Music genre classification based on knn algorithm

Li Linjin
1023040910
Nanjing University of Posts and Telecommunications
School of Computer Scienc
Nanjing, China

***Abstract*—In order to classify music genres and enhance the user experience of the music push function, this article introduces a KNN (k-nearest neighbor algorithm). The classification method introduced in this article mainly uses the KNN algorithm to predict and classify the Mel frequency cepstral coefficient (MFCC) features of audio. The Mel frequency is proposed based on the hearing characteristics of the human ear, and it has a nonlinear correspondence relationship with the Hz frequency. The Mel frequency cepstrum coefficient is the Hz spectrum characteristic calculated using this relationship between them. In experiments, the accuracy of prediction using the classification method introduced in this article reached more than 70 percent.**

***Index Terms*—Signal; Features; k-nearest neighbor algorithm; Audio; Classification**

I. Introduction

In recent years, with the maturity of information technology and the development of the Internet, massive digital music information has emerged rapidly and has shown explosive growth. How to efficiently retrieve users' favorite music from vast media resources is of great significance to retrieving digital music resources. . In the face of large-scale online music data, the retrieval and classification of digital music has become a current research hotspot, especially the classification of music genre/style has become an important branch of it. Music classification is an important part of music information retrieval, which can help users find their favorite music more conveniently. In the digital music era, the amount of music data is growing explosively, so an effective classification method is needed to manage and organize this music data. K-Nearest Neighbor algorithm (KNN, K-Nearest Neighbor) is a simple and effective classification algorithm based on instance learning and classification by measuring the distance between different data points. Its basic idea is that if a sample belongs to a certain category among the K most similar (that is, the closest in the feature space) samples in the feature space, then the sample also belongs to this category. In this article, we will focus on how to use the KNN algorithm to classify music and conduct experimental verification using the GTZAN dataset.

II. Related work

In the field of music classification, many researchers try to classify music using different algorithms and techniques. Common algorithms include minimum distance method, neural network, support vector machine, decision tree method, hidden Markov model, etc. These methods usually require large amounts of labeled data and require a lot of time and computing resources to train. In contrast, the KNN algorithm is simple, easy to implement, and does not require a large amount of labeled data. At the same time, the KNN algorithm is easier for beginners to get started, and they can quickly master the ideas of machine learning by classifying music. The key to classifying music genres using the KNN algorithm is to choose an appropriate distance metric and number of neighbors. You can use different distance measures, such as Euclidean distance, cosine similarity, etc., and choose a suitable number of neighbors, such as 3, 5 or 10 nearest neighbors. To improve classification performance, feature selection and dimensionality reduction techniques can be used to reduce the number of features and retain the most important features. For example, dimensionality reduction techniques such as principal component analysis (PCA) or linear discriminant analysis (LDA) can be used to reduce feature dimensions. Use appropriate evaluation metrics (such as precision, recall, F1 score, etc.) to evaluate the classification model, and optimize the model based on the evaluation results. For example, you can adjust the K value, choose different feature combinations, or use different distance measures to improve model performance.

III. Problem Statement

Using the knn algorithm to classify music genres, the related work we have to do includes:

A. Data preprocessing

Before classifying music genres, music data needs to be preprocessed, including cleaning and screening noise and irrelevant information. For example, use an audio feature extraction toolkit such as Librosa to extract audio features such as Mel Frequency Cepstral Coefficients (MFCC), etc.

B. Feature extraction

In music genre classification, feature extraction is a key step. Since music data usually contains a lot of noise and irrelevant information, it needs to be cleaned and filtered. Common features include audio features, metadata, etc.Extract features from music data, such as pitch, rhythm, volume, etc. These features can reflect different aspects of music and help classification algorithms better understand music content. The feature value extracted in this experiment is the Mel frequency cepstrum coefficient.

C. Training model

Use the KNN algorithm to train the classification model. The K value needs to be determined, that is, the most similar K samples are selected. Different metrics can be used to evaluate the classification effect, such as precision, recall, and F1 score. In this experiment, the k value was set to 5, and the classification effect was evaluated through accuracy.

D. Apply the model

Apply the trained model to actual music data to classify music genres. Music can be divided into different genres such as rock, pop, classical, etc.

IV. Algorithms

The principle of KNN is that when predicting a new value x, it determines which category x belongs to based on the categories of the K nearest points.

