Audio Segment Retrieval Using a Synthesized HMM

Atulya Velivelli
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
velivell@ifp.uiuc.edu

ChengXiang Zhai
Computer Science Department
University of Illinois at Urbana-Champaign
czhai@cs.uiuc.edu

Thomas S. Huang
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
huang@ifp.uiuc.edu

ABSTRACT

In this paper, we propose a general approach to audio segment retrieval using synthesized HMMs. The approach allows a user to query audio data of any length by one or more example audio segments and find similar segments. The basic idea of our approach is to first train a theme HMM using the given examples and a general background HMM using all the audio data, and then combine these individual HMMs to form a synthesized "Background-Theme-Background" HMM. This synthesized HMM can then be applied to any audio stream as a parser to detect the most likely theme segment. A major advantage of this approach is that it does not assume any predefined segment boundaries as in previous work, thus can be expected to retrieve theme segments with more accurate boundaries. Preliminary experiments with music detection have shown promising results.

1. INTRODUCTION

There has been a large increase in the amount of digital multimedia data over the past decade, specifically in the case of digital audio data. To make effective use of any audio data, we must be able to classify and retrieve audio information in terms of the audio content. Indeed, the problem of audio retrieval has attracted much attention recently [1].

A very interesting retrieval function is *query by example*, i.e., a user specifies a sample audio segment and the system would find similar segments from audio data. While there are other ways of querying audio data (e.g., by audio features or words), query by example is most convenient for a user and is often also a relatively accurate way of specifying the information need. For example if we have a video about the different rocket launches in the last century. The audio theme corresponding to a rocket launch is an explosion. The user may be interested in retrieving only those segments of audio which correspond to an explosion.

To address the retrieval problem in a general way, we do not make any assumption about the length of any interesting audio segment. This is quite different from some other works (e.g.,[2, 3]) that assume partitioned audio segments, and perform audio classification of these segments to match and retrieve the relevant segment. Indeed we also face the challenge of *identifying* the boundaries of a segment, in addition to matching a candidate segment with the query segment.

In this paper, we present a general audio segment retrieval method based on Hidden Markov Models (HMMs) that can *simultaneously* identify and match an audio segment with the query. We call the interesting audio segment that needs to be matched with the query as Theme audio segment. Our idea is based on the observation that a theme audio segment is often preceded and followed by sound track of a different audio category, and thus the problem of finding a theme audio segment boils down to identifying and detecting the pattern of a theme audio segment that is temporally "sandwiched" between any other kind of audio patterns, which we refer to as "general background audio." The audio information is modeled as a continuous Hidden Markov Model, representing the observation density as a Gaussian probability density function [4]. We design HMMs that can model such a simple "sandwich-like" structure in audio data . More specifically, given any query segment(s), we would first create a "theme HMM" that would capture the patterns of an interesting audio segment. We then train another "background HMM" to model the general patterns of all other *non-interesting* audio information. Finally, we construct a single synthesized HMM by putting the theme HMM in between two background HMMs to reflect the sequential "sandwich" structure. To retrieve interesting segments from any audio stream, we would fit such a concatenated HMM to the audio stream data and use the Viterbi algorithm to identify which part of the audio data corresponds to the theme HMM. This procedure is repeated in an iterative manner until N best segments have been retrieved. This provides an automatic and general way to simultaneously set the boundary for a segment and match the segment with the query segment.

There is some previous work on audio retrieval. For example, Liu and Huang perform a segmentation and unsupervised search by presenting an audio query example, uses a Gaussian mixture model(GMM) probability density function based metric as a similarity measure [5]. In [6, 7], HMMs are used to retrieve musical performances by using a short music piece as a query. Our work differs from the previous work in that we assume no predefined segment boundaries and use HMMs to perform audio segmentation and matching *simultaneously*.

Some preliminary evaluation of our method using the NIST Special Database 26 has shown promising results and suggests some interesting future research directions.

In the rest of the paper, we first propose a general approach to audio segment retrieval using a synthesized HMM, and then present some preliminary evaluation results of using the proposed method for music detection.

