A New Method for Calculating Music Similarity

Eric Battenberg and Vijay Ullal

December 12, 2006

Abstract

We introduce a new technique for calculating the perceived similarity of two songs based on their spectral content. Our method uses a set of hidden Markov Models to model the temporal evolution of a song. We then compute a dissimilarity distance measure based on finding log likelihood probabilities using Monte Carlo sampling. This method is compared to a previously established technique that performs frame clustering using Gaussian mixture models. Each method's performance is analyzed on a music catalog of 105 songs, and performance is subjectively evaluated.

1 Introduction

With digital music and personal audio players becoming more ubiquitous, the importance of a robust music similarity measure is evident. Such a measure can be utilized for playlist generation within one's own music collection or for the discovery of new music that is perceptually similar to an individual's preferences. Other applications include making music recommendations based on a user's song preferences (e.g., for music retailers) or effective organization of a music library. While there is little ground truth for music similarity since it can be quite subjective and depends on several factors, research within the music similarity field has developed substantially over the past few years [1].

Currently there are services that determine music similarity by hand; however, this becomes difficult and impractical in a large digital library. An alternative is to use collaborative filtering to compute the similarity between an individual's preferences and the preferences of others. However, this method has proven to be time-consuming and information from users can often be unreliable [2]. Thus, we are interested in automatically determining music similarity based on a song's audio content.

The calculation of music similarity depends on three main parts: selecting and extracting salient features, fitting a statistical model to the feature distributions within a song, and calculating a distance metric to compare two models. There are several potential features that may be extracted to determine music similarity; for example, low-level attributes such as zero-crossing rate, signal bandwidth, spectral centroid, and signal energy, as well as psychoacoustic features including roughness, loudness, and sharpness have been used in many audio classification systems. Mel-frequency cepstral coefficients (MFCCs), which estimate a signal's spectral envelope, have been widely used for both speech and music applications [3]. We focus on extracting the fluctuation patterns of sones, which will be described in detail in Section 4. Rather than using k-means or Gaussian mixture models to model the distribution of features, we utilize the temporal memory properties of hidden Markov Models.

2 Background

Logan and Salomon first introduced the idea of a similarity measure based on frame clustering. According to this method, a song is first divided into several frames of 20-30 ms duration. A set of MFCCs is extracted from each frame and each set is clustered using the k-means algorithm. A song's signature is determined by its set of clusters. Once every song's signature is computed, a distance between each song is calculated using a distance metric known as the Earth Mover's Distance (EMD). This distance measurement calculates the minimum amount of work to transform one song's signature into another's [2].

Aucouturier and Pachet improved upon this frame clustering idea. While still using MFCCs as their features, they use Gaussian mixture models to model the distribution of a song's MFCCs. Each GMM's parameters are initialized by using k-means clustering, and the model is trained using the Expectation-Maximization algorithm. After every song's Gaussian mixture model is computed, a distance measure which uses a Monte Carlo approach between each set of GMMs can be computed [4].

While these frame-based clustering methods have provided promising results, they do not take into account the temporal structure of a song. For example, if a song's spectral features change rapidly over a period of time, this information will be ignored by k-means clustering or GMMs. We believe that adding information describing transitions from one cluster (or state) to another may provide a more robust method of computing music similarity. We accomplish this task by modeling a song's distribution of features with a hidden Markov model. We compare our method to the music similarity method that won the 2004 International Conference on Music Information Retrieval (ISMIR) genre classification contest, which largely draws upon the work done by Aucouturier and Pachet [5].

3 ISMIR'04 method

In this section, we will describe the features, statistical model, and music similarity measure used in the ISMIR'04 genre classification contest-winning method, which we will refer to as the ISMIR'04 method. We implemented this method in Matlab using the Netlab toolbox [6] and the MA Toolbox for Matlab [7]. The songs that we use are first converted to mono and then downsampled to 11025 Hz. A diagram of this method can be seen in Figure 1.

3.1 Feature extraction

The ISMIR '04 method is a variant of the aformentioned frame-clustering method by Aucouturier and Pachet, which uses MFCCs as features. MFCCs are used to represent the spectrum of an audio signal, where low order MFCCs represent a slowly changing spectral envelope while higher order ones represent a highly fluctuating envelope [4].