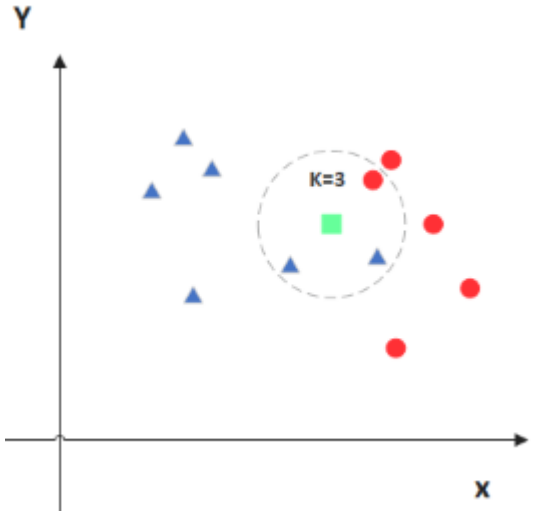

Fig. 1. Example of KNN.

The green point in the figure is the point we want to predict, assuming K=3. Then the KNN algorithm will find the three closest points (within the dotted circle) to see which category has more. In the example of the figure, there are more blue triangles, and the new green points will be classified into blue.

A. distance measure

The distance between two instance points in the feature space reflects the similarity of the two instance points. In the distance model KNN, there are many common distance measurement methods. Such as Euclidean distance, Manhattan distance, Minkovsky distance, Chebyshev distance and cosine distance.

- Manhattan distance, formally known as city block distance, also known as street distance, is the sum of the distances projected by line segments formed by fixed rectangular coordinates in Euclidean space. Its calculation method is equivalent to the 1st power representation of the Euclidean distance.

$$d(a,b) = \sum_{i=1}^{n}(|x_{ia} - x_{ib}|)$$

Fig. 2. Manhattan distance.

- Min's distance is not a kind of distance, but a set of distance definitions, which is a general expression of multiple distance measurement formulas. Both Euclidean distance and Manhattan distance can be regarded as a special case of Minkowski distance.

$$d(a,b) = \sqrt[p]{\sum_{i=1}^{n}(|x_{ia} - x_{ib}|)^p}$$

Fig. 3. Minkowski distance.

- Cosine similarity measures the cosine value of the angle between two vectors. The value range is [-1,1]. The larger the value, the more similar the two vectors are.

$$Similarity(A,B) = \cos(A,B) = \frac{A \cdot B}{\|A\| \|B\|}$$

Fig. 4. Cosine similarity.

- Among them, Euclidean distance is the most common.Defined in Euclidean space, the distance between two points or between multiple points is also called the Euclidean metric.

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Fig. 5. Euclidean distance.

B. k value selection

The choice of k value can have a significant impact on the results of the KNN algorithm.

- Reducing the value of k means that the overall model becomes complex and is susceptible to overfitting due to noise in the training data.
- An increase in the k value means that the overall model becomes simpler. If k is too large, the nearest neighbor classifier may misclassify the test sample because the k nearest neighbors may include data points that are far away and are not of the same type.

In applications, the k value is generally chosen to be a smaller value, and cross-validation is usually used to select the optimal k value.

V. Feature extraction

A. librosa

librosa is a Python library specialized for audio processing and analysis. It is based on the NumPy and SciPy libraries and provides a variety of data visualization methods, audio preprocessing tools, and audio feature extraction methods. It is one of the most popular audio processing libraries in the Python language.

Librosa has a wide range of applications in machine learning, especially in the field of audio processing and analysis.

- Feature extraction: Librosa provides a series of audio feature extraction methods, including spectral features, rhythm features, sound intensity features, etc. These features can be used to train machine learning models to achieve various audio-related tasks, such as audio classification, audio clustering, and audio recognition.
- Data preprocessing: Librosa can read audio files in various formats and convert them into time series data. By using Librosa, audio data can be preprocessed, such as adjusting the audio sampling rate, mixing and splicing audio, etc., to meet the requirements of the machine learning model.
- Data Visualization: Librosa provides a variety of data visualization methods to visualize audio data and other features. By using Librosa, you can intuitively understand the distribution of data and the extraction results of features, which helps to better understand the data and model.
- Audio recognition: Librosa can be used in conjunction with other machine learning libraries to implement various audio recognition tasks, such as speech recognition, music classification, and sound event detection. By using Librosa to extract audio features, you can train a classifier or use a deep learning model for audio recognition.
- Audio Analysis: Librosa can help analyze the structure and patterns of audio data. For example, you can use Librosa to analyze the rhythm and melody of music, or analyze intonation and other sound characteristics in speech. These analysis results can be used in applications such as music information retrieval, speech recognition, and sentiment analysis.