2. A SYNTHESIZED HMM APPROACH TO AUDIO SEGMENT RETRIEVAL

Given one or more examples of some theme clips, our task is to find similar audio segments in a stream of audio data. We thus need to accomplish two different, but related tasks: (1) Identify boundaries of any reasonable candidate segment; (2) Match a candidate segment with the query segment(s). Note that the two tasks interact with each other – the location of the optimal boundary of a theme segment is affected by how well the "proposed" segment would match the audio query; similarly whether a segment matches the query well highly depends on where you set the boundary for the segment. It is thus desirable to accomplish both tasks at the same time with a single mechanism. The synthesized HMM approach to be described below achieves this goal through modeling the *whole* stream of audio data with a compound HMM formed based on three individually trained HMMs. An HMM is essentially a probabilistic finite state automaton with output [4]. As a probabilistic model, it is especially suitable for modeling sequential data such as audio.

The main idea of the synthesized HMM approach is as follows.

1. We train a theme HMM based on the query example audio clips provided by a user. This HMM captures specific patterns associated with the theme that the user is interested (e.g., explosion).
2. We train a background HMM using a large amount of randomly selected audio data (e.g., all the data available to us). The background HMM thus reflects the general characteristics of any audio data. Since the training data used for training the background HMM are dominated by the non-interesting audio segments, the background HMM is a good approximation of a model for "non-interesting" segments.
3. We form a synthesized HMM by putting the theme HMM in between two background HMMs, resulting in an HMM with a left-to-right global structure like "background + theme + background". The HMM thus has two kinds of states – theme states and background states.
4. We apply the synthesized HMM to parse each audio document. Specifically, we use the Viterbi algorithm to find out the most likely underlying state transition sequence given the observed audio data. The part of the audio data that corresponds to all the theme states (as opposed to the background states) is regarded as the most likely interesting theme segment in the given audio data.
5. Once we detect the most likely theme segment from an audio document, we are left with potentially two residue subdocuments. The same synthesized HMM can be further applied to each to retrieve more theme segments if any.

We now describe some of these ideas in more detail.

2.1 Joining two HMMs

Since the synthesized HMM is formed by concatenating two background HMMs with a theme HMM, we first discuss how we can join any two HMMs in general.

The three major components in describing our HMM are: (1) A set of states, including a *start* and an *end* state; (2) A state-to-state transition matrix that gives the probability of going from one state to another; and (3)Output observation probability density function for each state, In case of continuous signals the observable process is modeled using a continuous observation density, which is defined for each state. It specifies how the observed data is generated by each state.

In our model, neither the start state nor the end start generates any output, so they are non-emitting states. The observation density for the audio data is a Gaussian probability density function [4].

Suppose H_1 and H_2 are two HMMs with states $S_1 = \{s_0, ..., s_M\}$ and $S_2 = \{s_{M+1}, ..., s_N\}$, respectively. s_0 and s_M are the start state and end state of H_1, and s_{M+1} and s_N are the start state and end state of H_2. A *joined HMM* H_{1-2} is simply the HMM formed by "forcing" an extra transition (with probability one) from the end state of H_1 to the start state of H_2. After eliminating the non-emitting states, the joined HMM can be defined more precisely as follows:

