Clustering Music by Genres Using Supervised and Unsupervised Algorithms

Kyuwon Kim, Wonjin Yun, and Rick Kim

***Abstract*—This report describes classification methods that recognize the genres of music using both supervised and unsupervised learning techniques. The five genres, classical(C), EDM(E), hip-hop(H), jazz(J) and rock(R), were examined and classified. As a feature selection method, discrete Fourier transform (DFT) converted the raw wave signals of each song into the signal amplitude ordered by their frequencies. Based on the analysis of the characteristics of data set, final feature set were collected by averaging the amplitudes of corresponding two different frequency division (X_L and X_M). For supervised learning, a training set ($m_{train} = 50/genre$) was used to train the CART (Classification and Regression Tree), and the performance of the genre prediction by CART classifier was evaluated using a test set ($m_{test} = 10/genre$). A recognition rate of 86.7% for three genre classification (C, H, and R) was observed, and 60.7% for five genre classification (C, E, H, J, and R) was obtained. For unsupervised learning algorithm, K-means clustering was performed on an unlabeled set of data ($m = 60/genre$) to cluster the music into genres, and showed purity of 84.4% for three genre classification, and 62.0% for five genre classification.**

***Index Terms*—Music clustering, K-means, CART**

I. Introduction

Most music websites have song recommender systems. There are two commonly used mechanisms in most websites: collaborative filtering and content-based filtering. In collaborative filtering, websites gather information about their users' behavior and preferences to recommend similar items for a user based on peer users' choices. On the other hand, in content-based filtering, keywords of music such as year, genre and musician are required to predict similar songs. While these mechanisms are effective in recommending similar songs for a user, both mechanisms require a large amount of external data. Collaborative filtering needs sufficient peer users' data, and content-based filtering requires pre-processed labels on the music. Here, we explore methods to cluster similar songs only using their raw wave files.

A genre is a label often used to group similar types of music. Although genres of music are usually determined manually by human, we attempt to predict genres of music without any pre-processed labels or peer choices. In this paper, we have tried to construct models that cluster songs based on their components of raw wave signals and test how well these models can predict the genres determined by human. Given an unknown music file, we aim to implement automatic classification by genres using machine learning techniques.

Both supervised and unsupervised learning methods were tested in our study. In the supervised learning test, we used Classification And Regression Tree (CART) to find a model that can effectively classify a new piece of music using some features from the wave signals. In the unsupervised learning test, we used k-means clustering to cluster a set of unlabeled songs into groups of similar features. Confusion matrices and the purity of classification were used to evaluate the performance of the clustering.

II. Related Works

There have been a number of studies on predicting the genre of music using features in the wave signals of a music file. The temporal structure of a music piece was often used to recognize the genre of a music. Soltau *et al.* used the sequence of activation in hidden units and a neural network to classify rock, pop, techno and classical[1]. Shao *et al.* used a hidden Markov model with sequences of features such as Mel-frequency cepstral coefficients(MFCCs) to cluster pop, country, jazz and classical [4].

Some approaches focused on the direct similarity between the feature sets than the temporal structures. Cilibrasi *et al.* used a distance function between feature vectors and generated trees by the distances to visualize the similarity between samples of classical, rock and jazz[3]. Peng *et al.* used features from the signals to perform a k-means clustering, and used some labels as constraints to perform a constraint-based clustering to group a set of songs by their artists.[5]. Tsai *et al.* used a hierarchial agglomerative clustering method which sequentially merges similar pieces of music[6].

MFCCs were commonly used in many studies, because the Mel-scale represents the perceptual scale of pitches for a person's hearing. However, some studies also introduced new features such as linear prediction coefficients(LPCs)[4] and Renyi Entropy Cepstral Coefficients(RECCs)[6]. Short-term Fourier Transform features were also used to capture the timbral textures from wave signals[5].

