Music Genre Classification of Audio Signals

Hakan Tekgul
University of Illinois Urbana-Champaign
Urbana, Illinois
tekgul2@illinois.edu

Raimi Shah
University of Illinois Urbana-Champaign
Urbana, Illinois
rsshah2@illinois.edu

ABSTRACT

With the increasing number of audio processing applications in the music industry, music genre classification became a very interesting and significant challenge for Music Information Retrieval. Because of the existence of thousands of genres from different cultures and the general subjective nature of music genres, it is very hard to create a system that can classify any type of music to its genre. In this project, we do a comparative study of audio music genre classification where we use different machine learning approaches to classify musical genres. Specifically, we analyze different machine learning algorithms such as k-Nearest Neighbor, multi-class SVMs, k-Means clustering and Gaussian Mixture Models. We additionally create a super classifier that uses the most common predictions of other classifiers. Finally, we also use PyTorch to build convolutional neural networks and perform genre classification with deep learning. We use Mel Frequency Cepstral Coefficients (MFCC) features and apply Principcal Component Analysis (PCA) to extract useful information from our dataset. Our experimental results suggest that we can achieve up to 90% accuracy for 5 musical genres.

KEYWORDS

Audio Processing, Music Genre Classification, Mel Frequency Cepstral Coefficients (MFCC), Principal Component Analysis(PCA), Neural Networks, k-Nearest Neighbors, Support Vector Machines, Clustering

ACM Reference Format:
Hakan Tekgul and Raimi Shah. 2018. Music Genre Classification of Audio Signals. In *Proceedings of Machine Learning for Signal Processing Conference (CS 598 PS Fall 18')*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the past decade, with the introduction of technology that can store huge amounts of data, a large amount of musical data is increasingly available to public on different application platforms such as Spotify. As the amount of musical data in our phones and Internet is increasing, there is a need to characterize each music track so that finding a specific song in a large archive of music would not be a problem. Musical genres are commonly used to describe and characterize songs for music information retrieval. Pachet [1] suggests that genre of music can be the best general information for music content description. Hence, a system that can classify musical genres can solve the problem of locating a specific sound track on any device.

The only problem with musical genre classification is the fact that the definition of genre is very subjective by its nature and there exists thousands of genres or sub-genres. It is also important to note that, the definition of music genre tends to change with time, as what we call rock song today is very different from the rock songs twenty years ago. Even though musical genres are subjective, there are certain features that can easily distinguish between different genres. By using features such as distribution of frequency or the number of beats, it is possible to classify main genres of music. For classification of musical genres, various approaches have been proposed. Unfortunately, most of these approaches have been proven to show accuracy around 60-70% [8]. Therefore, new approaches that can maximize classification accuracy must be considered.

Hence, we try to improve the classification accuracy of music genre classification of audio signals in this work. Specifically, we use a wide range of machine learning algorithms, including k-Nearest Neighbor (k-NN) [7], k-Means Clustering [9], Support Vector Machines [5], Gaussian Mixture Models [1] and different types of Neural Networks to classify the following 5 genres: metal, classical, blues, pop, country.

Our main goal in this study is to maximize the classification accuracy of 5 genres and compare different methods of machine learning for classification of audio signals. We use state-of-the-art machine learning platforms such as PyTorch [6] to introduce deep learning into our project. We experiment with different neural network architectures and types of neural network. Moreover, we use Mel Frequency Cepstral Coefficients (MFCC) [3] to extract useful information from musical data as recommended by past work in this field. To summarize, we make three main contributions in this paper:

- We experiment with a wide range of machine learning algorithms and state their classification accuracy for 5 different genres.
- We propose a method for feature extraction and audio processing that is dependent on both MFCC and PCA. We also discuss the significance of such methods.
- We report experimental data that describe the overall effectiveness of our classification methods by including confusion matrices.

Our experimental results suggest that Convolutional Neural Networks produced the best classification results, with around 90%, whereas k-Nearest Neighbor algorithm outputted the least accurate classification, with 81%. We observed that Support Vector Machines and Gaussian Mixture Models are also quite effective, with accuracy results around 85%. Additionally, the super classifier

CS 598 PS Fall 18', December 2018, Champaign

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

we created by combining GMM, SVM, k-NN, and a simple 3-layer neural network showed promising results, with accuracy around 90% as well. Furthermore, we found that 16 dimensions is the best reduction of our data through Principal Component Analysis.

