COMP9417 Machine Learning Project
Recommender system using collaborative filtering and Optimization

Zhihao Li z5144901
Peng Liu z5178833
Jingjing Wang z5188107
Kaiwen Luo z5100899

Abstract

Collaborative Filtering(CF) algorithm is widely applied in personalized recommender systems, which can help user make choices based on the opinions of other users. Our project is to implement user-based and item-based algorithm to test a reliable and highly accurate method for movie recommendation. In this project, we use python language to implement a movie recommendation sytem with different method，and we come up with a method to optimize original algorithm according to the weakness we found in the result.

Introduction

With the rapid development of internet and technology, people are facing information explosion, which means the increase of information is far more quicker than people process it, especially in e-commerce, user may get trouble in finding out useful information from a large amount of product information.

In this case, the way to deal with massive data and information has become a important research topic. In order to solve this predicament, recommendation system is proposed to adjust to the new internet scenario, which is a subset of information filtering system that seek to predict the ‘rating’ or ‘preference’ that customer may evaluate an item [3]. The recommender system aims to help choose the most similar and interesting items from the clustering. Predication and inference of most related products are based on existing user groups past behavior and opinions.

Related Work

In this assignment, collaborative filtering is used to achieve the recommendation system. There are three types of collaborative filtering, first one is user-based (User-based CF) [4], second is item-based (Item-based CF) [5]and last one is model-based .The movie score is used to present different users’ preferences on movies and we calculate the similarity between different users‘ movie scores to give user related movie recommendations.

1. Similarity measurement method

In order to find the nearest neighbor of the target user, the similarity between the users must be measured, and then several users with the highest similarity are selected as the nearest neighbor of the target user. The nearest neighbor query of the target user is accurate or not, which is directly related to the recommendation quality of the entire recommendation system. Accurately querying the nearest neighbor of the target user is the key to the success of the entire collaborative filtering recommendation.

The user rating data can be represented by a m × n matrix A (m, n), m rows represent m users, n columns represent n items, the cell $R_{i,j}$ represent user i' s rating score to item j. The score of the user rating data is shown in Figure 1.

	$Item_1$	...	$Item_k$	...	$Item_n$
$User_1$	$R_{1,1}$	...	$R_{1,k}$	...	/
...	...	...	...	...	...
$User_i$	$R_{i,1}$	...	$R_{i,k}$	...	/
$User_j$	$R_{j,1}$	...	$R_{j,k}$	...	$R_{j,n}$
...	...	...	...	...	...
$User_m$	$R_{m,1}$	...	$R_{m,k}$	...	$R_{m,n}$

Fig.1 User rating data matrix

The method of measuring the similarity between user i and user j is firstly to obtain all the items scored by user i and user j, and then calculate the similarity between user i and user j through different similarity measure methods and record it as:

$$sim(i,j)$$

In terms of methods, we mainly use Euclidean Distance similarity, Pearson Correlation similarity and cosine similarity to implement basic movie recommendation system.

1) Euclidean distance similarity

The Euclidean distance is the "ordinary" straight-line distance between two points in space.

$$distance = \sqrt{\sum_{i=j=0}^{n}(x_i - x_j)^2} \qquad (1)$$

Formula (1) can figure out the distance between the sores of clusters, which means the more similar of users' preferences, the less distance are. It can be presented as

a function (2) to show the more similar the user's preference, the larger the function value:

$$similar\_distance = \frac{1}{distance} \qquad (2)$$

However, considering the situation when distance equals to zero, the function has been improved as $similar\_distance = \frac{1}{distance+1}$. We use this function to formulate the Euclidean distance among users' preferences.

It can be observed that it is simple to implement Euclidean distance, however, it still have some weakness. When user 1 always tends to give lower mark than other user who originally have high similarity with user 1, the function value will expose to dissimilarity. One explanation for the results could be that user 1 has more stricter evaluation criterion than many other users on same type of movies, but in fact they have same taste of movie. In this case, we try to implement Pearson Correlation Scored to increase accurrency.

