Implementation of Music Genre Classifier Using KNN Algorithm

Xuchuan Mu *

Department of Computer Science, Queen'sUniversity, Kingston, Ontario, K7L 3N6, Canada

* Corresponding Author Email: 18xm24@queensu.ca

Abstract. As music history grew, music began to diversify into different genres. Thisstudy aims to implement a music genre classifier using the KNN algorithm and a faster method. The KNN algorithm is accurate but with long execution time. This study implements a new method that can speed up the process of the KNN algorithm, and the K-means clustering algorithm inspires the idea. The dataset is preprocessed using the new idea. The program will select the song that is the centroid of the genre and use the method of the KNN to return the closest genre based on the distance from the test sample to the centroid. In conclusion, the new method did not perform well in accuracy but sped up the program. This study provides a great reference for the music genre classification problem in the machine learning domain. The study investigates an infeasible method in preprocessing data for the KNN algorithm optimization.

Keywords: KNN algorithm, K-clustering, GMM, Music genre classification.

1. Introduction

Music is one of the essential parts of human history. With the growth of the modern world, the music genre's diversity has also become more extensive. The genre of music is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions [1]. Music developed to this day has become different kinds of genres, very familiar genres such as country music, electronic music, and rock music. Although each genre has its strict definition, it is possible for one song that combines multiple elements from different genres of music. Music lovers use techniques and their ears to identify the distinction between songs and classify them based on that. The internet is growing fast, and multiple online music players have appeared these days; using manual force to distinguish music genres would be redundant with high rate of mistakes. Therefore, using machine learning to implement music genre classification became a new concept for most music player platforms such as Spotify.

Since this topic's challenge was published, different approaches using machine learning have been exploited. Four common approaches used nowadays are:

(1) Support Vector Machines (SVM).

(2) K-means clustering.

(3) K-nearest neighbors.

(4) Convolutional neural networks (CNN).

In this study of music genre classification, the K-nearest neighbors algorithm is used as the basic logic of the classifier. The K-nearest neighbors algorithm is a non-parametric supervised machine learning algorithm [2]. This kind of algorithm is used to solve classification and regression problems. The basic logic for the KNN algorithm is as follows: when entering the training set and test set, first, the features of the training set and test set need to be extracted as the characteristics for each data. The algorithm will calculate each distance between the test data with the entire training set. Next, the distances are sorted, and the k nearest data is selected. Finally, the corresponding class of the data is obtained. In general, it is a sufficient but lazy algorithm that doesn't perform any training until it receives the training sets.

The biggest obstacle to making the KNN algorithm have a long execution time is the dataset. For each time a classification is predicted for a test sample, the algorithm will have to iterate the entire dataset to return the k nearest samples. Therefore, if the test set contains many samples, the algorithm

will become slow and inaccurate. To have a faster classifier, the key is to preprocess the dataset. The idea of data preprocessing is inspired by the process of the k-means clustering algorithm. The basic logic for the k-means clustering algorithm is to divide the sample space into k multiple clusters. The selection of the clusters is based on the centroid or the mean of the cluster. The program uses similar logic to the k-means algorithm to select one centroid for each genre in the dataset. Then each cluster can be treated as one spherical region with the furthest data point as the radius. Therefore, the program only needs to compute the closest distance from the test sample to the cluster and return the corresponding class.

2. Methodology

2.1. Data

The dataset based on this study on music genre classification is the GTZAN dataset downloaded from Kaggle [3]. It is a collection of 10 genres of songs with 100 songs in each genre. Each song is a 30-second segment stored as wav file. The dataset iteratively reads from the file. Each iteration will standardize the current data and store the mean value and covariance file called my.dat. This purpose is to make the latter calculation of the distance to be concise.

