this paper, we consider enharmonics (e.g. Eb and D#) to be completely equivalent.

1.2 Previous Work

Previous papers describe probabilistic models to solve music related problems. Using graphical models, Cemgil [6] introduces a somewhat complex probabilistic 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 computationally 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 and Stoddard [19] use graphical models for labeling MIDI data with traditional Western chord symbols. Lavrenko and Pickens [14] 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 using abstract chord representations as a smoothing technique towards better generalization. Begleiter et al. [2] provide an interesting comparison of variable order Markov models used for polyphonic music prediction, but without any use of chord information to overcome long term dependencies. Harmonization is the generation of a chord progression for a given melody. Allan and Williams [1] designed a harmonization model for Bach chorales using HMMs. While generating excellent musical results, this model has to be provided polyphonic music with a specific 4 voice structure as input, restricting its applicability to very specific settings. The models proposed in this paper are more general in the sense that it is possible to extract the appropriate chord representation from any polyphonic music, whatever the specific labeling, harmonic structure, or musical style.

1.3 Comparing Chord Representations

Despite the simple and relatively universal chord building principles, many different notations have been used through music history to represent chord progressions [7]. We face the same problem in the computer science literature, where each author uses a notation corresponding to his musical background [1,16,19,20,23]. All these papers describe arbitrary chord representations embedded in probabilistic models, which are mostly variants of Hidden Markov Models (HMMs). However, to the best of our knowledge, there is no available quantitative comparative study of the effect of the choice of particular chord representations to solve practical applications. Each chord representation carries specific information that could be more adapted to certain tasks (or musical styles) than others. At the same time, all of these notations encapsulate basic information about a chord, such as its root.

What are the most appropriate chord representations in terms of statistical polyphonic music modeling? Also, what are the most appropriate objective criteria for parameter estimation and evaluation of such representations? This paper explores these issues by using melodic prediction [22] given chords as a benchmark task to evaluate the quality of four chord representations defined in Section 2, along with various graphical models involving different temporal dependencies. Likelihoods and conditional and unconditional prediction error rates are used as complementary measures of the quality of each of the proposed chord representations in Section 3. Finally, we discuss our empirical results and draw conclusions in Section 4.

The generative models described in this paper are trained with the EM algorithm. It should be noted that probabilistic models can also be trained with discriminative training methods [12]. However, a generative approach is more appropriate here, since the objective of the reported experiments is not to optimize absolute melodic prediction performance, but to compare chord representations in a more general way.

2 Melodic Prediction Models

In order to assess the effect of using particular chord representations for melodic prediction, we propose two kinds of probabilistic models, described using the graphical model framework.

2.1 Graphical Models and EM

Graphical models [13] are useful to define 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 of all the variables in the model. The process of computing 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.

The Expectation-Maximization (EM) algorithm [8] can be used to estimate the conditional probabilities of the hidden variables in a graphical model. Hidden variables are variables that are neither observed during training nor during evaluation of the models. These variables represent underlying phenomena that have an impact on the actual observations, but that cannot be observed directly. The EM algorithm proceeds in two steps applied iteratively over a dataset until convergence of the parameters. Firstly, 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, given the expected values of the hidden variables.

2.2 A Local Model

The first proposed modeling strategy is to look at the direct effect of particular chord representations without any influence from past observations.

Let $\mathbf{m}^j = (m_1^j, \ldots, m_n^j)$ be the j-th melody in a dataset where each song have length n. Each m_t^j is a discrete random variable representing the melodic note played at time t in song j. In this paper, we assume octave invariance for the melodic observations. In other words, all notes belonging to the same pitch-class are considered to be the same (e.g. all C notes regardless of octave are associated to the same random variable value). Hence, we assign one possible value of m_t^j to each pitch-class, plus one extra value for silence, leading to a total of 13 possible melodic values.