2 RELATED WORK

The development of music genre classification has been increasing rapidly in the past decade. Many approaches have been proposed that build different models for genre classification. Some approaches concentrate on the processing of audio signals, whereas some approaches try to combine audio signals with lyrics from each musical track to increase accuracy. Some of the related work to our project is presented below.

Firstly, Tzanetakis and Cook [8] introduced different features to organize musical tracks into a genre by using k-NN and Gaussian Mixture Model (GMM) methods. Three different feature sets for speaking to tumbrel surface, rythmic substance and pitch substance of music signs were suggested. They also introduced a dataset for music genre classification (GTZAN Dataset [8]), which is widely used today in many projects, including ours.

Furthermore, Aucouturier and Pachet [1] used GMM and utilized Monte Carlo procedures for evaluation of KL divergence, which was used in a k-NN classifier. They conveyed some significant component sets for musical information retrieval that we use in our work, specifically the MFCC.

Apart from models such as GMM or k-NN, Feng [2] proposed an approach that uses Restricted Boltzmann machine algorithm to build deep belief neural networks. By generating more dataset from the original limited music tracks, he shows great improvement in the classification accuracy and describes the significance of neural networks for music genre classification.

Xing et. al. [11] proposed a similar approach that uses convolutional neural networks. By combining max and average pooling to provide more statistical information to higher neaural networks and applying residual connections, Xing et. al. [11] improved the classification accuracy on the GTZAN data set greatly. Li, Chan and Chun [4] recommend a very similar technique to concentrate musical example included in audio signals by using convolutional neural networks. They present their revelation of the perfect parameter set and best work on CNN for music genre classification.

Finally, Smaragdis and Whitman [10] presented a very interesting musical style identification scheme based on simultaneous classification of auditory and textual data. They combined musical and cultural features of audio tracks for intelligent style detection. They suggest that addition of cultural attributes in feature space improves the proper classification of acoustically dissimilar music within the same style.

As compared to these works, we do a comparative study of genre classification where we experiment with 7 different classifiers. Even though our feature extraction method is very similar to other works, not many works use PCA right after extracting MFCC features. Furthermore, to the best of our knowledge, this project is the first to create a super classifier, which is basically an ensemble method that uses 4 different classifiers and combines their predictions. Finally, in terms of classification accuracy, our experimental results are better than previous works that also used only 4 or 5 genres instead of 10.

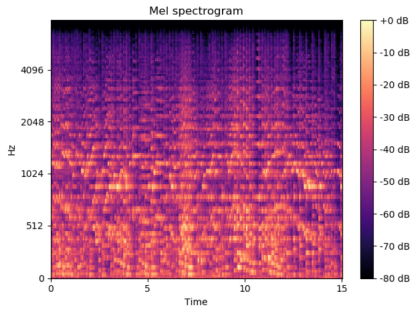

Figure 1: Mel-spectrogram of classical genre.

Specifically, there are only a few works that were able to be beat the 90% accuracy threshold. Hence, we believe our contribution to music genre clasification research is significant.

3 PROPOSED APPROACH

3.1 Musical Dataset

For musical data, Marsyas is an open source software framework for Music Information Retrieval with the GTZAN Genre Collection Database, which has 10 genres and each genre has 100 30-second audio tracks. All the tracks are 22050 Hz Mono 16-bit audio files in .au format.

For this project, we chose five distinct genres; classical, metal, blues, pop, country. Hence, our dataset was 500 songs total, from which we used 80% for training and 20% for testing. We chose five very distinct genres as previous works [4] suggest more than 5 genres can decrease accuracy a lot and introduce many problems.

3.2 Feature Extraction: Mel Frequency Cepstral Coefficients (MFCC)

Previous works [3] on music classification and processing of audio signals directed us to use MFCC (Mel Frequency Cepstral Coefficients) as a method for feature extraction so that time domain waveforms can be represented in the frequency domain in a mel-scale. For the process of MFCC, we first computed the spectrogram of each waveform by using Fast Fourier Transform and a Hamming Window. Then, we mapped each frequency to mel scale, as mel scale is the best scale for human ears. The mel spectrogram of a song from each genre is shown on Figures 1 through 5, so that the difference between each genre can be visualized. After computing the mel-spectrograms of each song, we applied discrete cosine transform (DCT) and then removed the very high frequency values from our data. At the end, we had an MFCC array of each song, where we stacked all them together, created appropriate labels for each genre and constructed our training and testing datasets. As stated, we used 80% for training and 20% for testing our classifiers.