III. Methodology

A. Data and Feature selection

Five different genres of music were chosen as our data classes: classical, EDM, hip-hop, jazz and rock. 60 samples from each genre were randomly streamed from YouTube. Each song was sampled using a sampling rate of 44.1kHz, and stereo wave signals were merged to mono wave signals. The whole data set ($m = 60/genre$) was used without labels to test the unsupervised learning algorithm. For the supervised learning

K.Kim is with the Department of Mechanical Engineering, Stanford University.

W.Yun is with the Department of Energy Resource Engineering, Stanford University.

R.Kim is with the Department of Biology, Stanford University.

algorithm, 10 samples from each genre were randomly chosen as our test set ($m_{test} = 10/genre$), and the remaining data were used as our training set ($m_{train} = 50/genre$).

To collect features from each sample, Discrete Fourier Transform (DFT) was performed on each sample. An array of frequencies $\{f_j\}_j$ were defined to divide our frequency range of interest into frequency bands of $B_j = [f_j, f_{j+1}]$. Given an array $\{f_j\}_j$ with length $n+1$, and a raw wave data $w^{(i)}$, our feature vector $x^{(i)} \in R^n$ is defined as the following :

$$\hat{x}_j^{(i)} = \int h_j(s) DFT(w^{(i)}) ds \tag{1}$$

$$x^{(i)} = \hat{x}^{(i)} / ||\hat{x}^{(i)}||_2 \tag{2}$$

$$h_j(s) = \begin{cases} 1 & \text{if } s \in B_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Each feature vector was normalized to satisfy $||x^{(i)}|| = 1$. Two different feature sets X_L and X_M were used for our test as illustrated in Fig. 1. Initially, we generated 10 frequency bands in ($0 \sim 5000$ Hz), and the contribution of each feature to classification of the songs was assessed by CART as shown in Section. III-B and Table. I. Since the lower frequency region was found to show more impact on determining the genres in our preliminary results, X_L uses the average values of 20 low frequency bands ($0 \sim 200$Hz) with bandwidth of 10 Hz, and 7 mid/high frequency bands ($200 \sim 5000$ Hz) with bandwidth of $100 \sim 1000$ Hz. On the other hand, X_M uses the average values of a Mel-scale frequency division for $20 \sim 2000$ Hz (20 bands).

B. Supervised Learning

CART, a conceptually simple yet powerful method, was used to perform a recursive partitioning. In this process, our data set was split into two regions R_1 and R_2 (first and second music genre for our study) and the splitting process was repeated until the stopping rule is applied. The CART algorithm automatically decided the splitting variables j and split points s until it found the best binary partition. A greedy algorithm efficiently and quickly determines the best pair (j, s) [7]. In the greedy algorithm [7], the pair of half-planes is defined as in Eq. 4. The splitting variable and the split point are solved by Eq. 5 and the the inner minimization is solved by Eq. 6.

$$R_1(j,s) = \Big\{ X \mid X_j \le s \Big\} \; ; \; R_2(j,s) = \Big\{ X \mid X_j > s \Big\} \tag{4}$$

$$\arg\min_{j,s} \left[\min_{c_1,c_2} \left(\sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right) \right] \tag{5}$$

$$\hat{c_1} = E\big(y_i \mid x_i \in R_1(j,s)\big) \; ; \; \hat{c_2} = E\big(y_i \mid x_i \in R_2(j,s)\big) \tag{6}$$

For our study, CART (classification tree algorithm rather than regression tree) was applied to prioritize a number of features among the entire feature space shown in Table.I.

