
A Probabilistic Model for Melodies

Anonymous Author(s)

Affiliation

Address

email

Abstract

We propose a generative model for melodies in a given musical genre, using symbolic representation of musical data. We first compute melodic features that represent the plausibility of sequences of three consecutive notes. Their probabilistic modeling is an interesting intermediate problem since the cardinality of such features is much lower than the number of sequences of three notes. We then introduce a probabilistic model of melodies *given* chords and rhythms based on these features. This model leads to significantly higher prediction rates than a simpler Input/Output Hidden Markov Model. Moreover, sampling this model given appropriate musical contexts generates realistic melodies.

1 Introduction

In this paper, we present graphical models that capture melodic structures in a given musical style using as evidence a limited amount of symbolic MIDI¹ data. Predictive models have already been proposed for melodies [1, 2]. However, proper generative models would be desirable when modeling melodies, since one would want to estimate the probability of any arbitrary melodic sequence in most information retrieval applications. In this respect, models based on Markov random fields [3] are very general, but would benefit from using more specific musical knowledge. On the other hand, dictionary based predictors [4] generate subjectively impressive musical results, but we are not aware of musicological evidence to support such modeling of melodic sequences. In contrast to all of these approaches, we propose in Section 2.3 a melodic model based on constraints directly derived from musicological substantiation [5]. Moreover, we are not aware of proper quantitative comparisons between generative models of music, that is for instance in terms of out-of-sample prediction accuracy, as we do in Section 3.

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 in a song 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 [6] than beginnings and endings of musical notes, which can happen almost everywhere in music signal. Thus, knowing the relations between such chords and actual notes would certainly help to discover long-term musical structures in tonal music.

It is fairly easy to generate interesting chord progressions given melodies in a particular musical genre [7, 8]. However, the dual problem that we address in this paper is much more difficult. Music data contains very strong long-term dependencies, and these statistical relationships are very difficult to capture with traditional machine learning methods [9]. In Section 2.2, we describe melodic features that put useful constraints on melodies based on musicological substantiation [5]. We then

¹MIDI stands for Musical Instrument Digital Interface, an industry-standard interface used on electronic musical keyboards and PCs for computer control of musical instruments and devices. In our work, we only consider note onsets and offsets in the MIDI signal.

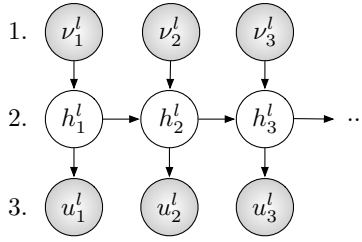


Figure 1: Variant of an IOHMM model for MIDI notes given chords. The variables in level 1 are always observed and correspond to chords. Variables in level 2 are hidden, while variables in level 3 correspond to melodic notes. All variables in grey are observed during training.

introduce in Section 2.3 a probabilistic model of melodies *given* chords and rhythms that leads to significantly higher prediction rates than a simpler Markovian model, as shown in Section 3. Reliable generative models for music could improve the poor performance of state-of-the-art transcription systems; it could as well be included in genre classifiers, automatic composition systems [10], or algorithms for music information retrieval [11].

2 Melodic Model

We propose a model for melody notes *given* chord progressions that is divided into two modules. We first model features that represent the plausibility of sequences of notes. These so-called “Narmour” features, introduced in Section 2.2, are computed for each sequence of three consecutive notes. Their prediction is an interesting intermediate problem since the cardinality of such features is much lower than the number of sequences of three notes. Moreover, Narmour features are descriptive of the perceptual expectancy of a particular group of three notes. As stated in Section 1, chords can be seen as latent variables (local in time) that condition the probabilities of choosing particular notes in a melody. However, chords do not describe longer term melodic structure. This is why we propose to use Narmour features as sequences of constraints on the choices of melody notes. In Section 2.3, we describe a probabilistic model for melody notes given Narmour features *and* chord progressions.

Results reported in Section 3 show that using sequences of Narmour features as constraints leads to much better prediction accuracy than the direct baseline approach using the IOHMM model described in the following section.

