A Distance Model for Rhythms

Jean-Francois Paiement [a] Yves Grandvalet [a]
Samy Bengio [b] Douglas Eck [c]

IDIAP–RR 08-33

May 2008

published in
J.-F. Paiement, Y. Grandvalet, S. Bengio, and D. Eck. A Distance Model for Rhythms. The 25th International Conference on Machine Learning (ICML 2008).

[a] IDIAP Research Institute, Case Postale 592, CH-1920 Martigny, Switzerland
[b] Google, 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
[c] Université de Montréal, Department of Computer Science, CP 6128, Succ. Centre-Ville, Montréal, Québec H3C 3J7, Canada

IDIAP Research Report 08-33

A Distance Model for Rhythms

Jean-Francois Paiement Yves Grandvalet Samy Bengio Douglas Eck

May 2008

published in
J.-F. Paiement, Y. Grandvalet, S. Bengio, and D. Eck. A Distance Model for Rhythms. The 25th International Conference on Machine Learning (ICML 2008).

Abstract. Modeling long-term dependencies in time series has proved very difficult to achieve with traditional machine learning methods. This problem occurs when considering music data. In this paper, we introduce a model for rhythms based on the distributions of distances between subsequences. A specific implementation of the model when considering Hamming distances over a simple rhythm representation is described. The proposed model consistently outperforms a standard Hidden Markov Model in terms of conditional prediction accuracy on two different music databases.

1 Introduction

Reliable models for 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, which has proved very difficult to achieve with traditional statistical methods. Note that the problem of long-term dependencies is not limited to music, nor to one particular probabilistic model (Bengio et al., 1994).

Music is characterized by strong hierarchical dependencies determined in large part by *meter*, the sense of strong and weak beats that arises from the interaction among hierarchical levels of sequences having nested periodic components. Such a hierarchy is implied in western music notation, where different levels are indicated by kinds of notes (whole notes, half notes, quarter notes, etc.) and where bars establish measures of an equal number of beats. Meter and rhythm provide a framework for developing musical melody. For example, a long melody is often composed by repeating with variation shorter sequences that fit into the metrical hierarchy (e.g. sequences of 4, 8 or 16 measures). It is well know in music theory that *distance patterns* are more important than the actual choice of notes in order to create coherent music (Handel, 1993). In this work, distance patterns refer to distances between subsequences of equal length in particular positions. For instance, measure 1 may be always similar to measure 5 in a particular musical genre. In fact, even random music can sound structured and melodic if it is built by repeating random subsequences with slight variation.

Many algorithms have been proposed for audio beat tracking (Dixon, 2007; Scheirer, 1998). Probabilistic models have also been proposed for tempo tracking and inference of rhythmic structure in musical audio (Whiteley et al., 2007; Cemgil & Kappen, 2002). The goal of these models is to align rhythm events with the metrical structure. However, simple Markovian assumptions are used to model the transitions between rhythms themselves. Hence, these models do not take into account long-term dependencies. A few generative models have already been proposed for music in general (Pachet, 2003; Dubnov et al., 2003). While these models generate impressive musical results, we are not aware of quantitative comparisons between models of music with machine learning standards, as it is done in Section 3 in terms of out-of-sample prediction accuracy. In this paper, we focus on modeling rhythmic sequences, ignoring for the moment other aspects of music such as pitch, timbre and dynamics. However, by capturing aspects of global temporal structure in music, this model should be valuable for full melodic prediction and generation: combined with an audio transcription algorithm, it should help improve the poor performance of state-of-the-art transcription systems; it could as well be included in genre classifiers or automatic composition systems (Eck & Schmidhuber, 2002); used to generate rhythms, the model could act as a drum machine or automatic accompaniment system which learns by example.

Our main contribution is to propose a generative model for distance patterns, specifically designed for capturing long-term dependencies in rhythms. In Section 2, we describe the model, detail its implementation and present an algorithm using this model for rhythm prediction. The algorithm solves a constrained optimization problem, where the distance model is used to filter out rhythms that do not comply with the inferred structure. The proposed model is evaluated in terms of conditional prediction error on two distinct databases in Section 3 and a discussion follows.

2 Distance Model

In this Section, we present a generative model for distance patterns and its application to rhythm sequences. Such a model is appropriate for most music data, where distances between subsequences of data exhibit strong regularities.