While there are different approaches to compute MFCCs for a given frame, in the ISMIR'04 method, they are computed in the following manner. First the Discrete Fourier Transform (DFT) of the frame is computed. Second, the spectral components from the DFT are collected into frequency bins that are spaced according to the Mel frequency scale. The human auditory system does not percieve pitch in a linear manner, and the Mel scale accounts for this by mapping frequency to perceived pitch [8]. Next, the logarithm of the amplitude spectrum is taken and lastly, the Discrete Cosine Transform (DCT) is performed to obtain a compressed sequence of uncorrelated coefficients.

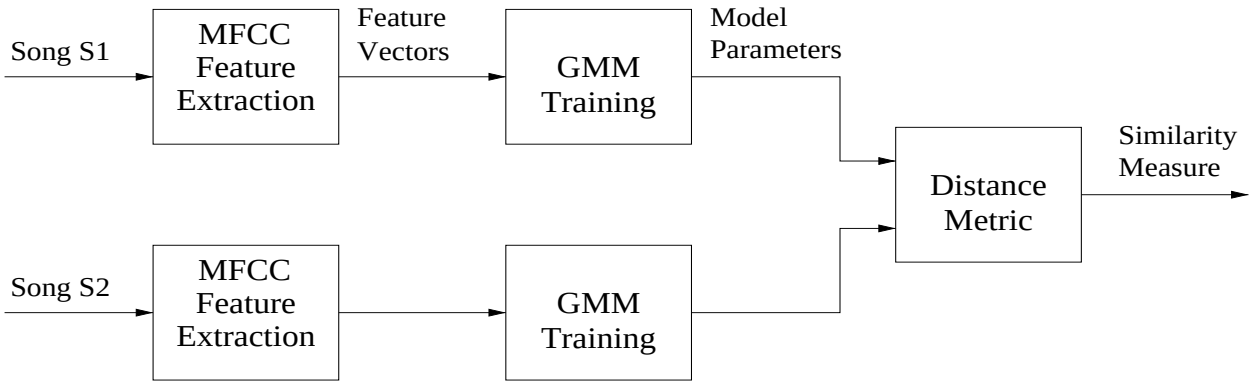

Figure 1: *Overview of ISMIR'04 Method*

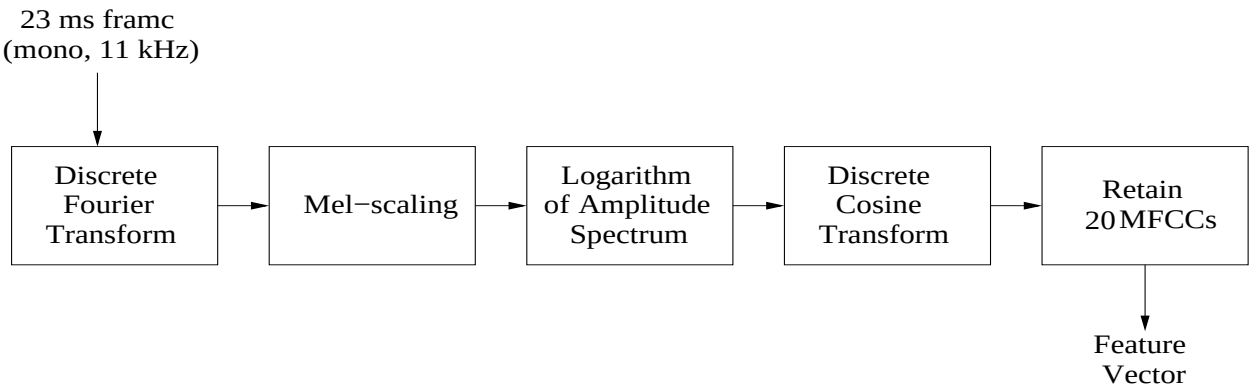

Figure 2: *MFCC feature extraction*

In the case of the ISMIR'04 method, the audio signal was divided into 23 ms half-overlapping frames. While 40 MFCCs are computed for each frame, only the first 20 MFCCs were retained [5]. The feature extraction process is described by Figure 2.

