Recommender Systems: Collaborative Filtering and Content-based Recommender System

Xuechao Yan[1], Shuhan Qi[2], Chang Chen[1]

[1]Department of Informatics, King’s College London, London, WC2R 2LS, United Kingdom
[2]Northeast YuCai Bilingual School, Shenyang ,110000, China

1207529008@qq.com

Abstract. There are three algorithms of recommender systems proposed by this paper, which are item collaborative filtering(itemCF), user collaborative filtering(useCF) and content-based recommender system(CBRS). The principal goal of this paper is to try to ascertain which algorithm has the highest precision, after training based on the same dataset. In accordance with the data we chose and ceaseless testing, we observe itemCF contains the most accurate rate. However, we theoretically and empirically conceive each algorithm owns different advantages and drawbacks, should be used in the specific circumstance.

Keywords: recommender systems, collaborative filtering, content-based recommender system.

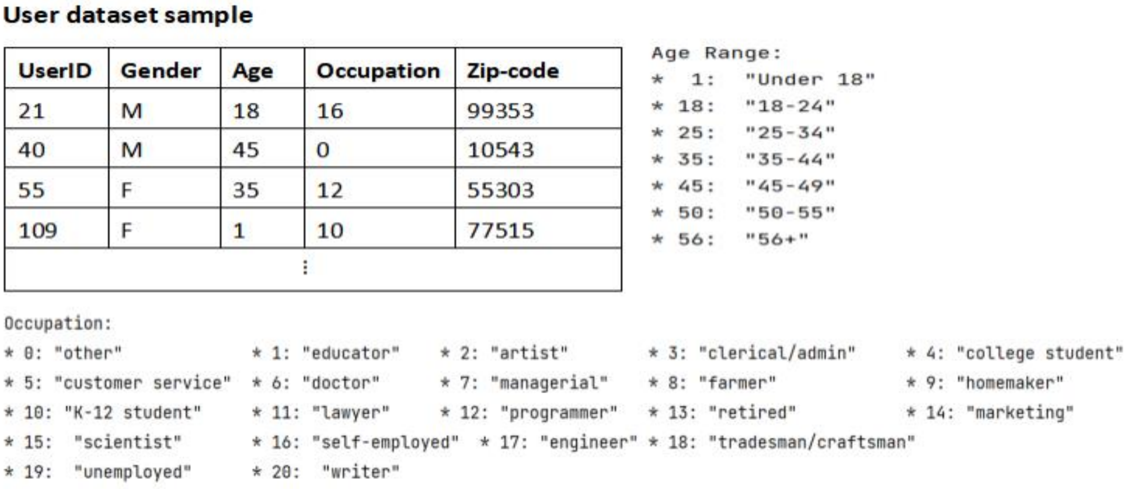
1. Introduction

With the progress of times and the rapid development of the internet, massive data has appeared in people's daily life, forcing us to enter an information explosion era, which generates abundant information at every moment. However, as enormous data enrich life, there is a new challenge on human decision making, some of the information may not be concerned and required, slowing down the searching and selecting speed and increasing the complexity. On the other hand, the information producer, who generates data and sends it to the user, suffers this circumstance simultaneously. Due to people having different tastes, they rack their brains to separate and post information based on people's interests. In this situation, the recommender system is applied to serve both users and information producers by generating significant recommendations about information that users are interested in [1].

Recommendation systems have wide applications on the Internet and e-commerce [2]. Almost all Internet platforms have applied recommendation systems, such as content recommendations for information news/film and television dramas/knowledge communities, like Youtube [3], product recommendations for e-commerce platforms, like Amazon [1]. As above mentioned, these recommender systems are implemented by diverse algorithms, but most of them are following the basic concept, based on the user's browsing habits, determining the user's interests, by discovering the user's behaviour, recommending the appropriate information or products to the user, meeting the user's personalized requirement [4].

In this paper, we particularly focus on three different recommender systems, which itemCF, userCF, CBRS, the working principle and performance of each system. The dataset about movies, called MovieLens, which is intended for training and testing recommender systems, was introduced in the next part, demonstrating how we pre-process these data before implementing the algorithm. In the method section, we provide the principle and logic of the system separately and the mathematical knowledge required in the algorithms. Eventually, the performance of the recommender systems will be analysed and evaluated, which concentrates on the strength and weaknesses and functionality, in the analysis part by comparing the testing outcome.

