CONTINUOUS HMM AND ITS ENHANCEMENT FOR SINGING/HUMMING QUERY RETRIEVAL

Jyh-Shing Roger Jang **Chao-Ling Hsu** **Hong-Ru Lee**
Multimedia Information Retrieval Laboratory
Computer Science Department, National Tsing Hua University
Hsinchu, Taiwan
{jang,leon,khair}@wayne.cs.nthu.edu.tw

ABSTRACT

The use of HMM (Hidden Markov Models) for speech recognition has been successful for various applications in the past decades. However, the use of continuous HMM (CHMM) for melody recognition via acoustic input (MRAI for short), or the so-called query by singing/humming, has seldom been reported, partly due to the difference in acoustic characteristics between speech and singing/humming inputs. This paper will derive the formula of CHMM training for frame-based MRAI. In particular, we shall propose enhancement to CHMM and demonstrate that with the enhancement scheme, CHMM can compare favourably with DTW in both efficiency and effectiveness.

Keywords: HMM, Continuous HMM, Query by Singing/Humming, Melody Recognition via Acoustic Input (MRAI), Speech Recognition.

1 INTRODUCTION

It is well known that HMM (Hidden Markov Models) is a standard and successful approach to speech recognition for the past few decades. In particular, for large-vocabulary speaker-independent speech recognition tasks, CHMM (Continuous HMM) has been exceedingly successful in creating practical real-word applications. However, the use of HMM for melody recognition via acoustic input (MRAI) has not been extensively reported. In fact, most previous approaches to MRAI using HMM employ note-based method, in which the user needs to hum in "ta" or "da" in order to facilitate note segmentation. The note-specific information is then used to build the DHMM (discrete HMM) of a given song (Shifrin et al., 2002; Meek & Birmingham, 2001; Liu & Li 2003). Shih et al. (Shih et al., 2002) proposed the use of CHMM for humming transcription and note segmentation, but it is not used for frame-based MRAI directly. Needless to say, the most obvious drawback of the note-based approach rested in the requirement of humming in "ta" or "da", which could greatly limit the presentation style of the user. On the other hand, if we allow the user to sing the lyrics, then the note segmentation will be error-prone due to the continuous variations in acoustic intensity. As a result, we have proposed a frame-based approach using DTW(Jang et al., 2001; Jang & Lee, 2001; Jang et al., 2004) to achieve high accuracy while allowing the user to use any presentation style that he/she feels most comfortable with.

While DTW is highly effective in retrieving relevant songs, it is computation intensive. An alternative is to use mathematical analysis to design a hierarchical filtering method (Jang & Lee, 2001). In this paper, we look into another direction that uses a frame-based CHMM for achieving a better balance between accuracy and efficiency. Specifically, we shall derive the evaluation and training procedures for CHMM. Moreover, we shall propose an enhancement scheme to be used with CHMM for achieving both efficiency and effectiveness in MRAI. Experimental results demonstrate that the proposed method compares favourably with DTW, and scale up nicely when the size of the database increases.

In order to promote the research of MRAI, we have put the song dataset and the singing voice corpus on the first user's homepage for public access. As far as we know, this is also the first singing voice corpus for MRAI evaluation.

2 CHMM FOR SINGING/HUMMING QUERIES

In our CHMM-based singing query retrieval system, each song (or theme) in the database is represented by a CHMM. We will describe the CHMM structure used in our system. We will also propose two approaches to the parameter identification of the CHMM.

2.1 CHMM Structure

An intuitive way to model the probabilistic characteristics of the pitch of each song (or theme) is by a left-right CHMM where each note is represented by a state. If there are consecutive notes of the same pitch level, then they are represented by a joined state in a CHMM. The following is the CHMM for the song "10 little Indians":

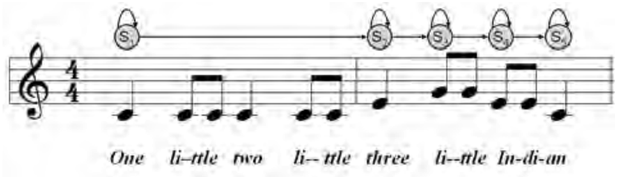

Figure 1. CHMM for "10 Little Indians"

