IDENTIFYING COVER SONGS USING DEEP NEURAL NETWORKS

Marko Stamenovic
University of Rochester
mstameno@ur.rochester.edu

ABSTRACT

A cover song, cover version, or simply cover, by definition, is a new performance or recording of a previously recorded, commercially released song. It may be by the original artist themselves or a different artist altogether. Automatic cover song detection has been an active research area in the field of Computer Audition for the past decade. In this paper, we propose a novel method for cover song detection using automatic extraction of audio features with a stacked auto-encoder (SAE) combined with beat tracking in order to maintain temporal synchronicity.

1. INTRODUCTION

The proliferation of cheap digital media creation tools and free web based publishing platforms has led to an ever-expanding universe of audio-visual content available for all to access on the world wide web. Although much of this content is original in nature, a stunning amount is cover material. For example, a recent search of a popular video sharing site for the term “Beatles cover” turned up 3.97 million matches. Over their entire career, The Beatles released a total of 257 songs.

A cover song may vary from the original song in tempo, timber, key, arrangement, instrumentation and/or vocals. More often than not, the most prevalent parts of an original song carried over to the cover song include the melody of the song and the chorus, especially in the case of pop music. The wide variety of variables creates a very challenging and interesting classification problem. Although cover song identification is not an outwardly extremely important problem, it is a form of music similarity recognition which is one of the key aspects of music information research.

2. OVERVIEW

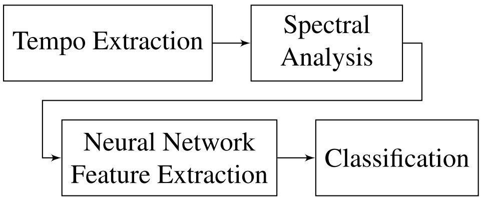


Figure 1. System block diagram

Ours is a four-step process. First the song is analyzed for tempo using the beat tracking algorithm implemented by D. Ellis [5]. The length of one beat of the song, or a 1/4 note assuming a 4/4 time signature, is saved. Second, a spectro-temporal analysis is performed on the entire audio file. We try both a constant Q transform (CQT) and 12-semitone shifted chroma vectors. The window size is set as the length of the extracted beat and the overlap is set at 50%. The spectrogram is then segmented into patches, each containing 8 frame of the spectrogram. This corresponds exactly to four beats with 50% overlap or one musical measure, assuming 4/4. Thirdly, each patch is reshaped into a vector and fed into a two-layer SAE. The SAE uses backpropagation to automatically extract relevant features from the audio. It is trained on an input set consisting on all of the original songs. Finally, each original and corresponding cover song is fed into the SAE to create an output feature vector. The ”distance” of each cover song’s feature vector is measured from each original song’s using a dynamic time warping and euclidean distance. If the correct cover song is closest to its corresponding original song, the classification is deemed a success.

3. RELATED WORK

Previous work in this field has seen a variety of methods, the most successful of which have used beat-by-chroma feature extraction and cross-correlation to identify matches [2] [3] [4]. Beat-by-chroma feature analysis is a method of spectral analysis which bins the entire spectrum of audio into 12 frequency bins corresponding to the 12 notes of the semitonal musical scale and indexes them in time to match the song’s tempo. Our method attempts to build on this method by introducing the automatic feature extraction of the neural network to learn time and spectral features which may be lost using other methods.

4. IMPLEMENTATION

The proposed system consists of the four main components shown visually in Figure 2 and described in the Overview section. An additional preprocessing step to prepare the audio for feature extraction is also performed.

4.1 Dataset

We use a slightly augmented “covers80” dataset [4] proposed at MIREX 2007 to benchmark cover song recognition systems. This dataset contains 80 sets of original and cover songs - 166 total - spanning genres, styles and live/recorded music. The dataset is biased towards western pop/rock music. Most songs contain only one cover version however some songs contain up to three. For speed of data processing and iteration, we trimmed the dataset to 80 pairs of original and cover songs, for 160 songs total split evenly.

4.2 Preprocessing

The files are converted from monophonic 16 kHz mp3 files into monophonic 16 kHz wav files. They are then input into an open-source automatic beat-tracker [5]. The software computes a vector containing beat onset times. The files are then truncated to start at the first beat and end at the last beat, in order to mitigate any intro or outro discrepancies caused by common cover music attributes such as clapping from live performances or extended introductory speeches. The minimum time between beats was chosen from the beat tracking vector for use in determining window size of the spectral analysis.