Also, let $\mathbf{c}^j = (c_1^j, \ldots, c_n^j)$ be the chord progression in song j where each c_t^j is a discrete random variable with a possible number of values depending on the chosen chord representation, as described in Section 2.4. Each c_t^j is the chord that is occurring while melodic note m_t^j is played in the j-th song of the dataset. A very simple approach is to model the joint distribution of $\mathbf{m}^j$ and $\mathbf{c}^j$ with

$$p_{\text{Local}}(\mathbf{m}^j, \mathbf{c}^j) = \prod_{t=1}^{n} p(c_t^j) p(m_t^j | c_t^j) \ , \qquad (1)$$

where both $p(c_t^j)$ and $p(m_t^j | c_t^j)$ are multinomial distributions. The factorization of the joint distribution in Eq. (1) is illustrated by the graphical model in Figure 1, not considering the dashed arrows. The upper indices of the random variables have been removed in the figure for clarity. In the remainder of this paper, we refer to this model as the `Local` model, because each time-step is independent of the others. Variables in level 1 are observed during training *and* testing.

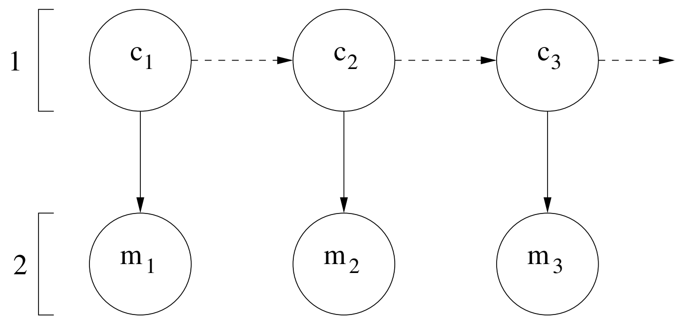

Fig. 1. A simple probabilistic model corresponding to the factorization in Eq. (1), where the upper indices of the random variables have been removed for clarity. The influence of each chord on melodic observations is direct. Circles represent random variables and arrows represent conditional probability distributions. The dashed lines are not present in the `Local` model, thus making each observation completely local in time. Variables in level 2 correspond to melodic observations.

The chosen melodic representation does not account for note similarities. In the proposed models, the probability of the melodic observations given appropriate other random variables is modeled by multinomial distributions. Such a distribution does not embed any notion of similarity between its possible outcomes. However, we chose this melodic representation for its simplicity and also to avoid introducing bias while measuring the quality of the chord representations for melodic prediction. While this model is overly simplistic for practical purposes, it has the advantage of isolating the direct effect of particular chord representations on the choice of melodic notes. Since all the variables are observed during training, parameters for this model can be easily learned from a dataset by standard maximum likelihood techniques.

2.3 IOHMM Model

A more realistic model can be designed by adding extra hidden variables in the previous model to consider influences from the past when trying to predict a melody note. Let $\mathbf{h}^j = (h_1^j, \ldots, h_n^j)$ be a vector of discrete hidden variables. The joint distribution of $\mathbf{m}^j$, $\mathbf{c}^j$, and $\mathbf{h}^j$ can be factorized by

$$p_{\text{IOHMM}}(\mathbf{m}^j, \mathbf{c}^j, \mathbf{h}^j) = p(c_1^j)p(h_1^j|c_1^j)p(m_1^j|h_1^j)\prod_{t=2}^{n} p(c_t^j)p(h_t^j|h_{t-1}^j, c_t^j)p(m_t^j|h_t^j) \quad ,(2)$$

where all the distributions are multinomial. This particular factorization is illustrated by the graphical model shown in Figure 2, where the upper indices of the variables have been removed for clarity. This model is very similar to an Input/Output Hidden Markov Model (`IOHMM`) [4]. Chord variables c_t^j in level 1 are always observed. Hidden variables h_t^j in level 2 are used to introduce dependencies between time frames in the model. Again, melodic variables c_t^j

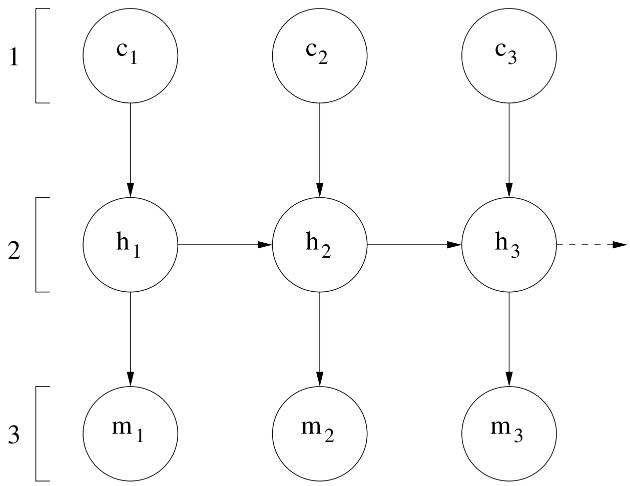

Fig. 2. Variant of an `IOHMM` model, as expressed by the joint factorization in Eq. (2). The upper indices of the variables have been removed for clarity. The variables in level 1 are observed and correspond to chord observations. Variables in level 2 are hidden, while variables in level 3 correspond to melodic observations.

