UNSW
AUSTRALIA

COMP9417

Machine Learning and Data Mining

Assignment 2

Recommender system using collaborative filtering

Group Name: **wam84.4**

Group Member:

Zhi He	z5123906
Hao Fu	z5102511
Shengtao Xu	z5099565
Bowen Fan	z5137961

Abstract

The recommendation system has been evaluated in many unparalleled ways. In this article, we reviewed the key decisions for collaborative filtering recommendation systems : the users task being evaluated, the type of analysis, the dataset being used, the way in which forecast quality is measured, and the user-based evaluation of the system as a whole.
Collaborative filtering (CF) is the process of filtering or evaluating items through the opinions of other people. CF technology brings together the opinions of large interconnected communities on the web, supporting filtering of substantial quantities of data. In this chapter we introduce the core concepts of collaborative filtering and design decisions regarding rating systems and acquisition of ratings. We also discuss how to evaluate CF systems. We close the chapter with a conclusion.

General Terms: Measurement, Performance
Key words: Recommender systems, Collaborative filtering, Metrics, Evaluation, Pearson Correlation

1. Introduction

In recent years, E-commerce markets have actively promoted automation personalized services to analyze customer behavior and purchase factor. As a successful technology application,recommender systems use the opinions of a community of users to help individuals in that community more effectively identify content of interest from a potential overwhelming set of choices.The most popular technique it used is collaborative filtering.
Collaborative filtering and its modifications is one of the most commonly used recommendation algorithms. While the term collaborative filtering has only been around for a decade, CF takes its roots to share opinions with others. For years people have stood over the back fence and discussed books they have read, movies they have seen and restaurants they have tried , then used these discussions to form opinions. For example, when enough of Michael's friends say they liked the latest released movie, he might decide that she also should to see it. Better yet, Michael might observe that Ben recommends the types of films that he finds enjoyable, Peter just seems to recommend everything. So he learns whose opinions should be listen to and how these opinions can be applied to help her determine the quality of an item.

2. Recommendation filtering techniques

There are two types of collaborative filtering : User-based collaborative filtering and Item-based collaborative filtering. Each methods approach different solutions effectively. User-based collaborative filtering is an effective way to recommend useful content to users by exploiting items that similar users may prefer.The algorithm tried to find the user's neighbor based on the user's similarity by analyzing their Pearson Correlation, then employ k-means algorithm to merge the scores of neighboring users in order to get the prediction of data of target user.

In both cases this recommendation engine has two steps:

1. Find out how many users/items in the database are similar to the given user/item.
2. Assess other users/items to predict what grade you would give the user of this product, given the total weight of the users/items that are more similar to this one.

Item-based collaborative filtering use the same schema as User-based collaborative filtering. Instead of nearest neighbors, it considers items. And the methods to find similarity change from the Pearson Correlation to cosine similarity. This algorithm analyze the items target user have already rated, then find the users who share the same interest on same items, find the items these users liked and use them as recommendation

