STRUCTURAL SEGMENTATION OF MUSIC WITH FUZZY CLUSTERING

Daniel Graves[1], **Witold Pedrycz**[1]
[1]Dept. of Electrical & Comp. Eng., University of Alberta, Edmonton, 9107 – 116 St., AB, Canada T6G 2V4

1. INTRODUCTION

Often in music we talk about its structural components as in intro, verse, chorus, bridge, outro, etc. In this research our objective is to automatically break music into these basic musical structures found in most popular music. The motivation behind musical segmentation is the many potential applications such as music thumbnail generation for sampling music from a database, fast-forward mechanisms for jumping to the next musical structure, music information retrieval, music summarization, and searching or browsing a musical database.

The goal of this research is to segment a digital music file such as an MP3 file. Digital music files are widely available due to the popularity of Internet music retailers; hence, the structural segmentation can be applied to any song purchased or converted to this digital format.

Musical segmentation was performed by [1] using MPEG-7 features and constrained clustering based on K-Means. Goto, c.f. [2], develops a method called RefraiD that detects the chorus sections of music and can even detect key changes in choruses using a 12-dimensional chroma feature vector. Peeters, c.f. [3], investigates musical segmentation of structural components using Mel Frequency Cepstral Coefficients (MFCC) and compares the sequence approach of structural segmentation with the state approach (HMM) and conclude that the state approach is more robust and computationally efficient. Authors in [4] propose a method of musical segmentation by detecting boundaries first, followed some aggregation. Many features are used including MFCCs. Abdallah et. al., c.f. [5], build a musical segmentation architecture based on a Bayesian framework.

2. METHOD

The MPEG-7, a well known standard for description of media and its digital storage, is employed in the automatic structural segmentation of music. Much pre-processing is performed similar to that done in [1]. All music files were converted to mono and had a sampling rate of 44.1kHz. A band spacing of 1/8th octave is used for the AudioSpectrumEnvelope audio descriptors outlined in the MPEG7 standard. The hop size described in the standard was set to the period of the beat in the song so that each sample in the spectrum corresponded to one beat. The beat was detected by the algorithm implemented in the software tool Matlab-XM (http://mpeg7.doc.gold.ac.uk/mirror/index.html). The spectrum is normalized by the L_2-norm and the dimensionality of the spectrum is reduced to 20 dimensions by principal components analysis (PCA) producing a feature vector of 21 dimensions called AudioSpectrumProjection in the MPEG-7 standard where the 21st dimension is the relative power of the beat.

A hidden Markov model (HMM) is trained on the entire AudioSpectrumProjection sequence where the output of each state is modeled by a single Gaussian distribution. The Viterbi algorithm is used to determine the most likely sequence of states for the observed vector sequence. Since each sample in AudioSpectrumProjection corresponds to a beat in the music, every beat corresponds to a state produced from the HMM. As suggested by [1] the number of states denoted by N in the HMM is selected to be large, i.e. N=80, since the HMM is unable to capture structural information in terms of individual states; however, structural information can be automatically distinguished using the local distribution of states mathematically denoted by the vector $\mathbf{x}_t = [c_0, c_1, .. c_{N-1}]$ where N is the number of states in the HMM and c_0 is the count of states W=11 samples from time t where W is the size of the window for generating the local distributions.

Hence the feature space of each song consists of the tuplet of pairs $\mathbf{X} = \{\mathbf{z}_t = (t, \mathbf{x}_t) \mid 0 \le t < L\}$ where L is the length of the song. The features are clustered using a modified version of Fuzzy C-Means (FCM) clustering where time and the distributions are treated as semantically distinct features in clustering. Since the time t and the local distribution $\mathbf{x}_t$ at time t are semantically distinct features, they should be distinguished as two separate blocks of features in the clustering, namely by modifying the distance function

$$d(\mathbf{z}_1, \mathbf{z}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\| + \alpha \|t_1 - t_2\| \qquad 1.$$

where $d(\mathbf{z}_1, \mathbf{z}_2)$ is the distance metric in standard FCM clustering. Just as with FCM, the number of clusters c must be specified beforehand. The update of membership values u_{ik} for i=1…c and k=1…L is identical to FCM except in the replacement of the metric $d(\mathbf{z}_1, \mathbf{z}_2)$. The update of the cluster centers is accomplished by the modified expression

$$\mathrm{v}_i = \frac{\sum_{k=1}^{L} u_{ik}^m \mathbf{z}_k}{\sum_{k=1}^{L} u_{ik}^m} \qquad 2.$$

where m is the fuzzification coefficient and $\mathbf{v}_i$ is the centroid distribution for each cluster i=1…c.

