Learning to Listen: Matching Song Covers to Original Songs via Supervised Learning Methods

Aroma Mahendru* MAROMA@VT.EDU
Arijit Ray* RAY93@VT.EDU
Bijaya Adhikari* BIJAYA@VT.EDU
Jason Granstedt* EVENETH@VT.EDU

*Equal Contribution

Abstract

Cover song identification is a trivial problem for human beings. A human can easily identify whether a song is a cover of another or not, without using much effort. However, cover song identification is not an easy task for machines, as the numerical audio data differs significantly. Since a major goal of Machine Learning and Artificial Intelligence is to make machines mimic human-like cognition capabilities, cover song recognition seems to be an interesting problem in the field. This project deals with matching cover songs to the original ones by using features extracted from the Million Songs Dataset, which provides features for about one million songs. Additionally, SecondHandSongs data provides clusters of the original and cover songs grouped together. In this project, we combine these two data-sets to train traditional and slightly modified versions of traditional machine learning algorithms to match cover songs to their original versions.

1. Introduction

Music plays a very important role in people's lives. Ever since the growth of the digital era, the search space of music data is ever increasing. Thus, to manage and utilize all the data on the internet effectively, it is very important to understand the semantic nature of songs and use it to catalog them. If we can make machines do the task for us, it will be another tedious job off the list for humans.

Matching a cover song to its original one is one such task. Humans can remarkably recognize music similarity tunewise, even when there are major changes made in the style of music. Even without the lyrics, we can easily recognize instrumental covers of popular songs based on tune similarity. It is hence a trivial task for human beings. However, tune similarity is a very abstract concept. A piece of audio can be deemed similar in tune even when the frequency, pitch and harmonics seem different numerically. For example, a drum beat cover of a song originally in violin will have completely different frequencies and harmonics, but will have a similar tune as perceived by humans. This project aims at learning the inherent concept of musical similarity based on audio features by solving the task of matching cover songs to their original ones using machine learning. We delineate the performance of various vanilla machine learning models for this problem and provide insights for further exploration.

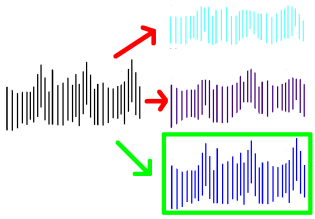

Figure 1. This figure illustrates the underlying theme of the project. The idea is to learn to match a cover song to original song by learning a model which learns relevant features.

As we will see from literature review, as well as the preliminary performance of our models, the problem of cover song matching remains a challenge yet to be solved. The fact that humans have a high performance in identifying matches based on audio input alone (no lyrics) show that there is a lot of room for improvement if proper data and models are used.

Virginia Tech CSx824/ECEx424 final project report.

2. Related Work

A lot of work has been done in the area of understanding music semantics. Genre classification, artist classification, musical instrument identification is a few of many such tasks.

In machine learning for example, taks like guitar cord recognition, audio features detection and drum pattern classification (Humphrey et al., 2012; Battenberg & Wessel, 2012) have been solved using convolution and recurrent networks. The most popular task however is the task of music recommendation. For example, websites like Spotify and Pandora use state-of-the-art recommender systems to suggest music to their customers on the basis of their past listening history.

Specifically. the problem of cover song matching is also not new. However, it has mostly been approached from an audio/signal processing point of view (Serrá et al.; Wolkowicz et al.; Jensen et al., 2008; Bertin-Mahieux & Ellis, 2011; 2012). One of the works (Ellis & Poliner, 2007), extracts specific features from the raw audio files and use a dynamic programming based algorithm to match input pair(s) audio file to give a similarity score. There is no training involved at all.

Another approach (Unal et al., 2011) is from an information theory point of view and slightly close to machine learning. The idea is to use n-gram model of extracted harmonic progression features from the 'training set'. Using these features, perplexity based similarity score is calculated between the audio files.

