COMP9417 2018S1

Machine Learning and Data Mining

Assignment 2

Topic 3.4

Recommender system using collaborative filtering

Group Member:

Z5121715 Yiming He

Z5089071 Yongjia Wang

Z5047518 Kun Dong

Introduction

A **recommender system** or a **recommendation system** (sometimes replacing "system" with a synonym such as platform or engine) is a subclass of information filtering system that seeks to predict the "rating" or "preference" that a user would give to an item. Recommender systems have become increasingly popular in recent years, and are utilised in a variety of areas including movies, music, news, books, research articles, search queries, social tags, and products in general.

In our project, we developed the movie ratings recommend system by applying collaborative filtering as instructed, and the dataset we used is the MovieLens Data Sets with size of 100k. We implemented both ITEM-ITEM collaborative filtering and USER-USER collaborative filtering aim to gain a more convincing comparison and conclusion. And the result is evaluated by calculating the Root Mean Squared Error and Mean Absolute Error.

Implementation

The *UserBasedRecommender filtering* takes a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked. Alternatively, *ItemBasedRecommender* looks for items that are similar to the movies that user has already rated and recommend most similar movies.

Methodology analysis

Collaborative filtering systems have two forms overall, one is USER-USER filtering:

1. Look for users who share the same rating patterns with the active user.
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.

Another one is ITEM-ITEM filtering:

1. Build an item-item matrix determining relationships between pairs of items.
2. Infer the tastes of the current user by examining the matrix and matching that user's data.

First step is to make a choice, we managed to select one method with better performance, so we analysed the accuracy of recommendation achieved by the ITEM-ITEM filtering and USER-USER filtering. The following part I will analyse every method by calculating the similarity of an particular instance.

USER-USER filtering

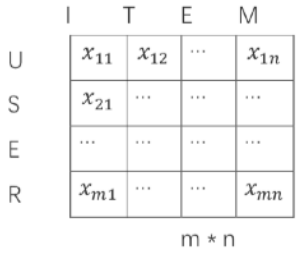
Step 1: Consider user A

Step 2: Find Set B of other users whose ratings are similar with A's ratings

Step 3: Estimate A's ratings based on ratings of users in B

Step 4: Calculate Pearson correlation coefficient

$$\text{Sim}(x,y) = \frac{\sum_{s \in Sxy}(r_{xs}-\bar{r}_x)(r_{ys}-\bar{r}_y)}{\sqrt{\sum_{s \in Sxy}(r_{xs}-\bar{r}_x)^2} \times \sqrt{\sum_{s \in Sxy}(r_{ys}-\bar{r}_y)^2}}$$



For example, we have the following user-rating table:

	HP	Avengers	Thor	Avatar	Saw	IM
A	4	2	5		4	
B		5	3		2	4
C	1		4			3
D	5	4	2			

common movies

We choose user A and B.

Common movies: Avengers, Thor, Saw

Sum of ratings of A: 2+5+4 = 11

Sum of ratings of B: 5+3+2 = 10

Mean of A's ratings = 11/3 = 3.67

Mean of B's ratings = 10/3 = 3.33

Covariance of A and B = [(2-3.67) *(5-3.33) +(5-3.67) (3-3.33) +(4-3.67) (2- 3.33)]/3 = -1.22

Standard deviation of A: square {[(2-3.67)^2+(5-3.67)^2+(4-3.67)^2]/3}

Standard deviation of B: square {[(5-3.33)^2+(3-3.33)^2+(2-3.33)^2]/3}

Then we use the formula above, Sim(A,B) = -0.7857

Then we choose C and A, calculate the similarity is Sim(A,C) = 1.3416

Because of |sim (A, B) |<|sim (A, C) |, thus user A and B are more similar than A and C.

But there is a critical problem in this method, if the users do not have many common rated movies, the result will be not accurate, so let us see if ITEM-ITEM filtering can solve the problem. And in practice, user preferences change quickly and the entire system model had to be recomputed, which is both time and computationally expensive.