Librosa is widely used in the field of audio processing and analysis in machine learning. By using Librosa to extract audio features, perform data preprocessing and visualization, and combine with other machine learning libraries, a variety of audio-related tasks and applications can be achieved.

B. Python_speech_features

Python_speech_features is a Python toolkit for audio feature extraction, specifically used for audio signal processing. It provides a variety of commonly used spectral feature extraction methods, such as Mel spectrum, MFCC (Mel frequency cepstrum coefficient), CQT (Constant-Q transform), etc. The library also provides other audio features such as cepstrum coefficients, linear predictive coding, etc.

The Python_speech_features library is mainly used for audio feature extraction in machine learning, especially in the fields of speech recognition and speech analysis.

- Feature extraction: The Python_speech_features library provides a variety of functions for audio feature extraction. These functions can be used to extract key information in audio signals, such as Mel spectrum, MFCC (Mel Frequency Cepstral Coefficients), etc., which can provide rich information about the audio signal, such as pitch, intensity, and timbre.
- Audio classification: By extracting audio features, machine learning models can be trained to perform audio classification tasks, such as speech recognition, music classification, etc. These classification tasks can help us understand the intrinsic structure and patterns of audio data.
- Audio clustering: By extracting audio features, you can also use clustering algorithms to cluster audio data, such as K-means clustering. This can help us discover similarities and differences in audio data.
- Audio recognition: Combined with other machine learning libraries, such as scikit-learn or TensorFlow, Python_speech_features can be used to implement various audio recognition tasks, such as speech recognition, music recognition, etc. By training a classifier or using a deep learning model, a specific audio signal or sound can be identified.
- Audio analysis: The Python_speech_features library can also be used to analyze the structure and patterns of audio data, such as analyzing the rhythm and melody of music, or analyzing intonation and other sound features in speech. These analysis results can be used in applications such as music information retrieval, speech recognition, and sentiment analysis.

C. Audio features

Music feature extraction is a crucial part of the classification task and plays a key role in the final music classification effect. Trabelsi et al. extracted the Mel Frequency Cepstrum Coefficient (MFCC) in the process of classifying music, and confirmed that MFCC is an important characteristic parameter for distinguishing music types. Baniya et al. extracted two types of feature parameters, MFCC and beat histogram, as music-related features to classify music. Yang Xiaoyu et al. performed a more accurate classification of music by extracting short-term energy, short-term average zero-crossing rate and short-term average amplitude as feature values.Lei Wenkang extracted the characteristics of the music signal (timbral characteristics, rhythm characteristics, pitch characteristics) and spectrogram characteristics (time-frequency diagram obtained by short-time Fourier transform, Mel spectrum, constant Q spectrum) and input them into the recurrent neural network Categorize music genres. Liu Yuting

classifies music through content-based music features (timbral features, listening features and beat features)

This experiment uses mfcc as the feature value to classify music genres, and selects python_speech_features as the feature value extraction library.

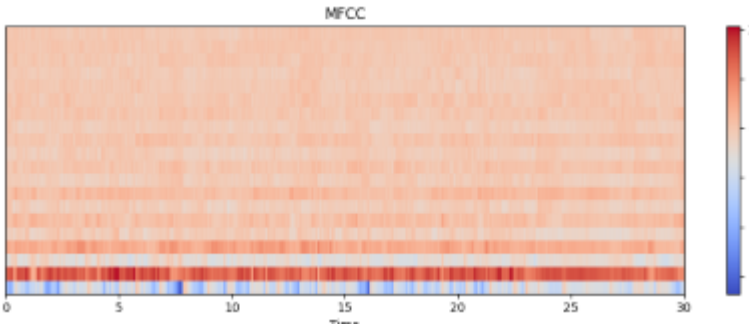

Fig. 6. MFCC Feature.

VI. Evaluation

A. GTZAN

The music data set used in this experiment is GTZAN, a genre collection data set collected in 2000-2001. It consists of 1000 audio files, each with a duration of 30 seconds. There are 10 classes (10 music genres), each containing 100 audio tracks. Each audio track is in .wav format. It contains audio files of the following 10 types: blues, classical, country, disco, hip-hop, jazz, metal, pop, music, reggae, rock

B. calculate distance

The knn distance metric calculation method selected in this experiment is not a single distance metric, but a method that combines multiple distance metrics. It includes three parts, Euclidean distance, cosine similarity, and log-likelihood similarity. The log-likelihood similarity measures the similarity between two probability distributions, and the larger the value, the more similar they are.This method of mixing multiple distance measures can capture more feature information, sometimes better than a single distance measure. But at the same time, the weights and parameters of each part need to be carefully adjusted to adapt to specific application scenarios.

