Music Modeling with Random Fields

Victor Lavrenko, Jeremy Pickens
Center for Intelligent Information Retrieval, Department of Computer Science
University of Massachusetts, Amherst, MA 01002 USA
{lavrenko,jeremy}@cs.umass.edu

1. INTRODUCTION

Recent interest in the area of music information retrieval is exploding. However, very few of the existing music retrieval techniques take advantage of recent developments in statistical modeling. In this report we discuss an application of Random Fields to the problem of statistical modeling of polyphonic music. With such models in hand, the challenges of developing effective searching, browsing, and organization techniques for the growing bodies of music collections may be successfully met. [1]

Polyphonic music can be thought of as a two-dimensional stochastic process. Unlike text, the musical vocabulary is relatively small, containing at most several hundred discrete note symbols. What makes music so fascinating and expressive is the very rich structure inherent in musical pieces. Whereas text samples can be reasonably modeled using simple unigram or bi-gram language models, polyphonic music is characterized by numerous periodic symmetries, repetitions, and overlapping short- and long-term interactions that are beyond the capabilities of simple Markov chains.

Random Fields are a generalization of Markov chains to multi-dimensional spatial processes. They are incredibly flexible, allowing us to model arbitrary interactions between elements of data. Recently random fields have found applications in large-vocabulary tasks, such as language modeling and information extraction. One of the most influential works in the area is the 1997 publication of Della Pietra et al. [2], which outlined the algorithms used in parts of this paper. Berger et al. [1] were the first to suggest the use of maximum entropy models for natural language processing.

While our work was inspired by applications of random fields to language processing, it bears more similarity to the use of the framework by the researchers in computer vision. In most natural language applications authors start with a reasonable set of features (which are usually single words, or hand-crafted expressions), and the main challenge is to optimize the weights corresponding to these features. This works well in natural language, where words bear significant semantic content. In our case, induction of the random field is the crucial step. We will use the techniques suggested by [2] to automatically induce new high-level, salient features, such as chords and melodic progressions.

[1]This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-9905842, and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 1-58113-646-3/03/0007.

2. RANDOM FIELDS FOR MUSIC

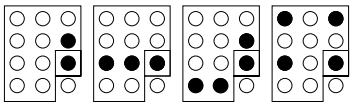
For this work, the music information appropriate to our model is pitch. A piece of music is organized into sequences of 12-bit (octave equivalent) binary vectors, where the value of a bit ($n_{i,t}$) represents the presence or absence of an onset of the pitch value i at time t in the sequence:

$$\begin{array}{l} n_{0,1}\ n_{0,2}\ n_{0,3}\ n_{0,4}\ \ldots\ n_{0,t} \\ n_{1,1}\ n_{1,2}\ n_{1,3}\ n_{1,4}\ \ldots\ n_{1,t} \\ \ldots \\ n_{11,1}\ n_{11,2}\ n_{11,3}\ n_{11,4}\ \ldots\ n_{11,t} \end{array}$$

Our goal is to develop a model that will allow us to predict the value $n_{i,t}$ from the values of the surrounding variables. In other words, we would like to develop an estimate for the probability distribution $P(n_{i,t}|\{n_{j,s} : j{\neq}i \text{ or } s{\neq}t\})$. It is important to stress that we do not want to assume independence among the variables, or restrict the conditioning to the immediate neighbors of $n_{i,t}$. On the contrary, we believe that the value of $n_{i,t}$ is strongly influenced by both its short-range and long-range neighbors in the lattice. However, for the scope of this paper we will impose two limitations on what kind of dependencies may exist in our field.

The first limitation we impose concerns the temporal nature of music. In our initial model will restrict the dependencies to only those notes that precede the target note in the sequence. For every note $n_{i,t}$ we define the concept of *history* or *neighborhood* $H_{i,t}$ to include the notes that either occur before time t, or notes that occur at time t, but have an index lower than i. Notes in $H_{i,t}$ are the ones that can be examined (observed) when we are making the prediction regarding $n_{i,t}$. In other words, we assume that the probability of note i playing at time t is completely determined by $H_{i,t}$.

The second limitation we impose on the conditional probability $P(n_{i,t}|H_{i,t})$ concerns the nature of dependencies that will be modeled by a field. For the sake of simplicity we will deliberately restrict dependencies to binary questions of the form: "*was some set of notes S played at some point before t?*". The answer to a question of this form will be called the feature function f_S, and S will be referred to as the *support* of f. Defined in this manner, our feature functions are always binary.



The above figure depicts a few examples of musical feature functions that may be induced to predict the probability of note 2 being played at time t. Black circles represent notes that are part of the feature function. The boxed black circle denotes the note $n_{2,t}$. The boxed area represents the history $H_{2,t}$. From left to right, the features are: $\{n_{2,t}\ n_{1,t}\}$, $\{n_{2,t}\ n_{2,t-1}\ n_{2,t-2}\}$, $\{n_{2,t}\ n_{1,t}\ n_{3,t-1}\ n_{3,t-2}\}$, $\{n_{2,t}\ n_{0,t}\ n_{2,t-2}\ n_{0,t-2}\}$

3. EXPONENTIAL FORM

At this point we are ready to select the parametric form that we will be using for computing the probabilities $P(n_{i,t}|H_{i,t})$. There are a number of different forms we could choose, but it turns out that for random fields there is a natural formulation of the distribution that is given by the maximum-entropy framework:

$$\hat{P}(n_{i,t}|H_{i,t}) = \frac{1}{Z_{i,t,\Lambda,\mathcal{F}}} \exp\left\{\sum_{f\in\mathcal{F}} \lambda_f f(n_{i,t}, H_{i,t})\right\} \quad (1)$$

In equation (1), the set of scalars $\Lambda = \{\lambda_f : f \in \mathcal{F}\}$ is the set of Lagrange multipliers for the set of structural constraints $\mathcal{F}$. Intuitively, the parameter λ_f ensures that our model predicts feature f as often as it should occur in reality. $Z_{i,t,\Lambda,\mathcal{F}}$ is the normalization constant that ensures that our distribution sums to unity over all possible values of $n_{i,t}$.

Our goal now is to develop a probability distribution $\hat{P}(n_{i,t}|H_{i,t})$ that will accurately predict the notes $n_{i,t}$ in the music. The maximum entropy principle leads us to select the most uniform distribution $\hat{P}(n|H)$ that is consistent with the structure imposed by the random field $\mathcal{F}$. To clarify what we mean by the structure consistency, suppose $f \in \mathcal{F}$ is a feature of the field. Let $\tilde{E}[f]$ denote the *empirical* expected value of f, which is simply how often the feature actually occurs in the training data $\mathcal{T}$:

$$\tilde{E}[f] = \frac{1}{12T}\sum_{t=1}^{T}\sum_{i=0}^{11} f(n_{i,t}, H_{i,t}) \quad (2)$$

Similarly, our estimate $\hat{P}(n|H)$ gives rise to the *predicted* expectation $\hat{E}[f]$ for the function f. This is simply how often our model "thinks" that f should occur in the training set:

$$\hat{E}[f] = \frac{1}{12T}\sum_{t=1}^{T}\sum_{i=0}^{11}\sum_{n\in\{0,1\}} \hat{P}(n|H_{i,t}) f(n, H_{i,t}) \quad (3)$$

The maximum entropy principle leads us to choose a model such that (2) and (3) are as similar as possible, but otherwise the least amount of assumptions about the data are made.

4. FEATURE INDUCTION

Our model from the previous section depends on two primary components. The first is the the structure of the field $\mathcal{F}$ itself, which is given by a set of constraints or feature functions $f\in\mathcal{F}$. The second component is the set of weights $\Lambda = \{\lambda_f\}$, one for each feature $f\in\mathcal{F}$. Learning the feature weights is a heavily studied problem and is beyond the scope of this paper [3]. In this section we instead describe how we can incrementally induce the structure $\mathcal{F}$ of the field, starting with a very flat, almost meaningless structure and slowly improving on it.