ITEM-ITEM filtering

Step 1: Consider movie A

Step 2: Find other similar movies

Step 3: Estimate the rating of A based on other similar movies

Step 4: Implementing cosine similarity metrics and prediction functions as in user-user model

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

	HP	Avengers	Thor	Avatar	Saw	IM
A	4	2	5		4	
B		5	3		2	4
C	1		4			3
D	5	4	2			

Choose movie Avengers and Thor.

Common users: A, B, D

Similarity(Avengers, Thor) = (2*5+5*3+4*2)/ (sqr [(2^2+5^2+4^2) *(5^2+3^2+2^2)]) = 0.799

Choose movie Thor and Saw.

Common users: A, B

Similarity(Thor, Saw) = (5*4+3*2)/(sqr(5^2+3^2)*(4^2+2^2)) = 0.99

Similarity(Avengers, Thor) < Similarity(Thor, Saw), so the movie Thor and Saw are more similar.

Comparing with the USER-USER filtering, in the real life, the average item has a lot more ratings than the average user. So an individual rating doesn't impact as much. So we can conclude this method is quite stable in itself as compared to User based collaborative filtering.

Prediction

Since we have the similarity of the movies or users, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items.

USER-USER collaborative filtering prediction

1. Let r_x be the vector of user x's ratings

2. Let N be the set of k users most similar to x who have rated item i

3. Prediction for item s of user x

$$r_{xi} = 1/k \sum_{y \in N} r_{yi}$$

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$$

	HP	Avengers	Thor	Avatar	Saw	IM	Similarity
A	4	2	5		4	?	1
B		5	3		2	4	0.7
C	1		4			3	0.2
D	5	4	2			3	-0.6

If we want to predict the rating of IM by user A, and we choose top 2 nearest neighbourhood. We just select top 2 similar users with A, user B and D.

Prediction = (4*0.7 + 3*|-0.6|) / (0.7+|-0.6|) = 3.54 **(4*0.7 + 3*0.2) / (0.7+0.2) = 3.78**

ITEM-ITEM collaborative filtering prediction

The formula to calculate rating is very similar to the user based collaborative filtering except the weights are between items instead of between users. And we use the current users rating for the item or for other items, instead of other users rating for the current items.

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

s_{ij} similarity of movie i and j
r_{xj} rating of user x on movie j
$N(i;x)$ set of movies rated by user x similar to j

Common Practice

1. Define similarity s_{ij} of items i and j
2. Select k nearest neighbours $N(i;x)$, Items most similar to i, that were rated by x
3. Estimate rating r_{xi} as the weighted average:

$$r_{xi} = b_{xi} + \frac{\sum_{j \in N(i;x)} s_{ij} \cdot (r_{xj} - b_{xj})}{\sum_{j \in N(i;x)} s_{ij}}$$

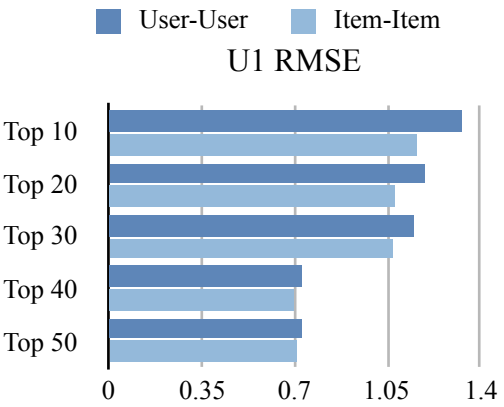
Evaluation

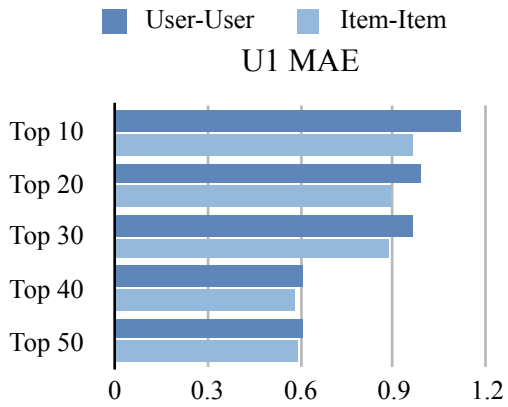
with dataset ml-100k/ u*.base for training and u*.test for testing