3. EXPERIMENTS

A number of experiences were conducted to evaluate the performance of the structural segmentation on some popular songs by the band Coldplay. Two evaluation criteria are used: classification rate (CR) and the f-measure (F). Precision and recall which are used to calculate the f-measure are denoted P and R respectively. Both measures are based on a sample-by-sample basis meaning that samples from each cluster are labelled according to the largest reoccurring class in each cluster and compared with the class label provided by a human expert. The number of classes was set to the number of different structural segments in the music and the number of clusters was set to the number of structural segments in the music (i.e. 1 intro, 3 verse, and 2 chorus segments gives c=6 clusters and k=3 classes). Tables 1 and 2 summarize the results.

Table 1. Results using reference distributions

Song	CR	F	P	R
Yellow (k=5)	89.0%	79.8%	80.2%	79.5%
A Message (k=4)	84.3%	74.0%	77.5%	70.9%
Fix You (k=6)	86.2%	77.2%	80.6%	74.0%
Swallowed in the Sea (k=4)	75.5%	62.6%	73.3%	54.7%
Talk (k=5)	77.4%	64.3%	60.8%	68.3%
A Rush of Blood to the Head (k=4)	89.4%	83.4%	92.4%	76.0%
In My Place (k=5)	86.0%	75.7%	77.5%	73.9%
Politik (k=6)	81.2%	68.9%	69.9%	68.0%
God Put A Smile Upon Your Face (k=6)	91.1%	83.0%	85.5%	80.6%
The Scientist (k=5)	83.1%	69.0%	67.4%	70.7%

Table 2. Results using FCM-DFS clustering

Song	CR	F	P	R
Yellow (c=10,m=1.8,a=0.1)	77.7%	70.2%	65.3%	75.9%
A Message (c=8,m=2,a=0.15)	89.2%	82.8%	88.1%	78.1%
Fix You (c=8,m=2,a=0.25)	79.8%	77.7%	71.1%	85.6%
Swallowed in the Sea (c=7,m=1.8,a=0.05)	89.7%	84.9%	79.2%	91.6%
Talk (c=8,m=1.4,a=0.7)	80.6%	68.4%	67.6%	69.1%
A Rush of Blood to the Head (c=8,m=1.5,a=0.1)	86.6%	81.0%	81.3%	80.6%
In My Place (c=9,m=1.4,a=0.1)	83.5%	71.2%	74.5%	68.1%
Politik (c=9,m=1.4,a=0.15)	88.4%	79.9%	79.0%	80.9%
God Put A Smile Upon Your Face (c=11,m=1.45,a=0.05)	71.3%	67.6%	62.6%	73.6%
The Scientist (c=8,m=2,a=1)	83.9%	73.5%	71.3%	75.9%

The results from the reference distributions are obtained by constructing reference distributions for each class that are the centroid or prototype distribution for each class as done by [1]. Distributions are labelled according to the reference distribution that is the closest in the Euclidean distance sense. The results produced by using reference distributions do not take advantage of time information. Hence, when clustering with reference distributions, the number of clusters (c) equals the number of classes (k).

With FCM-DFS, the number of clusters (c) can be greater then the number of classes (k) which accounts for the fact that there can be several chorus sections in a song. Since several choruses will be at different time instances and since we are making use of time information in clustering, each instance of the chorus is its own cluster. Varying the number of clusters (c) in clustering was not done in these experiments and will be the focus of future research. The number of clusters was fixed to be the number of segments in the song, i.e. total number of verses, choruses, etc.

4. DISCUSSION

The observations of the results are interesting as in a number of the test songs, the FCM-DFS performs better then the results from the reference distributions. Since the results from using the reference distributions do not employ the additional information about time, it is observed that FCM-DFS, which makes use of time information while clustering, is beneficial to the problem of musical segmentation.

The song "Yellow" and "God Put A Smile On Your Face" give relatively poor results compared with the other songs and compared with the technique of using reference distributions. A likely reason is that some of the structural segments are short and FCM is having trouble capturing these short segments.

The direction of future research will focus on looking at more songs and comparing FCM-DFS with current state of the art techniques in musical segmentation.

REFERENCES

[1] Levy, M., Sandler, M., (2008). Structural segmentation of music audio by constrained clustering. IEEE Trans. on Audio, Speech, and Language Processing, 16(2), 318-326.

[2] Goto, M. (2003). A chorus detecting method for musical audio signals. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 437-440.

[3] Peeters, G. (2003). Deriving musical structures from signal analysis for music audio summary generation: "sequence" and state approach, in CMMR (LNCS2771) Lecture Notes in Computer Science, New York: Springer-Verlag, 142-165.

[4] Ong, B.S., Herrera, P. (2005). Semantic segmentation of music audio contents, Proc. Int. Conf. Computer Music.

[5] Abdallah, S. (2005). Theory and evaluation of a Bayesian music structure extractor, Proc. ISMIR, 420-425.

ACKNOWLEDGMENTS

Support from the NSERC and Canada Research Chair (CRC) Program (W. Pedrycz) is gratefully acknowledged. This research is supported by a post-graduate masters NSERC fellowship and an iCore fellowship (D. Graves).