Music Recommendation System

analyzed from Million Song Dataset

Yuan Gao Weijie Ye Xining Wang Chong Hu

Ve572 Project, Group 3

August 2, 2019

Contents

Introduction

In order to help Reapor build the music platform, we designed the music recommendation system

- Preparations
  - Analyze songs from Million Song Dataset (MSD) [1]
  - Use Hadoop cluster to store data and do operations
  - Execute Drill query to search for useful information
- Recommendation System
  - Breadth First Search for similar artists
  - Genre prediction
  - Final Pipeline Structure

Contents

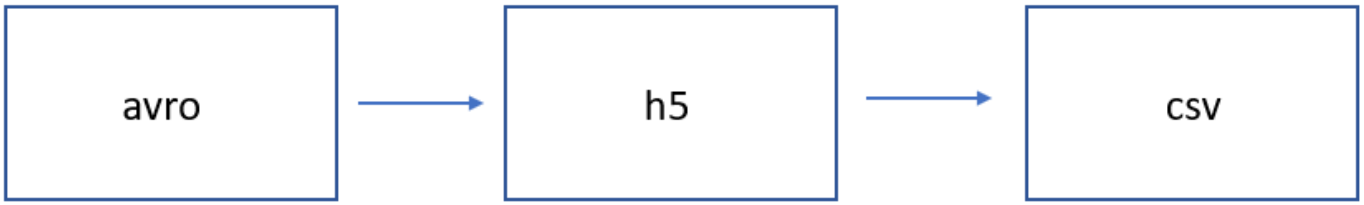
Data Processing

We extract h5 files from avro files on hdfs

- avro schema: filecontent, filename, checksum

Then, we convert h5 files to csv files

- use hdf5-java
- extract the only one artist term which has the largest weight
- 27 attributes, such as, similar_artists, artist_hotness, tempo...



Contents

Drill

- The dataset we use for Drill is A directory to N directory (A2N.csv)
- The range of dates covered by the songs in the dataset, i.e. the age of the oldest and of the youngest songs.

```
apache drill> select MIN(columns[8]), MAX(columns[8])
. .semicolon> from dfs.`/home/pgroup3/A2N.csv`
. .semicolon> where columns[8] <> '0' and columns[8] <> 'year';
+---------+---------+
| EXPR$0  | EXPR$1  |
+---------+---------+
| 1922    | 2011    |
+---------+---------+
1 row selected (56.949 seconds)
```

- Find the hottest song (0.9) that is the shortest

Answer

title: Wake (Album Version)
hotness: 0.9271
duration: 100.91s

Drill

- highest energy with lowest tempo. However, all energy is 0.
- the name of the album with the most tracks

