COMP9417-Assignment 2

Machine Learning and Data Mining

Project 3.4 Collaborative Filtering Recommendation System

Group member:

z5147046 Meiyan Pan

z5149155 Qian Cheng

z5124474 Yichen Zhang

1. Introduction

Collaborative filtering is a typical way to utilize collective intelligence and it has been widely used in many fields nowadays e.g. Amazon automatically recommends books that users might be interested in based on users' purchase and browsing history.

The goal of the project is to learn to predict the ratings of movies on the data from the GroupLens research group : MovieLens datasets ml-100k. Three models(Memory based) are built to train the data :item-based model ,user-based model and item-user-based model. NearestNeighbors is applied to find k most similar users or movies of the target user or the movies this user has watched.

2. Method

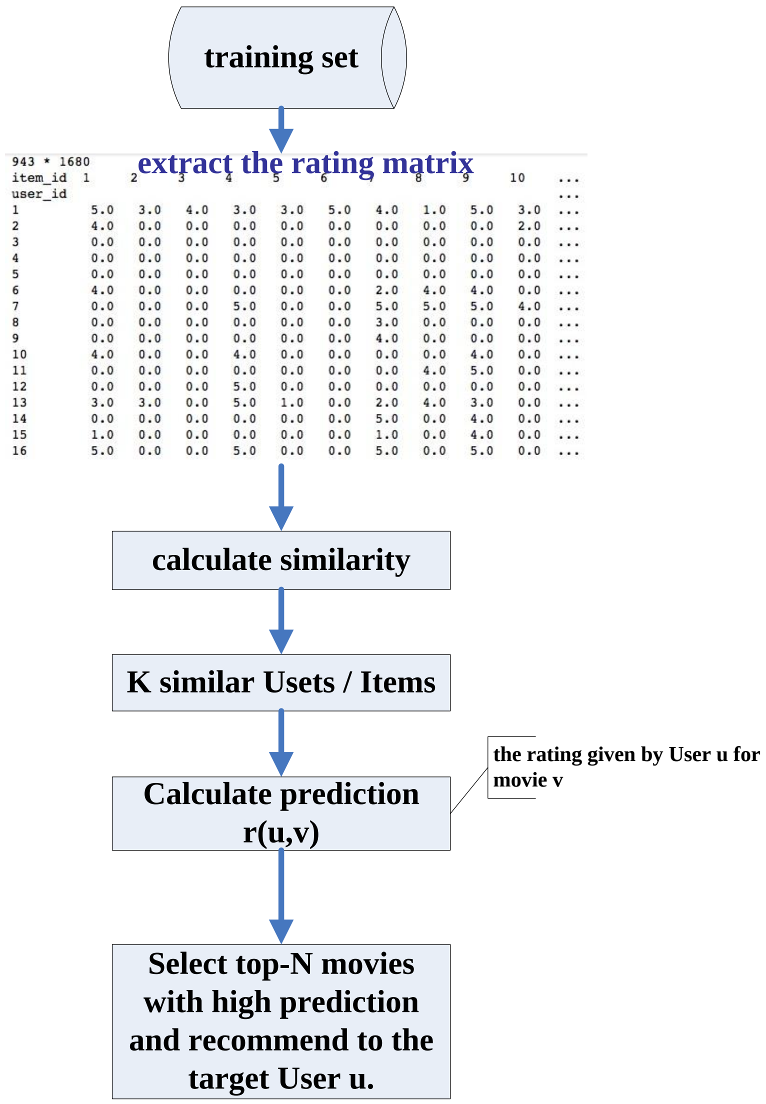

Figure 2.1 The flow diagram of the Recommedation System

2.1 Similarity and K-neighbors

In general, we applied three kinds of coefficients to describe the similarity degree:

1 Cosine similarity

$$\mathbf{T}(\mathbf{x},\mathbf{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|^2 \times \|\boldsymbol{y}\|^2} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