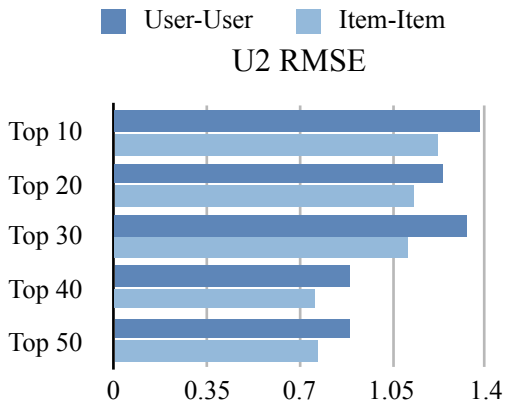
After make the prediction, we evaluated the prediction result by comparing the actual value and prediction to get the accuracy of our recommender system.

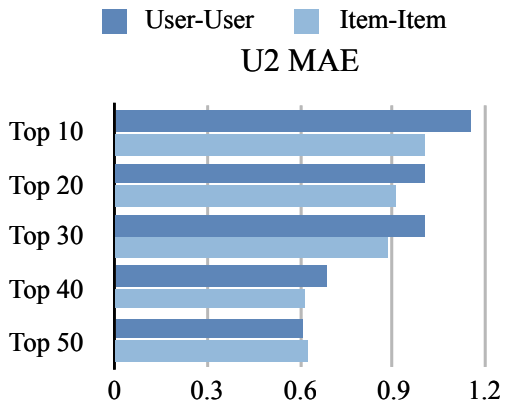
We chose two types of evaluation criterion, one is Root Mean Squared Error, which is the most common and popular metric used in evaluating accuracy of predicted ratings. The other one is Mean Absolute Error, is a measure of difference between two continuous variables. The lower RMSE and MAE indicate that our recommendation system is better.

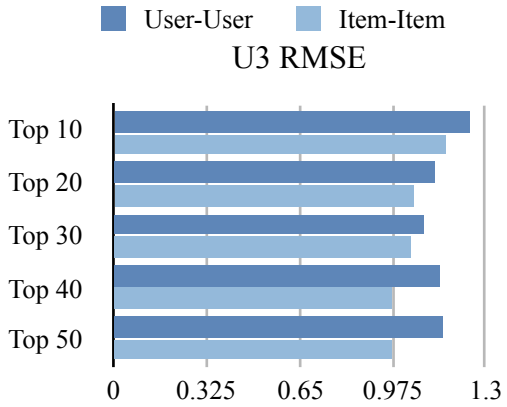
We use the training dataset and test dataset provided by the GroupLens, and for every dataset we calculate an RMSE and MAE with 5 different top k nearest neighbourhoods conditions.

