Information Retrieval

Content based Music Retrieval System

Tushar Goel & Aryan Sawhney

Problem Statement

The project requires us to implement a content based music retrieval (CBMR) system that uses Approximate Karbunen Loeve transform for noise reduction. Additionally, we have also implemented a user interface for this project.

Background of the Problem

Digital music data on the Internet are explosively growing. Therefore, applications of content-based music retrieval (CBMR) system are more and more popular. Searching music by a particular melody of a song directly is more convenient than by a name of a song for people. Moreover, according to the survey from the United Nations, the 21st century will witness even more rapid population ageing than did the century just past; therefore, it is important to develop an efficient and accurate way to retrieve the music data.

The motivation of the problem lies in the fact that developing such kind of retrieval systems helps the user in exploring a new world of music. The user can input a music clip without knowing the name of the music clip and can then retrieve the song from such kind of music retrieval systems. If the user has a noisy sample, he/she can use this project to find himself/herself the original song.

This project has also helped us in giving an insight as to how music is represented in the digital world and how to index data using AFPI trees in order to retrieve data quickly. We have extensively used librosa and saxpy libraries in python to implement this project. Librosa is used to extract features from the audio clip and saxpy is used to convert the extracted features into a SAX representation. Later the SAX representation is added to an AFPI tree.

Introduction

A CBMR method is a more effective approach for a music retrieval system than the text-based method. A CBMR system aims to retrieve and query music by acoustic features of music, while a text-based music retrieval system only takes names, lyrics, and ID3 tags of songs into consideration. A feature extraction method was used to extract the pitches and volumes of the input audio file.The data was used to build an index structure via advanced fast pattern index (AFPI) and Alignment as its searching technique.

Lu et al. proposed an extraction mechanism that regards audio data as a sequence of music notes, and then a hierarchical matching algorithm was performed. Finally, the similarity scores of each song to the query were combined with respect to the pitches and the rhythm by a linear ranking formula. This approach is accurate when an instrumental clip is given as the search key; however, the accuracy decreases when the input is the humming voice from a human.

Presently, state-of-the-art QBS-CBMR (Query by Singing - Content Based Music Retrieval) systems can achieve high accuracy under clean environments. However, under noisy environments, the performance might degrade due to the mismatch between the noisy feature and the clean-trained model. Motivated by this concern, the paper which we followed to implement this project (Robust and efficient content- based music retrieval system) extends the previous work. Noise effects are further reduced by applying Approximate Karbunen Loeve transform (AKLT) for preprocessing. The proposed robust QBS-CBMR system has four stages:

1. **Noise reduction**: Considering the real case of music retrieval, the noise in the music clips may impact the results. Therefore, we use AKLT as preprocess to reduce the influence of the noise for all music clips.
2. **Feature extraction**: The input music clip is first converted into a 39-dimensional Mel- frequency cepstral coefficients(MFCCs) and 12-dimensional Chroma. For each music clip,there are totally 51 dimensional features. Second, each dimension of features is transformed into symbolic sequences using the adapted symbolic aggregate approximation (adapted SAX) method, which is proposed in this work. These symbolic sequences are also called the SAX representation.

3. **The AFPI tree structure**: Following feature extraction stage, the input music is transformed into 51 symbolic sequences with respect to 51 features. In the proposed QBS-CBMR system, symbolic sequences are regarded as a search key. Finally, these symbolic sequences are stored by a tree structure called the AFPI tree due to high efficiency for the retrieval task.
4. **Score calculation**: The results of the music retrieval task are determined by the scores. After the two stages mentioned above, music clips are transformed into 51 AFPI trees. A partial score is calculated for each AFPI tree first. The final score is then obtained by the weighted summation of all partial scores, where the weighting of each partial score is determined by its entropy [12]. The higher scores denote the higher similarity between the query music clip and the songs in the database.