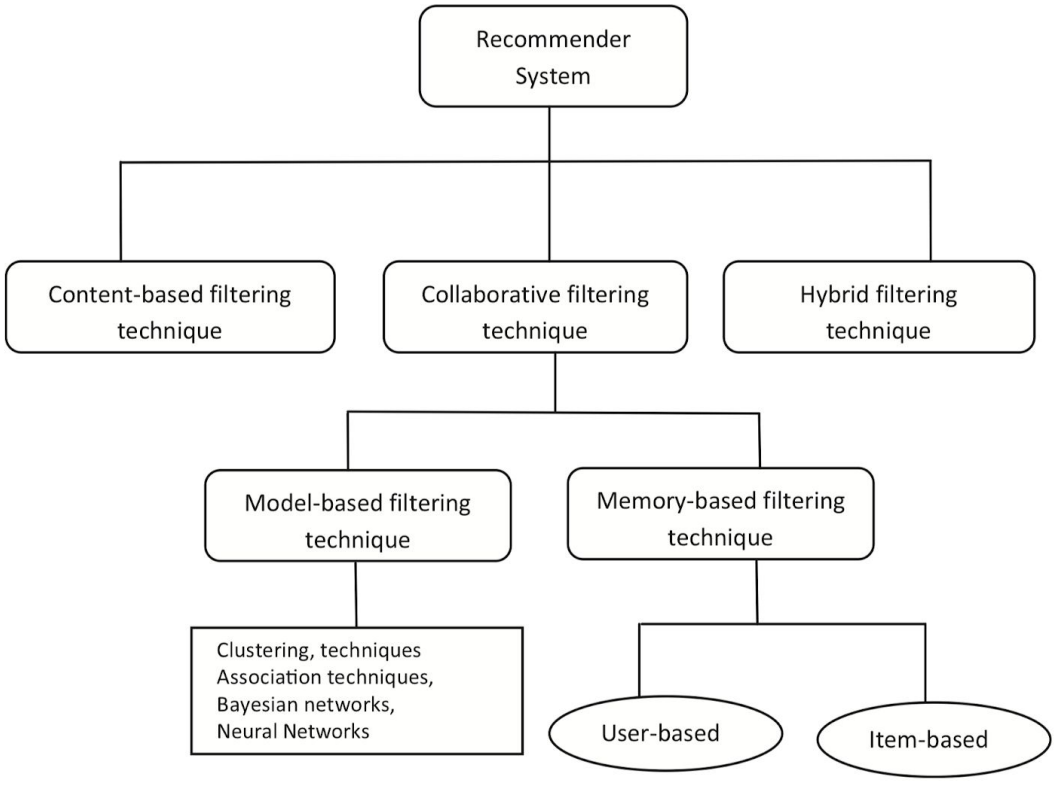

Figure 1 Recommendation techniques.

3. Implementation

3.1 Core Concept of collabrating filtering(CF)

The basic idea of CF is to recommend new items based on the similarity of users. In this section, we will discuss in following part:

1. How to measure similarity between users or objects.
2. Using the cosine similarity to measure the similarity between a pair of vectors.
3. How to use model-based collaborative filtering to identify similar users or items.

3.2 Dataset

The data we employed in this experiment is the GroupLens Research. It is consisted by 100,000 ratings from 1000 users on 1700 movies. Each user has rated at least 20 movies. The structure of the datasets can be seen in the following charts

User ID	Item ID	Rating	Timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
196	51	2	889606923
22	346	1	886397596
186	474	4	884182896

Table 1 Raw dataset of MovieLens

Item User	242	302	377	51	346	…..
196	3	…..	…..	2	…...	….
186	…...	3	4	2	1	…...
22	3	…...	1	2	1	…….

Table 2 User-Item Matrix by raw dataset

3.3 Similarity Method

When compute a similarity, the idea is to calculate the distance between two objects. The higher the distance, the farther apart they are. In the other word, the higher similarity between two objects, the closer they are. Usually similarity metrics return a value between 0 and 1, where 0 signifies no similarity and 1 signifies they are exactly the same.

We could easily tell how far apart these two vectors are. The common approach would be to calculate the Euclidean distance or Manhattan distance. When calculate the similarity between two vectors, we would like to evaluate the angle between them. The smaller the angle is, the more similar the two vectors are. If we restrict the vectors to non-negative values, as in the case of movie ratings, then the angle between two vectors is between 0 degree to 90 degree, corresponding to cosine similarities between 1 and 0. There is one thing important should be noticed: the cosine similarity is a measure of orientation, not the magnitude. Two vectors can have the same direction but different magnitudes.

3.3.1 Adjusted Cosine Similarity

One fundamental difference between the similarity computation in UBCF and IBCF is that in case of UBCF the similarity is computed along the rows of the matrix but in case of the IBCF the similarity is computed along the columns, Computing similarity using basic cosine measure in item-based case has one important drawback-the differences in rating scale between different users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding user average from each co-rated pair. Formally, the similarity between items i and j using this scheme is given by

3.3.2 Prediction Computation

The most important step in a collaborative filtering system is to generate the output interface in terms of prediction. Once we isolate the set of most similar items based on the similarity measures, the next step is to look into the target users ratings and use a technique to obtain predictions. Here we consider two such techniques.

