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

Most of cover song identification algorithms are based on the pairwise (dis)similarity between two songs which are represented by harmonic features such as chroma, and therefore the choice of a distance measure and a feature has a significant impact on performance. Furthermore, since the similarity measure is query-dependent, it cannot represent an absolute distance measure. In this paper, we present a novel approach to tackle the cover song identification problem from a new perspective. We first construct a set of core songs, and represent each song in a high-dimensional space where each dimension indicates the pairwise distance between the given song and the other in the pre-defined core set. There are several advantages to this. First, using a number of reference songs in the core set, we make the most of relative distances to many other songs. Second, as all songs are transformed into the same high-dimensional space, kernel methods and metric learning are exploited for distance computation. Third, our approach does not depend on the computation method for the pairwise distance, and thus can use any existing algorithms. Experimental results confirm that the proposed approach achieved a large performance gain compared to the state-of-the-art methods.

1. INTRODUCTION

A cover song, or simply cover, is a new version of existing music that is recorded or arranged by another musician. A cover reuses the melody and lyrics of the original song, but it is performed with new singers and instruments. The other musical factors such as key, rhythm, and genre can be reinterpreted by the new artist. Since the copyright of composition and lyrics of the cover still belongs to the author of the original song, releasing a cover song without permission of the original author may cause a legal conflict. Another case is music sampling, which is the act of process that reuses a snippet of existing sound recordings. The sampling is widely considered to be a technique for creating music today, but licensing that the original creator authorizes its reuse is a legal requirement. Cover song identification is a task that aims to measure the similarity between two songs. It can be used to prevent the infringement of copyright, and also to be an objective reference in case of conflict.

For a decade, many approaches for cover song identification have been proposed. Humans generally recognize the cover through the melodic or lyric similarity, but separation of the predominant melody from a mixed music signal is still not at a reliable level, and extraction of the lyrics can be attempted only if it is clearly separated. For this reason, most of the existing algorithms use the harmonic progression represented by an acoustic feature such as chroma [6], and measure the similarity in the features to determine the distance between two songs.

Cover song identification generally consists of two main stages: feature extraction and distance calculation. In most related works, chroma or harmonic pitch class profile (HPCP) are usually chosen, as well as its variants such as CENS [9], CRP [8], and MPLPLC [2]. It is reported that the abstraction of the chroma-like feature to focus on the chord progression rather than instantaneous note changes improves the identification performance [2, 15]. In early days, the feature was synchronized with the beat to take into account the covers with different tempo [4]. However, since the error in beat tracking degrades the performance and the tempo change is usually not extreme, the hop size with a fixed length is recently preferred [14]. Besides, two-dimensional Fourier transform magnitude (2DFTM) of the chroma feature is applicable for large-scale cover song identification [1]. The 2DFTM is key-invariant and thus does not require any preprocessing for key transposition. Also, regardless of the duration of the song, its fixed size has the advantage of keeping the locality.

In respect to the distance calculation, an early approach finds the best-correlated point using cross-correlation of the beat-synchronous chroma [4]. The next popular approach is based on dynamic time warping (DTW), which can be sensitive to tempo changes even when the hop size is fixed [14]. This approach uses the overall distance after aligning over the whole region of the two given songs. On the other hand, a more recent approach called similarity matrix profile (SiMPle) yields a high similarity when many local similar regions are found [15].

The conventional approaches described above calculate

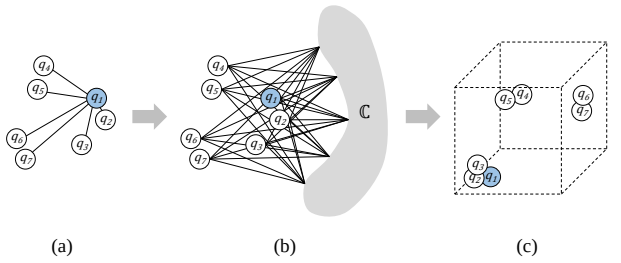

Figure 1. (a) The original distance between a query q_1 and the other songs. (b) The distance between each query and the core set $\mathbb{C}$. (c) New representation of songs in the $|\mathbb{C}|$-space.

the distance between a query and the songs to be compared, and determine that the song with the nearest distance is highly likely to be a cover. Since this process is separate from each query, the result from "another version of the same cover" cannot be taken into account. If it is possible, songs with different lengths can be represented in the same space. Furthermore, if similar/dissimilar song pairs are known, the metric to measure the song distance can be optimized, rather than using the Euclidean distance. Instead of taking the distance matrix directly to rank the similarity, 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.

