

# **Music Recommendation**

**Group: RUC**

**Group Member: Tongpeng Zhang, Ao Guo, Junjie Feng**

# 1.SUBJECT INTRODUCTION:

1. Automatic music recommendation has become an increasingly relevant problem in recent years.
2. Our project is based on ALS algorithm of spark.
3. After model training, our algorithm will give 5 recommendation products.
4. Also we could give a result of which artists that user have listened before.

## 2.PROBLEM STATEMENT:

1. Most music recommendation systems implement content-based recommendation algorithm.
2. Our Algorithm implement Collaborative filtering recommendation using ALS.
3. We choose implicit feedback instead of explicit feedback.

### 3.COMPARISON:

a1.Previous work needs get the exact rating of the music

a2.Our algorithm only needs get how many times that user have listened of the artists.

b1.Previous work inputs and read whole data file

b2.If the source is very huge, Our algorithm could split the whole data to some parts,and implement independently.

# COMPARISON:

c1.To one same artist who has different ID, previous work  
Solves independently.

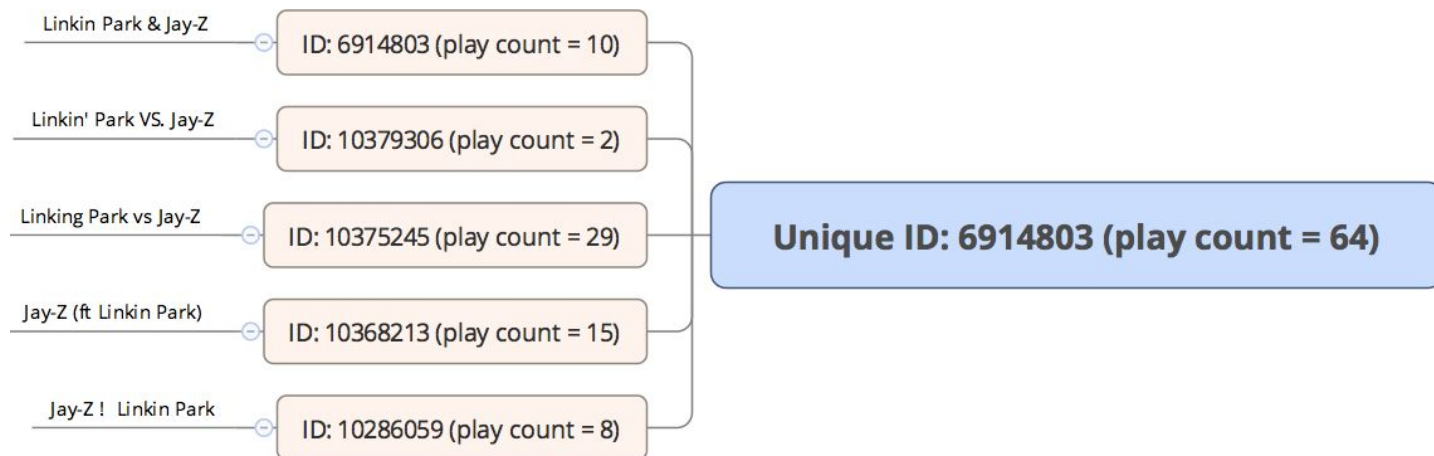
c2.Our algorithm will check all same artist with different  
ID, and put to a new same ID.

(About the size of our data: we get this data package from  
open sources website. The whole file includes 141000  
users,1.6 millions artists, and 24.2 millions music played  
records.)

# 4.1 OUR CONTRIBUTION

Clustering User & Artist --Merge the several artist alias ID into one unique ID.

- a. Increase the play count of 1 artist  $\Rightarrow$  increase the recommendation precision.



# RESULT:

1. Choose user ID 2093760 as an example, recommend 4 artists for him.

Without merger

(2093760,1300642,0.026266728)

(2093760,1308254,0.023869438)

(2093760,1114670,0.022670428)

(2093760,1046450,0.022248158)

With merge,rank=10, $\lambda$ =0.012,iteration=5

(2093760,1300642,0.026453131)

