Improving Artist Recognition Based on Million Song Dataset

Shujian Liu
Department of Mechanical and Industrial Engineering
University of Massachusetts Amherst
Amherst, MA 01003
shujian@umass.edu

Abstract

Artist recognition is to predict the artist ID of a song based the other information: analysis sample rate, bars start, beats starts and so on. Machine learning algorithms can learn from the training data and predict the artist ID based on test data. This article is focused on finding the appropriate model and tuning the parameters for high accuracy. The Million Song Dataset is used for the research. It is a large dataset of music consisting of audio features and metadata for one million songs. A variety of methods including supervised learning and unsupervised learning are tested. Good result is generated from LDA method (7.201% compared to 2.103% in KNN with k=1).

1 Introduction

Columbia University has released a large music dataset named "Million Song Dataset", for the purposes of encouraging research on machine learning algorithms for commercial scale, providing a reference to evaluate research and help new researchers to get started in the music information retrieval field [1-2].

This dataset enables plenty of interesting task such as year recognition (as in homework 3), artist recognition, imputation of missing data, cover song recognition. The original authors have made attempts about artist recognition problem with K Nearest Neighbors algorithm (KNN), however, only low accuracy is achieved.

This article is focused on obtaining better accuracy with the machine learning algorithm taught in class. There are significant commercial values in the problem. A successful application of artist/song recognition is Shazam, which is best known for music identification capabilities. Other services include ACRCloud, Audible Magic and Gracenote.

2 Million Song Dataset

The entire Million Song Dataset (MSD) contains 1 million songs and 44,745 unique artists based on their Echo Nest ID, out of that, 18,073 have at least 20 songs (the entire dataset takes 280 GB). Due to limitation of my PC and technical question about accessing it in Amazon AWS, this article used the subset (10,000 songs, 1.8 GB before compression) as provided.

The subset can be downloaded from: http://static.echonest.com/millionsongsubset_full.tar.gz

In the dataset, each file represents one track with all the related information. The actual file format of each file is HDF5. The wrapper code is provided but user need to install *libhdf5-7* and *python-tables* packages to access data. There are 90 features in MSD, such as analysis sample rate, artist information, bar confidence and start, beats confidence and start.

A list is in http://labrosa.ee.columbia.edu/millionsong/faq under Field list.

3 Related Work

Recognizing the artist from the audio is an interesting task that provides an opportunity for both audio features and machine learning. Artist recognition has been studied for decades, there some early works in [4-7]. However, there was no large scale dataset available to the academics.

Realizing this problem, the Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) of Columbia University has built Million Song Dataset and made it available. MSD offers is the 18, 073 artists who have at least 20 songs (compared to 5 artists reported a decade ago).

One popular Kaggle completion was made this dataset. In this Million Song Dataset Challenge, participants are working on predicting which songs a user will listen to [8]. It aims at being the best possible offline evaluation of a music recommendation system. Any type of algorithm can be used: collaborative filtering, content-based methods, web crawling.

In this artist recognizing problem, they provide two split of the data: 1) in the regular split, each artist has 15 songs in the training set, the rest is in the testing set. 2) in the unbalanced split, each artist has 2/3 of his songs in the training set, the rest in the testing set. This article only considers the unbalanced split which it easier to work with.

The authors achieved accuracy of 9.578% for unbalanced split on the entire dataset (each artist has 2/3 of his songs in the training set, the rest in the testing set) with KNN model (K=1). In the present research, using their code on the subset (2805 songs in the test case) leads to an accuracy of 2.2103%. The reason of lower accuracy may be the subset only contains selected tracks with ID starting with A and B. Artist who has tracks from A and B may be more difficult to recognized compare to C-Z.

A more recent article [9] uses MLP with backpropagation for this dataset. They boost the accuracy using MLP with backpropagation.

4 Proposed Solution

4.1 Supervised Learning

This classification problem is first solved by supervised learning methods. Supervised learning can analyze the training data and products a model, this model can be used to predict on new data.