2.1 IOHMMs

Let $\mathcal{U} = \{\mathbf{u}^1, \dots, \mathbf{u}^n\}$ be a dataset of varying length monophonic melodies, where each melody \mathbf{u}^l has g_l notes, $\mathbf{u}^l = (u_1^l, \dots, u_{g_l}^l)$. Each melodic line is composed of notes u_i^l in the MIDI standard, $u_i^l \in \{0, \dots, 127\}$. Also, note indices correspond to the chronological order of the notes in the songs. Hence, the only rhythmical information we consider in this particular model is the order in which the notes are played. In addition, let $\nu^l = (\nu_1^l, \dots, \nu_{g_l}^l)$ be the chord progression corresponding to the l -th melody. Each ν_i^l represents the chord that is in effect while note u_i^l is played. Each ν_i^l takes a discrete value within the number of different chords in the dataset. Finally, let $\mathbf{h}^l = (h_1^l, \dots, h_m^l)$ be a sequence of states of discrete hidden variables synchronized with sequence \mathbf{u}^l . The joint probability of each sequence \mathbf{u}^l , its associated chord progression ν^l , and hidden states \mathbf{h}^l can be modeled by

$$p_{\text{IOHMM}}(\mathbf{u}^l, \nu^l, \mathbf{h}^l) = p_i(\nu_1^l) p_\pi(h_1^l | \nu_1^l) p_o(u_1^l | h_1^l) \prod_{t=2}^{g_l} p_i(\nu_t^l) p_\sigma(h_t^l | h_{t-1}^l, \nu_t^l) p_o(u_t^l | h_t^l) . \quad (1)$$

This model, shown in Figure 1, is a specific Input/Output Hidden Markov Model (IOHMM) [12].

The model is described in the figure using the standard graphical model framework [13]. Usual IOHMMs have additional links connecting directly the input variables (level 1) to the outputs (level 3). We removed these links to decrease the number of parameters in the model, and thus being less prone to overfit the training data.

The probability distributions p_π , p_i , $p_{\bar{o}}$, and p_o are multinomials. The model is learned by the standard EM algorithm [14]. Marginalization must be carried out in this 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 usually tractable in IOHMMs because of their limited complexity. Performance of the IOHMM in terms of melodic prediction accuracy given chords is presented in Section 3.

2.2 Narmour Features

In this section, we introduce melodic features that will prove to be useful for melodic prediction. The Implication-Realization (I-R) model [5, 15] has been developed as a theory of musical expectation. This fairly complex musicological model was then simplified and implemented [16] as a formal analysis of each sequence of three consecutive notes, according to five perceptual items: registral direction, intervallic difference, registral return, proximity, and closure, as described later in this section. The model returns five scores measuring expectancy according to these five criteria, and, according to Narmour’s theory, high perceptual expectancy incurs high cumulative scores. This model was empirically shown to be relevant in information retrieval applications [11].

In this paper, our goal is quite different. Instead of quantifying melodic expectancy, we design a probabilistic model of melodic sequences given chords. We propose to collectively use the Narmour principles as discrete features to characterize each sequence of three consecutive notes. In the remainder of this paper, we refer to these features as *Narmour features*. There is much less possible Narmour features (108 in our implementation) than possible groups of three notes (128^3 if we consider all MIDI notes). Given that observation, we expect that modeling sequences of Narmour features should be easier than modeling actual sequences of notes. We describe in Section 2.3 how we propose to generate actual melodies given sequences of Narmour features.

Our particular implementation of the Narmour features is mostly derived from [16]. We simply define the *interval* v_t between two notes u_t and u_{t-1} to be the difference $v_t = u_{t-1} - u_t$ between their MIDI note numbers. Interval has to be taken here in its musicological sense, which is not related to the usual mathematical definition: an interval is an integer that counts the number of semi-tones between two notes. Each Narmour principle can be computed for any sequence of three consecutive notes, corresponding to two intervals. In Narmour’s theory, the first interval is referred to as the *Implication* while the second interval corresponds to the *Realization* of a melodic pattern of three notes. We define the sign function as

$$\text{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} .$$

The registral direction principle states that continuation in pitch direction is expected after small intervals and that large intervals imply a change of direction. We define

$$\text{rm}_t = \begin{cases} 0 & \text{if } |v_{t-1}| > 6 \text{ and } \text{sgn}(v_{t-1}) = \text{sgn}(v_t) \\ 1 & \text{if } |v_{t-1}| \leq 6 \\ 2 & \text{if } |v_{t-1}| > 6 \text{ and } \text{sgn}(v_{t-1}) \neq \text{sgn}(v_t) \end{cases}$$

to be the Narmour feature scoring the registral direction principle computed on arbitrary MIDI notes u_{t-2} , u_{t-1} , and u_t .