4.3 Spectral Analysis

Two forms of spectral analysis are implemented during iterative testing of our system, a constant-Q transform (CQT) and a 12-semitone shifted stacked set of chroma features.

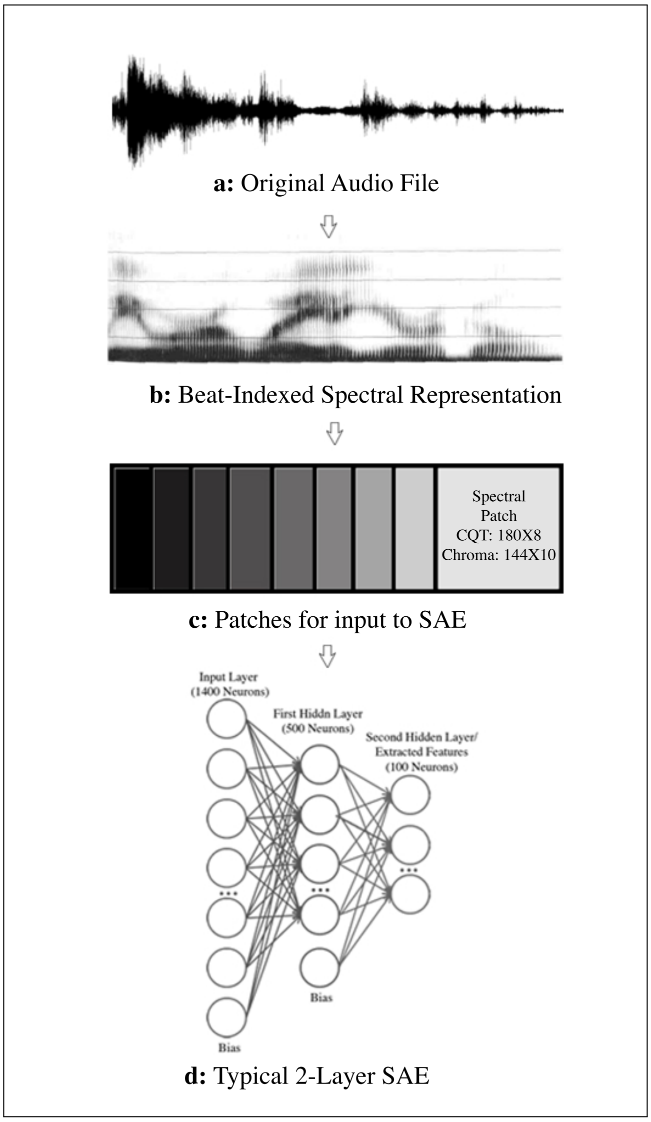
4.3.1 CQT

A 9-octave (20-8000 Hz) Constant-Q Transform (CQT) is then employed to calculate the cover song spectrogram using the MATLAB CQT toolbox [6]. Each octave contains 20 bins with a total of 180 frequency bins. The time frame hop size corresponds to one beat calculated during beat tracking. We use CQT instead of short-time Fourier transform (STFT) because the log-frequency scale in CQT better corresponds to human auditory perception.

In order to effectively train the SAE, the CQT spectrogram is segmented again into fixed-size patches. The length of each patch is set to four times the previously calculated mean beat time. As most of the music in the dataset is pop music, it is likely written in the 4/4 time signature. Taking a 4-beat long patch will be equivalent to taking one measure with a 50% overlap in case of any beat tracking discrepancies. One measure is chosen as it is a sufficiently granular chunk of time to have many iterations per song with which to train the SAE. However, it is also long enough to exhibit temporal musical evolution which we are hoping the SAE will learn and eventually add to the feature representation of each song for more accurate classification.

