A Robust Approach to Mining Repeated Sequence in Audio Stream

Jiansong Chen¹, Lei Zhu¹, Bailan Feng¹, Peng Ding¹, Bo Xu¹²

¹Digital Content Technology Research Center, Institute of Automation
²National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100190, China

{jshen, lzhu, blfeng, pding, xubo}@hitic.ia.ac.cn

Abstract

In multimedia stream, repeated sequences, e.g., commercials, jingles, usually imply potentially significant information. Therefore, mining repeated sequence is an important approach to analyzing multimedia content. This paper reports on a robust unsupervised technique of discovering repeated sequence in audio stream. Different from former research, our approach transforms the repeated sequence detection task into a Hidden Markov Model (HMM) decoding problem in a similarity trellis. To resist the false and missing matches in real application, we present a soft definition of repeated sequence, termed as maximal loosely repeated sequence (MLRS), as the objective for detection, and use a Viterbi-like algorithm to mine all the MLRSs in the stream. In addition, we propose a novel metric to evaluate the repeated sequence detection algorithm. Experiments both on simulated data and real broadcast data demonstrate the effectiveness of our method.

Index Terms: repeated sequence detection, HMM decoding problem, Viterbi-like algorithm

1. Introduction

With the rapid development of multimedia and internet technology, more and more multimedia resources are easily accessible nowadays. Mining and discovering meaningful information from huge amount of multimedia data becomes a significant issue to be resolved. Specially, repeated sequences in the multimedia stream usually associate with high-level concepts, e.g., TV commercials, jingles etc. Therefore, detecting repeated sequence is an important approach to analyzing multimedia content.

Much effort has been devoted on the problem of detecting repeated sequence in multimedia stream. The methods could be concluded into four steps as shown in Figure.1. Firstly, a long multimedia stream is segmented into a sequence of basic units, such as clips, video shots, etc. Secondly, audio or visual descriptors are used to perform the description of the basic units. Based on the description, thirdly, the units are measured by their content similarity and the similar ones are grouped together. In this process, indexing or clustering technique is commonly used for acceleration. Finally, repeated sequences are identified from the repeated units.

Most of the former research has focused on robust audio or visual feature extraction and fast similarity calculation. For example, Gauch et al [1] [2] propose a shot-based solution, in which video shots are described by moment vectors and indexed using perceptual hashing. Covell et al [3] construct an advertisement detection system by both audio and visual repetitions. Herley [4] exploits dimension reduction techniques on the audio portion of multimedia stream to make search and buffering feasible. Both Wang [5] and Berrani [6] introduce clustering technique to accelerate the similarity calculation process. In [7], Yuan et al analyze the complexity of the task and propose an efficient algorithm to mine repeated sequences by finding continuous paths in similarity trellis.

As for the core algorithm of repeated sequence identification, the main idea of above mentioned methods just roughly merge the adjacent repeated units into repeated sequences. However, on one hand, due to the noise in the signal or/and the usage of indexing technique, not all the similar units could be retrieved for every unit by the system. Thus, a whole repeated sequence is possible to be chopped into several repeated parts. On the other hand, adjacent units may have considerable content redundancy, consequently they may be identified as repeated units to the same unit simultaneously. All the problems considerably degrade the performance of the traditional approaches in real applications.

In this paper, we propose a robust approach to detecting repeated sequences in a formal framework. The approach follows the typical system flowchart of Figure.1. It first splits the multimedia stream into a sequence of segments of identical length with overlaps. Subsequently, a kind of audio fingerprint feature is extracted for similarity measurement. In order to quickly retrieve the repeated segments, a hashing indexing technique is used in the similarity search process. Each segment is returned by the system a list of similar segments leading to a similarity trellis as illustrated in Figure.2. Then, we transform the repeated sequence identification task to a HMM decoding problem. In the framework, we define the detection objective as MLRS, which can be regarded as local optimal path in the trellis. Traditional Viterbi algorithm for HMM decoding could not be applied directly under the context, thus we adopt a Viterbi-like algorithm to detect all the local optimal paths. Due to the soft definition of MLRS and the Viterbi-like algorithm, the approach have a strong resistance to signal deformations. In addition, a novel metric is proposed to effectively evaluate the performance of the approach.
2. Similarity Trellis Construction

As mentioned before, the audio stream is chopped into a sequence of audio segments with identical length of 4 seconds and overlap of 2 seconds. Each segment is labeled by its temporal index. The audio feature used in our system is based on an audio fingerprinting method [8] proposed by Haitsma and Kalker in 2002. In the fingerprint extraction, the audio signal is windowed into 32ms length and 12ms overlap and then transformed to the frequency domain using the discrete Fourier transform. 33 non-overlapping frequency bands ranging from 300Hz to 2000Hz over logarithmic spacing are selected to calculate their adjacent spectral energy differences (simultaneously along the temporal and frequency axes). It results in a 32-bit binary sub-fingerprint feature for every frame. Normalized hamming distance is adopted to measure the distance of two frames. To improve the search efficiency, an indexing technique based on hash table [8] is implemented.

