Music Recommendation System

Yuan Gao, Weijie Ye, Xining Wang, Chong Hu

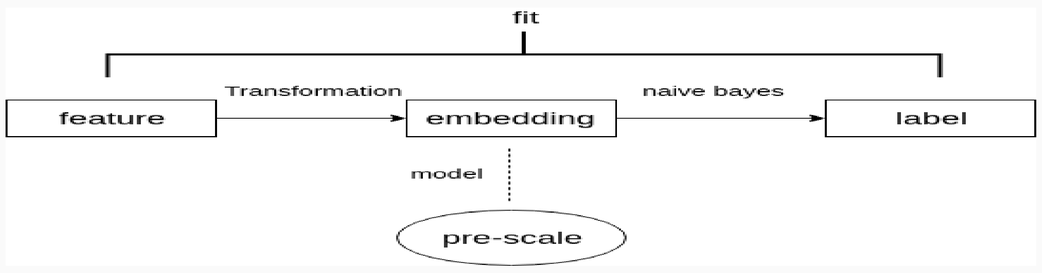
Ve572 Project, Group 3

Introduction

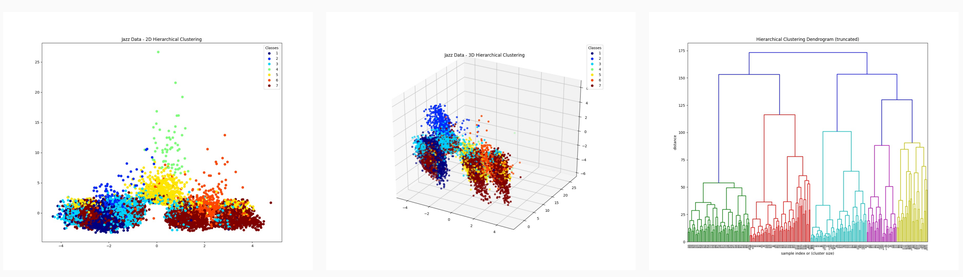
We have implemented a music recommendation system based on the analysis of Million Song Dataset [1]. We use Hadoop as the big data platform to execute operations that are too resource-consuming for a single computer. We store avro files on hdfs to save the disk capacity. We have implemented a breadth first search (BFS) algorithm to search for all the similar artists for a song' singer. We implement this algorithm both in MapReduce and Spark and we compare the advantages of those two methods. Besides that, we have developed a genre classification algorithm to search for songs with similar genre.

Genre Prediction

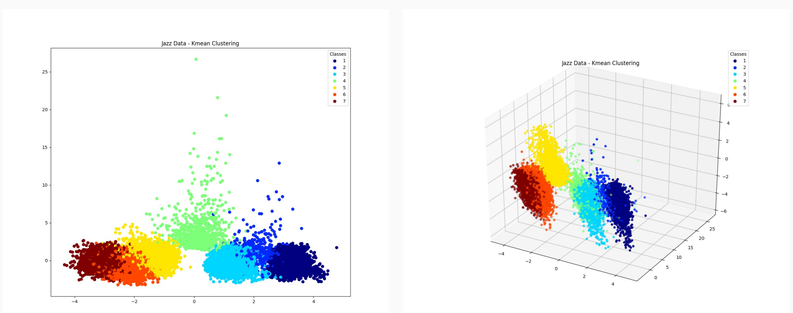
- Feature Selection:
  - We drop all the data with zero years
  - We use the artist_term as standard and choose the variable with large variance.
- Pre-Scaling based on Naive Bayes: We use the jazz label as the prior to help our scaling, so that we could retrive more information from current data.



- Hierarchical clustering andK-mean clustering
  - For K-mean clustering, we use K-mean++ strategy to assign initial value and run several time to find the global optimum.
  - For hierarchical clustering, we apply Ward's method as criterion in cluster analysis.
  - To visualize the effectiveness of two clustering method, we use the first two and three components to draw 2D and 3D plots.
- Hierarchical clustering



- K-mean clustering



Drill Query

According to what Reapor needs, some basic information should be retrieved from Drill with database queries as a start of the music recommendation system.