In general, each state in the CHMM model is characterized by a d-dimensional Gaussian probability density function (PDF):

$$g(x,\mu,\Sigma)=\frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left[-(x-\mu)^T\Sigma^{-1}(x-\mu)/2\right],$$

where x is a d-dimensional observation vector, μ and Σ are the mean vector and covariance matrix of the PDF, respectively. For simplicity, we assume the observation is the pitch of the query input, hence x is a scalar and the original d-dimensional Gaussian PDF reduces to the following well-known 1-dimensional version:

$$g(x,\mu,\sigma^2)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right],$$

where x is a scalar observation, μ and σ^2 are the mean and variance of the PDF, respectively. For a given observation x, the value of $g(x,\mu_j,\sigma_j^2)$ for state j is referred to as the "state probability" of observation x in state j, denoted by $p_s(x,j)$. In our implementation, we use "the sum of log probabilities" instead of "the product of probabilities" to avoid round-off errors. We also define a "transition probability" between state i and j, denoted by $p_t(i,j)$. For our left-right model, possible next moves for observation at state i are only state i (self transition) or state $i+1$ (next transition). Therefore we have the following constraints on transition probabilities:

$$\begin{cases} 0\le p_t(i,j)\le 1, \forall i,j \\ p_t(i,j)=0,\ if\ \ j<i\ \ or\ \ j>i+1 \\ p_t(n,n)=1 \\ p_t(i,i)+p_t(i,i+1)=1,\ \forall i \end{cases}$$

In the above constraints, n is the number of states and $p_t(n,n)=1$ indicates that the only possible move from the last state is self-transition to itself.

2.2 CHMM Evaluation

Given a CHMM (with corresponding parameters for state and transition probabilities) and a sequence of observations (pitch vector), we need to evaluate the model by finding the likelihood of the observations generated by the model. This is usually accomplished by dynamic programming. More specifically, for a given observed pitch vector $x=[x_1,x_2,...x_m]$ of length m, and a CHMM represented by n states, we need to construct an $m\times n$ table H, where each entry of the table is computed by the following recurrent equation:

$$H(i,j)=\ln p_s(x_i,j) + \max\begin{cases} H(i-1,j)+w\ln p_t(j,j) \\ H(i-1,j-1)+w\ln p_t(j-1,j) \end{cases},$$

where w is a weighting factor for the transition probability. In other words, $H(i,j)$ denotes the maximum cumulated log likelihood of the event that the first i observations $[x_1,x_2,...x_i]$ are generated by the first j states of the CHMM model. The initial conditions for the above recurrent equation are:

$$\begin{cases} H(1,1)=\ln p_s(x_1,1) \\ H(1,j)=-\infty\ \ for\ \ j>1 \end{cases}$$

And the overall maximum likelihood of the observations $x=[x_1,x_2,...x_m]$ generated by this model is equal to $\max_{j=1\sim n} H(m,j)$. Once the maximum likelihood is found, we can back track to find the optimum path that assign each pitch point to a state.

Therefore for a given observed pitch sequence $x=[x_1,x_2,...x_m]$, we can apply the above formula to find the likelihood of each song's model. The model with the maximum likelihood is then picked as the answer to the singing/humming query. For simplicity, we have two assumptions in this study:

1. The singing/humming query is from the beginning of the intended song. In other words, the first state is the "start state" and the first pitch point must belong to this state.
2. A user is required to keep singing for 8 seconds. Therefore any state could be the "end state" which holds the last pitch point.

The case of "match beginning" of assumption 1 can be relaxed by setting every state to be a "start state" to accommodate "match anywhere". It is also straightforward to change the recurrent equation and the initial conditions to take this case of "match anywhere" into consideration.

Figure 2 demonstrates a typical example of model evaluation, where the model is the song "10 Little Indians" and the observation is a pitch vector derived from an 8-second clip of a real user's singing.

