### music structural segmentation by hmm clustering

*Hua Xiao, Si Chen*

Center for Music Technology

Georgia Institute of Technology

#### Abstract

In this paper, we describe an algorithm of segmenting audio files into structural sections based on a trained Hidden Markov Model (HMM) of low-level spectral features. Histograms of neighbouring HMM states are then clustered into distinct segment-type. The boundary for each segment is marked by its starting point and end point in the format of pre-annotated ground truth. The results are pretty close to those of the original paper that proposes the method.

**Index Terms—**Audio, segmentation, structure, HMM, clustering

**1. Introduction**

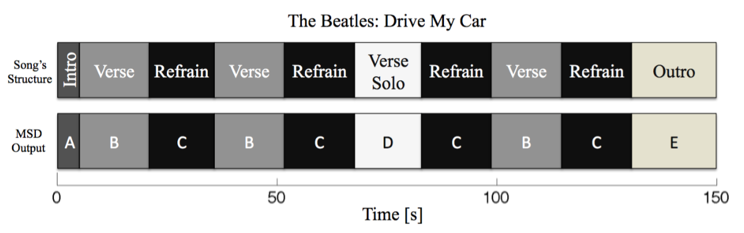
Music is composed using a highly structured language and thus produces structured musical segmentations. Structural segmentation of music means dividing an input audio signal according to its structural parts consistently without overlapping [2].

Most methods for the structural segmentation of music focus on the western popular music. With this understanding of structure, music piece is often segmented into sections that are commonly labeled as intro, verse or chorus.

**1.1. Problem definition**

In this task, we will be aiming at automatically derive structure boundaries from music audio files.

The task of music structure segmentation algorithms thus aims at retrieving a sequence of musically meaningful sections within the audio signal. The typical output example of such algorithms is illustrated in Figure below.



**Figure 1**. Output example of the structural analysis of the song Drive My Car by the Beatles.

**1.2. Application**

The detection of sections’ boundaries and of their repetitions is highly related to music annotation. In most cases, annotating a music piece manually is a time consuming process. However, providing users with structural sections would definitely improve the quality and efficiency of this process. Moreover, web services such as SoundCloud have revealed a great interest for collaborative annotation [2]. In short, providing users with automatically generated tags - such kind of services would be in great need. The

**1.3. Paper organization**

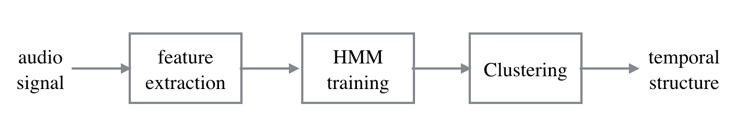
In section 3, we introduced main algorithm in detail, including low-level spectrum feature extraction, HMM training as well as histogram clustering. We then evaluate results using pair-wise F-measure in section 4 and give discussion as well as conclusion in latter parts.

**2. Related work**

Most previous research has been based on analysis of a self- similarity matrix comparing all window block features for a given track. In Levy’s paper [1], they introduce a method of using low-level HMM state labelling, and a histogram clustering using constraints.

**3. algorithm overview**

Algorithms for structural segmentation of music take an audio signal as input and give output information about its temporal structure. In our approach, a couple of subtasks can be identified. A rough overview of these subtasks is given in Figure 2. This section will be organized accordingly.



**Figure 2**. System overview: music structural segmentation by clustering based on HMM.

**3.1. Low-level feature extraction**

Our method of low-level feature extraction is based on the audio spectrum envelope, audio spectrum projection and sound model as described in the MPEG-7 standard [3]. This step yields a 21-dimensional audio spectrum projection feature vectors. The first 20 dimensions represent the spectral shape, by reducing dimension of the original normalized power spectrum envelope (64-dimension) using PCA method. And the final dimension is the relative power of each block window, which will be discussed in section 3.1.3.

*3.1.1. Beat tracking*

The dataset audios are mixed from stereo to mono by taking the mean of two channels. The original sampling rate is 44100Hz which is not necessary to be that high, for the sake of efficiency, we use the sampling rate of 11025Hz.

We use a hop size equals to the beat-length of the music (typically 400-600ms), and a window of three times the hop. While beat-lengths are estimated by a beat-tracking algorithm developed by labROSA [4]. For accuracy, we use the highest estimation of tempo instead of the lowest estimation (typically half of the highest tempo).

