Yahoo Music Recommendation System

Zhibo Yang

Under the guidance of

Professor Mr. Rensheng Wang

Abstract

Big data problem were extensively studied in the last few years. As one of big data problem, recommendation system is one popular problem. There are also variety of recommendation systems such as movie recommendation system and shopping recommendation system. Here what we constructed is a music recommendation system based on Yahoo! music dataset. Yahoo! Music dataset consists of more than 40 thousand users' histories and 10 thousand ratings. Rated items are multi-typed, including albums, artists, and genres. Albums and artists are kind of fixed which means one item belongs to only one album and one artist. However, genre is relatively flexible, since each item could belong to even tens of genres.

We first simply add album ratings and artist ratings together to get predictions for users. Then we take genre ratings for account.

We also tried matrix factorization which is a commonly used recommendation method. Matrix factorization treated dataset as matrix and separate it into two matrices. Then by multiplying these two matrices together, we finally get our prediction result.

After getting 10 results from above methods, we introduced ensemble method to our project. This method combines all the results together using certain algorithm. By applying ensemble method, the result was enhanced a lot.

Introduction

Programming Language:

Python spark.

Mostly python, partly Python Spark

Data Preparation:

We first use data: 1. trackData2.txt; 2. testIdx2.txt; 3. testTrack_hierarchy.txt to generate a hierarchy structure and match each item with a rating.

This hierarchy structure is like:

User ID | item ID | Album | Artist | Genre1 | Genre2 | Genre3 | ……

1 | 188135 | None | None | None | None | None | None
1 | 250273 | 90 | 90 | None | None | None | None| None | None
1 | 60248 | None | 90 | None | None | None | None | None
1 | 187953 | 90 | 90 | None | None | None | None | None
1 | 108088 | None | None | None | None | None | None
1 | 52615 | 90 | 90 | None | None | None

…… …… …… …… …… …… …… …… …… …… …… …… ……

38465 | 245719| 90 | 90 | 80| None | None | None | None

38465 | 54238 | 90 | 90 | 80| None | None | None

38465 | 228598 | None | None | None | None | None

38465 | 217068 | 90 | 90 | None | None | None

38465 | 257173 | None | None | None

38465 | 31358 | None | None | 80 | 80 | None | None | None

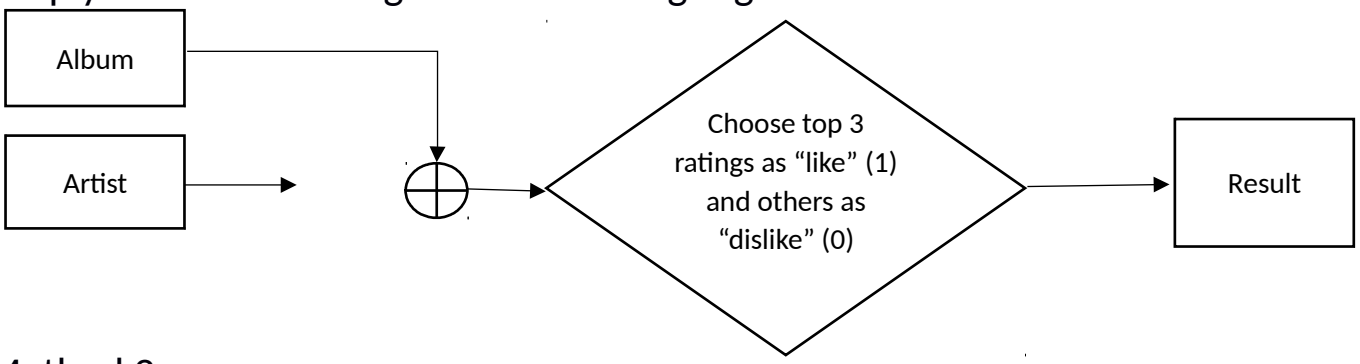
This structure was saved in test_raw_score.txt file.