Answer

Album: Intro Count: 831

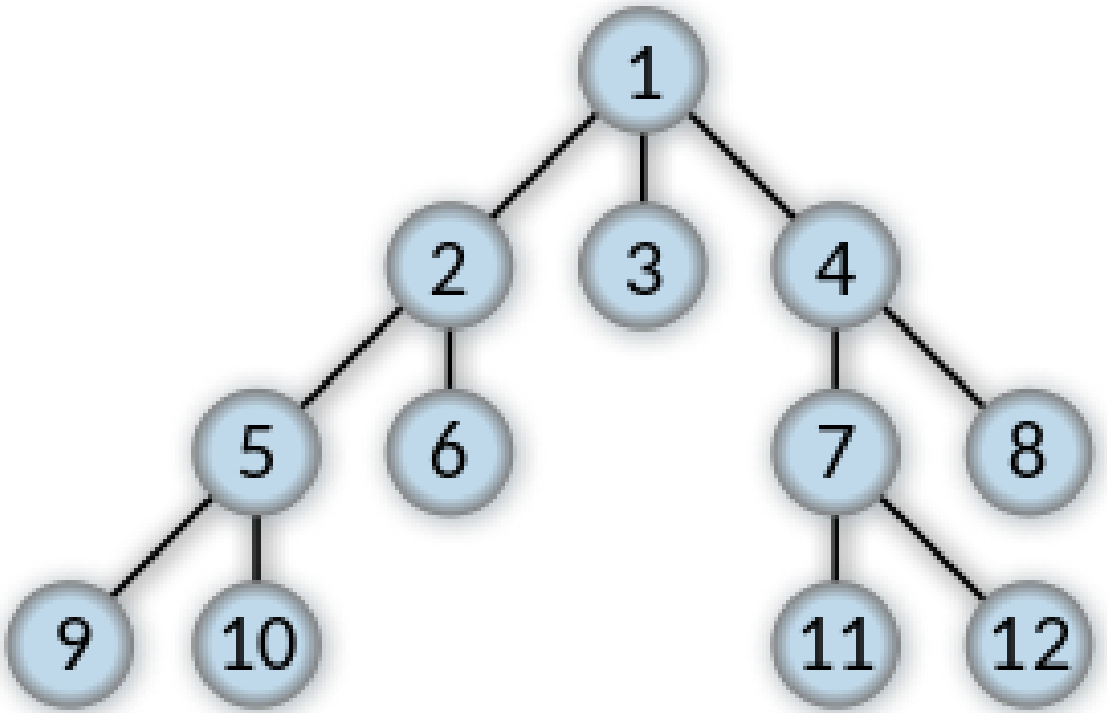
- the band who recorded the longest song. (no band, artist instead)

Answer

album: Alice Coltrane duration: 999.76s

Contents

Parallel BFS



- Maintain a result list of nodes: (artist, (similar artist, d, status))
- Map function: Explore and add nodes to the list
- Reduce function: Combine the nodes with same key
- Iterate until no node to explore

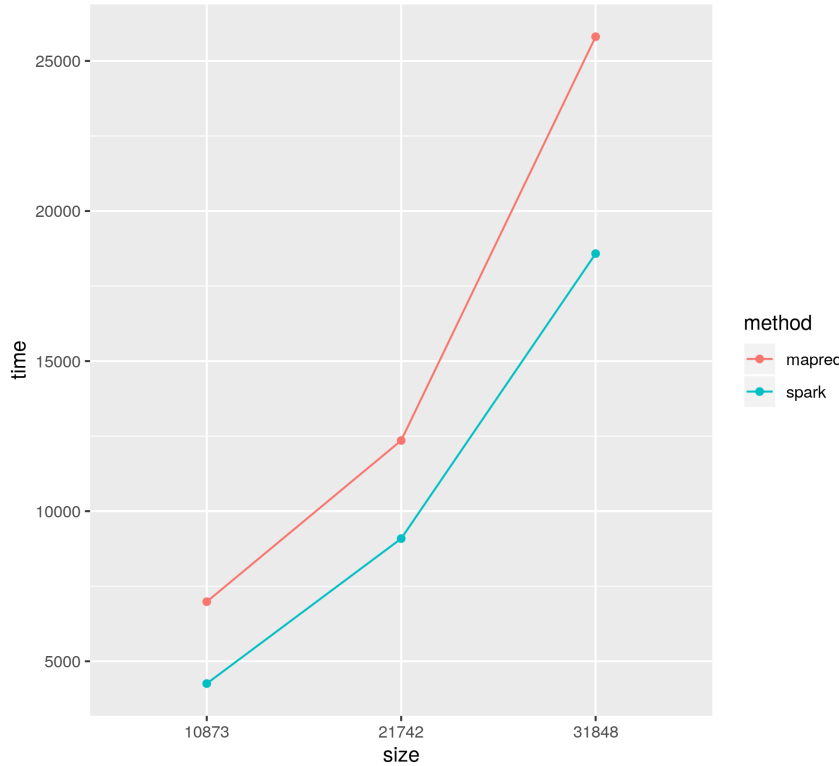
Results of BFS

```
ARRNQMS11F50C4E389,2
AR0IAWL1187B9A96D0,3
ARYQLYJ11F50C49FC2,4
ARQ6LU71187B98A31E,4
AR06SRC1187B98F3B6,3
ARTWBUH12454A43277,1
ARL3HWI1187FB5638A,4
AR553M51187B98BCE7,2
ARKGBWV1187B988D5E,2
ARHF0RW1187B9B509B,2
ARGSFAJ11F50C4DEEB,3
ARMNDIJ1187B9A09EA,3
ARR960Z1187B990E59,2
AR8QJHN1187FB3616E,2
ARQL9SK1187FB41D84,2
ARR32JQ1187FB42A2A,2
ARRXSSI12BAB079E45,3
AR1LWLE1187FB3FBBC,2
ARC5EHQ1187FB4834F,1
ARZFYGS12AA61C8E37,4
ARYQOXF12454A38B47,3
ARKY0VI1187B9952E1,3
ARR7MUX1187FB56F65,3
ARLJNIT1241B9CB4F4,2
ARC5JAZ1187B98EA82,2
ARMCGQI1187FB49DDE,2
ARPXZES1187B99DD71,3
ARGC7LW1187B9AF7BB,2
```

