An Unsupervised Learning Approach to Musical Event Detection

Sheng GAO[*]*, Chin-Hui LEE*[†] *and Yong-wei ZHU*[*]

[*]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore, 119613

[†]School of Electrical & Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

{gaosheng, ywzhu}@i2r.a-star.edu.sg chl@ece.gatech.edu

ABSTRACT

Musical signals are highly structured. Untrained listeners can capture some particular musical events from audio signals. Uncovering this structure and detecting musical events will benefit musical content analysis. This is known to be an unsolved problem. In this paper, an unsupervised learning approach is proposed to automatically infer some structure of music from the segments generated by beat and onset analysis. A top-down clustering procedure is applied to group these segments into musical events with the similar characteristics. A Bayesian information criterion is then used to regularize the complexity of the model structure. Experimental results show that this unsupervised learning approach can effectively group similar segments together and automatically determine the number of such musical events in a given music piece.

1. INTRODUCTION

Musical content analysis is an emerging research with many potential applications, such as music information retrieval, indexing and organization of digital audio library, and music summarization [2, 7, 12]. To effectively represent musical signals, we must exploit the underlying music structures. Even the untrained listeners can capture some of these structures and identify some musical events. However, the perceptual structures in music signals are not readily available. Automatically uncovering of these structures from only the acoustical signal is a non-trivial task without any expert knowledge of musicology. It is known that the spectrogram of a musical signal displays strong structures, which can often be characterized by a set of musical events. From our experience in spectrogram reading and speech analysis, such outstanding events (or landmarks) often occur at places with significant changes in spectral and temporal characteristics. It is therefore possible to find musical structures using the data-driven approaches.

In music analysis, the signal is generally divided into a frame sequence with a fixed-length frame window (e.g. Hamming), from which spectral (e.g. mel-frequency cepstrum, pitch) and temporal (e.g. energy, zero-crossing) features are extracted. Various methods have been proposed to segment and group the audio frames into the segments [2, 3, 4, 7, 11, 12]. These techniques map the feature sequence into its similarity representation by calculating the distance between any pair of the frames, and then locating the local segment boundary position using some heuristic methods [2, 4, 7, 12]. In [4], the segmentation is accomplished by calculating a diagonal band of a similarity matrix with a width of a checkerboard kernel. Dynamic programming (DP) algorithms can then be applied to segmentation with pre-defined costs of insertion and deletion [7, 12]. These frame-based methods need to compute the similarity of all frame pairs and it is not efficient for real-time analysis of music excerpts that could last for a few minutes. To address this problem, the fix-length window based similarity is used [2, 4, 12], where each window contains a few consecutive frames. The size of the window, which is often constant for a given piece of music, determines the granularity or scale of the analysis. To merge similar segments, unsupervised clustering methods, such as heuristic clustering [12], *k*-means clustering [2, 4], and hidden Markov models [2, 11], are used.

In the above methods there are still some issues not handled satisfactorily. The frame-based segmentation is costly in computation and many pseudo spurs occur. To remove the effects of the pseudo spurs in segmentation, complex heuristics (e.g. threshold setting, rules) are used. But these algorithms strongly depend on the particular data and they are often not robust to diverse musical structures. Although the similarity matrix can be calculated on a block basis, its size cannot be determined flexibly according to the particular characteristics (e.g. tempo and genre) of a piece of music. For example, the window should be wide for slow tempo and rhythm while a narrow window may be better for music with fast tempo. Moreover, the analysis block is often subjectively determined without considering any musical knowledge. The number of clusters to group similar segments cannot be automatically determined according to the signal complexity.

In this paper we propose an unsupervised, data-driven approach to musical event detection. Given an acoustical realization of a piece of music, the musical structure is uncovered in three steps, namely: (1) dividing a long music excerpt into smaller segments using beat and onset detection; (2) grouping segments with similar spectral and temporal characteristics into musical event clusters using an unsupervised top-down clustering technique; and (3) determining the number of musical events using model selection with a Bayesian information criterion.