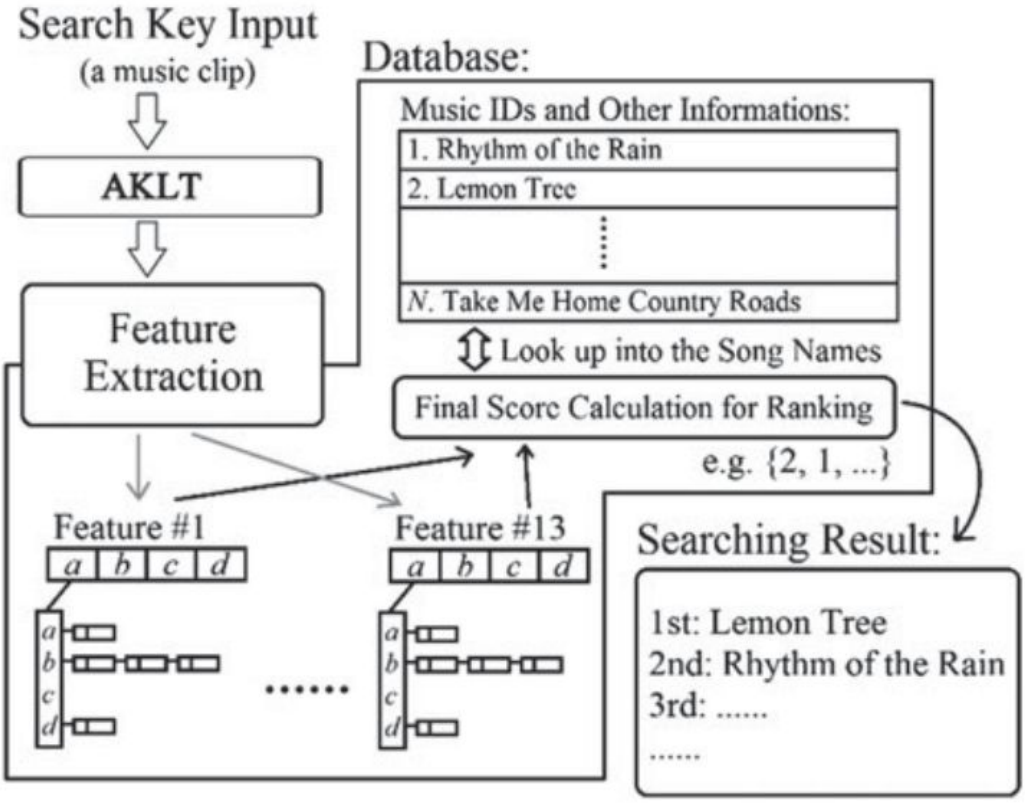

Fig. 1. The main structure of t he proposed music retrieval system.

Literature Survey

A. Music Content Representation

The MFCCs were first proposed by Davis and Mermelstein in 1980. The MFCCs are non-parametric representations of the audio signals and are used to model the human auditory perception system. Therefore, MFCCs are useful for audio recognition. This method had made important contributions in music retrieval to date. Tao et al. developed a QBS system by using the MFCCs matrix. For improved system efficiency, a two-stage clustering scheme was used to reorganize the database. On the other hand, the Chroma feature proposed by Shepard has been applied in studies of music retrieval with great effectiveness. Xiong et al. proposed a music retrieval system that used Chroma feature and notes detection technology. The main concept of this system is to extract a music fingerprint from the Chroma feature.

Sumi et al. proposed a symbol-based retrieval system that uses Chroma feature and pitch features to build queries. Moreover, to make the system with high precision, conditional random fields has been used to enhance features.

Chroma features can work well when queries and reference data are played from different music scores. It has been found that Chroma features can identify songs in different versions. Hence, we can use Chroma features to identify all kinds of songs, even cover versions . This research extends our previous work. Compared with, a new feature vector containing 39-MFCCs features and 12-Chroma features are extracted.

B. Noise Reduction

The actual application must eliminate environmental noise. Otherwise, the accuracy of the music retrieval results decreases. Shen et al. proposed a two-layer structure Hybrid Singer Identifier, including a preprocessing module and a singer modeling module. In the preprocessing module, the given music clip is separated into vocal and non-vocal segments. After the audio features are extracted, vocal features are

fed into Vocal Timbre and Vocal Pitch models, and non-vocal features are fed into Instrument and Genre models. It had been proven that the work of is robust against different kinds of audio noises. However, the noise is not removed and so the performance will be still affected by noise.

Mittal and Phamdo proposed a Karhunen Loeve transform (KLT)-based approach for speech enhancement. The basic principle is to decompose the vector space of the noisy speech into two subspaces, one is speech-plus noise subspace and the other is a noise subspace. The signal is enhanced by removing the noise subspace from the speech-plus-noise subspace. The KLT can perform the decomposition of noisy speech. Since the computational complexity of KLT is very high, the proposed system uses AKLT with wavelet packet expansion to process the noise reduction of input music clips.