MapReduce vs Spark

Environment:

OS	4.15.0-55-generic Unbuntu18.04
CPU	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz x8



- Spark is more efficient because less I/O

Genre Prediction

Our job in genre prediction is to investigate genre inside different jazz. The total dataset of jazz we used is roughly about 30000.

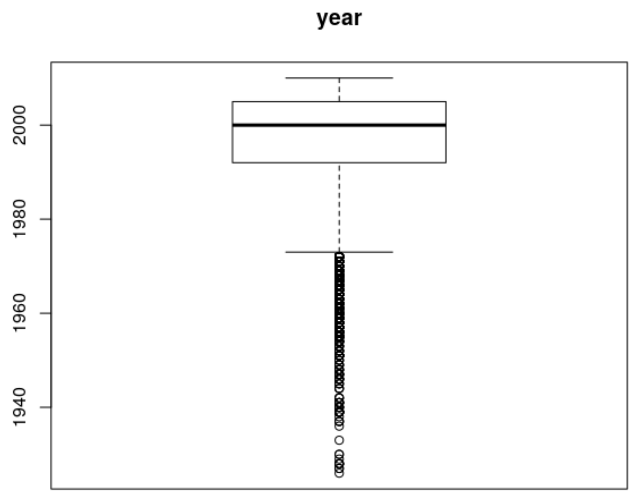
1. Feature Selection
2. Pre-Scaling based on Naive Bayes
3. Comparison between Hierarchical and K-mean Clustering
4. Drawbacks and Corresponding Solution

Feature Selection

Our strategy to select feature is to pick useful variables that could be easily handled. In this scenario, we pay more attention to numerical variables.

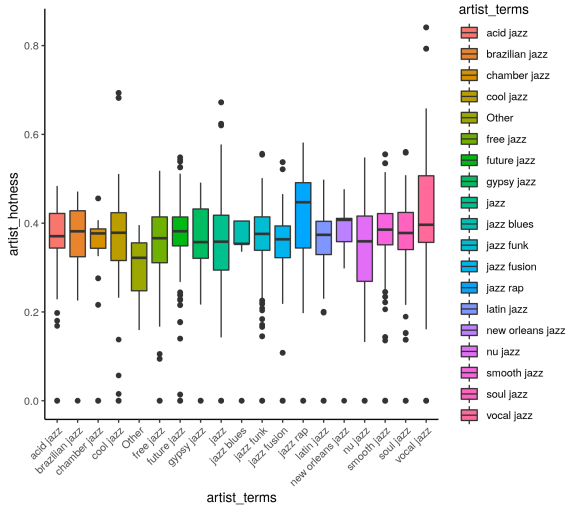
- How to drop variables.