3.3 Dimensionality Reduction with Principal Component Analysis (PCA)

After feature extraction and construction of final dataset, we thought of using dimensionality reduction before putting our data through

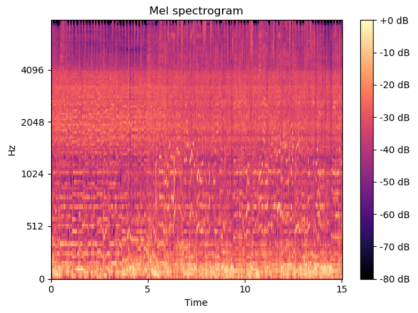

Figure 2: Mel-spectrogram of metal genre.

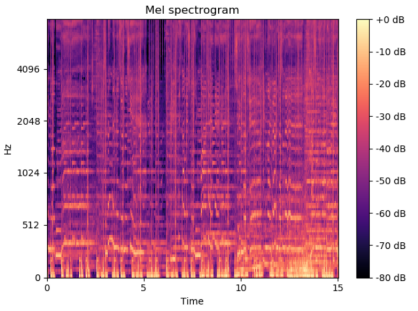

Figure 3: Mel-spectrogram of pop genre.

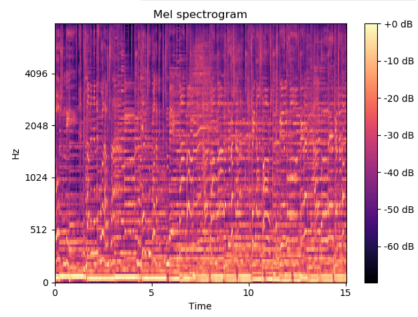

Figure 4: Mel-spectrogram of country genre.

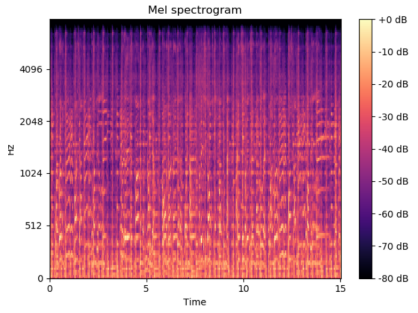

Figure 5: Mel-spectrogram of blues genre.

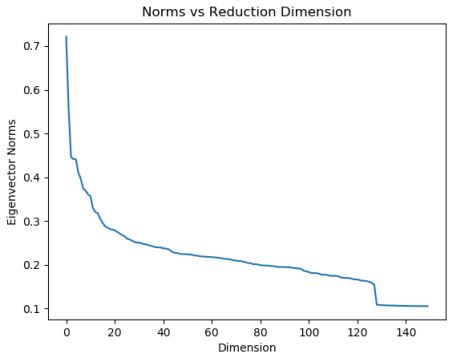

Figure 6: Norms of eigenvectors of our data plotted with respect to PCA dimensions. Note that we only want to keep the most significant components.

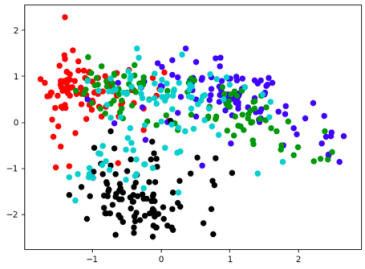

Figure 7: 2-dimensional scatter plot of our data with different genres.

classifiers. Since Principal Component Analysis (PCA) is a well-known and effective method for reduction of dimensions, we used PCA on our dataset. A realistic choice for number of reduced dimensions is to visualize the data with different PCA values and then pick the minimum dimension that can keep at least 95% of the significant components. The visualization of our data with respect to different PCA dimensions and the corresponding eigenvectors is shown on Figure 6. Note that figures 7 and 8 also present a visualization of each genre in 2 and 3 dimensions. After extensive analysis of the visualization and experimentation with our classifiers, we reduced the dimensionality to 16. Even though 16 dimensions performed very well on classifiers such as k-NN or SVM, we had to use much bigger dimensions for our neural network, since neural networks need much more data in practice.