4.1.1 Basic classifiers

With the starter code from original authors, the hyperparamter of KNN method K is tested. To better resolve this classification problem, a variety of classifiers are going to be used, which include decision tree, Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), Gaussian Naive Bayes (NB), logistic regression (LR) from Scikit-Learn library. Multilayer perceptron (MLP) neural network models such as sknn.mlp, scikit-neuralnetwork and pybrain.

4.1.2 Feature Selection

After testing these classifiers with default settings, feature selection is used on the most accurate model. SelectKBest model from Scikit-Learn library. This model can select features according to the k highest scores where k is a user input. The f_classif model is used to compute the ANOVA F-value for the provided sample. In feature selection methods, redundant or irrelevant features can be removed without incurring much loss of information. This can both decrease the computation time and increase the predicting accuracy.

4.2 Unsupervised Learning

Unsupervised learning methods are also used for this article. Principal Component Analysis

(PCA) as a well-used dimensionality reduction model is tested in this part. PCA has the advantages of its low noise sensitivity, the decreased requirements for capacity and memory, and increased efficiency given the processes taking place in a smaller dimension. The hyper-parameter of PCA is the number of components to keep. An exhaustive search over possible to find the best value of hyper-parameter.

5 Experiments and Results

With the starting KNN code from the Columbia University, a range of K is tested for the unbalanced split of data. It turns out that only k=1 works (59 out of 2805 songs). With k>1, the predicting accuracy is always zero. In KNN methods, everything is far from everything else in high dimensions. Using the whole set of features will make poor prediction since unrelated features will made the songs from one artist far away. In this case, only k=1 can make some good prediction. Trying to take majority vote of several neighbors is unlike to make any correct prediction.

Several classifiers are tested and the accuracy is shown in Table 1.

Table 1: Accuracy of classifiers

Classifier Name	**Accuracy**
KNN (k=1)	2.103%
Decision tree	0.749%
LDA	7.201%
Gaussian NB	2.460%
LR	2.103%
SVM	failed

The best prediction is given be LDA while decision tree has poor performance. LDA parameters are learned using maximum likelihood, which reduces to using sample estimates. The assumptions LDA makes will rarely be correct for real-world problems. However, the induced linear decision boundaries did perform reasonably well. On the other hand. Decision-tree learners can create over-complex trees that do not generalize the data well.

In the first and second homework, the well-used ensemble method RandomForestClassifier is also tested here. However, it always return out-of-memory error with a deep tree of large number of forest. The dataset is too large for Random Forest to run in a virtual machine on PC.

It is worth mentioned that several neural network models (sknn.mlp, scikit-neuralnetwork and pybrain) has been used in current project, however none of them can work properly or generate reasonable output.

By digging into the 90 features. Intuitively, features such as artist_name, the musicbrainz.org ID for this artists, ID of that artist on the service playme.com cannot be used for training. It is cheating. Features such as The Echo Nest song ID, song title are useless or even disturbing. Features such start time of each beat, time of the end of the fade, general loudness of the track are interesting and important for predicting. Feature selection may help increase the accuracy. SelectKBest with f_classif from Scikit-Learn library is used on the LDA and NB since they have best performance in the last part. The number of top features to select k has been searched in range of 40 to 90. The result is shown in Figure 1.

Feature selection has negative effects on LDA but slightly improved Gaussian NB. The other way tried to deal with redundant or irrelevant features is dimensional reduction. PCA is used in this part. Number of components to keep has been searched in range of 50 to 90. The results are shown in Figure 2. The negative effect of PCA is even strong for LDA.

The main reason that both feature selection and dimensional reduction cannot improve accuracy may be that the irrelevant features are still hard to be find with automatic feature selection and dimensional reduction methods. One potential direct of future work can be manually filtering of these features and reduce them in the dataset to boost the accuracy.

