A New Prediction Approach Based on Linear Regression for Collaborative Filtering

Xinyang Ge, Jia Liu*, Qi Qi, Zhenyu Chen
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
Software Institute, Nanjing University, Nanjing, China
*Corresponding author: liujia@software.nju.edu.cn

***Abstract*—Recommender systems using collaborative filtering help users filter information based on previous knowledge of users' preferences. Most of existing recommender systems make predictions using weighted average method. This paper introduces a new prediction approach based on an effective linear regression model. One fundamental idea behind this approach is that there exist patterns among different users' preferences. And we propose a linear regression model to characterize the inner relationships among different users' rating habits. The major contribution of this approach is that it can make more accurate predictions via utilizing the exact linear correlation indicated by Pearson Correlation Coefficient directly. The preliminary experiments show that our approach can improve the accuracy of prediction thus make recommendations more appealing to users.**

***Keywords*-Collaborative Filtering; Recommender System; Prediction; Linear Regression**

I. INTRODUCTION

Recently, recommender systems have become quite popular for the ability to distinguish what users are really interested in from the others [1]. As the amount of information is increasing explosively, it is more and more difficult for users to identify content effectively. In such situation, recommender systems, as a new technology, help users figure out what they want from potentially overwhelming set of choices.

Collaborative filtering, one of the most successful recommendation technologies, is now widely used in commercial system. The basic idea behind this wonderful technology is to build a community of users and recommend items to a certain user according to many similar users' preferences [2]. For instance, douban.fm is a popular music site that helps you enjoy music. One important feature of this music site is that users needn't to manage their songs list since it will recommend songs based on some "rules". In the process of recommendation, douban.fm accepts users' opinions (both implicit and explicit) about what they are listening, which will help the system to cluster similar users. Gradually, users will be more likely to enjoy what the system recommend, and it's the power of recommender system. Traditionally, collaborative filtering employs two fundamental steps in prediction: first, some users, known as neighbors, are selected due to their similarities to the active user; second, a weighted average of neighbors' ratings are used to predict the rating value. Long-term practices of collaborative filtering reveal that it is effective indeed when compared with content-based recommendation technologies.

The work described in this article was partially supported by the National Natural Science Foundation of China (10871091), the Humanities and Social Sciences Foundation of Ministry of Education of China (10YJC870020, 10YJC630283).

Traditional prediction approaches use weighted average technology with preference scales taken into consideration. However, they neglect specific relationships among user ratings. Furthermore, they simply assume the relationship between the active user's rating and his neighbors' is $f(x) = x$, which we would have detailed discussion in Section III.

To overcome these problems, we proposed a new prediction approach utilizing exact relationships among user ratings. Based on these relationships, recommender systems can make more accurate predictions without compromise of system performance, which means not only lower Mean Absolute Error (MAE) but also good user experience.

The rest of the paper is organized as the follows. The next section will have a brief overview of related work especially on collaborative filtering. The third section would be a complete description of the newly proposed approach based on linear regression. We then introduce ways of evaluation in section four and our experiment with its result in section five. Finally, conclusion is presented in section six.

II. RELATED WORK

In this part, we'll have a brief review of related work, especially the classical process of a whole recommender system.

A. Collaborative Filtering Overview

Collaborative Filtering (CF) has been studied since the last decade. It is one of the most important recommendation technologies owing to its excellent quality and simplicity [3]. A basic assumption of CF is that similar users prefer similar items, or that a user expresses similar preferences for similar items. Thus, rather than distinguishing among great quantity of resources (movies, music, stories, etc.) [4], user-based CF focuses on the similarity among different users and believes similar users share similar interests thus they would prefer the same group of resources. While for content-based technologies, such as text analysis and scanning, it faces difficult challenges on determining inner relationships among different items. Although natural language like English and Chinese has its own grammar, it is still hardly possible for a computer system to understand what people are "talking" about [5]. In fact, many content-based recommending system failed to give satisfactory recommendations due to inaccurate descriptions of items' inner relationships, and the situation goes worse where items are not text-based, such as movies, songs etc.

