SUBMISSION TO MIREX AMS TASK 2011 – SPARSE CODING SIMILARITY LEARNING METHOD

Yin-Tzu Lin
CMLab, GINM
National Taiwan University
known@cmlab.csie.ntu.edu.tw

Wen-Huang Cheng
MCLab, CITI
Academia Sinica
whcheng@citi.sinica.edu.tw

Ja-Ling Wu
CMLab, GINM
National Taiwan University
wjl@cmlab.csie.ntu.edu.tw

ABSTRACT

This paper describes our submissions to the MIREX 2011 audio music similarity and retrieval task. The proposed method is based on a machine learning technique – sparse coding (SC). The music similarity is not directly obtained from computed distance measures on audio contents, instead, we predict a higher level similarity scores to match the listener's subjective perceptions based on these distance measures and our pre-trained models. At the training stage, we will record the mapping of computed distance measures and the associated high level similarity scores which is estimated from human (expert) tags. Using the sparse coding techniques, this mapping is recorded using two jointly learned dictionaries that respectively store the representative computed distance and the high level similarity score patterns. Both the computed distance measures and the similarity scores can be represented as a sparse linear combination of the elements in the corresponding dictionary. At the testing stage, we will find the ratios that each element in the dictionary contributes to the newly computed distance measures, and then use this ratio to predict the corresponding similarity scores.

1. ALGORITHM OVERVIEW

We illustrate the flowchart of our algorithm in Figure 1. At the training stage, we first take the expert labelled tags of the training audio to estimate multifaceted similarity scores, such as, general-similarity, style-similarity, mood-similarity, etc. Then, for each song, we extract a set of computable acoustic features. Using these features, we compute multiple distance measures for each song-pair. Next, we jointly learn two dictionaries for the representative distance measure patterns and the associated similarity score patterns. Given any two testing songs in the testing stage, we will compute its distance measures to estimate the similarity scores.

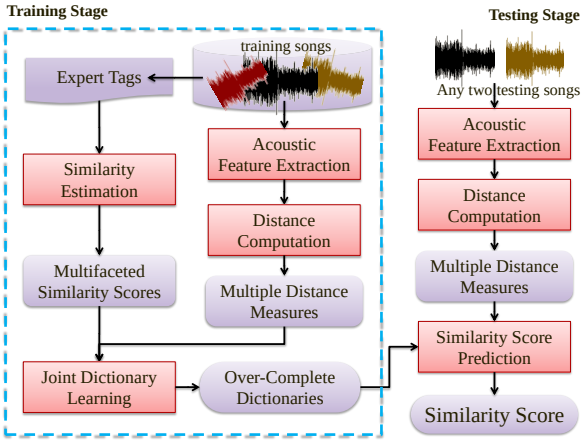

Figure 1. Algorithm Overview

2. DATASETS AND SIMILARITY SCORE ESTIMATION

We use the CAL500 dataset [4] as our training data. The dataset contains 500 songs labelled with 1700 human tags. The tags include 6 types of music properties: "emotion", "genre", "instrument", "song characteristics", "usage" and "vocal type". Each song was labelled by 2 to 4 people. The authors of the CAL500 dataset have combined all the annotations of each song into a single annotation vector by observing the level of agreement over all annotators. The single annotation vector of a song represents the degree of each tag's belonging to that song.

To estimate the multifaceted similarity scores, we calculate the scores for each tag type. We adopt the definition from psychology [5] that songs with more common attributes are more similar to each other. As a result, the similarity score y of type q between two music pieces is calculated as follows,

$$y_q = \frac{\mathbf{t_{iq}} \cdot \mathbf{t_{jq}}}{|\mathbf{t_{iq}}|\,|\mathbf{t_{jq}}|}, q = 0, 1, \ldots, 6, \quad (1)$$

where $\mathbf{t_{iq}}$ and $\mathbf{t_{jq}}$ are the annotation vectors of type q on two songs i and j, respectively. Let $q = 0$ represents the tags of all types, i.e. the general similarity.

3. ACOUSTIC FEATURES AND DISTANCE MEASURES

We used two acoustic features: the normalized MFCC codeword histograms and the onset interval histograms. For the MFCC codeword histogram feature, the MFCCs are extracted using the MIR toolbox given in [2]. We then follow McFee *et al.* [3]'s steps to extract the codeword histograms where we use a 1000-word codebook.

For the onset interval histogram feature, we extract onsets via the Beatroot 0.5.8 toobox [1]. Then, we compute the BPM (beat per minute) value for each onset interval, i.e. $BPM = 60/L$, where L is the length of an onset interval (in second). Finally, we count the number of onset intervals that lie between 20 BPM and 1000 BPM to form a 981-bin histogram.

For our task, we choose five commonly used distance measures, including: χ^2-kernel, Euclidean distance, Manhattan distance, histogram intersection, and cosine distance. Some other commonly used distances like KL divergence, Jensen-Shannon divergence, and Bhattacharyya Distance are not adopted because the numerical range is hard be normalized to between 0 and 1 [1]. All the distance measures are applied on the two used acoustic features such that ten measures will be obtained.