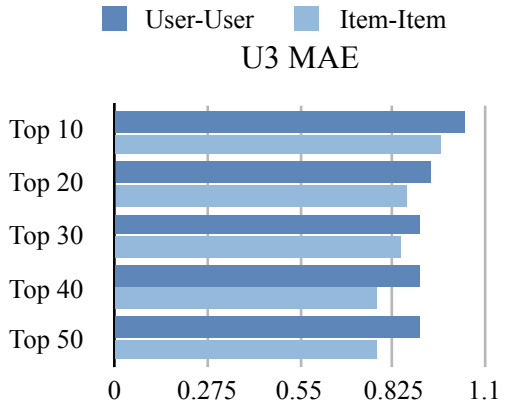


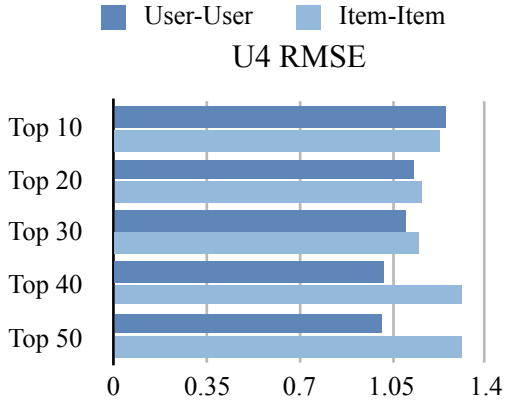


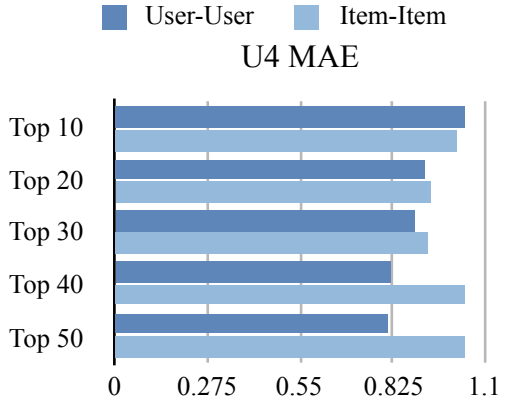

User-User
Item-Item
U2 RMSE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.35
0.7
1.05
1.4

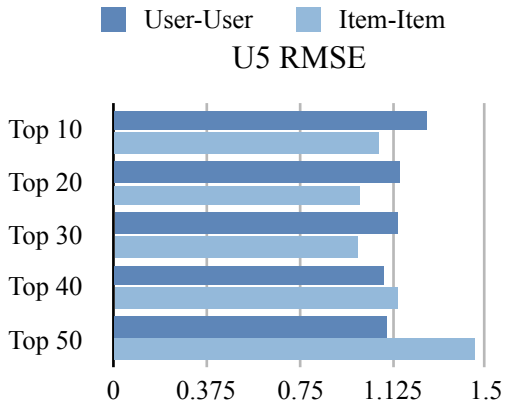

User-User
Item-Item
U2 MAE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.3
0.6
0.9
1.2

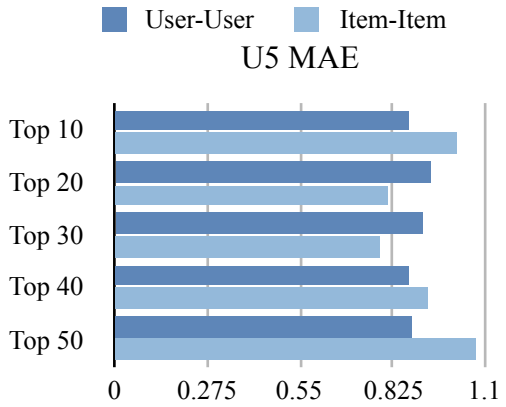

User-User
Item-Item
U3 RMSE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.325
0.65
0.975
1.3


User-User
Item-Item
U3 MAE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.275
0.55
0.825
1.1


User-User
Item-Item
U4 RMSE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.35
0.7
1.05
1.4


User-User
Item-Item
U4 MAE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.275
0.55
0.825
1.1


User-User
Item-Item
U5 RMSE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.375
0.75
1.125
1.5


User-User
Item-Item
U5 MAE
Top 10
Top 20
Top 30
Top 40
Top 50
0
0.275
0.55
0.825
1.1

Result Analysis

Overall, in the most datasets the Item based collaborative filtering's RMSE and MAE are both small than the User based collaborative filtering, which means the Item based collaborative filtering is the better recommendation system. Furthermore, with the increasing of the K nearest neighbourhood, the RMSE and MAE keep decreasing, which means the recommendation system result is more accurate.

But there are some special cases. In the U4 and U5 dataset, the RMSE dramatically increased with the top 50 nearest neighbourhood, which indicates that the movies in the top 50 probably is not that uniform, the ratings of them are quite different, so the result is relatively less accurate.