2) Pearson correlation similarity

The Pearson correlation coefficient reflects the degree of linear correlation between two variables, which is between [-1, 1]. When the linear relationship between the two variables is enhanced, the correlation coefficient tends to 1 or -1; when one variable increases and the other variable increases, it indicates that they are positively correlated, and the correlation coefficient is greater than 0; when one variable increases and other variable decreases, it indicates that they are negatively correlated, the correlation coefficient is less than 0; if the correlation coefficient is equal to 0, there is no linear correlation between them.

$$\rho = \frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}} = \frac{E\{(X-E(X))(Y-E(Y))\}}{\sqrt{D(X)}\sqrt{D(Y)}} = \frac{E(XY)-E(X)E(Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

$$\rho = \begin{cases} > 0, & positive \\ < 0, & negative \end{cases}$$

3) Cosine similarity:

Cosine-based similarity is a measure of similarity between two non-zero vectors of an inner product space that measures cosine of the angle between them. The scores of user i and user j on the n-dimensional matrix can be represented as vector i, j respectively, then the similarity $sim(i,j)$ between user i and user j is:

$$sim(i,j) = cos(i,j) = \frac{i \cdot j}{||i|| * ||j||}$$

2. Collaborative filtering (CF):

Collaborative Filtering (CF) is to use the past behaviors or opinions of existing user groups to predict what the current user is most possibly to like or what to interest. The algorithm input is a user-item scoring matrix. There are generally two types of output data: the current user's predicted value or rating of the item's likes and dislikes and the list of n items of recommended items (excluding items that the current user has already purchased). The main and most basic implementations are:

1) User-based CF:

User-based CF is an algorithm to find out the users in the data set who have the same preference as the target customers. According to the similarity between the nearest neighbor and the target user, the degree of preference of the target user to the target object is predicted[6].Specifically for each item that the current user has not seen, use the score of user's K neighbors to tiem, then select the top N items with the highest rating of all products and recommend it to the current user. Here is two calculation methods: Euclidean distance similarity and Pearson correlation similarity. These two methods have been introduced above. Then we will calculate the correlation between all the people and calculate the ranking from high to low.

2) Item-based CF：

The Item-based CF can be divided into two parts:

a) Calculating the similarity between items.

b) Generating a list of recommendations based on the similarity of items and the user’s historical behavior.

An item similarity matrix is constructed to describe the similarity between two items. The predicted score of u versus p is obtained by determining the most similar items to the item p and calculating the weighted sum of the scores of these neighboring items; the number of neighbors is limited by the number of items the user has scored, because of this The number of items is generally small, so the process of calculating the predicted value can be completed in a short time allowed by the interactive application on the line[7].

3. Benchmark of recommendation system

According to Jannach(2010), the items(N) which are recommended for user u we can sign as R(u); T(u) means the items which user u prefer on test set; I means the all the items in the data set. The functions of benchmark of recommendation system as follow:

1) Recall

It shows the percentage of user-item ratings in the final recommendation list:

$$Recall = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u T(u)}$$

2) Precision

It shows the percentage of the final recommendation list is appeared user - item score records:

$$Precision = \frac{\sum_u |R(u) \cap T(u)|}{\sum_u R(u)}$$

3) Coverage

It reflects the ability of the recommendation algorithm to discover the items which do not always appear(long-tail):

$$Coverage = \frac{\cup_{u \in U} R(u)}{|I|}$$

4) Popularity

It reflects the average popularity of the items on the recommendation list. If the recommended items are very popular, which means that the novelty of the recommendation is relatively low, otherwise it means that the recommendation results are relatively popular.

$$Popularity = \frac{\log_e(1+P)}{N}$$

Dataset Engineering

1. Data overview

This recommended system uses the MovieLens 100K data sets. The data set is divided into 6 files: movie score u.data, movie category u.genre, data set summary u.info, movie information u.info, occupation list u.occupation, user information u.user. Dataset has been split into train set and test set, percentage is 80% and 20% respectively.

2. Date preprocessing

1) UserID, MovieID is the same as original form. Occupation is processed into continus number form (0-20). Genre, title are ignored. Age, Gender are transformed into 0 and 1. Zipcode, rating are targets.

2) Rating score are transformed into two classifications, 4-5 represents 1, 0-3 represents 0, which will be used in logistics regression optimization.

