COMP9417 Machine Learning Project

Recommender System Using Collaborative Filtering

COMP 9417, 2019 S2

Group Members:

Jiayi Cui z5124061.

Xiaoqing Zhang z5143873

Zedong Li z5158623

Zhiwei Wang z5135560

1. Introduction

With the rapid development and popularity of Internet technology, many movies will be released on the Internet after they are shown in cinemas, which led to the increasing popularity of watching movies on the Internet. Therefore, movie websites pay more attention to people's movie-watching habits on the Internet. In the face of a large amount of movie information, it is important to have a perfect movie recommendation system.

This article focuses on recommendation systems based on collaborative filtering algorithms. Recommend movies to users by analyzing their ratings of movies (Goldberg et al, 1992). The system is built in the Python language. Using code to solve matrix similarity and recommendation scores can effectively increase the time of the algorithm. The advantage of this project is to share the preferences of others and recommend similar information. The disadvantage is that for new users, since the recommended quality of the system depends on the historical data set, the recommended quality is poor, and there are sparseness and scalability issues.

2. Implementation

The film recommendation system is based on massive data mining (Breese et al, 1998). It analyzes the user's historical data to understand the user's needs and interests, and thus actively recommend the movie of interest to the user. The essence is to establish the user and the movie. contact. A complete recommendation system usually consists of 4 modules: extract data, user similarity calculation, recommendation module, and evaluation module.

2.1 Extracting files: create two matrices, one based on the user, with userID as row, moiveID as the column, and another based on the movie, with moiveID as row and userID as the column. User similarity and item similarity can be calculated together. Enhance the accuracy of the recommendation system. There are three ways to measure the similarity between users, including the following three methods: cosine similarity, Pearson similarity and Euclidean distance (Sarwar et al, 2001).

2.2 Cosine similarity: the user rating is treated as an n-dimensional vector project space. If the user does not rate the item, the user's rating on the item is set to 0, and the cosine angle between the vectors can similarity measure between users. Let the scores of user *i* and user *j* on the two-dimensional space be represented as vectors *i*, *j* respectively, then the similarity sim(*i, j)* between user *i* and user *j* is：

$$sim(i,j) = cos(\vec{i},\vec{j}) = \frac{\vec{i}.\vec{j}}{\|\vec{i}\|\|\vec{j}\|}$$

2.3 Pearson similarity: we introduce the Pearson correlation coefficient to measure the linear correlation between two variables, because sometimes we will encounter that because the data between two users is due to data expansion, one party has large data and one party has small data, but the two are called obvious linear relationships (Hill et al, 1995).

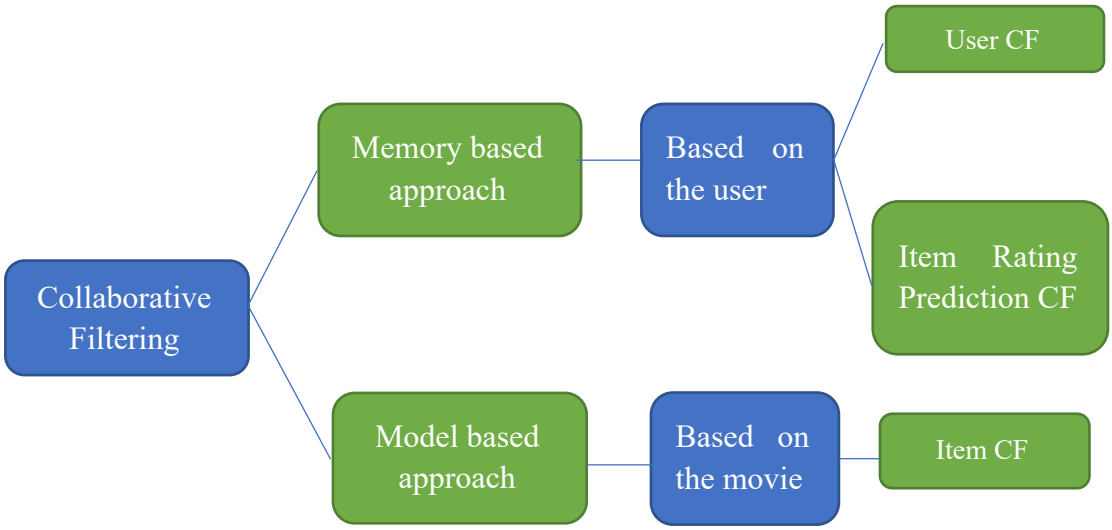
Pearson (-1~1): "-1" refers to a completely negative correlation. "1" is a completely positive correlation. "0": Not relevant. The Pearson formula is expressed as:

