Content-Based Features in the Composer Identification Problem
CS 229 Final Project

Sean Meador (smeador@stanford.edu) Karl Uhlig (knuhlig@stanford.edu)

December 12, 2008

1 Overview

Classification of digital music (also called Music Information Retrieval, or MIR), is a long-standing problem in machine learning, one with many potential real world applications. For example, providing accurate recommendations is extremely important to online music websites (Pandora, iTunes Genius, etc.) as this is the primary means by which users discover new music. Many classification schemes have therefore been proposed, with some attempting to group music according to characteristics like mood and genre, and others attempting to identify the particular artist or composer who created the work.

Due to the complex nature of audio waveforms, however, nearly all music classification algorithms begin with a feature extraction step. Raw digital audio is essentially unusable for direct training on a learning algorithm; therefore, it is first necessary to process the data and distill key identifying features from the audio clips.

Until recently, researchers were getting very good results using only relatively low-level signal features of the audio [5]. However, after the discovery of a design flaw now known as the "album effect", the performance of such classifiers has dropped dramatically [1]. In this paper, we focus on classical composer identification, and propose a content-based feature set which addresses the limitations of current classifiers caused by the album effect.

2 The Problem

Composer identification is a relatively well studied multi-value classification problem. At the Music Information Retrieval Evaluation eXchange (MIREX), teams of researchers test their classification algorithms in a number of different contests. Composer identification is one such task, and many entries in this competition have yielded good results [5, 7]. A survey of these classifiers seems to suggest that feature selection is not simply a necessary pre-processing step in the overall algorithm, but is in fact the *key component* of a good classifier. In general, standard machine learning algorithms have been applied successfully to extracted features (mixtures of Gaussians, DAG-SVM, etc.); teams achieve different results, then, primarily because of the features they select.

J.S. Downie, a key contributor to the MIREX competition, recently proposed the following prevalent issues with current music classification schemes [1]:

1. Album Effect – prior to 2005, data sets in the MIREX competition were not filtered by album; that is, audio clips from the same album appeared in both the training and testing sets. Because of this, timbral feature sets (MFCCs, zero-crossing rates, spectral centroid, etc.) were picking up on "trivial production qualities" of the albums rather than actual music theoretical content. When album filtering was applied to the data, performance in the competition declined substantially.

2. Artist Filtering – Similar to the album effect, data sets were also not being filtered by artist. The results of this effect were particularly pronounced: in the genre classification task, a timbral-based classifier

achieved 79% accuracy without filtering, and 27% with filtering. This evidence further supports the notion that spectral and timbral features are a 'naïve' representation of music.

Together, these effects contribute to what Downie calls the 'glass ceiling' for spectral and timbral feature sets. In other words, these features alone appear to have an upper performance limit because they are not capturing real music information in a truly meaningful way. Improvements in this field will thus come only when classifiers account for more sophisticated, music-theoretical features of the data they examine.

3 Features

To address this limitation, we propose a feature set which is based on the broad chord changes which take place during a piece of music.

3.1 Spectral Features

The standard feature sets are summarized below, and are described in detail in [3, 6]:

1. Mel-Frequency Cepstral Coefficients (MFCCs) – these features are borrowed from the field of speech recognition, where they have been applied with great success.

   The MFCCs are the coefficients (typically the first 13) of the audio signal on the mel-frequency cepstrum. After obtaining the frequency-domain representation of the waveform via the Discrete Fourier Transform, the signal is converted to the Mel scale, a perceptually-motivated logarithmic scale which more accurately models human perception of pitch. Finally, the Discrete Cosine Transform is applied. The resulting cepstrum gives a fairly accurate representation of the timbre of the signal, and has the benefit that most of the energy is located in the first few bands of the spectrum.

2. Zero-Crossing Rate – This feature is simply computed as the number of times the signal crosses zero. It gives an overall measure of the noisiness of the signal.

3. Spectral Centroid – This feature indicates the 'center of mass' of the audio signal and, perceptually, is stronly correlated to the brightness of the sound.