2 Pearson Correlation Coefficient

$$\mathbf{p}(\mathbf{x},\mathbf{y}) = \frac{\sum x_i y_i - n\overline{xy}}{(n-1)s_x s_y} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - \left(\sum x_i\right)^2}\sqrt{n\sum y_i^2 - \left(\sum y_i\right)^2}}$$

3 Tanimoto Coefficient(Jaccard)

$$\mathbf{T}(\mathbf{x},\mathbf{y}) = \frac{\boldsymbol{x}\cdot\boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}\cdot\boldsymbol{y}} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} + \sqrt{\sum y_i^2} - \sum x_i y_i}$$

For users, greater similarity indicates that user_x and user_y have more similar preferences. And for items, greater similarity means that item_x and item_y may have similar characteristics and are potentially attractive to same users.

Based on the similarity, we applied the K-NN algorithm in this project to find the nearest neighbors of the current user or item.

2.2 User-based recommendation

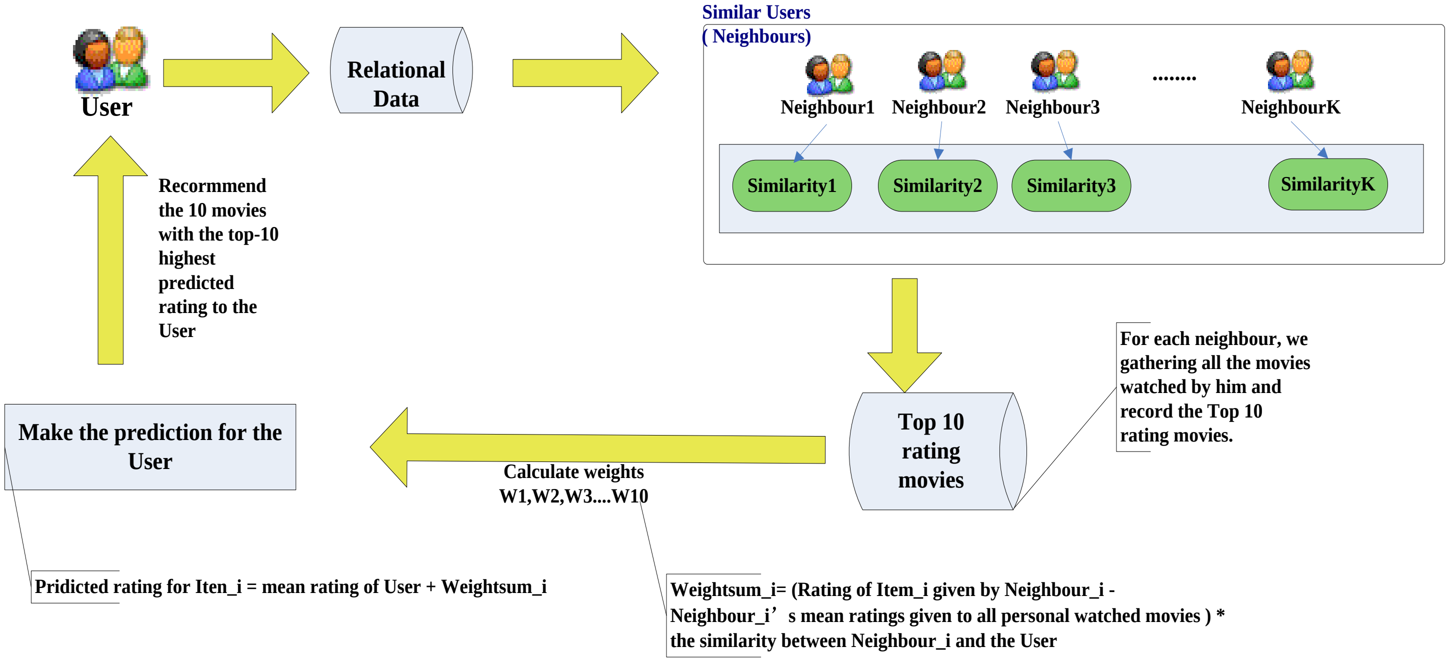

Figure 2.2 The flow diagram of the User-based recommendation process

In user-based recommendation, if we want to make the recommendation to user_u the overall processes are roughly as follows:

Step 1: Find K nearest neighbours (user_i, where i =1,2,3,...K) and record the similarities as well.

