COMP9417 ASSIGNMENT

Recommender system using collaborative filtering

Group Name : EatChicken

Group Member :

Li Xu	z5136365
Ran Bai	z5187292
Wenxun Peng	z5195349

1 Abstract

The collaborative filtering recommendation algorithm is the earliest and well-known recommendation algorithm. The main function is prediction and recommendation. The algorithm discovers the user's preferences by mining the user's historical behavior data, and classifies the users based on different preferences and recommends similar items. The basic steps of the memory-based collaborative filtering algorithm are to first collect and organize user preferences, then find similar users or items, and finally give recommendations by comparing similarities.

General Terms: Measurement, Performance

Key words: Collaborative Filtering, Recommendation, Pearson correlation, Cosine similarity

2 Introduction

The research of the recommender system can be roughly divided into three phases. The first phase is based on traditional services, the second phase is based on current social network services, and the third phase is the upcoming Internet of Things. This has produced many basic and important algorithms, such as collaborative filtering (including user-based and item-based), content-based recommender system algorithms, hybrid recommender system algorithms, statistical theory-based recommender system algorithms, and social network-based information (following Filter recommender system algorithm, group recommender system algorithm, location-based recommender system algorithm, attention, trust, popularity, credibility, etc. The neighborhood-based collaborative filtering recommender system algorithm is the most basic, core, and most important algorithm in the recommender system. The algorithm is not only deeply researched in academia, but also widely used in the industry, based on neighborhood. The algorithm is mainly divided into two categories, one is user-based collaborative filtering algorithm, and the other is the item-based collaborative filtering algorithm. In addition, the item-based recommendation algorithm is also widely used, two basic algorithms will be described in detail as below.

3 Collaborative Filtering

It is recommended that the system apply data analysis technology to find out what users are most likely to like and recommend it to users. Many e-commerce websites now use this algorithm. At present, the more mature recommendation algorithm is Collaborative Filtering (CF) recommender algorithm. The basic idea of CF is to recommend items to users according to their previous preferences and other users with similar interests.

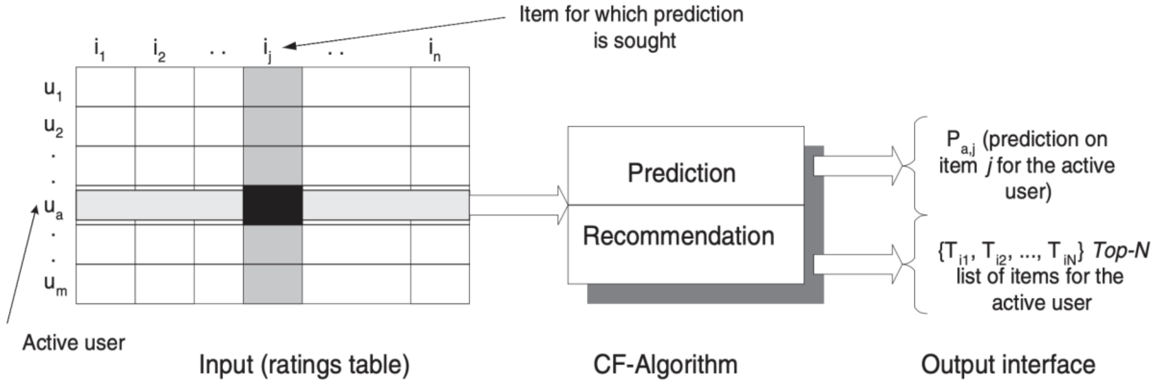

Item for which prediction is sought
i_1 i_2 . . i_j . . i_n
u_1
u_2
u_a
u_m
Active user
Prediction
Recommendation
$P_{a,j}$ (prediction on item *j* for the active user)
$\{T_{i1}, T_{i2}, ..., T_{iN}\}$ *Top-N* list of items for the active user
Input (ratings table)
CF-Algorithm
Output interface

Figure1 The Collaborative Filtering Process[i]

As shown in Figure 1, in the CF, the $m \times n$ matrix is used to indicate the user's preference for the item, and the score is generally used to indicate the user's preference for the item. The higher the score, the more like the item, and the 0 indicates that the item has not been bought. The row in the figure represents a user, the column represents an item, and the U_{ij} represents the scoring of the item j by the user i. CF is divided into two processes, one for the prediction process and the other for the recommendation process. The prediction process predicts the user's possible score for items that have not been purchased. The recommendation is to recommend one or Top-N items that the user would most like based on the results of the prediction phase.