2.2. Extraction Method

One of the most necessary steps is to use a method that can extract the feature of one audio segment. Standard techniques of acoustic feature extraction like MFCC, LPC, and PLP are popular in speech recognition. In this project, the Mel-Frequency Cepstral Coefficients are used to extract features of the GTZAN dataset. It is a spectral-based parameter used in audio recognition [4]. In the project, the implementation of MFCC is to call the embedded python speech features library. The process of MFCC in this project is as follows:

(1) The audio file divides into small frames with lengths of 20ms.

(2) Fast Fourier Transformation (FFT) applies to each frame for identifying different frequencies.

(3) Discrete Cosine Transformation (DCT) is applied where it sorts the coefficients extracted from the last step according to significance [5]. The least significant coefficients are eliminated to achieve noise discarding.

2.3. Gaussian Mixture Model and Distance Measurement

The form of the Gaussian Mixture Model is used to represent each audio file as one of the data points in the database. This form of Gaussian distribution is commonly used to represent an acoustic speech segment, a standard technique when we want to measure the distances on the probabilistic representation [6].

The measurement between two data points is a widely used calculation in the KNN algorithm. Choosing an appropriate distance measurement method to compute the distance between two GMM is critical in speaker verification. In math, there are three ways to measure the distance between two Gaussian distributions: the Mahalanobis distance, the Bhattacharyya distance, and the Kullback-Leiber divergence. In this study, the Kullback-Leiber divergence is used as the primary logic for calculating two GMM. The KL divergence is a method to measure the difference between one probability distribution with the other. It uses the mean and covariance transformed from the extracted data and applies Formula1[7].

$$D_{KL} = \frac{1}{2}\left[log\frac{|\Sigma_2|}{|\Sigma_1|} - k + (\mu_1 - \mu_2)^T\Sigma_2^{-1}(\mu_1 - \mu_2) + tr\{\Sigma_2^{-1}\Sigma_1\}\right] \quad (1)$$

2.4. K-Nearest Neighbor Algorithm

The K-nearest neighbors algorithm is a non-parametric supervised machine learning algorithm [2].This kind of algorithm is used to solve classification and regression problems. It is a sufficient but lazy algorithm that does not perform any training until it receives the training sets [8].

The KNN algorithm is the basic logic of our implementation. It can be performed through the following steps:

(1) We have a training set of A which A = {a_1, a_2, a_3, …., a_n} with its corresponding classes C = {c_1, c_2, c_3, …., c_s} and a test set of T which T = {t_1, t_2, t_3, …, tn}.

(2) We want the initial k neighbors in the training set that have the closest distance to the test set.

(3) In the program, we iterate the entire set of A for each iterate of the test set C. For one iteration the program calculates the KL divergence between each element a_i for 1<= i<= n with the one element from the test set CJ, 1<= j <= n.

(4) The distance will be stored with the corresponding class in a list. The list is sorted in descending form and return the first k neighbors.

(5) Each neighbor inside the k neighbors list will vote for their own classes, and the class with the most votes will be returned as the final output.

(6) The program will go back to step 3 for looping.

2.5. Data Preprocessing

The basic logic of data preprocessing is inspired by Wang et al. [9]. The key to improve the KNN algorithm is to preprocess the dataset. In the study of Wang et al., a method of faster K-mean clustering algorithm based on using probability to initialize the centroid of the data is proposed with a new name of pk-mean++ clustering. Using pk-mean++ to divide the dataset into multiple spherical regions with their corresponding centroid and radius. The preprocessed dataset will then be used as the input dataset for the KNN algorithm.

In this study, the idea from the above is applied to music genre classification. The process of data preprocessing is like the K-means clustering but with several differences. The K-means clustering is a method to partition one dataset into k clusters, with each cluster having a centroid.Here is the basic idea: first, the program will cluster the samples into k non-empty clusters randomly. These clusters will cover the entire sample space. The centroids for each cluster are computed. Each sample will be assigned with the newest centroid to form the new clusters. The process will go back to step 2 and repeat until the centroids for each cluster don't have a modification or reach the maximum iteration number.