$$\rho_{x,y} = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{(\sum X^2 - \frac{(\sum X)^2}{N})(\sum Y^2 - \frac{(\sum Y)^2}{N})}}$$

2.4 Euclidean distance: Find the similarity of each user (or each movie) in a two-digit array. The Euclidean metric is a commonly used distance definition, referring to the actual distance between two points in a two-bit space. So the closer the user is, the closer the similarity and hobbies are. Euclidean formula in two-dimensional space is expressed as:

$$f(x,y) = 1/(1 + \sqrt{\sum_1^N (x_i - y_i)^2})$$

This report will recommend relevant movies to users based on three different aspects.



User CF : Suppose the recommendation system needs to recommend the movie to user A. First, obtain a list of movies that user A and other users have commented at the same time (named temp_neighbor), and then use cosine theorem to calculate the similarity between user A and each other user who likes each movie (Dempster et al, 1977), obtain a new list of neighbors ranked by A similarity with his interest, and finally generate a recommended movie list according to neighbor, recommend to User A and use Pearson coefficient to judge the selection accuracy (Thiesson et al, 1998).

Item Rating Prediction CF: Based on the above system, Grover (2017) claims that the system predicts the user's rating of a recommended movie based on the rating of a movie by the neighbor and the similarity between the neighbor and the user.

Item CF: An item-based CF recommends items to users that are similar to items they were interested in in the past.

Each user's interest is limited to some aspects. If two items belong to a user's interest list, then two items may belong to a limited number of fields; if two items belong to many users' interest lists, then they may belong to the same field. So if a user is interested in one, he is likely to be interested in the other. Finally, calculate the items he is most interested in and recommend them to him.

In addition, the movie recommendation system recommends the movie according to the user's previous preferences. If a similar set of a certain user cannot be found, the system will recommend the user's popular movie (the movie that is scored the most).

3.Results

Firstly, the three different methods of recommendation system (User CF, tem Rating Prediction CF, Item CF) are based on cosine similarity for longitudinal experiments, select the best method; then carry out the lateral contrast experiment on cosine similarity and Euclidean distance on the same data set, compare The recommended quality of the three algorithms proposed in this paper is analyzed and the test results are analyzed.

The similarity measure method uses the cosine similarity to experiment on the data set and calculates the average accuracy of the recommended results. The number of neighbors increases from 1 to 36, and the interval is 2^1.

Figure 1 shows the User CF algorithm based on cosine similarity. It can be seen from the figure that the accuracy of this algorithm is better and the accuracy can be maintained at about 50%.