- Its state set S is the union of the states of H_1 and H_2, excluding the end state of H_1 and the start state of H_2. That is, $S = S_1 \cup S_2 - \{s_M, s_{M+1}\}$. The total number of states is $N - 2$.
- Its observation density functions are the union of the observation density functions of H_1 and H_2. That is, we keep all the observation density functions. The observation density function of a continuous observation sequence $\mathbf{O}$ from state s_i, $b_i(\mathbf{O})$ can be written as
$$b_i(\mathbf{O}) = \begin{cases} b_i^1(\mathbf{O}) & \text{if } 1 \le i \le M-1 \\ b_i^2(\mathbf{O}) & \text{if } M+1 \le i < N \end{cases}$$
where the superscripts 1 and 2 indicate the observation density function of H_1 and H_2 respectively.
- Its transition matrix A is constructed by (1) keeping all the transitions in H_1 except those going into the end state; (2) keeping all the transitions in H_2 except those going out from the start state; (2) adding a new transition from s_i to s_j where s_i is any state in H_1 that goes to the end state and s_j is any state in H_2 to which there is a transition from the start state. Formally, the probability of a transition from s_i to s_j in H_{1-2} $(a_{i,j})$ is defined as
$$a_{i,j} = \begin{cases} a_{i,j}^1 & \text{if } 0 \le i, j \le M-1 \\ a_{i,j}^2 & \text{if } M+1 \le i, j \le N \\ a_{i,M}^1 a_{M+1,j}^2 & \text{if } 0 \le i \le M-1, \\ & \text{ but } M+1 \le j \le N \\ 0 & \text{otherwise} \end{cases}$$
Note that we use $a_{i,j}^1$ and $a_{i,j}^2$ to indicate the transition probability given by the original H_1 and H_2, respectively.

An example of joining two HMMs is illustrated in Figure 1.

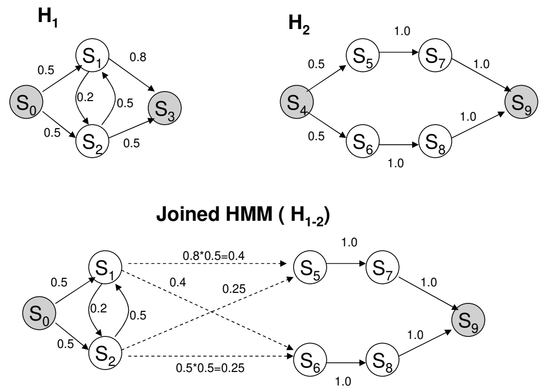

Figure 1: An example of joined HMM. Shaded states are start or end states.

2.2 Detecting and ranking theme segments

To detect a theme segment in any audio stream, we use a synthesized HMM that is formed by joining a background HMM with a theme HMM and then with another background HMM as shown in Figure 2. Treating the given audio data as the output from such a synthesized HMM, we can use the Viterbi algorithm to find the most likely transition state path given the observed audio data. The part of the audio data corresponding to the states of the theme HMM is then regarded as the detected theme segment. Note that the two background HMMs are functionally identical, but their states are *distinct* (essentially "re-labeled") in the synthesized HMM. As a result, the left background HMM would generate any audio data before the theme segment while the right one would generate any audio data after the theme segment.

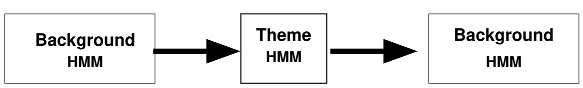

Figure 2: Synthesized HMM with the theme HMM "sandwiched" between two background HMMs.

Our synthesized HMM is designed to detect one single most likely theme segment, but we can apply the HMM repeatedly to any remaining part in the audio data to detect more theme segments, as shown in Figure 3. Specifically, once we detect one theme segment, it separates the audio document to two "residue documents." We then use the same HMM to detect a theme segment from each residue document. As shown in the Figure 3, the bar in the middle indicates the first theme segment detected from the output of the Viterbi decoding. We then apply Viterbi decoding again to the two neighboring audio segments of the first theme segment, to its left and right respectively. This returns two candidate theme detections as indicated by the two bars on the sides. Naturally, these two new theme segments should be ranked below the first one that we have detected. But we still need to decide which one of the two new segments is relatively more significant. For this purpose, we use the following likelihood ratio to measure the "significance" of a detected segment. Suppose **O** represents the audio data from which we have detected a theme segment, we compute the ratio of the likelihood of the data given by the synthesized HMM (i.e.,HMM_{B-T-B}) to that of the data given by just the background HMM (i.e., HMM_B), and use this ratio to rank the theme segments.

$$ratio = \frac{p(\mathbf{O}/HMM_{B-T-B})}{p(\mathbf{O}/HMM_B)} \tag{1}$$

Intuitively, since HMM_{B-T-B} differs from HMM_B mainly by the presence of the theme HMM, this likelihood ratio indicates how much better the theme HMM can help fit the data. The more helpful is the theme HMM, the more likely that the data would contain a true theme segment, thus the segment detected from the data should be ranked higher.