One fundamental step of CF is to find some "neighbors" who have similar preference with an active user and then make recommendations for this user based on his "neighbors". However, sometimes it is not easy to find similar users since the system has less information about active user, or in other words, the active user is a new user. Some researchers suggest combining content-based technologies with collaborative filtering to deal with the so-called "cold start" problem [6, 7].

To achieve this, researchers have proposed various algorithms, which can be divided into two groups: model-based and memory-based. The former makes predictions via learning a model, such as a cluster model or a Bayesian network model, from historical data while the latter stores raw preference information in computer memory and access it when needed. Memory-based collaborative filtering algorithms can further be divided into two categories: user based and item based. User-based CF focuses on similarities among users while the other centers on similarities among items. In this paper, we will pay more attention to user-based and model-based Collaborative Filtering.

B. An overview of the whole process of recommendation using collaborative filtering

Firstly, the recommender system will receive an identifier of a user for recommendation. For example, a site notices a user logins the system.

Secondly, recommender systems will find his "neighbors" who have similar tastes via some popular algorithms. Actually, different algorithms will generate different neighbors [8], which depend on the calculation of similarities among users.

Thirdly, the system will predict a potential rating of the active user towards a desired item based on his neighbors. The objective of this paper is to introduce a brand-new prediction approach, which has been proved to be more accurate than traditional one.

Finally, after making potential ratings the system will recommend some items the user are most likely to enjoy.

C. Similarity Calculation

Calculating the similarity among users is a crucial step in the whole process mentioned above since it will directly influence prediction accuracy. Among various approaches, Pearson Correlation Coefficient is widely used since it describes how one's rating vector is linearly correlated to another's [9]. In statistics, Pearson Correlation Coefficient ranges from [-1, 1] to indicate linear correlation between two arrays of numbers.

Thus, it is reasonable to use this value as similarity since it indicates how effective it is to use one's rating record to predict the others no matter their preferences are very similar or totally opposite.

In order to simply, we use the term "similarity" to indicate Pearson Correlation Coefficient. More specifically, the similarity between user *u* and *v* is calculated as,

$$sim(u,v) = \frac{\sum_{i\in I_u\cap I_v}(r_{ui}-\overline{r_u})(r_{vi}-\overline{r_v})}{\sqrt{\sum_{i\in I_u\cap I_v}(r_{ui}-\overline{r_u})^2 \sum_{i\in I_u\cap I_v}(r_{vi}-\overline{r_v})^2}} \quad (1)$$

In the formula, r_{ui} denotes the rating value from user *u* to item *i*. $\bar{r}_u$ is the average rating from user *u* to the items which are co-rated by both user *u* and *v*. I_u is a set of items which *u* has rated.

D. Rating Prediction

In order to predict a potential rating from user *u* to item *i*, the system examines similar users' ratings to this item and then makes an aggregation. One basic assumption behind this is the overall similar users' attitudes to a given item can, at least to some degree, represent the active user's attitude. Normally, different neighbors' ratings are averaged using similarity as their weights. And in order to take different users' preference scales into consideration, some algorithm will adjust their ratings by subtracting a user's average rating during calculation. Here is an example to illustrate it. User *A* is more optimistic than *B* and he always rates 4 for those he thinks just so-so, while *B* uses 2. To address this problem, we need to take users' rating habits into consideration and average rating is considered the rating for normal items. So, the formula is just like this,

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v\in\mathfrak{V}} sim(u,v)(r_{vi}-\overline{r_v})}{\sum_{v\in\mathfrak{V}}|sim(u,v)|} \quad (2)$$

where $\mathfrak{V}$ denotes the neighbors of *u*, and $\bar{r}_u$ means average rating of *u* to all items he has viewed, which is slightly different from the definition in similarity calculation phase.

It is difficult for a recommender system to maintain real-time performance if all rating information is adopted by the system. Moreover, noise can be introduced if neighbors with low similarity are involved. Thus, in practice, K nearest neighbor (KNN) is often adopted both for prediction accuracy and system performance [10]. The KNN method identifies *k*, which is determined by dataset, users who are most similar to the active user and use their ratings for prediction.

E. Recommendation

The final objective of a recommender system is to generate accurate recommendations. Since the system now knows the possible preference of this active user towards all items that he has not viewed, it is much easy to recommend items via examining the database.

III. PREDICTION BASED ON LINEAR REGRESSION

After looking again at the traditional prediction formula, we can find that

$$\hat{r}_{ui} - \bar{r}_u = \frac{\sum_{v\in\mathfrak{V}} sim(u,v)(r_{vi}-\overline{r_v})}{\sum_{v\in\mathfrak{V}}|sim(u,v)|} \quad (3)$$

Behind this aggregation, there is a valid assumption that relationship between $\hat{r}_{ui} - \bar{r}_u$ and $r_{vi} - \overline{r_v}$ is a special linear correlation,

$$f(x) = x \quad (4)$$

where x denotes $r_{vi} - \overline{r_v}$ and $f(x)$ represents $\hat{r}_{ui} - \bar{r}_u$.

However, fact is that Pearson Correlation Coefficient represents the degree of linear correlation between two rating

records, or in other words, it reflects how similar two users are according to their attitudes towards what they both rate. So, a natural idea is to use the same formula as the weighted average method but instead of aggregation directly using the special linear correlation above, recommender systems can find out the exact ones among different users' ratings and adjust the ratings via linear regression. Thus, we modify the special linear correlation to a more general one,

$$f(x) = a + bx \tag{5}$$

So,

$$\hat{r}_{ui} - \bar{r}_u = a + b(r_{vi} - \bar{r_v})$$

Finally, it turns to

$$\hat{r}_{ui} = a' + b' r_{vi} \tag{6}$$

in which $a' = a + \bar{r}_u - b\bar{r_v}$ and $b' = b$.

Here comes a challenge of how to determine the value of a' and b' (or a and b). The general process is that a recommender system learns historic data of two given users and then finds the most proper values of a' and b'. So, different learning methods will lead to different value pairs. We proposed an intuitive learning method, which can finally turn this challenge to optimization problems. More details of this method and ideas behind it were discussed at the following sections.

Here is an example illustrating this idea,

TABLE I.

User\Item	I_1	I_2	I_3	I_4
U_1	1	2	3	?
U_2	2	3	4	5

In this example, U_1 and U_2 have both rated three items, and now the system needs to predict the value of (U_1, I_4). And it is easy to find out that,

$$R_1 = R_2 - 1 \tag{7}$$

in which R_1 is the rating record of U_1's.

So, according to U_2's rating towards I_4, it is reasonably to guess U_1's rating would be 4. Moreover, the formula above is so-called regression function, which we would have detailed discussion below.

A. Regression Function

In order to have a complete and accurate description of regression function in the proposed model, we will introduce some definitions below.

Def. $E(v_1, v_2) = \sum_{i=1}^{n} |v_1[i] - v_2[i]|$, in which v_1 and v_1 are two vectors having the same dimension.

Def. A regression function from v_2 to v_1 is a linear function $f_{v_1 v_2}(x) = a + bx$ that will make value $E\big(v_1, f_{v_1 v_2}(v_2)\big)$ the least.

Here is an example,

$$v_1 = [1, 2, 3\,],\ v_2 = [2, 3, 4]$$

Now $E(v_1, v_2) = |1 - 2| + |2 - 3| + |3 - 4| = 3$

In fact, the regression function from v_2 to v_1 is $f_{v_1 v_2}(x) = -1 + x$. So, after this regression, $v_2{}' = [1, 2, 3]$, $E\ (v_1, f_{v_1 v_2}(v_2)) = E\ (v_1, v_2{}') = |1 - 1| + |2 - 2| + |3 - 3| = 0$.

Actually, not all situations are as obvious as the example above. Downhill simplex algorithm will help determine such a regression function at any situation.

B. Weighted Average Using Regression Function

The main objective of regression function is to simulate the active user's rating preference based on his neighbors' rating records. One fundamental belief behind this approach is that there exists a pattern in rating records between any given two users, and it is helpful in prediction to find the pattern, or the inner relationship among users' rating habits. There is also an assumption behind this approach that patterns keep stable in a short period of time. Since people's preferences are stable in a short period of time, the assumption mentioned above is accepted theoretically and would be further proved in our experiments. Finally, the approach can be summarized as prediction based on neighbors' ratings and patterns between each neighbor and active user. Thus new prediction approach can be represented as the following,

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathbb{U}} sim(u,v) f_{uv}(r_{vi})}{\sum_{v \in \mathbb{U}} |sim(u,v)|} \tag{8}$$

where f_{uv} denotes the regression function from *v* to *u*.

IV. EVALUATION

Evaluation measures how well a recommender system is in several aspects including prediction accuracy, system performance. Due to the diversity of recommender systems, there is no unified metric that can evaluate all aspects of a recommender system. Furthermore, different metric can be combined for overall evaluation. In this section, we focus more on accuracy, which is generally considered the most important criteria.

Statistical accuracy metrics measure how close the predicted ratings generated by various kinds of algorithms are to actual user ratings. Mean Absolute Error – the average absolute error between the predicted rating and the actual rating given by a user - is one of the most widely used predictive accuracy metric in evaluation of recommender systems. The advantages of MAE are obvious. Firstly, it is simple thus well understood, which makes it easily implemented and convincing. Furthermore, as many researchers use it to evaluate predictive accuracy, comparisons are easily achieved among different approaches. To specifically, the formula below shows how to calculate MAE,

$$MAE = \frac{\sum_{i=1}^{n} |\hat{r}_i - r_i|}{n} \tag{9}$$

The lower the MAE, the more accurate the prediction is. Besides MAE, Root Mean Squared Error (RMSE), and Correlation are also used as statistical accuracy metric.

Decision support accuracy metrics centers on how helpful the recommendation is, or in other words, are users interested in what the system recommend. Thus, this kind of metrics simplifies the prediction into binary result, good or bad.

In our experiments, we use MAE as our choice.

V. EXPERIMENTS

In this section, a series of experiments designed to demonstrate these two different prediction approaches are described and a variant, k neighbors, is taken into consideration. First we explore the effect of number of neighbors on each approach and then compare them at each point. Then we will examine tendency of each approach and make the conclusion that the approach based on linear regression model is more accurate.

A. Dataset

In our experiment, we use MovieLens dataset (MLDS) since it is one of the most widely used datasets thus the results can be more convincing [11]. This dataset consists of 1682 movies, 943 users, 10,000 ratings (1-5) and other information including timestamp of ratings, occupations of users. Besides, in order to remove noise of cold start problem, all users in this dataset has at least rated 20 movies. In practice, we use 80% ratings of each user for training and remaining 20% ratings for testing. And the dataset comes with 5 predefined splits (u[n].base for training and u[n].test for testing, in which n ranges from 1 to 5) for unified use.

B. Regression Function Generation

As mentioned above, not all cases are as easy as the example showed in the previous example. So, it is essential to find a general approach to calculate a, b in regression function $f(x) = a + bx$.

In mathematical optimization theory, downhill simplex algorithm is a popular algorithm for numerically solving linear programming problems. In practice, it is remarkably efficient in computing regression functions.

C. Experiments Result

In presenting our result, we firstly investigate the influence of parameter k in both models and then compare accuracy and trends of them.

The traditional approach computes ratings using weighted average technology while our approach gives predictions based on a linear regression model. To evaluate the trends of different algorithms on the value of k, we performed the experiment where k ranges from value 1 to 25 in increments of 2. And we can observe that k does affect the quality of prediction.

As we can see in Fig.1, the traditional method seems to achieve better results when k is small. However, as k becomes larger, the approach based on linear regression model becomes much better than traditional one. After studying trends of both approaches, we can see both get stable in the end.

After showing that the approach based on linear regression model can provide more accurate predictions, we focus more on performance. In fact, in a short period of time the relationship between two distinct users' rating habits are stable thus precomputation can be achieved to increase performance. Besides, other technologies such as caching can be applied to a real recommender system in order to maintain efficient recommendations.

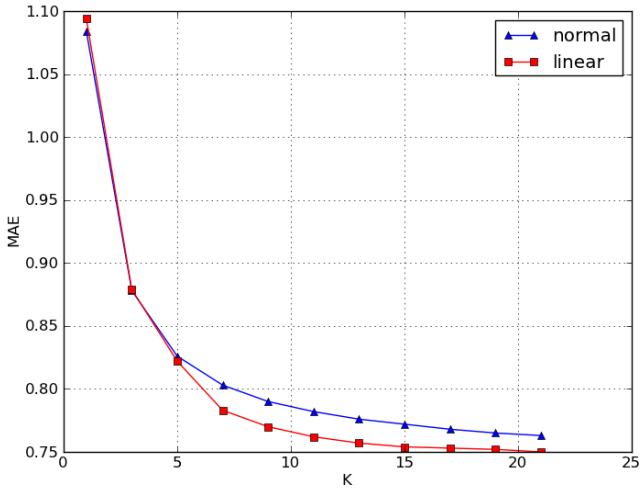

Figure 1. MAE trends

D. Limitations

Firstly, performance is not taken into consideration during this experiment. However, performance is even more important in a real system because of the impact of user experience on users. In other words, users have no time to "wait" for recommendations and system needs to make a balance between accuracy and performance. To solve this problem, a lot of techniques can be used when implementing a real system such as caching technique and delay refreshing. Secondly, as MovieLens is a dataset of movies and ratings, it is unclear how results would be when this new approach is applied to other domains. Movies are special items due to the relationship between users and movies. Finally, as the focus of this approach is to model the inner relationships among different users' habits of rating, more ways based on it can be applied when predicting. Moreover, experiments should be tested on more different datasets and thus can be more reliable.

VI. CONCLUSION AND FUTURE WORK

Recommender systems are a powerful technology for both service providers and users. For an e-commercial site, this kind of technology can help it to increase sales by recommending products that users are most likely to buy. And for users, it can help them get rid of overwhelming data and find what they really want in a more elegant way.

In this paper, we proposed a new prediction approach based on linear regression. This approach aims at modeling the inner relationship among different users' rating habits and make predictions based on them. We experimentally evaluate the quality of both approaches in terms of Mean Absolute Error

(MAE). Our results show that approach based on linear regression can generate more accurate recommendations.

Further research issues include making modifications to other recommending phases and combining them to produce better results. Moreover, higher dimensional functions can be experimented, where similarity calculation needs to be retaken into consideration for its incompatibility with Pearson Correlation Coefficient.

VII. REFERENCES

[1] Resnick, P. and Varian, H.R.1997. Recommender Systems. Commun. ACM 40, 56-58

[2] Dunn, G., Wiersema, J., Ham, J., and Aroyo, L.2009. Evaluating Interface Variants on Personality Acquisition for Recommender Systems. In: Houben, G.J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) User Modeling, Adaptation, and Personalization. LNCS, vol. 5535, pp. 259-270. Springer, Heidelberg.

[3] Hu, R. and Pu, P. 2010. A Study on User Perception of Personality-Based Recommender Systems. In: P.De Bra, A.Kobsa, and D.Chin (Eds.): UMAP 2010, LNCS 6075, pp 291-302.

[4] Linden, G., Smith, B, and York, J.2003. Amazon.com recommendations: Item-to-Item collaborative filtering. IEEE Internet Computing, Jan/Feb.: 76-80

[5] Raymond J. Mooney, Content-Based Book Recommending Using Learning for Text Categorization, 1999 ACM

[6] Rashid, A.M., Albert, I. , Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., and Riedl , J., Getting to Know You: Learning New User Preferences in Recommender System, Proc. Int'l Conf. Intelligent User Interfaces, 2002

[7] Ahn, H.J.2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences 178: 37-51

[8] Herlocker, J.L. and Konstan, J.A., Content-Independent Task-Focused Recommendation, IEEE Internet Computing, 2002

[9] Adomavicius, G. and Tuzhilin, A.2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowledge and Data Eng., 17, 6(2005), 734-749

[10] Sarwar, B.M., Karypis, G., Konstan, J.A., and Riedl, J.2000. Analysis of recommendation algorithms for E-commerce. In Porceedings of the 2nd ACM Conference on Electronic Commerce. ACM, New York. 285-295

[11] Lathia. N., Halies, S., and Capra, L. kNN CF: A Temporal Social Network. In Preceedings of Recommender Systems, 2008