3.3.3 Weighted Sum

This method computes the prediction on an item i for a user u by computing the sum of the ratings given by the user on the items similar to i. Each ratings is weighted by the corresponding similarity $s_{i;}j$ between items i and j

Basically, this approach tries to capture how the active user rates the similar items. The weighted sum is scaled by the sum of the similarity terms to make sure the prediction is within the predefined range.

3.3.3.4 Regression

This approach is similar to the weighted sum method but instead of directly using the ratings of similar items it uses an approximation of the ratings based on regression model.The basic idea is to use the same formula as the weighted sum technique, but instead of using the similar item N's 'raw' ratings values Ru;N 's, this model uses their approximated values R 0u;N based on a linear regression model. If we denote the respective vectors of the target item i and the similar item N by Ri and RN the linear regression model can be expressed as
The regression model parameters and are determined by going over both of the rating vectors. is the error of the regression model.

3.4 Memory-base techiques

The items that were already rated by the user before play a relevant role in searching for a neighbor that shares appreciation with him. Once a neighbor of a user is found, different algorithms can be used to combine the preferences of neighbors to generate recommendations. Due to the effectiveness of these techniques, they have achieved widespread success in real life applications. Memory-based CF can be achieved in two ways through user-based and item-based techniques.

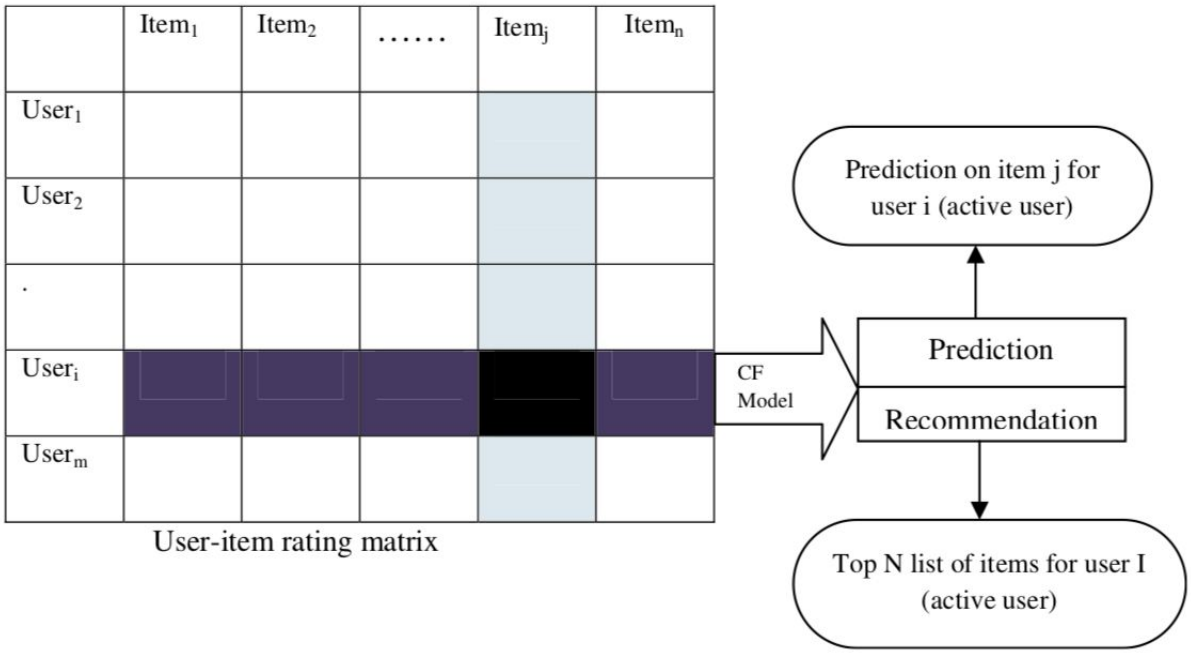

Figure 2 Collaborative filtering process

.

3.3.1 User-based collaborative filtering(UBCF)

User-based collaborative filtering method forecasts items to target users which similar users choose.

User/Movie	Movie1	Movie2	Movie3	Movie4
User1	5	5	4	0
User2	5	4	?	0
User3	3	4	5	?
User4	2	2	5	?