A number of MFCCs larger than 20 is unnecessary and often detrimental since the spectrum's fast variations are correlated with pitch. Thus, when incorporating pitch, spectrally similar frames (i.e., frames with similar timbres but different pitches) may not be clustered together [9].

3.2 Model training

From the feature extraction step, a vector of dimension 20 is obtained for each frame. In order to reduce the quantity of data and provide a more concise representation, the distribution of each song's MFCCs is modeled as a mixture of Gaussian distributions. A Gaussian Mixture Model (GMM) estimates a probability density as the weighted sum of a number of Gaussian densities, which are referred to as states of the mixture model. According to this model, the density function of a given feature vector can be represented by the equation:

$$p(f_t) = \sum_{i=l}^{M} \pi_i \mathcal{N}(f_t, \mu_i, \Sigma_i), \tag{1}$$

where f_t is the feature vector, M is the number of clusters in the GMM, π_i is the weight assigned to the ith cluster, and μ_i and Σ_i are the mean and covariance of the ith cluster.

The parameters of the GMM are first estimated using k-means clustering, and then the model is trained with the Expectation-Maximization algorithm. In the Expectation step, the parameters of the model are used to estimate the state that a feature vector belongs to. In the Maximization step, the estimates are used to update the parameters. The process is iterated until the log likelihood of the data no longer increases [4]. The ISMIR'04 method uses a mixture of 30 Gaussian distributions, but a number as low as 3 has produced favorable results.

3.3 Music similarity measure

Once models for songs have been trained, a distance measure between two songs can be calculated. This can be done by computing the likelihood that the MFCCs from Song 1 were generated by Song 2. However, this method requires access to a song's MFCCs, and the storage and computation of these features is expensive. However, one can still produce a distance measure only from the models of two songs. While it is easy to calculate the distance between only two Gaussian distributions using the Kullback-Leibler distance, it is more difficult to calculate the distance between two sets of Gaussian distributions [4].

The Monte Carlo sampling method provides a way to approximate the likelihood that a set of features is produced from a different song's model. A certain number of MFCC feature vectors can be generated or "sampled" from the GMM representing Song 1. The likelihood of these samples given the model of Song 2 can then be calculated. After the measure is made symmetric and normalized, the logarithm is taken to obtain the following distance metric:

$$d(1,2) = -\log \frac{p(S_1|M_2)p(S_2|M_1)}{p(S_1|M_1)p(S_2|M_2)}, \tag{2}$$

where $d(1,2)$ represents the distance between Song 1 and Song 2, S_1 represents a sample obtained from the model of Song 1, M_1 represents the model parameters of Song 1, and $p(S_1|M_2)$ represents the likelihood that a sample of Song 1 is generated by the model of Song 2 [9]. The number of samples chosen in the ISMIR'04 method was 2000.

4 HMM method

The method we propose uses hidden Markov Models. We implement this method in Matlab, using the Netlab Toolbox [6] and MA Toolbox for Matlab [7] for feature extraction and computing the music similarity measure, and the HMM Toolbox [10] for training the hidden Markov Models. As before in the first method, songs are converted to mono and downsampled to 11025 Hz. Figure 3 provides a concise flow diagram of our method.

4.1 Feature extraction

Rather than using MFCCs as features, we extract the fluctuation patterns (FPs), or modulation spectrum, of sones in twenty frequency sub-bands spaced according to the Bark scale. A unit of one Bark within this psychoacoustic scale represents a critical band in the human auditory filter. The main reasoning behind our choice of sones is that we would rather obtain features from all frames within a collection of songs and then perform dimensionality reduction. In contrast, the