Methods

This hierarchy structure could be easily used to do prediction using following methods:

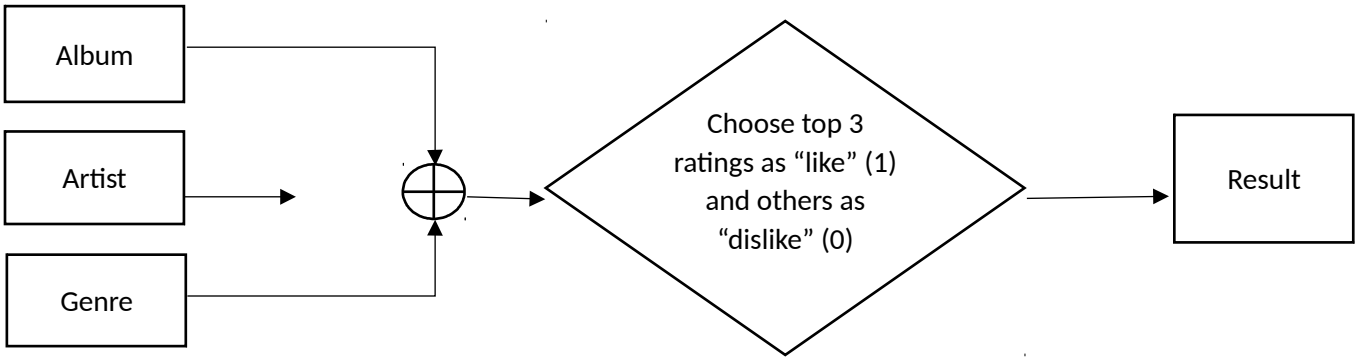
Method 1:

Simply add album rating and artist rating together



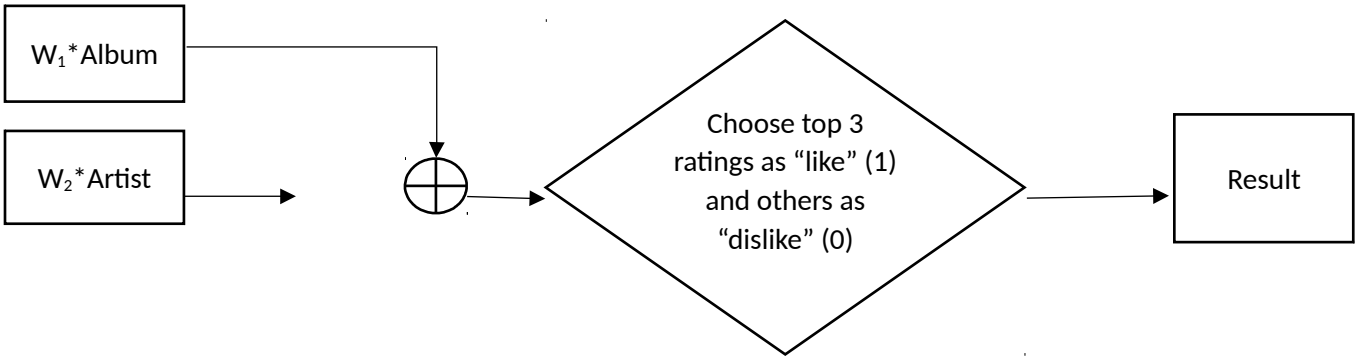
Method 2:

Add album rating, artist rating, and mean of genre ratings together



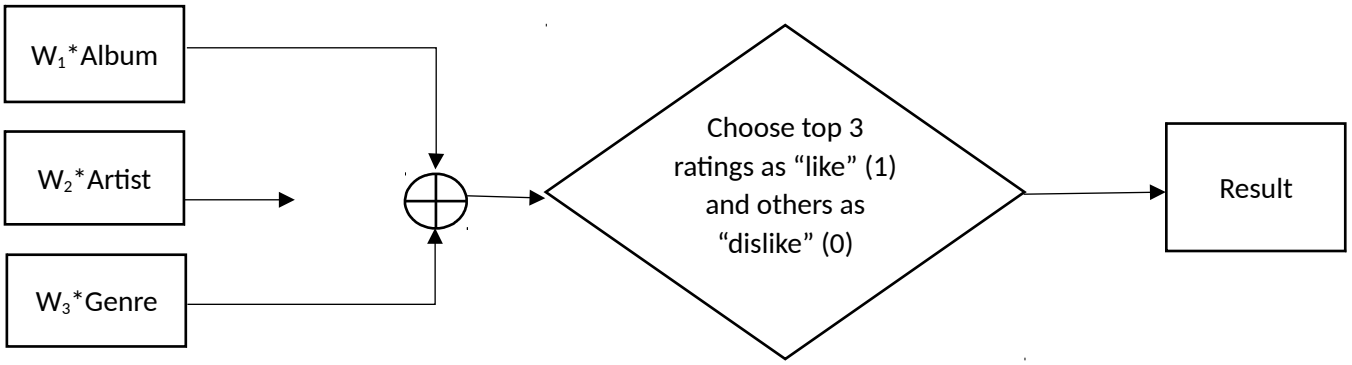
Method 3:

Based on Method 1, assign album and artist with different weights, then add them together.



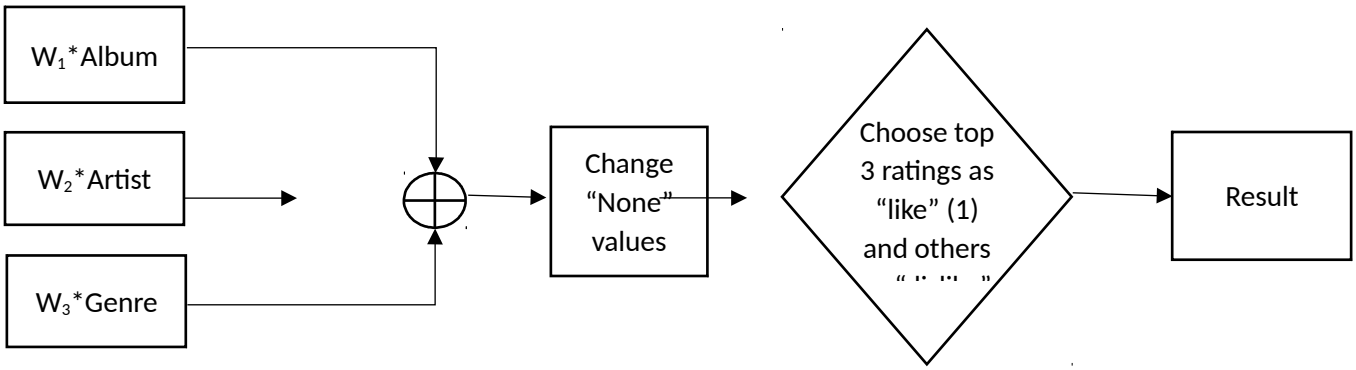
Method 4:

Based on Method 2, assign album, artist, and genre with different weights, then add them together.