The intervallic difference principle says that small intervals imply similar-sized realized intervals and that large implicative intervals imply relatively smaller realized intervals. Formally,

$$\text{id}_t = \begin{cases} 1 & \text{if } |v_{t-1}| < 6 \text{ and } \text{sgn}(v_{t-1}) \neq \text{sgn}(v_t) \text{ and } ||v_{t-1}| - |v_t|| < 3 \\ 1 & \text{if } |v_{t-1}| < 6 \text{ and } \text{sgn}(v_{t-1}) = \text{sgn}(v_t) \text{ and } ||v_{t-1}| - |v_t|| < 4 \\ 1 & \text{if } |v_{t-1}| > 6 \text{ and } |v_{t-1}| \geq |v_t| \\ 0 & \text{otherwise} \end{cases}$$

is the Narmour feature scoring the intervallic difference principle.

The registral return principle states that the second tone of a realized interval is expected to be very similar to the original pitch (within 2 semi-tones). Thus, we define the following scoring function

$$\text{rr}_t = \begin{cases} 1 & \text{if } |v_t + v_{t-1}| \leq 2 \\ 0 & \text{otherwise.} \end{cases}$$

Then, the closure principle states that either melody changes direction, or that large intervals are followed by a relatively smaller interval. This feature is scored by

$$\text{cl}_t = \begin{cases} 2 & \text{if } \text{sgn}(v_{t-1}) \neq \text{sgn}(v_t) \text{ and } |v_{t-1}| - |v_t| > 2 \\ 1 & \text{if } \text{sgn}(v_{t-1}) \neq \text{sgn}(v_t) \text{ and } |v_{t-1}| - |v_t| < 3 \\ 1 & \text{if } \text{sgn}(v_{t-1}) = \text{sgn}(v_t) \text{ and } |v_{t-1}| - |v_t| > 3 \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the proximity principle favors small realized intervals. We define

$$\text{pr}_t = \begin{cases} 0 & \text{if } |v_t| \geq 6 \\ 1 & \text{if } 3 \leq |v_t| \leq 5 \\ 2 & \text{if } 0 \leq |v_t| \leq 2 . \end{cases}$$

We define this feature with less possible states than in [16] in order to limit the dimensionality of the Narmour representation. Besides, the actual numerical values for each of the Narmour features do not correspond to those of [16], where the goal was to quantify numerically the subjective melodic expectation. In the context of this paper, these values only correspond to discrete ordered values summarizing triplets of notes.

From these definitions, the Narmour features for the note triplet (u_{t-2}, u_{t-1}, u_t) are defined as

$$\gamma_t = (\text{rm}_t, \text{id}_t, \text{rr}_t, \text{cl}_t, \text{pr}_t) .$$

Such features have 108 possible different discrete states.

As an example, the sequence of MIDI notes $(u_1, u_2, u_3, u_4) = (71, 74, 72, 84)$ would lead to the Narmour features $\gamma_3 = (1, 1, 1, 1, 2)$ and $\gamma_4 = (1, 0, 0, 1, 0)$.

2.3 Melodic Model

In this section, we describe a probabilistic model for melodies given rhythms and chord progressions. While the IOHMM in Section 2.1 was directly modeling the choice of notes given chords, the model described here proceeds in two steps. We first model sequences of Narmour features given rhythm. Then, we model the actual choice of melodic notes, given sequences of Narmour features generated in the last step and chord progressions.

2.3.1 IOHMM for Narmour Features

An IOHMM like the one presented in Section 2.1 can be used to model sequences of Narmour features given rhythms. We first compress the rhythms in the dataset in a form that is synchronized with Narmour features: we define $\mathbf{a}^l = (a_2^l, \dots, a_{g_l-1}^l)$ to be the l -th sequence of note lengths in the dataset, ignoring the first and last note lengths a_1^l and $a_{g_l}^l$. Also, we denote by $\gamma^l = (\gamma_3^l, \dots, \gamma_{g_l}^l)$ the sequence of Narmour features associated to the l -th melody. This sequence starts with index 3 because each Narmour feature spans three notes.