Our experimental analysis shows that the proposed unsupervised learning approach is effective and efficient in grouping similar segments together and automatically determining the number of such musical events in a given music piece. The proposed algorithm can also be used as a component of a music information retrieval system in which the events detected by the algorithm could form the basis for indexing and retrieving large collection of music documents.

2. MUSICAL EVENT SEGMENTATION

In music perception, a musical event is often defined by a set of coherent characteristics with some striking properties (e.g. the simultaneous roll of the drum, clash of the cymbal, and brief pulse of noise from the woodwinds) [1]. To segment the musical signal into these coherent events, we first apply beat and onset detection algorithms to infer some low-level music structures. For drum-less music we do not expect to accurately detect beat onset all the time, but try to detect the places with significant changes in the spectrum [6, 10]. Our experience in spectrogram reading tells us that the places of onset for drum music show obvious spectral differences when comparing with its previous and following contexts. For drum-less music we expect we can also do it. Using segmentation based on beat and onset detection, further clustering technique can be applied to infer more high-level

0-7803-8603-5/04/$20.00 ©2004 IEEE.

structure than the beat-based level. Figure 1 show an amplitude envelope and spectrogram of a piece of music. This music shows clear structure and two distinct musical events can be observed.

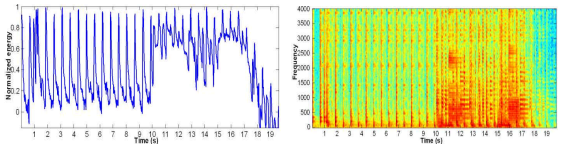

Figure 1 The amplitude envelope and its corresponding spectrogram (music "Pop Superstition" [9,10])

Many algorithms on beat and onset detection have been proposed up to now. Here we adopt the *maximum a posteriori* (MAP) based adaptive learning approach [10]. There are many advantages. First, it can propagate the learned knowledge on the beat from the previous excerpt of music to the following ones. This property makes the estimated beat and onset more robust, more consistent and less variant. Second, it is a flexible statistical framework that can easily fuse different knowledge sources (e.g. temporal or spectral features) to improve beat and onset detection. Third, the estimated posterior probability of the tempo can serve as a confidence measure for further processing.

2.1 Adaptive Beat Detection

The beat of a piece of music is a sequence of equally spaced phenomenal impulses, which defines a tempo for the music [6, 10]. Given a piece of music, a feature sequence can be extracted [6,10]. Let $X=(\vec{o}_1,\cdots,\vec{o}_t,\cdots,\vec{o}_T)$ denote a sequence of *D*-dimension feature vectors, and *T* be its length. A temporal window is applied to analyze the beat. In general its size should cover a few periods of the slowest tempo of interest. Assume that the window (or block) size is *L*, and there are *M* blocks in the feature sequence, then *X* can be re-denoted as $X=(O_1,\cdots,O_t,\cdots,O_M)$, where $O_t=(o_1^t,o_2^t,\cdots,o_L^t)$. If only the tempos in a range of $[t_a,t_b]$ are considered, then tempo induction can be formulated as

$$t^* = \arg\max_{t\in \text{All possible tempo sequence}} P(t|X) \quad (1)$$

where $t=(t_1,\cdots,t_t,\cdots,t_M), t_t\in[t_a,t_b]$ is any possible tempo sequence, and $t^*=(t_1^*,\cdots,t_t^*,\cdots,t_M^*)$ the optimal one.

To simplify the optimization problem in Eq. (1), we assume that the optimal tempo t_t^* is estimated only from the block O_t but with a conditional probability, i.e. $P(t_t|t_{t-1})$, which is derived from its previous block. With this assumptions, Eq. (1) can be simplified as

$$t_t^* = \arg\max_{t_t\in[t_a,t_b]}(1-h)\cdot\log\left(P\left(O_t|A^t,t_t,\Sigma^t\right)\right)+h\cdot\log\left(P(t_t|t_{t-1})\right), \quad t\in[1,M]. \quad (2)$$

Here h is a constant weight from prior knowledge. The first term in the right hand is the likelihood of the sub-sequence O_t. And the second is our model about the tempo for the block, O_t, given the known previous tempo.