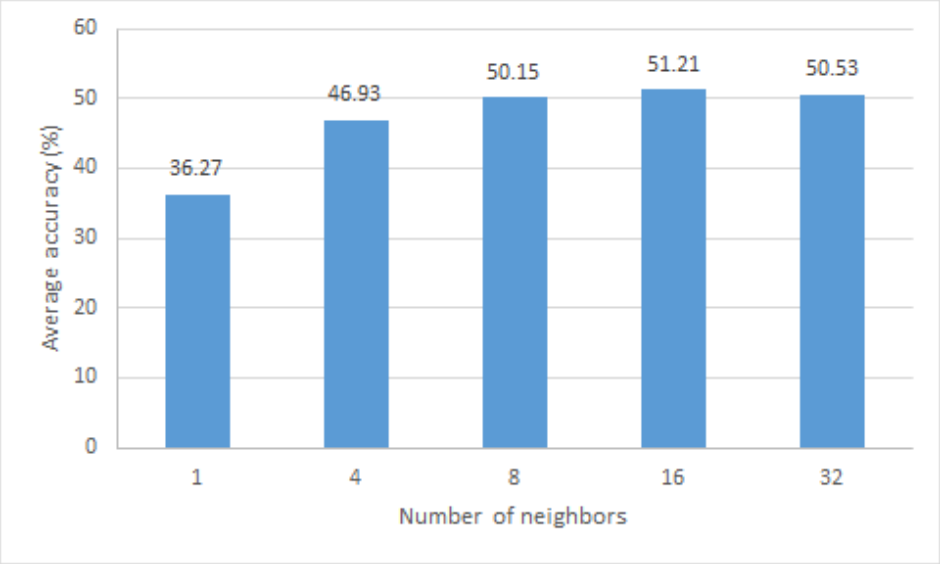


Fig.1 User CF (cosine similarity)

Figure2 shows the User CF algorithm based on cosine similarity and Euclidean distance similarity. As seen from the figure that the accuracy of this algorithm based on cosine similarity is better, generally.

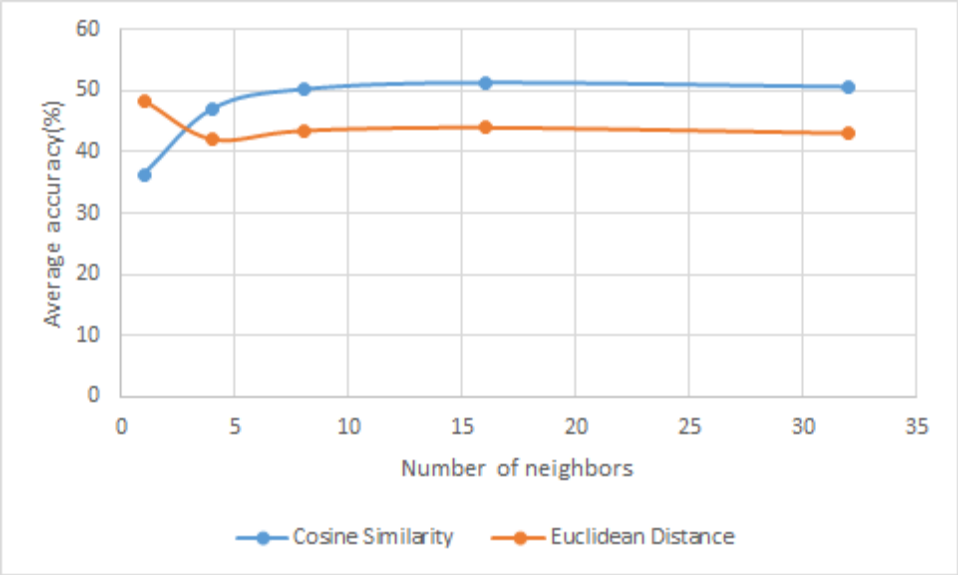


Fig.2 Comparison of accuracy of User CF

Figure 3 illustrates the Item Rating Prediction CF algorithm based on cosine similarity and correlation similarity. The average accuracy of this algorithm is around 65%, and which slowly decreases with the number of neighbors increasing. In additions, it is shown that the average accuracy of cosine similarity is lower than the correlation similarity at the same number of neighbors. As a result, the correlation similarity measure is better.