The joint probability of each sequence of Narmour feature γ^l , its associated sequence of note lengths \mathbf{a}^l , and hidden states \mathbf{h}^l can be modeled by

$$p_{\text{NIOHMM}}(\mathbf{a}^l, \gamma^l, \mathbf{h}^l) = p_i(a_2^l) p_\pi(h_1^l | a_2^l) p_o(\gamma_3^l | h_1^l) \prod_{t=4}^{g_l} p_i(a_{t-1}^l) p_{\bar{o}}(h_{t-2}^l | h_{t-3}^l, a_{t-1}^l) p_o(\gamma_t^l | h_{t-2}^l) . \quad (2)$$

This model is shown in Figure 2. As in Equation (1), the probability distributions p_π , p_i , $p_{\bar{o}}$, and p_o are multinomials, and the model is learned by the standard EM algorithm.

As can be seen in Equation (2), we arbitrarily chose to condition the Narmour features on the *previous* note length. This is due to the empirical observation that greater intervals tend to occur after long notes while smaller intervals tend to occur after short notes. Other models of Narmour features given *current* length, a longer past context, or even no note length at all could be considered. We let this exploration for future work.

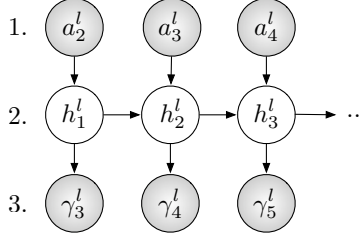


Figure 2: Variant of an IOHMM model for Narmour features given note lengths. The variables in level 1 are always observed and correspond to *previous* note lengths. Variables in level 2 are hidden, while variables in level 3 correspond to Narmour features. All variables in grey are observed during training.

2.3.2 Notes Model

We introduce in this section a model for MIDI notes given Narmour features and chord progressions. The combination of this model with the IOHMM for Narmour features introduced in the last section leads to a complete generative model of melodies given chord progressions.

We first decompose the chord representation defined in Section 2.1 into two parts: $\nu_i^l = (\eta_i^l, \tau_i^l)$, where η_i^l is the structure of the chord and τ_i^l is the root pitch class. Chord structures are just the chord definitions aside of the name of the root (e.g. “m7b5” is the chord structure in the chord “Bm7b5”). Each different chord structure is mapped to a specific state of the variables η_i^l . The sequences $\eta^l = (\eta_1^l, \dots, \eta_{g_t}^l)$ and $\tau^l = (\tau_1^l, \dots, \tau_{g_t}^l)$ are respectively the *chord structure* and the *root progressions* of the l -th song in the dataset.

Let \tilde{u}_t^l be an arbitrary MIDI note played at time t . We define

$$\phi(\tilde{u}_t^l, \tau_t^l) = ((\tilde{u}_t^l \bmod 12) - \tau_t^l) \bmod 12$$

to be the representation of the pitch class associated to the MIDI note \tilde{u}_t^l , *relative* to the root of the current chord. For instance, let $\tilde{u}_t^l = 65$ (note F) be played over the D minor chord. In that case, we have $\tau_t^l = 2$, meaning that the pitch class of the root of the chord is D. Hence, $\phi(65, 2) = 3$ for that particular example, meaning that the current melody note pitch class is 3 semi-tones higher than the root of the current chord.

It is easy to estimate $p(\eta_t^l | \tilde{u}_t^l, \tau_t^l)$ with a multinomial distribution conditioned on the values of $\phi(\tilde{u}_t^l, \tau_t^l)$. This distribution can be estimated by maximum likelihood over a training set. Hence, we learn a simple distribution of the chord structures η for each possible pitch classes of the melodies relative to the roots of the corresponding chords. For instance, this distribution could learn the fact that we often observe a minor seventh chord when playing a minor third over the tonic in the melody.