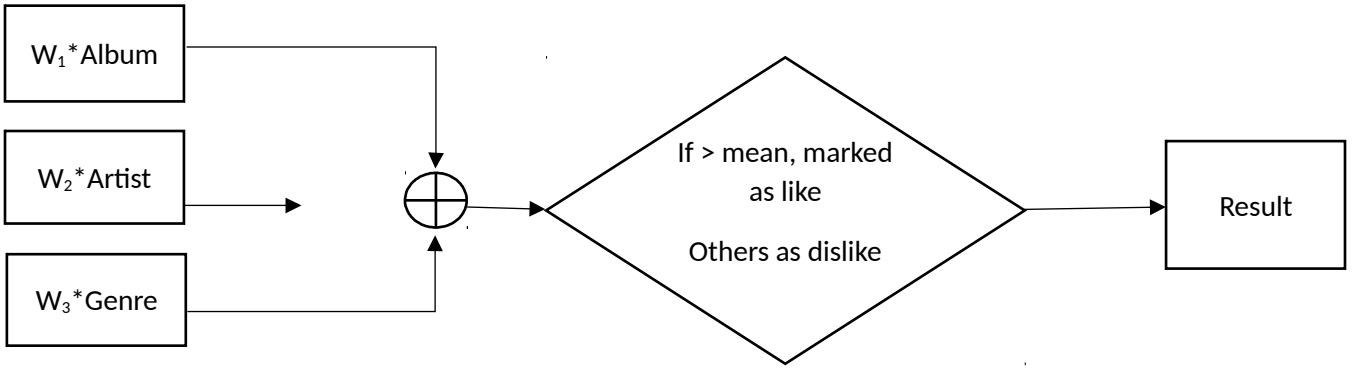
Method 5:

Based on Method 1, Method 2, Method 3, and Method 4, replace “None” with different values.



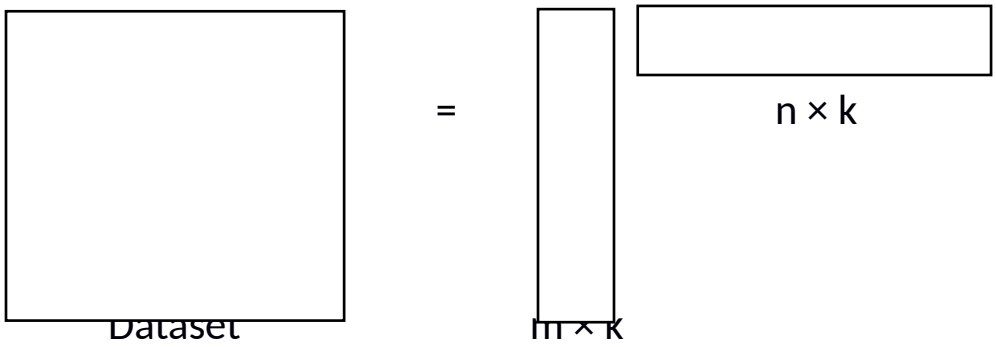
Method 6:

Based on Method 1, Method 2, Method 3, and Method 4, calculate means of user’s ratings instead of mark top 3 rating as like and others as dislike.



Method 7:

Matrix Factorization



Here we need to explain more about this method.

Since we already have one m × n matrix, first we separate it into two part, and then multiply the two parts together.

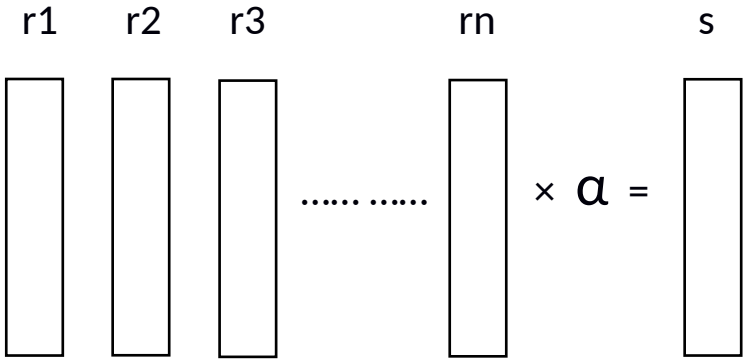
As this m × n matrix is a sparse matrix and not every position have a value,

So by doing matrix factorization, we can exactly restore the ones who already have a value and predict the ones who do not have a value.

This method was commonly used in recommendation field.

Method 8:

Ensemble all the results obtained from above methods.



Here r1, r2 …… rn are different test results obtained from different methods. S is the combined test result. A is parameters for each result.