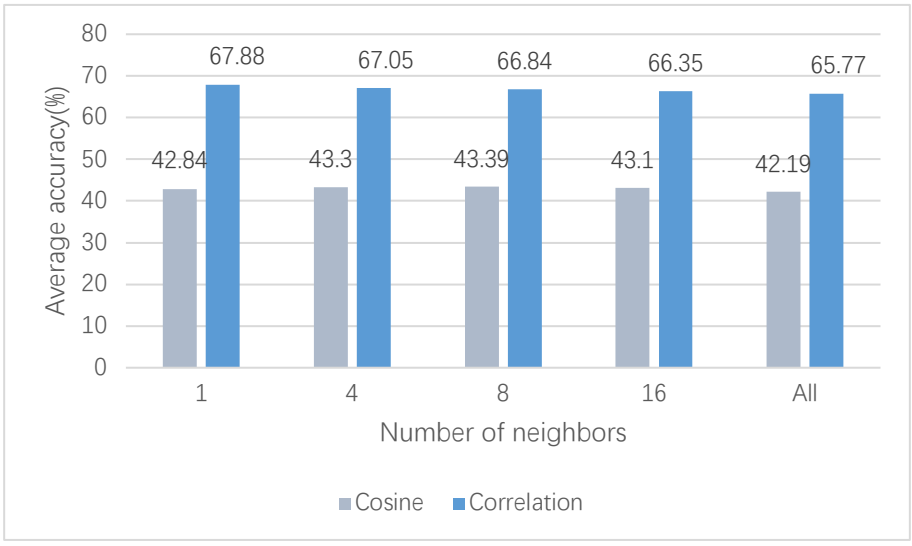


Fig.3 Item Rating Prediction CF (cosine and correlation)

Figure 4 explains the Item Rating Prediction CF algorithm based on cosine similarity and Euclidean distance similarity. As can be seen from the figure that the average accuracy of Euclidean distance similarity is better, but when the similar set is particularly small and especially large, the accuracy of the two is almost the same.

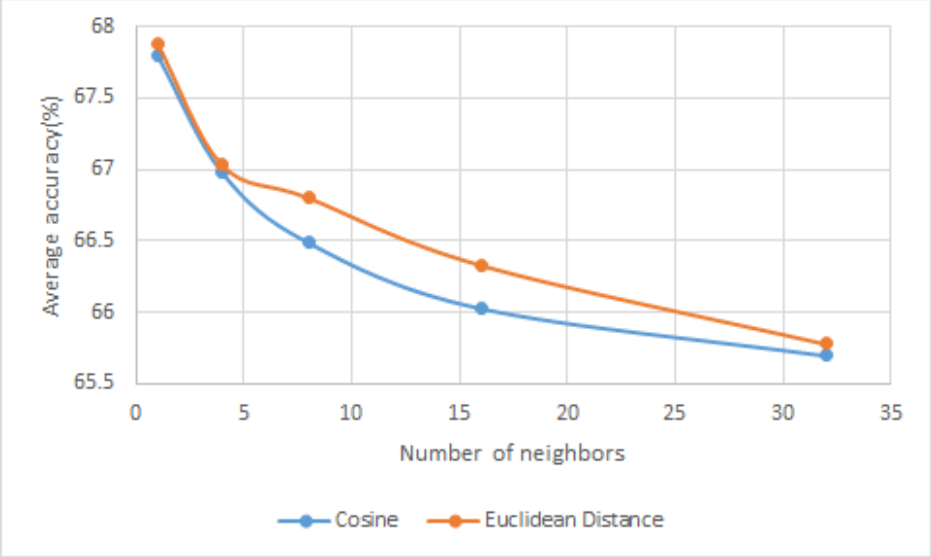


Fig.4 Comparison of accuracy of Rating Prediction

Figure 5 shows the Item CF algorithm based on cosine similarity. The column shows that the accuracy of this algorithm is better and the accuracy can be maintained at about 40%.