Let $\tilde{\gamma}_t^l(u_{t-2}^l, u_{t-1}^l, \tilde{u}_t^l)$ be the extracted Narmour feature when notes u_{t-2}^l and u_{t-1}^l are followed by the arbitrary note \tilde{u}_t^l . Also, let $\tilde{\kappa}_t^l$ be an arbitrary random variable such that

$$p(\tilde{\kappa}_t^l = 1 | \tilde{u}_t^l, u_{t-2}^l, u_{t-1}^l, \gamma_t^l) = \begin{cases} 1 & \text{if } \gamma_t^l = \tilde{\gamma}_t^l(u_{t-2}^l, u_{t-1}^l, \tilde{u}_t^l) \\ 0 & \text{otherwise.} \end{cases}$$

In words, $\tilde{\kappa}_t^l$ is equal to 1 if and only if the Narmour feature produced when playing arbitrary note \tilde{u}_t^l is equal to the *given* Narmour feature γ_t^l , given the two previous notes.

We define a factorization of the joint probability of the variables $\tilde{u}_t^l, u_{t-1}^l, u_{t-2}^l, \eta_t^l, \tau_t^l, \gamma_t^l$, and $\tilde{\kappa}_t^l$ with

$$p(\tilde{u}_t^l, u_{t-1}^l, u_{t-2}^l, \eta_t^l, \tau_t^l, \gamma_t^l, \tilde{\kappa}_t^l) = p(u_{t-1}^l) p(u_{t-2}^l) p(\gamma_t^l) p(\tilde{\kappa}_t^l | \tilde{u}_t^l, u_{t-2}^l, u_{t-1}^l, \gamma_t^l) p(\tilde{u}_t^l) p(\tau_t^l) p(\eta_t^l | \tilde{u}_t^l, \tau_t^l) \quad (3)$$

at each time t . This factorization is shown by the graphical model in Figure 3.

We want to estimate the probability of playing any arbitrary MIDI note \tilde{u}_t^l at time t in the l -th song of the dataset given the two previous observed notes u_{t-2}^l and u_{t-1}^l , the current Narmour feature γ_t^l ,

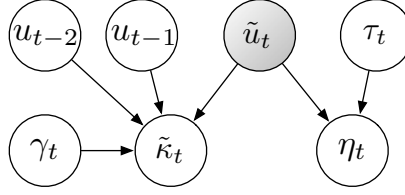


Figure 3: Graphical model representation of the factorization of the joint probability defined in Eq. (3).

and the current chord $\nu_t^l = (\eta_t^l, \tau_t^l)$. Given the factorization in Equation (3), we have that

$$p_{MEL}(\tilde{u}_t^l | u_{t-1}^l, u_{t-2}^l, \eta_t^l, \tau_t^l, \gamma_t^l, \tilde{\kappa}_t^l = 1) = \frac{p(\tilde{\kappa}_t^l = 1 | \tilde{u}_t^l, u_{t-2}^l, u_{t-1}^l, \gamma_t^l) p(\tilde{u}_t^l) p(\eta_t^l | \tilde{u}_t^l, \tau_t^l)}{\sum_{\tilde{u}_t^l} p(\tilde{\kappa}_t^l = 1 | \tilde{u}_t^l, u_{t-2}^l, u_{t-1}^l, \gamma_t^l) p(\tilde{u}_t^l) p(\eta_t^l | \tilde{u}_t^l, \tau_t^l)} \quad (4)$$

where $p(\tilde{u}_t^l)$ is the prior probability of observing \tilde{u}_t^l . The distribution $p(\tilde{u}_t^l)$ is a multinomial that can be simply estimated by maximum likelihood on the training set.

Hence, a simple strategy to find the most likely MIDI note \tilde{u}_t^l given $u_{t-1}^l, u_{t-2}^l, \eta_t^l, \tau_t^l$, and γ^l is to solve

$$\arg \max_{\{\tilde{u}_t^l | \tilde{\kappa}_t^l = 1, u_{t-1}^l, u_{t-2}^l, \gamma^l\}} p(\tilde{u}_t^l) p(\eta_t^l | \tilde{u}_t^l, \tau_t^l),$$

since the denominator in the right-hand side of Equation (4) is the same for all values of \tilde{u}_t^l . In other words, we search for the most likely melodic note (with respect to the current chord) among all the possible notes given the current Narmour constraint and the current chord. Despite the fact that this model only predicts one note at a time, it is able to take into account longer term melodic shapes through the constraints imposed by the sequences of Narmour features.