Step 2: For each neighbour user_i, find the top-10 rating movies wached by user_i

Step 3: Calculate the weightsum as follows,

weightsum_i= (Rating of movie_i given by user_i – User_i's mean ratings)* Sim(user_u,user_i)

Step4: Make prediction for movie_i by the forluma:

predicted rating = mean rating of user_u + weightsum.

Step 5: The recommendation is the top-10 movies with highest predicted rating.

2.3 Item-based recommendation

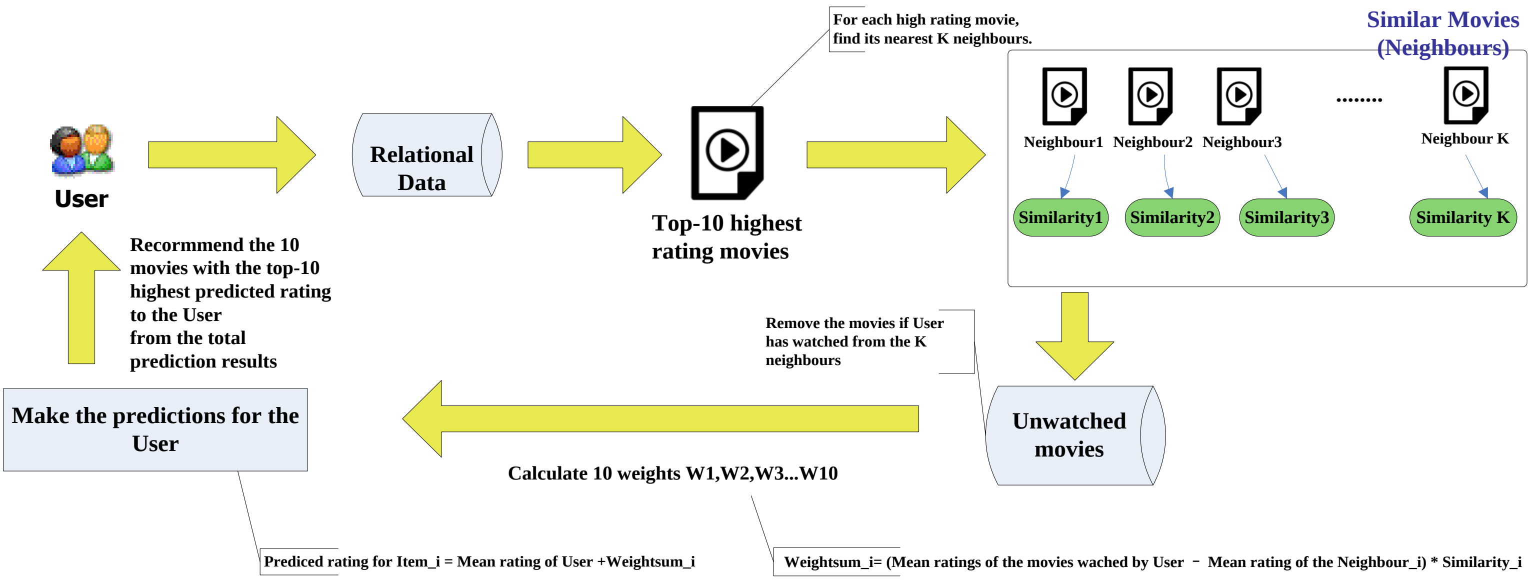

Figure 2.3 The flow diagram of the Item-based recommendation process

Item-based recommendation is based on the similarity between items. Following is the description of the process , if we want to recommend 10 movies to user_u:

Step 1: Find top-10 rating movies (movie_i, where i=1,2,3...10) of user_u.

Step 2: For each movie_i , find its K nearest neighbours (movie_j, where j =1,2,3...K) and record the similarities.