Table 3 User-based matrix by raw dataset

The similarity computation needs both rates to identical movies from one user and the k-nearest neighbors. The rates from different users are considered as vectors, Pearson Correlation Coefficient is implemented in the computation to measure the similarity between these vectors, usually the coefficient is between 0 and 1. The higher coefficient it is, the more similar the two vectors are, which means it is more closer to 1.

The formula of Pearson Correlation Coefficient is

$$sim(i,j) = \frac{\sum_{u \in S_U}(R_{u_i} - \overline{R_i})(R_{u,j} - \overline{R_i})}{\sqrt{\sum_{u \in U}(R_{u,j} - \overline{R_i})^2} \times \sqrt{\sum_{s \in S}(R_{(u,j)} - \overline{R_j})^2}}$$

The system will predict some rates based on the rates which similar users have chosen. For example, select user1 as item1,select user2 as item2, Common movies are [movie1, movie2, movie4],

$$sim(user1,user2) = \frac{(5-3.33)(5-3) + (5-3.33)(4-3) + (0-3.33)(0-3)}{\sqrt{(5-3)^2 + (4-3)^2 + (0-3)^2} \times \sqrt{(5-3)^2 + (4-3)^2 + (0-3)^2}}$$

$$= \mathbf{0.734}$$

select user2 as item1, user3 as item2, Common movies are [movie1, movie2],

$$sim(user2,user3) = \frac{(5-4.5)(3-3.5) + (4-4.5)(4-3.5)}{\sqrt{(5-4.5)^2 + (4-4.5)^2} \times \sqrt{(3-3.5)^2 + (4-3.5)^2}}$$

$$= \mathbf{0.500}$$

When adding a N/S to put a weight consider into this formula,

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})(r_{ys} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \overline{r_x})^2} \times \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \overline{r_y})^2}} \times \frac{N}{S}$$

N/S(Movie1,Movie2) = ¾, N/S(Movie1,Movie3) = ½

Because sim(user1,user2) > sim(user2,user3), we consider User1 and User2 are more similar than User2 and User3.

3.3.2 Item-based collaborative filtering(IBCF)

Item-Based Recommender gets the item, finds the user who likes the item, and finds other items that these users or similar users like. It requires a project and outputs other projects as suggestions.We use collaborative filtering to predict movie ratings.

Users rates movies using 0 to five stars.

Person/Movie	Movie1	Movie2	Movie3	Movie4
Person1	5	5	4	0
Person2	5	4	2	0
Person3	3	4	5	?
Person4	?	?	5	4

Table 4 Item_based matirx by raw data

For each user j learning with parameter θ. Predict user j as rating movie i with $(\theta^{(i)})^T(x^{(i)})$ stars. (? means a missing value)

We use following forumla to measure the the similarity.

$$s_u^{cos}(i_m, i_n) = \frac{i_m * i_n}{||i_m|| * ||i_n||} = \frac{\sum x_{a,m} x_{a,n}}{\sqrt{\sum x_{a,m}^2 \sum x_{a,n}^2}}$$

For instance, select movie1 as item1 and movie2 as item2.

Common users = [Person1, Person2, Person3]

$$cos(movie1, movie2) = \frac{i_m * i_n}{||i_m|| * ||i_n||} = \frac{(5*5 + 5*4 + 3*4)}{\sqrt{(5^2 + 5^2 + 3^2) * (5^2 + 4^2 + 4^2)}} = 0.983$$

select movie1 as item1 and movie3 as item2.

Common users = [Person1,Person2]

$$cos(movie1,movie3) = \frac{i_m * i_n}{||i_m|| * ||i_n||} = \frac{(5*4+5*2)}{\sqrt{(5^2+5^2)*(4^2+2^2)}} = 0.949$$

When adding a N/S to put a weight consider into this formula,

$$s_u^{cos}(i_m, i_n) = \frac{i_m * i_n}{||i_m|| * ||i_n||} = \frac{\sum x_{a,m} x_{a,n}}{\sqrt{\sum x_{a,m}^2 \sum x_{a,n}^2}} * \frac{N}{S}$$

N/S(Movie1,Movie2) = ¾, N/S(Movie1,Movie3) = ½

$$cos(movie1,movie2) * \frac{N}{S}(movie1,movie2) = 0.983 * \frac{3}{4} = 0.737$$

$$cos(movie1,movie3) * \frac{N}{S}(movie1,movie3) = 0.949 * \frac{2}{4} = 0.4745$$

Because $cos(movie1,movie2) > cos(movie1,movie3)$, we consider Movie1 and Movie2 are more similar than Movie1 and Movie3.