The goal of this paper is to examine whether the distance metric learning can be effective to retrieve the similarity between songs. Also, this paper aims to achieve the best performance in cover song identification by applying the metric learning to the distance matrix generated by existing algorithms. Currently, MIREX hosts an annual task for cover song identification, but the dataset is not publicly available. In the later section, we report a performance comparison using our own dataset with the same specification as that of the MIREX.

The rest of this paper is organized as follows. Section 2 defines some important terms throughout this paper, and summarizes three popular algorithms for measuring the distance between songs. In section 3, we describe the technical method for better representation of songs and metric learning. After that, the experimental setup and results are presented in section 4. Finally, the conclusions of this paper are drawn in section 5.

2. DISTANCE MATRIX

The distance matrix is defined by a two-dimensional matrix that contains the pairwise distances for all possible

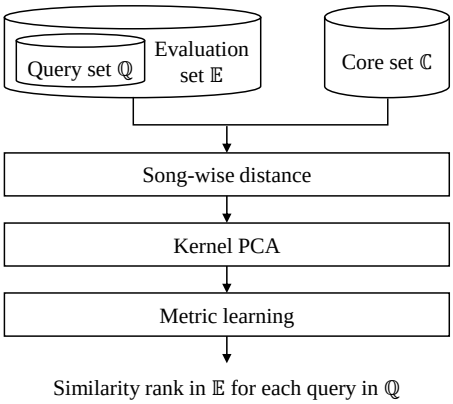

Figure 2. Block diagram of the proposed method.

combinations of two songs. The range of distance may vary depending on the algorithm, but it should be low between songs belonging to the same cover group, and should be high if they are not associated.

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).

Among these sets, $\mathbb{Q} \subset \mathbb{E}$ and $\mathbb{E} \cap \mathbb{C} = \varnothing$ should be satisfied.

The distance matrix is a square matrix calculated from all the songs in the three sets. We employed three algorithms for measuring the song-wise distance: dynamic time warping (DTW), Smith–Waterman algorithm, and similarity matrix profile (SiMPle). In the following subsections, we give a brief overview of each algorithm to construct the distance matrix.

2.1 Dynamic Time Warping

DTW performs dynamic programming to retrieve the optimal path that minimizes the warping cost. Given a sequence A of length n and a sequence B of length m, it constructs an n-by-m matrix that contains the Euclidean distance $\delta_{i,j}$ between both sequences at two time instances i and j. The cumulative distance $\gamma_{i,j}$ is the sum of the distance in the current point and the minimum cumulative distance from the three adjacent points,

$$\gamma_{i,j} = \delta_{i,j} + \min\left(\gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1}\right). \quad (1)$$

The overall distance between two sequences A and B is determined by the cumulative distance at the end of the

path,

$$d_{A,B} = \gamma_{n,m}. \tag{2}$$

To prevent unrealistic warping and reduce the number of paths to consider, DTW can be implemented with global and local constraints. The two popular global constraints are Sakoe–Chiba band [11] and Itakura parallelogram [5]. On the other hand, the local constraints allows deviations of the double or half the original tempo by using warpings $(i-1, i-1)$, $(i-2, j-1)$, and $(i-1, j-2)$ [10].