The closest work to ours perhaps is (Ravuri & Ellis, 2010) where after feature extraction, SVM and multilayer percetron are used for classification of cover songs. However, all these works approach the problem as classification of a query pair which is a binary classification problem. Our paradigm however involves multi-class classification.

3. Approach

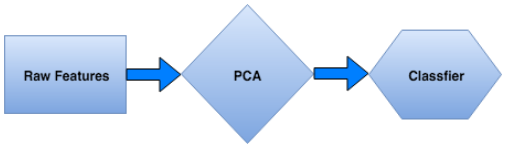

Figure 2. Basic Approach

As illustrated in Figure 2, we first applied PCA to the feature set and subsequently compared the performance of various vanilla machine learning models to classify various cliques of cover, original songs. Specifically, we implemented and compared the following supervised learning models:

1. Nearest Neighbors
2. Clustered Nearest Neighbors
3. Multi-Layered Neural Networks
4. Multi-Layered Perceptrons

3.1. k-Nearest Neighbors

Here, we match songs directly according to the minimum euclidean distance between feature vectors. As the features are comprehensive about the entire song, and not temporal in nature, we hoped that a naive feature matching will do comparable to more complex machine learning models. In fact, this method performs the best in the ablated features experiment.

3.2. Clustered Nearest Neighbors

Here, we measure mean feature vectors of each cluster and assign test song to cluster with minimum euclidean distance of feature vector to mean cluster features. We hoped that each clique is based on a similar set features that describe the song and thus, clustering in a supervised manner would be a reasonable approach for this problem.

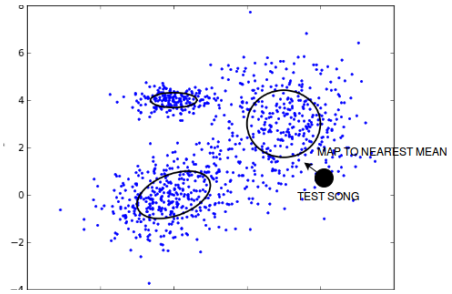

Figure 3. Nearest Neighbor Approach

3.3. Multi Layer Neural Network

Here, we train Neural Networks to learn the cluster ID for each song. We can either measure loss as a Softmax with cross entropy over an output space of 4128 (there are 4128 cover song cliques). Or, we can simply map each song feature vector to the cluster ID (numbered 1 to 4128) and use a mean-squared error loss. The former performs better as expected. We use a 4-Layered Neural Network with mixed ReLU and Tanh non-linearity in each of the hidden layers.

3.4. Multi Layer Perceptron

This is similar to the above approach on neural networks, except that we remove all non-linearities. This performs superior to a neural network model with non-linearities at the hidden layers.

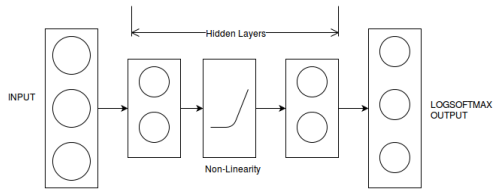

Figure 4. Neural Network

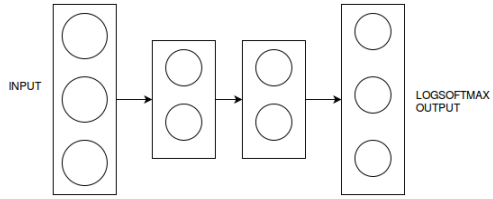

Figure 5. Multi Layered Perceptron with no non-linear activations

patterns. These features are made available as MSD Benchmarks (Schindler et al., 2012) which have been extracted from downloaded audio samples, mostly in the form of 30 or 60 second snippets.

