# Music Genre Verification

**Karan Hora**
Department of Computer Science
University at Buffalo, SUNY

**Nitin Nataraj**
Department of Computer Science
University at Buffalo, SUNY

**Priyanshi Shukla**
Department of Computer Science
University at Buffalo, SUNY

## Abstract

In this paper, we apply probabilistic graphical models, supervised learning algorithms and deep learning to perform genre verification on pairs of songs i.e. to check whether or not two songs belong to the same genre. The genre classification process of music has two main steps: feature extraction and classification. We apply our models on the Free Music Access dataset. We compare the accuracies and results of these models.

## 1 Introduction

Music Genre classification is an important process of music information retrieval over which a lot of music applications have been developed that work on user recommendation algorithms to suffice the needs of the users by providing recommendations according to their choice of music. In this paper, we attempt to verify if two songs belong to the same genre or not. For this purpose, the line of work includes feature extraction from music database/repositories and then following a supervised machine learning algorithm to train the model over the dataset and then testy its validity.

## 2 Related Work

Music Genre Verification is very similar to being a classification task, and hence we explore some of the related work in this area. Music genre classification is a widely studied area in Music Information Retrieval for categorizing and describing enormous amount of music. Tzanetakis et al. [1] apply Gaussian classifiers and Gaussian mixture models. They present a hierarchy of musical genres and an elaborate section on feature extraction. Yet their classification results in only 61% accuracy over ten genres. Salamon et al. [2] describe an approach using high-level melodic features for their classification. Various algorithms are compared including support vector machines, random forests, k-nearest neighbour networks and Bayesian networks. Recognition rates of over 90% are reported. This approach though requires the existence of a melody in an audio file, which is not the case for all genres. In [3], the authors proposed a music genre classification method using multilayer support vector machine learning. They used various factors such as beat spectrum, linear prediction coefficients, zero crossing rates, short time energy and mel-frequency cepstral coefficients to categorize music content. Support vector machines are developed to obtain optimal class boundaries between different kinds of music genres by learning from training data. The authors show that multi-layer support vector machines have better performance compared to traditional Euclidean distance based methods and statistical learning methods.

In [1], the authors explore automatic classification of audio signals into a hierarchy of musical genres. They propose three feature sets for representing timbral texture, rhythmic content and pitch content. Gaussian mixture model (GMM) and K-nearest neighbor (KNN) classifiers are used to obtain an accuracy of 61% (non-real time) and 44% (real time) when there are ten musical genres present.

There has been a lot of research in related to extracting representative features from music signals to heavily improve the performance of classifications. CNNs have gained prominence over the recent past due to their unparalleled success in the field of image classification. Motivated by this success, CNNs have been used for various music classification tasks such as music tagging [4, 5], genre classification tasks [6, 7] and user-item latent feature predictions for recommendations [8]. End-to-end architecture training mean that less effort is required to engineer specific features to improve performance, as the capacity of the framework is such that several hierarchical features are learned via the hidden layers of the neural network.

## 3 Dataset

The FMA (Free Music Archive) [11] dataset was used for training our models. The dataset is a dump of the Free Music Archive (FMA), an interactive library of high-quality, legal audio downloads. The FMA aims provides 917 GiB and 343 days of Creative Commons-licensed audio from 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres. It provides full-length and high-quality audio, pre-computed features, together with track- and user-level metadata, tags, and free-form text such as biographies. For the purpose of our experiments, we have considered a smaller version of this dataset - fma_small. This dataset consists of 8000 tracks, each of 30 seconds duration, coming from 8 balanced genres. This decision was made based on the availability of time and compute resources.

## 4 Implementations

The purpose of this project was to implement different classification algorithms and compare their performance. The three approaches we used for verification were Bayesian Networks, statistical machine learning (SVMs, k-nearest neighbor, Random Forest) and Deep Learning. All algorithms were implemented in Python and using relevant additional packages.

### 4.1 Bayesian Networks

The dataset used to build the Bayesian network was the echonest.csv. Each row defines one track and its characteristics including qualitative features like acousticness, danceability, energy, instrumentalness, liveness, speechiness, tempo et cetera.

As these values were continuous variables between 0 and 1, relationship between them was found by calculating the covariance between pairs of these features. The feature pairs with the highest absolute covariance were used to build the structure of the first Bayesian network PGM1 as shown in figure 1. Continuous features were converted to categorical features while constructing the Bayesian Network.