2.1 Motivation

Let $\mathbf{x}^l = (x_1^l, \ldots, x_m^l) \in \mathbb{R}^m$ be the l-th rhythm sequence in a dataset $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ where all the sequences contain m elements. Suppose that we construct a *partition* of this sequence by dividing it into ρ parts defined by $y_i^l = (x_{1+(i-1)m/\rho}^l, \ldots, x_{im/\rho}^l)$ with $i \in \{1, \ldots, \rho\}$. We are interested in modeling the distances between these subsequences, given a suitable metric $d(y_i, y_j) : \mathbb{R}^{m/\rho} \times \mathbb{R}^{m/\rho} \rightarrow \mathbb{R}$. As was pointed out in Section 1, the distribution of $d(y_i, y_j)$ for each specific choice of i and j may be more important when modeling rhythms (and music in general) than the actual choice of subsequences y_i.

Hidden Markov Models (HMM) (Rabiner, 1989) are commonly used to model temporal data. In principle, an HMM is able to capture complex regularities in patterns between subsequences of data, provided its number of hidden states is large enough. However, when dealing with music, such a model would lead to a learning process requiring a prohibitive amount of data: in order to learn long range interactions, the training set should be representative of the joint distribution of subsequences. To overcome this problem, we summarize the joint distribution of subsequences by the distribution of distances between these subsequences. This summary is clearly not a sufficient statistics for the distribution of subsequences, but its distribution can be learned from a limited number of examples. The resulting model, which generates distances, is then used to recover subsequences.

2.2 Decomposition of Distances

Let $D(\mathbf{x}^l) = (d_{i,j}^l)_{\rho \times \rho}$ be the distance matrix associated with each sequence $\mathbf{x}^l$, where $d_{i,j}^l = d(y_i^l, y_j^l)$. Since $D(\mathbf{x}^l)$ is symmetric and contains only zeros on the diagonal, it is completely characterized by the upper triangular matrix of distances *without* the diagonal. Hence,

$$p(D(\mathbf{x}^l)) = \prod_{i=1}^{\rho-1} \prod_{j=i+1}^{\rho} p(d_{i,j}^l | S_{l,i,j}) \tag{1}$$

where

$$\begin{array}{rl} S_{l,i,j} = \{d_{r,s}^l | & (1 < s < j \text{ and } 1 \leq r < s) \\ \text{or} & (s = j \text{ and } 1 \leq r < i)\} \ . \end{array} \tag{2}$$

In words, we order the elements column-wise and do a standard factorization, where each random variable depends on the previous elements in the ordering. Hence, we do not assume any conditional independence between the distances.

Since $d(y_i, y_j)$ is a metric, we have that $d(y_i, y_j) \leq d(y_i, y_k) + d(y_k, y_j)$ for all $i, j, k \in \{1, \ldots, \rho\}$. This inequality is usually referred to as the *triangle inequality.* Defining

$$\begin{array}{rcl} \alpha_{i,j}^l & = & \min\limits_{k \in \{1,\ldots,(i-1)\}} (d_{k,j}^l + d_{i,k}^l) \text{ and} \\ \beta_{i,j}^l & = & \max\limits_{k \in \{1,\ldots,(i-1)\}} (|d_{k,j}^l - d_{i,k}^l|) \ , \end{array} \tag{3}$$

we know that given previously observed (or sampled) distances, constraints imposed by the triangle inequality on $d_{i,j}^l$ are simply

$$\beta_{i,j}^l \leq d_{i,j}^l \leq \alpha_{i,j}^l \ . \tag{4}$$

One may observe that the boundaries given in Eq. (3) contain a subset of the distances that are on the conditioning side of each factor in Eq. (1) for each indexes i and j. Thus, constraints imposed by the triangle inequality can be taken into account when modeling each factor of $p(D(\mathbf{x}^l))$: each $d_{i,j}^l$ must lie in the interval imposed by previously observed/sampled distances given in Eq. (4). Figure 1 shows an example where $\rho = 4$. Using Eq. (1), the distribution of $d_{2,4}^l$ would be conditioned on $d_{1,2}^l$, $d_{1,3}^l$, $d_{2,3}^l$, and $d_{1,4}^l$, and Eq. (4) reads $|d_{1,2}^l - d_{1,4}^l| \leq d_{2,4}^l \leq d_{1,2}^l + d_{1,4}^l$. Then, if subsequences y_1^l and y_2^l are close and y_1^l and y_4^l are also close, we know that y_2^l and y_4^l cannot be far. Conversely, if subsequences y_1^l and y_2^l are far and y_1^l and y_4^l are close, we know that y_2^l and y_4^l cannot be close.