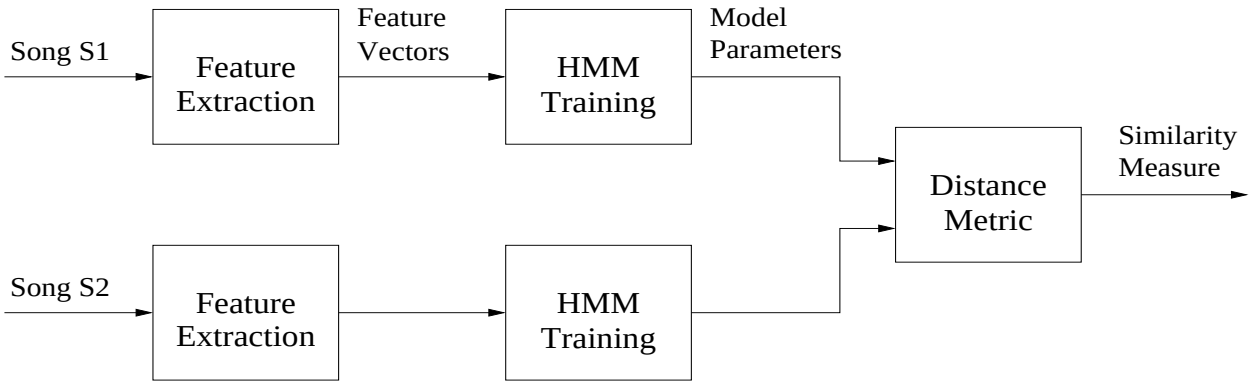

Figure 3: *Overview of HMM method*

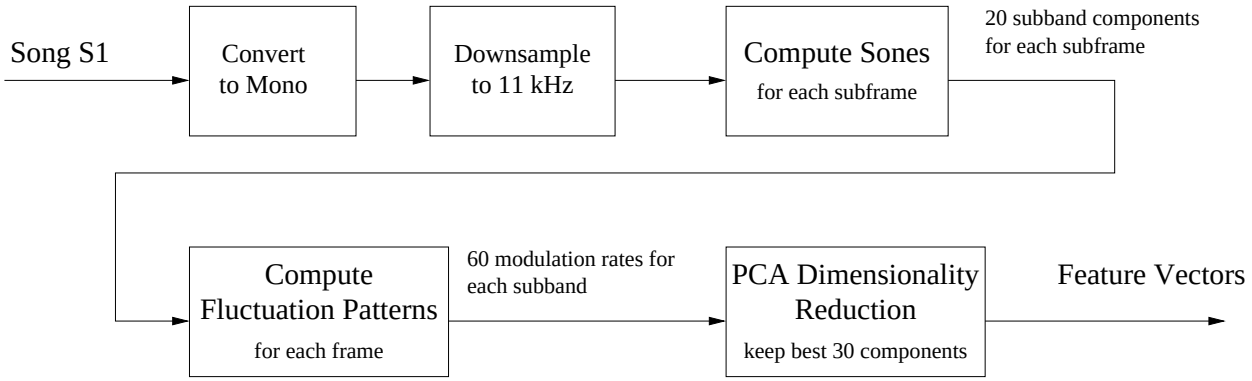

Figure 4: *Fluctuation pattern feature extraction*

dimensionality of a set of MFCCs for a given frame is reduced by the Discrete Cosine Transform. Thus, when using MFCCs, dimensionality reduction on the feature vector is performed before all frames from all songs are extracted. In addition, in our implementation, sones better represent perceived loudness and spectral masking than do MFCCs. While MFCCs represent changes within the spectral envelope, the modulation spectrum is represented though the sone fluctuation patterns [5].

The sones are extracted from 23 ms half-overlapping sub-frames. The FPs of the sones are then calculated for a set of 128 sub-frames, corresponding to a frame size of around 1500 ms. The resulting FP for a frame of music can be seen in Figure 5. This figure represents a matrix of 1200 values, with 20 rows corresponding to Bark sub-bands and 60 columns corresponding to modulation frequencies between 0 and 10 Hz. The values within the FP are vectorized to obtain a 1200-dimension feature vector that corresponds to one frame. Once features from all frames of all songs are computed, PCA is performed to reduce the dimensionality of the feature vector from 1200 to 30. This value is comparable to the number of MFCC coefficients retained in the ISMIR'04 method. A block diagram of our feature extraction method is presented in Figure 4.