Repeated segments are detected in two steps. The first step is based on the frame collisions in the hash table. For a query segment, we take all its frames to query in the hash table. To guarantee all repeated segments can be retrieved, we reserve the frames whose bit errors are less than 3 with the query frame instead of reserving only perfect matching frames. The segments that the retrieved frames lied in are considered as the candidate repeated segments for the query segment. The second step is to validate the candidate segments based on the acoustic similarity of the whole segments. Since temporal shift may exist, it is not proper to accumulate the distances of corresponding frames from two segments directly. Thus Dynamic Time Wrapping (DTW) distance [9] which can compensate the shift bias is adopted to measure the distance of two segments. For two segments, if their DTW distance is less than a threshold \( T \), then they are identified to be the repeated segments. In the similarity search, we only reserve the segments which are prior to the query segment in temporal axis to eliminate redundant repetitions. For the \( n^{th} \) segment in the stream, the system will retrieve a list of similar segments, denoted as \( l_n \). Finally, it yields a similarity trellis as shown in Figure 2.

3. Repeated Sequence Identification

In this section, we formulate the repeated sequence identification problem as a HMM decoding problem. Before that, we state the HMM decoding problem as follows. Given an observation sequence \( O = \{ o_1, o_2, \ldots, o_N \} \) and a HMM model \( \lambda = \{ A, B \} \), calculate the most likely sequence of hidden states \( S = \{ s_1, s_2, \ldots, s_N \} \) that produces this observation sequence \( O \). In the HMM model, \( A \) is the state transfer probability matrix, and \( B \) is the observation probability matrix.

An audio stream consisting of \( N \) segments can be directly regarded as the emission sequence in HMM. A retrieved segment \( r_{n,j} \) in list \( l_n \) represents \( j^{th} \) state \( s_{n,j} \) in the \( n^{th} \) stage. Note that there are different states at different time instants. Naturally, the probability distribution of an observation \( o_n \) in the state \( s_{n,j} \) is calculated as the acoustic similarity scores of corresponding segments:

\[
b_{n,j} = p(o_n|s_{n,j}) = \begin{cases} 
score(o_n, r_{n,j}), & \text{if } r_{n,j} \in l_n, \\
0, & \text{otherwise} \end{cases} \tag{1}
\]

where, \( 1 \leq j \leq M_n \), and \( M_n \) is the number of members in \( l_n \).

For two state \( s_{n,j} \) and \( s_{n+1,j \downarrow} \), ideally, \( s_{n,j} \) can transfer to \( s_{n+1,j \downarrow} \), only if the segment \( r_{n+1,j \downarrow} \) is the exactly next segment of \( r_{n,j} \) along the temporal axis. However the requirement is really strict in practice. As mentioned above, the similarity search step may cast numbers of false and missing matches due to kinds of signal deformations. So we relax the state transformation constraint as follows:

\[
a_{i,j} = p(s_j|s_i) = \begin{cases} 
\theta, & \text{if } j - i = 1 \\
(1 - \theta)/\Delta t, & \text{if } j - i = 0, 2, \ldots, \Delta t \\
0, & \text{otherwise} \end{cases} \tag{2}
\]

where, \( s_i \) and \( s_j \) are two states and footnotes \( i \) and \( j \) are temporal labels of their corresponding segments; \( \theta \) is the state transition probability, when \( r_j \) is the exactly next segment of \( r_i \), and it is set to 0.6 in the experiments; \( \Delta t \) is a positive integer parameter, representing the temporal distance between the current segment and its furthest transferable segment. The intuition behind equation (2) is that, a reference segment is more probable to transfer to its exact next segment, besides, it could also transfer to itself and the other following segments in lower odds.