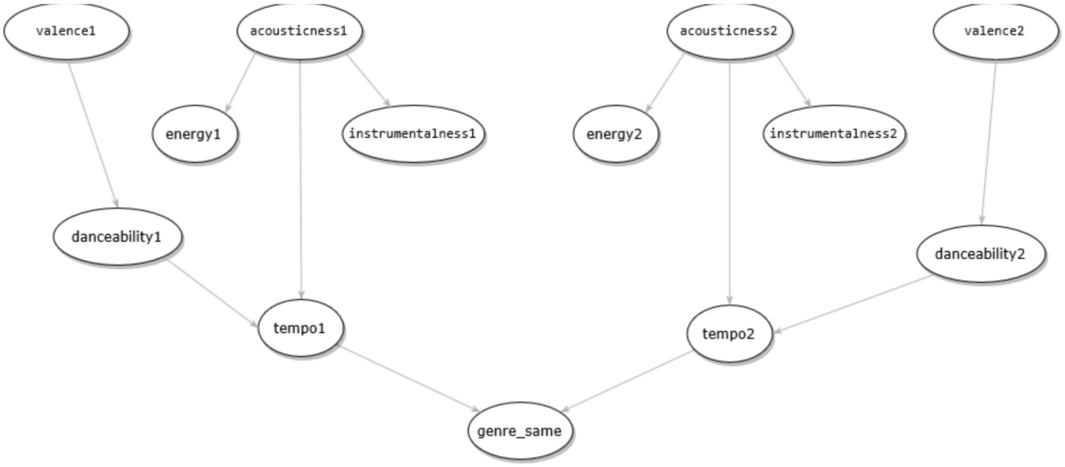

Figure 1: PGM1 structure. Here, the nodes tempo1, danceability1 et cetera define the characteristic of the first song, and songs ending with 2 define the characteristic of the second song. The 'genre_same' node indicates whether the songs have the same genre or not.

Other network structures for verification tasks were explored such as defined in Yi.Tang et al. [9]. We also created a second model using HillClimb (PGMHillClimb) approach wherein the model performs local hill climb search to estimates the Bayesian network structure that has optimal score, according to the scoring method supplied. Starts at model start and proceeds by step-by-step network modifications until a local maximum is reached. We calculate the log-loss and F1-scores for this model as well.

## 4.2 Statistical Machine Learning

We implement three different machine learning algorithms for our verification task - Linear SVM, Random Forest and K-Nearest Neighbour classifiers.

### 4.2.1 Support Vector machines

These classifiers are based on the concept of hyperplanes that differentiate decision boundaries between a set of points belonging to separate classes. The task is to find the best separator function that maximizes the margin between two separate classes. The difference is in the regularization term, which is there to make the SVM less susceptible to outliers and improve its overall generalization.

### 4.2.2 K-Nearest Neighbours

In this classifier the final output is the class membership of an object as decided by the majority vote of its K nearest neighboring objects, where K is a positive integer. Here the number of neighbors to use as default i.e. n_neighbors=10 is fed as a parameter to the KNeighborsClassifier of the Scikit- learn [10].

### 4.2.3 Random Forest Classifier

This is a classifier where a large number of decision trees is created at training time for the purpose of learning and the output is the class that is the mode of the classes i.e. mean prediction of individual trees. During construction of tree the node chosen to split no longer remains the best split, instead the split chosen is the best split among a random subset of features chosen. As a result, the bias increases but the variance decreases, thus compensating for the bias. Here the number of trees in the forest i.e. n_estimators is randomly set as 100 as a parameter fed to the RandomForestClassifier of Scikit-learn [10].

### 4.3 Deep Learning Approach

#### 4.3.1 Data Processing

The FMA dataset [11] consists of audio samples and their associated labels. For the deep learning approach, we decided to use raw audio files for feature extraction and deep learning. Although this is perfect for a genre classification task, in this paper we are trying to explore the task of genre verification, which takes two audio samples and provides an output of 0 if the genres are similar, and 1 if dissimilar. Before providing input to the Deep Learning framework, there are a few preprocessing steps that we performed.

Feature extraction methods explore ways of representing the audio samples in other vector spaces, in order to make it simpler for the classifier to learn accurate functions over the new space. The audio data was transformed from time-series data into spectrograms, which are a visual representation of the various frequencies present in the audio. Spectrograms are widely used in the fields of music and speech processing. They are commonly visualized as graphs with three dimensions, where one axis represents the time, the second represents the frequency, and possess a third dimension to represent the magnitude of the frequency component present at each point in time.