The first term can be easily estimated from the observed data if a linear regression model is assumed. Given a block of sub-sequence evidence, O_t, a linear regression model (e.g. [10]) is defined as

$$\vec{o}_k^t = \mathrm{A}^t\cdot\vec{o}_{k-t_t}^t+\Theta^t \quad (3)$$

where $\Theta^t=(q_1^t,q_2^t,\cdots,q_D^t)^T$ is a prediction error vector, and A^t is a transformation matrix. Eq. (3) implies that the *k*-th observation is predicted by the $(k-t_t)$ previous observations with a prediction error Θ^t. In this paper, a diagonal transformation matrix is chosen, and the prediction error vector variable, Θ^t, is assumed to be a multivariate Gaussian distribution with a zero mean and a diagonal covariance, Σ^t.

With the above assumptions, the probability distribution of the observed feature, $\vec{o}_k^t$, is also a Gaussian distribution with a mean equal to $A^t\cdot\vec{o}_{k-t_t}^t$ and a covariance, Σ^t, i.e.

$$P\left(\vec{o}_k^t\middle|A^t,t_t,\Sigma^t\right)\sim N\left(A^t\cdot\vec{o}_{k-t_t}^t,\Sigma^t\right) \quad (4)$$

So the likelihood of the evidence, O_t, in Eq. (2) can be derived from Eq. (4) as

$$\log\left(P\left(O_t\middle|A^t,t_t,\Sigma^t\right)\right)=\sum_k\log\left(P\left(\vec{o}_k^t\middle|A^t,t_t,\Sigma^t\right)\right) \quad (5)$$

Because the likelihood defined in Eq. (5) is a function of a tempo, the second term in Eq.(2) can be approximated by a logistic function, i.e. given a block O_{t-1}, the conditional probability, $P(t_t|t_{t-1})$, can be defined as

$$P(t_t|t_{t-1})=\frac{1}{1+\exp\left(-l\cdot\left(P\left(O_{t-1}\middle|A^t,t_{t-1},\Sigma^{t-1}\right)-b\right)\right)} \quad (6)$$

where l is a scale coefficient and b is a bias. The normalization is performed to make $\sum_{t_t}P(t_t|t_{t-1})=1$.

With the above definitions, the optimal tempo, $t^*=(t_1^*,\cdots,t_t^*,\cdots,t_M^*)$, can be estimated from the feature sequence according to Eq. (2) using the EM algorithm based on the MAP criterion.

2.2 Beat Onset Decision

After the tempo or beat period is determined with Eq.(2), the beat onset can be decided. Assume that the detected beat period is t_t^* for a sub-sequence $O_t=(o_1^t,o_2^t,\cdots,o_L^t)$, and its corresponding energy envelope is $En_t=(en_1^t,en_2^t,\cdots,en_L^t)$, the sub-sequence is equally divided by its beat period. Let $O_t(i)=(o_{(i-1)\cdot t_t^*+1}^t,o_{(i-1)\cdot t_t^*+2}^t,\cdots,o_{(i-1)\cdot t_t^*+t_t^*}^t)$ be the feature vector in the *i-th* beat period (with $i\in[1,L/t_t^*]$), and $En_t(i)=(en_{(i-1)\cdot t_t^*+1}^t,en_{(i-1)\cdot t_t^*+2}^t,\cdots,en_{(i-1)\cdot t_t^*+t_t^*}^t)$.