Step 3: Remove the movie_i if it has been wached by user_u.

Step 4: Calculate the weightsum as follows,

weightsum_i= (User_u's mean ratings – Mean ratings of movie_j)* Sim(movie_i,movie_j).

Step 5: Make prediction for movie_i by the forluma:

predicted rating = mean rating of user_u+ weightsum.

Step 6: The recommendation is the top-10 movies with highest predicted rating.

2.4 User-Item-based recommendation

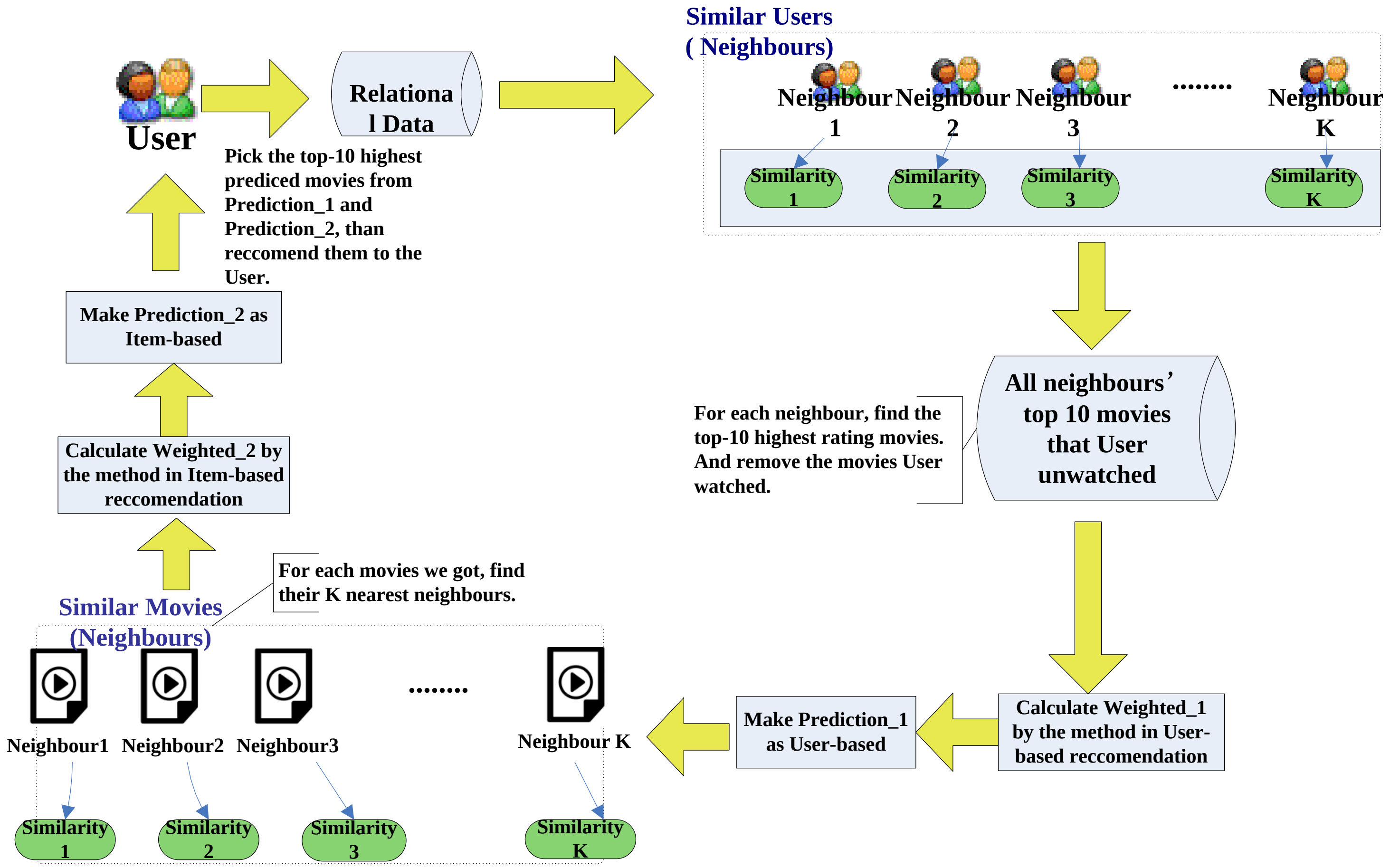

Figure 2.4 The flow diagram of the User-Item-based recommendation process

It is a complex type recommendation model, the operation method fuses the above two models. First, get Prediction_1 by the user-based recommendation. Then, get Prediction_2 by the item-based recommendation. Finally, work out the predicted result combining both Prediction_1 and Prediction_2. The details are shown in **Figure 2.4**.

3.Results and Discussion

3.1 evaluation metric

Mean Absolute Error (MAE) are selected as the evaluation metric as it is one of the most common metrics used to measure accuracy for continuous variables.

