MIREX 2017: COVER SONG IDENTIFICATION WITH METRIC LEARNING USING DISTANCE AS A FEATURE

Hoon Heo[1] **Hyunwoo J. Kim**[2] **Wan Soo Kim**[1] **Kyogu Lee**[1]
[1] Music and Audio Research Group, Seoul National University, Republic of Korea
[2] Department of Computer Sciences, University of Wisconsin–Madison, USA
cubist04@snu.ac.kr, hwkim@cs.wisc.edu, wansookim@snu.ac.kr, kglee@snu.ac.kr

ABSTRACT

The purpose of this MIREX 2017 submission is to test a recently proposed cover song identification algorithm using metric learning [2]. Most cover song identification algorithms use pairwise (dis)similarity between two songs. This makes the choice of a harmonic feature and a distance measure critical to the performance of the algorithm. In this submission, we approach this with a different perspective. We represent each song in a high-dimensional space where each dimension indicates the pairwise distance between a given song and a song from the pre-defined set of songs. Kernel methods and metric learning can be applied to this representation because all songs are transformed into the same high-dimensional space. We submit two versions of our cover song identification algorithm. Our first submission, MARG-1, uses both kernel methods and metric learning and our second submission, MARG-2, only uses kernel methods. Additionally, we submit alternative algorithms for each version named MARG-fast-1 and MARG-fast-2. These algorithms are about two times faster than the original algorithm by sacrificing acuity of SiMPle calculation.

1. INTRODUCTION

A cover song is a new version of an existing music that is modified by another musician. A cover song reuses some component such as melody and lyrics of the original music. Identifying a cover song is an important task because it can prevent copyright infringement and help music search systems.

Most cover song identification algorithm determines the relationship between two songs by comparing the harmonic progression represented by an acoustic feature such as chroma. In this submission we express a song by its relationship to a pre-defined set of songs instead of a single acoustic feature. This method represents songs with different lengths in the same high-dimensional space and makes it possible to optimize the metric to measure the distance between songs using some known similar/dissimilar song pairs.

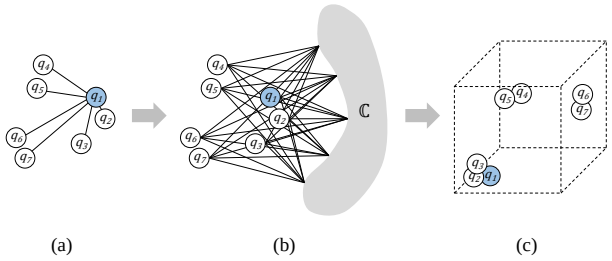

Figure 1. (a) The original distance between a query q_1 and the other songs. (b) The distance between each query and the core set $\mathbb{C}$. (c) New representation of songs in the $|\mathbb{C}|$-space.

We define three sets of songs as follows:

- Query set ($\mathbb{Q}$): A set of songs to be a query for identification. Each cover group consists of the same number of versions.
- Evaluation set ($\mathbb{E}$): A set for performance evaluation which includes the query set $\mathbb{Q}$. The remainders are "confusing songs" that are not associated with any cover groups.
- Core set ($\mathbb{C}$): An additional set of songs for embedding and training in the proposed method. It is good to select songs in the core set with diverse musical styles (i.e. genre, tempo, instruments). In this submission we combine the query set and the evaluation set and use them as the core set.

We first perform a nonlinear transformation using kernel principal component analysis (KPCA) to rearrange each song in the high-dimensional space. Next, the distance metric is learned from song pairs in the new representation and their labels. We select "core songs" with diverse musical properties and use them for both embedding and training. In summary, our approach assumes that the distance between the core set and each song can be a discriminating feature to easily group the same covers. The conceptual illustration of this new representation is shown in Figure 1.

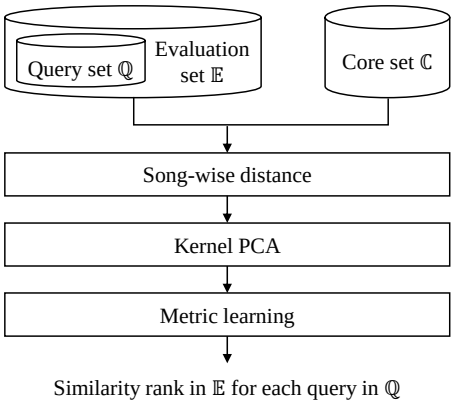

Figure 2. Block diagram of the proposed method.

2. SYSTEM OVERVIEW

Figure 2 shows a block diagram of our system.

2.1 Song-wise Distance

In our submission, we used chroma energy normalized statistics (CENS) extracted every half a second as a time-frequency representation of the raw audio and similarity matrix profile (SiMPle) to measure the song-wise distance.

Similarity matrix profile (SiMPle) efficiently evaluates similarities between songs based on subsequence similarity joins in the features [5]. For a time-frequency representation A of length m and B of length n, SiMPle identifies the nearest neighbor of each continuous subsets in A from all continuous subsets in B. Euclidean distance between the subset of A with time index i and the subset of B with time index j, $d_{i,j}$, is calculated using MASS (Mueen's Algorithm for Similarity Search). [3]. The subsequence length we used in our submission is 20. In the alternative algorithms, MARG-fast-1 and MARG-fast-2, we only calculate MASS for even indexed subsequences.

$$d_{i,j} = \text{MASS}(A[i], B[j]) \tag{1}$$

SiMPle P_i is obtained by choosing the minimum value in the distance between a subset of A and each subset of B.

$$P_i = \min(d_{i,1}, d_{i,2}, \cdots, d_{i,n}) \tag{2}$$

The overall distance between two sequences A and B is defined as the median value of SiMPle [5].

$$d_{A,B} = \text{median}(P_i) \tag{3}$$

Since SiMPle is not a symmetric measure, we symmetrized the distance matrix by $d'_{i,j} = \frac{1}{2}(d_{i,j} + d_{j,i})$, where $d_{i,j}$ is defined in Eqn (3).