4.3.2 Chroma Features

As a comparison, a parallel analysis was done using chroma spectrum features. Chroma features collect spectral energy from each semitone in each octave and combine them into one spectral bin per semitone. They are a convenient way to spectrally represent music, they compress the spectral content into a musically relevant plane. Our chroma extraction begins with the implementation in [9]. The spectral extraction is again windowed by beat length with a 50% overlap. This results in a chromagram of 12 semitone frequency bins by 1/4 note temporal bins.


a: Original Audio File

b: Beat-Indexed Spectral Representation

c: Patches for input to SAE

d: Typical 2-Layer SAE

Figure 2. Illustration showing an overview of our process. a: original audio file. b: typical beat-matched spectral representation either CQT or wrapped Chroma features. c: patching of spectral representation for input into the SAE. Each patch is one measure long with 1/8th note hop size. d: 2-layer SAE such as the one we use for feature-extraction.

In order to account for any key change that might occur in the cover version, the chromagram is shifted or “wrapped” to cover all possible key changes. This is accomplished by creating a one-semitone shifted chomagram for each possible key change and then stacking them all together. The output after stacking is a 144 semitone frequency bin by 1/4 note temporal bin chromagram. Thus when feeding the chromagram into the SAE, the SAE learns the song’s chroma features in every possible key at once.

Again, the data is formatted into patches for the SAE. In order to keep parity between the neural networks, the chroma features are segmented into patches of 10 spectral frames, corresponding to one measure plus one beat of the music. Although not as clean a representation as on measure per patch, this still allows a large amount of data inputs into the SAE and additional length for temporal musical evolution.

4.4 Feature Extraction

Each patch is reshaped into a vector and fed into the neural network. Feature extraction is performed by a two hidden layer stacked auto-encoder with 1400 input neurons, 500 first layer hidden neurons and 100 output neurons. Each neuron is fully connected to every other neuron in an adjacent layer. There is an additional bias weighting for each hidden layer that is also connected to every neuron in that layer as shown in Figure 2.d. Initial forward activations for each hidden layer are computed independently of the overall structure. Back-propagation is calculated using the L-BFGS algorithm [10].

Due to the nature of the data chunks fed into the by the SAE, we hope to phrase the input so that the model learns features corresponding to the evolution of the music over time. This temporal evolution, which is easily detectable by humans, cannot be replicated by strictly spectral feature extraction systems such as Chroma analysis and Mel-Frequency Cepstral Coefficients (MFCC's). We hope it also adds meaningful information over a combination of simply beat-indexed spectral features, as the neural network will learn the patterns in musical change over time, rather than a rigid index of spectrum vs time.

Furthermore, features learned automatically by neural networks, specifically SAE's, have recently shown promise in musical feature extraction. An SAE similar to the one in [1] is chosen for this system due to its proven efficacy in extracting features from audio.

Figure 3 shows a visualization of the first 100 features of the first hidden layer. There are 500 hidden features in this layer but only the first 100 are shown for illustrative purposes.

4.5 Classification

Two distance measurements are then used to evaluate similarity of extracted SAE features against ground truth. Distance is measured between the original song feature vectors and all candidate song feature vectors using two metrics: dynamic time warping (DTW) and Euclidean Distance. Figure 4 shows a visualization on the extracted feature matrices of three songs for classification.

4.5.1 Normalized Dynamic Time Warping

DTW is an algorithm used to efficiently measure time-series similarity between two temporal sequences which may vary in speed or length. It minimizes the effects of time shifts by allowing an 'elastic' transformation of time series in order to detect similar shapes with different phases [8]. DTW has been used extensively to measure distance and align temporally shifted audio, particularly in voice recognition applications [7] but also in song recognition systems [3]. Since our output features vectors have different lengths, we also normalize the DTW over the sum of the lengths of the feature vectors being compared. This is

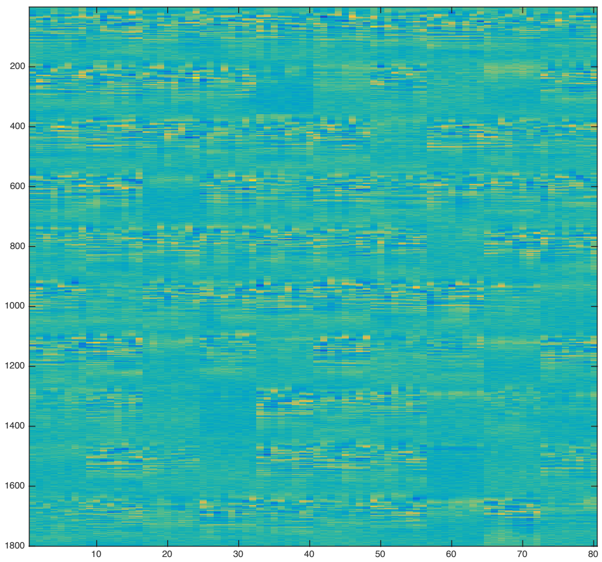


Figure 3. 10X10 visualization of the first 100 features of the first hidden layer of the SAE. Each feature is a 180X8 square patch