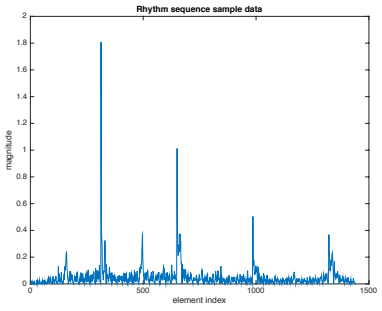

Figure 6. Sample rhythm sequence

4. Experimental Setup

4.1. Dataset

The Million Song Dataset (Bertin-Mahieux et al., 2011) is also a cluster of complementary datasets contributed by the community like cover songs, lyrics, song-level tags and similarity, user data and genre labels. We planned to use the SecondHandSongs (Bertin-Mahieux et al.) dataset for cover songs, which has 12,960 training and 5236 test songs. The dataset includes features related to how the song sounds like as in loudness, tempo as well as purely informational features like artist, song title, etc.

We removed all song identification features for this task of song matching. The features were finally 29 dimensional. Since the training and test data had separate sets of song 'cliques' (set of original and all its cover songs), we decided to split the training set itself into 10,000 songs for training and the rest 2,960 for validation. Note that this dataset contains a total of 4,128 song cliques. This gives an average of 3.1 songs per clique.

We however know that the real world scenario will not just be identifying the correct clique but in fact a majority of songs will belong to no cliques at all. Hence, we added random 4000 songs from the Million Songs Dataset for the experiment which mimics this scenario.

Enriched Features We found that the above features were not discriminative enough for the complex task of multi-class classification amongst 4k categories. Hence, performing classification over a more enriched set of features was a reasonable choice. For this, we used the 1440 dimensional rhythm patterns extracted from the SecondHandSongs dataset. Choosing rhythm patterns was a reasonable choice since similar sounding songs have similar rhythm

4.2. Setup

For the above dataset, we performed main three sets of experiments. We first evaluated all of our approaches on the basic dataset. Following that we evaluated our approaches on the rhythm pattern dataset. Finally, we add noisy songs which don't belong to any of the cliques to make the dataset unbalanced and mimic the real world case and perform evaluation on this set.

5. Results

5.1. PCA Results

Looking at the variance values sorted in descending order by the PCA algorithm (Figures 7,8), we see that most of the variance is captured within the 5 features on the basic 29 features data and within 200-300 features on the full 1440 features data. This shows that using PCA might help in the pipeline for our classification.

5.2. Quantitative Results

We now present results on all the three experiments. Table 1 shows the results on the dataset with the basic set of features. As expected, we find that basic approaches like nearest neighbor classification and clustered nearest neighbors perform quite better than complicated approaches like various variants of neural networks. This was expected since there are 3.1 songs per clique on an average and total 4128 classes in all. However, since the random baseline is much lower, it is safe to say that there does exist a pattern in the data and there is a scope of learning some trend.

Table 2 shows the results on the full 1440 features. As expected the neural networks perform better with more fea-

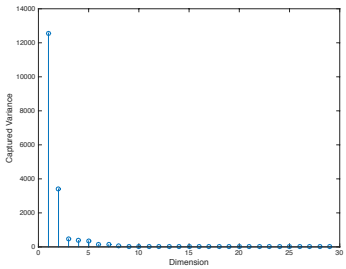

Figure 7. Histogram of features for basic dataset after PCA.

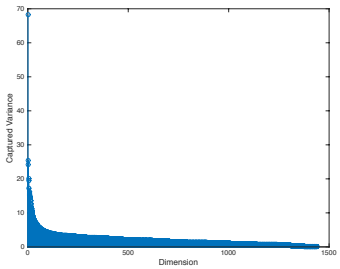

Figure 8. Histogram of features for enriched rhythm dataset after PCA.

APPROACH	TOP-5	TOP-10	TOP-100
NEAREST NEIGHBORS	56	98	502
CLUSTERED NN	39	66	341
NEURAL NETWORK	32	51	231
MLP	5	8	93
RANDOM BASELINE	3.5	7	72

Table 1. Number of songs grouped in the right clique in the top-k rank using 29 basic features

APPROACH	TOP-5	TOP-10	TOP-100
NEAREST NEIGHBORS	60	88	389
CLUSTERED NN	61	87	334
NEURAL NETWORK	43	67	252
MLP	46	86	412
RANDOM BASELINE	3.5	7	72