in level 3 have 13 possible values as in the `Local` model presented in Section 2.2. There is no link between level 1 and level 3 variables on Figure 2, contrary to standard IOHMMs [4]. The number of possible values is highly variable from one chord representation to another. Considering that, we chose to remove the usual links between inputs and outputs in IOHMMs in order to limit the impact of the particular choice of a chord representation on the capacity of the model. This way, the number of possible values of the chosen chord representation has an impact on the parameterization of the conditional distribution of the hidden variables, but not on the conditional distributions of the predicted melodic variables. Learning in the model is done with the standard EM algorithm [8]. Marginalization must be carried out in the proposed model both for learning (during the expectation step of the EM algorithm) and for evaluation. Exact marginalization with the standard Junction Tree Algorithm [13] is tractable in IOHMMs because of their limited complexity.

2.4 Chord Representations

The chord representations that we introduce in this section consider chord symbols as they are represented in musical analysis instead of actual instantiated chords. In other words, we observe chord symbols such as they appear in music sheets [21] instead of observing the notes that would be played by a musician reading these chord symbols. As we noted in Section 1, chords can be seen as a latent variable (local in time) that conditions the probabilities of choosing particular notes in other music components, such as melodies or accompaniments. The chord symbol “C Maj7” is usually constructed using the pitch classes C, E, G, and B. However, it really defines a conditional probability over *all* pitch classes. For instance, the pitch class D would normally be much more likely over this particular chord than the pitch-class Bb. Note that it is easy to infer valid chord symbols from the actual notes in most datasets with deterministic methods, 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 with minimal preprocessing effort.

Four chord representations have been used in the experiments described in Section 3. First, what we call a `Naive` representation is to consider every chord (including the choice of the root) as a distinct observation. This representation has the disadvantage of excluding any notion of chord similarity. Moreover, this representation leads to a large number of states for the associated random variables (e.g. 152 in the current experiments, corresponding to each different chord found in the dataset described in Section 3). This can be harmful when learning over small datasets because of the high number of parameters. Despite all these drawbacks, such a representation can be useful if the notions of chord similarities are included in others parts of the models, such as in the conditional probabilities between variables [15].

Another possible chord representation is to discard any information except the root, yielding random variables with 12 possible values. While having a reasonable number of possible values, such a representation introduces a lot of smoothing in the models. It is possible to automatically detect the key and the mode (major or minor) of a song. Given that information, the root is very often sufficient to predict the whole structure of the rest of the current chord. For instance, given a song in C major and observing a root C, it is very likely that the complete chord associated to this root is a variant of C major.

We can also restrict ourselves to a subset of all possible chords [3,20] by mapping more complex observed chord symbols to a subset of simpler ones. Such a representation is also used in the experiments described in this paper. We define only whether a chord is minor, major or dominant, leading to chord random variables with only 3 possible values. In this case, if we observe for instance the chord C7#5b9, we map this chord to the value corresponding to a dominant (7) chord. In the remainder of the paper, this representation is referred to as the `mM7` (**m**inor-**M**ajor-dominant **7**th) representation.

Finally, we can combine the `Root` representation and the `mM7` representation. This leads to discrete random variables with 36 possible values (12 roots times 3 chord qualities).

3 Comparing Chord Representations

53 jazz standards melodies [21] were recorded by the first author in MIDI format. Corresponding chord labels were also manually added to this corpus. The complexity of the chord sequences and melodies found in the corpus is representative of the complexity of common jazz and pop music. The songs were transposed to the key of C. This simplification has no impact on the generality of the experiments since automatic key detection is relatively reliable. Every song was 16 bars long, with four beats per bar, and with one chord change every two beats. The shortest notes in the dataset are eighth notes. Hence, two melodic observations were made for each beat, yielding observed sequences of length 128. Since chords span multiple melodic observations, their symbols were repeated multiple times (e.g. a 6 beat chord is represented as 12 distinct observations). This has no impact on the quality of the model since chords are acting as latent variables. Hence, the fact that chords are repeated in input sequences does not imply actual repetitions in the corresponding melodic signal.