4. Evaluation

The method we use to compare the predicted results and the actual output to know the accuracy of our recommendation system is evaluation function. By applying several statistical formulas, we could get the bias when predicting. The following is two main statistical formulas: RMSE and MAE.

4.1 Method

Root Mean Squared Error (**RMSE**) is the most popular metric used in evaluating accuracy of predicted ratings. RMSE represents the sample standard deviation of the differences between predicted values and observed values.

$$RMSE = \sqrt{\frac{1}{|S|} \sum_{(u,i) \in S} (\hat{r}_{ui} - r_{ur})^2}$$

Mean Absolute Error **(MAE)** is another popular alternative, is a measure of difference between two continuous variables, it given by

$$MAE = \sqrt{\frac{1}{|S|} \sum_{(u,i) \in S} |\hat{r}_{ui} - r_{ur}|}$$

The lower the RMSE and MAE are, the more accurately our recommendation system predicts.Because of the square compared to MAE, RMSE amplifies and severely punishes large errors, which means RMSE could detect error more sensitive. As a simple example, suppose we are given a test set with four missing items, and we have two recommender systems.

For example :

System 1: makes an error of 2 on three ratings and 0 on the fourth.

System 2: makes an error of 3 on one rating and 0 on all three others.

RMSE would prefers the first system, while MAE would prefer the second system.

4.2 Result

We started our evaluation by first dividing the data set into training and test portion. Before we make prediction, we need to generate similarity matrix, at this step we should test different similarly function, after compare the prediction accuracy based on different similarly function, we could find the most appropriate similarly function for each collaborative filtering algorithm. Then for each training and test data set, we compare the prediction accuracy of user based recommender and item based recommender algorithm. We also need to experiment the influence of neighbourhood size (the number of similar items or users we choose) in same data set for each recommender algorithm. For conducted a 5-fold cross validation of our experiments by randomly choosing different training and test sets each time and taking the average of the RMSE and MAE values.

In this section, we present our experimental results of item-based collaborative filtering, user-based collaborative filtering, effect of neighbourhood size and cross validation. We generate bar charts and line charts according to the output data of our program.

4.2.1 Experiments with top Match Item size

In order to tests the impact of top Match Item size (the number of similar items or users we choose) we make 10 times prediction for each data set and each collaborative filtering algorithm. As the chart shown below, when we use data set ua to make training and test. We change the number of similarly items and users from 10 to 100 as the step of 10. It is obviously that with the increase of size of top Match Items, the MAE and RMSE is decreasing, and the counting number is increase. Which means the accuracy of our prediction get better. The distribution of User-based collaborative filtering and Item- based collaborative filtering are also quite similar.(User-based distribution graph left, Item-based distribution

graph right). The abscissa indicates: the number of top match items from 10 to 100. The ordinates indicates: MAE or RMSE (represent the accuracy of prediction).