$$\mathbf{MAE} = \frac{1}{n}\sum_{j=1}^{n}\left|y_j - \hat{y}_j\right|$$

$\hat{y}_j$ -- is the true rating.

y_j – is the corresponding prediction by the model.

A smaller MAE indicates better recommendation quality.

3.2 results and discussion for “ua” dataset

“ua.base” is used as the training set which has 943 users and 1680 movies in total.
“ua.test” is used as the testing set which has 943 users and 1129 movies.

1 Experiments with similarity method:

The results of User-based MAE distribution for different similarity calculation methods are in **Figure 3.1**.

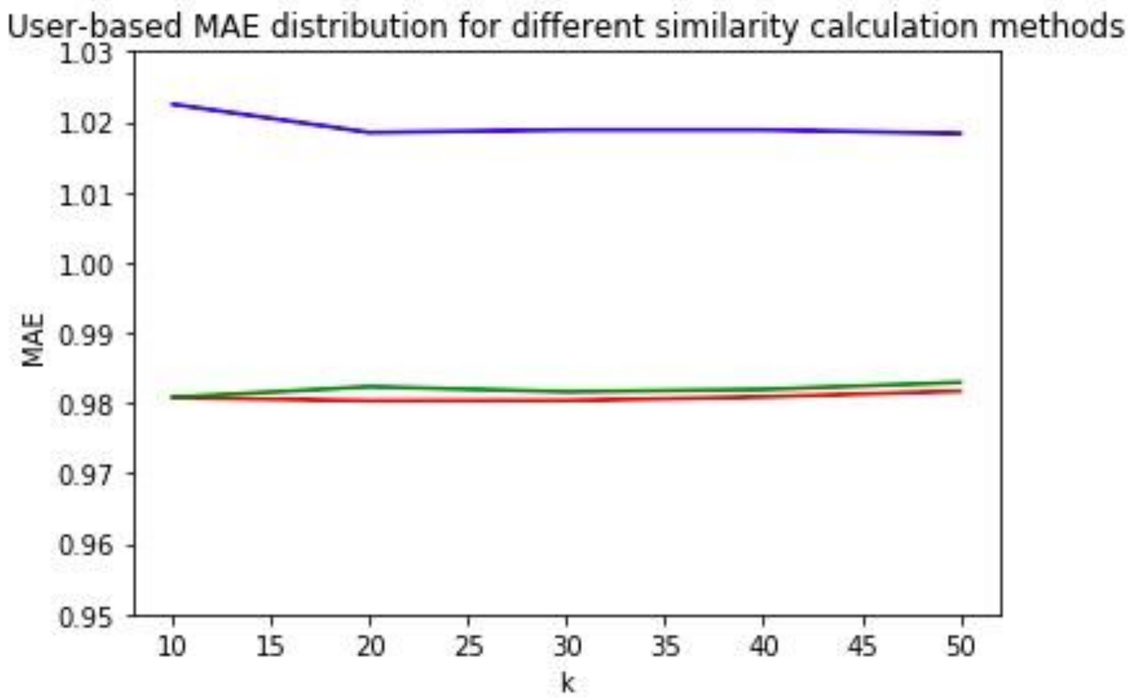

Figure 3.1

(blue - Tanimoto Coefficient(Jaccard) ; green - cosine similarity ; red - Pearson Correlation Coefficient)

As can be seen, “Pearson Correlation Coefficient” has the greatest performance and “Tanimoto Coefficient” is not as accurate as the other two in this case. So we select “Pearson Correlation Coefficient” for the system eventually.