3.1 Local Model

We chose to use cross-validation [10] to evaluate the proposed models instead of dividing the dataset into two parts (i.e. a single training set and a single test set) because of the small size of our experimental dataset. Assume that the dataset is divided into K folds $T_1, \ldots, T_K$ (each containing different sequences), and that the k-th fold T_k contains η_k test sequences. We also define $k(j)$ such that $\forall j,\ j \in T_{k(j)}$. In other words, $k(j)$ is the index of the fold containing the sequence j. Let

$$\hat{m}_t^j = \arg\max p_{\mathrm{Local}(k(j))}(m_t^j|\mathbf{c}^j) \tag{3}$$

where the parameters of $p_{\mathrm{Local}(k(j))}(m_t^j|\mathbf{c}^j)$ are learned on all the sequences of the dataset, *without* the sequences in $T_{k(j)}$. Hence, $\hat{m}_t^j$ is the most probable melodic observation at time t in test song j given the whole chord progression, according to the `Local` model described in Section 2.2. Given Eq. (1), we have that

$$\hat{m}_t^j = \arg\max p_{\mathrm{Local}(k(j))}(m_t^j|c_t^j) \ .$$

Thus, a melodic prediction in the `Local` model only depends on the current chord observation. Finally, let $\varepsilon_t^j = 1$ if $\hat{m}_t^j \neq m_t^j$. When using cross-validation,

Table 1
Out-of-sample classification error and average negative out-of-sample log-likelihood for local models. The lower the better for both evaluation measures. The number of possible states for each chord representation is given in the second column.

Model	Chord states	% of Error	Neg. Log-Likelihood
Naive	152	**77.39**	357.48
Roots + mM7	36	80.93	316.33
Roots	12	81.40	**293.95**
mM7	3	82.41	295.29
Freq	0	83.34	301.40

the out-of-sample classification error of the local model is given by

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\eta_k}\sum_{j\in T_k}\frac{1}{n}\sum_{t=1}^{n}\varepsilon_t^j \ . \tag{4}$$

In words, the out-of-sample error is just the average number of times the algorithm makes a mistake when trying to predict melodic observations over songs *unseen* during training. Out-of-sample classification errors (computed with 5-fold cross-validation) for the `Local` model described in Section 2.2 and for each of the chord representations given in Section 2.4 are presented in Table 1.

It should be pointed out that the classification error is really the criterion we want to minimize when developing models for practical applications. In such a context, the system *must* make a decision at each time step. As an example, a melodic model could be appended to a transcription algorithm. In such a context, the model would have to guess what is the most likely next note, given the previous notes and the audio measurements.

On each iteration of cross-validation, one fold of the dataset is not used for training. This subset of the dataset (referred to as the *test set* in machine learning literature) can be used to evaluate the model. Hence, it is always possible to observe a chord symbol during evaluation (or testing) that has not been observed previously when training the model. Dirichlet priors [11] have been used on all the chord variables in the algorithms described in this paper in order to avoid propagating zero probabilities in the models in this case. Note that we did not include these priors in the descriptions of the proposed models in order to clarify the presentation. Dirichlet priors on multinomial distributions amounts to assume that some observations have been made by the models before learning the parameters. In the reported experiments, we simply assumed that every variable states given every conditioning variable states have been observed one time.

Parameters are tied over time in the `Local` model presented in Figure 1. In other words, the arrows between level 1 and level 2 always correspond to the same probability table in one fold of the cross-validation. As a benchmark, we also introduce the `Freq` model in Table 1, which is simply an algorithm that always selects the most frequent melodic observation in the training set. It corresponds to removing level 1 random variables in Figure 1, just leaving independent observations.