In this ensemble method, we first need to replace “0” with “-1” in r1, r2….. rn, and then multiply all test results together. This step will help us to make sure which predictions are always same in variety of methods.

From class, Prof. Wang said that in order to calculate α here, we use the following formula:

$$\alpha = \left(r^T r\right)^{-1} r^T s$$

Where

$$r^T s = 2 * \left(correct\, rate - \frac{1}{2} \right) * N$$

Where

N is the total number of items in r (or number of rows in r).

Result

Method	Correct Rate	Test Result File
Method 1	0.8046	TestResult1
Method 2	0.8335	TestResult2
Method 3.1	0.8498	TestResult3
Method 3.2	0.8465	TestResult4
Method 3.3	0.8495	TestResult5
Method 4.1	0.8495	TestResult6
Method 4.2	0.8498	TestResult7
Method 4.3	0.8465	TestResult8
Method 5.1	0.8450	TestResult9
Method 5.2	0.8060	TestResult10
Method 5.3	0.8500	TestResult11
Method 6.1	0.8706	TestResult12
Method 6.2	0.8664	TestResult13
Method 7.1	0.6734	TestResult14
Method 7.2	0.7024	TestResult15
Method 8	0.8791	EnsembledResult

Note: Here are just part of our result. Because we have tried much more parameters than what we have listed here, and we cannot list them all here. But I will put all my results as an attachment in project submission.

Result Analysis:

I will analyze all the results listed here.

Method 1: This method just simply add artist and album ratings together.

Method 2: Add artist, album, and means of genre ratings together.

Method 3.1: w_1=1.2, w_2=0.8

Method 3.2: w_1=0.8, w_2=1.2

Method 3.3: w_1=1.2, w_2=1.1

By assigning above 3 pairs of weights, we find the result would be better if we choose $w_1 > w_2$. This means, in practice, album rating is more important than artist rating in music recommendation problems.

Method 4.1: w_1=1.2, w_2=1.1, w_3=0.5

Method 4.2: w_1=1.2, w_2=0.8, w_3=0.4

Method 4.3: w_1=0.8, w_2=1.2, w_3=0.6

Here we have tired exactly far more weights of genre, but it seems weights do not play as an important role as album and artist. By changing weights of genre, the result actually do not change a lot. Therefore we just pick up the same weights of album and artist as those in Method 3.

Method 5.1: replace “None” with 0

Method 5.2: replace “None” with 50

Method 5.3: replace “None” with -35

By replacing “None” with different values, we find when we replace “None” with -35, the result have best correct rate. Although this kind of replacing do not seem rational, but it really works. And this make me feel sometimes the parameters which have real meanings do not really match to its real meanings. As long as it works, the parameters are good parameters.

Method 6.1: w_1=1.2, w_2=1.1, w_3=0.5

Method 6.2: w_1=1.2, w_2=0.8, w_3=0.5

In Method 1—5, we just simply mark top 3 ratings as “like” and others as “dislike”. While in Method 6, we use users’ means of ratings to judge if we should mark it as “like” or “dislike”.

Here, we mark ones who are larger than users’ mean as “like” and ones who are smaller than users’ mean as “dislike”.

The result by applying this method enhanced a lot from previous methods. Also we have tired far more parameters than what we listed here. We will also attach them all together as attachment.

Method 7.1: Rank = 8 Iteration = 10

Method 7.2: Rank = 8 Iteration = 20

Theoratically, matrix factorization is one of the best methods in recommendation field. But here in our project, the results are not good enough.

Since our music dataset is really large and is actually a sparse matrix (which means most of the matrix are empty), there are too many values need to be predicted. This will definitely cause inaccuracy of results.

Hence, in order to improve the efficiency of matrix factorization, we need to do some more preprocessing work on our training dataset such as reducing dimension of matrix and ignoring items who do not have any values.

Method 8: This method combines all the previous results together in the way we have introduced in previous section.