The beat onset is defined as the time with the maximal energy. To extract the beat onset in each beat period, the averaging beat onset, $o\bar{n}^t$, is first calculated from the averaging energy envelope according to the following:

$$o\bar{n}^t=\arg\max_{j\in[1,t_t^*]}\frac{1}{t_t^*}\sum_{i=1}^{L/t_t^*}en_{(i-1)\cdot t_t^*+j}^t \quad (7)$$

With the assumption that the onset in each beat period will have a bias (here maximum bias is set to 10% of the beat period) centered at the average onset, the real onset is obtained by searching the time with the maximal energy during the above constrained range. It is determined as

$$on^t(i)=\arg\max_{j\in[o\bar{n}^t-bias_t^*,o\bar{n}^t+bias_t^*]}en_{(i-1)\cdot t_t^*+j}^t \quad (8)$$

3. MUSICAL EVENT DETECTION

After a musical signal is segmented with the beat onset, its beat-level structure is obtained. This segmentation has some perceptual meaning, especially for percussion music. Each segment can be treated as an audio shot, just like the shot in video analysis, because the segment boundaries occur at places with significant spectral changes. Comparing to the fixed-length window block [2, 4, 7], its granularity is varying according to its tempo. Due to the highly structured nature of music, many repetitive measures are often observed. To group these segments into some meaningful musical events without using any knowledge, unsupervised clustering is applied.

3.1 Top-down Clustering

Assume that the detected onset boundaries divide the feature sequence, $X = (\vec{o}_1, \vec{o}_2, \cdots, \vec{o}_T)$, into *N* segments. Denoted the onset sequence as $S = (s_0, s_1, \cdots, s_i, \cdots, s_N)$, where s_i is the onset place measured as the number of frames, and the range of the i-th segment is denoted as $[s_{i-1}, s_i]$. Our task is to group the *N* segments into musical event clusters based on a statistical representation of each segment and a chosen similarity metric.

For each segment, there are some frame-based features (e.g. MFCC, energy, etc.). To characterize each segment some frame-based low-level statistical features are extracted. In this paper the mean and covariance of the frame-based features are summarized for each segment. For the i-th segment its mean and covariance are denoted as m_i and Σ_i, respectively. Of course, other temporal features (e.g. energy envelope) can also be used. Now a segment is described by its mean and covariance without concerning its frame-based features when computing the similarity between the segments. Furthermore, we assume that the frame-based features within a segment are characterized by a single Gaussian distribution.

With the above representations, the similarity measure can be calculated between any pair of the segments. Here we use the popular Kullback-Leibler (KL) distance metric, to measure the distance between any two Gaussian distributions [4, 8]. The KL distance between the i-th segment and the j-th segment is defined as

$$d(m_i, \Sigma_i; m_j, \Sigma_j) = \frac{1}{2}\left[Tr(\Sigma_i \Sigma_j^{-1}) + Tr(\Sigma_j \Sigma_i^{-1}) + (m_i - m_j)^T (\Sigma_i^{-1} + \Sigma_j^{-1})(m_i - m_j)\right] - D \quad (9)$$

To reduce computation in Eq. (9), a diagonal covariance matrix is adopted. Given the *N* segments and the chosen KL distance metric, a top-down *k*-means clustering procedure, described as follows, is used to group these segments:

1. Initialization
   a) Calculate the global mean and covariance from the feature sequence $X = (\vec{o}_1, \vec{o}_2, \cdots, \vec{o}_T)$.
   b) Set the total cluster number, *numC=1*.
   c) Set the split cluster index *splitId=1*.
2. Set *numC=numC+1*.
3. Split the cluster *splitId* into 2 clusters.
4. Use *k*-means to estimate the mean and covariance of the *numC* clusters by minimizing the summarized intra-cluster similarities measured by the KL distance.
5. Assign *splitId* to the cluster with maximal intra-cluster distance as the next to be split. If its sample size is less than a predefined threshold, then assign *splitId to* the cluster with the maximal sample size.
6. Check if *numC* reaches the maximum number of clusters allowed. If so, then exit. Else, go to step 2.

3.2 Model Selection

Given any piece of music, it is not easy to know about how many musical events are sufficient to describe it. Significant differences of the musical structure are often observed in diverse music pieces. Some music excerpts are simple, which maybe played by a single instrument with repeats of a few similar events, while others are complex where many instruments play simultaneously with diverse chords and rhythms. This implies that the number of the musical events should partially depend on the complexity of music. For music with a simple structure, only a few clusters may be sufficient, while much more clusters are needed for modeling music with the complex structure.