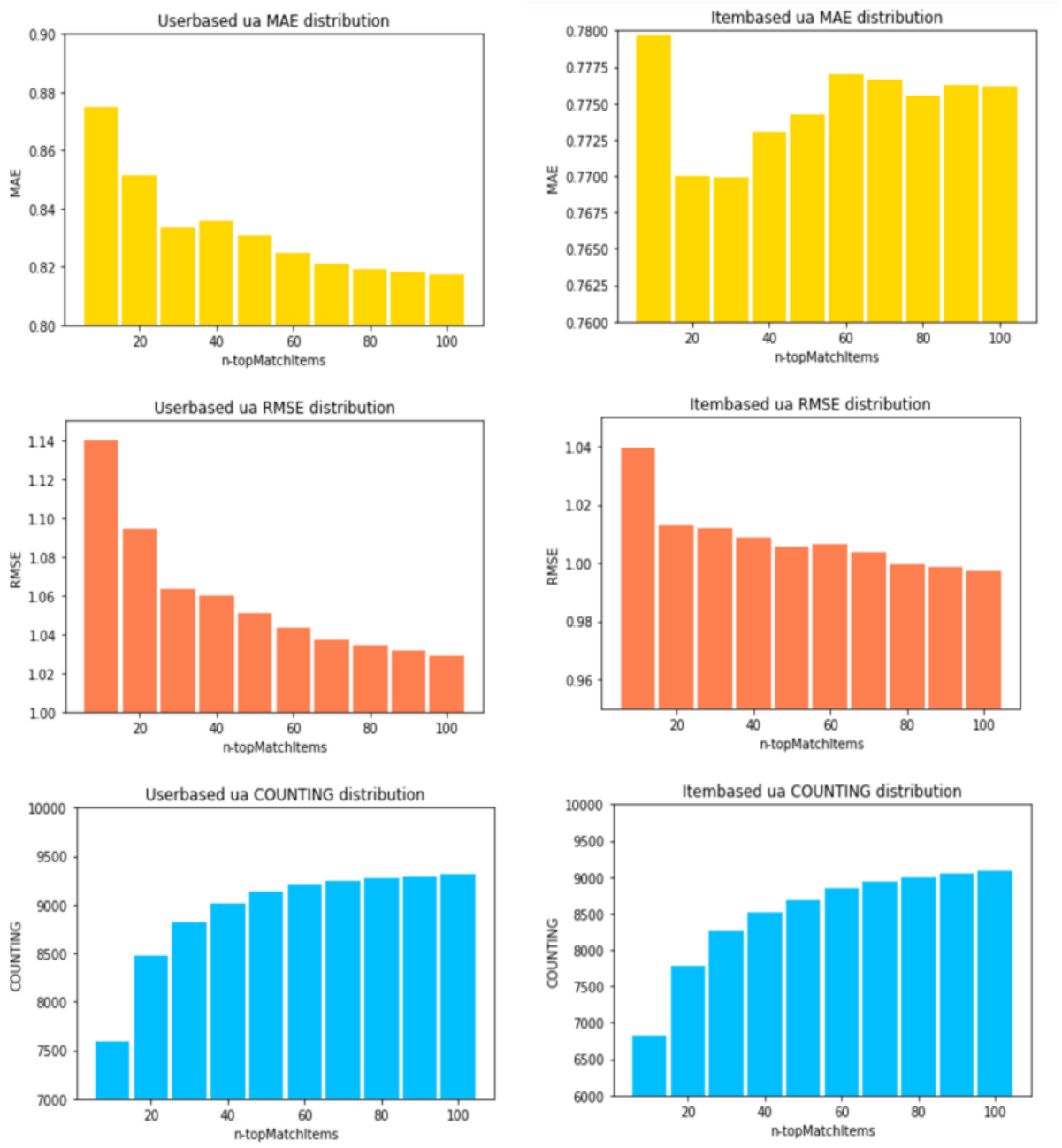

Figure 4 Collaborative filtering process.

4.2.2 Experiments with prediction quality

Compare User-based and Item-based collaborative filtering, we test all the data sets we have and use the output data to generate the graphs below and also perform cross-validation. (User-based distribution graph left, Item-based distribution graph right). The

abscissa indicates: the number of top match items from 10 to 100. The ordinates indicates: MAE or RMSE (represent the accuracy of prediction)

