COVERNET: COVER SONG IDENTIFICATION USING CROSS-SIMILARITY MATRIX WITH CONVOLUTIONAL NEURAL NETWORK

Juheon Lee, Sungkyun Chang, Donmoon Lee, Kyogu Lee
Music and Audio Research Group & Center for Superintelligence, Seoul National University, Korea
{ juheon2, raynol, lunideal, kglee } @snu.ac.kr

ABSTRACT

This paper describes the algorithm submitted for the MIREX 2018 cover song identification task. We propose a system that identifies a cover song based on the similarity on the melodic sequence. For determining the melodic similarity of the two songs, we trained the convolutional neural network based classifier that predicts the cover probability. We also suggest ranking strategies using the relationship between the neural network outcomes of various songs. By using this strategy, we can have the robust performance with the misclassification of the neural network.

1. INTRODUCTION

The cover song identification task aims to find other recordings of the original song. It is considered as a challenging task because a covering of one song is not a fixed type conversion. The cover song may have a change in the player, tempo, or key while maintaining other conditions. In some cases, the structure, the type of instrument used, the language of the lyrics, the genre, or the complex changes may have completely different characteristics from the original one. However, it retains its identity such as unique melody pattern, so that anyone familiar with the original song can easily recognize it.

Conventional approaches tried to solve this problem by finding the similar melodic features in audio such as mel-frequency cepstrum coefficients (MFCCs), chroma-gram, mel-spectrogram, constant Q transform (CQT), or etc. For finding their sequential similarity, a similarity matrix is commonly used. It has advantages in showing the relationship between all possible subsequences. However, in reverse, it has disadvantages from the abundant data, which is the data redundancy and the difficulty in analysis. Thus, direct use of melodic features such as dynamic time warping (DTW) or SiMPle [8] have shown remarkable performance.

We propose a convolutional neural network (ConvNet) based approach to identify a cover song with as a way to analyze the similarity matrix. We assumed that even considering the data redundancy in the similarity matrix, it is still informative and the ConvNet can be useful tool for finding patterns in the similarity matrix. As shown in Figure 1, the similarity matrix of the two songs in the cover relationship has an unique image characteristics such as a diagonal shape. Therefore, we hope ConvNet, which has shown particularly promising results in the area of image pattern recognition, will be an useful tool for this task. In addition, we propose the relationship representation vector for the more robust ranking system. Since we assumed that when the network works as a mapping function to another feature space, the relationship among the songs could be a more strong indicator for cover relation than the individual output of the network.

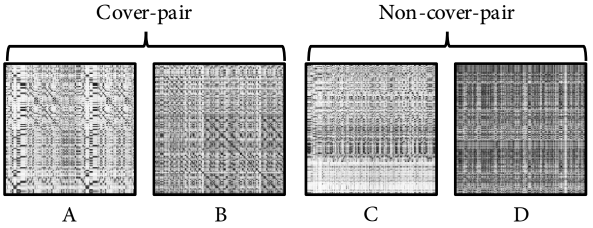

Figure 1. Cross-similarity matrix from cover-pair (A,B), and non-cover-pair (C,D).

2. PROPOSED SYSTEM

Figure 2 shows the overview of our system. Our system determines whether the songs are in a cover relation or not based on a pair and finds the cover songs based on the ranking of the entire data. First, all songs are converted to chroma feature and a cover probability is calculated using a ConvNet-based decision network for every possible pairs. When all the probabilities are calculated, the system defines a relationship representation vector for every song with calculated probabilities. Finally, the cover rank is measured using the correlation between the representation vector.

2.1 Chroma Feature Extraction

We used the first 180-second clip of songs for our experiments, since we assumed that it is enough time for containing the main melody which best represents the identity of