4.2 Training a model

The 30-dimensional feature vectors belonging to a single song are then used as observations to learn a hidden Markov Model that best represents that song. A single multivariate Gaussian

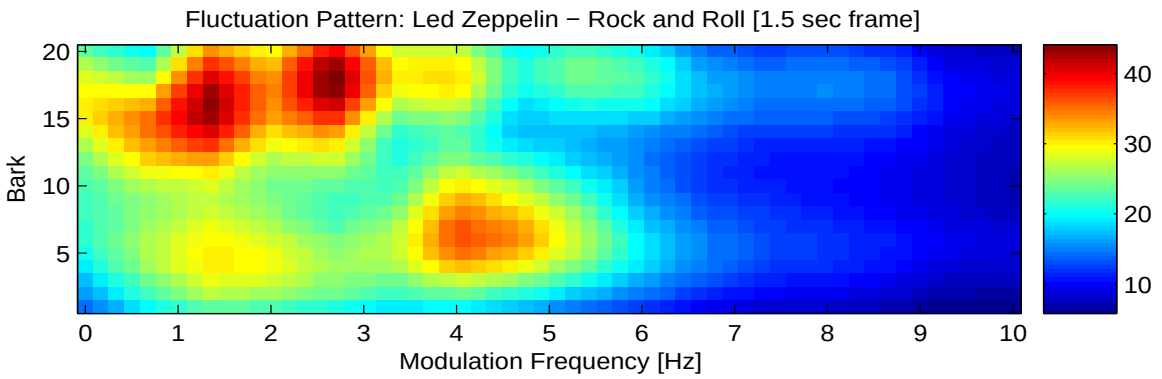

Figure 5: *Example fluctuation pattern*

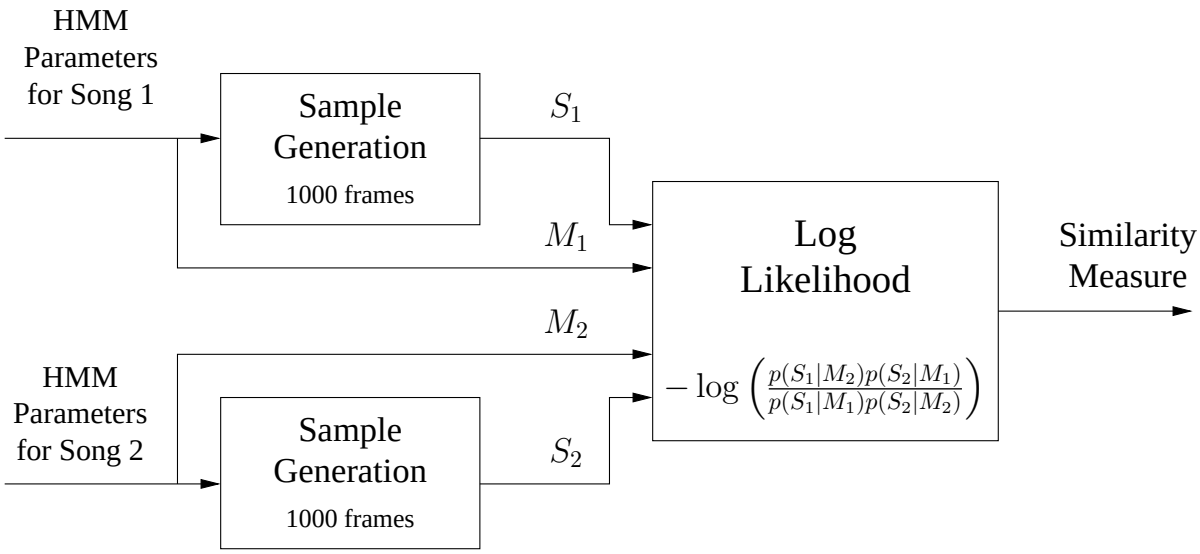

Figure 6: *Calculation of similarity measure*

distribution defines the posterior distribution of each hidden state. Each posterior distribution is initialized using k-means clustering of the vectors followed by covariance estimation. The priors and transition matrix are estimated using the frequency of each cluster. After initialization, the Baum-Welch EM algorithm is used to refine the parameter estimates until a local maximum of the log likelihood of the training sequence is reached.

