CS229 Final Project: Audio Query By Gesture

by Steinunn Arnardottir, Luke Dahl and Juhan Nam
{steinunn,lukedahl,juhan}@ccrma.stanford.edu

December 12, 2008

1 Introduction

In the field of Music Information Retrieval (MIR) researchers apply machine learning techniques to systems that allow users to access a large database of music files by humming a fragment of a melody. These "query by humming" systems are intended for use in commercial settings such as music recommendation. Machine learning techniques can also have application in the creative domain of music creation and performance. We are developing an audio "query by gesture" system to be used as part of a computer-based music performance instrument. The query by gesture system allows a user to make a brief physical gesture with a pen and tablet interface which is then used to find and play a particular audio recording from a database of recorded short "musical gestures." This paper describes how we apply machine learning techniques to determine which musical snippet most closely matches a physical query gesture.

2 Description of Problem

Our task is to construct a system that learns to choose an audio sample when queried by a trackpad gesture. A user interacts with this system in two stages: during the training stage the user is presented with audio samples from the database, and he or she responds by making a gesture that they think matches the sound. We assume that the user's gestures correspond in some way to various aspects of the music, such as changes in dynamics, pitch or timbre, according to principles which may be either intentional or unconscious. These implicit principles can be considered as defining a mapping between the space of possible gestures and the space of possible musical snippets. Our system must use the training examples to infer this mapping, and then use the same mapping during the performance stage to respond to new gestures with appropriate musical samples.

3 Audio and Gesture Data

We recorded thee-hundred three-second long "musical gestures" performed on the saxophone by musician Adnan Marquez-Bourbon. From these we chose sixty-two samples that had mostly contiguous pitched melodic material and relatively simple melodic shapes (i.e. not having too many changes of direction.)

We define a gesture to be a three second long movement using a Wacom Bamboo tablet, which measures pen position and pressure over a 5.8" by 3.7" area. In order to gather the gesture data we created a program in the Max/Msp/Jitter environment [1] which plays a sample from the audio database, and records the first three seconds of gesture data after the pen first touches the tablet. As the user makes a gesture the audio sample is repeated to provide a reference. We found that this helped the user make gestures that more closely matched the time features of the audio.

Figure 1: *Training the query by gesture system*

4 Audio and Gesture Features

The selection of features is of critical importance since the features must encompass the elements of both gesture and audio that allow us to associate a gesture with a sound. After some experimentation we arrived at the following features. For audio we begin with two base-level features, the log frequency and the log rms energy in the signal, measured from overlapping windowed segments occurring every 10 milliseconds. The pitch tracking uses time-based autocorrelation[2]. Our base-level gesture features are the x and y pen position and pressure measured every 10 milliseconds as well as the x and y velocities. We then smooth each of these time series by approximating them with the first 20 Discrete Cosine Transform coefficients.

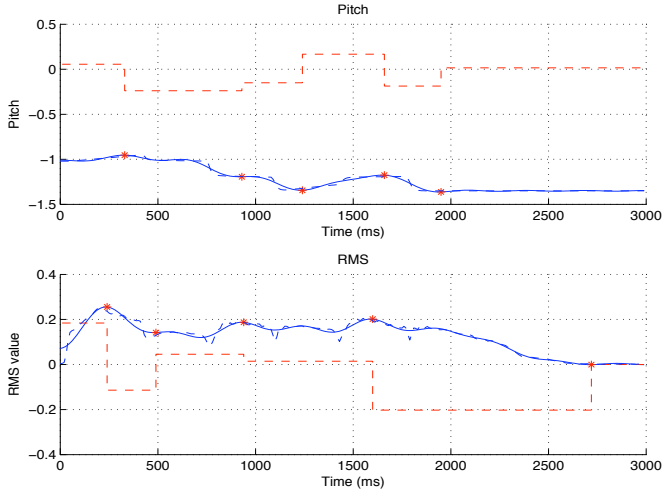

Figure 2: *Audio features - log rms energy and pitch. Dashed Blue: original feature, Solid: DCT-smoothed feature, Dashed Red: slopes, Star: turning points.*