We then scale these spectrograms according to the mel-scale. The mel scale is a perceptual scale coined by Stevens, Volkmann, and Newman in 1937. It represents a scale of pitches as judged by listeners who are taken to be equal in distance from one another. These mel-scaled spectrograms are then converted into the log space and then taken as inputs to our architecture.

In order to create a new dataset for the verification task, we follow a few steps to ensure that every genre is represented with equal proportions. We first draw a stratified sample from the initial set of images, and then pair each image with every other image, and store this metadata id information in a csv file, along with the corresponding labels and similarity indicators. The melspectrogram features are loaded on-the-fly during the execution of the program, in stratified batch samples.

#### 4.3.2 Model Architecture

Deep Learning frameworks, especially those using Convolutional Neural Networks (CNNs) have become exceedingly popular in image classification tasks over the last few years. The ability of CNNs to capture increasingly informative features in a hierarchical manner has been very useful for a variety of tasks. We have transformed our time-series audio information to spectrograms, which are essentially images. Hence, in our paper, we pose our problem as that of image verification, wherein we compare two spectrograms and aim to classify them as coming from the same or different genre. Our problem has thus been reduced to a binary classification task with inputs being the mel-scaled spectrogram features, and the output being a binary variable.

In this paper, we explore the use of a Siamese neural network based architecture for image verification, as explored by Chopra et. al. [12]. Instead of using a distance metric as in their paper, we concatenate the two vectors that are obtained from the sister networks of the Siamese network, and append a fully connected layer to produce a binary output after sigmoid non-linearity. Each sister network in our Siamese architecture consists of 5 alternating convolutional and maxpool layers, with each convolutional layer equipped with identical optimization layers, namely dropout, batch normalization and ReLU (Rectified Linear Unit) activation layers. However, we did not use the batch normalization layer on our input data, as this resulted in huge memory issues that our devices were incapable of handling. The binary output is then compared with the truth label as generated by our verification transformation task in order to determine the performance metrics. We display the F1 scores and the accuracy metrics for our task. The deep learning architecture can be seen in Figure 2.