The capacity of a set of functions [24] can be related to the size of the set: the more functions in the set, the higher the capacity. Searching for the optimal function in a set with not enough capacity will yield *underfitting*, while searching in a set with too much capacity will yield *overfitting*. Hence, techniques such as cross-validation can be used to select the appropriate set size. With the `Local` model, the only direct way to optimize capacity is to vary the parameter of the Dirichlet prior on the distribution over melodies (which we did not do in the experiments). However, using a chord representation with more possible values also increases capacity in this model. Looking at the obtained results, we see that the rate of error increases when using chord representations with *fewer* possible values. It is then possible that all these models somehow underfit, making the `Naive` representation (with 152 possible chord values in the current experiments) the best choice in this context. Not surprisingly, the `Freq` model, which always chooses the most frequent melodic observation, is the worst model in terms of classification error rate. Generalization capabilities of a model can benefit from the smoothing produced by a simplified chord representation, since this simplification is done by clustering perceptual properties of chords (e.g. all chords with the same root are clustered together in the `Root` representation). This is obviously not the case when using the `Freq` model.

Table 1 also shows the average negative out-of-sample log-likelihoods obtained with the same models again using cross-validation. Using the same notation as in Equation (4), this measure is given by

$$-\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\eta_k}\sum_{j\in T_k}\frac{1}{n}\sum_{t=1}^{n}\log\left(p_{\mathrm{Local}(k)}(m_t^j|\mathbf{c}^j)\right) \quad . \tag{5}$$

where the parameters of $p_{\mathrm{Local}(k)}(m_t^j|\mathbf{c}^j)$ are learned on all the sequences of the dataset, *excluding* the ones that are in T_k. In other words, we compute each likelihood on examples that have not been used to learn the parameters in the model. Note that this performance measure is the one that was optimized over the training set when learning the parameters of the models with the maximum likelihood algorithm. The likelihoods of the different models are comparable since all the proposed models observe the *same* melodic representation.

As can be observed, the negative log-likelihood results are not coherent with the classification error. For instance, the `Freq` model has a lower negative log-likelihood than the `Roots + mM7` and `Naive` models! While being surprising, this result is plausible since the reported log-likelihoods have been computed on sequences that have not been used to train the models. Hence, a model with lower capacity can have better generalization properties than a too complex model (i.e. an overly complex model may *overfit* the training dataset.) However, this result is counterintuitive since one would expect that adding current chord information would help the model to guess what would be the current melody note.

The observed discrepancies between classification error rate and negative log-likelihood could be explained by the fact that the negative log-probabilities in Eq. (5) are not bounded. Suppose that a model fits most of the data quite well but some of the out-of-sample examples have very low probabilities. Then, the terms associated to these examples in Equation (5) can take very large values that could dominate the average negative log-likelihood for all the examples. On the other hand, the cost of encountering a very unlikely out-of-sample sequence (with respect to the model being evaluated) in Eq. (4) is only proportional to $\frac{1}{N}$, with $N = \sum_{k=1}^{K} \eta_k$ being the total number of examples in the dataset. This observation raises the following interesting question: Is the likelihood of the model over the observed data the best criterion to optimize when what we *really* want to do is to minimize the error of classification? We further discuss this issue in Section 4.

3.2 IOHMM Model

Classification error and average negative out-of-sample log-likelihoods can also be computed for the `IOHMM` defined in Eq. (2). Let

$$\tilde{\mathbf{m}}^j = \arg\max p_{\mathrm{IOHMM}(k(j))}(\tilde{\mathbf{m}}^j|\mathbf{c}^j) \tag{6}$$

with $\tilde{\mathbf{m}}^j = (\tilde{m}_1^j, \ldots, \tilde{m}_n^j)$. Again, the parameters of $p_{\mathrm{IOHMM}(k(j))}(\mathbf{m}^j|\mathbf{c}^j)$ are learned on all the sequences of the dataset, *excluding* the sequences in $T_{k(j)}$. The out-of-sample classification error for the `IOHMM` is given by Eq. (4), with $\varepsilon_t^j = 1$ if $\tilde{m}_t^j \neq m_t^j$. In other words, the model tries to guess the whole melodic sequences given the corresponding chord progressions.

Table 2
Out-of-sample classification error and average negative out-of-sample log-likelihood for the `IOHMM`. The lower the better for both evaluation measures. The number of possible states for each chord representation is given in the second column.