The disadvantages of conventional K-means clustering are that the outcome will be largely impacted by the initial selection of centroid, which is random. Poor initial selection will lead to the poor partition of the samples [10].The other disadvantage is the number of initial clusters to choose from. In this study of music genre classification, this disadvantage can be neglected. The GTZAN dataset used in this study contains a partitioned dataset which means that the dataset is clustered with an exact number of clusters. Each genre forms a corresponding cluster for itself. Therefore, the key to preprocessing GTZAN is to find the centroid data for each one of the genres. The implementation is presented as follows:

(1)The program calculates the distance from one point to each point inside one genre. Sum all the distances.

(2)Each point will have a sum of distances to every other points.

(3)the program returns the point with the lowest sum of distance and save as the centroid of the cluster.

(4)The radius will be the distance from the centroid to the furthest point.

2.6. Improved distance measurement

After the preprocessing part, the centroid data and corresponding radius for each cluster are used as input data in the KNN algorithm. Using the idea of the improved spherical KNN Algorithm KNN^{PK++}[9] published by Wang et al., the distance measurement can be modified. The process of KNN^{PK++} is presented as follows: The program calculates the distance from each test point with the centroid of each cluster.The distance will minus the radius of the region to get the distance from the test point to the spherical region. When the distance is smaller than zero, this means that the test point

is inside the cluster. The corresponding cluster will then be used as the new dataset for the KNN algorithm application. Otherwise, the program will pick the spherical region closestto the test point and apply the KNN algorithm.

With the guidance of KNN^{PK++}, we have the following modified distance measurement:

(1)The KL divergence is used to measure the distance between the test point and centroids.

(2)The calculated distance will minus the radius of each corresponding cluster to get the distance to each cluster.

(3)The program will return the corresponding genre of the cluster.

3. Result

After several executions of the program, the following charts show the accuracy and execution time for both methods separately. All the programs are tested under the same GTZAN dataset with 70% of the training set and 30% of the test set. The accuracy is calculated by comparing the number of the predicted music genres with the original genre. The execution time is measured when the function is loaded with preprocessed data and ends when the accuracy of the overall program is given. Since each program gives a random execution time and accuracy, the following charts collect five times of execution with results. In result, the accuracy of genre prediction in Table 1 decreases significantly after the preprocessing of the dataset but with shocking execution time increasing, as shown in Table 2. In the KNN implementation, the program will go over the entire dataset and return the distance with each data point. It will take a long time for every sample in the test set to go through the dataset once. The time complexity of KNN implementation is O(n*m) for them to be the size of the test set and the n for the size of the training set. The implementation after preprocessing the dataset has a time complexity of O(k*n) for the k to be the number of clusters in the dataset and the n to be the size of the test set. A large decrease in time complexity can be observed after analysis.However, the result coming from the KNN implementation is solid, with the input dataset split into 70% of training set and 30% of test set, an average accuracy of 70% shows a good standard of the machine learning model. As a result, the only improvement of the new algorithm is an increase in execution speed.

Table 1. Accuracy

Execution	Before preprocessing	After preprocessing
1	72.27%	6.574%
2	71.52%	11.111%
3	67.51%	7.718%
4	69.06%	9.090%
5	71.14%	12.121%

Table 2. Execution time

Execution	Before preprocessing (seconds)	After preprocessing (seconds)
1	89.035	1.058
2	74.601	1.038
3	85.603	1.041
4	85.087	0.994
5	79.715	1.128

The following two tables (3 and 4) are the confusion matrix for the classifier before the preprocessing and after, respectively. The matrix is formed using the test set and the predicted results provided by the program. In the second matrix, it is obvious that the results concentrate on the fourth genre.