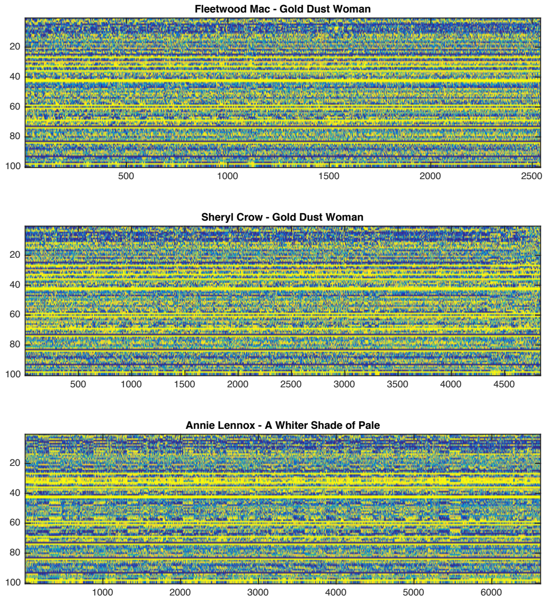


Figure 4. Visualization of extracted features for three songs. The first two graphs show a correctly identified original/cover pair while the third song is unrelated and included for comparison. Columns correspond to extracted features, rows correspond to time in beats.

done in order to not give preferential weighting to songs that are of similar lengths.

4.5.2 Euclidean Distance

Euclidean distance is used to compute a“bag-of features” distance between song feature representations in order to evaluate the benefit of DTW. The mean of the time-feature representations is averaged over time to form a single 100-dimensional feature-vector for each song. Euclidean distance is then measured between feature vectors between each original and cover.

5. EXPERIMENTAL RESULTS

The overall results show a maximum classification accuracy of 13.75%, or 11 correctly identified pairs out of 80 for both types of spectral analysis as shown in Figure 5. DTW gives better results than bag of features for both as well, indicating time alignment of the extracted features does provide significant benefits. Although it is not as high as the DTW, the bag of features classification shows a better performance for extracted features of chroma spectral analysis. This indicates that chroma features may contain more useful spectral data for classification to be fed to the neural network.

1/2 Note Patch Overlap Accuracy		
	DTW	Bag of Features
CQT	13.75%	7.5%
Chroma Features	13.75%	10%
Random Guess	1.25%	1.25%

Figure 5. Results for our system using both Chroma and CQT spectral data with a 1/2 note patch overlap. Results are calculated using DTW similarity and bag of features euclidean distance similarity.

Additional tests were performed using smaller patch hop sizes to train and classify the data which seemed promising, but were not completed due to CPU and hard disk processing restraints. For example, the same experiment as above was run using 1/8th note patch hop size and chroma feature extraction. This hop size results in four times as many output features as using a 1/2 note patch hop size. Bag of words results for this test showed 17.5% accuracy or 14 correctly classified songs out of 80. The result actually is better than DTW for a corresponding analysis with smaller hop size. The DTW calculation for 1/8th note hop size and chroma features could not be calculated due to time constraints. DTW comparison between one cover song and the entire song set took approximately one hour, depending on song length. However, when extrapolating the DTW improvement over bag of words for larger patch size to the bag of words results for the smaller hop size, we would expect an accuracy in the range of 19% percent or 15 correctly classified songs out of 80.

An overall results matrix is shown in Figure 6. The rows correspond to original songs while the columns correspond to cover songs. Green indicates a higher similarity while red indicates a lower similarity. 100% classification would show solid green values down the diagonal of the matrix from top left to bottom right. Although there are some clusters of green around the diagonal, the more prevalent trends seem to be that certain rows or columns are classified as being closer to many of the songs, while other rows or columns are classified as being further from a large group of the songs. For example the strong red column shown in the figure corresponds to the cover song “Happiness is a Warm Gun” by Tori Amos. This song was judged as being less similar to all the songs in the cover set than any of the other songs tested.