4.3 Music similarity measure

The Monte Carlo sampling method used in the ISMIR'04 method was employed to calculate a similarity metric. Figure 6 shows an overview of this method applied to hidden Markov Models. A large number of samples (in this case, 1000 sequences, where each sequence represents a frame) are generated from a model using Viterbi decoding. Equation 2 can then be used to obtain a similarity measurement between two songs [9].

Artist	No. of songs
Beatles	8
Beach Boys	5
The Who	7
Led Zeppelin	6
Pink Floyd	5
Journey	4
Foreigner	4
Eagles	4
Green Day	5
Stone Temple Pilots	5
Soundgarden	5
Metallica	6
Jack Johnson	6
Dave Matthews Band	7
Beastie Boys	6
Notorious B.I.G.	4
Snoop Dogg	4
Lil John	5
50 Cent	5
Michael Jackson	4

Table 1: Composition of test database

5 Evaluation

To test our system, we assembled a list of 105 songs consisting primarily of rock and rap tracks from the 1960's to the present. The list contained at least 4 songs from each artist to increase the likelihood of finding a good match for each song. The song listing by artist is shown in Table 1.

We test three methods: the ISMIR '04 MFCC clustering method, our HMM-FP method, and a third method combining fluctuation pattern features and GMM clustering. For the third method, we use a total of 5 mixture components. The performance of each method is evaluated subjectively by the authors of this paper. After calculating the dissimilarity matrix (see Figures 7 and 8) using each of the three methods, the best 5 matches from each method for a single query song are displayed on screen. The best list is chosen and tallied and the procedure is repeated for the next song. This process is carried out in a randomized double-blind fashion to ensure that the testers do not know which list belongs to which method. Ties for best matching list were allowed as well as "no matches" when all three lists lacked any relevancy.

After tirelessly evaluating top-5 lists, we arrived at the conclusion that all three methods performed similarly overall. The ISMIR '04 clustering method produced the best list for 40% of the songs. Our HMM method was best for 44% of the songs. The clustered fluctuation pattern method was chosen for 44% of the songs. Remember that ties for best were allowed which is why the sum of the percentage values is over 100%.

Although the ISMIR '04 method performed slightly worse overall, it was consitently better for the timbrally distinct heavy metal music of Metallica. Its main defeats were in the genre of rap