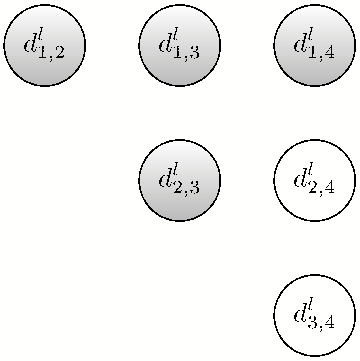

Figure 1: Each circle represents the random variable associated with the corresponding factor in Eq. (1), when $\rho = 4$. For instance, the conditional distribution for $d^l_{2,4}$ possibly depends on the variables associated to the grey circles.

2.3 Modeling Relative Distances Between Rhythms

We want to model rhythms in a music dataset $\mathcal{X}$ consisting of melodies of the same musical genre. We first quantize the database by segmenting each song in m time steps and associate each note to the nearest time step, such that all melodies have the same length m[1]. It is then possible to represent rhythms by sequences containing potentially three different symbols: 1) Note onset, 2) Note continuation, and 3) Silence. When using quantization, there is a one to one mapping between this representation and the set of all possible rhythms. Using this representation, symbol 2 can never follow symbol 3. Let $A = \{1, 2, 3\}$; in the remaining of this paper, we assume that $\mathbf{x}^l \in A^m$ for all $\mathbf{x}^l \in \mathcal{X}$.

When using this representation, $d^l_{i,j}$ can simply be chosen to be the Hamming distance (i.e. counting the number of positions on which corresponding symbols are different.) One could think of using more general edit distance such as the Levenshtein distance. However, this approach would not make sense psycho-acoustically: doing an insertion or a deletion in a rhythm produces a translation that alters dramatically the nature of the sequence. Putting it another way, rhythm perception heavily depends on the *position* on which rhythmic events occur. In the remainder of this paper, $d^l_{i,j}$ is the Hamming distance between subsequences y_i and y_j.

We now have to encode our belief that melodies of the same musical genre have a common distance structure. For instance, drum beats in rock music can be very repetitive, except in the endings of every four measures, without regard to the actual beats being played. This should be accounted for in the distributions of the corresponding $d^l_{i,j}$. With Hamming distances, the conditional distributions of $d^l_{i,j}$ in Eq. (1) should be modeled by discrete distributions, whose range of possible values *must* obey Eq. (4). Hence, we assume that the random variables $(d^l_{i,j} - \beta^l_{i,j})/(\alpha^l_{i,j} - \beta^l_{i,j})$ should be identically distributed for $l = 1, \ldots, n$. As an example, suppose that measures 1 and 4 always tend to be far away, that measures 1 and 3 are close, and that measures 3 and 4 are close; Triangle inequality states that 1 and 4 should be close in this case, but the desired model would still favor a solution with the greatest distance possible *within* the constrains imposed by triangle inequalities.

All these requirements are fulfilled if we model $d_{i,j} - \beta_{i,j}$ by a binomial distribution of parameters $(\alpha_{i,j} - \beta_{i,j}, p_{i,j})$, where $p_{i,j}$ is the probability that two symbols of subsequences y_i and y_j differ. With

[1] This hypothesis is not fundamental in the proposed model and could easily be avoided if one would have to deal with more general datasets.