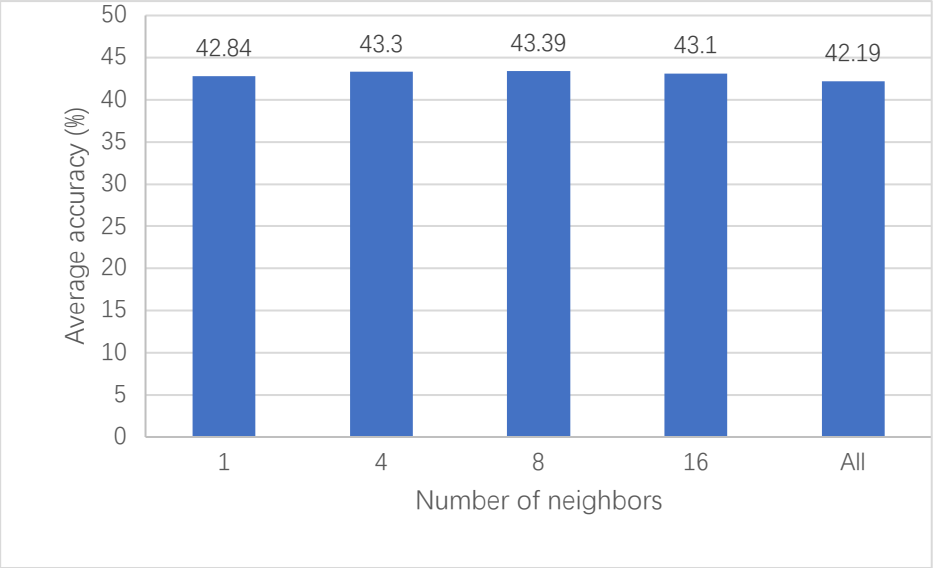


Fig.5 Item CF(cosine similarity)

Figure 6 shows the Item CF algorithm based on cosine similarity and Euclidean distance similarity. From the figure that the accuracy of this algorithm based on cosine similarity is better, however, as the number of movies the user has seen increases, the accuracy of using Euclidean distance similarity increases, while the accuracy of cosine similarity decreases.

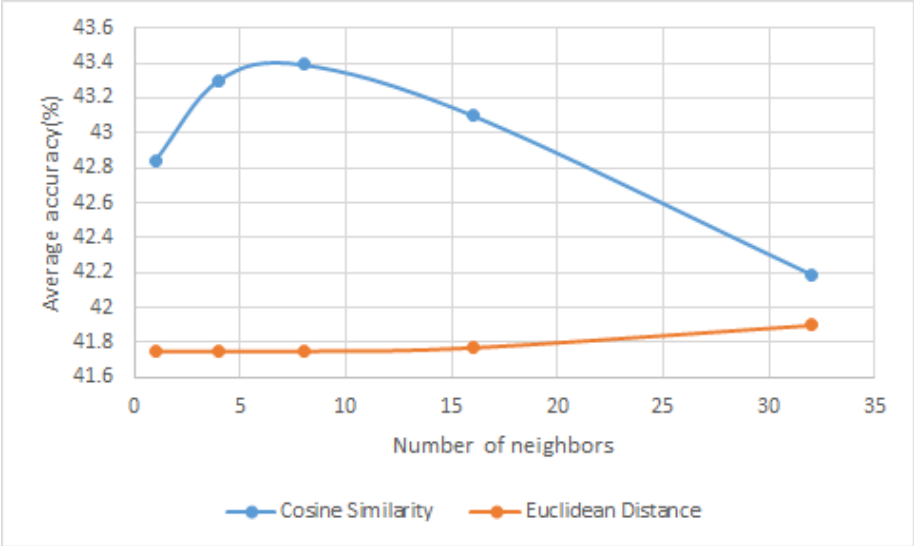


Fig.6 Comparison of accuracy of Item CF

It can be seen from figure 7 that from the perspective of a single user, that is, given a user, to see if the recommendation list given by the system is diverse, that is, to compare the similarity between the items in the recommended list, it is not difficult to think of this measurement method. The diversity of Item CF is not as good as User CF, because the recommendation of Item CF is the most similar to what was seen before.