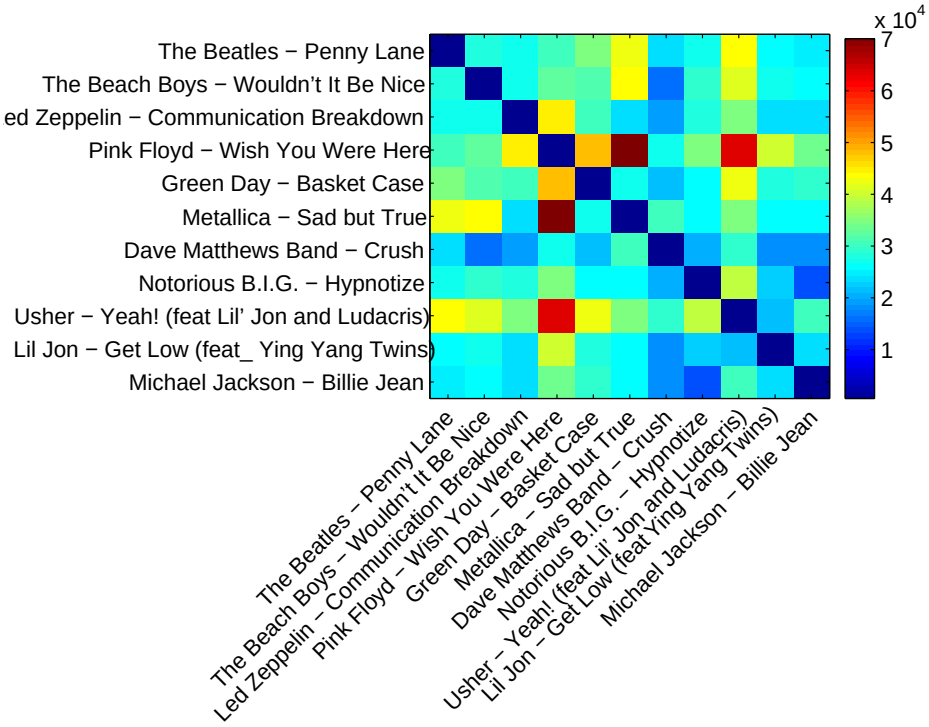

Figure 7: *Example dissimilarity matrix produced by the ISMIR '04 method*

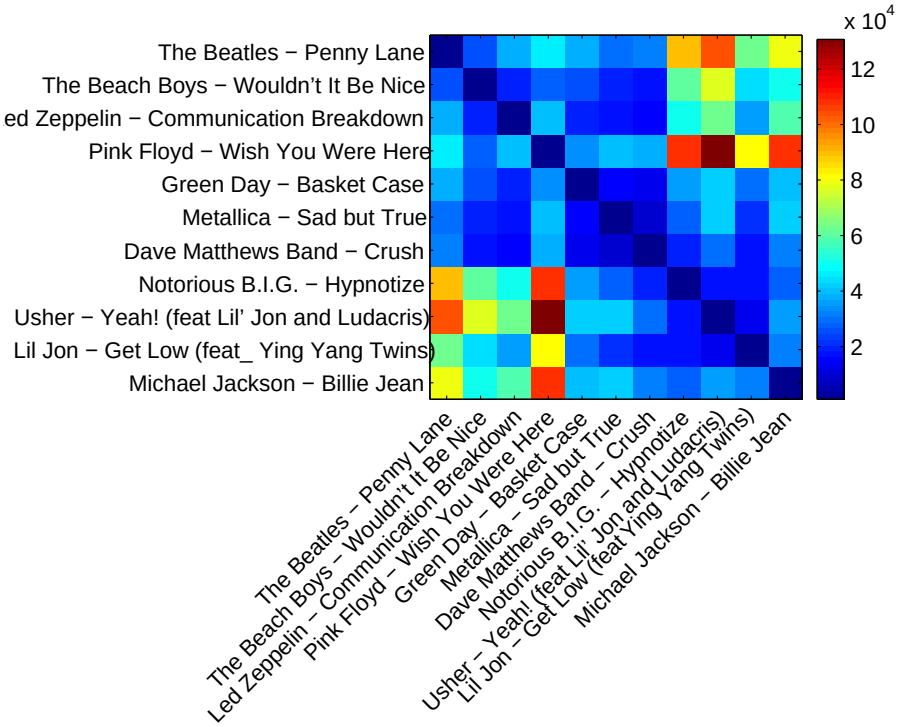

Figure 8: *Example dissimilarity matrix produced by the HMM-FP method*

ISMIR '04

1. 027_Pink Floyd - Welcome to the Machine.wav
2. 017_The Who - Who Are You (single edit version).wav
3. 105_Michael Jackson - Smooth Criminal.wav
4. 043_Eagles - Hotel California.wav
5. 021_Led Zeppelin - Black Dog.wav

HMM-FP

1. 027_Pink Floyd - Welcome to the Machine.wav
2. 031_Pink Floyd - Hey You.wav
3. 028_Pink Floyd - Comfortably Numb.wav
4. 029_Pink Floyd - Wish You Were Here.wav
5. 008_The Beatles - Strawberry Fields Forever.wav

ISMIR '04

1. 062_Metallica - The Shortest Straw.wav
2. 060_Metallica - ___and Justice for All.wav
3. 063_Metallica - Blackened.wav
4. 061_Metallica - Harvester of Sorrow.wav
5. 064_Metallica - Sad but True.wav

HMM-FP

1. 062_Metallica - The Shortest Straw.wav
2. 041_Eagles - Take It Easy.wav
3. 072_Dave Matthews Band - Crush.wav
4. 060_Metallica - ___and Justice for All.wav
5. 094_Petey Pablo - Freek A Leek (Ft_ Lil Jon).wav

Figure 9: Example top five lists

where rhythm is more of a salient feature. Also, the rhythmically distinct music of Pink Floyd was consistently categorized better by HMM-FP. The two methods employing fluctuation patterns were superior for nearly every rap song. Example top-5 lists for the ISMIR '04 method and our HMM FP method are shown above in Figure 9. The number one song in each list is the query song.

