COVER SONG IDENTIFICATION USING SIAMESE CONVOLUTIONAL NETWORKS

John Banta, Noah Schaffer, Paulina Tarasul

Northwestern University

1. INTRODUCTION

Cover song identification tasks determine whether a particular recording is a cover of another recording, e.g. it is a different performance of the same song by a different artist. Using Convolutional Neural Networks, we created a system which takes in two songs and determines if one is a cover song of the other by creating similarity representations of two songs.

Music-hosting websites often want to know whether or not a song is a cover of an already published song, so that they can make sure it is appropriately labeled as such. In addition, this is of interest in the music industry when it comes to plagiarism; often songs are covered or used without the permission of the initial artist. Since the neural network is being trained to identify cover songs based on similarity of content, this could possibly be expanded to identify plagiarism.

The ranking based on similarity of content could also be extended for use in music recommendation systems, as a more effective method than recommendation by similarity of user history and preferences. A neural network that can take in an audio file and identify whether or not it is a cover song would be useful for processing the large volume of such songs. In terms of intellectual interest, it will be very interesting to work with a neural network aimed at classifying something as complex as a song, and to find out what a network needs in order to do so effectively.

2. RELATED WORK

Seetharaman and Rafii (Seetharaman and Rafii, 2018) used 2D Fourier Transform sequences to identify cover songs. They represented the songs as Constant Q Transforms, a close relative of the fourier-transform which is key-invariant due to linear frequency shiftings mirroring pitch shifting in the music, and created diagonal matrices to match the CQTs of potential covers to the CQTs of originals. It compared its methodology to machine-learning methods using Mean Average Precision, P@10, and MR1 (mean rank of correct cover). This paper was extremely useful since it introduced us to the concept of the CQT, but it did not use machine learning at all, and as a result its ability to learn covers became far worse as melodies became more and more dissimilar (e.g. in a jazz rendition of My Favorite Things with extensive melody variations).

Yu et al. (Yu et al., 2019) used a CNN with temporal pyramid pooling to learn key-invariant representations of songs for cover song identification similarly to the way CNNs learn shift-invariant representations of images for image recognition. However, this network did not simultaneously feed two songs into the network, instead using the network to extract representations of each song where the song with the shortest euclidean distance from a reference song was identified as a cover song. While this paper was helpful in understanding why convolutional architecture is useful in cover song identification, it was unclear how to train the network to extract meaningful representations of songs that will cause covers to have lower euclidean distances than two pairs of random songs.

Stamenovic (Stamenovic, 2020) introduced using Siamese convolutional networks for cover song identification. The network was trained on pairs of songs and used binary cross entropy loss to create a representation of distance between two songs. For evaluation, the network compared a reference song to every song in the cover set and calculated P@1, where the network was correct if the reference song was closest to its cover song and incorrect otherwise. While our network has a similar architecture to this, we focused on evaluating cover songs based on average ranking of a cover song rather than just the precision. We wanted to focus on not only how precisely our network can identify cover songs, but also how far off the cover song is from the highest ranking when it does not correctly identify a cover song as its closest song.

3. MODEL AND TRAINING

3.1. Overview of Approach

Using a Siamese Convolutional Network modeled after the network introduced in the Stamenovic paper (Stamenovic, 2020), the system takes in a pair of songs, one from the a set of reference (original) songs and one from a set of cover songs. A reference/cover pair is labeled 0 if it contains two renditions of the same song and 1 if it does not. The network uses these labels to learn a similarity representation of two songs by the distance of the representation of these songs.

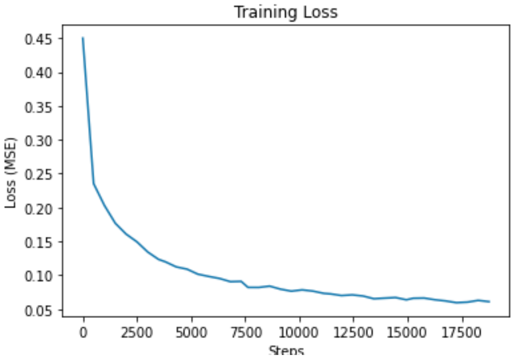

Fig. 1. MSE Loss over 5 epochs of training

3.2. Dataset and Input Representation

For both the training and testing set, we used online datasets of cover songs, collected as MP3s and converted to the Constant Q Transform format using the Librosa library. For the training set, a collection of MP3 files was curated using the Million Song Dataset [Bertin-Mahieux et al, 2011]. The original dataset was stored as a list of links to the Secondhand Songs dataset, which contained information and youtube video performances for cover songs as well as their original versions; we converted these youtube videos into MP3s and collected 173 reference songs and 353 cover songs. With these cover songs, we created 61069 reference/cover pairs.

3.3. Network Architecture and Training

Our network architecture is a four-layer convolutional network followed by a linear layer, a sigmoid, and a second linear layer. Two songs are passed through the convolutional network, and then through a linear layer and a sigmoid layer. The absolute value of the difference of the sigmoid representation of both songs is then passed through a final linear layer, which outputs a prediction for the label of the two songs.