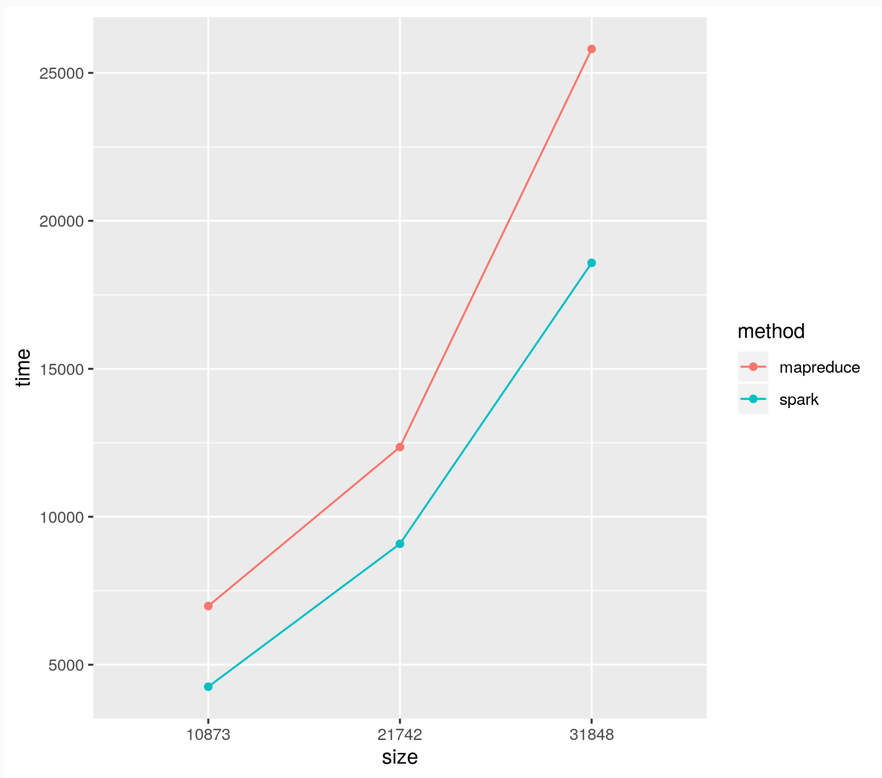
The query includes the span of the age of songs, hottest song that is the shortest, as well as the album with the most tracks, which are all required by Reapor.

Drill could be a good starting tool for us to get familiar with the dataset and have a rough understanding of the dataset, with its fast speed and useful result.

BFS

Given a source artist, our BFS algorithm could output a file that contains all the other artists' distance. The distance is calculated by the attribute similar_artists. All the similar artists have distance 1 and our BFS algorithm could iterate all the similar artists' similar artists and output a cumulative distance. If an artist can not be found in all the iterated similar artists, the distance will be labeled as infinity.

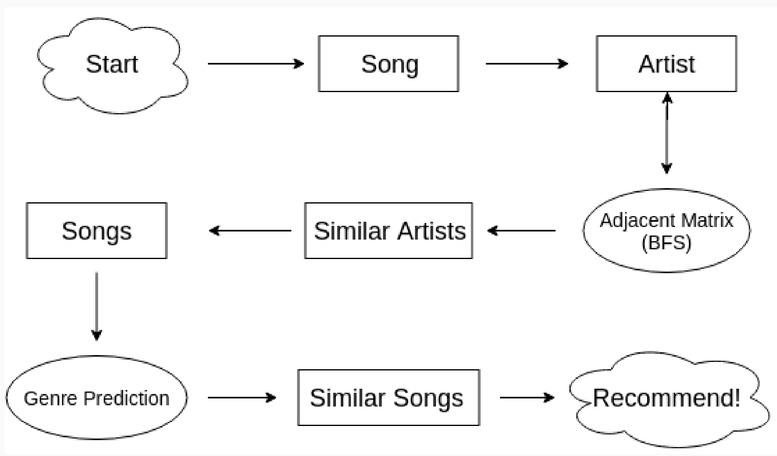
We implemented this algorithm in both MapReduce and Spark. We compared the efficiency of those two implementations.



It could be found that in the same environment Spark is more efficient than MapReduce.

Combination of BFS and Clustering

Finally, we combined our genre prediction clustering algorithm with the BFS similar artists algorithm to reach a music recommendation system. We could retrieve the artist from a song and use BFS to find all the similar artists and their songs. Then, we could find all similar songs by the genre prediction clustering algorithm. Then, we got the recommended songs!



Reference

[1] Thierry Bertin-Mahieux et al. "The Million Song Dataset". In: Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011). 2011.