2.2 Smith–Waterman Algorithm

Similar to DTW, Smith–Waterman algorithm performs dynamic programming to find the optimal path that maximizes the similarity score between two sequences [16]. The main difference to the classic DTW is that the optimal path is produced locally. That is, it is not necessary that the path with the maximum similarity covers the whole sequence. Given a sequence A of length n and a sequence B of length m, it constructs an $(n+1)$-by-$(m+1)$ scoring matrix H. The first row and column are initialized with 0. The recursion formula to fill the rest of the scoring matrix is,

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ \max_{k \geq 1}\{H_{i-k,j} - W_k\}, \\ \max_{l \geq 1}\{H_{i,j-l} - W_l\}, \\ 0 \end{cases} \tag{3}$$

where $s(a_i, b_j)$ is the similarity score between ith element of A and jth element of B, and W_n is the penalty of a gap with length n. The overall similarity of the Smith–Waterman algorithm is defined as the maximum value on the scoring matrix.

2.3 Similarity Matrix Profile

Similarity matrix profile (SiMPle) efficiently evaluates similarities between songs based on subsequence similarity joins in the features [15]. 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), the fastest known algorithm for distance vector computation [7].

$$d_{i,j} = \text{MASS}(A[i], B[j]) \tag{4}$$

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

$$P_i = \min(d_{i,1}, d_{i,2}, \cdots, d_{i,n}) \tag{5}$$

The overall distance between two sequences A and B is defined as the median value of SiMPle [15].

$$d_{A,B} = \text{median}(P_i) \tag{6}$$

Note that SiMPle is not a symmetric distance measure, i.e., $d_{B,A} \neq d_{A,B}$.

3. DISTANCE METRIC LEARNING

Distance metric learning has been studied in machine learning literature. Classical metric learning algorithms are motivated by Mahalanobis distance given as

$$d(x_1, x_2) = \sqrt{(x_1 - x_2)^T \Sigma^{-1} (x_1 - x_2)}, \tag{7}$$

where Σ is the covariance matrix of X. The main intuition behind Mahalanobis distance is that it calculates the Euclidean distance in a linearly transformed space by R, where $R^T R = S^{-1}$. Mahalanobis distance is a convenient metric since it is scale-invariant, and it takes the correlations of data set into account. The linear transform R makes the data have the isotropic covariance as the same as the covariance of multivariate normal distribution. The goal of metric learning algorithms is to learn A, which corresponds to the precision matrix (Σ^{-1}) based on a variety of criterion.

$$d(x_1, x_2) = \sqrt{(x_1 - x_2)^T A (x_1 - x_2)}, \tag{8}$$

where A is a symmetric positive semidefinite matrix $A \succeq 0$, $A = A^T$. Training may require additional labels such as classes and similar/dissimilar pairs depending on the objective of the frameworks.

The main difficulty to apply the classical metric learning algorithms to cover song identification problems is that the songs should be represented in a vector space. One simple approach is to extract a set of fixed length features from songs, e.g., mean MFCCs, mean Chroma, and beats per minute (BPM). But these features do not capture the temporal information within a song. So, a variety of time series analysis methods has been shown to be more effective such as dynamic time warping (DTW).

Can we embed songs in a vector space preserving the temporal information? If this is possible, then distance metric learning algorithms are able to find a better distance between songs with both the temporal information and additional labels (similar/dissimilar pairs or classes). One option is kernel PCA. Fortunately, distance metric learning can be extended in the context of kernel methods as well. The kernel methods do not require the original data to be in a vector space. We can get a gram matrix (or inner product matrix) by pairwise dissimilarity measures. For embedding, other embedding algorithms can be used for instance multidimensional scaling (MDS), ISOMAP, locally linear embedding (LLE) and so on. We discuss our framework to calculate the gram matrix and embed songs in a vector space shortly.

3.1 Embedding of songs

As discussed above, we start from a pairwise dissimilarity measures. We calculate the distance matrix as described in Section 2. The gram matrix in the conventional kernel methods should be symmetric positive-semidefinite matrix. If the matrix is given as not symmetric (e.g. SiMPle), it needs to be symmetrized by $d'_{i,j} = \frac{1}{2}(d_{i,j} + d_{j,i})$, where $d_{i,j}$ is defined in Eqn (6).

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

$$C = \frac{1}{N}\sum_{i}^{N} x_i, x_i^T. \tag{9}$$

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

$$\bar{C} = \frac{1}{N}\sum_{i}^{N} \Phi(x_i)\Phi(x_i)^T, \tag{10}$$

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

$$\begin{aligned} k(x_i, x_j) &= \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \\ &= \exp\left(-\frac{(d'_{ij})^2}{2\sigma^2}\right) \end{aligned} \tag{11}$$

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 [12].