In general the selection of N best segments is done as shown in Figure 4

3. MUSIC DETECTION EXPERIMENTS

In this section, we present some preliminary evaluation results of using the proposed synthesized HMM for music detection. We use the NIST Special Database 26 [1], which has about 3 GB of video data, for training and testing purpose. The audio data of three files in the database is annotated for music data. This music data is used for training purpose.

3.1 Background HMM construction

We use the mediaconvert audio decoder software on SGI workstation to extract the sound track of all the training files to different audio files, and using the HTK toolkit[8] we represent these audio files using Mel-frequency cepstrum coefficients+energy coefficient(Mfcc+E) vectors as features. Using the HTK toolkit we model this Observation sequence of (Mfcc+E) vectors as a 5 state continuous observation density Hidden Markov model. This HMM is Left to right as shown in Figure 5, though the figure does not show all the transition arcs in the left to right directions, all the transitions in the left to right direction are possible. Each state observation density $b_j(\mathbf{O})$ is modeled as a Gaussian probability density function.

$$b_j(\mathbf{O}) = \mathcal{N}(\mathbf{O}, \mu_\mathbf{j}, \mathbf{U_j}), 1 \leq j \leq N.$$

$\mu_\mathbf{j}$, $\mathbf{U_j}$ are the mean vector and the Covariance matrix for state j. The first state and the last state are non-emitting.

3.2 Theme(Music) HMM construction

In the case of music segment detection the "theme" is music, hence we construct a music HMM. From the audio files, we manually extract only those segments that correspond to music and represent using (Mfcc+E) vectors. Using the HTK toolkit we model this observation sequence of (Mfcc+E) vectors as a 4 state continuous observation density Hidden Markov model. Similar to the Background HMM , each state observation density is modeled as a Gaussian probability density function. This HMM is Left to right as shown in Figure 5, though the figure does not show all the transition arcs in the left to right directions, all the transitions in the left to right direction are possible.

3.3 Synthesized HMM

Using the approaches described in Section 2, we can first combine the background HMM and the theme (music) HMM, and then

[1]See http://www.nist.gov/srd/nistsd26.htm for more information.

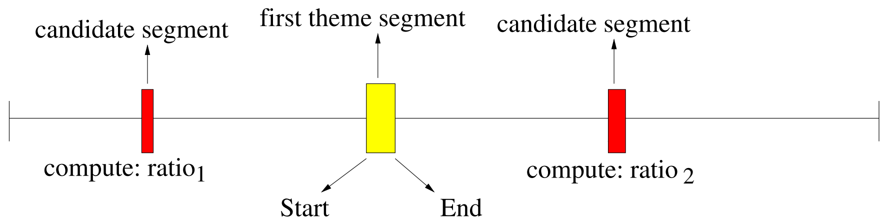

Figure 3: $ratio_1$ and $ratio_2$ are evaluated over the duration of the candidate audio segments before and after the first theme segment detected.

•$j \leftarrow 1, i \leftarrow 1$
• While $j \leq N$
    Compute the likelihood ratio for the candidate segment on the left of $audio_{j-1} \rightarrow ratio_i$
    $seg_i \leftarrow [start_i \; end_i]$
    $i \leftarrow i + 1$
    Compute the likelihood ratio for the candidate segment on the right of $audio_{j-1} \rightarrow ratio_i$
    $seg_i \leftarrow [start_i \; end_i]$
    $i \leftarrow i + 1$
    $audio_j \leftarrow seg_{argmax\{ratio_1, ratio_2, \ldots, ratio_i\}}$
    $ratio_{argmax\{ratio_1, ratio_2, \ldots, ratio_i\}} \leftarrow 0$
    $j \leftarrow j + 1$
end

Figure 4: Audio Segment Ranking Algorithm. $audio_j$ is the j^{th} best audio segment detected among the N best, i is the index of each candidate segment seg_i, $start_i$ and end_i denote the start and end of seg_i.

combine the background+music HMM with the background HMM to obtain a synthesized "background+music+background" HMM shown in Figure 5.