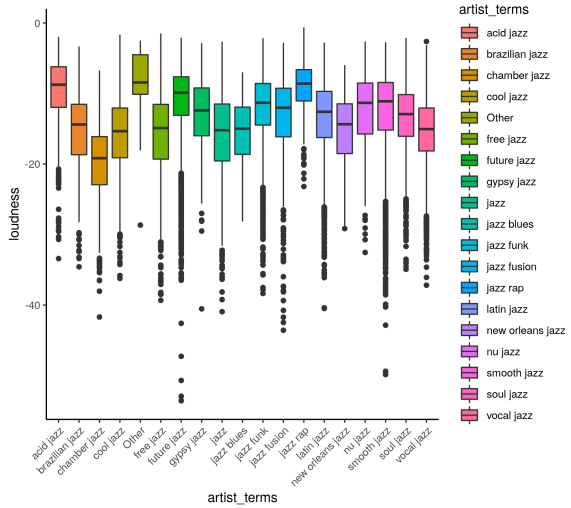
elements	elements with zero year
31847	15493

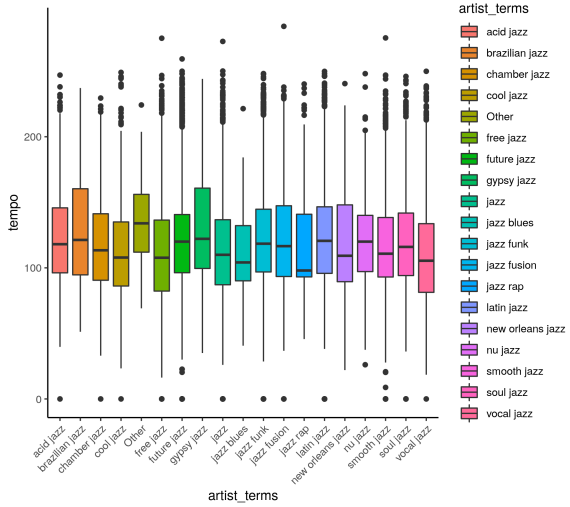


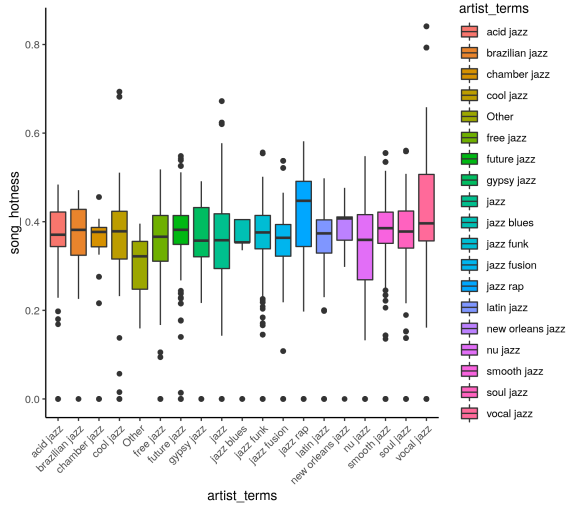
Feature Selection

- How to select variables: Use the `artist_term` as standard since it has a lot of hidden information. Variable that has large variety in `artist_term` would contain more unique features of that song.









Pre-Scaling based on Naive Bayes

Why need to pre-scale data

Different variables have different varieties; different variables have different scaled importance of embedded information.

How to scale data

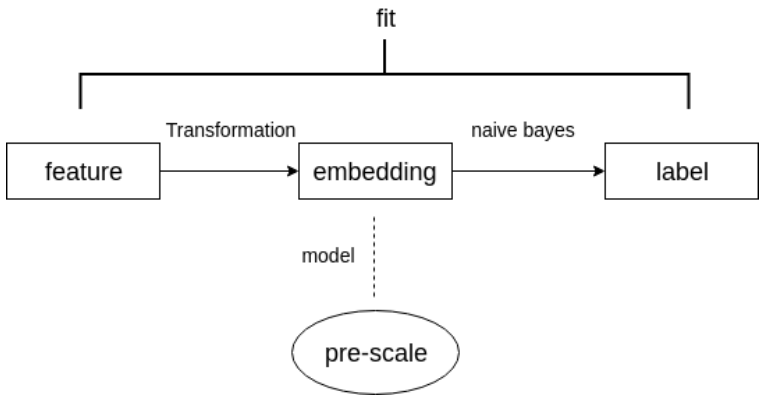
- It would be common to treat this as a single classification problem to tell which label the song has.
- However, classification couldn't rely too much on the label and ignore some features in certain degree. Besides, different jazz label may also have a large overlap.
- To solve those problems, our strategy is to use jazz label as prior to guide our scaling. In this way, we can combine the label information and features inside data.