Implementation and Result

1. User-based CF

In this method, dataset is calculated with Euclidean Distance similarity and Pearson correlation similarity respectively. The specific process is as follows:

1) Euclidean Distance similarity

a) Calculate the K users closest to the users who need to be recommended using the Euclidean distances similarity formula

b) Obtain the movie collections evaluated by the K users, set a score threshold, for example, the threshold is equal to 4, and the movie whose screening score is greater than or equal to the threshold will be added to the recommendation list.

c) Recommend N the highest rated movie in the recommended list as the final recommendation.

k	5	10	20	40	80
recall	57.910	75.320	86.835	92.765	95.870
precision	0.808	0.681	0.571	0.485	0.423
coverage	79.636	82.182	84.000	84.909	85.091
popularity	4.295	4.161	4.010	3.851	3.695

Fig.2 Benchmark Table of user-based CF (Euclidean Distance similarity)

d) As can be seen from the table, when k increases, the recall and the coverage increase accordingly. However, the key indicator, precision and popularity decrease. This is because K determines the interest of other users who are similar to your interests when User-based CF makes recommendations for you. If K is larger, the more people refer to it, the result will become closer to the global hot items. Moreover, because the number of users and the number of movies are too many, the number of movies that interact with each other is small, the sparsity of the data will be large, and the similarity of the obtained users is not reliable. It is difficult to make predictions, so the accuracy will decrease, and the coverage will increase.

2) Pearson correlation similarity

a) It is basically the same as the above steps. The difference is that the distance formula needs to be replaced by the Pearson correlation similarity formula, and then the score of the unevaluated movie that needs to be predicted before is recommended. Here the average value is needed, which means calculating the average for each user, and then calculate the average of the scores for each movie that the user has not evaluated.

b) In this method, it is necessary to predict the rating score of each movie for each user, which results in considerable time and computer memory when calculating the following values. The result is that, we only can calculate the top N recommendation list. For benchmark result, we cannot calculate it due to the high time complexity and calculation power limitation. It will be improved in future to do, which is expected to reduce the amount of calculation and improve the code efficiency.

2. Item-based CF

In this method, dataset is calculated with cosine similarity. The specific process is as follows:

1) The cosine function is used to calculate the similarity between the two movies and stored in the cosine similarity matrix. In this case, the dictionary is used instead of the matrix.

2) After obtaining the similarity between the items, Item-based CF calculates the interest of user u on an item j by the following formula:

$$p_{u,j} = \sum_{i \in N(u) \cap S(j,k)} w_{ij} r_{ui}$$

3) The more similar an item is to an item of interest in the user's history, the more likely it is to get a higher ranking in the user's recommendation list. Find the highest N scores.

4) Calculate precision, recall, coverage, and popularity to get the following table (N=10):

k	5	10	20	50	100
recall	9.730	9.955	9.905	9.610	9.670
precision	20.636	21.113	21.007	20.382	20.509
coverage	23.455	17.333	14.727	13.455	13.030
popularity	5.387	5.481	5.507	5.504	5.510

Fig.3 Benchmark Table of Item-based CF (cosine similarity)

5) It can be seen that the precision (precision rate and recall rate) of the recommended system is not linear relation with the parameter K. We can find that, choosing K=10 will result in higher rate of precision and recall. Therefore, choosing the right K is important for obtaining high accuracy of the recommended system. For popularity, as K increases, the popularity of the results will gradually increase, but when K increases to a certain extent, the popularity will not change significantly. Meanwhile, increasement in K will reduce the coverage of the system.

3. Comparison of User-based CF and Item-based CF

According to Liden[7] ,in the user-based method, with the increasing number of users, "nearest neighbor search" within a large number of users will become the problem of the algorithm. An item-based method replaces the similarity between users by calculating the similarity between items. For items, the similarity between them is much more stable, so the similarity calculation step with the largest workload can be completed offline, which greatly reduces the online calculation amount and improves the recommendation efficiency.

UserCF recommends to the user items that are of interest to users who share a common interest with him. ItemCF recommends to the user items that are similar to the items he liked before. Therefore, the recommendation results of User-based CF focus on reflecting the hotspots of small groups similar to user interests which is social. The recommendation results of Item-based CF focus on maintaining the user's historical interest which is personalization [9].