Melodic prediction *without* observing Narmour features can be done with this model in two steps. We first generate the most likely sequence of Narmour features given rhythms with the IOHMM model described in Section 2.3.1. Then, we can use the melodic prediction model described in the current section to predict MIDI notes given chord progressions. Such a model is shown in Section 3 to have much better prediction accuracy than using a simpler IOHMM model alone.

One can listen the the audio examples provided with this submission as additional material. Even for the non musician, it should be obvious that the sequences generated by sampling the melodic model introduced in this section are much more realistic than sequences generated by sampling the IOHMM model described in Section 2.1. Both models generate notes that are coherent with the current chord. However, the sequences generated by the IOHMM model do not have any coherent temporal structure. On the other hand, melodies generated by the melodic model presented here tend to follow the same melodic shapes than the songs in the training sets. These melodic shapes are constrained by the conditioning sequences of Narmour features used as inputs.

3 Melodic Prediction Experiments

Two databases from different musical genres were used to evaluate the proposed model. Firstly, 47 jazz standards melodies [17] were interpreted and recorded by the first author in MIDI format. The complexity of the melodies and chord progressions found in this corpus is representative of the complexity of common jazz and pop music. We used the last 16 bars of each song to train the models, with four beats per bar. We also used a subset of the Nottingham database² consisting of 53 traditional British folk dance tunes called ‘‘hornpipes’’. In this case, we used the first 16 bars of each song to train the models, with four beats per bar.

The goal of the proposed models is to predict or generate melodies *given* chord progressions and rhythms. Let $\mathbf{u}^j = (u_1^j, \dots, u_{g_j}^j)$ be a test sequence of MIDI notes and \hat{u}_t^j to be the output of

²<http://www.cs.nott.ac.uk/~ef/music/database.htm>.

Table 1: Accuracies (the higher the better) achieved by both models on the two databases, for various prediction starting bars s (all the songs contain 16 bars in the experiments).

s	Jazz		Hornpipes	
	IOHMM	Narmour	IOHMM	Narmour
5	2.0%	8.9%	2.5%	4.6%
9	1.7%	8.1%	2.6%	4.8%
13	2.2%	8.3%	2.6%	4.9%

the evaluated prediction model on the i -th position when given $(u_1^j, \dots, u_{i-1}^j)$ and the associated rhythm sequence \mathbf{x}^j . Assume that the dataset is divided into K folds T_1, \dots, T_K (each containing different sequences), and that the k -th fold T_k contains n_k test sequences. Finally, let s be the first bar from which the evaluated model try to guess what would be the next notes in each test songs. When using cross-validation, we define the “prediction accuracy” Acc of an evaluated model to be

$$Acc = \frac{1}{K} \sum_{k=1}^K \frac{1}{n_k} \sum_{j \in T_k} \frac{1}{g_j - \zeta_s^j + 1} \sum_{i=\zeta_s^j}^{g_j} \tilde{\varepsilon}_i^j \quad (5)$$

where $\tilde{\varepsilon}_i^j = 1$ if $\hat{u}_i^j = u_i^j$, and 0 otherwise, and ζ_s^j is the smallest note index in bar s of the j -th song. Hence, prediction models have access to *all* the previously observed notes when trying to guess what would be the next note. This rate of success is averaged on the last $16 - s + 1$ bars of each song (which correspond to the last $g_j - \zeta_s^j + 1$ notes of each song.)

Between 2 to 20 possible hidden states were tried in the reported experiments for the baseline IOHMM model of Section 2.1 and the “Narmour” IOHMM of Section 2.3.1. Both models try to predict out-of-sample melody notes, given chord progressions and complete test rhythm sequences \mathbf{x}^j . The same chord representations are used as input for both models. 5-fold cross-validation was used to compute prediction accuracies. We report results for the choices of parameters that provided the highest accuracies for each model. The IOHMM model of notes given chords is a stronger contender than would be a simpler HMM trained on melodies, because the prediction given by the IOHMM takes advantage of the current input.

Results in Table 1 for the jazz standards database show that generating Narmour features as an intermediate step greatly improves prediction accuracy. Since there is 128 different MIDI notes, a completely random predictor would have a local accuracy of 0.8%. Both models take into account chord progressions when trying to predict the next MIDI note. However, the Narmour model favors melodic shapes similar to the ones found in the training set.