2. Dataset introduction



User dataset sample

UserID	Gender	Age	Occupation	Zip-code
21	M	18	16	99353
40	M	45	0	10543
55	F	35	12	55303
109	F	1	10	77515
⋮				

```
Age Range:
*  1:  "Under 18"
* 18:  "18-24"
* 25:  "25-34"
* 35:  "35-44"
* 45:  "45-49"
* 50:  "50-55"
* 56:  "56+"
```

```
Occupation:
* 0: "other"            * 1: "educator"    * 2: "artist"       * 3: "clerical/admin"      * 4: "college student"
* 5: "customer service" * 6: "doctor"      * 7: "managerial"   * 8: "farmer"              * 9: "homemaker"
* 10: "K-12 student"    * 11: "lawyer"     * 12: "programmer"  * 13: "retired"            * 14: "marketing"
* 15:  "scientist"      * 16: "self-employed"  * 17: "engineer" * 18: "tradesman/craftsman"
* 19:  "unemployed"     * 20:  "writer"
```

Figure 1. User dataset sample.

Table 1. Movie dataset sample

Movie ID	Title	Genre
1	Toy Story (1995)	Animation | Children’s | Comedy
6	Heat (1995)	Action | Crime | Thriller
100	City Hall (1996)	Drama | Thriller
153	Batman Forever (1995)	Action | Adventure | Comedy | Crime
⋮		

Table 2. Rating dataset sample

User ID	Movie ID	Rating	Time stamp
1	527	5	978824195
1	2321	3	978302205
2	2427	2	978299913
3	1261	1	978297663
⋮			

In this experiment, the dataset used was published by the University of Minnesota, called the MovieLens dataset [5], collected and provided by the GroupLens Research project from the MovieLens website. It involves 1 million rating data (integer of 1 to5) about 3900 movies, generated by more than 6000 users. Three data files, named distinctively users.dat, movies.dat and ratings.dat built up this dataset. As shown in Figure 1, table 1 and 2, the dataset in every file is stored like the sample above.

In order to use the MovieLens dataset in the itemCF and userCF, we split the rating data in the file "rating.dat" into two parts. First is the training containing 80% of the original datasets, and the other is the test set involving the rest of the dataset. However, CBRS processes the MovieLens dataset in another complex method, combining the three files in the dataset into a new one and then splitting it into a training set and test set, with the same ratio as the previous algorithm. The reason we implement this is to extract the user's features and movie's features, using them in our neural network model in CBRS.

3. Method

3.2. Collaborative filtering

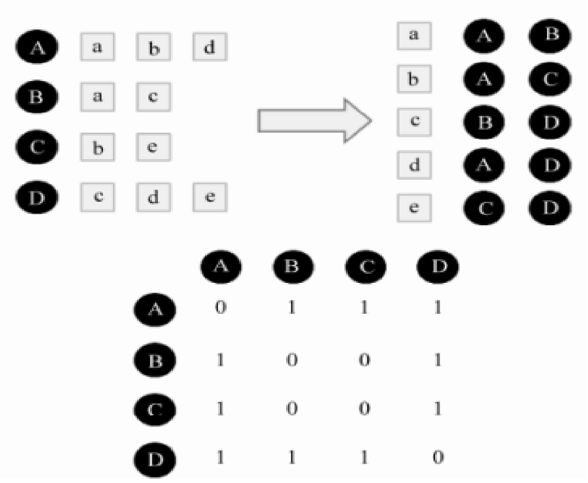
UserCF

UserCF is a typical Collaborative Filtering algorithm based on Users, this algorithm includes two main steps:

Step 1: Given the user u and the user v we set N(u) represents the collection of items the user u is interested in and N(v) represents the collection of items the user v is interested in. Then we calculate the similarity of interests by cosine similarity formula (1) here:

$$W_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{| N(u)||N(v)|}} \qquad \text{Cosine Similarity Formula (1)}$$

In this way, we can get the similarity of interest between all users, but we realise that the time complexity of this algorithm has reached O(N^2), and the efficiency becomes lower because many users have no common preference at all, that is, the numerator will be 0, so we can use an inverse table matrix as shown in Figure 2 to solve this problem. With this matrix, we can just exclude the users which have the numerator as 0.



	A	B	C	D
A	0	1	1	1
B	1	0	0	1
C	1	0	0	1
D	1	1	1	0

Figure 2. inverse table matrix sample.