	User-based CF	**Item-based CF**
Complexity	It is suitable for situations where there are fewer user. if there are many users, the complexity of calculating user similarity matrix will be very higher.	It is suitable for situations where the number of items is obviously less than the number of users. If there are many items, the cost of calculating the similarity matrix of items will be very high.
Real-time performance	The user has a new behavior and does not necessarily cause the recommendation result to change immediately.	The user has a new behavior, which will definitely cause the recommendation result to change immediately.
user personalization	It can be applied in areas where user personalization is not obvious[9]	Better performance in long-tails items, and it is suitable for areas where users have strong personal needs

Fig.4 Comparison Table of user-based CF and Item-based CF

After comparison, we can summarize the application scenarios of the two algorithms.

1) News Recommendation

There are three reasons for using User-based CF in news recommendation:

a) Users' interests are not particularly specified. Most users are most likely to watch popular news.

b) Items update faster than new users join, and for new users, can recommend him the most popular news.

c) The number of users of such websites is relatively stable, and the cost of maintaining user similarity matrix is relatively small.

2) Books, e-commerce and movie websites items Recommendation

a) In these kind of websites, users' interests are more specified and lasting, which means that they are not so sensitive to the popularity of items.

b) The updating speed of items on these websites will not be particularly fast. It is acceptable for them to update the similarity matrix of items once a day without causing too much loss.

c) The number of items on this kind of website is relatively stable, and the cost of maintaining the similarity matrix of items is relatively small.

Model Optimizations

1. User-based CF optimized by Logistic Regression Optimization

1) The logistic regression algorithm is used to optimize User-based CF result, because in dataset the rating score has been divided into 1-5, there is no need to treat this problem into a multi-class problem, so applying logistic regression algorithm can make this problem into a two-category problem. Therefore, in data preprocessing, we mark the label of rating >= 4 as 1, and the label of rating < 3 as 0.

2) Because the model is based on user-based CF, the category of movies and the projection time are not considered when predicting, so our dataset has user id, movie id, user age, user gender, user occupation, and user postcode.

3) Subsequently, using user-based Euclidean distance to find K users with the highest similarity. Then their recommended movies are added to the prediction list. According to the user ID recommended by the user, the trained logistic regression model is used to predict the recommended probability of these movies. Here we set a threshold equals to 0.6, If the probability of film is greater than 0.6, which is what should be recommended to the user. On the contrary, the film will not be recommended.

4) Select the top N movie recommendation with the highest recommendation probability. Then calculate precision, recall, coverage, and popularity to get the following table (N=10):

k	5	10	20	40	80	160
recall	38.390	39.995	40.435	40.545	40.570	40.580
precision	0.960	0.851	0.791	0.761	0.745	0.738
coverage	27.091	27.273	27.455	27.818	28.121	28.182
popularity	4.622	4.555	4.491	4.441	4.404	4.381

Fig.5 Benchmark Table of optimized user-based CF (Euclidean Distance similarity)