Let $z_1, \cdots, z_N$ be the new representation of songs from KPCA described above. In our experiments, the number of basis functions and the bandwidth σ in Eqn (11) were empirically selected.

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 regardless of the metric learned by A in Eqn (8), 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. The empirical performance gain is provided in Section 4.3. The normalized vector representation will be used for metric learning.

3.2 Metric Learning

We adopt the Information-Theoretic Metric Learning (ITML) [3] 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

$$\begin{aligned} \min_A \sum_{(i,j)\in\mathbb{S}} &\max(0, \mathrm{Tr}(AZ_{ij}Z_{ij}^T) - u) \\ &+ \sum_{(i,j)\in\mathbb{D}} \max(0, l - \mathrm{Tr}(AZ_{ij}Z_{ij}^T)), \\ \text{s.t. } A \succeq 0 &\text{ and } A^T = A, \end{aligned} \tag{12}$$

where $Z_{ij} = z_i - z_j$ and $\mathrm{Tr}(\cdot)$ is the trace. The input z_i for the metric learning in Eqn (12) 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 way we label the relationship between songs naturally yields highly skewed labels. For example, if two out of ten songs are the only covers, then we have one similar pair against $\binom{10}{2} - 1 = 44$ dissimilar pairs. Interestingly, it turns out that the skewness of labels does not hurt the performance of our framework. Rather, as the number of dissimilar pairs increases, the performance increases. Our experiment evidences this phenomenon, see Section 4.3.