The CF algorithm is divided into two categories, one is memory-based, the other is Model-based. The User-based and Item-based algorithms are Memory-based.

The basic idea of User-based is that if user A likes item a, user B likes items a, b, c, and user C likes a and c, then user A is considered similar to users B and C because they all like a, but like The user of a also likes c, so recommend c to user A. The algorithm uses the nearest neighbor (nearest-neighbor) algorithm to find a set of neighbors of a user. The users of the set have similar preferences to the user, and the algorithm predicts the user according to the preference of the neighbors.

The basic idea of Item-based is to calculate the similarity between items based on historical preference data of all users in advance, and then recommend the items similar to those that the user likes to the user. Taking the previous example as an example, you can know that items a and c are very similar, because users who like a also like c, and user A likes a, so recommend c to user A.

4 Implementation

4.1 General Process

General process of User-based and Item-based CF algorithms:

1. Calculate similarity (inter-user or inter-item similarity): usually adopt the pearson correlation or cosine similarity.
2. Prediction (the attraction of the target user's unpurchased item to the target user): the target user's predicted rating for the unpurchased item.
3. Recommendation: recommend the top K nearest items to the target user.

4.2 Dataset

The dataset we chose is MovieLens 100K Dataset. It contains 100,000 ratings (1-5) from 943 users on 1682 movies, and each user has rated at least 20 movies. The fields of each table have been shown as below:

UserID	ItemID	Rating	Timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

Figure 2: Original data (u.data) from dataset

The raw data(u.data) is first converted into the user-item matrix before starting the recommendation system algorithm.

Item	242	302	377	51	346

User					
196	3	NULL	NULL	NULL	NULL
186	NULL	3	NULL	NULL	NULL
22	NULL	NULL	1	NULL	NULL
244	NULL	NULL	NULL	2	NULL
166	NULL	NULL	NULL	NULL	1

Figure 3: User-Item matrix

4.3 Calculate Similarity

In the recommendation system, it is usually necessary to calculate the distance between two feature vectors, that is, the similarity between the two. Different similarity measures have great differences for the results of the algorithm. Therefore, it is necessary to select a suitable similarity calculating method according to the characteristics of the data.

The three most commonly used calculation methods are: Euclidean distance, Pearson similarity coefficient, cosine similarity, and adjusted cosine similarity. The following will analyse the characteristics of the three methods.

4.3.1 Euclidean Distance

$$\begin{aligned} &euclidean\_simlarity \\ &= \sqrt{(m_1^1 - m_1^2)^2 + (m_2^1 - m_2^2)^2 + (m_3^1 - m_3^2)^2 + (m_4^1 - m_4^2)^2 + (m_5^1 - m_5^2)^2} \end{aligned}$$

The subscript indicates the serial number of the movie, and the superscript indicates the user serial number. Since the Euclidean distance similarity is in the range of 0 to 1, the similarity of the Euclidean distance is between 0 and 1 by the following formula.

$$euclidean\_simlarity = \frac{1}{1 + euclidean\_simlarity}$$

When the Euclidean distance is 0, the similarity is 1, and when the Euclidean distance tends to infinity, the similarity tends to zero.

Calculating the similarity using the Euclidean distance, when the user rates 0, the similarity will be 1. To avoid this problem, the similarity can be calculated only when the users have scored the same product.

