Popular Music Retrieval by Independent Component Analysis

Yazhong Feng Yueting Zhuang Yunhe Pan
Department of Computer Science, Zhejiang University
Hangzhou 310027, China
86-571-87951853
fengyz_zju@263.net yzhuang@cs.zju.edu.cn panyh@sun.zju.edu.cn

1. INTRODUCTION

Digital music download activity is becoming the dominant traffic stream on Internet; research on content-based music retrieval tool is increasing. A rich range of researchers are contributing to content-based music retrieval systems, most of the systems deal with MIDI music and use melody contour to represent music and string matching strategies to retrieval music.

2. OUR APPROACH

In published literature, usually, acoustic input singing or humming is pitch-tracked and segmented into notes or converted into three or five level melody contour, melody contour is also extracted from music in database, our approach (Figure 1) does not try to segment notes at all, we employ statistic model to extract singing from popular music, user's input singing and extracted singing are converted to self-similarity sequence, which is a curve in 2D space, music retrieval is equivalent to comparison of these curves. Indices on music database are the weights of recurrent neural network; similarity of query key with music in database is represented by their correlation degree.

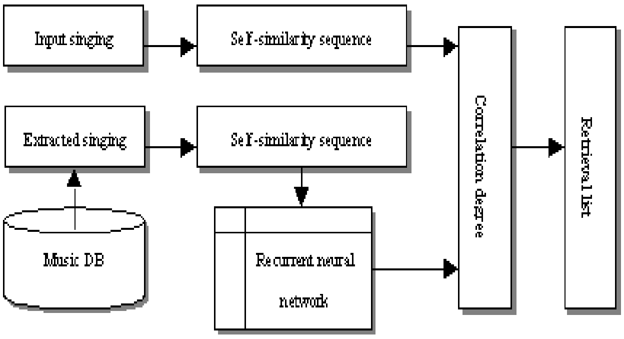

Figure 1. The diagram of our approach.

3. EXTRACTING SINGING FROM RAW AUDIO MUSIC

Independent Component Analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. The application of ICA to singing extraction is straightforward and the result is so good that it is worthy of doing more research on this direction.

3.1 Independent Component Analysis

Assume that N linear mixtures $\mathbf{x}=[x_1, x_2, \ldots, x_N]^T$ of M independent components $\mathbf{s}=[s_1, s_2, \ldots, s_M]^T$ are observed

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \qquad \textbf{Eq. 1}$$

Where $\mathbf{A}$ is a full rank $N \times M$ scalar matrix. In the ICA model, assume that each mixture x_j as well as each independent component s_k is a random variable instead of a time signal. Without loss of generality, assume that both the mixture variables and the independent components have zero mean. If the multivariate probability density function (pdf) of $\mathbf{s}$ can be written as the product of the marginal independent distributions,

$$p(\mathbf{s}) = \prod_{i=1}^{M} p_i(s_i) \qquad \textbf{Eq. 2}$$

The components of $\mathbf{s}$ are such that at most one source is normally distributed, and then it is possible to extract the sources from the mixtures. This statistical model is called independent component analysis (ICA).

3.2 Singing Extraction from stereo Popular Songs

The number of independent components is equal to that of observed variables in classic ICA, that is $M = N$, FastICA [2] can be used to perform independent component analysis. Application of ICA to singing extraction from stereo popular music is straightforward; just regard two channels of signal as observed values, singing and accompaniment as two resources.

Experiments show that FastICA performs very well in extracting singing from popular music except that drum blurs the singing in some cases.

3.3 Singing Extraction from Single Channel Popular Song

To extract two sources, singing and accompaniment from single channel recording $\mathbf{Y}$, we assume that

$$\mathbf{Y} = \mathbf{Y}_1 + \mathbf{Y}_2, \qquad \textbf{Eq. 3}$$

$$\mathbf{Y}_i = \lambda_i \mathbf{x}_i, \qquad \textbf{Eq. 4}$$

where $\mathbf{Y}_i = \{y_i(t) | t \in [1, T]\}$. It is forced that

$$\lambda_1 + \lambda_2 = 1, \qquad \textbf{Eq. 5}$$

we use an exponential power density for resource $\mathbf{s}$, which is zero mean, e.g.

$$p(\mathbf{s}) \propto \exp(-|\ \mathbf{s}\ |^q), \qquad \textbf{Eq. 6}$$

at every time point $t \in [1, T - N + 1]$ a segment $y_1(t)$ of contiguous N samples is extracted from $\mathbf{Y}_i$. Then independent source can be inferred as $s_1(t) = \frac{1}{\lambda_1} \mathbf{W}_1 y_1(t)$. Figure 2 shows a segment of single channel music and its components, the singing and accompaniment.

