GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

AN ADJUSTMENT SIMILARITY MEASURE FOR IMPROVING PREDICTION IN COLLABORATIVE FILTERING

Yasser El Madani El Alami[*1], El Habib Nfaoui[2], Omar El Beqqali[3]
[*1,2,3] FSDM Sidi Mohammed Ben Abdellah University, Fez, Morocco

ABSTRACT

"Collaborative filtering" (CF) methods provide a good solution for recommendation systems. One of the main phases in CF is the neighborhood selection phase. It relies on selecting users according to their similarity to the active user. Unfortunately, almost all used similarity measures do not take into account many useful parameters associated with the users that can help computing similarity more accurately. This paper presents a comparative study of adjustment similarity measures that combines Pearson correlation with various set-similarity measures (such as Jaccard similarity) as a correction coefficient. The focus is to improve computing similarity phase among users (items) to reflect as much as possible their real relationships. Finally, experiments using FilmTrust dataset show that combing Jaccard coefficient with Pearson similarity give more predictions accuracy than the traditional collaborative filtering.

Keywords*—Collaborative filtering; Recommender system; Similarity measure; set-similarity measure.*

I. INTRODUCTION

Ever since the 90s, the amount of information has been increased in exponential way. In order to deal with this problem, there have been a great interest in recommendation systems. According to [1] recommendation systems have been considered as an effective means to reduce complexity in information retrieval. Recommendation system helps users to deal with information overload and provides personalized recommendations, content and services to them. It suggests the appropriate items for each user according to his/her interests. Although, recommendation systems are largely used in both e-commerce applications such as Amazon [2] and academic researches such as MovieLens [3], they are being extended to other domains such as digital libraries, e-learning, etc. Authors in [4] show that the most used methods in recommender systems are collaborative filtering methods. They rely on users' evaluation to identify "useful" items to these users. Unfortunately, many typical drawbacks are noticed in collaborative filtering approaches which weaken thereafter the quality of the recommendations. Our work relies on using a reformed similarity measure that combines the well-known Pearson correlation with set-similarity measures as adjustment coefficient. It's applied to the user based collaborative filtering approach which yields good results. This paper is organized as follows: in section 2 we give an overview of the traditional collaborative filtering. Section 3 describes our proposals. In section 4, we present the experiments and evaluation results of our proposals. At the end, we give some perspectives, and a conclusion.

II. BACKGROUND

In this section we focus on presenting a detailed background that our work is based on. Collaborative filtering paradigm is based on mutual aid of users who share similar tastes and preferences to recommend the suitable items. As a result, CF based systems can predict a user's rating (or behavior) for an unknown item [5] or create a top-N list of recommended items [6] for a target user (called active user).

The formal definition of collaborative filtering approach is: we denote the space of users by C and the space of items by I. let U be a utility function that measures usefulness of item i to user c and:

$$U: C \times I \rightarrow R \qquad (1)$$

Where R represents the space of rating values of user u to item i. Rating represents how a given item is interesting to a particular user. R can be a discrete set (R= {like, dislike} or positive integer) or a real number in a given range (for example R= [1, 5]). According to [7], we can distinguish between two classes of CF algorithms: memory-based algorithm and model-based algorithms. In what follows we focus on the memory based approach

Memory based algorithm

Memory based approach builds predictions based on the whole set of ratings that users assigned to items before. Previous ratings are grouped in a matrix referred to as ratings matrix. In this matrix, the cell rsj refers to the rating given by user s to item j according to a specific scale (for example 1-5 rating scale). In most cases, this ratings' matrix is typically sparse [8] as most users do not rate viewed items regularly. Besides, the most popular algorithm in memory based is neighbor-based algorithm which predicts ratings based on either users [7] who are similar to the active user or similar items to the requested item [9]. In what follows, we focus on the User-Based CF method (UBCF). Generally, According to [10] there are three steps into processing a recommendation based on CF system: i) Representation, ii) Neighborhood formation, iii) Recommendation generation.

Representation: The first step in UBCF consists on building a rating matrix and assigning values to the unrated items to fill the porous ratings matrix. Two processes [11] can be used: Default rating and Pre-processing using average.

Neighborhood formation: The second step consists of measuring similarity between the other users. The most commonly used algorithm is the Pearson correlation. In fact, it has become a standard way of calculating correlation [10]. Using Pearson correlation, similarity between user u_a and u_b is calculated with the following formula:

$$sim_{a,b} = \frac{\sum_{j=1}^{n}(r_{aj}-\overline{r_a})(r_{bj}-\overline{r_b})}{\sqrt{\sum_{j=1}^{n}(r_{aj}-\overline{r_a})^2 \sum_{j=1}^{n}(r_{bj}-}} \quad (2)$$

Where n is the cardinal of the set of items, r_{ai} is the rating given by user a to item j and is the average rating given by user a for all the items he rated. As an output, similarity process returns a user similarity matrix which determines correlation between pairs of users. Thus, building similarity between users allows forming the requested neighborhood. Two techniques have been employed [12]: "threshold-based" where user is considered as neighbor when his/her user similarity exceeds a given threshold, and "K nearest users" where k is given as input. Also, it can be computed for each user.

Recommendation generation: This phase relies on generating predicted rating of user s to item i. It's calculated as aggregation of similarity between the active user and his neighborhood, and their ratings. Various aggregation functions are employed in predictions. The one most used is calculated as the weighted average of neighbors' ratings:

$$p_{s,i} = \bar{r}_s + \frac{\sum_{p=1}^{k}(r_{p,i}-}{\sum_{p=1}^{k}|sim} \quad (3)$$

K represents the size of selected neighborhood.

Therefore, based on computed predictions, recommender system may suggest unknown or new items that the active user may like.