4.3.2 Pearson Similarity Coefficient

Pearson's similarity is insensitive to the magnitude of the user relative to the Euclidean distance. For example, two users, one user usually likes to evaluate 5 stars, while another user likes to evaluate 1 star. The Pearson correlation coefficient considers the two vectors to be equal. The Pearson correlation coefficient ranges from -1 to 1. The Pearson correlation coefficient can be obtained by the numpy corrcoef function, and the range of similarity is transformed between 0 and 1 by 0.5+0.5*corrcoef().

$$r = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}$$

4.3.3 Cosine Similarity

$$cos\theta = \frac{x_1x_2 + y_1y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

The cosine similarity is to calculate the angle between two vectors. If the angle is 90 degrees, the similarity is 0. If the angle is 0 degrees, that is, the directions of the two vectors are the same, the similarity is 1. The range of cosine similarity is also between -1 and 1, and the range can be changed to between 0 and 1 by the above method.

The cosine similarity uses the cosine of the angles of the two vectors in the vector space as the measure of the difference between the two individuals. Compared to distance metrics, cosine similarity pays more attention to the difference in direction between two vectors, rather than distance or length.

4.3.4 Adjusted Cosine Similarity

The cosine similarity is more to distinguish the difference from the direction, but not the absolute value. Therefore, it is impossible to measure the difference of each dimension value, and the insensitivity of the cosine similarity to the value results in the error of the result.

$$sim(i,j) = \frac{\sum_{u \in U}\left(R_{u,i} - \bar{R}_u\right)\left(R_{u,j} - \bar{R}_u\right)}{\sqrt{\sum_{u \in U}\left(R_{u,i} - \bar{R}_u\right)^2}\sqrt{\sum_{u \in U}\left(R_{u,j} - \bar{R}_u\right)^2}}$$

The mean of each dimension of the vector is calculated first, then each vector is subtracted from the mean in each dimension, and then the cosine similarity is calculated.

4.4 Calculate Prediction Set

Model-based collaborative filtering is currently the most popular type of collaborative filtering. In the data set we use, only some users and some data have scoring data, and other parts are blank. In this case, we need to use existing ones. Partially sparse data to predict the score relationship between those blank items and the data, and find the highest rated item recommended to the user.

4.4.1 Weighted Summation

By weighting the scores of the items that the user *u* has scored, the weight is the similarity between each item and the item *i*, and then averaging the sum of the similarities of all the items, and calculating the user *u* to score the item *i*, the formula is as follows:

$$P_{u,i} = \frac{\sum(all - similar - items, N)(S_i, N^* R_{u,N})}{\sum_{(all-similar-items,N)}(\,|s_i, N)}$$

Among them, $S_{i,\,N}$ is the similarity between the item *i* and the item *N*, and $R_{u,\,N}$ is the score of the item *N* by the user *u*. The meaning of this formula is equivalent to taking a weighted average of the possible scores of the user u for the item *i*.

4.4.2 Regression Algorithm

Similar to the method of weighted summation above, but the regression method does not directly use the score of the similar item *N*, because there is a misunderstanding when calculating the similarity by cosine method or Pearson correlation method, that is, the two scoring vectors may be far apart (Euclidean Distance), but there may be a high degree of similarity. Because different users have different habits, some prefer to score high, and some prefer to score low. If both users like the same item, their Euclidean Distance may be farther because of different scoring habits, but they should have a higher similarity. In this case, the calculation of the score value of the user's original similar item may result in a poor prediction result. By re-estimating a new value using linear regression, the same method as above is used for prediction. The calculation method is as follows:

$$\bar{R}'_N = \alpha R_i + \beta + \epsilon$$

The item N is a similar item of the item i, α and β are obtained by linear regression calculation of the scoring vectors of the item N and i, ϵ is the error of the regression model. Collaborative filtering with regression algorithms looks more natural than classification algorithms. Our score can be a continuous value rather than a discrete value. Through the regression model, we can get the target user's prediction score for a certain product.

4.5 Memory-based Filtering Technique

Item User	243	125	198	22
233	5	5	4	0
256	5	4	?	0
78	3	4	5	?
139	2	2	5	?