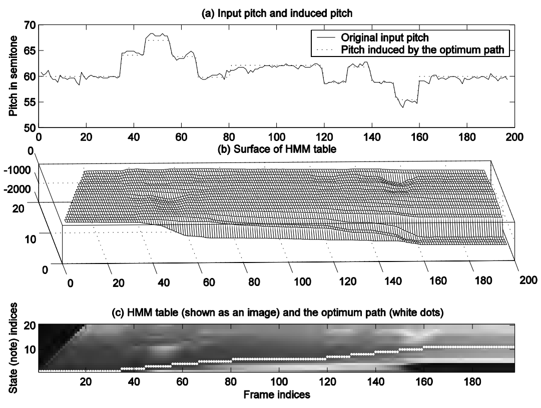

Figure 2. A typical example of model evaluation, where the model is "10 Little Indians". (a) The original observed pitch and the induced matched pitch. (b) The H table shown as a 3D surface. (c) The H table (shown as an image) and the corresponding optimum path (shown as white dots.)

Before model evaluation, we should take key transposition into consideration since the user may not sing or hum in the same key level as that of the intended song in the database. This issue will be addressed in subsection 4.2.

Obviously the retrieval results are greatly influenced by the parameters of the model. In the next subsection, we will explain how the optimum parameters can be derived.

2.3 CHMM Training

Before we can perform model evaluation, we need to identify the optimum parameters for a CHMM. This can be achieved based on the availability of either music scores or clips of users' inputs. We shall refer to these two methods as "score-based training" and "corpus-based training".

Score-based training

Each song in the database is actually a monophonic sequence of music notes, where each note is represented by its pitch (in semitone) and duration (in second). For a given music score of this format, we can find the corresponding CHMM according to the following steps:

1. Basically, each state in the CHMM represents a music note (or several consecutive notes of the same pitch level) in the song. The μ of each state is equal to the note's pitch level in semitone.
2. The σ^2 of each state is set to 1 subjectively.
3. Let k equal to the pitch points within this state. Since only the last pitch point of this state will transit to the next state, therefore the transition probability $p_t(i,i+1)$ is equal to $\frac{1}{k}$. Hence $p_t(i,i)$ is set to $1-p_t(i,i+1)$. For the last note, we have $p_t(n,n)=1$ and $p_t(n,n+1)=0$. For instance, in our implementation, sampling rate = 8000 Hz, frame size = 256, overlap = 0, then for a state representing a note of duration 0.5 second, the corresponding value of k is

$$0.5\times\frac{sampling\ rate}{frame\ size-overlap}=0.5\times\frac{8000}{256-0}\approx 16$$

In our system, the actual music scores are in MIDI format and they could be polyphonic. However, for the purpose of singing/humming retrieval, we only consider the melody or vocal track of a MIDI file for constructing the corresponding CHMM.

The above procedure for "score-based training" is quite intuitive. Only the value of σ^2 is set in a subjective manner; all the other parameters are set according to the song's characteristics. However, it does not reflect the characteristics of recordings from users' singing/humming. Therefore we have developed "corpus-based training" that can further tune the model to achieve better performance.

Corpus-based training

In order to proceed with corpus-based training of a model, we assume there is at least one recording of the corresponding song. The procedure of "corpus-based training" for a given model involves the following re-estimation procedures:

1. Use the result of "score-based training" as the initial parameters of the CHMM.
2. For each recording of the song, perform model evaluation and keep the optimum alignment path. Also compute the overall likelihood as the sum of the maximum log probabilities of all recordings.
3. For each state j in the model, collect all pitch points (from all recordings) corresponding to this state based on the optimum alignment path in the previous step. If we concatenate these pitch points into a vector $\lfloor y_1, y_2, ... y_q \rfloor$ (here we drop off the subscript j for simplicity), then the optimum parameters can be derived based on the principle of maximum likelihood estimate, as follows:

$$\mu_j=\left(\sum_{k=1}^{q} y_k\right)/q,$$

$$\sigma_j^2=\left[\sum_{k=1}^{q}(y_k-\mu_j)(y_k-\mu_j)^T\right]/q,$$

$$p_t(j,j+1)=\frac{r}{\sum_{k=1}^{r} d_k},$$

$$p_t(j,j) = 1 - p_s(j, j+1),$$

where r is the number of recordings for this model, and d_k the number of pitch points of recording k that belongs to state j.