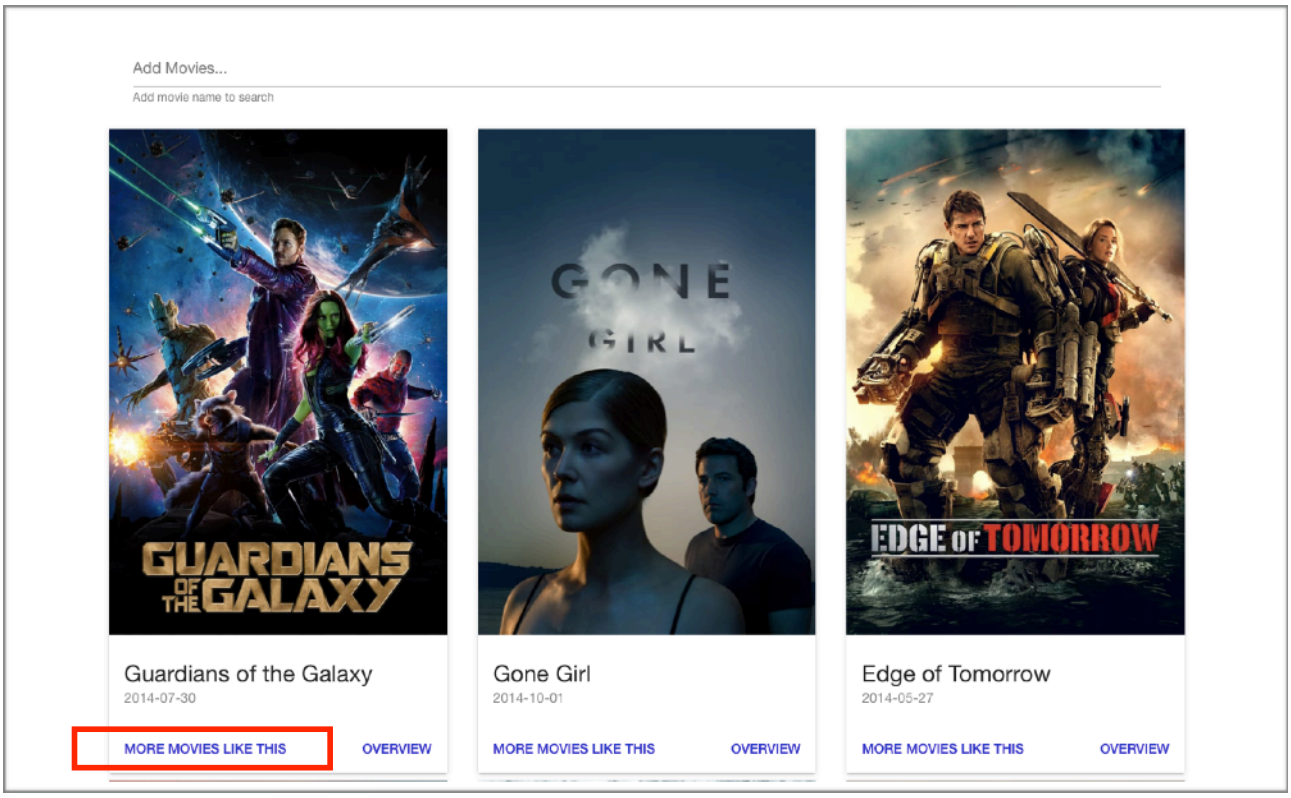
Considering running time, we only test 10% test data(2000 records) of each data set, which may not cover every user . So in U4 and U5, the randomness of picking test case could also cause this result

Extra Feature & Presentation

In order to present the recommend system in a clear and interactive way, we build a presentation platform on GitHub by Vue.js. The platform presents all the movies in the dataset, and user can easily click on the recommend button and top 10 similar movies will automatically pop up in the page. The platform is not involved with the users' previous preferences because we do not have a user login function, we just calculate the similarity of the movies by ITEM-ITEM collaborative filtering, but we will improve the platform in the future.