*3.1.2. Audio spectrum envelope*

The fundamental audio feature is the audio spectrum envelope. It is a standard MPEG-7 descriptor that uses logarithmic frequency scale and decibel scale to describe the power spectrum. Using logarithmic scale imitate the response of the human ear better.

We implement a constant Q transform (CQT) function to extract the audio spectrum envelope. The bin number of CQT is 8, which means there are 8 frequency bands in an octave. The lowest and highest frequency for CQT are 62.5Hz and 16kHz respectively. The output spectrum of the CQT is multiplied by its conjugate which yields a power spectrum.

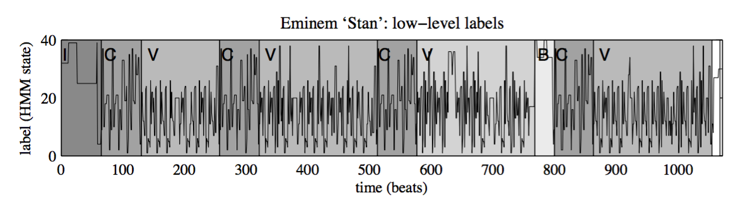
*3.1.3. Audio spectrum projection*

We applied the principal component analysis on the whole feature vector sequence to reduce the dimensionality of feature vector. Only the first 20 principal components are used. We also append the L2-norm, which represents the over all power, of each feature vector, to each feature vector. The norm is normalized by the max value, making all its values between [0, 1]. These 21-dimensional feature vectors, or audio spectrum projection (since PCA is a linear transform that project the feature vector to the principal components), are our low level features.

**3.2. HMM**

We use Baum–Welch algorithm to train an 80-state hidden Markov model on the entire sequence of feature vectors, with a single Gaussian output distribution for each state, and a single covariance matrix tied across all states. We then use Viterbi algorithm to decode the most likely state sequence, which is equivalent to assign a state to each beat.

Fig. 3 shows the resulting state sequence labels for the sample track.

**Figure 3**. Sequence of low-level labels for the sample track. The alphabetical labels are human annotation.

**3.3. Clustering**

We use a sliding window (16-beat long) to estimate the local state distribution at each beat of the sequence by counting 15 neighbouring states. This yields a 80-bin histogram of local states.

We then use kmeans with K=6 to cluster the histograms (state distributions). The value of K, 6, represents the number of common segmentation type of a song.

**4. Evaluation**

For the segmentation results to be evaluated and compared with a ground truth, music pieces must be previously manually annotated. We use the Beatles dataset for evaluation due to its relatively complete annotation and its popularity. We use 4 beats as our tolerance, only label the boundaries of each section (the label doe not have musical meaning, we just care about the boundaries). Since we always choose the higher tempo of the two tempos given by beat tracking algorithm, it is often actually a 2-beat tolerance.

**4.1. Pair-wise F-measure**

The evaluation method of using the pairwise precision, recall and F-measure was introduced in [1]. The temporal segmentation step produces a set of boundaries between sections. These sections are evaluated by their comparison with the annotated ground truth.

The performance of our project is represented by table 1.

|  |  |  |  |
| --- | --- | --- | --- |
| State number | 20 | 40 | 80 |
| Average precision | 0.38 | 0.41 | 0.47 |
| Average recall | 0.69 | 0.67 | 0.63 |
| Average Fmeasure | 0.47 | 0.49 | 0.51 |

**Table 1**. The performance of Kmeans histogram clustering with different state number.

**5. discussion**

The reference paper [1] uses constrained clustering, by which the clustering result could be improved considerably. For some reasons, we have not implemented it. Our result is pretty close to that of the kmeans histogram clustering in the original paper.

**6. conclusion**

Without constrained clustering, the result is not satisfactory. From table 1 we may conclude that increasing the number of states does not improve the result saliently.

**7. References**

[1] M. Levy and M. Sandler. “Structural Segmentation of Musical Audio by Constrained Clustering,” *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 2, pages 318–326, 2008.

[2] F. Kaiser, “Music Structure Segmentation,” Diss. Universitätsbibliothek der Technischen Universität Berlin, 2012.

[3] M. Casey, “General sound classification and similarity in mpeg-7,”  *Organised Sound*, vol. 6, no. 2, p. 153..164, 2001.

[4] D. Ellis., “Beat Tracking by Dynamic Programming,” *J. New Music Research*, Special Issue on Beat and Tempo Extraction, vol. 36, no. 1, pp. 51-60, 2007.