In general, increasing the number of clusters can improve the fit of the data in a piece of music. But the risk of over-fitting will also increase at the same time. To balance these two facets, we adopt a Bayesian information criterion (BIC) [8] to choose an optimal model from a set of models, each of which is obtained with the top-down clustering procedure introduced earlier. For any piece of music, we assume that at least *minC* clusters are needed to represent it and the maximal number of the clusters is set to *maxC*. The set of models, each corresponding to a cluster set, is $\Phi = \{f(n) | n \in [\min C, \max C]\}$. And $f(n) = \{m(n,i), \Sigma(n,i) | i \in [1,n]\}$ is a candidate model of *n* clusters with *n* means, $m(n,i)$, and *n* covariance matrices, $\Sigma(n,i)$. BIC is used to score each of the models, $f(n)$, using a penalized function defined as:

$$BIC(n) = -L(n) - \frac{1}{2} k \cdot Q(n) \quad , \quad (10)$$

where the first term, $L(n)$, in the right hand side of Eq. (10), is the summary of all intra-cluster similarities (here the goal in clustering is to minimize the similarity not to maximize the likelihood), k is a penalty weight, and *Q(n)* is a measure of the complexity of the model. Because a diagonal covariance is used in our case, *Q(n)=2n*D*log(N)*. The optimal model, $f(n^*)$, is obtained by searching the set of all possible models, Φ, to find a model with the maximal BIC score,

$$f(n^*) = \max_{f(n) \in \Phi} BIC(n) \quad . \quad (11)$$

4. EXPERIMENTAL ANALYSIS

To analyze our proposed structure learning approach, a database with 807 pieces of music, with an average length 240 seconds, is first built. All of them are with the different formats (e.g. MP3, RAM, RM, WMA, etc) and encoding rates, collected from the web. Many genres (e.g. western popular music, Chinese classical music, songs by various singers, etc) are covered. All pieces of music are converted to the standard wav format with a 16-bit resolution and 8-kHz sampling rate. The 12-dimension MFCC vector is first extracted from a 32ms frame with 16ms overlap. Hamming window and 24 Mel-filter banks are used [5]. Then the first and second order differences of MFCC are calculated to form a 36-dimension feature vector. The normalized energy is also calculated.

In the beat and onset detection algorithm, h is set to 0.5, the interested tempo is between 60bpm and 250bpm, and the length of the block to analyze the tempo is 5 seconds. In the top-down clustering algorithm, the desired number of the

clusters ranges from 2 to 20. And the penalty weight in Eq. (10) is equal to 1.0.

4.1 Musical Event Detection Analysis

In our music dataset with 807 pieces of music, the proposed algorithm detected a total of 12164 musical events with an average of about 15 musical events and a variance of about 19 for a music piece. This implies that there is a great difference in music structure in the dataset. In this following we characterize the musical events detected.

First the detected onsets and musical events are displayed in Figure 2 for a piece of music (~20 sec) [9] with a simple structure, together with its corresponding spectrogram. The vertical lines label the onset position and the characters (e.g. "A" or "B") represent the distinctive musical events. This music has a Motown/Soul style, where the first 10-sec segment only has a drum and the next 10-sec segment has a drum and an electronic instrument with a pitch. The detected tempo is 98 bpm. And two musical events ("A" and "B") are found and segmented using our proposed algorithms. It matches well with human perception. This figure indicates that our algorithms work well for this simple music structure. Its beats, onsets, and musical events are correctly detected.

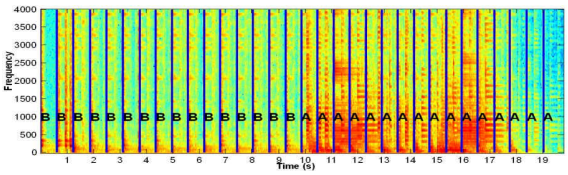

Figure 2 Musical events grouping of beat onset segments and the spectrogram of music "pop superstition" (Vertical line: onset position, "A" or "B": musical event)

Figure 3 shows another 10-sec excerpt from an ~280-sec music selected from our dataset. It is a Chinese classical piece played by the urheen, a Chinese instrument. Because it has no drum beat and its rhythm varies slowly, many onset insertions were observed. However, the detected musical events seem still well behaved. For this piece, 15 events, each with a similar spectrum, were detected.