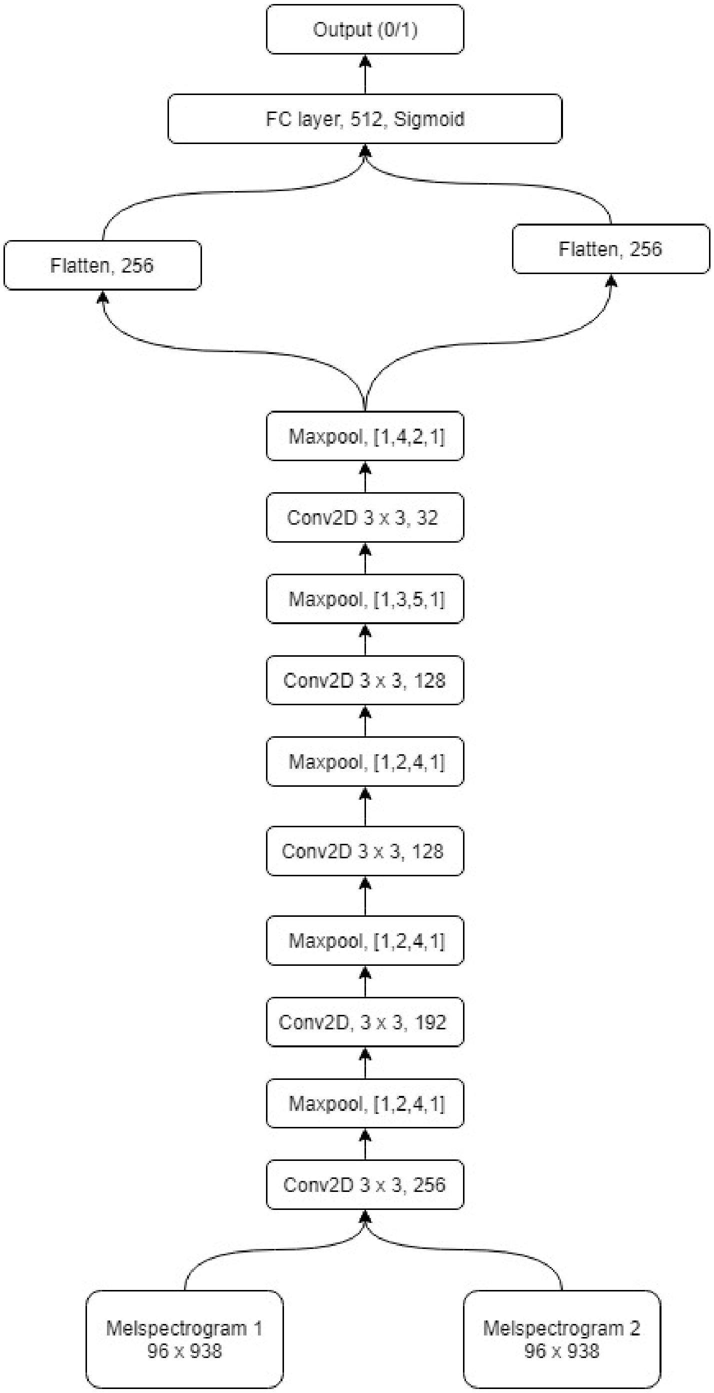

Figure 2: Model Architecture - Siamese network model architecture consisting of alternating convolutional and max pooling layers. Each layer interleaved with optimisation layers - batch normalization dropout and ReLU layers. The dropout layer keeps 50% of the neurons active during training time and all neurons active during testing time.

## 5 Results

Table 1: Results. Accuracy and F1 Scores are presented for all the classifiers, but the Log Loss, K2 and Average Log Likelihood Ratio Scores apply only to the PGM approaches.

| Method | Accuracy | F1 Score | Log Loss | K2 Score | Log-likelihood ratio |
|---|---|---|---|---|---|
| PGM-1 | 70.21% | 0.190 | 10.288 | -32382.432 | -0.102 |
| PGM-HillClimb | 72.10% | 0.228 | 9.639 | -34737.829 | 0.297 |
| KNN | 81.57% | 0.520 | - | - | |
| Linear SVM | 84.3% | 0.469 | - | - | |
| Random Forest | 92.9% | 0.801 | - | - | |
| Siamese Network | 50% | 0.667 | - | - | |

## 6 Discussions

The Bayesian network was run on knowledge-based features, the statistical machine learning approaches were run on human-engineered features and the premise for the deep learning approach was song audio files which were converted into the mel-scaled log frequency spectrum. The Random Forest classifier produced the best results, with an F1 score of 0.801 and an average accuracy of 92.9% with 1634 test samples. One key point to be mentioned here is that the datasets required by each method were markedly different from each other, and a conclusive empirical comparison cannot be conducted.

It would be interesting to see how our classification accuracies when working on the same data, and by combining the knowledge-based and human-engineered features. Lin Feng et. al. [13] explore a Paralleling Recurrent Convolutional architecture, and it would be interesting to see how that network performs on the FMA dataset, as audio data are inherently sequential and can benefit from recurrent connections. This would be something interesting to explore as future work.

### Acknowledgments

We are humbly grateful to Dr. Sargur Srihari for being a great source of inspiration and for providing us the opportunity and the guidance as we worked on this project. We would also like to thank our TAs, Mihir Chauhan and Mohammad Shaikh for their continued support.

### References

[1] G. Tzanetakis and P. R. Cook, “Musical genre classification of audio signals,” IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293–302, 2002.

[2] J. Salamon, B. Rocha, and E. Gomez, “Musical genre classification ´ using melody features extracted from polyphonic music signals,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25/03/2012 2012. [Online]. Available: files/publications/SalamonRochaGomezICASSP2012.pdf

[3] C. Xu, M. C. Maddage, X. Shao and F. Cao, “Musical genre classification using support vector machines”, Proceeding of International Conference on Acoustic, Sppech, and Signal Processing, (2003).

[4] Sander Dieleman and Benjamin Schrauwen, “End-to-end learning for music audio,” in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 6964–6968

[5] Keunwoo Choi, George Fazekas, and Mark Sandler, “Automatic tagging using deep convolutional neural networks,” in International Society of Music Information Retrieval Conference. ISMIR, 2016

[6] Siddharth Sigtia and Simon Dixon, “Improved music feature learning with deep neural networks,” in 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014

[7] Paulo Chiliguano and Gyorgy Fazekas, “Hybrid music recommender using content-based and social information,” in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016, pp. 2618–2622

[8] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen, “Deep content-based music recommendation,” in Advances in Neural Information Processing Systems, 2013, pp. 2643–2651

[9] Yi Tang, Evaluating the Probability of Identification in Forensic Science, Computer Science and Engineering 2012, Member of Technical Staff, Microsoft, Redmond, WA

[10] http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[11] https://github.com/mdeff/fma

[12] Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In CVPR, pp. 539–546, Washington, DC, USA, 2005. IEEE Computer Society.

[13] arXiv:1712.08370 [cs.SD]