We originally tried using as features some small number of DCT coefficients for each base-level feature time series. These represented the basic shape of each base-level feature, but they could not account for the relative time warping of feature curves that occurred between an audio sample's features and the features of its corresponding gesture. To capture the overall shape of our features and account for time-warping, we modeled

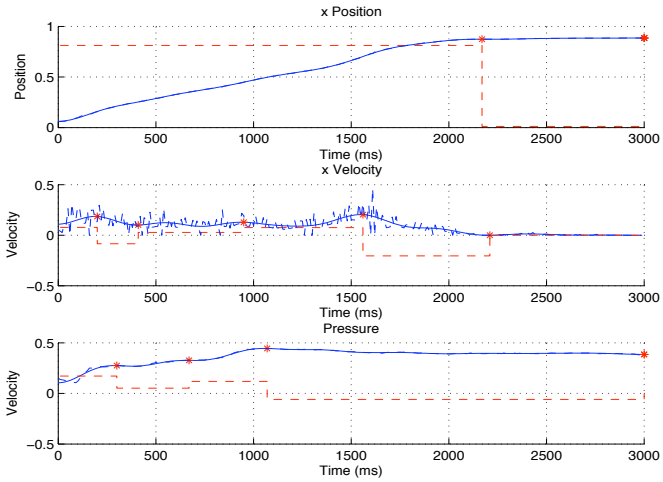

Figure 3: *Gesture features - position, velocity and pressure. Dashed: original feature, Solid: DCT-smoothed feature, Red: slopes, Star: turning points.*

each DCT-smoothed time-series as six sloped segments separated by five breakpoints. Figure 2 shows the base-level audio features for a specifc audio sample as well as its DCT approximation and the slopes and breakpoints our algorithm found. Figure 3 shows the features for the associated gesture from the training set.

For each of our two base-level audio features we get six slopes and five breakpoint times, resulting in twenty-two features that define our audio space. Similarly, the slopes and breakpoint times for each of our five base-level gesture features result in fifty-five gesture features.

5 Formulation as Classification

If we group the feature vectors for all of the audio samples into a set of classes, we can then formulate our mapping from gesture feature space to audio feature space as a classification problem. The audio feature groups become the class labels for the associated gesture feature vectors for the training examples. We can train a classifier to assign a class label to new gesture feature vectors, and then choose an audio sample from the same group.

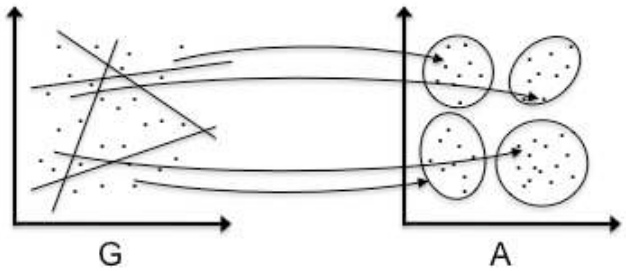

Figure 4: *Mapping from Gesture space to clusters in Audio space*

We used k-means clustering to group the elements of the audio database into six groups, using an euclidean distance measure in the audio feature space defined above. We found that the data fit easily into six categories. With more groups the separation

was not as well-defined and the algorithm would take much longer to converge. Figure 5 shows the number of members and confidence for each cluster.

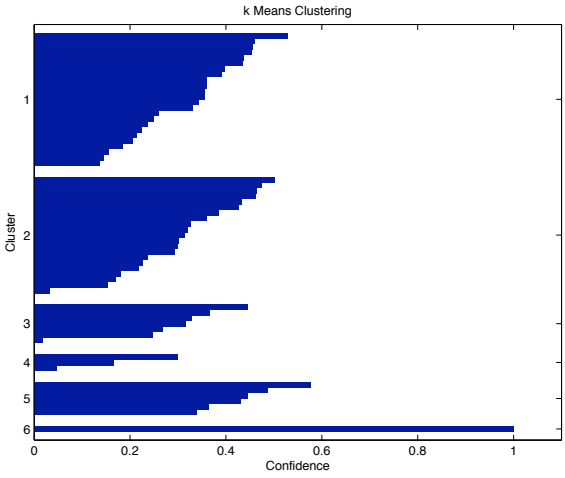

Figure 5: *K-Means Clustering*

It is interesting to consider what characterizes each audio category. Since the centroid of each audio category does not necessarily correspond to an existing audio example we can plot an approximate reproduction of the pitch and rms curves from the audio features of each centroid, as in Figure 6. If we listen to the audio files in each class we find that they do seem to belong together, and members of different classes are different. Notably, the sole member of class six is the only audio snippet that accidentally made it into our data set that has two short phrases within it.