4. Repeat step 2 to 3 until the overall likelihood does not increase.

It should be noted that the above procedure is repeated to find the optimum parameters of each CHMM of a song. Moreover, the above procedure is based on EM (Expectation Maximization) and the overall likelihood is guaranteed to be monotonically increasing until converged after several iterations.

3 OTHER APPROACHES TO MRAI

As we shall explain later, CHMM alone cannot guarantee good performance. Our error analysis indicates that we need to use other approaches jointly with CHMM to achieve better performance. Hence this section will introduce several approaches to MRAI, including LS (linear scaling) and DTW (dynamic time warping).

3.1 Linear scaling

LS (linear scaling) is one of the most simple and straightforward ways for comparing the input pitch with the song database. The basic idea is to compress or stretch the input pitch in a linear manner in order to match those in the song database. The following figure demonstrates the use of LS.

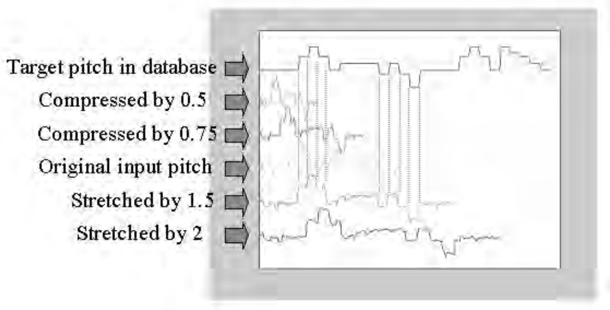

Figure 3. A typical example of linear scaling, where the input pitch vector is linearly scaled 5 times and the best match occurs when the input pitch is stretched by 1.5 of its original length.

3.2 Dynamic time warping

DTW (dynamic time warping) is one of the most effective methods for frame-based MRAI, as reported extensively in the literature. However, due to the space limit, we do not give technical details of DTW. Interested readers can refer to the corresponding references (Jang et al., 2001; Jang & Lee, 2001; Jang et al., 2004; Zhu & Shasha, 2003; Hu & Dannenberg, 2002).

4 OTHER MRAI RELATED ISSUES

In the previous two sections, we have introduced 3 different approaches to compare the input pitch vector with those in the song database, including the newly proposed CHMM, and previously proposed LS and DTW. In this section, we shall discuss the specific ways we tackle several MRAI-related issues in our implementation, including pitch tracking, key transposition, and rest handling.

4.1 Pitch tracking

In general, pitch tracking in singing or humming is easier than in speech signals due to the fact that the pitch variations in singing/humming are usually not as drastic. For simplicity, in this paper we should adopt the human-labelled pitch (which is a part of the singing corpus SQC) as the input to our melody recognition engine. By using the presumably correct pitch, we can concentrate on the error analysis of the proposed recognition approach, instead of worrying about the pitch tracking error.

4.2 Key transposition

To deal with key transposition, most note-based approaches apply the difference operator before invoking the comparison procedure. However, for frame-based approach proposed in this paper, the difference operator amplifies noises and deteriorates the performance. Hence we adopt a heuristic binary-search-like method (Jang & Lee, 2001) to shift the entire input pitch vector to a suitable position that can generate the minimum DTW distance.

4.3 Rest handling

General speaking, there are two ways to handle rests in both the input pitch and the pitch of the database songs:

1. Remove all the rests.
2. Replace the rests with its previous note's pitch level.

Both methods were used in our experiments and the results are discussed in the following section.

5 EXPERIMENTAL RESULTS

The section introduces the song database and the singing voice corpus used in our experiments, together with the experimental settings and the corresponding results.

5.1 Song database and singing voice corpus

We have collected a MIDI database of 38 children's songs and a corresponding singing voice corpus called SQC (Sung Query Corpus) for evaluating our MRAI approach. This corpus contains 1460 wave files (up to recordings of the year 2004) of 8 seconds each, recorded by 64 subjects (51 males and 13 females). Each subject was asked to record 20~30 songs (out of the 38 songs in the database) that he/she was most familiar with. All recordings in SQC are in Microsoft .wav format, with a sampling rate of 8000 Hz, single channel (mono), bit resolution of 8 bits. For simplicity, all recordings are from beginning of the intended songs. SQC is available from the first author's homepage at

http://www.cs.nthu.edu.tw/~jang

Please follow the link of "MIR corpora" to download the SQC corpus and corresponding documents.