Model	**Chord states**	**% of Error**	**Neg. Log-Likelihood**
Naive	152	**79.05**	281.84
Roots	12	82.09	223.67
Roots + mM7	36	83.17	247.48
mM7	3	84.71	212.47
HMM	0	86.56	**196.27**

On the other hand, the average negative out-of-sample log-likelihood for the `IOHMM` is given by

$$-\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\eta_k}\sum_{j\in T_k}\log\left(p_{\mathrm{IOHMM}(k)}(\mathbf{m}_t^j|\mathbf{c}^j)\right) \quad ,$$

following the notation in Eq. (5).

Table 2 shows out-of-sample classification error and average negative log-likelihood for the `IOHMM`. These results are qualitatively similar to those in Table 1 for the `Local` model. This time, the number of possible values for hidden variables in level 2 of Figure 2 was optimized using 5-fold double cross-validation, which is a recursive application of cross-validation where both the optimization of the parameters of the model and the evaluation of the generalization performance of the model are carried out simultaneously. Standard cross-validation as described in the beginning of this section was applied to each subset of 4 folds with each hyper-parameter setting and tested with the best set of parameters (on average) on the remaining hold-out fold. 2 to 20 possible hidden states for the variables $\mathbf{h}^j$ were tried in the reported experiments.

The same parameters are used over time to define the conditional probability distributions. For instance, all the vertical arrows between variables in level 1 and level 2 in Figure 2 represent the *same* probability table. The fact that it was possible in this context to optimize the capacity of the models by adjusting the number of states for the hidden variables makes the results in Table 2 more trustworthy than the ones found in Table 1, although they are similar. It should be pointed out that the capacity of the models was optimized with respect to the appropriate error measure. For instance, when reporting results about prediction error rates, capacity is optimized with respect to prediction error rate (while the models are trained by maximizing the likelihood with the

EM algorithm).

The `HMM` referred to in Table 2 is similar to the `IOHMM` but removing the chord inputs layer. Hence, the joint distribution of the melodic observations $\mathbf{m}^j$ and the hidden variables $\mathbf{h}^j$ estimated by this model is given by

$$p_{HMM}(\mathbf{m}^j, \mathbf{h}^j) = p(h_1^j)p(m_1^j|h_1^j)\prod_{t=2}^{n} p(h_t^j|h_{t-1}^j)p(m_t^j|h_t^j) \ ,$$

where all the distributions are multinomial. Again, all the parameters in this model can be learned with the standard EM algorithm [18]. This particular factorization can be represented by the model in Figure 1 with the horizontal dashed arrows being present and the variables c_t^j replaced by the corresponding variables h_t^j. In this case, variables in level 1 are hidden and variables in level 2 are still melodic observations. As expected, the `HMM` produces higher out-of-sample classification error than the `IOHMM`s, which can take advantage of the chord symbols given as inputs. Interestingly, the `Naive` representation for chords seems to be consistently efficient for melodic prediction. In Table 2, the `Naive` representation gives statistically significantly better results than the `Roots` representation with a confidence level of 99% [2]. This is an indication that developing probabilistic models with this representation could be a viable approach, especially if perceived relations between chords are included in the models via the conditional probabilities related to these chords [15]. The representation including only the roots also performs well. Given these results, this representation appears to be a good compromise given its relative simplicity and the fact that it inherently embodies perceptually relevant smoothing. Using basic chord information (`mM7` representation) does not seem to help with respect to unconditional classification error.

Again, average negative log-likelihoods contradict average classification error rates. Even worse, the `HMM`s performs much better than the `IOHMM`s in terms of likelihood! This is an indication that such a measure favors models that are more uniform in essence, thus giving a relatively high probability to unseen sequences. However, more uniform models are weaker when asked to predict a single new note, because they define distributions with modes less precisely adapted to the training data.

[2] We used a standard proportion test, assuming a binomial distribution for the errors and using a normal approximation.

3.3 Conditional Classification Error

The goal of the models presented here is to predict the melodies in the dataset. It is out of the scope of this work to evaluate the subjective artistic quality of the predicted melodies. A more interesting measure of melodic prediction is the out-of-sample *conditional* classification error for the `Local` model, given by Equation (4), but using instead

$$\hat{m}_t^j = \arg\max p_{\text{Local}(k)}(m_t^j|\mathbf{c}^j, m_1^j, \ldots, m_{t-1}^j)$$

in the definition of ε_t^j. The same technique can be applied to compute the classification error rate for the `IOHMM` if using the appropriate distribution derived from Eq. (2). This measure is very similar to the unconditional error described in Section 3.2. However, the models now have access to the true previous melodic observations when trying to guess the next one. This is different from the unconditional error rate, where each model tries to predict whole sequences when only given chord progressions.