3.4 Machine Learning Algorithms

3.4.1 ***K-Nearest Neighbor (K-NN).*** The first algorithm we used is the very famous and effective k-closest neighbors algorithm. k-NN is a non-linear algorithm that can detect direct or indirect spread of data. It is very effective for huge amounts of data. One downside of k-NN is the fact that it makes hard decisions and might produce low classification accuracy. Other than that, k-NN is computationally expensive since it does not learn any data, and it

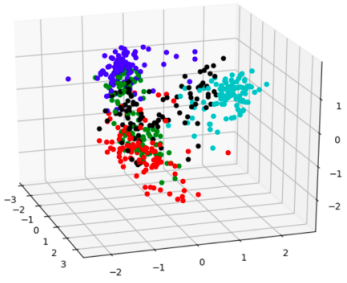

Figure 8: 3-dimensional scatter plot of our data with different genres.

has to compute the distance for every point in prediction. We used Euclidean distance for k-NN, which produced good results. After our experiments, we also found that k=5 produced the best results.

3.4.2 ***Support Vector Machines (SVM).*** The second technique we used is the support vector machine (SVM), which is a directed organization method that discovers the extreme boundary splitting two classes of information [5]. The idea behind the algorithm is to project data onto a higher dimensional space in order to separate classes in a better way. We used Radial Basis Function (RBF) for our kernel for SVM and changed the penalty to three.

3.4.3 ***K-Means Clustering.*** We attempted to use clustering by K-means which produces a hard assignment to a cluster for each data point. We initialized this algorithm randomly and assigned each point to the nearest cluster by using euclidean distance, and updated each cluster mean. This algorithm proceeds iteratively until the cluster means do not change. We found that reducing the data to 16 dimensions produced the best results. One challenge we had was evaluating clustering methods. We found that we could use Fowlkes-Mallows score to give an accuracy.

3.4.4 ***Gaussian Mixture Models (GMM).*** After K-Means clustering, we attempted to improve the clustering accuracy by implementing Gaussian Mixture Models. Each gaussian cluster has a mean and associated covariance, and each data point has a probability associated with each cluster. This gives soft assignments, which are usually better to deal with. We used sklearn's GMM class and experimented with different initialization techniques and covariance. We found that initialization with K-Means and diagonal covariance produced the best results.

3.4.5 ***Simple 3-layer Neural Network.*** After implementing different well-known classifiers, we wanted to experiment with neural networks since they generally produce promising results in machine learning applications. Firstly, we used PyTorch to process our dataset and constructed a 3-layer neural network that uses ReLU for nonlinearity. Then, we experimented with different PCA dimensions and different hidden layer sizes to produce the best accuracy. The architecture of the network that gives the best classification results is shown in Figure 9.

3.4.6 ***Convolutional Neural Network (CNN).*** We also wanted to experiment with Convolutional Neural Networks, since they can be much more effective than a simple 3-layer neural network or

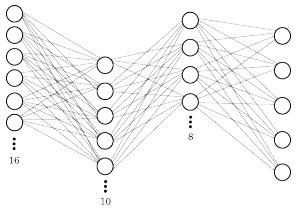

Figure 9: A summary of our final architecture of 3-layer Neural Network.

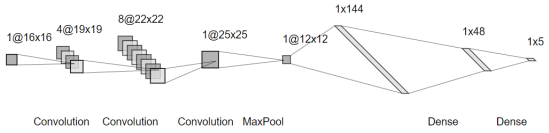

Figure 10: A summary of our final architecture of Convolutional Neural Network.

any other classifier. The downside of CNN is the fact that it requires a great deal of hyperparameter tuning. We experimented with different PCA dimensions, convolution kernel sizes, nonlinearity functions and the amount of dropout that we have to add. After testing our classifier with many different parameters, we were able to get a very good accuracy for music genre classification. The architecture that produced the best accuracy is shown on Figure 10.