The formulation in Eqn (12) 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) ∈ S then
6:              if max(0, Tr(A Z_ij Z_ij^T) − u) > 0 then
7:                  p = Z_ij Z_ij^T
8:          else
9:              if max(0, l − Tr(A Z_ij Z_ij^T)) > 0 then
10:                 p = −Z_ij Z_ij^T
11:         A = A − αp
12:         A = π_psd(A)
13:     update α
```

4. EVALUATION

4.1 Dataset and Metrics

We used two separate datasets to evaluation and train our method. The specification of our evaluation dataset resem-

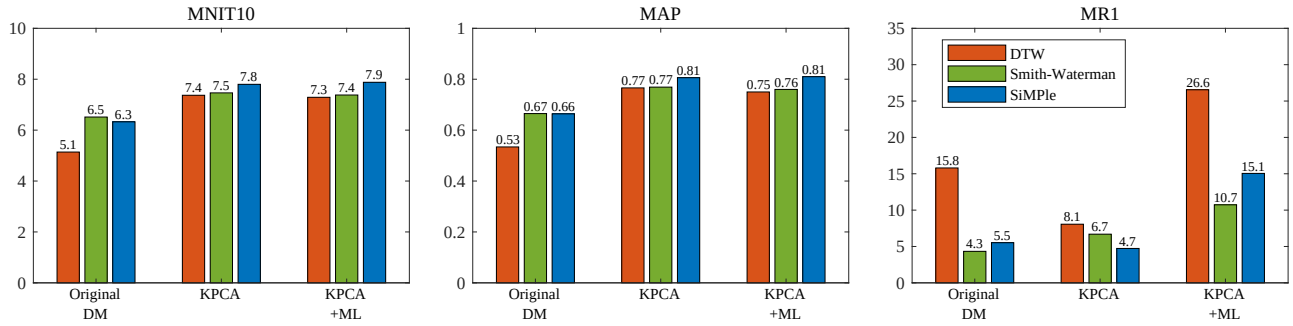

Figure 3. Improved performance by each step in the proposed method: the original distance matrix, kernel PCA, and metric learning with kernel PCA.

bles as in the MIREX cover song identification task [1] . The evaluation set $\mathbb{E}$ consists of 330 cover songs, which make the query set $\mathbb{Q}$, and 670 non-covers. There are 30 different kinds of cover songs and each has 11 cover versions. The training dataset consists of 254 covers and each cover has two to five different versions, and have 1,175 songs in total. It was used as the core set $\mathbb{C}$ in the experiments. Both datasets are disjoint, and contain various genres of Korean pops released from 1980 to 2016.

We employed four conventional metrics that have been used in the MIREX: total number of covers identified in top 10, mean number of covers identified in top 10 (MNIT10), mean average precision (MAP), and mean rank of the first correctly identified cover (MR1). In the experimental results, we skipped the first one because it is exactly the same as the second metric multiplied by $|\mathbb{Q}|$.

4.2 Experiments

Since selection of features and calculation of pairwise song distance are not our interest, the chroma energy normalized statistics (CENS) [9] was fixed as the feature vector and extracted for every half a second in all the following experiments. Also, before calculating the distance between two songs, we transposed one using the optimal transpose index (OTI) [13] so that both songs have the same key.

In the first experiment, we examined the effect of two proposed steps on identification performance: new representation transformed by the kernel PCA, and the metric learning using similar/dissimilar pairs in the core set. 135 basis functions were empirically selected, and 2435 similar pairs (for covers) and 687k dissimilar pairs (for non-covers) were used as training data for metric learning. This experiment allows reporting the maximum performance we could achieve and how each part of the proposed method contributes to the performance improvement.

The second experiment aims to verify that the metric learning converges to a higher performance as more training data are used. We tested different numbers of the training data, which are song pairs in the core set. Songs are randomly chosen with the given number of pairs in each class. Since we have much less similar pairs than dissimilar pairs, the training will be imbalanced when all possible similar pairs are used. In this experiment, we fixed the original distance measure by the SiMPle algorithm.

[1] `http://www.music-ir.org/mirex/wiki/2016:Audio_Cover_Song_Identification`

4.3 Results and Discussions

The first experimental result is shown in Figure 3. When comparing the original performance of the existing algorithms, Smith–Waterman algorithm achieved 26% higher performance than classic DTW. This is almost the same result as reported in a previous work [15]. The SiMPle algorithm, which we consider to be the state-of-the-art method, originally scored a slightly lower performance than Smith–Waterman algorithm in our experiment. However, the proposed method improved its original performance by 25% (in MNIT10), which was the largest improvement. Algorithms based on dynamic programming (DP) seem to have limitations in potential performance gain. One possible reason is that the differences in distance between similar and dissimilar pairs are not so discriminated; while the SiMPle mainly depends on local similarity joins with a fixed length of 10 seconds, DP-based algorithms may take much longer sequences into account. Meanwhile, MR1 was increased by the metric learning. This will be discussed in detail in the next paragraph.

Figure 4 shows the learning curve of the metric learning with different number of pairs. A hundred pairs for each class were not sufficient to converge. As more pairs were used for training, both MNIT10 and MAP converged to higher performance. This result was also obtained when more but imbalanced training data was used. Interestingly, the trend of MR1 increased after a certain number of iterations. This is caused by that the metric learning concentrates on the performance for a large majority of query songs, while it fails for very few queries. To support this, we first calculated the median instead of the arithmetic mean rank, and noticed that the correct cover had the highest similarity in most queries (i.e. median $= 1$) for every number of pairs and iteration. Nevertheless, since it is not suitable to show that the performance is getting improved with more iterations, the 90th percentile of rank of the first correctly identified cover ($P_{90}R1$) is shown instead in the figure.