4. LEARNING SIMILARITY VIA SPARSE CODING (SC)

The similarity learning method is inspired by Yang et al.'s Joint Dictionary Learning (JDL) [6]. Using sparse coding techniques, both computed distance measures and the associated similarity scores can be represented as a sparse linear combination of atom vectors in the corresponding dictionaries. Let $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$ be the vectors of multiple distance measures and the multifaceted similarity scores, respectively. Then, $\mathbf{x}$ can be represented as a linear combination of the basic elements in its over-complete dictionary $\mathbf{D_x} \in \mathbb{R}^{m \times K}$ that is,

$$\mathbf{x} = \mathbf{D_x}\boldsymbol{\alpha} \ , \tag{2}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^K$ is a sparse vector with few nonzero entries ($\ll K$). We assume the relationship between $\mathbf{x}$ and $\mathbf{y}$ can be written as $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{W}$ is an unknown transform matrix [2]. Then $\mathbf{y}$ can be written as,

$$\mathbf{y} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{D_x}\boldsymbol{\alpha} = \mathbf{D_y}\boldsymbol{\alpha} \ . \tag{3}$$

We define $\mathbf{D_y} = \mathbf{W}\mathbf{D_x}$ to be the dictionary of similarity scores. Although $\mathbf{W}$ is unknown, we can still solve $\boldsymbol{\alpha}$ in (2), and then $\mathbf{y}$ is obtained by multiplying $\mathbf{D_y}$ by $\boldsymbol{\alpha}$. To solve (2), we apply the same solver as suggested in [6].

For learning the dictionary pair, we modify the JDL algorithm [6] due to the vector normalization issue. The dictionary is usually learned from a set of training songs. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$ respectively be the distance measure vectors and the similarity score vectors of the training songs. Yang *et al.* concatenated the matrix vertically, that is $\mathbf{V} = \left[\mathbf{X}^T \ Y^T\right]^T$, $\mathbf{D} = \left[\mathbf{D}_\mathbf{x}^T \ \mathbf{D}_\mathbf{y}^T\right]^T$ and regarded it as a single dictionary learning problem. However, vector normalization is needed to normalize each column to a unit vector such that the magnitude information of vectors will be will eliminated. As a result, we did not concatenate the two. Instead, we learn the two dictionaries separately but share the same sparse coefficient $\mathbf{Z}$. That is,

(i) initialize $\mathbf{D_x}$ with randomly selected K vectors from $\mathbf{X}$

(ii) fix $\mathbf{D_x}$ and update $\mathbf{Z}$ by

$$\mathbf{Z} = \arg\min_{\mathbf{Z}} \left(\|\mathbf{X} - \mathbf{D_x}\mathbf{Z}\|_2^2 + \lambda \|\mathbf{Z}\|_1\right) \tag{4}$$

(iii) fix $\mathbf{Z}$ and update $\mathbf{D_y}$ by

$$\begin{aligned} &\mathbf{D_y} = \arg\min_{\mathbf{D_y}} \|\mathbf{Y} - \mathbf{D_y}\mathbf{Z}\|_2^2 \ , \\ &\text{s.t. } \left\|\mathbf{D}_\mathbf{y}^i\right\|_2^2 \leq 1, i = 1, 2, \ldots, K \ , \end{aligned} \tag{5}$$

(iv) fix $\mathbf{D_y}$ and update $\mathbf{Z}$ by substituting $\mathbf{D_y}$ in (4) instead of $\mathbf{D_x}$.

(v) repeat steps (iii) and (iv), but substitute $\mathbf{D_x}$ instead of $\mathbf{D_y}$.

(vi) fix Z and update $\mathbf{D_y}$ and $\mathbf{D_x}$ according to (5), respectively.

(vii) iterate step (ii)$\sim$(vi) until convergence.

5. IMPLEMENTATIONS

The uploaded package only includes the testing stage. That is, we extract acoustic features, compute distance measures among testing songs, solve $\boldsymbol{\alpha}$ in (2), and then obtain $\mathbf{y}$ by multiplying $\mathbf{D_y}$ by $\boldsymbol{\alpha}$. Since the first dimension of the similarity score vector is the general similarity (computed form all types of tag), we just select this dimension. To conform to the MIREX output format (output distance), We output $1-$(predicted similarity scores).

6. REFERENCES

[1] S. Dixon: "Evaluation of the Audio Beat Tracking System BeatRoot," *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.

[2] O. Lartillot and P. Toiviainen: "MIR in Matlab (II): A Toolbox for Musical Feature Extraction from Audio," *ISMIR*, 2007.

[3] B. Mcfee, *et al.*: "Learning Similarity From Collaborative Filters," *ISMIR*, 2010.

[4] D. Turnbull, *et al.*: "Semantic Annotation and Retrieval of Music and Sound Effects," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 467–476, 2008.

[1] Because the acoustic features are normalized histograms, the numerical range of all the distance measures we used can be preserved between 0 and 1.

[2] In the traditional regression problem, $\mathbf{y}$ is a scalar and the aim is to find a proper $\mathbf{W}$.

[5] A. Tverski: “Features of Similarity,” *Psychological Review*, vol. 84, no. 2, pp. 327–352, 1977.

[6] J. Yang, *et al.*: “Image Super-Resolution via Sparse Representation,” *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.