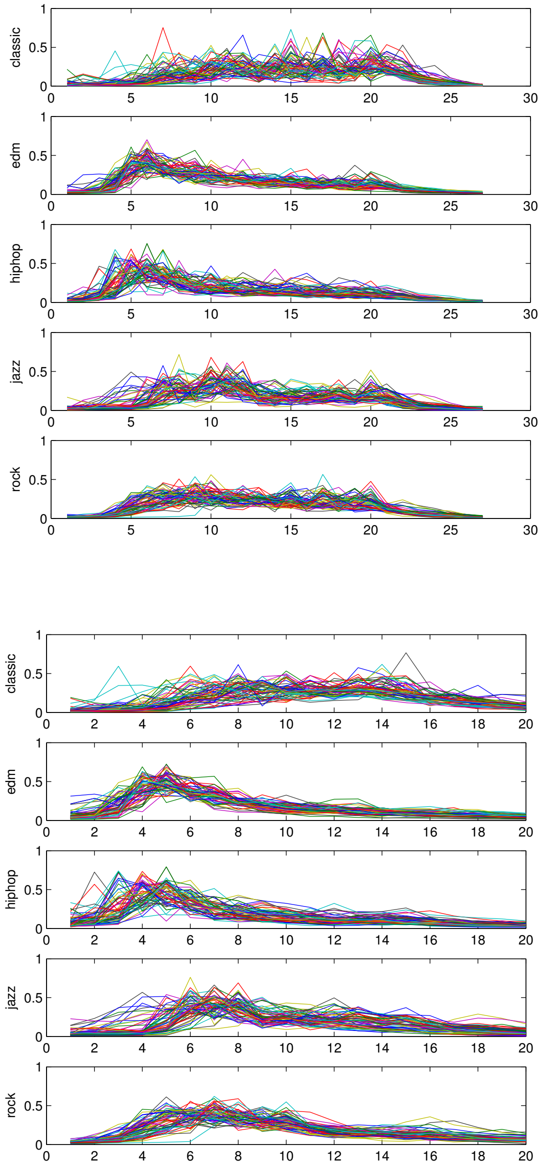

Figure 1. Features X_L (upper) and X_M (lower) sampled from 5 genres : classical, EDM, hip-hop, jazz and rock.

For supervised learing, a CART classifier model was trained on a balanced training data set ($m_{training} = 50/genre$). A built-in algorithm, *rpart* [8], was used in R software to perform CART and the optimal classification tree size was adaptively chosen from the data by *cost-complexity pruning* [8] using the node impurity, $Q_m(T) = 1 - \hat{p}_{mk(m)}$ described in Eq. 7.

$$\hat{p}_{mk(m)} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k(m)) \tag{7}$$

where $\hat{p}_{mk}$ is the proportion of class k (genre for our study) observation in node m.

The recursive binary tree is illustrated in Fig. 2 that fully describes the feature space partition corresponding to the partitioning of the songs in the regions R_m (three or five genres for our study).

Table I
CART FEATURE IMPORTANCE

10 features (Hz)	Feature	2nd (50-100)	1st (0-50)	3th (100-200)
	Importance	26	20	19

Table 1 : Classification tree (CART) trained on 90 pre-labeled songs with 10 features ranged from 0 Hz to 5000 Hz shows that 46 songs among 90 songs were recognized by the first two features (frequency band from 0 to 100 Hz).

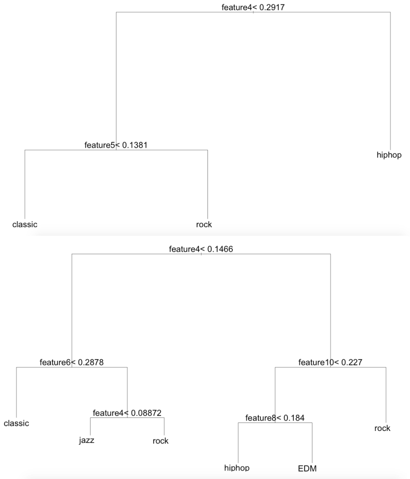

Figure 2. CART for 3 (Upper:$m_{train} = 150$)) genres and 5 (Lower:$m_{train} = 250$) genres are demonstrated using X_L.