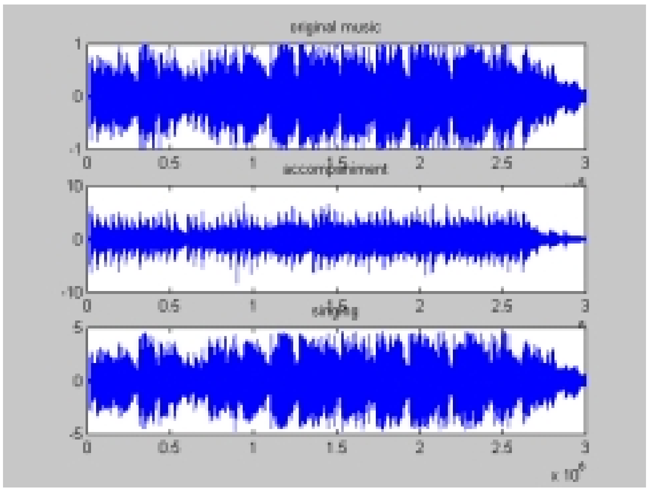

Figure 2. Single channel original music is separated to accompaniment and singing.

4. SELF-SIMILARITY SEQUENCE

Music is self-similar, lay people tend to singing with their own style, they introduce some errors such as, tempo variation, insertion or deletion of notes, but they also tend to keep the same error. These evidences support our using of self-similar sequence to represent input singing and entities in music database in our music retrieval system.

4.1 MFCCs as Features

MFCCs has been used to model music or audio, some audio retrieval system based on a cepstral representation of sounds, because that the use of Mel scale for modeling music is at least not harmful in speech/music discrimination, we also use MFCCs as the feature to form the self-similarity sequence.

4.2 Self-similarity of Audio

[1] represents acoustic similarity between any tow instants of an audio recording in a 2D representation, similarity matrix, we borrow this idea to form the self-similarity sequence, define

$$s(i, j) = diff(v_i, v_j) \qquad \textbf{Eq. 7}$$

as the element of similarity matrix $\mathbf{S}$ of an audio segment, where v_i is the feature vector of i^{th} frame, $i, j \in [1, N]$, there are N frames in this segment, define the self-similarity sequence of the same segment of audio as

$$ss(i) = \frac{\sum_{j=i}^{N} diag(S, j-1)}{N+1-i}, i \in [1, N]. \qquad \textbf{Eq. 8}$$

Self-similarity sequence is a fault-tolerant representation of music and input singing because of its not using the exact acoustic input or music information but retain their latent structures. In our system, Self-similarity sequence is used as the index of music database.

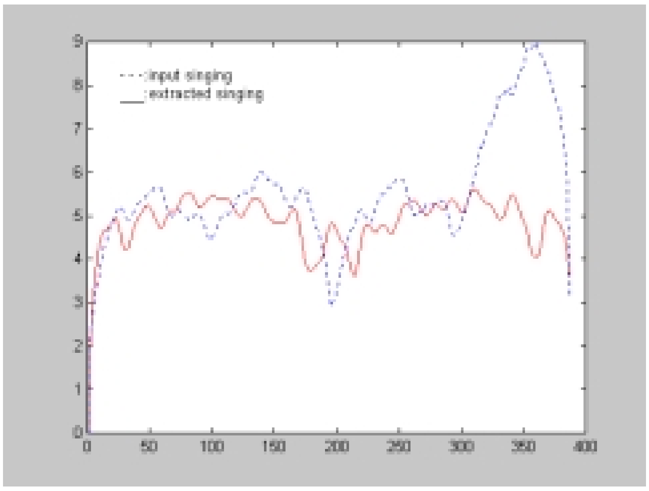

Figure 3. Self-similarity sequences: red solid curve is the extracted singing of 15 seconds segment in song "Happy birth day to you.", blue dash curve is input singing.

5. INDEXING ON MUSIC DATABASE

After singing is extracted from music in database and being converted into self-similarity sequence, recurrent neural network (RNN) is employed to remember this sequence, for each piece of music, we train a corresponding RNN. It is obvious that index size is linear to the size of music database. We do not know in previous which part of a piece of music users will singing, so the system must be robust enough for users to singing any part of the song. Recurrent neural network is of strong ability in time series prediction, the node size of input layer, output layer, hidden layer and context layer is 1, 1, 10, 10, respectively in our system, the weights between different layers store what information the network remembers.

When feeding self-similarity sequence of input singing to RNNs, we obtain a corresponding sequence from output layer, calculate the correlation degree [3] of the input and output, the bigger the correlation degree is, the more similar the input is with the music represented by this RNN.

6. EXPERIMENT RESULT

Our test database is composed of 120 pieces of raw audio popular music, our approach achieves the successful rate of top 1 and top3 is 79% and 86%, respectively. The database is rather small, but is enough to test our idea. The inaccurate retrieval results may result from bad singing extraction because of too many drum sound and improperly selected features used to calculate self-similarity sequence. Further research should be done on more accurate extraction algorithms; performance evaluation of ICA should be paid attention. Self-similarity is an interesting character of music, but which feature is more appropriate for its calculation is open for discussion.

7. REFERENCES

[1] Foote, J. Visualizing Music and Audio using Self-Similarity. In Proceedings of ACM on Multimedia, 1999.

[2] Hyvärinen, A. and Oja, E. Independent Component Analysis: Algorithms and applications. Neural Networks, 13(4-5): 411-430, 2000.

[3] Feng, Y.Z., Zhuang, Y.T. and Pan, Y.H. Query Similar Music by Correlation Degree. In Proceedings of IEEE PCM, 2001, pp.885-890.