The Narmour model still provides consistently better prediction accuracy on the hornpipes database, as can be seen in the same table. However, prediction accuracies are lower on the hornpipes database than on the jazz database for the Narmour model. Note onsets occur on most rhythm positions in this database. This means that rhythm sequences in this database have relatively low entropy. Hence, rhythm sequences are less informative when used as conditioning inputs to generate sequences of Narmour features. Another observation is that the chord structures in this database are almost always the same (i.e. simple triads). The melodic model of Section 2.3 is directly modeling the distribution $p(\eta_t^l | \tau_t^l, \tilde{u}_t^l)$ of relative MIDI notes given chord structures. This distribution was probably more helpful for melodic prediction in the jazz database than in the hornpipes database. Despite these two drawbacks, the melodic model of Section 2.3 has a prediction accuracy twice as good as what is obtained with the simpler IOHMM model in the hornpipes database.

While the prediction accuracy is simple to compute and to apprehend, other performance criteria, such as ratings provided by a panel of experts, could be more appropriate to evaluate the relevance of music models. The fact that the Narmour model accurately predict “only” about 8% of the notes on out-of-sample sequences does not mean that it is not performing well when generating the other “wrong” notes. Many realistic melodies can be generated on the same chord progression in a given musical genre. Moreover, some mistakes are more harmful than others. For most applications, a model that would have very low prediction accuracy, but that would generate realistic melodies, would be preferable to a model with 50% prediction accuracy, but that would generate unrealistic notes the other half of the time.

4 Conclusion

The main contribution of this paper is the design and evaluation of a generative model for melodies. While a few generative models have already been proposed for music in general [4, 1], we are not aware of quantitative comparisons between generative models of music. We first described melodic features [5] that put useful constraints on melodies based on musicological substantiation. We then defined a probabilistic model of melodies that provides significantly higher prediction rates than a simpler, yet powerful, Markovian model. Furthermore, sampling the proposed model given appropriate musical contexts generates realistic melodies.

References

- [1] F. Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, September 2003.
- [2] D. Espi, P.J. Ponce de Leon, C. Perez-Sancho, D. Rizo, J.M. Inesta, F. Moreno-Seco, and A. Pertusa. A cooperative approach to style-oriented music composition. In *Proc. of the Int. Workshop on Artificial Intelligence and Music, MUSIC-AI*, pages 25–36, Hyderabad, India, 2007.
- [3] V. Lavrenko and J. Pickens. Polyphonic music modeling with random fields. In *Proceedings of ACM Multimedia*, pages 120–129, Berkeley, CA, November 2-8 2003.
- [4] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. Using machine-learning methods for musical style modeling. *IEEE Computer*, 10(38), October 2003.
- [5] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. Univeristy of Chicago Press, Chicago, 1990.
- [6] Kyogu Lee and Malcolm Slaney. A unified system for chord transcription and key extraction using hidden markov models. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2007.
- [7] M. Allan and C. K. I. Williams. Harmonising chorales by probabilistic inference. In *Advances in Neural Information Processing Systems*, volume 17, 2004.
- [8] J.-F. Paiement, D. Eck, and S. Bengio. Probabilistic melodic harmonization. In *Proceedings of the 19th Canadian Conference on Artificial Intelligence*, pages 218–229. Springer, 2006.
- [9] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.
- [10] Douglas Eck and Juergen Schmidhuber. Finding temporal structure in music: Blues improvisation with LSTM recurrent networks. In H. Bourlard, editor, *Neural Networks for Signal Processing XII, Proc. 2002 IEEE Workshop*, pages 747–756, New York, 2002. IEEE.
- [11] M. Grachten, J. LI. Arcos, and R. Lopez de Mantaras. Melody retrieval using the implication/realization model. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005.
- [12] Y. Bengio and P. Frasconi. Input/output HMMs for sequence processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, 1996.
- [13] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- [14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.
- [15] Eugene Narmour. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. University of Chicago Press, 1992.
- [16] E. Schellenberg. Simplifying the implication-realization model of musical expectancy. *Music Perception*, 14(3):295–318, 1997.
- [17] Chuck Sher, editor. *The New Real Book*, volume 1-3. Sher Music Co., 1988.