Figure 4: Example of User-Item matrix

4.5.1 User-based Collaborative Filtering

When a user A needs a personalized recommendation, he can first find other users who have similar interests, and then recommend those movies that the users like and the user A has seen to A.

a) Find a collection of users with similar interests to the target user. To calculate the similarity of interest between the two users. Here, the collaborative filtering algorithm mainly uses the similarity of behaviors to calculate the similarity of interest.

According to Pearson Similarity Coefficient,

$$sim(i,j) = \frac{\sum_{u \in S_U} \left(R_{u,i} - \bar{R}_i\right)\left(R_{u,j} - \bar{R}_i\right)}{\sqrt{\sum_{u \in U} \left(R_{u,i} - \bar{R}_i\right)^2} \sqrt{\sum_{u \in U} \left(R_{u,j} - \bar{R}_i\right)^2}}$$

$$sim(U_{233}, U_{256}) = \frac{(5-3.33)(5-3)+(5-3.33)(4-3)+(0-3.33)(0-3)}{\sqrt{(5-3)^2+(4-3)^2+(0-3)^2} \times \sqrt{(5-3)^2+(4-3)^2+(0-3)^2}} = 0.734$$

$$sim(U_{256}, U_{78}) = \frac{(5-4.5)(3-3.5)+(4-4.5)(4-3.5)}{\sqrt{(5-4.5)^2+(4-4.5)^2} \times \sqrt{(3-3.5)^2+(4-3.5)^2}}$$

b) Find items in this collection that the user likes and that the target user has not heard of are recommended to the target user.

Weight(m243, m125)=3/4, weight(m243, m198)=1/2.

As above, similarity of (User$_{233}$, User$_{256}$) is greater than (User$_{256}$, User$_{78}$), and (User$_{233}$, User$_{256}$) is more similar than (User$_{256}$, User$_{78}$).

The key to step (a) is Given User$_{233}$ and User$_{256}$, there are User$_{233}$ that have had a higher scored set, and User$_{256}$ has ever had a higher scored movie collection.

4.5.2 Item-based Collaborative Filtering

Recommend items to users that are similar to the items they liked before. Firstly need to calculate Cosine Similarity,

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

$$\cos(\mathrm{m243}, \mathrm{m125}) = \frac{i_m * i_n}{||i_m|| * ||i_n||} = \frac{(5 \times 5 + 5 \times 4 + 3 \times 4)}{\sqrt{(5^2+5^2+3^2) \times (5^2+4^2+4^2)}} = 0.983$$

$$\cos(\mathrm{m243}, \mathrm{m198}) = \frac{i_m * i_n}{||i_m|| * ||i_n||} = \frac{(5 \times 5 + 5 \times 2)}{\sqrt{(5^2+5^2) \times (4^2+2^2)}} = 0.949$$

Adjusted Cosine Similarity with their weight,

Weight(m243, m125)=3/4, weight(m243, m198)=1/2.

$$\cos(\mathrm{m243}, \mathrm{m125}) \times \frac{N}{S}(\mathrm{m243}, \mathrm{m125}) = 0.983 \times \frac{3}{4} = 0.737$$

$$\cos(\mathrm{m243}, \mathrm{m198}) \times \frac{N}{S}(\mathrm{m243}, \mathrm{m198}) = 0.949 \times \frac{2}{4} = 0.4745$$

Since cos(m243, m125) > cos(m243, m198), the similarity of movie$_{243}$ and movie$_{125}$ is greater than movie$_{243}$ and movie$_{198}$.

5 Evaluation

5.1 Method Intro

5.1.1 MAE

The mean absolute error (MAE) in statistical accuracy metrics is widely used to evaluate the recommended quality of collaborative filtering recommendation systems. Therefore, the recommended quality assessment uses the common average absolute error. MAE firstly uses the recommendation system to predict the user's score on the test set, and then calculates the deviation of the two based on the actual score of the user in the test set, which is the value of MAE.