How to scale data

How to apply the label prior information:

- Gaussian Naive Bayes model

$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y)$$

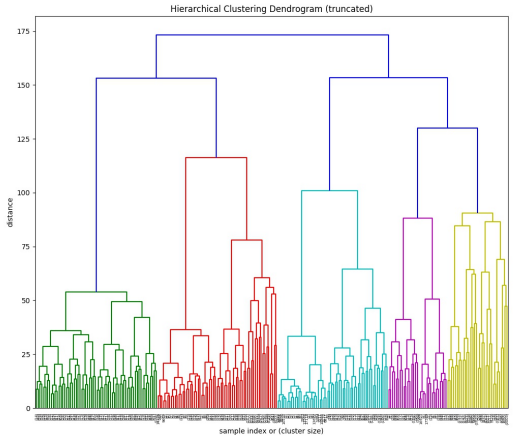


Based on the embedding information, we could run the clustering algorithm to dig more features inside data.

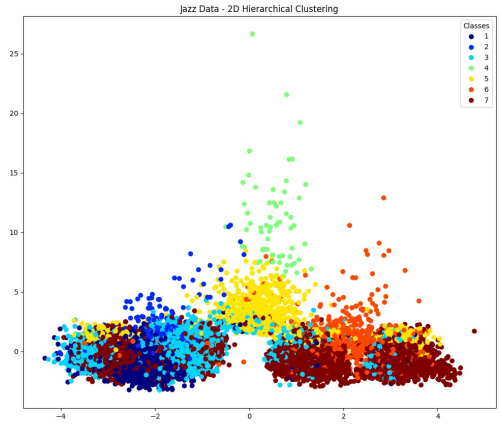
To select a better algorithm, we run both hierarchical clustering and K-mean clustering on the data set.

- 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.