C. Unsupervised Learning

K-means clustering was performed on our data set to cluster the samples. One sample from each genre were randomly chosen as our initial pivots. Prior to k-means, PCA was performed to extract the significant components from our feature sets. Only the top 10 components were used to run the clustering. Fig. 3 and Fig. 4 show the data distribution in top 3 principal components. The test was performed for i) a three genre classification (classical, hip-hop, rock), and ii) a five genre classification. The whole data set ($m = 60$/genre) was used for the classification. To evaluate the performance of our classification, the *purity* and *Rand Index*(RI) were used. The purity of a clustering represents how homogeneous each cluster is in average, and RI gives us the accuracy of the classifier when any two random samples are chosen from the data set and compared.

IV. RESULTS

A. Supervised Learning

CART was performed for a three genre classification test and a five genre classification test, using feature sets X_L and X_M. In a 3-genre classification, the classifier showed recognition rates of 77.2% ($\sigma = 7.42$) with X_L and 86.7% ($\sigma = 4.27$) with X_M, displaying a significantly better performance for X_M. On a 5-genre classification, the recognition rates declined to 54.7% ($\sigma = 6.02$) and 60.7% ($\sigma = 4.32$) respectively, also displaying a better performance for X_M.

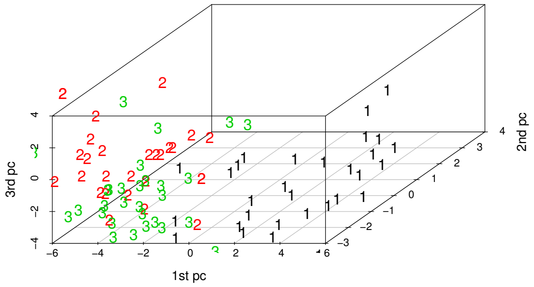

Figure 3. PCA result for three genres using feature set X_M. PCA plot from 3 genres effectively visualizes the distance between songs. In the plot, first two principal components efficiently classifies classical (1) from the other genres, and third principal components separate the hip-hop(2) from rock (3).

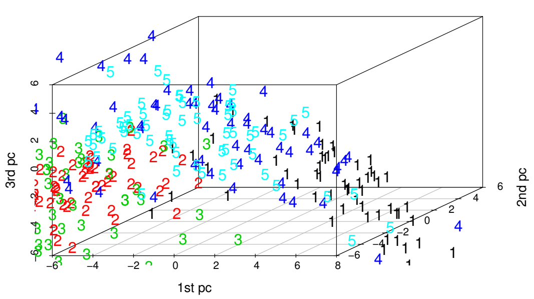

Figure 4. PCA result for five genres using feature set X_M. PCA plot from 5 genres effectively visualizes the similarity and difference between songs. The plot exhibits the genre similarity between EDM (2) and hip-hop (3); jazz (4) and rock (5). And, it can be noted that the plot demonstrates the widest span of jazz (4) compared to the highly-localized classical (1).

B. Unsupervised Learning

The purity for a three genre classification was 0.822 with feature set X_L and 0.844 with X_M. There was a significant drop in the purity when the number of genres increased from 3 to 5, showing 0.620 with X_L and 0.597 with X_M, resulting in -0.224 decline in average. It also shows that feature set X_M was more effective for classifying the three genres, while X_L appeared to have a better performance when the number of genres increased.

Meanwhile, the RI for a three genre classification was 0.790 with X_L and 0.817 with X_M. It also did not show much decline (-0.033 in average) when the number of genres increased, resulting in 0.767 for X_L and 0.774 for X_M.

These values could not be directly compared with the results from other studies since the data set and the chosen genres were different. Also, some studies [6] use different ways to define the purity and RI. Although we belive our results have shown satisfying performance, there are possibilities to improve the performance.