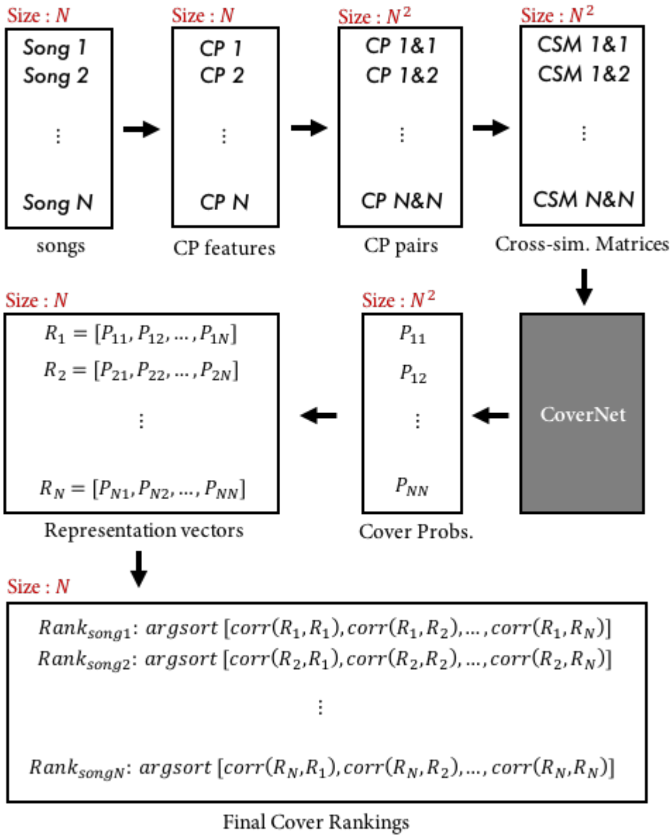
Size : N
Song 1
Song 2
Song N
songs
Size : N
CP 1
CP 2
CP N
CP features
Size : N^2
CP 1&1
CP 1&2
CP N&N
CP pairs
Size : N^2
CSM 1&1
CSM 1&2
CSM N&N
Cross-sim. Matrices
CoverNet
Size : N^2
P_{11}
P_{12}
P_{NN}
Cover Probs.
Size : N
$R_1 = [P_{11}, P_{12}, ..., P_{1N}]$
$R_2 = [P_{21}, P_{22}, ..., P_{2N}]$
$R_N = [P_{N1}, P_{N2}, ..., P_{NN}]$
Representation vectors
Size : N
$Rank_{song1}$: argsort $[corr(R_1,R_1), corr(R_1,R_2), ..., corr(R_1,R_N)]$
$Rank_{song2}$: argsort $[corr(R_2,R_1), corr(R_2,R_2), ..., corr(R_2,R_N)]$
$Rank_{songN}$: argsort $[corr(R_N,R_1), corr(R_N,R_2), ..., corr(R_N,R_N)]$
Final Cover Rankings

Figure 2. System Overview: A schematic diagram for finding cover songs of N songs.

a song. We, then, calculated a 12-semitone *Chroma-Pitch* (CP) [2] feature without overlap using the 1-second analysis window. Therefore, all songs are converted to the matrix with the shape of (180 x 12) which correspond to the number of frames and the number of chroma bin. The implementation was done using MatLab chroma toolbox [6].

2.2 Cross-Similarity Matrix

A cross-similarity matrix (CSM) is defined between a pair of songs. In this experiment, we used the CP feature of each song to calculate the CSM. Therefore, for a given song pair, we have a CSM with the shape of (180, 180). The (i, j) element is the Euclidean distance between the i th frame of the first song which is 12-dimensional chroma vector and the j th frame of the second of the song. Considering the key difference between the two songs, we normalized the key using the OTI method [7] before calculating CSM.

2.3 Decision Network

The ConvNet-based decision network (Figure 3) calculates the cover-probability for a given CSM. The input of the network is a CSM with the shape of (180 x 180) which is followed by 5 convolutional blocks. The convolutional block consists of two convolutional layers, a max pooling layer with size of (2 x 2), and a batch normalization layer [1]. The depth of filter map is fixed to 32 and kernel size is (3 x 3), except first kernel size of (5 x 5). Then, it connected to the fully-connected layer and the binary prediction layer with drop out regularization [9]. The rectified linear unit (ReLU) is used as a non-linearity function except the softmax in prediction layer.