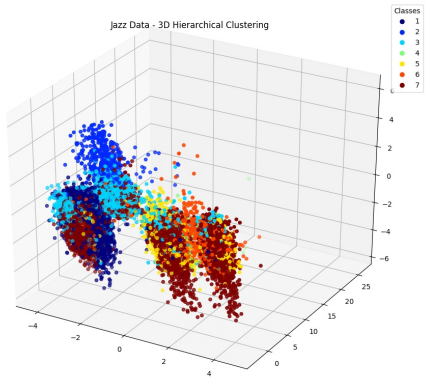
Hierarchical Clustering



Hierarchical Tree

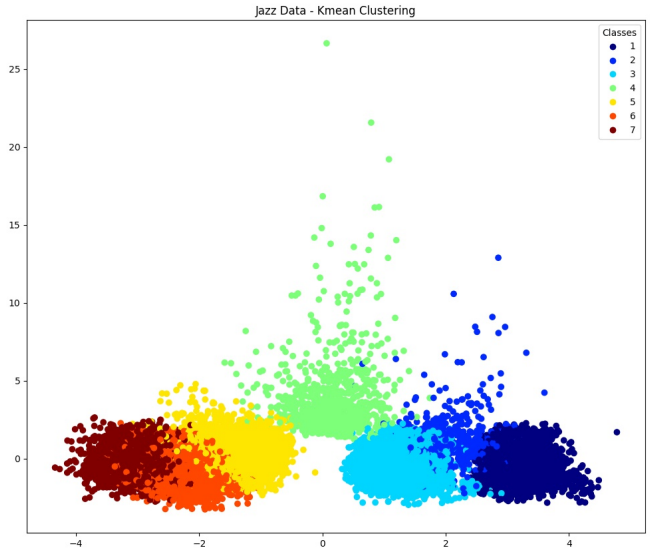


Hierarchical 2D scatter plot

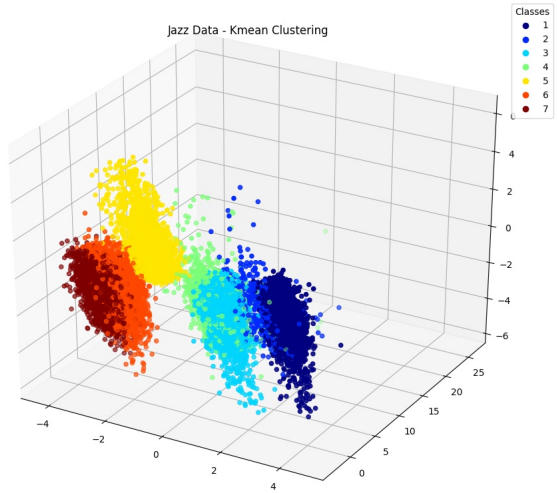


Hierarchical 3D scatter plot

K-mean Clustering

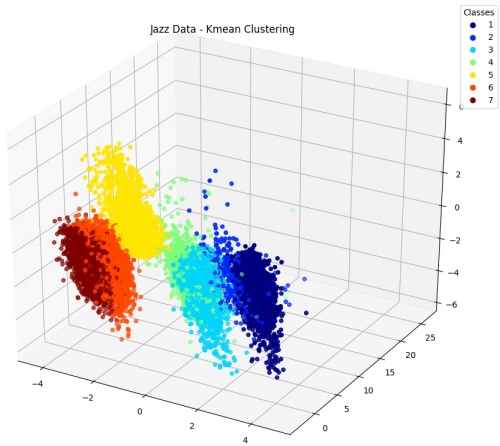


K-mean 2D scatter plot

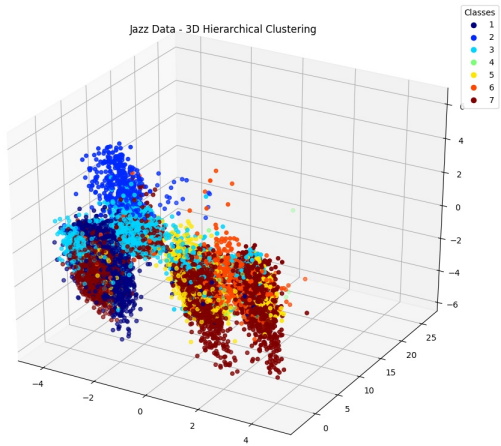


K-mean 3D scatter plot

From previous plots, we can clear see that K-mean could better deal with clustering in this problem.