2 Experiments with recommendation method:

The results of MAE distribution for three models are in **Figure 3.2** .

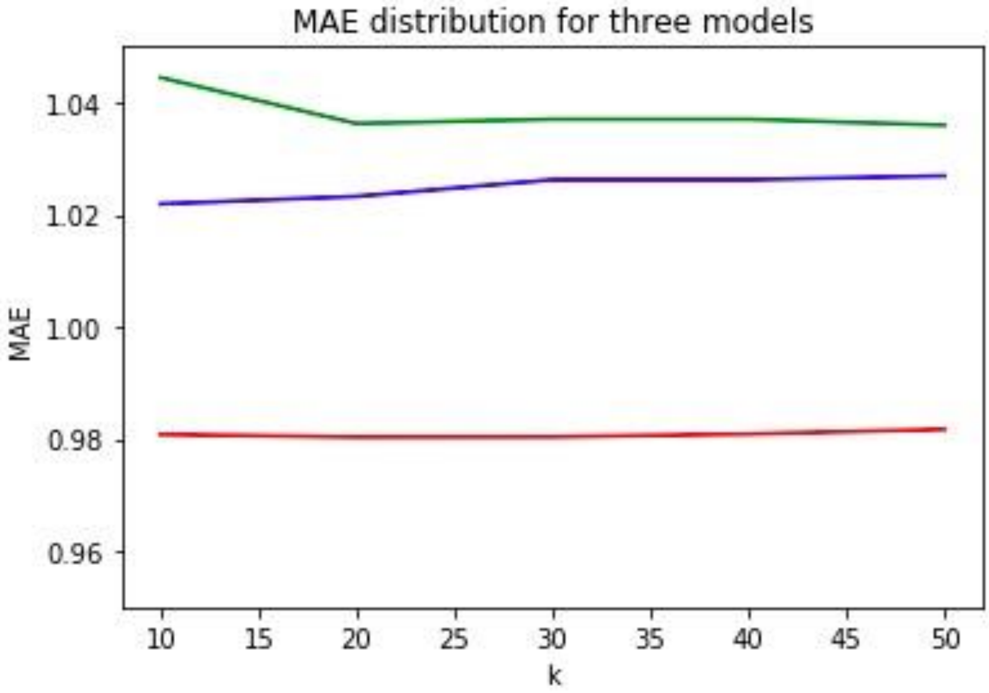

Figure 3.2
(blue-user-item-based ; green-item-based ; red- user-based)

As can be seen, user-based recommendation has the best performance and item-based recommendation is less accurate. As what we expected, user-item-based recommendation should beat the other two. However, in reality, it is not as efficient as the other two and the performance does not worth the trade-off.

3 Experiments with k (number of neighbors):

The results of User-based MAE correlation distribution as k increases are in **Figure 3.3**.