In summary, our experiments confirm that the use

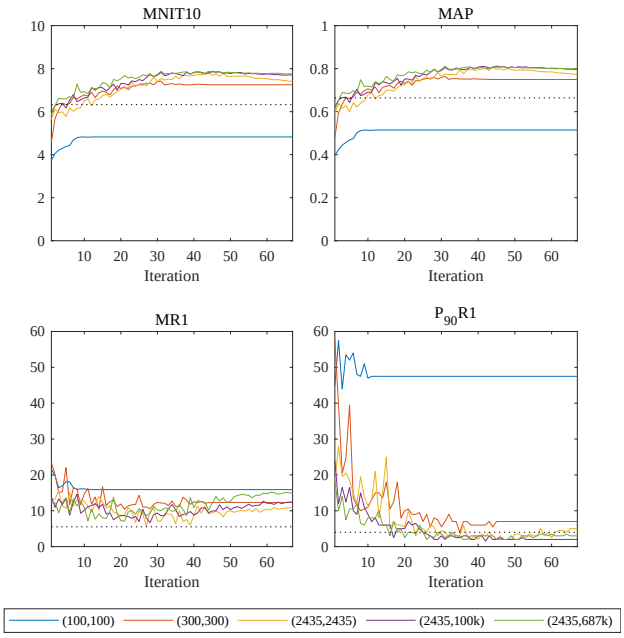

Figure 4. Learning curve of the metric learning with different number of pairs (similar, dissimilar). The black dotted line indicates each metric resulted from the original distance matrix.

of KPCA and metric learning on the SiMPle algorithm achieves the highest performance in a general situation. Although MR1 was increased by metric learning, it was explained by the second experiment showing that the trained metric failed only for a very small number of queries, while it was optimized for the most of queries. Since metric learning takes longer computation time and its performance improvement was not prominent as much as KPCA, it is possible to expect a good performance gain using empirically optimized parameters of KPCA for a fixed dataset. However, considering that scalability is an important issue in cover song identification, metric learning cannot be excluded especially for large-scale collections.

In the new representation through KPCA, each dimension represents the distance from each core song. This implies that core songs with diverse styles of music allows dimensions to be nearly orthogonal, and may yield better performance. In the metric learning, on the other hand, higher performance could be achieved with a sufficient number of similar and dissimilar pairs for training. It is not easy to satisfy both of the above conditions simultaneously, because collection of songs with various styles includes songs that are not very popular and rarely covered. Therefore, when a high recall is required (to avoid very low identification performance for very few queries), it is expected that it can be more important to have many similar pairs than various styles.

5. CONCLUSIONS

In this paper, we have presented a novel approach to improve the performance of existing algorithms for cover song identification. Our approach exploits an external set of core songs so that all the given songs are newly represented by the distance between each core song. Through the distance metric learning after embedding of songs using kernel PCA, the original performance of the state-of-the-art method was improved by more than 20%.

With different features and distance measures, the proposed method can be easily applied to similarity analysis of other tag-based data such as genre, mood, and style. We plan to further explore our approach to many other MIR tasks, and seek for proper criteria to choose the core set from large-scale collections. A sufficient number of well-organized core songs and efficient computation for metric learning will be also studied in the next step.

6. ACKNOWLEDGEMENTS

This research project was supported by Ministry of Culture, Sports and Tourism (MCST) and from Korea Copyright Commission in 2017. [Development of predictive detection technology for the search for the related works and the prevention of copyright infringement]

7. REFERENCES

[1] Thierry Bertin-Mahieux and Daniel PW Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, pages 117–120. IEEE, 2011.

[2] Ning Chen, J Stephen Downie, Haidong Xiao, Yu Zhu, and Jie Zhu. Modified perceptual linear prediction liftered cepstrum (mplplc) model for pop cover song recognition. In *International Society for Music Information Retrieval Conference*, pages 598–604, 2015.

[3] 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.

[4] Daniel PW Ellis and Graham E Poliner. Identifying cover songs' with chroma features and dynamic programming beat tracking. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1429. IEEE, 2007.

[5] Fumitada Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1):67–72, 1975.

[6] Kyogu Lee. Identifying cover songs from audio using harmonic representation. *MIREX task on Audio Cover Song Identification*, 2006.

[7] 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`.

[8] Meinard Muller and Sebastian Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.

[9] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *International Society for Music Information Retrieval Conference*, volume 2005, page 6th, 2005.

[10] Cory S Myers and Lawrence R Rabiner. A comparative study of several dynamic time-warping algorithms for connected-word recognition. *Bell System Technical Journal*, 60(7):1389–1409, 1981.

[11] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

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

[13] Joan Serra, Emilia Gómez, and Perfecto Herrera. Transposing chroma representations to a common key. In *IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, pages 45–48, 2008.

[14] Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.

[15] 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.

[16] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.