4. Other Features – Other low level features commonly used in classification algorithms include spectral roll-off, root-mean-square energy, delta spectrum, kurtosis, skew, flatness, and entropy (we calculate these features using [2], a comprehensive auditory toolbox for MATLAB).

Based on the above feature sets, it should be clear that spectral feature sets do little to capture the actual musical content of an audio file. They certainly capture relevant features like timbre and brightness, but do not represent the overall music from which they are extracted.

3.2 Content-Based Features

In order to address the limitations outlined above, we propose a new feature set which is based on the beat and chord patterns found in a music clip. We hypothesize that a composer can be modeled roughly by the types of chord progressions they employ in their music. To that end, we extract a feature vector from each clip which is based on the number and type of each chord transition we encounter.

1. First, we use a standard beat detection algorithm to find the most likely positions of beats within the song. In western music, it is the case that chord transitions rarely occur between beats; therefore, we assume that the audio between beats is encompassed by a single chord. This allows us to slice the audio into beat-length frames, each of which represents a single chord.

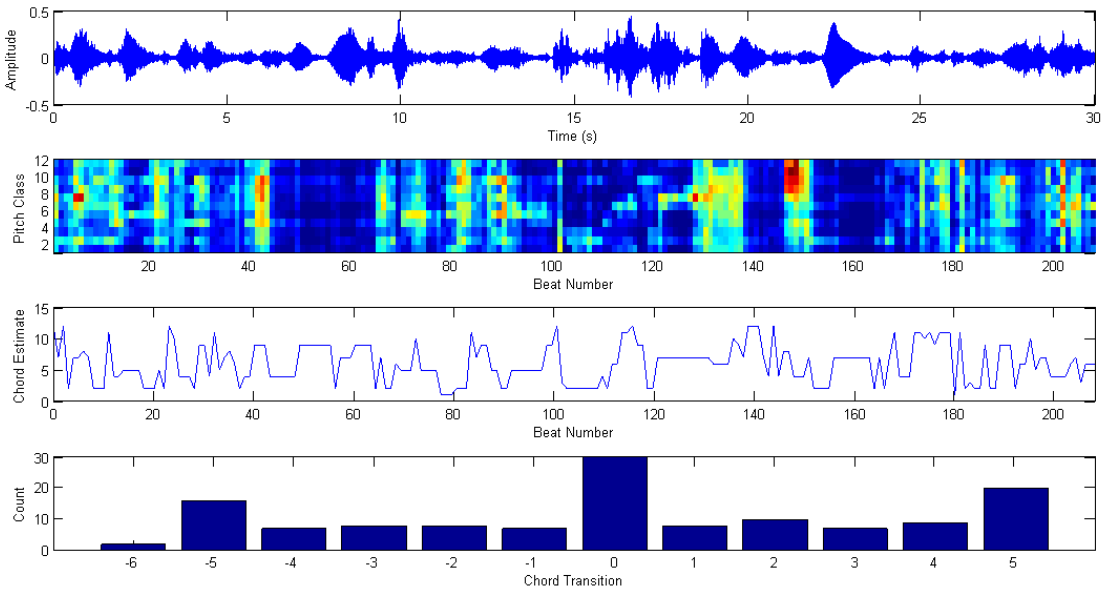

Figure 1: This diagram shows the four-step transformation from waveform to chord-transition vector.

2. For each beat frame, we compute its Harmonic Pitch Class Profile. This is calculated by dividing the frequency spectrum into logarithmic pitch bands, and then folding these bands into a single octave. The result is a 12-dimensional vector representing the relative strengths of each chroma (pitch) in the frame.

3. The HPCP is translated into an actual chord by correlating its values against those of pre-computed chord templates. In our testing, we limited our scope to the major and minor triads for each pitch (24 chords in all).

4. Finally, we compute the transition between each pair of chords in the sample. For example, the transition from C to G major represents the transition 'up by a 5th'. This yields a final, 48-dimensional vector of pitch transitions, and has the benefit of being invariant to the key of the song.

