Real Time Extraction of Musical Parameters From Audio

Victor Shepardson

October 2, 2014

1 The Problem

The aim is to convert a stream of audio from a performer into some useful musical parameterization in real time. Hand engineered solutions don't work very well for polyphonic sounds; intuition says that we should be able to leverage knowledge about what kind of sounds an instrument can make into a better algorithm.

The approach is to start with a synthesis process which characterizes the parameters and a corpus of audio which characterizes the instrument, and learn a regression model. We sample the output space and run the gathered parameter vectors through our synthesis model to generate corresponding inputs. If a generated input is perceptually similar to elements of our data set, we add the pair to our model.

Specifically, given

a set of audio fragments $D \subset \mathbb{R}^{\mathrm{n}}$

a synthesis process $S : \mathbb{R}^{\mathrm{m}} \mapsto \mathbb{R}^{\mathrm{n}}$

a perceptual distance function $\Psi : (\mathbb{R}^{\mathrm{n}} \times \mathbb{R}^{\mathrm{n}}) \mapsto \mathbb{R}$

we generate $D' \subset (\mathbb{R}^n \times \mathbb{R}^m)$ and use it to train a regression model, or directly in a nonparametric model.

With a musical parameterization in hand, we can alter the synthesis algorithm to produce new timbres or control another process entirely.

2 Subproblems To Be Solved

2.1 Sampling Procedure

Sampling the output space presents a dimensionality problem for output vectors of modest size. To make generation of D' from D tractable, we need a more effective sampling procedure. A good starting point might be to sample from a plausible hand-constructed distribution, rejecting points which are above some threshold τ in distance from the closest element of D. A better approach will doubtless be necessary to get useful data for high dimensional outputs; Cappé et al. [1] describe approaches to adaptive sampling which may be relevant here.

2.2 Perceptual Distance

We need a measure of distance between audio segments which is quickly evaluated and corresponds well to perceived difference to a human observer. A good place to start might be magnitude of difference in power spectra weighted by equal loudness contours. This could be implemented by a Fast Fourier Transform and a few matrix operations.

2.3 Synthesis Model

Ideally the algorithm will be able to adapt to different sythesis models which encode different parameters, but we will need a reasonable model to focus on for testing purposes. A simple FM synthesis model based on Chowning [2] for each string of a guitar, for example, parameterized by carrier frequency c_j, modulating frequency m_j, index of modulation i_j and gain g_j for string j:

$$S\left(t\right)=\sum_{j=1}^{6} g_j \cos\left(2\pi c_j\left(1+i_j\cos\left(2\pi f_j t\right)\right)t\right)$$

2.4 Learned Model

Having acquired a training set by sampling, we can use it directly in a nonparametric model or use linear regression along with basis functions. The model used to evaluate new inputs needs to reliably operate with low latency as perceived by a human; a new feature vector must be evaluated every few milliseconds.

For a nonparametric model, this seems to rule out hard drive access but admit DRAM access. The ideal size for D' would then be on the order of 100MB, for reasonable impact as a part of an audio processing software ecosystem. If each fragment of audio is ˜1kB, we get on the order of 1,000,000 feature-output pairs to interpolate. A good starting point would be to interpolate the associated outputs of the k-nearest features in D'. The literature on basics of nonparametric models here seems to descend into the confusing world of mathematical statistics from fifty years ago, but Benedetti [3] seems like a good starting point.

Pachet and Aucouturie [4] discuss common practices for converting short segments of audio to feature vectors.

3 Data

The ideal data set would be an exhaustive demonstration of a instrument's technique, recorded dry. A full audio CD is about 80MB; a few CDs of a solo performer recorded relatively clean should provide ample data. Personalizing to a given performer/instrument would involve the moderately arduous task of recording an hour or so of representative playing on which to train the algorithm.

4 Goals

Scope: a working prototype in MATLAB which does not operate interactively but could in principle once performance-optimized and ported to a real-time audio framework.

Milestone: simple or placeholder solutions to each of the above problems assembled in MATLAB to the point of producing initial results; at least one useful data set curated.

References

[1] Olivier Cappé et al. Adaptive importance sampling in general mixture classes. Statistics and Computing 18.4 (2008): 447-459.

[2] John M. Chowning. 1977. The Synthesis of Complex Audio Spectra by Means of Frequency Modulation. Computer Music Journal, Vol. 1, No. 2 (April, 1977), pp. 46-54

[3] Jacqueline K. Benedetti. On the Nonparametric Estimation of Regression Functions. Journal of the Royal Statistical Society. Series B (Methodological), Vol. 39, No. 2 (1977), pp. 248-253

[4] Francois Pachet and Jean-Julien Aucouturier. Improving timbre similarity: How high is the sky? Journal of negative results in speech and audio sciences 1.1 (2004): 1-13.