Step 2: After getting the user similarity, we started calculating the user's interest in the item. The user u's level of interests in item i can be calculated by User's Interests Formula (2) here:

$$p(u, i) = \sum_{v \in S(u,K) \cap N(i)} W_{uv} \, R_{vi} \qquad \text{User's Interests Formula (2)}$$

S(u, K) is the set of K users who have the closest interests with user u. N(i) is the set of the users who made actions on item i. Since we want to predict the level of user u's interest in item i, we should pick out the users made actions on item i from S(u, K) . Therefore, we take the intersection between S(u,K) and N(i). Wuv is the similarity of interest between user u and user v and Rvi represents user v's interest in item i. Rvi uses implicit feedback data of single behaviour, so Rvi is equal to 1. After training through all the training sets, we start to recommend and justify the accuracy.

ItemCF

To some extent, ItemCF is similar to the UserCF because they are both algorithms that belong to collaborative filtering. The basic principle of these two algorithms is the same. ItemCF includes two main steps. The first step is calculating the similarity between items and the second step is calculating the level of the user's interest in the item.

Step 1: We use the cosine similarity formula (3) to calculate the similarity between items.

$$W_{ij} = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i)||N(j)|}} \qquad \text{Cosine Similarity Formula (3)}$$

N(i) represents the set of users who like i items, and N(j) represents the set of users who like j items. Same as we noted in UserCF, we still need to use the inverse table matrix to exclude the part of users with no relations with the item i and j to increase the efficiency of calculation.

Step 2: After getting the item similarity, we use the User's Interests Formula (4) below to calculate the level of the user's interest in item j.

$$p(u,j) = \sum_{i \in S(j,K) \cap N(u)} W_{ji} R_{ui} \qquad \text{User's Interests Formula (4)}$$

S(j, K) represents the set of k items most similar to j items. N(u) indicates the collection of items that the user like. Wji indicates the similarity between items. Rui is the user u's interest in item i. Rui could be set to 1. After training through all the training sets, we start to recommend and justify the accuracy.

3.2. Content-based recommendation

Content-based recommender system (CBRS) systems implement by collecting items information that users used to like in form of ratings and determining recommendations according to the items information. The approaches for this system can be summarized in the following four steps:

Step 1: We combine all features from users and movies files used as input for CBRS.

Step 2 and 3: An extra embedding layer is used to map four attribute information of users into vector representation and input it into the fully connected layer, then adding these four vectors as the user feature. The movie ID and the movie type are also mapped instead of using the embedding layer. The names of the movie need to get the long vector through a text convolution network. Finally, add all the vectors as the movie feature. During the addition of the feature vectors, different fully connected layers are used to map different feature vectors to equal-length vectors, then it is convenient to merge into a vector. we can train the model on CPU, setting the learning rate 0.01 and training 5 epochs. the result shows that loss is converging. This means the training system takes effect.

Step 4: The method of calculating the similarity between features is cosine similarity, as in Cosine Similarity Formula (5):

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{A + B} = \frac{\sum_i^n A_i \times B_i}{\sqrt{\sum_i^n (Ai)^2 + \sum_i^n (Bi)^2}} \qquad \text{Cosine Similarity Formula (5)}$$

ere θ is the angle between two vectors, n is the number of vectors of a user, Ai and Bi is the separate vector in the total feature vectors. In this equality, θ can express the degree of similarity. θ equals 0 means completely similar.

we can calculate the similarity matrix of users and all movies using the feature we trained. Before we recommend, we can add some random factors to ensure the novelty of the recommendation.

4. Analysis

Table 3. Precision test table.

	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10	Mean
ItemCF	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007	0.3007
UserCF	0.3057	0.3057	0.30255	0.3057	0.3057	0.3056	0.3056	0.3056	0.3057	0.3057	0.30565
CBRS	0.2796	0.2934	0.2855	0.2681	0.2748	0.2824	0.2593	0.2778	0.2672	0.2704	0.27585

Table 4. Runtime table (unit: second).

	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9	Test10	Mean
ItemCF	483	483	480	478	468	476	466	465	461	461	472.1
UserCF	230	238	255	235	235	240	232	244	227	228	236.4
CBRS	1.187	1.203	1.116	1.227	1.193	1.092	1.120	1.202	1.101	1.134	1.1575

The Table 3 records the outcomes about the precision of 10 tests for these 3 recommender systems. It is obvious to see that userCF has the highest accuracy, around 0.30565. On the contrary, CBRS is the lowest, which is 0.27585. The most significant point is that itemCF retains at the same value, 0.3007, representing the stability of this recommender system. Therefore, under this circumstance, the most worth to suggest recommender system is UserCF.