this choice, the conditional probability of getting $d_{i,j} = \beta_{i,j} + \delta$ would be

$$B(\delta, \alpha_{i,j}, \beta_{i,j}, p_{i,j}) = \binom{\alpha_{i,j} - \beta_{i,j}}{\delta} (p_{i,j})^{\delta} (1 - p_{i,j})^{(\alpha_{i,j} - \beta_{i,j} - \delta)} \ , \quad (5)$$

with $0 \leq p_{i,j} \leq 1$. If $p_{i,j}$ is close to zero/one, the relative distance between subsequences y_i and y_j is small/large. However, the binomial distribution is not flexible enough since there is no indication that the distribution of $d_{i,j} - \beta_{i,j}$ is unimodal. We thus model each $d_{i,j} - \beta_{i,j}$ with a binomial *mixture* distribution in order to allow multiple modes. We thus use

$$p(d_{i,j} = \beta_{i,j} + \delta | S_{i,j}) = \sum_{k=1}^{c} w_{i,j}^{(k)} B(\delta, \alpha_{i,j}, \beta_{i,j}, p_{i,j}^{(k)}) \quad (6)$$

with $w_{i,j}^{(k)} \geq 0$, $\sum_{k=1}^{c} w_{i,j}^{(k)} = 1$ for every indexes i and j, and $S_{i,j}$ defined similarly as in Eq. (2). Parameters

$$\theta_{i,j} = \{w_{i,j}^{(1)}, \ldots, w_{i,j}^{(c-1)}\} \cup \{p_{i,j}^{(1)}, \ldots, p_{i,j}^{(c)}\}$$

can be learned with the EM algorithm (Dempster et al., 1977) on rhythm data for a specific music style.

In words, we model the *difference* between the observed distance $d_{i,j}^{l}$ between two subsequences and the minimum possible value $\beta_{i,j}$ for such a difference by a binomial mixture.

The parameters $\theta_{i,j}$ can be initialized to arbitrary values before applying the EM algorithm. However, as the likelihood of mixture models is not a convex function, one may get better models and speed up the learning process by choosing sensible values for the initial parameters. In the experiments reported in Section 3, the k-means algorithm for clustering (Duda et al., 2000) was used. More precisely, k-means was used to partition the values $(d_{i,j}^{l} - \beta_{i,j}^{l})/(\alpha_{i,j}^{l} - \beta_{i,j}^{l})$ into c clusters corresponding to each component of the mixture in Eq. (6). Let $\{\mu_{i,j}^{(1)}, \ldots, \mu_{i,j}^{(c)}\}$ be the centroids and $\{n_{i,j}^{(1)}, \ldots, n_{i,j}^{(c)}\}$ the number of elements in each of these clusters. We initialize the parameters $\theta_{i,j}$ with

$$w_{i,j}^{(k)} = \frac{n_{i,j}^{(k)}}{n} \quad \text{and} \quad p_{i,j}^{(k)} = \mu_{i,j}^{(k)} .$$

We then follow a standard approach (Bilmes, 1997) to apply the EM algorithm to the binomial mixture in Eq. (6). Let $z_{i,j}^{l} \in \{1, \ldots, c\}$ be a hidden variable telling which component density generated $d_{i,j}^{l}$. For every iteration of the EM algorithm, we first compute

$$p(z_{i,j}^{l} = k | d_{i,j}^{l}, \alpha_{i,j}^{l}, \beta_{i,j}^{l}, \hat{\theta}_{i,j}) = \frac{\psi_{k,i,j,l}}{\sum_{t=1}^{c} \psi_{t,i,j,l}}$$

where $\hat{\theta}_{i,j}$ are the parameters estimated in the previous iteration, or the parameters guessed with k-means on the first iteration of EM, and

$$\psi_{k,i,j,l} = \hat{w}_{i,j}^{(k)} B(d_{i,j}^{l}, \alpha_{i,j}^{l}, \beta_{i,j}^{l}, p^{(k)}) \ .$$

Then, the parameters can be updated with

$$p_{i,j}^{(k)} = \frac{\sum_{l=1}^{n} (d_{i,j}^{l} - \beta_{i,j}^{l}) p(z_{i,j}^{l} = k | d_{i,j}^{l}, \alpha_{i,j}^{l}, \beta_{i,j}^{l}, \hat{\theta}_{i,j})}{\sum_{l=1}^{n} (\alpha_{i,j}^{l} - \beta_{i,j}^{l}) p(z_{i,j}^{l} = k | d_{i,j}^{l}, \alpha_{i,j}^{l}, \beta_{i,j}^{l}, \hat{\theta}_{i,j})}$$

and

$$w_{i,j}^{(k)} = \frac{1}{n} \sum_{l=1}^{n} p(z_{i,j}^{l} = k | d_{i,j}^{l}, \alpha_{i,j}^{l}, \beta_{i,j}^{l}, \hat{\theta}_{i,j}).$$

This process is repeated until convergence.