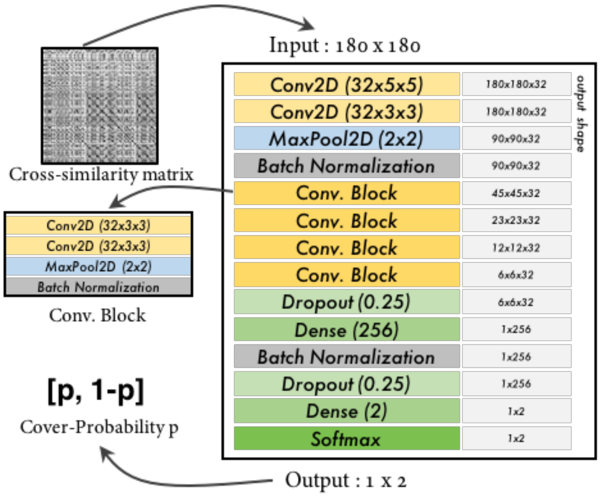

Figure 3. Architecture of CoverNet: *Conv2D (filter depth x kernel width x kernel height)*, *MaxPool2D (pool width x pool height)*, denotes a convolution layer and max pooling layer, respectively. The elements of output shape are correspond to width, height, and channel.

2.4 Ranking Strategy

In order to rank, we define the distance of cover relation using the relationship representation vector between the songs rather than direct use of the cover-probability which is the output of the decision network. By adopting the relationship representation vector, we expect more robust performance even when there are incorrect predictions made by the decision network.

The relationship representation vector ($R_i \in \mathcal{R}^{1 \times N}$) for song i defined as follows:

$$R_i = [P_{i1}, P_{i2}, P_{i3}, ..., P_{iN}]$$

where, P_{ij} is the cover-probability of song i and j calculated by decision network and N is the number of the songs in data set. The distance between the two songs is derived based on the sample Pearson correlation coefficient,

$$dist(R_i, R_j) = 1 - \frac{(R_i - \bar{R}_i) \cdot (R_i - \bar{R}_j)}{\|R_i - \bar{R}_i\|_2 \, \|R_j - \bar{R}_j\|_2}$$

where $\bar{R}$ is the mean vector of R and $\cdot$ means the inner product.

Fianlly, we estimate cover-ranking for song i ($Rank_i \in \mathcal{R}^{1 \times N}$) by sorting in ascending order:

$$Rank_i = \mathbf{argsort}(dist(R_i, R_1), ..., dist(R_i, R_N))$$

2.5 Network Training

For training decision network, we used a MIREX-like data set consisting of 1,175 Korean pop songs. There are 2,113 cover-pairs and 687,612 non-cover-pairs in the training set and 322 cover and non-cover pairs in the validation set. We use 2,113 cover-pairs and randomly drawn 100,000 non-cover pairs for training. The size of the appropriate dataset was determined through a model validation. Detailed validation results are listed in the previous study [3].

3. ACKNOWLEDGEMENT

This work was supported partly by Kakao and Kakao Brain corporations and partly by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2017M3C4A7078548).

4. REFERENCES

[1] Ioffe, Sergey and Szegedy, Christian: "Batch normalization: Accelerating deep network training by reducing internal covariate shift" *arXiv preprint arXiv:1502.03167*, 2015

[2] Jiang, Nanzhu and Grosche, Peter and Konz, Verena and Müller, Meinard: "Analyzing chroma feature types for automated chord recognition" *Audio Engineering Society Conference: 42nd International Conference: Semantic Audio*, 2011

[3] Juheon Lee, Sungkyun Chang, Sangkeun Choe, Kyogu Lee: "Cover Song Identification Using Song-to-Song Cross-Similarity Matrix With Convolutional Neural Network" *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)* , 2018

[4] Kingma, Diederik P and Ba, Jimmy: "Adam: A method for stochastic optimization" *arXiv preprint arXiv:1412.6980*, 2014

[5] Lee, Kyogu: "Identifying cover songs from audio using harmonic representation" *extended abstract submitted to Music Information Retrieval eXchange task*, 2006

[6] Müller, Meinard and Ewert, Sebastian: "Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features" *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011. hal-00727791, version 2-22 Oct 2012.

[7] Serra, Joan and Gómez, Emilia and Herrera, Perfecto: "Transposing chroma representations to a common key" *IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, 45-48, 2008

[8] Silva, Diego F and Yeh, Chin-Chin M and Batista, Gustavo Enrique de Almeida Prado Alves and Keogh, Eamonn and others: "SIMPle: Assessing music similarity using subsequences joins" *International Society for Music Information Retrieval-ISMIR*, 2016

[9] Srivastava, Nitish and Hinton, Geoffrey and Krizhevsky, Alex and Sutskever, Ilya and Salakhutdinov, Ruslan: "Dropout: a simple way to prevent neural networks from overfitting" *The Journal of Machine Learning Research*, 15-1, 1929–1958, 2014