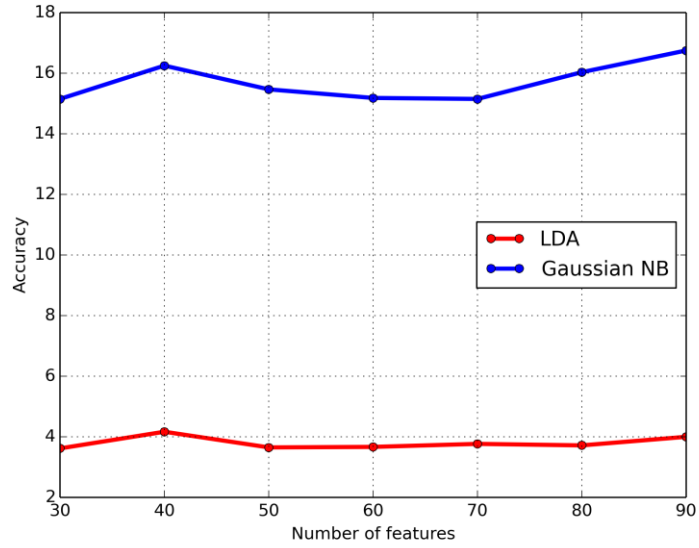

Figure 1: Accuracy with feature selection (SelectKBest)

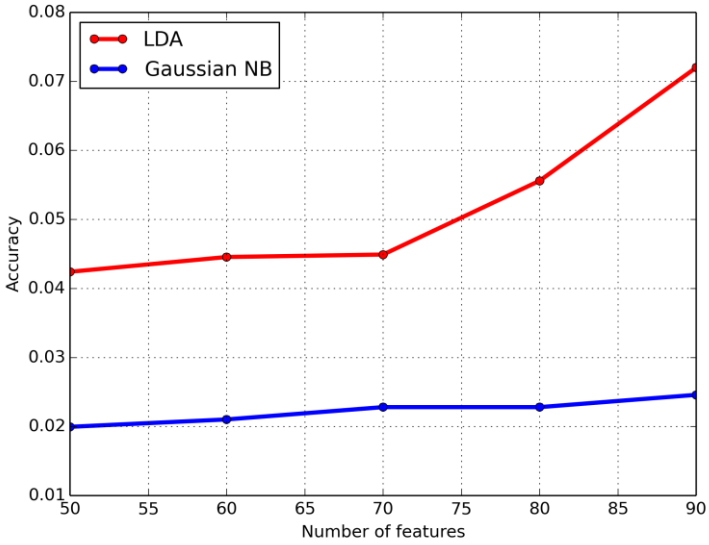

Figure 2: Accuracy with dimensional reduction (PCA)

5 Discussion and Conclusions

The task in real world is much more complicated than in the homework. A lot of effects are done to preprocessing the data and to install packages. The present research is focus on artist recognition problem with machine learning algorithms. It is a large dataset of music

consisting of audio features and metadata for one million songs named MSD is used. A variety of methods including supervised learning and unsupervised learning are tested. Both feature selection and dimensional reduction cannot improve the best case with LDA. Best result is generated from LDA method (7.201% compared to 2.103% in KNN with k=1). Tests with deep learning is suggested in the future work. Manually filter the data may be helpful to achieve a higher accuracy and provide more value for commercial use.

References

[1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[2] Million Song Dataset: http://labrosa.ee.columbia.edu/millionsong/pages/getting-dataset

[3] Bertin-Mahieux, Thierry. Large-Scale Pattern Discovery in Music. Diss. Columbia University, 2013.

[4] Bergstra, James, et al. "Aggregate features and AdaBoost for music classification." Machine learning 65.2-3 (2006): 473-484.

[5] Whitman, Brian, Gary Flake, and Steve Lawrence. "Artist detection in music with minnowmatch." Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop. IEEE, 2001.

[6] Kaminskas, Marius, and Francesco Ricci. "Contextual music information retrieval and recommendation: State of the art and challenges." Computer Science Review 6.2 (2012): 89-119.

[7] Henaff, Mikael, et al. "Unsupervised learning of sparse features for scalable audio classification." ISMIR. Vol. 11. 2011.

[8] Million Song Dataset Challenge. https://www.kaggle.com/c/msdchallenge

[9] Dieleman, Sander, Philémon Brakel, and Benjamin Schrauwen. "Audio-based music classification with a pretrained convolutional network." 12th International Society for Music Information Retrieval Conference (ISMIR-2011). University of Miami, 2011.