Note that using mixture models for discrete data is known to lead to *identifiability* problems. Identifiability refers here to the uniqueness of the representation (up to an irrelevant permutation of parameters) of any distribution that can be modeled by a mixture.

Estimation procedures may not be well-defined and asymptotic theory may not hold if a model is not identifiable. However, the model defined in Eq. (6) is identifiable if $\alpha_{i,j} - \beta_{i,j} > 2c-1$ (Titterington et al., 1985, p.40). While this is the case for most $d_{i,j}$, we observed that this condition is sometimes violated. Whatever happens, there is no impact on the estimation because we only care about what happens at the distribution level: there may be several parameters leading to the same distribution, some components may vanish in the fitting process, but this is easily remedied, and EM behaves well.

As stated in Section 1, musical patterns form hierarchical structures closely related to meter (Handel, 1993). Thus, the distribution of $p(D(\mathbf{x}^l))$ can be computed for many numbers of partitions within each rhythmic sequence. Let $\mathcal{P} = \{\rho_1, \ldots \rho_h\}$ be a set of numbers of partitions to be considered by our model, where h is the number of such numbers of partitions. The choice of $\mathcal{P}$ depends on the domain of application. Following meter, $\mathcal{P}$ may have dyadic[2] tree-like structure when modeling music (e.g. $\mathcal{P} = \{2, 4, 8, 16\}$). Let $D_{\rho_r}(\mathbf{x}^l)$ be the distance matrix associated with sequence $\mathbf{x}^l$ divided into ρ_r parts. Estimating the joint probability $\prod_{r=1}^{h} p(D_{\rho_r}(\mathbf{x}^l))$ with the EM algorithm as described in this section leads to a model of the distance structures in music datasets. Suppose we consider 16 bars songs with four beats per bar. Using $\mathcal{P} = \{8, 16\}$ would mean that we consider pairs of distances between every group of two measures ($\rho = 8$), and every single measures ($\rho = 16$).

One may argue that our proposed model for long-term dependencies is rather unorthodox. However, simpler models like Poisson or Bernoulli process (we are working in discrete time) defined over the whole sequence would not be flexible enough to represent the particular long-term structures in music.

2.4 Conditional Prediction

For most music applications, it would be particularly helpful to know which sequence $\hat{x}_s, \ldots, \hat{x}_m$ maximizes $p(\hat{x}_s, \ldots, \hat{x}_m | x_1, \ldots, x_{s-1})$. Knowing which musical events are the most likely given the past $s-1$ observations would be useful both for prediction and generation. Note that in the remaining of the paper, we refer to prediction of musical events given past observations only for notational simplicity. The distance model presented in this paper could be used to predict any part of a music sequence given any other part with only minor modifications.

While the described modeling approach captures long range interactions in the music signal, it has two shortcomings. First, it does not model local dependencies: it does not predict how the distances in the smallest subsequences (i.e. with length smaller than $m/\max(\mathcal{P})$) are distributed on the events contained in these subsequences. Second, as the mapping from sequences to distances is many to one, there exists several admissible sequences $\mathbf{x}^l$ for a given set of distances. These limitations are addressed by using another sequence learner designed to capture short-term dependencies between musical events. Here, we use a standard Hidden Markov Model (HMM) (Rabiner, 1989) displayed in Figure 2, following standard graphical model formalism. Each node is associated to a random variable and arrows denote conditional dependencies. Learning the parameters of the HMM can be done as usual with the EM algorithm.

The two models are trained separately using their respective version of the EM algorithm. For predicting the continuation of new sequences, they are combined by choosing the sequence that is most likely according to the local HMM model, provided it is also plausible regarding the model of long-term dependencies. Let $p_{\text{HMM}}(\mathbf{x}^l)$ be the probability of observing sequence $\mathbf{x}^l$ estimated by the HMM after training. The final predicted sequence is the solution of the following optimization

[2] Even when considering non-dyadic measures (e.g. a three-beat waltz), the *very large* majority of the hierarchical levels in metric structures follow dyadic patterns (Handel, 1993) in most tonal music.