![Figure 2: Illustrations of similarity trellis. Each node in the trellis denotes a fixed-length audio segment labeled by its temporal index. The sequence in the top-row denotes the audio stream of length \( N \), and each column represents the segments matched with the segment in the top-row. In the trellis, the segment 4 should be matched with segment 135, however, it is missed by the system resulting in a broken state sequence. On the other hand, because of large overlap between adjacent segments, the segments 132 and 133 are both matched with segment 273, which brings many redundant pathes. The proposed method enlarges the matching scope and can connect the broken sequences. In addition, it finds the local optimal sequence from all the potential sequences to wipe out redundant sequences.](https://example.com/figure2.png)
Another important issue in problem transformation is to define the objective for repeated sequence detection under the framework. In this paper, we define the objective as maximal loosely repeated sequence. Specifically, given an observation sequence $O(i,j) = (o_0, o_1, \ldots, o_m, \ldots, o_{j-1}, o_j)$ and one of its possible state sequence $S(i,j) = (s_1, s_1, \ldots, s_m, \ldots, s_{j-1}, s_j)$, if (1) any state in previous $\Delta t$ time instants cannot transfer to state $s_i$; (2) the states can transfer from the beginning state $s_i$ to the last state $s_j$ in the interior of the state sequence $S(i,j)$; (3) $s_j$ cannot transfer to any states in its posterior $\Delta t$ stages, then the state sequence $S(i,j)$ is identified as a maximal loosely repeated sequence, abbreviated as MLRS. In fact, the definition uses the relations of state transition to determine the beginning and end of the repeated sequences. Up to now, we have transformed the repeated sequence detection issue to the decoding problem under the HMM framework.

Standard algorithm for HMM decoding is the Viterbi algorithm traces back the local optimal paths at each time and check whether repeated sequences are correctly detected or not. It enlarges the matching scope and then can tolerate more noise.

Algorithm 1 Viterbi-like algorithm for repeated sequence identification.

Initialization: $t = 1, 1 \leq j \leq M_t$

$\delta(1,j) = b_{t,j}$

Path$(1,j) = (s_{1,j})$

Reursion: $2 \leq t \leq N-1, 1 \leq j \leq M_t$

1: Search the best state in previous time instants, which has the best score transferring to state $s_{t,j}$:

$\delta(t,j) = \delta(t^*,i^*) \cdot a_{i^*,j} + b_{t,j}$

2: Calculate the best score for any state $s_{t,j}$:

$\delta(t,j) = \delta(t^*,i^*) \cdot a_{i^*,j} + b_{t,j}$

3: Backtrack the best path for any state $s_{t,j}$:

if $a_{i^*,j} > 0$ then

$Path(t,j) = (Path(t^*,i^*), s_{t,j})$

else

$Path(t,j) = s_{t,j}$

end if

4: Check whether the best path to state $s_{t,j}$ is a MLRS:

if $Path(t,j)$ satisfy the third condition of MLRS then

Save $Path(t,j)$ to output set

end if

4. Experiments

In this section, we prepare two data sets, simulated data and real broadcast data, for evaluation. For simulation, we collect 200 distinct audio clips of commercials from SCTV, a Chinese TV channel. The length of the audio clips ranges from 5 seconds to 30 seconds. We randomly select 20 clips from the clip library and randomly position them five times in a stream. In addition, adjacent clips are separated by different remaining commercial clips. We repeat the process three times yielding three distinct test streams. To test the robustness of Algorithm 1, white noises are added to the original test streams with different Signal-to-Noise ratios (SNR): 5dB, 10dB, 15dB and 20dB etc. The real experimental data is collected from a Chinese radio broadcast channel, Beijing News Radio, about 24 hours. We obtain the ground truth by manually labeling the repeated objects. There are totally 178 repeated sequences with 584 pairs of repetitions in the database. The audio files in the experiments are saved as 16 kHz, 16-bit, mono wav format.

Most of the previous work [3] [7] evaluate the performance of repeated sequence detection by calculating precision rate and recall rate based on the number of occurrences of repeated sequences. However, a repetition is always corresponding to a channel, Beijing News Radio, about 24 hours. We obtain the ground truth by manually labeling the repeated objects. There are totally 178 repeated sequences with 584 pairs of repetitions in the database. The audio files in the experiments are saved as 16 kHz, 16-bit, mono wav format.

Most of the previous work [3] [7] evaluate the performance of repeated sequence detection by calculating precision rate and recall rate based on the number of occurrences of repeated sequences. However, a repetition is always corresponding to a channel, Beijing News Radio, about 24 hours. We obtain the ground truth by manually labeling the repeated objects. There are totally 178 repeated sequences with 584 pairs of repetitions in the database. The audio files in the experiments are saved as 16 kHz, 16-bit, mono wav format.