Table 2. Number of songs grouped in the right clique in the top-k rank using full 1440 features

APPROACH	TOP-5	TOP-10	TOP-100
NEAREST NEIGHBORS	57	90	282
CLUSTERED NN	53	79	306
NEURAL NETWORK	61	92	466
MLP	74	111	542
RANDOM BASELINE	3.5	7	72

Table 3. Number of songs grouped in the right clique in the top-k rank using full 1440 features on unbalanced dataset

tures, but the rise in performance isn't that high. When we remove all non-linearities from the Neural Network to make it a Multi-Layer Perceptron, it surprisingly performs much better. We hypothesize that this is because the negative gradients are not clipped in the MLP due to absence of ReLU's in the hidden layers. As expected, we observe that performance of all algorithms improved with increased number of features. As per our intuition, there was a significant increase in performance of complex models, namely Multi-Layer Perceptron, and Neural Network.

Table 3 shows the results obtained by adding songs that do not belong to any clique to the dataset. This shows superior performance compared to the previous experiments. We hypothesize this is partly because of more data and partly because the classifier has more information on how to distinguish between clique and non-clique songs. Once again, a MLP performs much better than a Neural Network in classifying the data.

Across different sets, there is a slight improvement from basic feature set to enriched feature set and a slight dip on the unbalanced set for the pure learning methods like Neural Network and Multi-Layer Perceptron. Performance stays more or less the same or in fact dips for Nearest Neighbor based techniques.

5.3. Qualitative Results

We picked random top-1 correct prediction by Nearest Neighbor model on the unbalanced dataset to see how the rhythm patterns of the songs look qualitatively (see Figures 9,10). See that it does pick out similar looking rhythm sequences. An interesting observation was that not only the exact same rhythm sequences were picked but also sequences with similar trends but different scales were also picked.

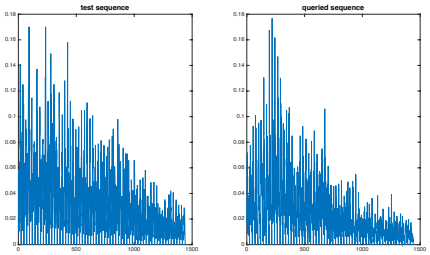

Figure 9. Example on the left shows the cover song and the queried song whose rhythm sequences are quite similar to each other (Top-1 Prediction). Song: "Hey! Little Child" originally by Alex Chilton.

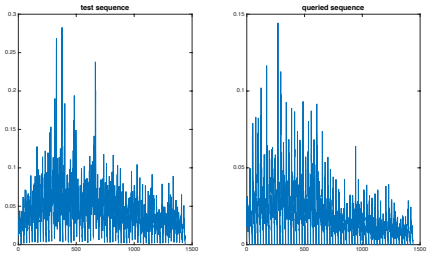

Figure 10. This example however shows an example in which the shape matches but the scale of the sequences are different. Song: "Up in the Air" originally by Hüsker Dü.

5.4. Feature Analysis with EM and Gaussian Mixture Model

We also examine the log likelihood of clusters when the data is clustered using GMM (Figure 11). As expected, the log likelihood of the data that includes songs that are not in any clique has a higher log likelihood than the data that doesn't. This is because the features for songs that are not in any clique are diverse and have a higher variance. Thus, a higher log likelihood here means that our GMM is capturing the variance as we expect it to. We can't compare log likelihood of basic ablated features with the others because of it being in a different feature space.

However, we can see that just at 14 clusters, the log likelihood becomes constant for all the feature sets showing that features aren't discriminative enough. Ideally we should get around 4,000 clusters.

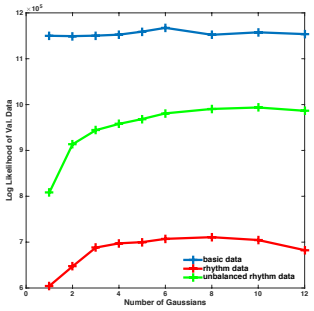

Figure 11. Log Likelihood for various number of clusters for different datasets

6. Conclusion and Future Work

The initial dataset that we used had insufficient features to generate good results. Although the models performed significantly better than the random baseline, the predictions were seldom accurate. Our modified nearest neighbors approach performed the best, followed by the neural network approaches.