3.4.7 ***Super-Classifier (SC).*** After completing a few other methods, we decided to try an ensemble method where we took 4 classifiers (GMM, k-NN, SVM, 3-layer Neural Network) and for each data point in the testing set, ran each classifier and took the most common label as the prediction. This produced better results than each of the methods individually. The motivation behind this implementation of the super classifier was the fact that each individual classifier had different least and most accurate genres.

4 EXPERIMENTAL RESULTS

4.1 Experiment Setup

The experimental setup for computation of classification accuracies were quite simple. After extracting features of our data through MFCC and applying PCA, we saved our training and testing datasets into a file, so that we do not have to do all the computations again. Then, we wrote a script that loads the training and testing datasets, and then puts the training data as an input to each of our classifiers with the corresponding genre labels. After training on each classifier, we computed predictions of each song and compared each label with its ground truth. Finally, we outputted the classification accuracy of each classifier and their confusion matrices, which are shown in Figure 11 through 18.

Table 1: Classification results of each classifier

Classifier Type	*Accuracy*	*Most Accurate Genre*	*Least Accurate Genre*
K-NN	81%	Metal	Classical
SVM	84%	Classical	Country
K-Means	83%	Classical	Country
GMM	85%	Metal	Blues
3-layer NN	88%	Classical	Pop
CNN	90%	Classical	Blues
SC	88%	Classical	Country

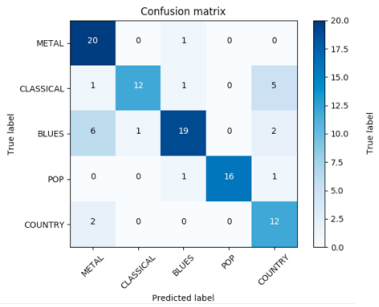

Figure 11: Confusion matrix for k-Nearest Neighbors.

4.2 Classification Accuracy

After splitting the dataset into training and testing, we ran each classifier sequentially for five times and recorded their average accuracies. After that, we also plotted the confusion matrix of each classifier and recorded each classifier's least and most accurate genre. The classification results in terms of accuracy are shown in Table 1. We can see from the table that CNN gave the highest accuracy with 90% and k-NN produced the least accuracy with 81%. Our results show that combining different classifiers' predictions increased the general accuracy to 88%. Other than that, the results show that metal and classical genres were the easiest to classify compared to other musical genres, whereas country music was the hardest one to classify correctly. Furthermore, it is also significant to state that Neural Networks were dominant in terms of classification accuracy and they produced very promising results for audio signal classification, as expected. The training losses of neural networks are shown in Figure 19 and Figure 20. It is also fair to state that Convolutional Neural Networks did not perform as well as we expected, since we can instead use a 3-layer Neural Network or a super classifier as well and get an accuracy close to 90%.

5 CONCLUSION

In conclusion, we implemented 7 different classifiers for music genre classification and it is fair to state that all the classifiers performed well, which is expected since we used 5 distinctive genres. It is observed that simple and easy to implement approaches such as k-NN and k-Means did worse than other more complicated classifiers such as SVMs or Neural Networks. Our goal was to achieve accuracy around 90% and we were able to accomplish that goal with Convolutional Neural Networks. We also observed that it was easy

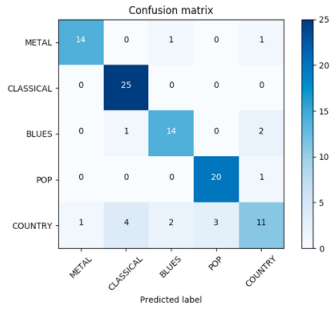

Figure 12: Confusion matrix for Support Vector Machines.

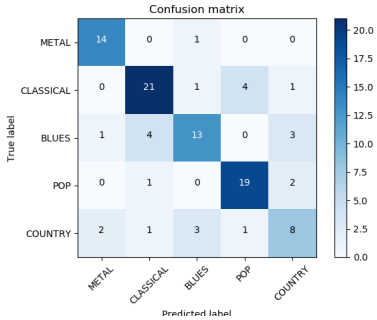

Figure 13: Confusion matrix for K-Means Clustering.

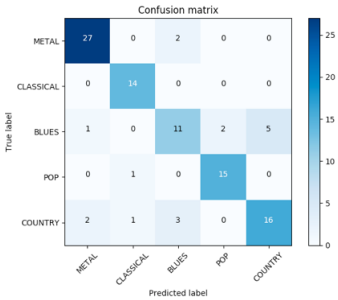

Figure 14: Confusion matrix for Gaussian Mixture Models.