Please visit: *https://unswddk.github.io/comp9417/#/*

Use a larger and latest dataset

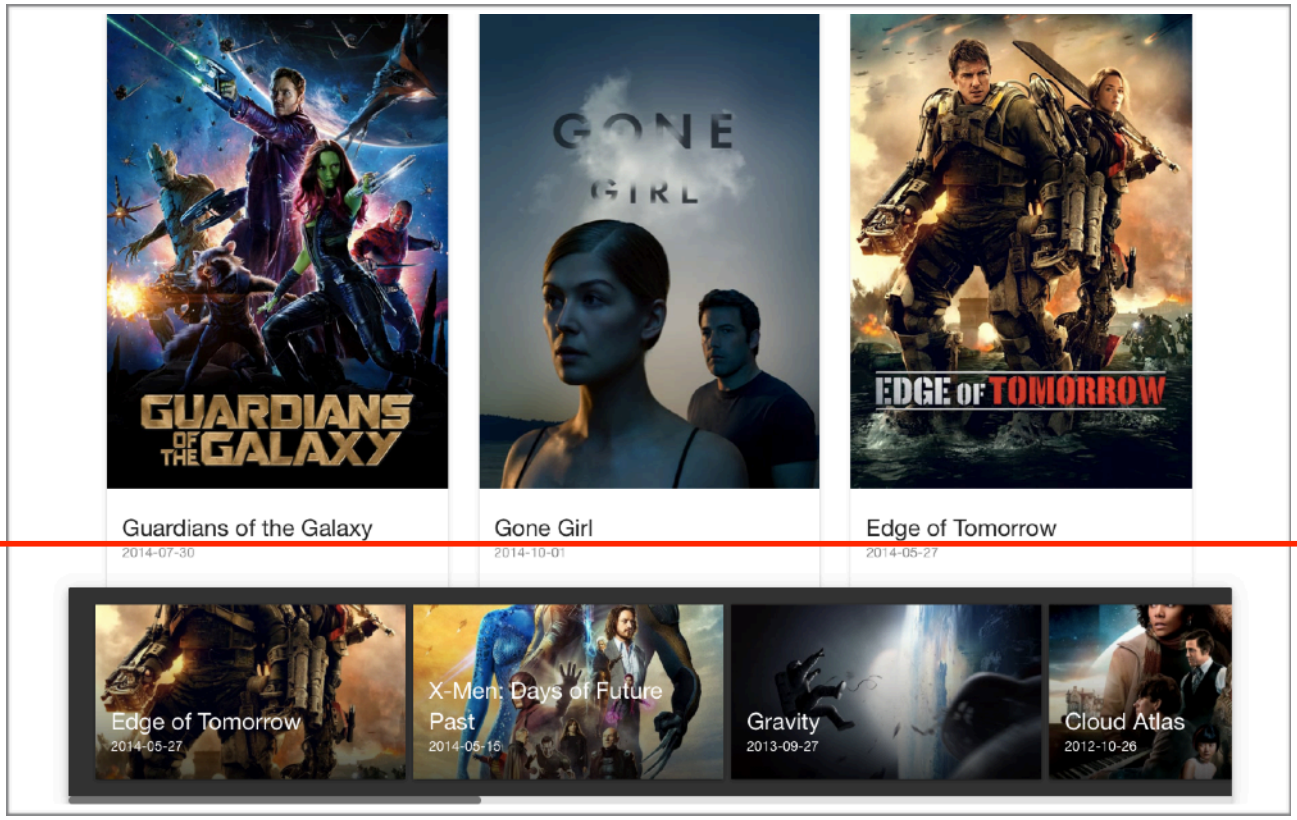


Platform Development:

Step 1: We store the movies.csv in the server, then we iterate the csv to retrieve all the movie information, splitting the information into movie year and movie name. Since we already have the movie's information, we can use api from IMDB to collect movie poster to present it in our page.

Step 2: In the server side, we already pre-process all the movies to calculate the similarity by our ITEM-ITEM collaborative filtering code, which took 20 hours to process the whole dataset. And then we store the top 10 recommended movies of every single movie in our server in JSON.

Step 3: When a user click on the 'more movie like this' button, the top 10 recommended movies will pop up from the bottom of the page.



The recommended movies of Guardians of the Galaxy

What's more, we also provide the search function, if the movie you like is not in the first page, you can find it by searching the movie name and it will present on the very first position. We hope this platform will give our users a better interaction with our system, and all the users' preferences from the website will be stored in our database and we can use them to improve our system in the future.

Conclusion

Theoretically, ITEM-ITEM collaborative filtering is quite stable in itself as compared to User based collaborative filtering, it can be prevented from an individual rating's negative effect. And in our experiment, ITEM-ITEM collaborative filtering provides better performance than the USER-USER collaborative filtering in most circumstances, also works well with cross validation.

References

J Schafer, D Frankowski, J Herlocker, S Sen "Introduction to recommender systems: Algorithms and evaluation", ACM Transactions on Information Systems (TOIS), 2007

JA Konstan "ACM Transactions on Information Systems (TOIS)", The adaptive web, 2004