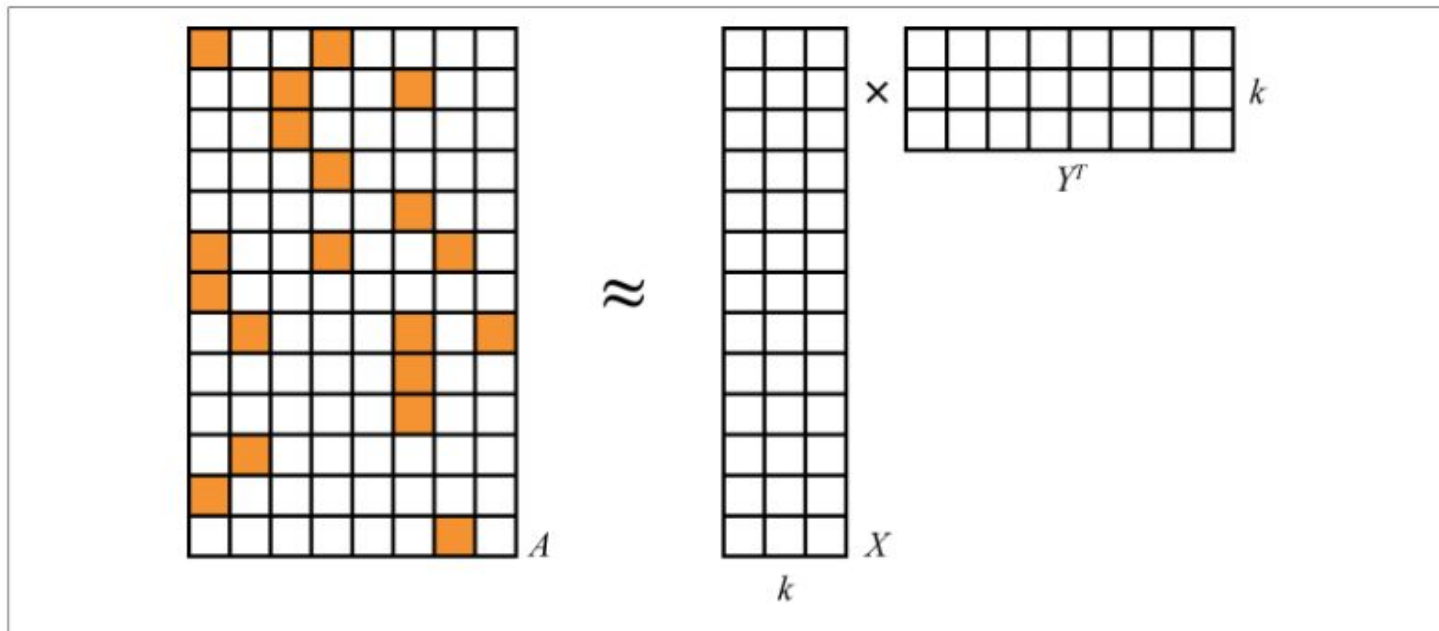
(2093760,2814,0.0262199577)

(2093760,1037970,0.025403633)

(2093760,1001819,0.0251077078)

## 4.2 OUR CONTRIBUTION

Adjust the input parameters of ALS(collaborative filtering algorithm) training function





## 4.2 OUR CONTRIBUTION

Adjust the input parameters of ALS(collaborative filtering algorithm) training function

```
Val model = ALS.trainImplicit(trainData, rank, Iteration, $\lambda$ ,alpha)
```

Rank: number of implicit features to use (default: 10)

$\lambda$ : regularization factor (default: 0.12)

Alpha: confidence parameter (default: 1)

Iteration: numbers of iteration (default: 5)

# RESULT:

Setting ALS parameters to different value:

rank=10

rank=12

(2093760,1300642,0.026266728)

(2093760,2814,0.0332615202)

(2093760,1308254,0.023869438)

(2093760,1811,0.0333170304)

(2093760,1114670,0.022670428)

(2093760,1001819,0.0334348967)

(2093760,1046450,0.022248158)

(2093760,1300642,0.0344533654)

The higher the rank, the better the recommendation precision

# RESULT:

Setting ALS parameters to different value:

$\lambda = 0.012$

$\lambda = 0.010$

(2093760,1300642,0.026266728)

(2093760,1001819,0.031332741)

(2093760,1308254,0.023869438)

(2093760,2814,0.0311111766)

(2093760,1114670,0.022670428)

(2093760,4605,0.0307145103)

(2093760,1046450,0.022248158)

(2093760,1811,0.0301325664)

When  $\lambda=0.01$ , the better the recommendation precision

# RESULT:

Setting ALS parameters to different value:

iteration=5

iteration=10

(2093760,1300642,0.026266728)

(2093760,1015537,0.028112695)

(2093760,1308254,0.023869438)

(2093760,1244723,0.027193579)

(2093760,1114670,0.022670428)

(2093760,5209,0.027075033716)

(2093760,1046450,0.022248158)

(2093760,5209,0.027075033716)

The more iterations ALS runs, the better the recommendation precision

## 4.3 OUR CONTRIBUTION

3. Since the raw dataset has some error line, we add code to check them and ignore error line, in order to improve the recommendation precision.

The improvement is too small.

4. The raw dataset has 24 millions records, Then spark could break the dataset into blocks and run these block in parallel in order to speedup the running time.

Previous: about 15 min;    Now: about 14 min (when we split it into 2 blocks, and the other conditions are not changed)

# FUTURE PROSPECTS

What we have done is to improve the recommendation precision. For future, we will make effort to make the program run faster.

1. Optimize ALS algorithm in order to reduce total number of Cartesian product computation. modifying the spark ALS source code.
2. Improve the speed of the file load.