Assuming that the predicted user scores {p_1, p_2..., p_n} correspond to the actual scores of { q_1, q_2, ..., q_n }, the calculation formula for MAE is[ii]

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|p_i - q_i|$$

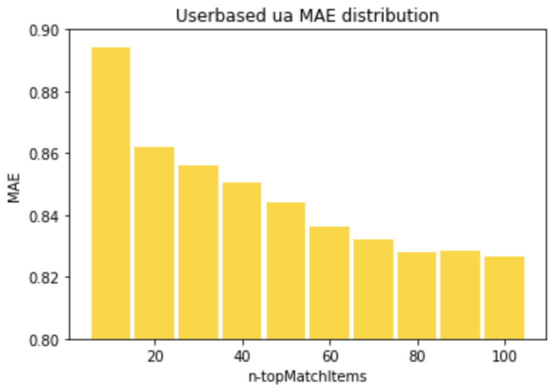
5.1.2 RMSE

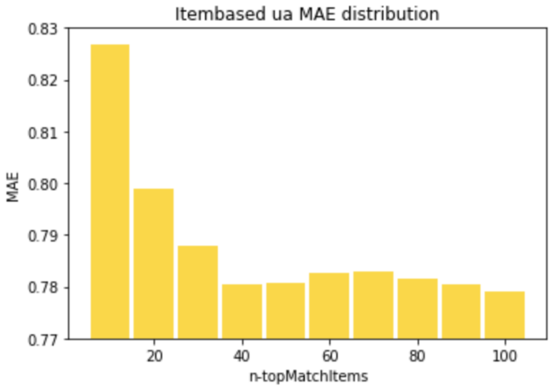
The root mean square error is the arithmetic square root of the mean square error. In other words, the square root of the ratio of the square of the observation to the true (or analog) deviation (rather than the deviation between the observation and its mean) and the number of observations n. In actual measurements, the number n of observations is always limited. The true value can only be replaced by the most reliable (best) value. The RMSE is very sensitive to very large or very small errors in a set of measurements, so the RMSE is a good reflection of the precision of the measurement. This is why RMSE is widely used in engineering measurements. Therefore, RMSE is used to measure the deviation between the observed value and the true value[iii].

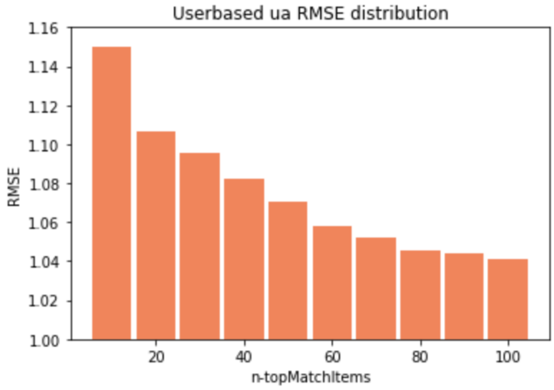
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - q_i)^2}$$

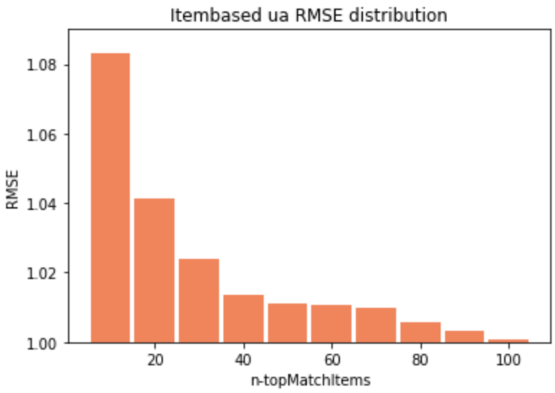
5.2 Evaluation

For more accurate testing and computational accuracy, we divided the data set into two, the training set increased from 10 to 100, and the test set decreased. As the test set and the training set change, the MAE and RMSE also changed.