We trained using an adaptive learning rate (ADAM) optimizer with an initial learning rate of .001. We measured error by taking the mean squared error of the output of our network with the label of the song pair [Figure 1]. We trained with a batch size of 16 over 5 epochs. When sampling our data for training, we used a custom PyTorch dataset which used random generation to sample approximately equal amount of data with a label of 0 and a label of 1.

4. TESTING AND RESULTS

4.1. Dataset

The testing dataset was the covers80 dataset [D. P. W. Ellis, 2007], a dataset collected by D.P.W. Ellis with 80 reference songs and one cover song per reference songs. This resulted in 3600 reference/cover pairs to be evaluated.

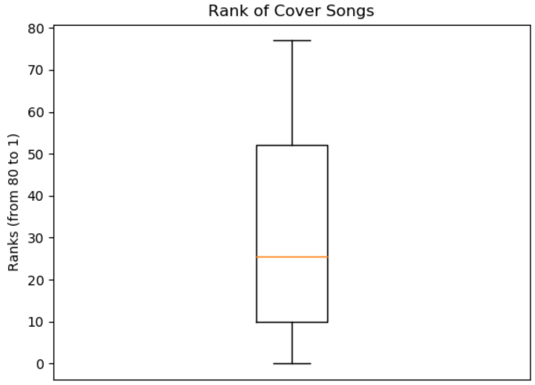

Fig. 2. Boxplot of ranks assigned by reference songs to their covers

4.2. Evaluation

We tested the network by comparing each reference song in the testing set to each of the cover songs, and producing a ranking of the cover songs by similarity to the reference song. For each reference song, we then found where the correct corresponding cover song had been ranked. We measured our network's success by how many cover songs were correctly ranked as their reference song's first most similar, as well as how many cover songs were correctly ranked in the top ten most similar to their reference song. We chose to compare our network's performance to the probable results of rankings set by chance alone.

4.3. Results

We compared our results to a baseline of identifying a cover song based on random chance, which would be that given two songs the network generates a random label between 0 and 1.

For the 80 reference songs and 80 corresponding cover songs, we found that the network correctly ranked 3/80 covers first for their reference songs, and 21/80 correct covers in the top ten. This result is certainly better than chance. If the cover songs were ranked purely by chance, then the probability of any reference song correctly ranking its cover first would be 1/80. The probability of 3 reference songs ranking their correct covers first, then, would be $1.95x10^{-6}$, making our network's performance significantly better than chance. The probability of 21/80 covers being correctly ranked in the top 10 for their reference songs would be $1.08x10^{-19}$ if the ranking were to happen by chance, making our network's performance in this respect significantly better than chance as well. This means that our network performs very well as compared to the baseline to which we chose to compare it, that of chance performance.

In figure 3 the ranks assigned by reference songs to their covers are shown; a rank of 1 means that this reference song is the most likely original for this cover, and vice versa for rank 80. The median rank was 25.5, which is significantly better than the random-chance median of 40.

Compared to a baseline of random chance, our model seems to have significant results for cover song identification of the Covers80 dataset. However, compared to other baselines such as the $P@1$ of the network in the Stamenovic paper (Stamenovic, 2020), our network does not perform nearly as well due to a variety of factors. Though we do see results better than random chance, improvements to our design can help the network perform at a level closer to other baselines.

5. FUTURE WORK

To create a CQT with equal dimensionality for every song, we clipped each song to the first 30 seconds. To improve accuracy, future iterations of the model could take in various windows from a single song or take in longer windows of a song. This could have impacted accuracy for songs that may have introductions without a distinctive melody or covers which added or omitted parts of the beginning of the song. In these cases, it would be hard to identify covers with just this short of a window. Taking longer windows or several windows throughout the song may help to better target similarities between songs.

Another way we could improve accuracy of our network is by changing our network to train on the ranking function using pairwise loss. Our current network only learns on a binary classification, which is 0 if it is a cover and 1 otherwise. Using a ranking function in training with pairwise loss may allow the network to more easily identify similarity between songs as opposed to simply classification as a cover or not. This may explain why some cover songs in our experiment were ranked so low, as the network wasn't able to fully learn similarity representations with only binary labels so anything that is not close to 0 is equally as likely to be close to 1 when passed through the network.

Finally, we could explore additional results using some of the more advanced techniques in the Seetharaman/Rafii paper like Adaptive Thresholding, in which values in a frequency array are set to 1 or 0 depending on whether they are above or below a threshold (Seetharaman and Rafii, 2018). This removes timbral information, thus making comparisons between covers instrument-independent and possibly increasing accuracy in the case of covers with a great deal of variation in instrumentation.

6. REFERENCES

1. Manocha, P., Badlani, R., Kumar, A., Shah, A., Elizalde, B., amp; Raj, B. . Content-based Representations of audio using Siamese neural networks. 15 February 2018; pp. 1-5

2. Seetharaman, P.; Rafii, Z. Cover song identification with 2D Fourier Transform sequences. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 616–620.

3. Stamenovic, M. Towards Cover Song Detection with Siamese Convolutional Neural Networks. May 2020; pp. 1-4

4. Yu, Z.; Xu, X.; Chen, X.; Yang, D. Temporal Pyramid Pooling Convolutional Neural Network for Cover Song Identification. In Proceedings of the International Joint Conference on Artifical Intelligence (IJCAI), Macao, China, 10–16 August 2019; pp. 4846–4852.