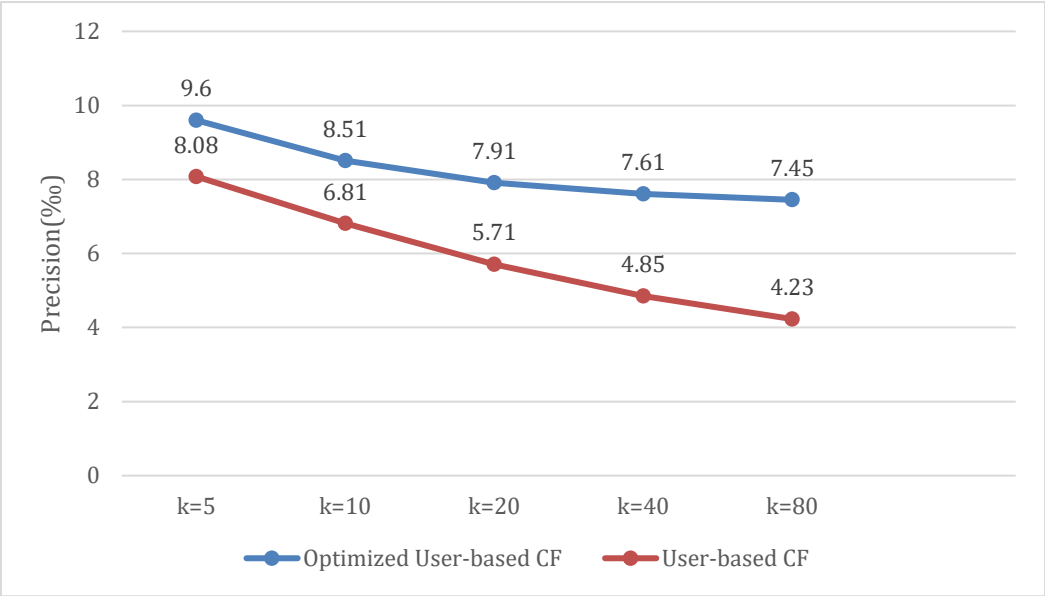

Fig.6 Comparison between optimized user-based CF and original user-based CF

2. Optimization analysis:

It can be seen from figure 6 that after applying logistic regression, compared with the original user-based CF, the precision of recommended movies has increased with the same K and N. Meanwhile, the coverage and the recall will not be as abnormal as the original user-based CF, which indicates that the optimization is successful. However, the shortcomings of the user-based CF recommendation algorithm still exist, for example, this test cost at least eight hours, which shows that the time complexity is significantly large.

Conclusion

By implementing Collaborative filtering recommendation system by 2 basic methods (User-based CF and Item-based CF) with 3 basic algorithms (Euclidean Distance similarity, Pearson correlation similarity and cosine similarity), facts and results can be summarized as follows:

1. User-based CF Versus Item-based CF

For e-commerce industry, the number of users is generally much larger than the number of goods, at which time the computational complexity of Item CF is low. in this case, the recommendation of Item CF has become a key method to guide users to browse. Item-based collaborative filtering algorithm is currently the most widely used recommendation algorithm in e-commerce like Amazon.

In non-social networking sites, the intrinsic link of content is an important recommendation principle, which is more effective than the recommendation principle based on similar users. User-based CF is a better choice in social network sites.

2. User-based CF Optimization

It can be found the limitations of user-based CF that if the number of users is extremely large, it results in extreme sparseness of user rating data and user matrix. In order to solve this issue, dataset is transformed with less classification in terms of rating, and a modified User-based CF based on logistics regression is introduced to optimize the result of original one. The result shows that the modified algorithm has improved the precision with the same K and N, which proves that it is a feasible method to optimize multi-classification issue.

Future Work

1. Optimize the item-based CF by applying regression by using the same method as user-based CF. furthermore, logistic regression can be directly applied to predict the user's most recommended movie scores to recommend.

2. It can be improved by implementing a self-constructing clustering algorithm to reduce the dimensionality related to the number of products. Similar products are grouped in the same cluster and dissimilar products are dispatched in different clusters. (2016)

3. Improve user-rating- data matrix**.** The data sparseness of user-rating-data matrix could be one of reasons to result in low performance, precision and personalized of

recommendation in basic user-based CF. It can be improved by avoiding huge workload which is produced by the computing pair-wise users' similarity.

4. Implementing cold start. It still has some shortages and deficiencies, such as cold-boot problem. When developers lack the data of new users, the system may not provide better recommendations. And at same time, it could not consider the differences of situations, like recommend differently based on scenarios and user's mood. Meanwhile, it also could not provide personal and minority interest.

References

1.https://en.wikipedia.org/wiki/Euclidean_distance.
2.Zhou, T., Chen, L. and Shen, J., 2017, July. Movie Recommendation System Employing the User-Based CF in Cloud Computing. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)* (Vol. 2, pp. 46-50). IEEE.
4.J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive 3.F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," Recommender Systems Handbook, vol. 1-35, no. 3, pp. 1–35, 2011.
algorithms for collaborative filtering," in Fourteenth Conference on Uncertainty in Artificial Intelligence, 2013, pp. 43– 52.
5.Q. Li and B. M. Kim, "An approach for combining content-based and collaborative filters," in International Workshop on Information Retrieval with Asian Languages, 2003, pp. 17–24.
6.Zhou, T., Chen, L. and Shen, J., 2017, July. Movie Recommendation System Employing the User-Based CF in Cloud Computing. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)* (Vol. 2, pp. 46-50). IEEE.
7.Linden, G., Smith, B. and York, J., 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1), pp.76-80.
8.Liao, C.L. and Lee, S.J., 2016. A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electronic Commerce Research and Applications*, *18*, pp.1-9.
9.Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G., 2010. *Recommender systems: an introduction*. Cambridge University Press