Our approach to inducing the structure of the field closely follows the algorithm proposed by Della Pietra et al. [2]. We start with field that contains only individual notes, without any dependencies: $\mathcal{F}^0 = \{n_{i,t} : i = 0\ldots11\}$. Now, suppose $\mathcal{F} = \{f_S\}$ is the current field structure. Also assume that the corresponding weights Λ are optimized with respect to $\mathcal{F}$. We would like to add to $\mathcal{F}$ a new feature g that will allow us to further increase the likelihood of the training data. In order to do that we first need to form a set of candidate features $\mathcal{G}$ that could be added. We define $\mathcal{G}$ to be the set of all one-note extensions of the current structure $\mathcal{F}$; in other words, we form new candidate features g taking an existing feature f and attaching a single note $n_{j,s}$ that is not too far from f in time (in our case, not more than by two simultaneities). Naturally, we do not include as candidates any features that are already members of $\mathcal{F}$.

Now, following the reasoning of [2], we would like to pick a candidate $g\in\mathcal{G}$ that will result in the maximum improvement in the objective function. Due to some simplying algebra made possible by the fact that both our data (the notes $n_{i,t}$) as well as the feature functions over that data are binary, the improvement or *gain* offered by g is computable in closed form. This final form is particularly interesting, since it represents the Kullback-Leibler divergence between two Bernoulli distributions with expected values $\tilde{E}[g]$ and $\hat{E}[g]$ respectively:

$$Gain = \tilde{E}[g]\log\frac{\tilde{E}[g]}{\hat{E}[g]} + (1-\tilde{E}[g])\log\frac{1-\tilde{E}[g]}{1-\hat{E}[g]} \quad (4)$$

So the algorithm proceeds as follows:

1. Initialize (learn) weights on $\mathcal{F}^0$.
2. Induce a feature by enumerating candidates and adding the one with the highest gain
3. Update all the weights of the new larger set of constraints
4. Goto 2 until there is no noticible change in likelihood
5. Return $\mathcal{F}$ and Λ as the induced field for that piece of music.

5. RESULTS AND CONCLUSION

We measure the performance of our model by its ability to predict the notes in a collection of polyphonic music. The collection consists of 2,806 classical pieces (16 million notes). We randomly split the collection into 1% training and 99% testing portions, induce a random field from the training portion and use it to predict every note $n_{i,t}$ in the testing set. The random field correctly predicted 82% of all testing notes (with a 41% precision at 40% recall). By comparison, a well-tuned (high-order) Markov chain model achieves 74% accuracy (26% precision at 40% recall).

Qualitatively, we examine the features produced by training a model on Variation 6 of "Ah vous dirai-je, maman" by Mozart (also known as "Twinkle, Twinkle Little Star"). Our algorithm quickly induced ≈8,000 features and learned their associated weights. Those features with the highest weights were a C major triad, the sequential notes *b-c-b*, a major 3^{rd} and a perfect 5^{th} on C, a semi-arpeggiated C major triad (the note c followed by the dyad *e-g*), an F major triad (subdominant of C), and a perfect 5^{th} on G (dominant of C). These features are highly characteristic of this piece and are strong qualitative evidence for the success of our modeling and feature induction techniques.

Future Works: Our algorithm is a starting point for an ad hoc music retrieval system in which a model $\hat{P}$, consisting of a set of induced features and their associated weights, is estimated for every piece of music in a collection. When a query is given, pieces are ranked by the likelihood of each piece's model having generated that query. The main advantage of Random Fields is their flexibility: rather than memorizing large chunks (Markov chain approach), they rely on inducing salient features of each piece and then discerningly matching these features to the query.

6. REFERENCES

[1] A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, pages 380–393, 1997.

[3] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *6th Workshop on Comp. Language Learning (CoNLL-2002)*, 2002.