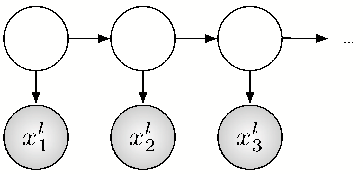

Figure 2: Hidden Markov Model. Each node is associated to a random variable and arrows denote conditional dependencies. During training of the model, white nodes are hidden while grey nodes are observed.

problem:

$$\begin{cases} \max\limits_{\tilde{x}_s,\ldots,\tilde{x}_m} & p_{\mathrm{HMM}}(\tilde{x}_s,\ldots,\tilde{x}_m|x_1,\ldots,x_{s-1}) \\ \text{subject to} & \prod\limits_{r=1}^{h} p(D_{\rho_r}(\mathbf{x}^l)) \geq P_0 \ , \end{cases} \tag{7}$$

where P_0 is a threshold. In practice, one solves a Lagrangian formulation of problem (7), where we use log-probabilities for obvious computational reasons:

$$\begin{array}{l} \max_{\tilde{x}_s,\ldots,\tilde{x}_m}[\log p_{\mathrm{HMM}}(\tilde{x}_s,\ldots,\tilde{x}_m|x_1,\ldots,x_{s-1}) \\ \qquad\qquad\qquad +\lambda\sum_{r=1}^{h}\log p(D_{\rho_r}(\mathbf{x}^l))] \ , \end{array} \tag{8}$$

where tuning λ has the same effect as choosing a threshold P_0 in Eq. (7) and can be done by cross-validation.

Multidimensional Scaling (MDS) is an algorithm that tries to embed points (here “local” subsequences) into a potentially lower dimensional space while trying to be faithful to the pairwise affinities given by a “global” distance matrix. Here, we propose to consider the prediction problem as finding sequences that maximize the likelihood of a “local” model of subsequences under the constraints imposed by a “global” generative model of distances between subsequences. In other words, solving problem (7) is similar to finding points between which distances are as close as possible to a given set of distances (i.e. minimizing a stress function in MDS). Naively trying all possible subsequences to maximize (8) leads to $O(|A|^{(m-s+1)})$ computations. Instead, we propose to search the space of sequences using a variant of the Greedy Max Cut (GMC) method (Rohde, 2002) that has proven to be optimal in terms of running time and performance for binary MDS optimization.

The subsequence $\hat{x}_s,\ldots,\hat{x}_m$ can be simply initialized with

$$(\hat{x}_s,\ldots,\hat{x}_m) = \max_{\tilde{x}_s,\ldots,\tilde{x}_m} p_{\mathrm{HMM}}(\tilde{x}_s,\ldots,\tilde{x}_m|x_1,\ldots,x_{s-1}) \tag{9}$$

using the local HMM model. The complete optimization algorithm is described in Figure 3. For each position, we try every admissible symbol of the alphabet and test if a change increases the probability of the sequence. We stop when no further change can increase the value of the utility function. Obviously, many other methods could have been used to search the space of possible sequences $\hat{x}_s,\ldots,\hat{x}_m$, such as simulated annealing (Kirkpatrick et al., 1983). We chose the algorithm in Figure 3 for its simplicity and the fact that it yields excellent results, as reported in the following section.

3 Experiments

Two rhythm databases from different musical genres were used to evaluate the proposed model. Firstly, 47 jazz standards melodies (Sher, 1988) were interpreted and recorded by the first author in

```
1. Initialize $\hat{x}_s, \dots, \hat{x}_m$ using Eq. (9);
2. Set $j = s$ and set end = true;
3. Set $\hat{x}_j = \arg\max_{a \in A} \log p_{\text{HMM}}(\hat{x}_s, \dots, \hat{x}_{j-1}, a, \hat{x}_{j+1}, \dots, \hat{x}_m | x_1, \dots, x_{s-1}) +$
       $\lambda \sum_{r=1}^{h} \log p(D_{\rho_r}(\mathbf{x}^*))$
   where $\mathbf{x}^* = (x_1, \dots, x_{s-1}, \hat{x}_s, \dots, \hat{x}_{j-1}, a, \hat{x}_{j+1}, \dots, \hat{x}_m)$.
4. If $\hat{x}_j$ has been modified in the last step, set end = false.
5. If $j = m$ and end = false, go to 2;
6. If $j < m$, set $j = j + 1$ and go to 3;
7. Return $\hat{x}_s, \dots, \hat{x}_m$.
```