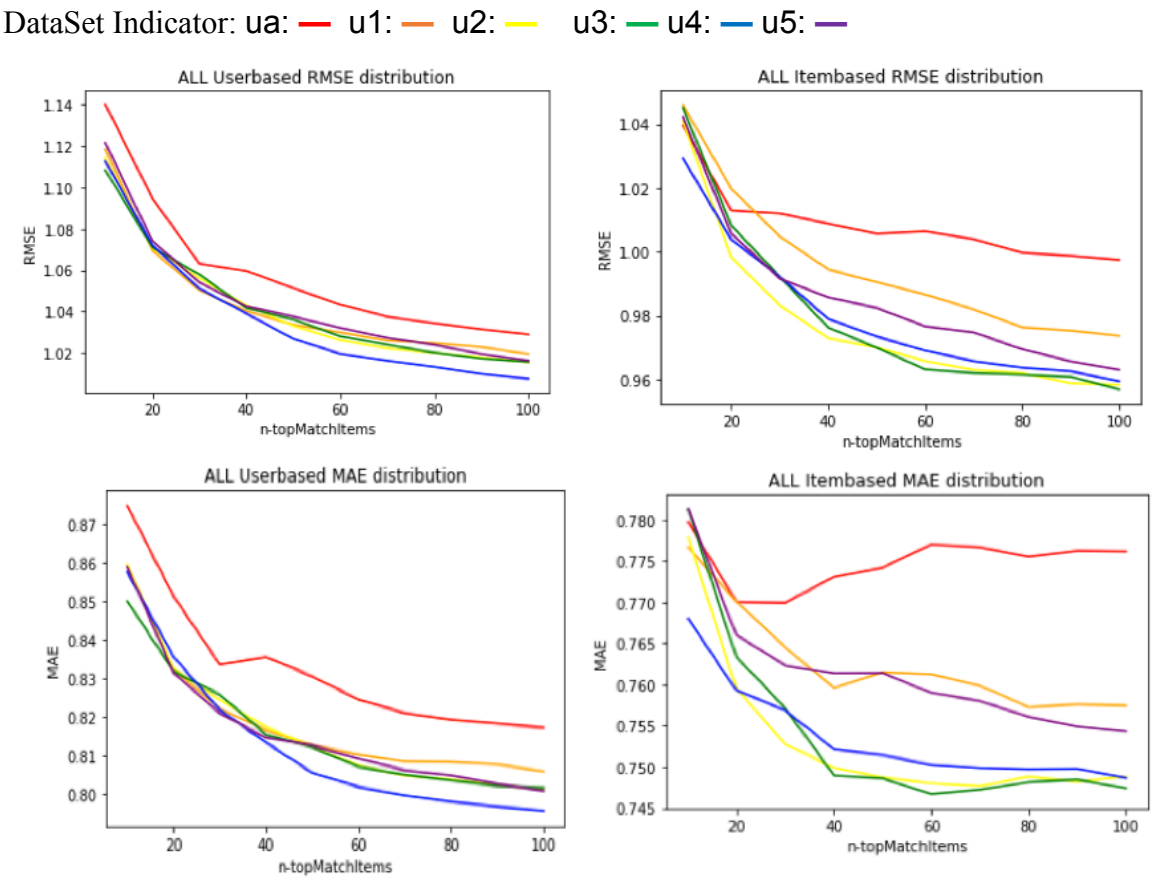

Figure 5 Prediction quality

It is obviously that RMSE of all data set prediction in Item-based are below 1.02 with 100-topMatchItems, which is much better than in User-based distribution. As the MAE distribution, Item-based also performance better than User-based, although in Item-based some data set performance vibrate but the MAE is always below 0.780, which is less than the minimum value (close to 0.80) in User-based MAE.

The different colour line means different data set we test, these charts shown the cross-validation we processed, each time we use one of this data set as the test set and all the rest data set as training set. As the charts shown 5 different data set prediction result performance similar tendency. By making cross-validation our result could reduce the effects of how the data gets divided and also overcome overfitting.

The conclusion of this experiment is that, Item-based algorithms provide better quality than the user-based algorithms at all circumstance, also works good with cross validation.

In our final results, we have accomplished some algorithm to enable the system to recommend certain movies to users. Also, we improved some algorithm after output and compared some results.

5. Reference

[1] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In *The adaptive web* (pp. 291-324). Springer, Berlin, Heidelberg.

[2] Resnick P, Varian H R. Recommender systems[J]. Communications of the ACM, 1997, 40(3): 56-58.

[3] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734-749.

[4] Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). springer US.

[5] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 5-53.

[6] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

[7] Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook* (pp. 73-105). Springer, Boston, MA.