K-mean 3D scatter animation



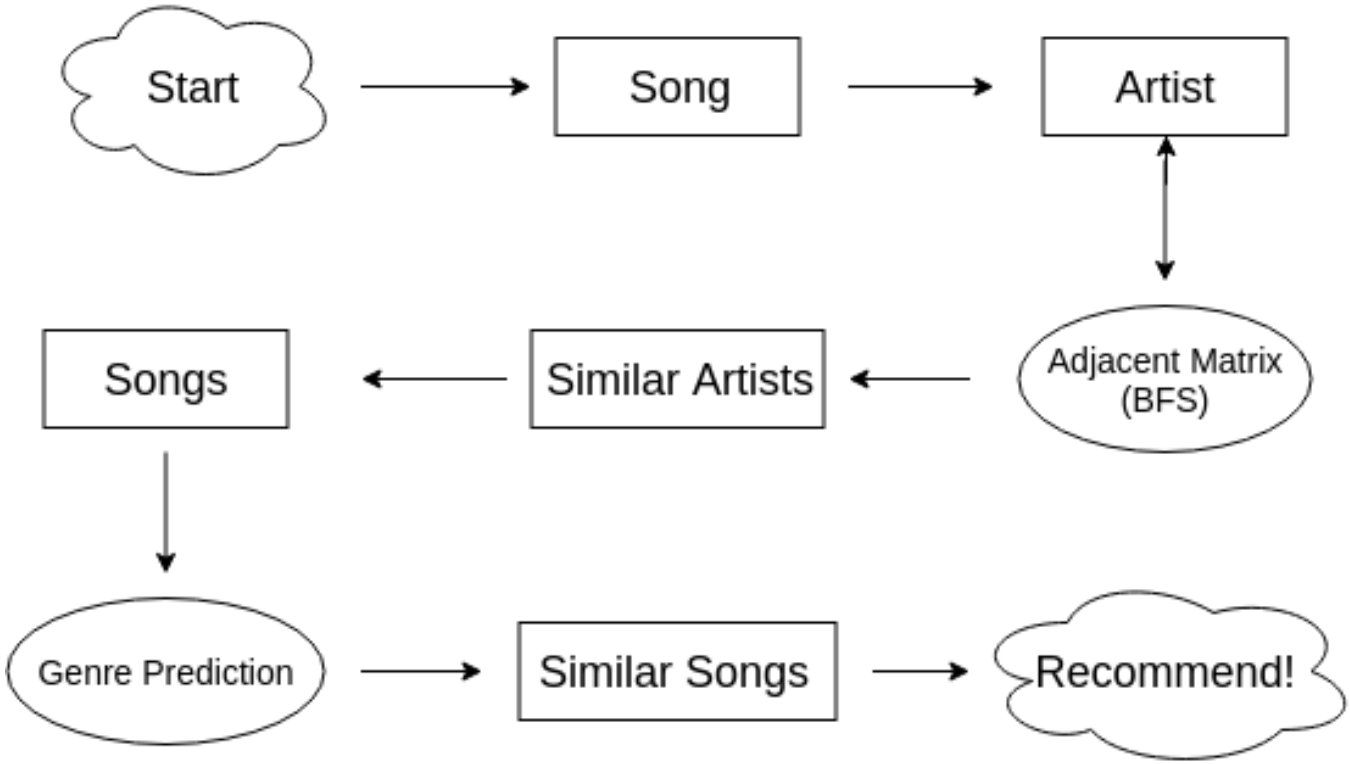
hierarchical 3D scatter animation

Drawbacks and Corresponding Solution

Drawbacks and Corresponding Solution

Since we may have some hard boundary in K-mean clustering some obscure data points may have multiple attributes. We decide to provide a granularity selection for user so that they can find more specific genre or more general songs

Diagram



Offer variations:

- The depth of BFS
- The number of class for the classifier

Demo


ARZSLJK1187FB55341
Depth=2
ARRJPSI11C8A42B77B
AR1OFHY1187FB4E141
ARWPADL12454A51F1C
ARTWBUH12454A43277
ARA5S1O1187FB3F6F3
...
SOWKHVU12A58A7A447
SOGEUYD12AB017DBAC
SOICBGQ12AB017F223
SOZXZYL12A6D4FB6E6
SOAKSJQ12AC4689697
...
SONSKNZ12AB0185BD6
Genre=7
SONSKNZ12AB0185BD6
SOIIJPV12A8C1333A7
SOHUOMZ12A58A78BA2
SOKXWRL12AB0187FE7
SOQELPT12AC3DFA040
...

Contents

Discussion

Remaining Issues

- Inefficient storing of data for adjacency matrix
- BFS can eat up memory easily

Further Improvements

- Other algorithms to build adjacency matrix
- Combine with user behavior

Contents

Conclusion

We successfully build a song recommendation system based on clustering.

Contribution

- Avro to hdf5 to csv:
  Yuan Gao
- BFS MapReduce:
  Chong Hu
- BFS Spark:
  Xining Wang
- Genre Prediction:
  Chong Hu, Xining Wang, Weijie Ye, Yuan Gao
- Drill:
  Weijie Ye
- poster:
  Yuan Gao, Weijie Ye
- ppt:
  Yuan Gao, Weijie Ye, Xining Wang, Chong Hu

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.