C. Confusion between genres

Both learning algorithms tried performed significantly better with a 3-genre classification. In Table. III, the confusion matrix

Table II
CONFUSION MATRICES FOR THREE GENRE CLASSIFICATION

(a) Feature set X_L

	C_1	C_2	C_3
Classical	**45**	0	15
Hip-hop	0	**49**	11
Rock	5	1	**54**

(b) Feature set X_M

	C_1	C_2	C_3
Classical	**50**	0	10
Hip-hop	0	**51**	9
Rock	5	4	**51**

Table II : Clustering results for a three genre classification using feature sets X_L (left) and X_M (right). The purity and RI are (a) 0.822, 0.790 and (b) 0.844, 0.817 respectively.

Table III
CONFUSION MATRICES FOR FIVE GENRE CLASSIFICATION

(a) Feature set X_L

	C_1	C_2	C_3	C_4	C_5
Classical	**37**	0	0	9	14
EDM	0	**51**	7	0	2
Hip-hop	0	30	**26**	0	4
Jazz	3	7	1	**39**	10
Rock	3	16	0	8	**33**

(b) Feature set X_M

	C_1	C_2	C_3	C_4	C_5
Classical	**47**	0	2	9	2
EDM	0	**48**	5	0	7
Hip-hop	0	35	**17**	0	8
Jazz	10	2	2	**23**	23
Rock	4	6	0	6	**44**

Table III : Clustering results for a five genre classification using feature sets X_L (upper) and X_M (lower). The purity and RI are (a) 0.620, 0.767 and (b) 0.597, 0.774 respectively.

shows that EDM was frequently confused with hip-hop, while jazz was often mistaken as rock. The PCA result in Fig. 4 also demonstrates how these genres are closely distributed on the principal component space. This shows that some genres can be closer than other genres in our feature set.

In order to examine the distance between genres, we constructed a neighbor graph based on the L2-norm distances of feature set X_M in Fig. 5. For each sample, only the seven most nearest samples were considered as neighbors. Proximity scores were calculated using the graph, which gives a higher score if the two samples are closely related to each other. The proximity score of two samples $x^{(i)}, x^{(j)}$ were calculated as the following :

$$s_{ij} \quad = \quad \sum_{k \neq i,j} \exp(-||x^{(i)} - x^{(k)}|| - ||x^{(j)} - x^{(k)}||) \quad (8)$$

High score implies that the two data has many common neighbors and are likely to be classified as same genre. Fig. 6 shows a heat map of the proximity scores, which visualize the similarity between genres clearly. It shows a close relationship between EDM and hip-hop, while classic does not show much relationship with other genres.

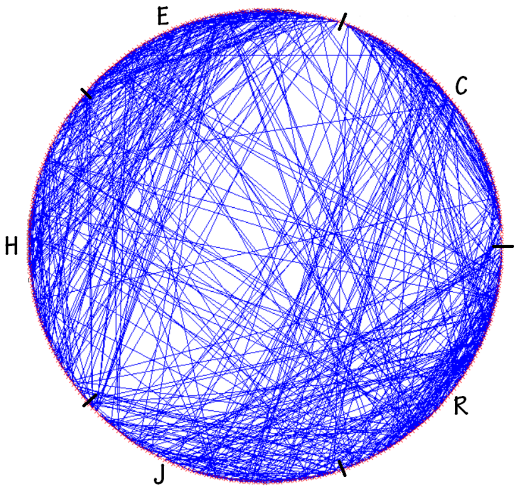

Figure 5. Graph showing the proximity between songs with 7 nearest neighbors. Each red dot represents a song, each blue edge represents a connection between two neighbors. Abbreviations: C is classical, E is EDM, H is hip-hop, J is jazz, R is rock

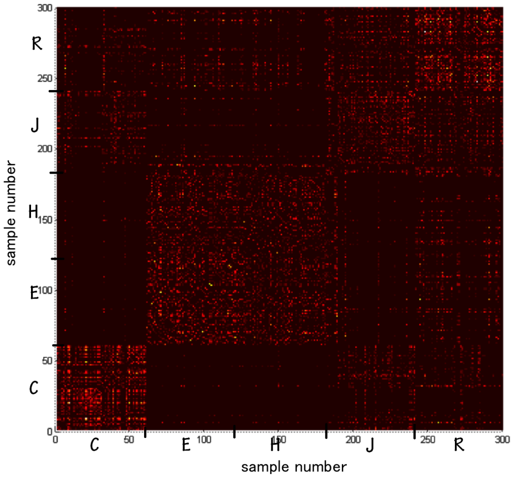

Figure 6. The heatmap of a proximity matrix between 300 songs of 5 genres in order - classic, EDM, hip-hop, jazz and rock. Abbreviations: C is classical, E is EDM, H is hip-hop, J is jazz, R is rock