Although our methodology does provide significantly better results than random guessing, they are far below the state of the art for this dataset of 67.50% [4].

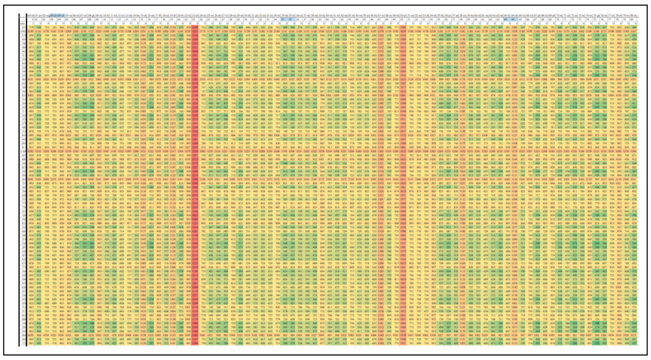


Figure 6. Color coded results matrix for the CQT DTW. Rows correspond to original songs and columns correspond to cover songs. Green indicates a higher similarity while red indicates a lower similarity.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented and tested a novel system for automatically classifying cover songs based on spectral analysis and automatic feature extraction using a stacked autoencoder. We used two different spectral analyses, a CQT and wrapped chroma features, to extract features and two different classifiers, DTW and bag of features, to analyze the results. Although our results are far below the current state of the art there are many avenues for improval.

The first would be to test various window/hop size combinations for both the spectral analysis and the patches. As mentioned in the Results section, additional testing showed promise for smaller patch hop sizes and larger spectral window sizes, allowing bag of words with a smaller patch hop to actually outperform DTW with a larger patch hop.

Another future line of work will be to analyze the results matrix in Figure 6 to determine any possible trends between highly similar songs and highly dissimilar songs.

Another future line of work would be to implement some kind of song part extraction system before the input to our system, in order to hopefully extract the chorus of the song. It has been mentioned in the introduction that the chorus of the cover song and original song are more often than not the most similar parts of the songs. Therefore it would stand to reason that comparing song choruses would yield a better match than comparing the entire song files. This would have the added benefit of compressing data along the time axis for faster calculation.

Finally, we would like to experiment with different neural networks both size of hidden layers and in depth. We believe that a deeper neural network will be able to extract more abstracted and relevant features.

7. REFERENCES

[1] Zhang, Yichi, Duan, Zhiyao. “Retrieving Sounds by Vocal Imitation Recognition,” *2015 IEEE International Workshop on Machine Learning for Signal Processing*, Boston, MA, 2015.

[2] Ellis, Daniel P.W. “Identifying ”Cover Songs” with Beat-Synchronous Chroma Features,” *Music Information Retrieval Evaluation Exchange*, 2006.

[3] Lee, Kyogu. “Identifying Cover Songs from Audio Using Harmonic Representation,” *Music Information Retrieval Evaluation Exchange*, 2006.

[4] Ellis, Daniel P.W., Cotton, Courtenay V. “The 2007 LabRosa Cover Song Detection System,” *Music Information Retrieval Evaluation Exchange*, 2007.

[5] Ellis, Daniel P.W. “Beat Tracking by Dynamic Programming,” *LabROSA, Columbia University, New York/*, New York, NY, 2007.

[6] Christian Schrkhuber and Anssi Klapuri. “Constant-Q transform toolbox for music processing,” *Proc. 7th Sound and Music Computing Conference*, Barcelona, Spain, pp. 3-64, 2010.

[7] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi. “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques,” *Journal of Computing*, Volume 2, Issue 3, March 2010.

[8] Pavel Senin. “Dynamic Time Warping Algorithm Review,” *Information and Computer Science Department University of Hawaii at Manoa*, December 2008.

[9] Labrosa.ee.columbia.edu, “Chroma Feature Analysis and Synthesis”, 2015. [Online]. Available: http://labrosa.ee.columbia.edu/matlab/chroma-ansyn/. [Accessed: 19- Dec- 2015].

[10] Malouf, Robert (2002). “A comparison of algorithms for maximum entropy parameter estimation.” Proc. Sixth Conf. on Natural Language Learning (CoNLL). pp. 49?55.