SUMMs for Audio Music Similarity & Retrieval

Dongying Zhang, Deshun Yang, Xiaoou Chen
Institute of Computer Science & Technology of Peking University
{zhangdongying, yangdeshun, chenxiaoou}@icst.pku.edu.cn

Abstract

A Markov chain based classification model is employed in out program, called SUMMs (Spectrogram Unit Markov Models system). First, we got a set of cluster models by doing cluster analysis on a corpus of spectrogram feature vectors extracted from a large collection of unlabeled songs, and build a codebook from these models. Then, for labeled songs, we extract their feature vectors and map the vectors to appropriate codes according to the codebook, thus converting songs into code sequences. Later we use code sequences of songs to train Markov chains for each song. The models are used to evaluate distances between songs.

1. Datasets

There are two datasets need to be specified.

The first dataset, called CBDS, is used to generate the codebook. The audio tracks of this dataset are gathered from the Internet, hopefully covering all the main genres and moods of western music.

The second dataset, should we call it ACDS, is provided by MIREX, which is used to evaluate the algorithms submitted by participants. The meta-data of the tracks in ACDS are known, but we don't know the details about it.

2. Procedures

There are mainly five steps: 1) Computing of spectrogram unit features of a collection of unlabeled audio tracks; 2) Clustering the corpus of feature vectors and generate the codebook; 3) Extracting the feature vectors of the labeled audio tracks and converting them into code sequences; 4) Training Models; 5) Querying.

The first two steps are applied to dataset CBDS. Steps 3-5 are prepared for dataset ACDS. The program we submit bring with it a codebook we've already generated upon CBDS, only do the jobs of steps 3-5.

2.1 Computation of the spectrogram unit features of audio tracks

First we apply CQ-transformation (CQT) [1] on each frame (about 20ms) of the audio tracks of CBDS. A CQ spectrum has 84 frequency bands, and the central frequencies of the bands are from 55Hz (A1) to 6644.9Hz (Ab8), each is being the central frequency of a semi-tone. So, 84 bands cover 7 continuous octaves. We represent each CQ spectrum by a 84-dimensional vector, with each component corresponding to a band (semi-tone).

Second, we divide the components of every spectrum vector into 7 octaves so that each octave has a 12-dimensional vector. For each octave, we concatenate the 12-dimensional vectors of every six consecutive frames to form a single vector, in other words we apply a six frames long window on each octave and gain a sequence of 72-dimensional vectors for each octave. These vectors we gathered are called spectrogram unit vectors.

2.2 Clustering spectrogram unit vectors and generate the codebook

We group spectrogram unit vectors by octaves, then we apply K-means clustering (with a fix number of clusters) on the corpus of vectors of each octave respectively. For each octave, we gain K centroids. Then we label all those centroids by giving them numeric IDs, and that's the codebook.

2.3 Mapping audio tracks into code sequences

For each audio track from ACDS, we extract spectrogram units in the same way as described in 2.1. For each spectrogram unit we search the codebook for the nearest centroid and give its ID to the unit. Thus we transform each track into 7 code sequences, each belonging to an octave.

2.4 Training Markov chains

We use Markov chain [2] to model those code sequences. For each track, we train 7 Markov chains on the code sequences of the track. Each of the 7 Markov chains corresponds to a certain octave, and is therefore trained on that octave's code sequence. These 7 Markov chains are combined into an integrated model by adding their output probabilities as the combined model's output probability.

2.5 Querying

We extract the code sequences and train Markov chains for each song from query list. For sequences of query song S_q and sequences of one of the song from ACDS S_c, with corresponding models called M_q and M_c, the distance is computed as:

$$D(S_q, S_c) = \frac{[M_q(S_q) - M_q(S_c)] + [M_c(S_c) - M_c(S_q)]}{2 * 7}$$

3. Usage

Please read README file in the submitted package to get more instructions.

References

[1] Judith C. Brown, “Calculation of a Constants Q spectral transform”, Journal of the Acoustical Society of America (1991)

[2] LR Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition”, Proceedings of the IEEE (1989)