System Architecture Overview

The feature extraction stage mainly contains two steps: (1) transform music files into 39 of the MFCCs features and 12 of the Chroma features [10]; (2) convert each dimension of MFCCs into a symbolic sequence by the piecewise aggregate approximation (PAA) method and the adapted SAX.

After feature extraction, each of the 51 symbolic sequences is then stored using a tree structure called the AFPI tree. Next, the 51 AFPI trees are used to generate a final score to evaluate the similarity between the query music clip and the songs in the database. Two components stored in the database for each song are:

1. 51 AFPI tree structures.
2. Music IDs and other information.

The searching process only accesses these components instead of the original audio files, so that the proposed music retrieval system is portable. In the proposed implementation, two music retrieval related operations are performed: adding a complete music file into the database (the ADD operation), and searching from the database with a music clip file (the SEARCH operation). Both operations run the music retrieving process and access the tree

structures in the database. The only two differences between ADD and SEARCH are: (1) ADD builds the structure, while SEARCH searches the structure; (2) SEARCH analyzes the result from the database structures, while ADD does not.

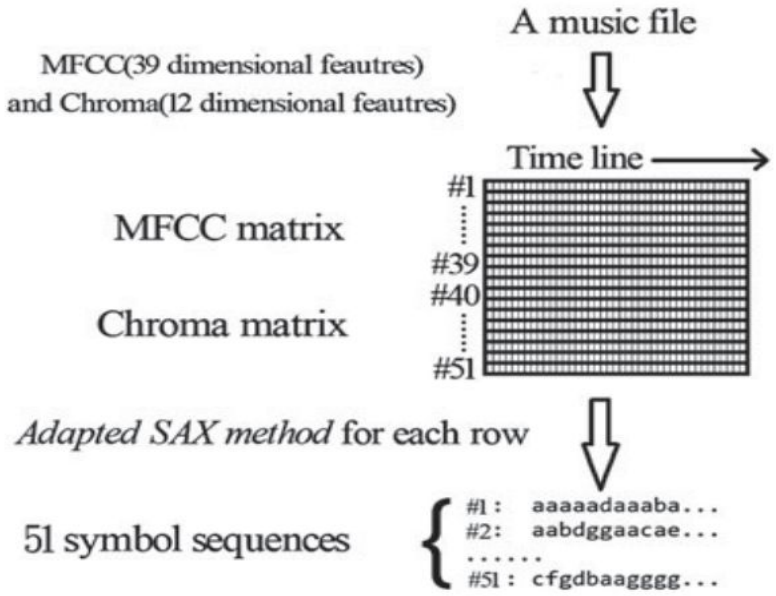

Fig. 2. A diagram for the *music retrieving process* in our system.

Evaluation Strategy

We have adopted a new method to evaluate the performance of our Information Retrieval system. We check whether our searched music clip is same as the top three music clips retrieved, if it is, then we assign it a score of one, otherwise we assign it a score of 0. Using this approach we have found the accuracy. The accuracy of the information retrieval system came out to be equal to $Accuracy = (\sum_{i=0}^{N} R(i))/N$

This accuracy is far more higher than the accuracy mentioned in our model paper (50%), we think this is because of new and improved versions of the libraries we have used to implement this project.

Research Gap

We have tried to implement the paper to the best of our capabilities but there is still scope for improvement that can increase functionality and performance even more. Currently we have implemented the project by converting the input music clip to a SAX representation, the research paper advises to us a ASAX representation. Doing so will help in improving the speed of the retrieval system.

Another research gap in our project is because of the limited resources we had while we were carrying out the project. Because of limited processor speed, we were unable to run our code on the full dataset of 500 songs. We however managed to run the code on 100 songs and find the accuracy for that.We our therefore unsure whether the accuracy will increase or decrease on increasing the size of the dataset.

Conclusions

The robust QBS music retrieval system proposed in this study first converts the 51-dimensional features, which include MFCCs and Chroma features, into symbolic sequences by applying adapted SAX methods. The symbolic sequence is then used to construct theAFPI tree. Finally, the entropy-weighting mechanism is proposed to determine the final ranking. Noise effects are further reduced by applying AKLT preprocessing. The experiment results show that the proposed QBS-CBMR system outperforms the baseline system. Future studies will optimize the parameters of the proposed method such as the length of symbolic sequence and the dimension of the PAA representation. Moreover, a large database can be used to demonstrate the efficiency of the system.