The only objective performance measure we can provide about a melodic prediction given a chord progression is to tell if a predicted melody is similar or not to the one provided in the dataset with the same chord progression. However, while the space of plausible melodies is huge, we only have access to a very small number of sequences to measure the performance of the models given a chord progression. Moreover, given a particular sequence of chords, one can imagine that a very high number of melodies would be considered more or less musically similar to the ones in the dataset. Among all these melodies, some may not share a single note with the true melody associated with this sequence of chords in the test set. A good melodic prediction model would be likely to generate any one of these melodies, thus producing a very high unconditional error rate.

The conditional error rate alleviates this problem by measuring the prediction performance of a model in regions of the observation space where data is present, leading to a much more reliable performance measure in this context. Moreover, distributions that would generalize well according to such a measure could also be sampled to generate realistic melodies given chord progressions and initial melodic motives. The Junction Tree Algorithm for marginalization [13] allows us to fix the state of some variables in the graphical model while finding the most probable states for other variables. Out-of-sample conditional classification error rates for the `IOHMM`s presented in Section 2.3 are shown in Table 3. We do not provide conditional classification error rates for the `Local` model described in Section 2.2, because they would be identical to the unconditional classification error rates shown in Table 1. When making a prediction, such models are completely unaware of previous observations

Table 3
Out-of-sample conditional classification error rates for the IOHMMs. The lower the better. The number of possible states for each chord representation is given in the second column.

Model	**Chord states**	**% of Error**
Roots	12	**57.41**
Roots + mM7	36	58.21
mM7	3	58.32
Naive	152	69.77
HMM	0	85.27

in time. Also, we do not provide average negative log-likelihoods, since this measure is not well adapted to prediction tasks, as we noted in Section 3.

Again, the HMM produces higher out-of-sample conditional classification error rate than the IOHMMs which benefit from chord symbols given as inputs. In Table 3, the conditional error rates are much lower than the unconditional ones for the same models. For each chord representation, the prediction accuracy gained when observing previous melodic notes is much higher than the differences in error rates for each chord representation obtained in Table 2. This means that observing previous melodic notes gives more information about the likely choices for the current melody than any chord information.

In Table 2, the Naive representation was the best one in terms of unconditional prediction error rate. On the other hand, this representation has the highest conditional prediction error rate among all the IOHMMs. When knowing nothing about the previous melodic observations, the model performs better when provided with a more detailed chord representation. However, given previous melodic observations, smoothed chord representations lead to better generalization in terms of prediction error rate. The Naive representation overfits the training data because it leads to models with higher capacity. Finally, no representation is statistically significantly better than another with a confidence level of 90% among the three best representations in Table 3.

4 Conclusions

The main motivation behind this paper was to better understand the statistical relations between chord representations and the actual choice of notes in polyphonic music. To this end, we compared four chord representations using melodic prediction as a benchmark task. Surprisingly, the Naive representation where each chord is conceived as a discrete observation apparently performs well in terms of unconditional prediction error rates. Nevertheless, this representation overfits when past melodic observations are used to condition the predictions. In this case, smoothed chord representations seems more appropriate. Given the obtained results, representing chords only by their roots seems to be a good compromise, especially when all the songs to be analyzed are transposed to the same key. While being extremely simple, this representation inherently includes smoothing related to perceptual relations between notes. The `Root+mM7` representation that is used in some important music information retrieval papers [3,19] is not optimal in terms of out-of-sample classification error and average negative log-likelihood for both probabilistic models presented in this paper. However, in practice, the actual choice of a chord representation should always be made considering the application to be developed.