There are a number of reasons why the models can not distinguish EDM and Jazz from hip-hop and rock, respectively . One key reason is because of the limited features. The features explored in this paper may not capture the key differences between those genres. In addition, our input data were selected randomly from Youtube. Our selection might include songs that are not representative of the designated genres. Some samples might be a mixture of different genres, in which the classification of genres are ambiguous since some songs can be classified as multiple genres. With a standard data set, we would expect the performance to improve.

V. CONCLUSION

In this report, we have explored methods to cluster sets of music by DFT data extracted from raw wave signals. The most challenging part was the feature selection. The DFT data were decomposed into two sets of frequency bands (X_L and X_M) within the frequency range (0-5000 Hz) and (0-2000Hz), considering the range of frequencies used by most music instruments in songs. Two different learning algorithms, CART and k-means clustering, were studied. Using CART allowed us to examine what features are most influential in making decisions as described in section III-B. PCA was also performed on the feature sets to extract the principal components.

For supervised learning, CART showed a recognition rate of 86.7% for 3 genres and 60.7% for 5 genres at maximum. Results from k-means clustering showed a purity of 0.844 for 3 genres and 0.620 for 5 genres. This indicates that the performance is better with a small number of genres than with a large number of genres. In contrast, the RI did not drop significantly.

The selection of features displayed differences in the performance of clustering, and proved to be an important factor for the effective classification of genres. Only features derived from the DFT of signals were used in our project, but using other features could improve the perfomance of our classifier. Other potential features to use include the average tempo of a music, repeated patterns in the music structure, and measures that can capture timbral textures of different instruments. A better selection of our data set could also improve our performance. The songs we used were randomly streamed by Youtube, and were not verified by specialists. Thus, it is possible that some of the songs may be hybrid of different genres. Also, by using a standard set of data that is commonly used to test music clustering, we would be able to compare our performance with other previous methods.

References

[1] Soltau, H., Schultz, T., Westphal, M., Waibel, A. (1998). *Recognition of music types.* In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference* on (Vol. 2, pp. 1137-1140). IEEE.

[2] Tzanetakis, G., Cook, P. (2002). *Musical genre classification of audio signals.* In *Speech and Audio Processing, IEEE transactions* on (Vol. 10, no. 5, pp. 293-302).

[3] Cilibrasi, R., Vitányi, P., De Wolf, R. (2004). *Algorithmic clustering of music based on string compression.* In *Computer Music Journal* on (Vol.28, no. 4, pp. 49-67). IEEE.

[4] Shao, X., Xu, C., Kankanhalli, M. S. (2004). *Unsupervised classification of music genre using hidden markov model.* In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference* on (Vol. 3, pp. 2023-2026). IEEE.

[5] Peng, W., Li, T., Ogihara, M. (2007). *Music Clustering with Constraints.* In *ISMIR* (pp. 27-32).

[6] Tsai, W.H., Bao, D.F. (2010). *Clustering music recordings based on genres.* In *Information Science and Applications (ICISA), 2010 International Conference* on (pp. 1-5). IEEE.

[7] Hastie, T., Tibshirani, R., Friedman, J. ”Additive Models, Trees, and Related Methods,” *The Elements of Statistical Learning*,2nd ed. New York, USA: Springer, 2009, ch.9 ,sec. 2 ,pp. 305–310.

[8] Breiman L., Friedman J. H., Olshen R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.