4 Classification

Support vector machines (SVMs) are among the best classifiers for their speed and accuracy. Surveying recent MIREX entries, we found that most teams used some form of SVM for classification. However, conventional SVMs are binary classifiers; in order to address the multi-valued case of classical composers, we instead use a decision directed-acyclic-graph SVM, or DDAG-SVM [8]. These classifiers work by performing a series of binary classifications, comparing only a single pair of classes at each step; thus, DDAG-SVMs work by gradual exclusion.

Training of a DDAG-SVM is the same as with conventional pairwise support vector machines. Namely, we need to determine $\binom{k}{2}$ decision functions for each pair of the k classes. However, testing is much faster using a DDAG-SVM because we only need to perform $(k-1)$ classifications, one for each level in the decision graph. In our case this means performing three classifications per sample, which is not a prohibitive cost. Finally, we used a Gaussian kernel in our algorithm; this choice seemed to yield the best results.

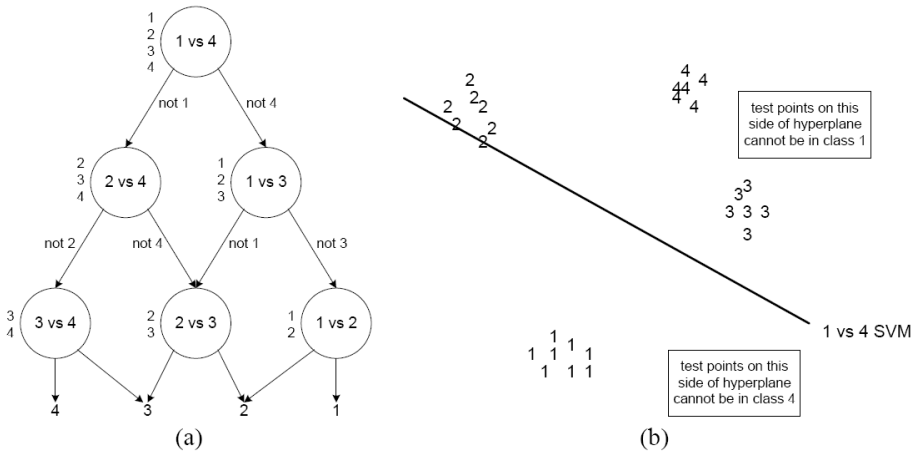

Figure 2: (a) The DDAG for finding the best of four classes. The equivalent list state for each node is shown next to that node. (b) A diagram of the input of a four-class problem. A 1-v-1 SVM can only exclude one class from consideration.