Our original intention was to use this limited feature set to detect cover songs using the entire million song dataset. However, the poor performance of the algorithms on our limited set of data forced us to reconsider the validity of this approach. To develop more accurate models, we used the rhythm based enriched features and limited the amount of songs we were considering. This raised the performance of all our classifiers as expected. However, the interesting fact to note was that the performance of MLP increased by a huge margin causing it to outperform all other models. The fact that MLP performed better than a non-linear neural network shows that the features were somewhat linearly separable to form cliques.

However, we see that the general performance of all our classifiers are not spectacular. The accuracies show that the problem of matching cover songs to original ones are far from being solved. Humans are experts at recognizing cover songs and thus, there is scope for improvement. However, if we look at how humans match cover songs to the original ones, we see that they largely make use of the temporal rhythm and lyrics data of the song. Our data doesn't contain any temporal data. The features used are largely comprehensive about the entire song in general. But seeing the performance increase in the full features dataset, we observe that features like rhythm sequence, although comprehensive of the entire song, helped. This gives hope that deeper models trained on temporal audio data will have the capability to learn an embedding of rhythm sequences

without us having to hand code it in. Thus, training deep neural models with embedding similarity for pairs of cover and original songs might be an interesting direction to look into for further study.

To summarize, this project delineates the performances of various machine learning models to solve a novel problem of cover song matching. We run experiments to observe the performances of supervised and unsupervised approaches with varying features, and draw intuitions from them for aid in further study and research.

References

Battenberg, Eric and Wessel, David. Analyzing drum patterns using conditional deep belief networks. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, October 8-12 2012.

Bertin-Mahieux, T. and Ellis, D.P.W. Large-scale cover song recognition using hashed chroma landmarks. In *Proceedings of IEEE WASPAA*, New Platz, NY, 2011. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA).

Bertin-Mahieux, Thierry and Ellis, Daniel P. W. Large-scale cover song recognition using the 2d fourier transform magnitude. In Gouyon, Fabien, Herrera, Perfecto, Martins, Luis Gustavo, and Mller, Meinard (eds.), *ISMIR*, 2012.

Bertin-Mahieux, Thierry, Ellis, Daniel P.W., Whitman, Brian, and Lamere, Paul. The second hand song dataset. http://labrosa.ee.columbia.edu/millionsong/secondhand.

Bertin-Mahieux, Thierry, Ellis, Daniel P.W., Whitman, Brian, and Lamere, Paul. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

Ellis, D. P. W. and Poliner, G. E. Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2007*, 2007.

Humphrey, Eric J., Bello, Juan P., and Lecun, Yann. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, October 8-12 2012.

Jensen, Jesper Hjvang, Christensen, Mads G., and Jensen, Sren Holdt. A chroma-based tempo-insensitive distance measure for cover song identification using the 2d autocorrelation, 2008.

Ravuri, Suman V. and Ellis, Daniel P. W. Cover song detection: From high scores to general classification. In *International Conference on Acoustics, Speech, and Signal Processing*, pp. 65–68, 2010. doi: 10.1109/ICASSP.2010.5496214.

Schindler, Alexander, Mayer, Rudolf, and Rauber, Andreas. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.

Serrá, Joan, Zanin, Massimiliano, and Andrzejak, Ralph G. Cover song retrieval by cross recurrence quantification and unsupervised set detection.

Unal, Erdem, Chew, Elaine, Georgiou, Panayiotis G., and Narayanan, Shrikanth S. A perplexity based cover song matching system for short length queries. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR)*, oct 2011.

Wolkowicz, Jacek, Brooks, Stephen, and Keselj, Vlado. Midivis: Visualizing music structure via similarity matrices.