Figure 3: Simple optimization algorithm to maximize $p(\hat{x}_i, \dots, \hat{x}_m | x_1, \dots, x_{i-1})$

MIDI format. Appropriate rhythmic representations as described in Section 2.3 have been extracted from these files. The complexity of the rhythm sequences 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. Two rhythmic observations were made for each beat, yielding observed sequences of length 128. We also used a subset of the Nottingham database [3] 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. Three rhythmic observations were made for each beat, yielding observed sequences of length 192. The sequences from this second database contain no silence (i.e. rests), leading to sequences with binary states.

The goal of the proposed model is to predict or generate rhythms *given* previously observed rhythm patterns. As pointed out in Section 1, such a model could be particularly useful for music information retrieval, transcription, or music generation applications. Let $\varepsilon_i^t = 1$ if $\hat{x}_i^t = x_i^t$, and 0 otherwise, with $\mathbf{x}^t = (x_1^t, \dots, x_m^t)$ a test sequence, and $\hat{x}_i^t$ the output of the evaluated prediction model on the i-th position when given $(x_1^t, \dots, x_s^t)$ with $s < i$. 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. When using cross-validation, the accuracy Acc of an evaluated model is given by

$$Acc = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{t \in T_k} \frac{1}{m - s} \sum_{i=s+1}^{m} \varepsilon_i^t \ . \tag{10}$$

Note that, while the prediction accuracy is simple to estimate and to interpret, other performance criteria, such as ratings provided by a panel of experts, should be more appropriate to evaluate the relevance of music models. We plan to define such an evaluation protocol in future work. We used 5-fold double cross-validation to estimate the accuracies. Double cross-validation is a recursive application of cross-validation that enables to jointly optimize the hyper-parameters of the model and evaluate its generalization performance. Standard cross-validation is applied to each subset of $K - 1$ folds with each hyper-parameter setting and tested with the best estimated setting on the remaining hold-out fold. The reported accuracies are the averages of the results of each of the K applications of simple cross-validation during this process.

For the baseline HMM model, double cross-validation optimizes the number of possible states for the hidden variables. 2 to 20 possible states were tried in the reported experiments. In the case of the model with distance constraints, referred to as the global model, the hyper-parameters that were optimized are the number of possible states for hidden variables in the local HMM model (i.e.

[3] http://www.cs.nott.ac.uk/~ef/music/database.htm.

Table 1: Accuracy (the higher the better) for best models on the jazz standards database.

Observed	Predicted	HMM	Global
32	96	34.5%	54.6%
64	64	34.5%	55.6%
96	32	41.6%	47.2%

Table 2: Accuracy (the higher the better) for best models on the hornpipes database.

Observed	Predicted	HMM	Global
48	144	75.1%	83.0%
96	96	75.6%	82.1%
144	48	76.6%	80.1%

2 to 20), the Lagrange multiplier λ, the number of components c (common to all distances) for each binomial mixture, and the choice of $\mathcal{P}$, i.e. which partitions of the sequences to consider. Values of λ ranging between 0.1 and 4 and values of c ranging between 2 and 5 were tried during double cross-validation. Since music data commonly shows strong dyadic structure following meter, many subsets of $\mathcal{P} = \{2, 4, 8, 16\}$ were allowed during double cross-validation.

Note that the baseline HMM model is a poor benchmark on this task, since the predicted sequence, when prediction consists in choosing the most probable subsequence given previous observations, only depends on the state of the hidden variable in position s, where s is the index of the last observation. This observation implies that the number of possible states for the hidden variables of the HMM upper-bounds the number of different sequences that the HMM can predict. However, this behavior of the HMM does not harm the validity of the reported experiments. The main goal of this quantitative study is to measure to what extent distance patterns are present in music data and how well these dependencies can be captured by the proposed model. What we really want to measure is how much gain we observe in terms of out-of-sample prediction accuracy when using an arbitrary model if we impose additional constraints based on distance patterns. That being said, it would be interesting to measure the effect of appending distance constraints to more complex music prediction models (Pachet, 2003; Dubnov et al., 2003) in future work.