3.4 Music segment retrieval

With this synthesized HMM, retrieving a music segment is essentially matching the "background+music+background" pattern in the audio data. Specifically, we do the Viterbi decoding of the audio observation sequence **O** to find the most likely state transition path, given the background+music+background HMM parameters. Since the HMM is left to right, the state transition path would in general consist of a sequence of background HMM states followed by a sequence of music HMM states and then by another sequence of background HMM states. The portion of the audio data corresponding to the music HMM states is the recognized music segment. We apply the algorithm described in Figure 3, to iteratively detect the top 4 most likely music segments for each audio document.

3.5 Results

For testing purpose, we use 4 audio files in the database which were not used for training the music HMM. The files in the database have short music segments with most of them of duration less than 10s interspersed in the sound track. Hence we decide to evaluate the performance of our scheme by retrieving the top N music segments from each file. Considering the duration of the test documents we decide that N=4 is an appropriate value, which means that we test the effectiveness of our scheme for the top 4 music detections. For each retrieved music segment, we manually verify whether it is indeed a music segment.

The results on all the four files are shown in Table 1. A "hit" means that the detected segment is music, while a "miss" means that the detected segment is not music. We see that in many cases, the retrieved music segments are indeed music segments. We also see that there are some mistakes. In the case of the first audio document the second music segment detected was not music, though as indicated by the ground truth there were actually 4 significant music segments present. In the case of the fourth audio document, the first music segment is hit while the other music segments are missed. But, the Ground Truth indicates that there is only one significant music segment present, which has been successfully detected.

Figure 6 shows the same results by plotting the precision for different number of music segments detected, which is defined as

$$precision = \frac{Number\ of\ correctly\ detected\ Musicsegments}{Total\ number\ of\ detected\ music\ segments}$$

The number of music segments detected is an indicator of recall.

We see that the precision of the top retrieved segments from all the files is 100%, which means all the top retrieved segments are true music segments. As we retrieve more segments, the precision drops. For example, among the top 4 retrieved segments, the precision drops to about 0.56. Note that the drop of precision may be due to the lack of additional music segments in a file as in the fourth audio document. This also illustrates a weakness of our method – we can only rank the segments, and the user must decide when to stop retrieving more segments. It would be interesting to explore how we can automatically judge whether there is any additional music segment in any remaining portion of an audio file. This is one of our future research directions.

Note that we cannot generalize these results in any way, since they

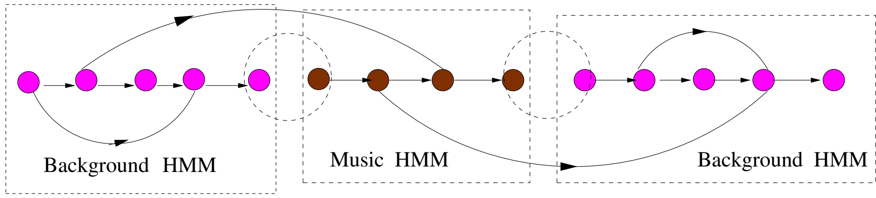

Figure 5: The background+music+background HMM is synthesized by combining by combining the non-emitting states of the background and music HMM's as shown by the circles.

Table 1: Detection results for the top 4 music segments

Audio	Music1	Music2	Music3	Music4	Ground Truth
1	hit	miss	hit	hit	4 significant music segments in this doc
2	hit	hit	hit	miss	4 significant music segments in this doc
3	hit	hit	miss	miss	3 significant music segments in this doc
4	hit	miss	miss	miss	1 significant music segment in this doc

are based on an extremely small scale of experiments. However, these preliminary results are quite encouraging, and they do provide some evidence that the proposed method is promising.