C. selection of k

In terms of k value selection, by solving the knn algorithm prediction accuracy for integers in the range of 1 to 20, a curve composed of a series of points can be obtained. The trend of the curve first rises and then falls, and at the k value The maximum value is obtained when it is 4. Calculating the variance for each value shows that when the k value is equal to 4, it has the best accuracy.

D. result

We divide music into different categories and use the KNN algorithm for classification. The final accuracy of classification prediction can reach more than 70%. Experimental results show that the KNN algorithm has better performance in music classification. Compared with traditional machine learning methods, the KNN algorithm has the advantages of being simple, easy to implement, and does not require a large amount of labeled data. In addition, we can also adjust the distance measurement algorithm of feature values to obtain better classification results.

E. Improvement and optimization

In this experiment, only one eigenvalue was used for distance measurement. If you want to obtain more accurate classification results, you can use multi-dimensional eigenvalue evaluation. However, due to the increase in the number of features, irrelevant and redundant features are prone to exist. On the contrary, irrelevant and redundant features will bring the risk of over-fitting, resulting in poor performance of the classification model. In order to avoid the problem of poor classification performance caused by too many features, it is necessary to filter features, select important features and eliminate irrelevant and redundant features.

- KPCA:Classic subspace learning methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) obtain low-dimensional robust features by finding projection matrices. While reducing the dimension, they do not perform effective Feature selection affects the accuracy of feature representation to a certain extent.Kernel principal component analysis is used to screen the input time domain features, frequency domain features and cepstral domain features. On this basis, the new filtered feature set is input into the improved KNN classification model to find the best one. Feature subset, by applying the best feature subset, improves the classification accuracy of the KNN algorithm.
- Improved KNN algorithm:When the extracted features are input into the KNN classification model, for KNN, since it treats each feature item and neighbor samples equally, the features of each dimension contribute the same to its distance. If there are certain n-dimensional features that have no impact on music classification, then their existence will reduce the accuracy of music classification. Therefore, the KNN algorithm was improved by introducing feature selection. This allows KNN to select feature attributes that are more valuable to the classification algorithm.

VII. Conclusion

This experiment shows that the KNN algorithm can be effectively applied to music classification. The KNN algorithm is simple, easy to implement, and does not require a large amount of labeled data. In addition, the KNN algorithm can also adjust parameters as needed to obtain better classification results.but,The knn algorithm also has some problems, including a large amount of calculation - the KNN algorithm has a large amount of calculation, especially when the data set is large or the feature dimension is high. In order to speed up calculations, some optimization techniques can be used,

such as data structures such as KD trees or ball trees to speed up nearest neighbor searches; sample imbalance problem - in some cases, the number of samples in some categories may be much smaller than other categories, resulting in Imbalanced classification problem. This may cause samples of certain categories to be ignored or underestimated; requires a lot of memory - the KNN algorithm needs to store the entire training data set, and for large-scale data sets, a large amount of memory and storage space may be required; cannot give intrinsic meaning - KNN The algorithm is an example-based learning method that cannot give the inner meaning and explanation of the data, which makes it more difficult to understand the model's decisions.

In the future, we will further study how to extract more effective music features to further improve the performance of music classification. At the same time, we will also explore the application of other machine learning algorithms in music classification and compare the performance differences of different algorithms, such as multi-class support vector machines, K-means clustering, and convolutional neural networks.

REFERENCES

[1] 宋扬, 王海龙, 柳林等. 融合KPCA与改进KNN的蒙古族音乐分类方法[J]. 复旦学报(自然科学版), 2022, 第61卷(5):573-580，588.

[2] 胡昭华，余媛媛.深度卷积神经网络在音乐风格识别中的应用[J].小型微型计算机系统，2018，39（9）：1932-1936.

[3] 辛欣,陈曙东,仝明磊,等.采用潜在概率语义模型和K近邻分类器的音频分类算法[J].华侨大学学报(自然科学版),2016,37(2):196-200.

[4] 刘万军, 王佳铭, 曲海成等. 基于频谱空间域特征注意的音乐流派分类算法[J]. 计算机应用, 2022, 第42卷(7):2072-2077.

[5] 史岩.从统计学习理论到应用领域与模型拓展：KNN综合探究[J].企业科技与发展,2023(10):27-30.DOI:10.20137/j.cnki.45-1359/t.2023.10.012.