Results in Table 1 for the jazz standards database show that considering distance patterns significantly improves the HMM model. One can observe that the baseline HMM model performs much better when trying to predict the last 32 symbols. This is due to the fact that this database contains song endings. Such endings contain many silences and, in terms of accuracy, a useless model predicting silence at any position performs already well. On the other hand, the endings are generally different from the rest of the rhythm structures, thus harming the performance of the global model when just trying to predict the last 32 symbols. Results in Table 2 for the hornpipes database again show that the prediction accuracy of the global model is consistently better than the prediction accuracy of the HMM, but the difference is less marked. This is mainly due to the fact that this dataset only contains two symbols, associated to note onset and note continuation. Moreover, the frequency of these symbols is quite unbalanced, making the HMM model much more accurate when almost always predicting the most common symbol.

In Table 3, the set of partitions $\mathcal{P}$ is not optimized by double cross-validation. Results are shown for different fixed sets of partitions. The best results are reached with “deeper” dyadic structure. This is a good indication that the basic hypothesis underlying the proposed model is well-suited to music data, namely that dyadic distance patterns exhibit strong regularities in music data. We did not compute accuracies for $\rho > 16$ because it makes no sense to estimate distribution of distances between too short subsequences.

Table 3: Accuracy over the last 64 positions for many sets of partitions $\mathcal{P}$ on the jazz database, given the first 64 observations. The higher the better.

$\mathcal{P}$	Global
$\{2\}$	49.3%
$\{2, 4\}$	49.3%
$\{2, 4, 8\}$	51.4%
$\{2, 4, 8, 16\}$	55.6%

4 Conclusion

The main contribution of this paper is the design and evaluation of a generative model for distance patterns in temporal data. The model is specifically well-suited to music data, which exhibits strong regularities in dyadic distance patterns between subsequences. Reported conditional prediction accuracies show that such regularities are present in music data and can be effectively captured by the proposed model. Moreover, learning distributions of distances between subsequences really helps for accurate rhythm prediction. Rhythm prediction can be seen as the first step towards full melodic prediction and generation. A promising approach would be to apply the proposed model to melody prediction. It could also be readily used to increase the performance of transcription algorithms, genre classifiers, or even automatic composition systems.

The choice of the HMM to initialize the model is not optimal. However, this has no impact on the validity of the reported results, since our goal was to show the importance of distance patterns between subsequences in rhythm data. In order to sample to models to generate subjectively good results (Pachet, 2003; Dubnov et al., 2003), one could use other benchmark and initialization techniques, such as repetition of common patterns.

Finally, besides being fundamental in music, modeling distance between subsequences should also be useful in other application domains, such as in natural language processing. Being able to characterize and constrain the relative distances between various parts of a sequence of bags-of-concepts could be an efficient means to improve performance of automatic systems such as machine translation (Och & Ney, 2004). On a more general level, learning constraints related to distances between subsequences can boost the performance of ”short memory” models such as the HMM.

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

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

Bilmes, J. (1997). A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models.

Cemgil, A. T., & Kappen, H. J. (2002). Rhythm quantization and tempo tracking by sequential Monte Carlo. *Advances in Neural Information Processing Systems 14* (pp. 1361–1368).

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

Dixon, S. (2007). Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, *36*, 39–50.

Dubnov, S., Assayag, G., Lartillot, O., & Bejerano, G. (2003). Using machine-learning methods for musical style modeling. *IEEE Computer*, *10*.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification, second edition*. Wiley Interscience.

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

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

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, *220, 4598*, 671–680.

Och, F. J., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, *30*, 417–449.

Pachet, F. (2003). The continuator: Musical interaction with style. *Journal of New Music Research*, *32*, 333–341.

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

Rohde, D. L. T. (2002). Methods for binary multidimensional scaling. *Neural Comput.*, *14*, 1195–1232.

Scheirer, E. (1998). Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, *103*, 588–601.

Sher, C. (Ed.). (1988). *The New Real Book*, vol. 1-3. Sher Music Co.

Titterington, D. M., Smith, A. F. M., & Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. Wiley.

Whiteley, N., Cemgil, A. T., & Godsill, S. J. (2007). Sequential inference of rhythmic structure in musical audio. *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 07)* (pp. 1321–1324).