Most of the previous work [3] [7] evaluate the performance of repeated sequence detection by calculating precision rate and recall rate based on the number of occurrences of repeated sequences. However, a repetition is always corresponding to a channel, Beijing News Radio, about 24 hours. We obtain the ground truth by manually labeling the repeated objects. There are totally 178 repeated sequences with 584 pairs of repetitions in the database. The audio files in the experiments are saved as 16 kHz, 16-bit, mono wav format.
rate (RR) for evaluation are defined as:

\[ PR = \frac{\# \text{ correctly detected pairs of repetitions}}{\# \text{ detected pairs of repetitions}} \]  

\[ RR = \frac{\# \text{ correctly detected pairs of repetitions}}{\# \text{ correct pairs of repetitions}} \]  

In the proposed method, \( \Delta t \) influences the boundaries of detected sequences. In addition, the threshold \( T \) determines the tolerance degree to the matching errors. So we have conducted experiments with different values of the two parameters to evaluate their effects on the system. The experiments are performed on the three pure simulated streams respectively and the precision and recall rates are averaged. Results are shown in Table 1.

In view of the columns in Table 1, it shows a tendency that the performance increases along with the raise of the threshold \( T \) when \( T \leq 0.3 \). However, it begins to decline when the value is bigger than 0.3. This is because a larger value of \( T \) gives more chances to retrieve the similar segments so as to identify a complete sequence. Nevertheless, oversize of the value is bigger than \( T \) also brings much false matches which degrades the precision a lot. Note that the recall rate also goes down a little when \( T > 0.3 \). A sound explanation is that some false matching segments are connected with the real repeated sequences and thus they are not able to be matched with the sequence in ground truth. Similarly, a bigger value of \( \Delta t \) means the system can resist more missing matches whereas oversize of the value may lead to performance degradation because of connecting irrelevant sequences.

To certificate the robustness of the proposed method, we compare it with the algorithm proposed in [7]. In our experiments, both of the two methods use the similarity trellis construction procedure described in Section 2. The experiments are conducted on simulated data with noise of different SNRs and the real broadcast data. The parameters \( T \) and \( \Delta t \) are set to 0.3 and 2 respectively. Table 2 shows that, the proposed method achieves better results than that of [7] on both simulated data and real broadcast data. This is, on one hand, due to the soft definition of MLRS, which allows existence of slight sequence-broken phenomenon. On the other hand, the HMM decoding framework have the mechanism to choose the local optimal paths removing lots of redundant paths. Table 2 also shows that as the decrease of SNR, both performances of the two methods degrade accordingly. Basically, the more noise is added the worse performances are obtained.

<table>
<thead>
<tr>
<th>( T )</th>
<th>( \Delta t = 1 )</th>
<th>( \Delta t = 2 )</th>
<th>( \Delta t = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( PR )</td>
<td>( RR )</td>
<td>( PR )</td>
<td>( RR )</td>
</tr>
<tr>
<td>0.18</td>
<td>0.928</td>
<td>0.637</td>
<td>0.969</td>
</tr>
<tr>
<td>0.22</td>
<td>0.931</td>
<td>0.897</td>
<td>0.982</td>
</tr>
<tr>
<td>0.26</td>
<td>0.942</td>
<td>0.983</td>
<td>1.000</td>
</tr>
<tr>
<td>0.30</td>
<td>0.950</td>
<td>0.992</td>
<td>1.000</td>
</tr>
<tr>
<td>0.34</td>
<td>0.944</td>
<td>0.988</td>
<td>1.000</td>
</tr>
<tr>
<td>0.38</td>
<td>0.831</td>
<td>0.983</td>
<td>0.891</td>
</tr>
<tr>
<td>0.42</td>
<td>0.262</td>
<td>0.980</td>
<td>0.243</td>
</tr>
</tbody>
</table>

Table 2: Comparison of different methods on different data.

<table>
<thead>
<tr>
<th>Data</th>
<th>Proposed method [7]</th>
<th>[Proposed method]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNR=20dB</td>
<td>0.883</td>
<td>0.888</td>
</tr>
<tr>
<td>SNR=15dB</td>
<td>0.796</td>
<td>0.768</td>
</tr>
<tr>
<td>SNR=10dB</td>
<td>0.679</td>
<td>0.407</td>
</tr>
<tr>
<td>SNR=5dB</td>
<td>0.201</td>
<td>0.140</td>
</tr>
<tr>
<td>Real broadcast data</td>
<td>0.902</td>
<td>0.913</td>
</tr>
</tbody>
</table>

6. Acknowledgement

The work was supported by “The National Key Technology R&D Program”, grant No. 2009BAH40B03.

7. References