In order to perform a fair evaluation, we have divided the corpus into two equal-size disjoint groups: A and B. For each song in the database, we tried to divide its recordings evenly into two groups. In our experiments, we used group A for training CHMM and group B for test, and vice versa for checking the consistency of these two groups. In the following subsections, we shall describe our experiments and the corresponding results.

5.2 CHMM and its enhancement

After detailed error analysis, we found that the optimum CHMM path can achieve the maximum cumulated log likelihood, but the path is not exactly what we wanted. For instance, a state can hold as few as a single pitch point, corresponding to a note duration of 1/32 second, which is not likely to occur in practical situations. To fix this problem, we have tried several methods and found the best one is to use LS to identify note boundaries and then use the note's Gaussian PDF compute the cumulated likelihood. Note that the Gaussian PDFs are obtained from either score-based training (denoted by CHMM1) or corpus-based training (denoted by CHMM2).

Figure 4 shows the recognition rates with respect to the computation time of each query with respect to 38 songs in the database. (Here the recognition rates are based on group B while group A was used as the training data. Moreover, the rests are removed and the weighting factor of transition probabilities is set to 1.)

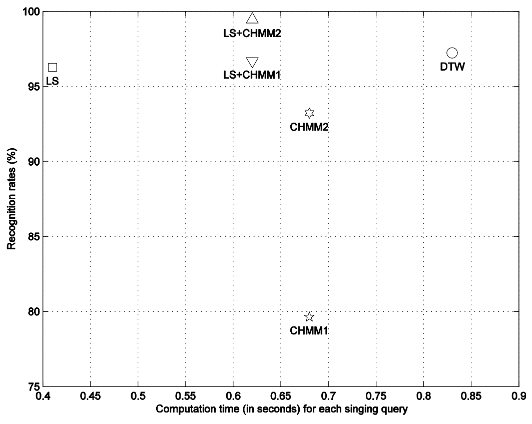

Figure 4. Top-1 recognition rates with respect to the computation time of various methods.

From the above figure, it is obvious that LS+CHMM2 has a higher recognition rate of 99.45% (with less computation time) than DTW (97.23%) and CHMM2 (93.21). LS (96.26%) also seems to be a decent competitor, however, it does not scale up well, as shown in section 5.3.

5.3 Rest handling and weight adjustment

As mentioned earlier, we can either remove the rests or extend the rests with their previous notes' pitch levels. For the weighting factor of transition probabilities, we employed a simple linear search to find its optimum value. The following bar chart shows the top-1 recognition rates of LS+HMM2 with respect to different settings.

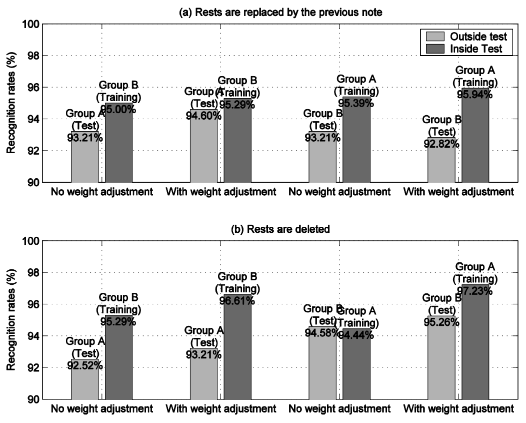

Figure 5. Top-1 recognition rates of LS+HMM2 with respect to different settings.

From the above bar chart, it seems that we can adopt either way to handle rests with only small difference in recognition rates. Moreover, the optimum weight (= 2.8) does increase the recognition rates in 3 out of 4 cases. For the experiments in the following next subsection, we used the optimum weight and the rests were all removed.

5.4 Recognition rates w.r.t. database sizes

It is essential know how these methods scale up properly or not. To explore along this direction, we increased the song database by adding other real-world songs sequentially till its size is 5000. Figure 6 plots the top-1 recognition rates with respective to the sizes of the song database. From the plot, it is obvious that "LS+CHMM2" has the highest recognition rate curve and degrades gracefully as the size of the database increases. CHMM2 also degrades gracefully, though its initial recognition rates are not so high. On the other hand, DTW, LS, and LS+CHMM1 all have a high recognition rates when the database size is small, but degrade rapidly when the database size increases. We can conclude that both LS+CHMM2 and CHMM2 scale up nicely due to its model identification via corpus-based training.