2.2 Kernel PCA

After symmetrization of the distance matrix, we perform a kernel PCA. PCA seeks for eigenvectors of the covariance matrix of the data given as

$$C = \frac{1}{N}\sum_{i}^{N} x_i, x_i^T. \tag{4}$$

Similarly, kernel PCA seeks for eigen functions of the covariance function. In other words, Given a nonlinear function $\Phi(\cdot)$ to map data to feature space, the covariance matrix is calucated by

$$\bar{C} = \frac{1}{N}\sum_{i}^{N} \Phi(x_i)\Phi(x_i)^T, \tag{5}$$

where $\Phi(x)$ is centered, i.e., $\sum_i^N \Phi(x_i) = 0$. Thanks to the kernel trick, without performing the map Φ, kernel methods can be computed by kernel functions $K_{ij} = k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$. In this paper, we used the Radial basis function (Gaussian kernel). The kernel function is given by

$$\begin{aligned} k(x_i, x_j) &= \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{(d'_{ij})^2}{2\sigma^2}\right) \end{aligned} \tag{6}$$

where d'_{ij} is the symmetrized dissimilarity measure (distance) and σ is a tuning parameter. So only with the pairwise dissimilarity measure, the gram matrix for kernel PCA is obtained. The remaining procedure is similar to classical PCA. For more details, we refer the reader to [4].

Let $z_1, \cdots, z_N$ be the new representation of songs from KPCA described above. In our submission, we used 135 basis functions.

Remarks. When KPCA embeds songs in a vector space based on dissimilarity measured by SiMPle, we found that in the vector representations of some songs may have extremely large norms. So these songs tend to have large distance from most of other songs. In other words, these songs cannot be detected as a cover song. To prevent this problem, we normalized the vector representation of songs $z_1, \cdots, z_N$ by their ℓ_2 norms. All songs now are on the unit sphere and the problem can be alleviated.

2.3 Metric Learning

We adopt the Information-Theoretic Metric Learning (ITML) [1] except the regularization to make A close to the prior A_0, which is selected by users. Let $\mathbb{S}$ and $\mathbb{D}$ be a similar set and a dissimilar set, respectively. Then optimization program is given as

$$\begin{aligned} \min_A \sum_{(i,j)\in\mathbb{S}} &\max(0, \text{Tr}(AZ_{ij}Z_{ij}^T) - u) \\ &+ \sum_{(i,j)\in\mathbb{D}} \max(0, l - \text{Tr}(AZ_{ij}Z_{ij}^T)), \\ \text{s.t. } A \succeq 0 &\text{ and } A^T = A, \end{aligned} \tag{7}$$

where $Z_{ij} = z_i - z_j$ and $\text{Tr}(\cdot)$ is the trace. The input z_i for the metric learning in Eqn (7) is the new (normalized) representation of ith song obtained by KPCA. The

objective of this metric learning is to seek for an A matrix, which make the distance of dissimilar pairs larger than a threshold l (and the distance of similar pairs smaller than a threshold u). A similar pair consists of an original song and its cover song, or it can be two cover songs from an original song. The dissimilar pairs in our experiments are all possible pairs of songs except the similar pairs.

The formulation in Eqn (7) is optimized by projected stochastic subgradient descent as in Alg. 1. Since the objective function is a nonsmooth and convex function, we used the subgradient descent function. Also for the symmetric positive semidefinite constraint, the projection is added in line 12. The step size α can be updated by any reasonable method.

Algorithm 1 Projected SSGD for metric learning.

1: **for** k=1:maxiter **do**
2: DATA′ = randperm(DATA)
3: **for** (i, j) = DATA′ **do**
4: $p = 0$
5: **if** $(i, j) \in S$ **then**
6: **if** $\max(0, \mathrm{Tr}(AZ_{ij}Z_{ij}^T) - u) > 0$ **then**
7: $p = Z_{ij}Z_{ij}^T$
8: **else**
9: **if** $\max(0, l - \mathrm{Tr}(AZ_{ij}Z_{ij}^T)) > 0$ **then**
10: $p = -Z_{ij}Z_{ij}^T$
11: $A = A - \alpha p$
12: $A = \pi_{\text{psd}}(A)$
13: update α

3. DATASET

In our submission we included a separate dataset to train our method. This training dataset consists of 254 covers and each cover has two to five different versions, and have 1,175 songs in total. This dataset contains various genres of Korean pops from 1980 to 2016.

4. CONCLUSIONS

In this submission, we proposed a new cover song identification algorithm with metric learning using distance as a feature. Our experimental results with Korean pop song data show that our method improved the performance of the state-of-the-art method by more than 20%.

5. REFERENCES

[1] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[2] Hoon Heo, Hyunwoo J. Kim, Wan Soo Kim, and Kyogu Lee. Cover song identification with metric learning using distance as a feature. In *International Society for Music Information Retrieval Conference*, 2017.

[3] Abdullah Mueen, Krishnamurthy Viswanathan, Chetan Gupta, and Eamonn Keogh. The fastest similarity search algorithm for time series subsequences under euclidean distance, August 2015. Available: `http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html`.

[4] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588, 1997.

[5] Diego F Silva, Chin-Chin M Yeh, Gustavo Enrique de Almeida Prado Alves Batista, Eamonn Keogh, et al. Simple: assessing music similarity using subsequences joins. In *International Society for Music Information Retrieval Conference*, 2016.