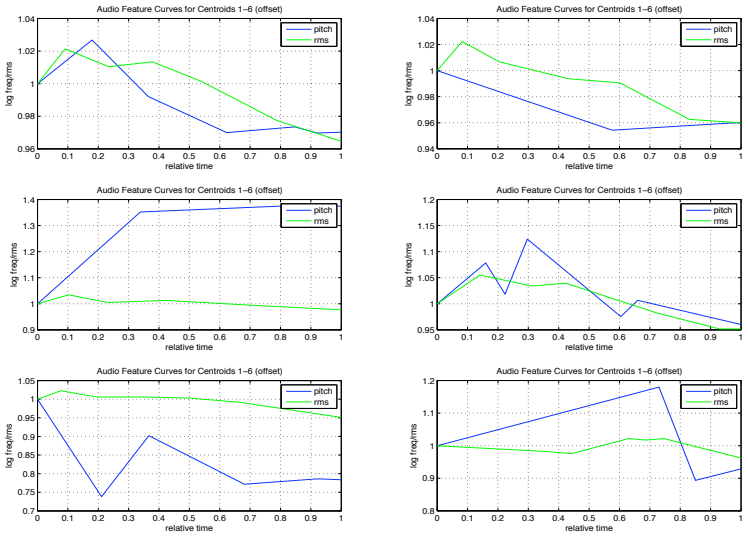

Figure 6: *Audio feature curves for cluster centroids*

6 Classification Results

Since our application is in a creative domain it is not crucial that a new gesture query retrieves a specific audio file, but it is important that the retrieved audio file be similar to the gesture in some way. For this reason we evaluate the success of our algorithm not on whether it retrieves the audio file which matches a given test example exactly. Rather we check whether it retrieves a sample from the same class.

The six categories from the audio clustering were used as class labels for the corresponding gestures in the training set. We trained five support vector machines to classify gesture vectors into one of these six classes. Each SVM estimates whether or not a test vector is a member the first five classes, and if it is not it falls into the sixth class.

We performed k-fold cross-validation to evaulate the classifier. With our sixty-two examples we tried 80/20 training to test ratio, 90/10 ratio, and leave-one-out cross validation. With all of these cases we achieved 60% to 65% accuracy in classifying test vectors. Achieving these accuracy results required tweaking the parameters of the SMO algorithm. We also tried using polynomial and gaussian kernals, but these did not result in noticeable improvements.

7 Conclusions and Future Work

It seems that the selection of features is crucial to successfully applying machine learning techniques. Designing a query by gesture system is especially difficult since we have to find features for representing both gestures and music. By modelling our base-features (pitch and rms for audio, x and y positions and velocities and pressure for gesture) as series of slopes with transition times between them, we hoped to create feature spaces in which curves of similar shape lay close to each other. However there may be other choices of base-features or models of them which work better for this task.

If we were to model the audio data as gaussian clusters and estimate the parameters using the EM algorithm, given a classification of a gesture we may be able to choose audio samples in a more interesting way. It may also be fruitful to model the time varying properties of both gesture and audio using Hidden Markov Models.

Regardless of our choice of features or learning algorithms, we believe that a larger training set will improve performance, and we plan to investigate that.

Once we our system works well consistently we can adapt our system to work with other gesture acquisition methods such as with accelerometers or video tracking. We can also use the parameters of a fully-trained query by gesture system to sythesize new musical sounds rather than retrieving pre-recorded sounds.

8 References

1) http://cycling74.com/products/max5
2) Garreth Middleton, http://cnx.org/content/m11716/latest/