The Table 4 records the runtime of each test in seconds, and the last column of the table shows the mean of the runtime of each algorithm. According to this table, it is obviously to see that ItemCF takes the longest average time to run, which is 472.1 seconds. By contrast, the time of running of CBRS is lowest among three algorithms, which is 1.1575 seconds. The efficiency of UserCF is moderate among the three algorithms, which is 236,4 seconds. Comparing these three algorithms we can see that CBRS runs much faster than other algorithms which indicates that the neural network used in CBRS has an incredible ability to process the data and output the result.

So if we consider the precision and time cost simultaneously, in most circumstances CBRS would be the first choice because of its good efficiency with a precision which is not bad.

Strength/weakness of each RS:

- UserCF:

UserCF based on similarity of use to recommend information compares each user's behaviours, then it can generate the recommendations. Therefore, it has high efficiency when the user group is small, as the user group increases, the complexity of algorithms will rise rapidly, consuming more time and space. The reason that Amazon and Netflix doesn't choose userCF, but the Social feature, userCF contains, strongly fulfils the requirement of a recommender system for news [6]. People more concentrated on hot spots in the news area, and userCF is suitable for tracing the hotspot and head items.

- ItemCF:

ItemCF resembles UserCF, but it uses similar items to create suggestions [7]. Consequently, the performance and efficiency will decrease as the number of items increases. However, it is proper to implement in the user has stable interest and need to recommend tail items, as the result lots of e-commerce, like Amazon decide it as a recommender system.

- CBRS:

The advantage of CBRS is to recommend new products to users, without rating voted by users. Therefore, even if the database does not contain user interests, it does not affect the accuracy of the recommendation results. But the requirements for datasets are higher, when we want to use CBRS, the dataset should be comprehensive enough to be used by CBRS.

5. Conclusion

In this paper, we perform a study on the three recommendation system algorithms, comparing each other to distinguish their superiority and drawbacks. Through the experiment, we recognise that userCF is more proper for the scenarios with fewer users because the massive amount of users causes the user similarity matrix becomes expensive to solve. The same drawback occurs for itemCF as the amount of products increases when it exceeds the number of the user group [8]. In contrast, CBRS doesn't cost as much as the others but needs a more comprehensive dataset to build the neural network model [9]. However, this experiment ignores the instability of human rating behaviours mentioned by Basu, C., Hirsh, H., & Cohen, W. in 1998 [10], some people prefer to give high feedback and vice versa, leading to an unbias rating, eventually decreasing the precision of the recommender system. Therefore, we still need to improve the algorithms used to balance the rating. We believe that every recommendation system can fulfil a specific requirement, so it is imperative to choose an appropriate algorithm when we need them.

References

[1] Melville, P., & Sindhwani, V. (2010). Recommender systems. Encyclopedia of machine learning, 1, 829-838.

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

[3] Covington, P., Adams, J., & Sargin, E. (2016, September). Deep neural net-works for youtube recommendations. In Proceedings of the 10th ACM confer-ence on recommender systems (pp. 191-198).

[4] Burke, R. (1999, July). Integrating knowledge-based and collaborative-filtering recommender systems. In Proceedings of the Workshop on AI and Electronic Commerce (pp. 69-72).

[5] Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4), 1-19.

[6] Wang, Z. (2020.3). Deep learning recommendation system. Beijing: Electronics Industry

[7] Jinxiushinian. (2019). UserCF & ItemCF.https://www.jianshu.com/p/8934dc19c7ee

[8] Zhangxiansheng-ninhao. (2021). Comparison of advantages and disadvantages of the collaborative filtering algorithm UserCF and ItemCF. https://blog.csdn.net/weixin_35154281/article/details/120377181

[9] The fourth paradigm celestial hub. (2019, August, 12). Recommendation sys-tem: Content-based filtering and its pros and cons. [Web log post]. Retrieved from https://zhuanlan.zhihu.com/p/77765572

[10] Basu, C., Hirsh, H., & Cohen, W. (1998, July). Recommendation as classifica-tion: Using social and content-based information in recommendation. In Aaai/iaai (pp. 714-720).