An interesting observation when looking at the results of the experiments done in Section 3 is that the behavior of the average out-of-sample likelihood does not follow the trends of the average prediction error rate (conditional or unconditional). On the one hand, the likelihood is a measure of the fit of a whole distribution to a dataset. However, the classification error seems to be a better descriptor of the fit of the *modes* of a distribution. These remarks are applicable to any domain where parameters of models have to be learned on small datasets. Provided a nearly infinite amount of data, the two evaluation measures that we presented would lead to the same ranking of the models. Thus, likelihood and prediction error would probably be more comparable when measured with models trained and evaluated with much more data. How much data is needed in this particular framework to obtain such a behavior is still an open question that needs to be addressed. Another important observation is that when learning parameters on small datasets, the chosen capacities for the models appear to be even more crucial than when learning on large dataset.

Also, given realistic datasets, optimizing the likelihood of a model with respect to training data may not be the best strategy when one is only interested in the modes of the distribution, which can be the case when doing prediction. An alternative learning strategy would be to maximize the sum of the differences between the probabilities of the observed classes and the probabilities of the most probable wrong classes instead of just maximizing the sum of the log probabilities of the observed classes. This approach is referred to as the minimum classification error (MCE) algorithm [12].

Finally, it would be interesting to append these models to existing music information retrieval algorithms to improve their performance. As a comparison, language models are regularly used in speech recognition algorithms to constrain their search space to reasonable solutions [17].

References

[1] M. Allan, C. K. I. Williams, Harmonising chorales by probabilistic inference, in: Advances in Neural Information Processing Systems, vol. 17, 2004.

[2] R. Begleiter, R. El-Yaniv, G. Yona, On prediction using variable order Markov Models, Journal of Artificial Intelligence Research 22 (2004) 385–421.

[3] J. P. Bello, J. Pickens, A robust mid-level representation for harmonic content in music signals, in: Proceedings of the Sixth International Conference on Music Information Retrieval, London, 2005.

[4] Y. Bengio, P. Frasconi, Input/output HMMs for sequence processing, IEEE Transactions on Neural Networks 7 (5) (1996) 1231–1249.

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

[6] A. T. Cemgil, Bayesian music transcription, Ph.D. thesis, Radboud University of Nijmegen (2004).

[7] C. Dahlhaus, Studies on the Origin of Harmonic Tonality, Princeton University Press, 1990, XV-389 p.

[8] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1977) 1–38.

[9] D. Eck, J. Schmidhuber, Finding temporal structure in music: Blues improvisation with LSTM recurrent networks, in: H. Bourlard (ed.), Neural Networks for Signal Processing XII, Proc. 2002 IEEE Workshop, IEEE, New York, 2002.

[10] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer series in statistics, Springer-Verlag, 2001.

[11] D. Heckerman, D. Geiger, M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Tech. rep., Microsoft Research (1994).

[12] B.-H. Juang, S. Katagiri, Discriminative learning for minimum error classification, in: IEEE Trans. on Signal Processing, vol. 10 (12), 1992.

[13] S. L. Lauritzen, Graphical Models, Oxford University Press, 1996.

[14] V. Lavrenko, J. Pickens, Polyphonic music modeling with random fields, in: Proceedings of ACM Multimedia, Berkeley, CA, 2003.

[15] J.-F. Paiement, D. Eck, S. Bengio, A probabilistic model for chord progressions, in: Proceedings of the 6th International Conference on Music Information Retrieval, 2005.

[16] J.-F. Paiement, D. Eck, S. Bengio, Probabilistic melodic harmonization, in: Proceedings of the 19th Canadian Conference on Artificial Intelligence, Springer, 2006.

[17] L. Rabiner, B.-H. Juang, Fundamentals of speech recognition, 1st ed., Prentice All, 1993.

[18] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–285.

[19] C. Raphael, J. Stoddard, Harmonic analysis with probabilistic graphical models, Computer Music Journal 28 (3) (2004) 45–52.

[20] A. Sheh, D. P. Ellis, Chord segmentation and recognition using EM-trained Hidden Markov Models, in: Proceedings of the 4th ISMIR, Baltimore, Maryland, 2003.

[21] C. Sher (ed.), The New Real Book, vol. 1-3, Sher Music Co., 1988.

[22] D. Temperley, A probabilistic model of music perception, in: Proceedings of the 7th Conference on Music Information Retrieval, Victoria, Canada, 2006.

[23] B. Thom, Predicting chords in jazz: the good, the bad, and the ugly, in: IJCAI-95, Music and AI Workshop, Montreal, Canada, 1995.

[24] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.