Modelling Expressive Musical Performance with HMMs

Graham Grindlay David Helmbold
University of California, Santa Cruz

October 15, 2004

1 Introduction

Although one can easily produce a literal audio rendition of a musical score, the result is usually bland and unappealing. In this paper, we consider the problem of modelling and synthesizing expressive performances of piano melodies. For piano, an expressive performance can be described by deviations from the written score in three primary dimensions: note duration, note articulation (the gap or overlap between consecutive notes), and note velocity (loudness). We describe a system using hidden Markov models that automatically learns the ways in which example performances in a corpus of training data deviate from their corresponding scores. Once a model is trained, it can synthesize expressive renditions of novel scores. Informal listening tests indicate that the renditions generated by our system are generally preferred over literal renditions and are even slightly preferred to human performances. In addition to generating pleasing renditions, the trained models can be used for a variety of other purposes as described in the concluding section.

2 The Expressive Synthetic Performance (ESP) System

Our approach is to learn a generative model for score and rendition features. We draw from the motion synthesis work of Brand [1] and Wang *et al.* [3] in order to model expressive performances using hidden Markov models (HMMs). In our system, HMM states represent abstract "musical contexts" that are learned from the data. Each state models a distribution of score features and rendition features that can be sampled to produce a note appropriate for the context. These distributions are efficiently represented using kernels and sampling techniques [3]. Because our score features are a mix of discrete and continuous features, the score's marginal distribution is actually a joint distribution of its discrete (multinomial) and continuous (mixture of Gaussian) components.

Our training data is encoded in MIDI format and was extracted directly from a Disklavier piano, thus avoiding the need to extract symbolic information from raw audio[1]. We extract features from both the example performances and their corresponding scores. The score features for each note include articulation information, the rhythmic relationship of each note to both its predecessor as well as successor, information regarding the inherent metric strength of the note's position in its measure, and an indication of whether or not the note begins (or ends) a phrase or score. Duration, articulation, and velocity are the three ways a performer can add expression to the performance. Our performance feature set includes measures of note duration relative to the score, articulation values relative to the score, and velocity relative to the performance's average. In addition, we include first-order differences of these features to help smooth the transitions between "musical contexts".

The ESP system uses an entropic bias due to Brand [1] that automatically prunes both model parameters and states in the HMM during learning. This technique effectively merges parameter estimation and model selection into the single problem of searching for the maximum *a posteriori* estimate of the model parameters θ which encode the transition probabilities and feature distributions at each state. An annealing schedule reduces the chance of getting trapped in local maxima. Each musical note has one score feature vector and one rendition feature vector associated with it, and each training performance consists of a sequence S of these score feature vectors and the corresponding sequence R of rendition feature vectors. The learned HMM parameters θ encode a distribution $\Pr(S, R|\theta)$ on paired sequences of score and rendition features.

The algorithm for producing a performance from a (new) score uses a maximum-likelihood technique. It attempts to find a sequence of rendition features R that, for the new sequence of score features S, maximizes

[1] We are deeply indebted to Bruno Repp at Yale and his 10 graduate student performers for making this data available.

the probability $\Pr(R|S,\theta)$. Once the probability $\Pr(R|S,\theta)$ converges, we create a MIDI file from the score and rendition features R. Unlike training, the synthesis algorithm often gets stuck in local maxima. Therefore we run it several times and use the sequence of rendition features having the highest conditional probability to produce the MIDI-encoded performance.

Although our approach can model and synthesize fully polyphonic music, we have been focusing on piano melodies. Even melodies are not necessarily monophonic (note durations often overlap), but with melodies we assume that at most one note onset occurs at each point in time. Consideration of fully polyphonic pieces is underway, and we hope to report on it at MIPS.

We conducted a series of informal listening tests which showed the ESP system to be capable of producing performances at a skill level approaching that of the performers who generated the training data. We asked 14 local undergraduate music students to rank performances of a melody line according to aesthetic preference. The students heard a synthetic rendition, an inexpressive rendition (literally rendered score), and the performance of either an advanced undergraduate music major or a graduate music student. The expressive synthesis was preferred 5 of 7 times over the undergraduate's performance and 4 of 7 times over the graduate student's performance. Although some listeners (4 of 14) preferred the inexpressive score to the expressive synthesis, most (10 of 14) ranked the expressive synthesis higher.

3 Applications & Future Work

We are extending the system's capabilities to handle full polyphony. This involves adding harmonic features to model the interplay between melodic and harmonic lines and new types of score deviation such as the rolling of chords.

Our system builds a probabilistic model $\Pr(R,S|\theta) = \Pr(R|S,\theta)\Pr(S|\theta)$ from the data. Thus far we have exploited the distribution $\Pr(R|S,\theta)$ to create synthetic pleasing renditions of new scores. With the appropriate features, one could conceivably sample $\Pr(S|\theta)$ to generate new scores in the style of the data. When there are multiple data sets, perhaps representing different composers, performers, or styles, we can learn a different model θ_i for each data set. Scores (or performances) can then be classified by finding the θ_i maximizing $\Pr(S|\theta_i)$ (or $\Pr(R|S,\theta_i)$). One way to measure the similarity between two performances is to train a model using one of them and then measure the probability of the other given the model. To measure the similarity between models of two or more data sets we could use a divergence function, such as Juang and Rabiner's cross-model entropy [2]. Going even further, one might create a set of n reference θ_i's and map scores (or performances) into R^n using (some function of) the values $\Pr(S|\theta_i)$ (or $\Pr(R|S,\theta_i)$) to map scores (or performances) into R^n.

Finally, our system has practical applications as a tool for "bringing life" to raw musical scores which are usually emotionally flat and uninteresting when rendered literally. Such a system could be useful to composers, who may find the ablility to hear a piece as it might sound when performed extremely helpful during the composition process. As a result of the entropic pruning technique, our system produces HMMs which are sparse (ie. have relatively low state interconnectivity) and therefore more interpretable than what classical training methods might produce. We believe that by examining the structure of a trained model, it may be possible to gain musicological insight into the relationship between musical performance and compositional structure.

References

[1] Matthew Brand. Pattern discovery via entropy minimization. In D. Heckerman and C. Whittaker, editors, *Artificial Intelligence and Statistics*. Morgan Kaufman, January 1999.

[2] B. H. Juang and L. R. Rabiner. A probabilistic distance measure for hidden markov models. *AT&T Technical Journal*, 64(2):391–408, 1985.

[3] Tian-Shu Wang, Nan-Ning Zheng, Yan Li, Ying-Qing Xu, and Heung-Yung Shum. Learning kernel-based hmms for dynamic sequence synthesis. *Graphical Models*, 65(4):206–221, 2003.