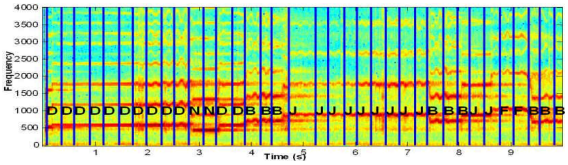

Figure 3 Musical events grouping of beat onset segments and the spectrogram of music "a Chinese classical piece" (Vertical line: onset position, "A", "B", ..., musical events)

4.2 Model Selection Analysis

Now we study the relation between the BIC score and the number of the clusters. The above two pieces of music are still chosen as the examples and the corresponding curves are depicted in Figure 4, in which the left figure is for the "pop superstition" piece, and the right for the Chinese classical music excerpt. In general, the likelihood in the right hand side of Eq. (10) monotonously increases with increasing number of clusters. This property is changed after the penalty is added. From the right curve it clearly shows that the maximal BIC score is observed with 15 musical event clusters.

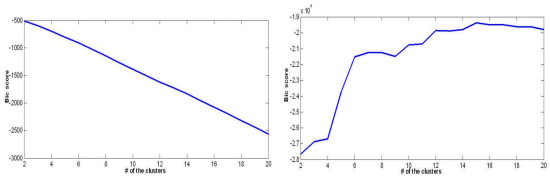

Figure 4 BIC score vs. the number of event clusters Left: pop superstition, Right: Chinese classical music; X-axis: the number of the clusters, Y-axis: BIC score)

5. CONCLUSION

We propose an unsupervised learning approach to detecting and segmenting musical events from beat structures in music. The proposed algorithm combines two data-driven techniques, top-down *k*-means clustering and BIC-based model selection, to automatically detect musical structures that can well characterize music signals. Our results show that the detected musical events have some perceptual meaning, and match well with our experience in spectrogram reading. These musical events are efficient and compact representation of the musical signals and can be used to index and organize a large collection of music documents. They can also be used in many musical content related applications, such as digital music library, music summarization, and music identification.

6. REFERENCES

1. E. D. Scheirer, "Bregman's chimerae: music perception as auditory scene analysis," *Prof. of 4th ICMPC*, 1996.
2. G. Peeters, A. L. Burthe and X. Rodet, "Toward automatic music audio summary generation from signal analysis," *Proc. of ISMIR*, 2002.
3. J. J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," *Proc. of 110th AES Convention*, 2001.
4. J. T. Foote and M.L. Cooper, "Media segmentation using self-similarity decomposition," *Proc. SPIE Storage and Retrieval for Multimedia Databases*, pp.167-175, 2003.
5. L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
6. M. Goto & Y. Muraoka, "Real-time Beat Tracking for Drumless Audio Signals: Chord Change Detection for Musical Decisions," *Speech Communication*, Vol. 27, No.3-4, pp.311-335, 1999.
7. R. B. Dannenberg and N. Hu, "Pattern Discovery Techniques for Music Audio," *Proc. of ISMIR*, pp.63-70, 2002.
8. R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Second edition, Wiley, 2001.
9. S. Dixon, "Automatic Extraction of Tempo and Beat from Expressive Performances," *Journal of New Music Research*, Vol. 30, No.1, pp.39-58, 2001.
10. S. Gao and C.-H. Lee, "An adaptive learning approach to music tempo and beat analysis," to appear in *Proc. ICASSP 2004*.
11. S. Gao, N. C. Maddage and C.H. Lee, "A Hidden Markov Model Based Approach to Music Segmentation and Identification," *Proc. of ICICS-PCM'03*, Singapore, Dec. 2003.
12. W. Chai and B. Vercoe, "Structural analysis of musical signals for indexing and thumbnailing," *Proc. of ACM/ IEEE Joint Conference on Digital Libraries*, 2003.