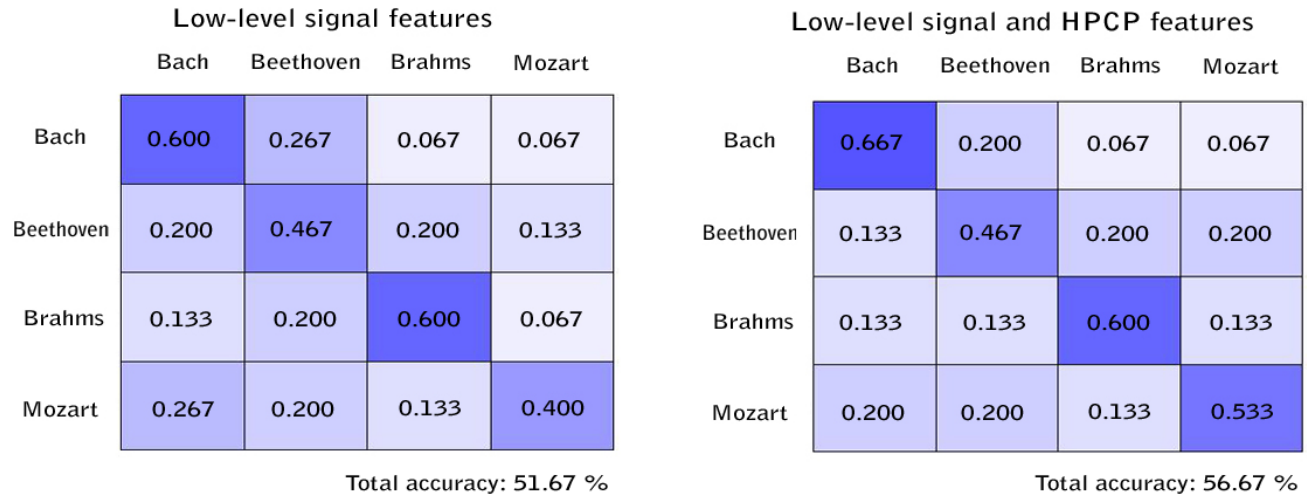

Figure 3: Confusion matrices resulting from training on baseline and augmented feature sets. Row labels represent the correct class, and columns represent the labels applied during classification.

5 Results

The MIREX competition's Classical Composer category includes eleven composers and a total of around 2700 30-second audio files. Due to time and resource constraints, we limited our scope to four of the most representative composers from the Baroque, Classical, and Romantic periods: Bach, Mozart, Beethoven, and Brahms. We collected one hundred 30-second samples for each composer, taking clips from a wide range of albums. Furthermore, we applied filtering to our training and testing sets to avoid the album effect described earlier.

To establish a baseline, we first train and test using only a standard spectral feature set (as described in section 3.1). We then augment our feature vectors with our chord analysis vector, and train and test using these features using 3-fold cross validation. Using our augmented feature vector, we achieved a 5% improvement over the baseline; our results are summarized in Figure 3.

6 Conclusion and Future Work

It is clear that the success of future classifiers will be dependent on their ability to capture music-theoretical aspects of the data they examine. We believe our feature set is a step in that direction, and we were able to achieve modest performance gains over spectral feature sets.

However, there are a number of limitations in our implementation which could see substantial improvement in the future. For one, we limited our chord analysis to only the major and minor triads. We found this limitation necessary because the techniques we used for chord detection did not provide the resolution required to identify more sophisticated chords. Beat and chord detection is an area of active research, and emerging techniques show promise for improving our model.

Furthermore, we limited our chord analysis to a simple transition-counting scheme; clearly, this method does not account for significant aspects of a musical work. In future refinements of our algorithm, we will want to look at more sophisticated structural aspects of the work like key modulation with respect to the base key, or longer chord patterns of chord changes (currently, transitions counts are patterns of length two).

References

[1] Downie, J.S. (2008). The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustic Science and Technology, 29*, vol. 4, 2008.

[2] Lartillot, O., & Toivianen, P. (2007). MIR in Matlab: A Toolbox for Musical Feature Extraction. *Unpublished.*

[3] Li, D., Sethi, I.K., Dimitrova, N., & McGee, T. (2001). Classification of general audio data for content-based retrieval. *Pattern Recognition Letters, 22*, pp. 533-544.

[4] Mandel, M., Poliner, G., & Ellis, D. (2006). Support Vector Machine Active Learning for Music Retrieval. *Multimedia Systems*, special issue on Machine Learning Approaches to Multimedia Information Retrieval, vol, 12, no. 1, pp. 3-13, Aug 2006.

[5] Mandel, M., & Ellis, D. (2008). Labrosa's Audio Classification Submissions. *MIREX 2008.*

[6] McKinney, M.f., & Breebaart, J. (2003). Features for Audio and Music Classification. *Proceedings of the International Symposium on Music Information Retrieval*, 2003.

[7] Peeters, G. (2008). A Generic Training and Classification System: Audio Music Mood, Audio Genre, Audio Artist, and Audio Tag. *MIREX 2008.*

[8] Platt, C., Cristianini, N., & Shawe-Taylor, J. (2000). Large Margin DAGs for Multiclass Classficiation. *MIT Press.*