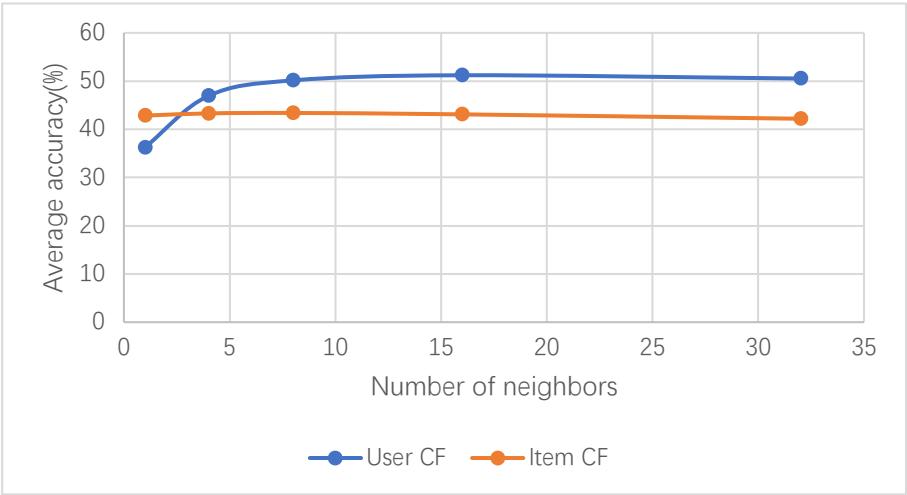


Fig.7 Comparison of accuracy of User CF and Item CF

In order to test the effectiveness of the proposed algorithm, we use cosine similarity and Pearson similarity as the similarity metrics to calculate the average accuracy. The number of neighbors increases from 1 to 36 with an interval of 2[1].

The biggest difference in the collaborative filtering methods of the three movie recommendation systems proposed in this paper is that the similarity methods for finding nearest neighbors are different. It can be seen from the test results that the recommended quality of the cosine similarity measurement is higher than that of the Euclidean method. In the collaborative filtering recommendation algorithm based on the project score prediction, the user is initially predicted by the similarity between the projects. The scoring of the project makes the users have more common scoring items, and then the similarity between users is calculated by the cosine similarity measurement method. The experimental results show that the collaborative filtering recommendation algorithm based on project scoring prediction has the highest recommendation quality.

Conclusion

This report firstly analyzes the problems of cosine similarity, Euclidean distance similarity and Pearson similarity measure in calculating the neighbor of the target user. For the above problems, three collaborative filtering recommendation algorithms are proposed. Experimental results show the collaborative filtering recommendation algorithm based on project score prediction can effectively solve the extreme sparsity of user score data and the recommendation quality of the recommendation system is significantly improved when the user score data is very sparse.

Future Work

With the advent of the recommendation engine, the way users get information from simple and targeted data search to more advanced information discovery is more in line with people's usage habits. Nowadays, with the continuous development of recommendation technology, the recommendation engine has achieved great success in

e-commerce (e-commerce, such as Amazon, Dangdang) and some social networking sites (including music, movies and book sharing, such as Douban). This further shows that in the Web2.0 environment, in the face of massive data, users need to be more intelligent and better understand their needs, tastes and preferences for information discovery mechanisms.

In the future, we can use collaborative filtering algorithms to better integrate neural networks and face recognition. More artificial intelligence fields can be combined to better accelerate the progress of industrial production and the development of human civilization. In addition, in the age of more and more information and expression types, the old information classification and filtering system can not be satisfied, we have more reasons to expect to use collaborative filtering methods in the future.

References

Breese, J. S., Heckerman, D. & Kadie, C. (1998) Empirical analysis of predictive algorithms for collaborative filtering, *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc.

Dempster, A. P., Laird, N. M. & Rubin, D. B. J. J. o. t. R. S. S. S. B. (1977) Maximum likelihood from incomplete data via the EM algorithm, 39(1), 1-22.

Goldberg, D., Nichols, D., Oki, B. M. & Terry, D. J. C. o. t. A. (1992) Using collaborative filtering to weave an information tapestry, 35(12), 61-71.

Grover, P. (2017) Various Implementations of Collaborative Filtering.

Hill, W., Stead, L., Rosenstein, M. & Furnas, G. (1995) Recommending and evaluating choices in a virtual community of use, *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2000) Analysis of recommendation algorithms for e-commerce, *EC*.

Sarwar, B. M., Karypis, G., Konstan, J. A. & Riedl, J. J. W. (2001) Item-based collaborative filtering recommendation algorithms, 1, 285-295.

Thiesson, B., Meek, C., Chickering, D. M. & Heckerman, D. (1998) Learning mixtures of DAG models, *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc.