2019 9th International Conference on Education and Social Science (ICESS 2019)

Research on Collaborative Filtering Recommendation Algorithm for Personalized Recommendation System

Yi Ren [1,a], Jingke Xu [1,b], Jie Huang [1,c] and Cuirong Chi [1,d]

[1.]Shenyang Jianzhu University Liaoning, China

[a.]1181134429@qq.com;[b.]1401308379@qq.com;[c.]1269028746 @qq.com;[d.]726733255 @qq.com

Keywords: Recommendation system; Collaborative filtering; Personalized recommendation

Abstract. In the era of information overload, recommendation system emerges at the historic moment. Collaborative filtering recommendation system algorithm is the most mature and extensive recommendation technology. This paper simulates the scene of film recommendation, applies different recommendation algorithms under the collaborative filtering recommendation system, and compares the recommendation quality to obtain a relatively objective experimental conclusion.

Introduction

With the development of information technology and Internet, information overload brings great challenges to users and producers. The recommendation system is created to deal with problems such as unclear user needs and long tail information. The recommendation system models users' interests by analyzing their historical behaviors so as to recommend information that meets their interests and needs [1,2]. Nowadays, collaborative filtering recommendation system is applied to various e-commerce platforms, such as amazon, JD Mall, taobao, toutiao.com and so on. Netflix is one of the most successful companies in the field of film recommendation. It applies the item-based recommendation algorithm in the collaborative filtering recommendation algorithm. Thanks to its successful recommendation system, Netflix has generated huge revenue growth [3]. Aiming at three popular model-based collaborative filtering recommendation algorithms, this paper applies them to the simulated film scoring environment and analyzes them. These include: Slope One, SVD, SVD++.

The basic idea behind collaborative filtering is that if users have preferences in the past, they will have similar preferences in the future. Based on this, if two users' preferences overlap a lot, they can recommend projects to each other [4]. This technique is called collaborative filtering because the selection of interested items comes from filtering results from a large number of collections, and users implicitly collaborate with others. At present, collaborative filtering recommendation algorithms are roughly classified into two categories: memory-based collaborative filtering algorithm and model-based collaborative filtering algorithm [5]. Among them, the collaborative filtering recommendation algorithm based on memory is divided into the nearest neighbor recommendation based on users and the nearest neighbor recommendation based on items.

Collaborative Filtering Recommendation Algorithm

The Nearest Neighbor Recommendation Based On Users.

By using the target user's historical behavior data to find out the user's preferences for goods. Then, find other users who have similar preferences with the target user in the past, that is, find the nearest neighbor of the target user. For item i that the target user has not seen, the predicted value is calculated by the score of its neighbor to i. If the results are positive, the target user will be recommended [6].

The Nearest Neighbor Recommendation based on Items.

The relationship between items can be quantified by the rating of different users on different items or other behaviors (such as adding wish lists, sharing with others, evaluation, etc.), so as to find the nearest neighbor of the item. If the target user has a positive attitude towards item i and item j is the neighbor of item i, then item j is recommended to the target user. In specific practice, users' attitudes towards items will be quantified so as to make more accurate recommendations [7].

Model-Based Recommendation Algorithm.

Model-based recommendations start to process raw data offline, such as using some dimensionality reduction techniques [8]. At run time, it is only needed to predict or "learn" the model to be able to predict.

Recommendation Algorithm of Film Recommendation System

In the Netflix competition in 2009, more teams used model-based recommendation algorithms to effectively improve the prediction accuracy of the recommendation system [9]. So this paper introduces three common model-based recommendation algorithms and applies them to experiments. The experimental data was the movie ratings of more than 30,000 users of Netflix's competition. This paper divides training set and test set according to the ratio of 8:2. RMSE (root mean square error) was used as the evaluation index. The smaller the RMSE value, the better the prediction quality [10].

Slope One Algorithm.

Algorithm idea: Taking a prediction based on a simple linear regression model. Example of algorithm: the following table 1 is the four users rating to three movies, so as to predict the rating of movie 3 by user 1.

Table 1 The four users rating to three movies

	movie 1	*movie 2*	*movie 3*
user 1	3	5	?
user 2	2	4	5
user 3	4		3
user 4	3	4	4

Calculating the average distance between movie 3 and movie 1: (3+ (-1) +1) /3=1, and the average distance between movie 3 and movie 2: (1+0) /2=0.5. According to user 1 user 1's rating of movie 1, the predicted rating of movie 3 is 4. According to user 1's rating of movie 2, the predicted rating of movie 3 is 4.5. Considering that the simultaneous rating of movie 3 and movie 1 is 3, and the simultaneous rating of movie 3 and movie 2 is 2, it is predicted that user 1's rating of movie 3 is:

$$red(user1, movie3) = \frac{4 \times 3 + 4.5 \times 2}{3 + 2} = 4.2$$

In the process of recommendation, in order to prevent users from habitually giving high scores to affect the quality of recommendation, it is necessary to specifically analyze the scoring habits of user 1. If rating 4.2 is a user's favorite category, movie 3 is recommended to user 1. The rating prediction formula of user u to movie j is as follows:

$$pred(u, j) = \frac{\sum_{i \in S(u)-\{j\}} \left(dev_{j,i} + u_i\right) * |S_{j,i}(R)|}{\sum_{i \in S(u)-\{j\}} |S_{j,i}(R)|}$$

User i is used to predict rating of the film; s(u) represents the set of rating elements in u; The mean deviation value on two films is:

$$dev_{j,i} = \sum_{(u_i, u_j) \in S_{j,i}(R)} \frac{u_i - u_j}{|S_{j,i}(R)|}$$

u_i represents rating of user u to movie i; R is the whole rating data; $S_{j,i}$(R) tag contains a rating set for both movie i and movie j.

SVD Algorithm.

Algorithm idea: Taking a prediction based on singular value decomposition model. Algorithm principle: mapping the large and sparse user rating matrix to the low-dimensional space by SVD algorithm, calculating the similarity between ungraded items and other items in the low-dimensional space, calculating the prediction rating, and then sorting the prediction rating from high to low, and return the first N items to recommend users. In other words, important features are extracted by compression matrix and then calculated and recommended. For example, movie features intuitively can be divided into thrillers and comedies.

The user rating matrix M is decomposed into three product, both U and V respectively are the left and right singular vectors, the value of Σ diagonal is singular value.

$$\mathrm{M} = \mathrm{U}\Sigma V^T$$

The matrix V corresponds to the user, and the matrix U corresponds to the item. In general, the entire matrix can be approximately expressed by only observing the most important characteristic of singular value. The common similarity calculation methods include Euclidean distance, Pearson correlation coefficient and cosine similarity.

SVD++ Algorithm.

Algorithm idea: SVD++ algorithm is an improvement of SVD algorithm, which is also an algorithm based on singular matrix decomposition model.

Algorithm principle: SVD++ algorithm introduces implicit feedback on the basis of SVD algorithm, and takes the user's historical browsing data, user's historical rating data, etc., as new parameters. Implicit feedback is combined with the domain model. Main formula are as follows:

(1) Rating prediction:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T\left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j\right) + |R^k(i;u)|^{-\frac{1}{2}} \sum_{j\in R^k(i;u)} (r_{uj} - b_{uj})w_{ij} + |N^k(i;u)|^{-\frac{1}{2}} \sum_{j\in N^k(i;u)} c_{ij}$$

(2) Root mean square error:

$$\min_{q_*,p_*,y_*,w_*,c_*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_u - b_i - q_i^T\left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j\in N(u)} y_j\right) - |R^k(i;u)|^{-\frac{1}{2}} \sum_{j\in R^k(i;u)} (r_{uj} - b_{uj})w_{ij} - |N^k(i;u)|^{-\frac{1}{2}} \sum_{j\in N^k(i;u)} c_{ij})^2 + \lambda({b_u}^2 + {b_i}^2 + \|q_i\|^2 + \|p_u\|^2 + \sum_{j\in N(u)} \|y_j\|^2 + \sum_{j\in R^k(i;u)} {w_{ij}}^2 + \sum_{j\in N^k(i;u)} {c_{ij}}^2)$$

After calculating the user's rating for unknown items, the rating quality was quantified by root mean square error. The significance of adding regularization parameters is to prevent overfitting. The symbols are shown in table 2.

Table 2 The symbols implication

symbol	*implication*
u	user
i	Item i (unknown rating)
j	Items similar to item i (known rating)
k	Rating r_{ui} known set
r_{ui}	User u real rating of item i
$\hat{r}_{ui}$	User u predicted rating of item i
b_u	User u rating deviation
b_i	Item i rating deviation
q_i^T	Factor vector of item i
p_u	Factor vector of user u
N(u)	User u's itmes set with an implicit preference
k	Number of items closest to item i
$S^k(i)$	Set of k items closest to item i
R(u)	All rating set of user u
w_{ij}	Weight from item j to item i
c_{ij}	Offset for user u's implicit preference for item j
λ	Regularization parameter

(3) The algorithm application results:

The test results of the three algorithms are shown in table 3.

Table 3 The algorithm application results

prediction result / *algorithm*	*RMSE value*
Slope One	0.8356574113243.2
SVD	0.8134704664413.3
SVD++	0.816237015507

Conclusion

Due to the differences in the results of running in different environments, this paper conducted tests on multiple machines. Preliminary conclusion: SVD++ algorithm recommendation effect is relatively more accurate. Slope One algorithm runs fast, but it is relatively simple in theory and fails to realize personalized recommendation. SVD algorithm and SVD++ are relatively basic algorithms, on which people keep improving. For example, timeSVD++ algorithm is an algorithm that takes time perception into consideration. In the aspect of recommendation system, people gather wisdom and integrate all aspects of knowledge to make the recommendation system more humanized and intelligent. However, we should continue to explore.

References

[1] G Dror,N Koenigstein,Y Koren and M Weimer. Journal of Machine Learning Research. Vol.18(2012),p.3.

[2] RM Bell and Y Koren. Lessons from the Netflix prize challenge. Acm Sigkdd Explorations Newsletter. Vol.9(2007)No.2,p.75.

[3] Goldberg D, Nichols D, and Oki B M, et al. Using Collaboration Filtering to Weave an Information Tapestry. Communications of the Association for Computing Machinery, Vol. 35(1992) No.12, p.61.

[4] X.W. Meng, S.D. Liu and Y.J. Zhang, et al. Social recommendation system research[J]. Journal of Software, Vol. 26(2015) No.6, p.1356-1372 .(In Chinese).

[5] Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Journal of Nature,Vol.401(1999) No.6755,p.788.

[6] Aggarwal C C, Wolf J L, Wu K L, et al. Proc.of KDD-99 :the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (San Diego,Calif,USA,Aug 15-18,1999). p.201.

[7] Nozomi Nori,Danushka Bollegala and Mitsuru Ishizuka. Proc.of International AAAI Conference on Weblogs and Social Media (ICWSM 2011) 5th.(2011). p.241.

[8] Gediminas Adomavicius and YoungOk Kwon. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. IEEE Transactions on Knowledge and Data Engineering. Vol.24(2012)No.5,p.896.

[9] R.Forsati, M.Mahdavi, M.Shamsfard, and M.Sarwat. ACM Transactions on Information Systems, Vol.32(2014) No.4, p.17.

[10] R. Forsati, I. Barjasteh, F. Masrour, A.H. Esfahanian, and H. Radha, Proc.of the RecSys '15: Proceedings of the 9th ACM Conference on Recommender Systems (Vienna,Austria,September 16-20,2015). p.51.