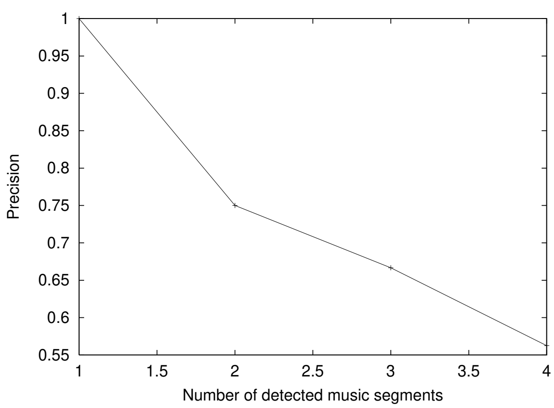

Figure 6: A plot of precision for different number of music segments detected.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a general approach to audio segment retrieval using synthesized HMMs. The approach allows a user to query an arbitrary audio stream by one or more example audio segments. A major advantage of this approach over other existing approaches is that it does not assume any predefined segment boundaries, thus can be expected to retrieve theme segments with more accurate boundaries.

Preliminary experiments with using the proposed approach for music detection show some quite interesting results. In particular, the first music segment detected in each audio file is correct in all cases. The subsequent detections do not have the same quality as the first detection, but are still reasonably accurate when there exist more music pieces in the remaining audio file. This indicates that the proposed approach is effective in retrieving music segments with flexible boundaries. Even though our experiments involve a very small data set and we have only tried one theme which is music detection, the results are quite encouraging.

There are many interesting future research directions. The most important one is to evaluate the method with more experiments at a larger scale and with more varieties of themes. In addition to the extraction accuracy, we should also evaluate the accuracy in detecting the segment boundaries. Another interesting direction is to study how to automatically stop the iterative detection process when there seems to be no additional theme segment in the remaining audio data. One possibility is to apply machine learning techniques (e.g., logistic regression) to train a decision boundary using known theme segments. We believe that, in addition to the likelihood ratio, the length of the detected theme segment may be also a good feature for this purpose and for ranking in general. Finally, although in our experiments the theme HMM trained with annotated music data out of only three audio files appears to be quite effective, in general, effective training of an HMM would require a fair amount of training data, thus a direct application of the proposed method to one single example segment is unlikely to work very well. It would be very interesting to further study how to effectively retrieve theme segments using only one or very few example segments. In such a case, techniques from information retrieval, such as pseudo feedback and query expansion, might be useful. Another possibility is to "pre-train" a number of generic HMMs

5. REFERENCES

[1] J. Foote, "An overview of audio information retrieval," *ACM Multimedia Systems J.*, 7(1):2-10, Jan 1999.

[2] T. Zhang and C. C Jay Kuo, "Hierarchical Classification of audio data for archiving and retrieval," *ICASSP 1999*, vol 6, pp. 3001-3004, Phoenix, March 1999.

[3] E. Wold, T. Blum, D. Keislar and J. Wheaton, "Content-Based Classification, search, and Retrieval of Audio," *IEEE Multimedia*, Vol. 3, No. 3, pp. 27-36, Fall 1996.

[4] L. Rabiner and B. H Juang, *Fundamentals of speech*

recognition. New Jersey: Prentice Hall International Inc, 1993.

[5] Z. Liu, Q. Huang, "Content-based Indexing and Retrieval-by-Example in Audio," *ICME 2000,* New York, NY, July 30 - Aug. 2, 2000.

[6] A. S Durey, M. A Clements. "Melody spotting using Hidden Markov Models," *International Symposium on Music Information Retrieval*, pp. 109-117, Bloomington, IN, October 2001.

[7] S. Shalev-Shwartz, S. Dubnov, N. Friedman, Y. Singer. "Robust temporal and spectral modeling for query By melody ," *ACM SIGIR 2002 conference on research and development in information retrieval*, pp. 331-338, 2002.

[8] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev and P. Woodland. *The HTK Book (for HTK Version 3.2)*, Cambridge University Engineering Department, 2002.