This fact leads us to conclude that extracting fluctuation patterns over the entire length of a song as features produces a useful representation of a song's rhythmic structure. While timbre was less of a factor in the FP-based methods, they were quick to discover interesting cross-genre rhythmic similaries between songs. For example, the refrain sung in "My Generation" by The Who is similar in pitch and rhythm to the synth sound repeated throughout "Freek-A-Leek" by Petey Pablo and Lil Jon. Though the two songs are quite distant from each other musically, we're curious to see what a good DJ could do with the two songs.

The competency of both methods using FPs in conveying rhythmic features of a song are promising; however the use of HMMs hasn't yet been justified since simply clustering FPs works just as well. We have yet to test the performance of the methods using a database of songs that have more diverse structures. HMMs can be useful along with Viterbi decoding to estimate the

hidden states of a song, thereby allowing the visualization of its structure.

6 Conclusion and future work

In this paper, we have presented a new method of computing music similarity and have reviewed an existing method based on frame clustering. Our method uses hidden Markov models to incorporate temporal evolution information in the form of transition probabilities between states. In previous research, HMM modeling has been applied to features extracted from short frames (23 ms) by [9] and was found to perform no better than GMM modeling. However, in our HMM model, each state corresponds to information extracted from longer frames of nearly 1500 ms in duration. In a crude subjective analysis, we found that the performance of our method is comparable to frame-based clustering using GMMs and k-means clustering. Additionally, we believe that our results point to better modeling of song structure and rhythm.

There are two main points to emphasize. First, the absence of "ground truth" makes objective evaluation of our method difficult. Some authors in the literature consider a song within the "same genre", "same artist" or "same album" (based on metadata) as the seed song as a good match. However, this method is clearly not optimal; for example, genre labels can often be too broad or too restrictive, or two songs from the same artist may not be perceptually similar. In order to thoroughly validate our method, our results should be tested against a larger database of subjective data. Second, it should be noted that frame-based clustering methods reach a glass ceiling of about 65% R-precision when compared with subjective grouping data. This suggests that important aspects of timbre and rhythm may be ignored by current music similarity methods. Additionally, varying certain parameters within the algorithm, such as number of MFCCs, number of components used in the GMM to model MFCCs, or number of points to sample in the Monte Carlo sampling method did not produce significant improvements [9].

A simple extension of our work would be to investigate the performance of a combination of the methods compared in this paper to better model both timbre and rhythm. Another possibility for future work includes segmenting a song into homogeneous regions and then fitting a model to these individual regions. Viterbi decoding could also be used to estimate the hidden states of each song allowing a comparison of song structure.

References

[1] E. Pampalk, S. Dixon, and G. Widmer, "On the evaluation of perceptual similarity measures for music," in *Proc of DAFx*, 2003.

[2] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proc of ICME*, 2001.

[3] M. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc of ISMIR*, 2003.

[4] J.-J. Aucouturier and F. Pachet, "Music similarity measures: what's the use?" in *Proc of ISMIR*, 2002.

[5] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio-based music similarity and genre classification," in *Proc of Sixth Intl Conference on Music Information Retrieval (ISMIR)*, 2005.

[6] I. Nabney, *Netlab: algorithms for pattern recognition*. Springer, 2001. [Online]. Available: http://www.ncrg.aston.ac.uk/netlab/

[7] E. Pampalk, "A Matlab toolbox to compute music similarity from audio," in *Proc of ISMIR*, 2004. [Online]. Available: http://www.ofai.at/ elias.pampalk/ma/documentation.html

[8] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *International Symposium on Music Information Retrieval*, 2000.

[9] J.-J. Aucouturier and F. Pachet, "Improving timbre similarity: How high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.

[10] K. Murphy, "Hidden Markov Model (HMM) Toolbox for Matlab." [Online]. Available: http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html