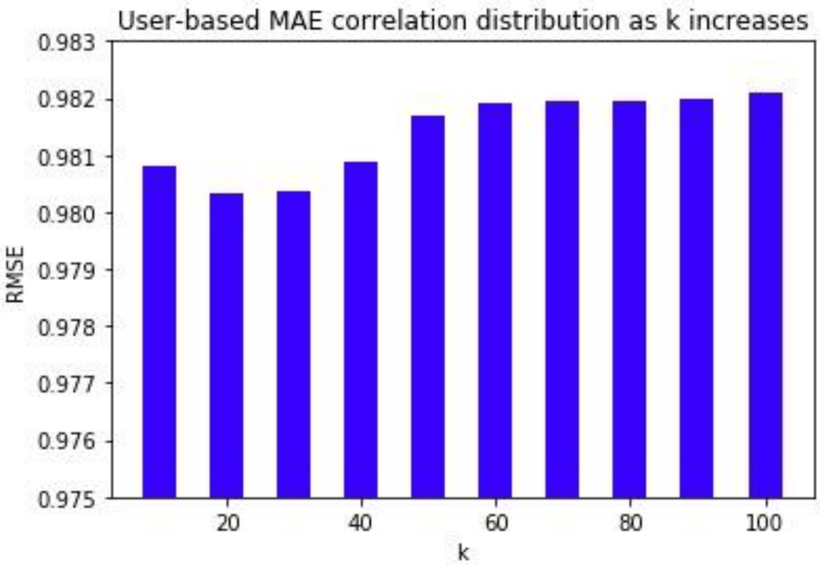

Figure 3.3

The results of Item-based MAE correlation distribution as k increases are in **Figure 3.4**.

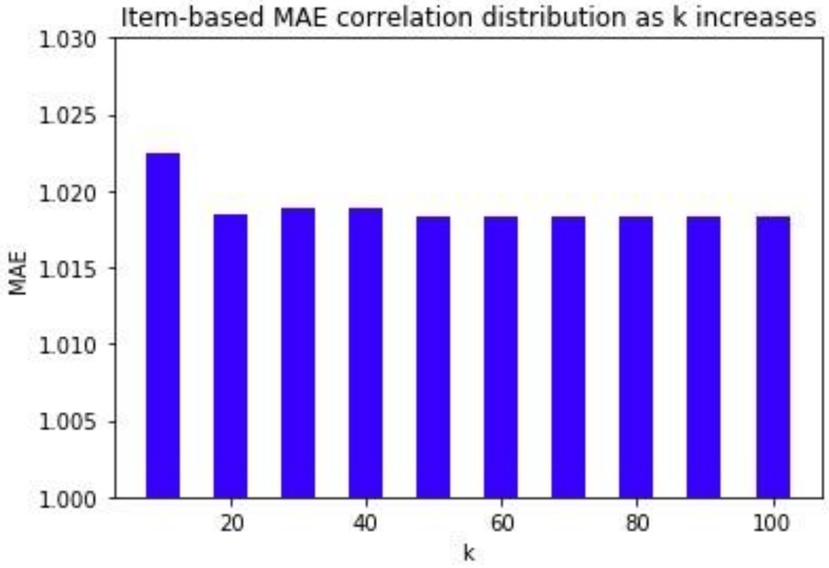

Figure 3.4

The results of User-Item-based MAE correlation distribution as k increases are in **Figure 3.5**.

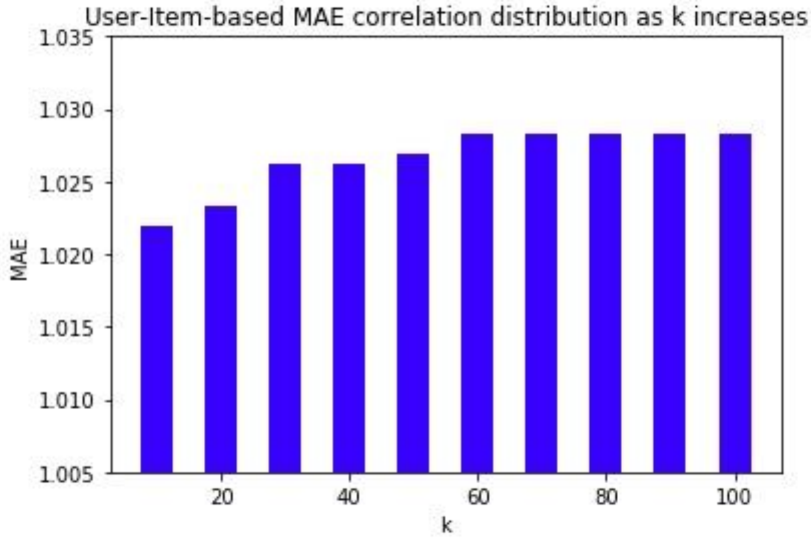

Figure 3.5

As can be seen from Figure 3.3 and Figure 3.4, the best performance occurs when k=20. And in Figure 3.5, k=10 leads to better quality of recommendation.