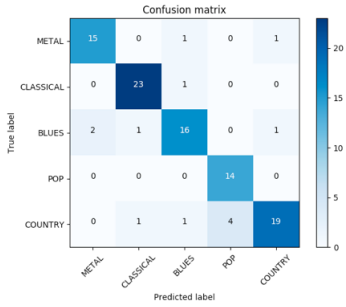

Figure 15: Confusion matrix for 3-layer Neural Network.

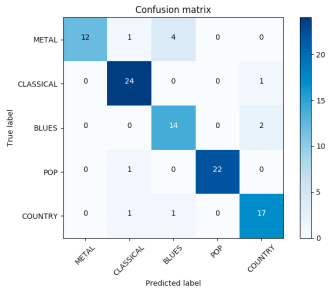

Figure 16: Confusion matrix for Convolutional Neural Network.

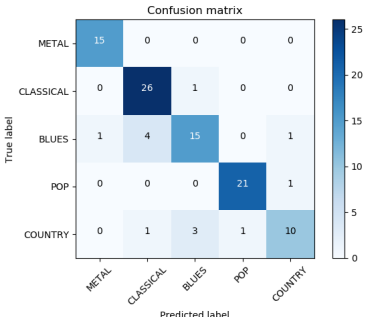

Figure 17: Confusion matrix for Super Classifier.

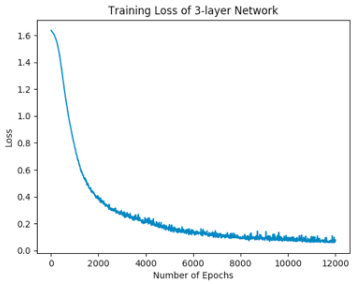

Figure 18: Training loss vs. number of epochs for 3-layer neural network.

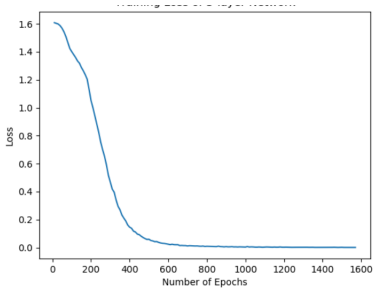

Figure 19: Training loss vs. number of epochs for Convolutional Neural Network.

to beat 80% accuracy for 5 genres of music and the easiest genres to classify or distingush were classical and metal. Other than that, most of the classifiers had problems with classifying country or blues music.

REFERENCES

[1] Jean-Julien Aucouturier. 2003. Representing Musical Genre: A State of the Art. *Journal of New Music Research* 32 (03 2003), 83–93. https://doi.org/10.1076/jnmr.32.1.83.16801

[2] Tao Feng. 2016. Deep learning for music genre classification. *Pattern Recognition Class Paper* (2016).

[3] Z. Fu, G. Lu, K. M. Ting, and D. Zhang. 2011. A Survey of Audio-Based Music Classification and Annotation. *IEEE Transactions on Multimedia* 13, 2 (April 2011), 303–319. https://doi.org/10.1109/TMM.2010.2098858

[4] Tom L. H. Li, Antoni B. Chan, and Andy HW. Chun. 2010. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network.

[5] Michael I. Mandel, Graham E. Poliner, and Daniel P. W. Ellis. 2006. Support vector machine active learning for music retrieval. *Multimedia Systems* 12 (2006), 3–13.

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[7] L. E. Peterson. 2009. K-nearest neighbor. *Scholarpedia* 4, 2 (2009), 1883. https://doi.org/10.4249/scholarpedia.1883 revision #137311.

[8] G. Tzanetakis and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10, 5 (July 2002), 293–302. https://doi.org/10.1109/TSA.2002.800560

[9] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 577–584. http://dl.acm.org/citation.cfm?id=645530.655669

[10] Brian Whitman and Paris Smaragdis. 2002. Combining Musical and Cultural Features for Intelligent Style Detection. In *ISMIR*.

[11] Weibin Zhang, Wenkang Lei, Xiangmin Xu, and Xiaofeng Xing. 2016. Improved Music Genre Classification with Convolutional Neural Networks. In *INTERSPEECH*.