Table 3. Confusion matrix before preprocessing

	blues	classical	country	disco	hiphop	jazz	metal	Pop	reggae	rock
Blues	20	0	1	1	0	1	4	0	0	1
Classical	0	28	0	0	0	0	0	0	0	0
Country	1	1	23	3	0	2	1	0	1	2
Disco	0	0	1	20	1	0	3	0	0	3
Hiphop	0	0	0	3	18	0	0	1	2	1
Jazz	0	2	0	1	0	15	2	0	0	5
Metal	1	0	0	0	1	0	23	0	0	0
Pop	0	0	2	2	0	0	0	29	1	0
Reggae	0	1	2	5	2	0	0	0	19	1
Rock	1	1	1	5	1	2	2	0	2	15

Table 4. Confusion matrix after preprocessing

	blues	classical	country	disco	hiphop	jazz	metal	Pop	reggae	rock
Blues	0	0	0	37	0	0	0	0	0	0
Classical	1	0	0	39	0	0	0	0	0	0
Country	0	0	0	27	0	0	0	0	0	0
Disco	0	0	0	33	0	0	0	0	0	0
Hiphop	0	0	0	26	0	0	0	0	0	0
Jazz	0	0	0	33	0	0	0	0	0	0
Metal	1	0	0	30	0	0	0	0	0	0
Pop	0	0	0	27	0	0	0	0	0	0
Reggae	0	0	0	34	0	0	0	0	0	0
Rock	1	0	0	16	0	0	0	0	0	0

4. Conclusion

In brief, the results provided by the improved method did not show an optimization of the original algorithm. One of the core concepts of this method is to use the advantage of the GTZAN dataset, which is the classified data. The disadvantage of the k-means clustering is picking k clusters that will return the best-formed clusters. In the program, since each genre of music is stored in one folder, each folder can be treated as one cluster in the dataset. Therefore, in the data preprocessing, the program jumped the part that selected the k cluster and used only the selecting centroid part. However, one music genre can be originated from another; one song in a particular genre can share a lot similar with the other song that belongs to a different genre. This will make each genre cannot be treated as a spherical region in the dataset. Therefore, using the centroid and radius of each cluster will have a great bias. The second reason is the radius. The method of measuring radius is measuring the distance between the furthest sample with the centroid. If one sample shares a lot of similar features with a song in a different genre, the radius of that cluster will be a lot greater, which will lead to a large spherical region that contains partial samples from other clusters. When the program calculates the distance from the test point to each cluster, the result will be highly impacted by the large radius provided by one specific cluster.

For the further improvement of this method, the data preprocessing part should apply the k-means clustering method to the whole dataset. Therefore, after the clustering, the data should form multiple spherical regions, with each region may contain different genres of music. The program will first return the closest region to the test sample and apply the KNN algorithm to the chosen region. This will give a higher accuracy, but the hardest part should be applying the K-means clustering to the GMM for each song. On the other hand, if the new method improved the way to store the feature extraction for each song, the application of clustering will become easier.

References

[1] Wikipedia contributors. (2022, September 2). *Music genre*. Wikipedia. https://en.m.wikipedia.org/wiki/Music_genre

[2] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information* theory, *13*(1), 21-27.

[3] Kaggle. 2020. GTZAN Dataset - Music Genre Classification. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

[4] Ittichaichareon, C., Suksri, S., &Yingthawornsuk, T. (2012, July). Speech recognition using MFCC. In *International conference on computer graphics, simulation, and modeling* (Vol. 9).

[5] Dave, N. (2013). Feature extraction methods LPC, PLP, and MFCC in speech recognition. *International Journal for advance research in engineering and technology*, *1*(6), 1-4.

[6] Goldberger, J., &Aronowitz, H. (2005). A distance measure between GMMS based on the unscented transform and its application to speaker recognition. *Interspeech*, 1985-1988.

[7] Bouhlel, N., &Dziri, A. (2019). Kullback–Leibler divergence between multivariate generalized gaussian distributions. *IEEE Signal Processing Letters*, *26*(7), 1021-1025.

[8] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." *Pattern recognition* 40.7 (2007): 2038-2048.

[9] Wang, H., Xu, P., & Zhao, J. (2021). Improved KNN Algorithm Based on Preprocessing of Center in Smart Cities. *Complexity*, *2021*.

[10] Arthur, David, and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Stanford, 2006.