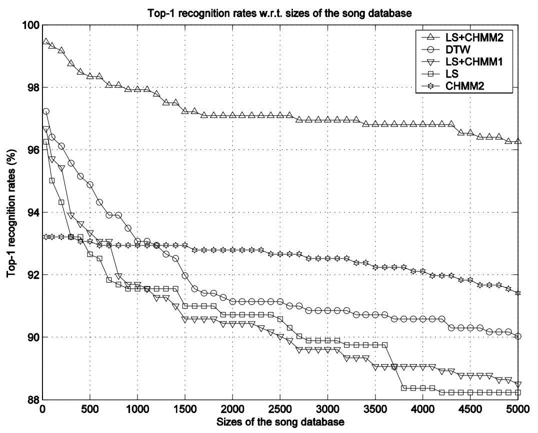

Figure 6. Top-1 recognition rates with respect to the sizes of the song database. CHMM1 is the CHMM via score-based training; CHMM2 is the CHMM via corpus-based training.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed the use of CHMM and its enhancement scheme for singing/humming query for music retrieval. The proposed LS+CHMM method can achieve a higher recognition rate and less computation time when compared with the state-of-the-art frame-based MRAI approach of DTW. Moreover, the proposed method also scales up nicely when the size of the song database increases. We have put the song database and the singing voice corpus on the web for public access.

There are a lot of potential directions for future work. Some of the immediate ones include the use of "rest" states in CHMM for appropriately representing rests, a better way of incorporating key transposition into the evaluation and training procedures of CHMM, and the use of more features (such as pitch and delta pitch) in CHMM for achieving a better recognition rate.

REFERENCES

Dannenberg, Roger B., Birmingham, William P., Tzanetakis, George, Meek, Colin, Hu, Ning and Pardo, Bryan, "The MUSART Testbed for Query-by-Humming Evaluation", International Symposium on Music Information Retrieval 2003, Baltimore, Maryland, USA, October 2003.

Hu, Ning and Dannenberg, Roger B.A, "Comparison of Melodic Database Retrieval Techniques Using Sung Queries", Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, Portland, Oregon, 2002.

Jang, J.-S. Roger, and Lee, Hong-Ru, "Hierarchical Filtering Method for Content-based Music Retrieval via Acoustic Input", The 9th ACM Multimedia Conference, PP. 401-410, Ottawa, Ontario, Canada, September 2001.

Jang, J.-S. Roger, Chen, Jiang-Chun, and Kao, Ming-Yang, "MIRACLE: A Music Information Retrieval System with Clustered Computing Engines", International Symposium on Music Information Retrieval 2001, Indiana University, Bloomington, Indiana, USA, October 2001.

Jang, J.-S. Roger, Lee, Hong-Ru, Chen, Jiang-Chuen, and Lin, Cheng-Yuan, "Research and Development of an MIR Engine with Multi-modal Interface for Real-world Applications", Journal of the American Society for Information Science and Technology, 2004.

Liu, Baolong, and Li, Yadong, "Linear hidden Markov model for music information retrieval based on humming", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pages 533-536, 2003.

Meek, Colin, and Birmingham, William, "Jonny Can't Sing: A Comprehensive Error Model for Sung Music Queries, " International Symposium on Music Information Retrieval 2002, Paris, France, October 2001.

Shifrin, Jonah, Pardo, Bryan, Meek, Colin, and Birmingham, Birmingham, "HMM-based musical query retrieval," Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, pages 295–300, 2002.

Shih, Hsuan-Huei, Narayanan, Shrikanth S., and Kuo, C.-C. Jay, "An HMM-based Approach to Humming Transcription", IEEE International Conference on Multimedia and Expo, vol. 1, pages 337-340, Aug. 2002.

Zhu, Yunyue and Shasha, Dennis, "Warping Indexes with Envelope Transforms for Query by Humming", Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, June 2003.