4 Experiments with N (number of movies recommended to user):k=10

The results of User-based MAE correlation distribution as N increases are in **Figure 3.6**. As can be illustrated, less error occurs as N increases.

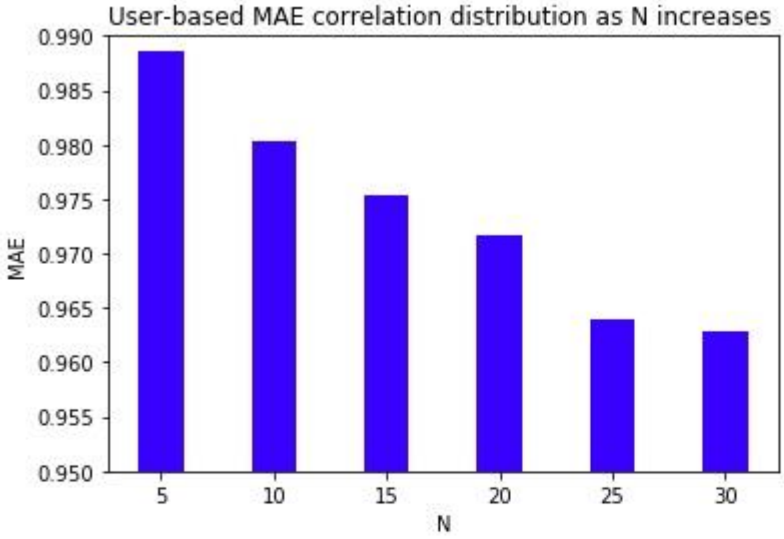

Figure3.6

3.3 Output example

1 User-based recommendation result

```
predict_users(u,ubase1,items)  #k=10
#3.688888889 is the mean rating of user 344
```

```
reccomend 10 movies for user 344:
predicted_rating   movieId
3.68888888889
```

```
[[4.0252838358965306, 271],
 [4.0252838358965306, 301],
 [3.9873016270485473, 1],
 [3.9873016270485473, 2],
 [3.9873016270485473, 28],
 [3.9873016270485473, 48],
 [3.9873016270485473, 50],
 [3.9873016270485473, 58],
 [3.9873016270485473, 64],
 [3.9873016270485473, 77]]
```

2 Item-based recommendation result

```
u=345#item based
predict_items(10,u,ubase1,items)
```

```
[[4.3139016116264948, 6],
 [4.3028090550308269, 120],
 [4.252082760160965, 49],
 [4.2298041980132961, 233],
 [4.223806624038553, 21],
 [4.2117315638013801, 78],
 [4.1963318969990357, 63],
 [4.1852059945123461, 236],
 [4.175326039329514, 180],
 [4.1683101924714183, 10]]
```

3 User-item-based recommendation result

```
reccomend 10 movies for user 98:
predicted_rating   movieId

[[5, 60],
 [5, 59],
 [4.9168058742211187, 120],
 [4.9143549174556211, 6],
 [4.9075526065995714, 545],
 [4.8931902183465974, 264],
 [4.877106928495504, 233],
 [4.8688698284149492, 407],
 [4.8582070270941262, 188],
 [4.8297529999662911, 267]]
```

4.Conclusions

From the results it can be concluded that memory based collaborative filtering recommendation especially user-based and user-item-based method generates high quality recommendations. And the performance improves as the number of neighbours(k) and the number of recommended movies to each user(N). Besides,“Pearson Correlation Coefficient” turns out to be the most suitable method for calculating the similarity for both users and items in this case.

5.References

Yichang.“Collaborative filtering recommendation based on item rating and characteristic information prediction”.China,2012

R. M. Bell and Y. Koren. “Scalable collaborative filtering with jointly derived neighborhood interpolation weights”. InIEEE ICDM, 2007.