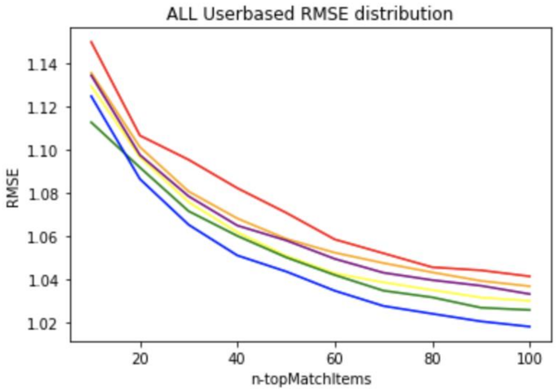


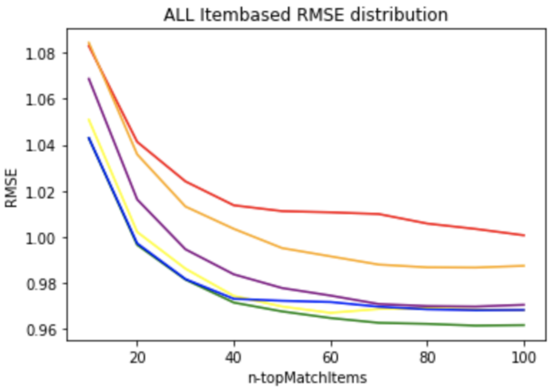


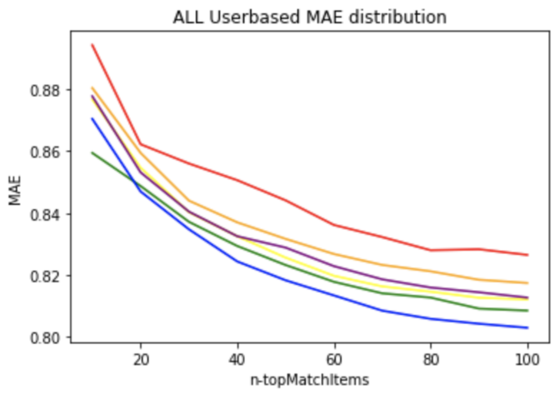


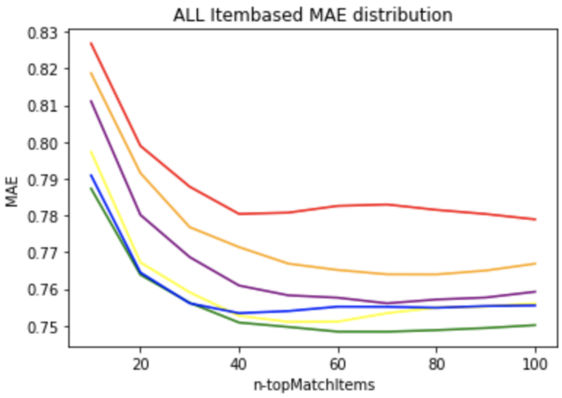


The conclusion charts of the dataset will be shown as below:









5.2.1 User-based Collaborative Filtering

For occasions where there are few users, if there are many users, it takes a lot of time to calculate user similarity. The timeliness is strong, and the user's personalized interest is not obvious. When a user has a new evaluation, it does not necessarily result in an immediate change in the recommendation result. The accuracy recommended for users is not very high.

5.2.2 Item-based Collaborative Filtering

Applicable to the case where the number of items is significantly smaller than the number of users. If there are many items, it takes too much time to calculate the similarity of the items. It is suitable for users to personalize recommendations, and can display changes in recommendation results in real time based on new data. As long as the user is interested in an item, he can be recommended to other items similar to the item. Users are more likely to accept the recommended results.

6 Reference

[i] Sarwar B M, Karypis G, Konstan J A, et al. Item-based collaborative filtering recommendation algorithms[J]. Www, 2001, 1: 285-295.

[ii] Willmott C J, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance[J]. Climate research, 2005, 30(1): 79-82.

[iii] Chai T, Draxler R R. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature[J]. Geoscientific model development, 2014, 7(3): 1247-1250.