Performance measures

In recommendation system, a great interest has been made in measuring system performance (scalability, accuracy, quality, etc). In our case we are interested by prediction performance measures. In this area, many indicators are used to evaluate the system accuracy. A case in point is Mean Absolute Error (MAE). In fact, MAE is a common way used to measure accuracy based on statistical metric. It calculates the average absolute difference between predicted ratings and real ones:

$$MAE = \frac{\sum_{(s,i)}|p_{s,i}-}{N} \quad (4)$$

In the formula above $p_{(s,i)}$ is the predicted rating for user s to item i, $r_{(s,i)}$ is the real rating given by user s to item i and N corresponds to the number of predicted ratings calculated during the test phase.

One of the major factors in collaborative filtering that greatly influences the recommendation accuracy is the selected similarity measure. In the literature, almost all works are based on the well-known Pearson correlation measure. As presented in formula (2), Pearson correlation measure is based only on the items that are co-rated by both users. It doesn't take into account other decisive parameters such as the items that are rated only by one of the two users or not rated by both of them knowing that these parameters provide meaningful information of how user's preferences are different. Consequently, it increases mainly the margin of error and reduces the confidence interval which leads to inaccurate recommendations.

III. PROPOSED APPROACH

In order to reduce the impact of the fallacious similarity on the computed recommendations, we propose to use other parameters associated with the users x and y. We consider that the users are described using a set of binary attributes also named presence/ absence. The set represents whether an item is rated by the user or not.

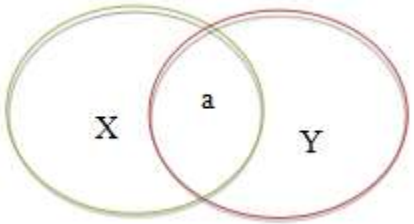

Figure 1 Space of Characteristics

We note the space of the whole users by (figure 1) S= {0, 1}p, X={i/xi=1}represents the set of attributes (the rated items) that user x has, Y={j/yj=1} the set of attributes (the rated items) that user y has, and | | the cardinal of a given set. In what follows, we use four parameters that are presented in many similarity measures such as Jaccard Similarity, Ochiai measure, etc. We note:

a=|X∩Y| the number of attributes (the rated items) which are present in user X and user Y.

b=|X-Y| the number of attributes (the rated items) which are present in user X and not in user Y.

c=|Y-X| the number of attributes (the rated items) which are present in user Y and not in user X.

d=| $\overline{X}$ | the number of attributes (the rated items) which are neither present in user X nor in user Y.

Table 1 Group 1 of measures

Similarity measure	Notation	Definition
Jaccard, 1908	S_{Jac}	$\frac{a}{a+b+c}$
Ochiai, 1957	S_{Och}	$\frac{a}{\sqrt{a+b}\sqrt{a+c}}$
Rifqi et al, 2000	S_{FD}	$\frac{F_{FD}(\varphi) - F_{FD}\left(\frac{\pi}{2}\right)}{F_{FD}(0) - F_{FD}\left(\frac{\pi}{2}\right)}$

The existing similarity measure can be divided into two groups. The first one doesn't take into account the items which are not rated by both users (table 1), contrary to the second group which considers this information as useful (table 2). The tables 1 and 2 below [13] give an overview of the well-known similarity measures in each group.

Rifqi et al similarity measure relies on Fermic-Dirac function defined as follows:

$$F_{FD}(\varphi) = \frac{1}{1+\exp(\frac{\varphi}{}}$$ With $\varphi = \arctan(\frac{b}{}$

Where T is a positive real number, and $\varphi_0 \epsilon [($. These parameters allow a better control of the power of discrimination similarity.

Similarity measure	Notation	Definition
Sokal & Michener, 1958	S_{SM}	$\frac{a+d}{a+b+c+d}$
Russel & Rao, 1940	S_{RR}	$\frac{a}{a+b+c+d}$

Table 2 Group 2 of measures

In our case we use these similarity measures (S_{SM}, S_{Jac}, S_{Och}…) as (noted *Coef)* which correct the computed similarity between two users. These adjustment coefficients rely on the associated parameters to each pair of users. Thus, the new similarity measure is presented as follows:

$$Sim_{xy} = Coef * S_c$$

Where S_c represents the traditional Pearson correlation and coef represents the used adjustment coefficient.

IV. EXPERIMENTS AND RESULTS

A) FilmTrust DataSet

For experiments we employed the dataset of FilmTrust project [14]. It is an academic research project being run by Jennifer Golbeck[1]. It's a movie recommendation website where users can rate and review movies. Users can give their opinion using a value on a rating scale from 0.5 to 4 stars where 0.5 means bad and 4 means excellent. The data is stored as semantic web annotations based on RDF and FOAF. It integrates semantic web-based on social networks with movie ratings so as to compute predictive movie recommendations. The collected dataset consists of 1856 users, 2092 movies and 759922 ratings. Thus, around 80.4% of the global ratings matrix is empty. It means that FilmTrust dataset represents a real situation of sparsity problem.

B) Experiment and results

Our experiment focuses on testing the traditional approach using the Simple Pearson correlation measure (PC), PC combined with S_{Jac}, PC combined with S_{Och}, PC combined with S_{SM} and PC combined with Rifqi2000 measure. (In our tests we set T=0.1 and $\varphi_0 = \frac{\pi}{4}$)

First, we divide the dataset DS into two subset: 70% for the training set and 30% for the test set. After that, We apply the prediction algorithm, as presented above, to the training set. The system predicts the ratings of all users in the data set by repeating this computation three times and then we give the average predicted rating. At the end, we compare the predicted value and the real value of the ratings using the MAE measure.

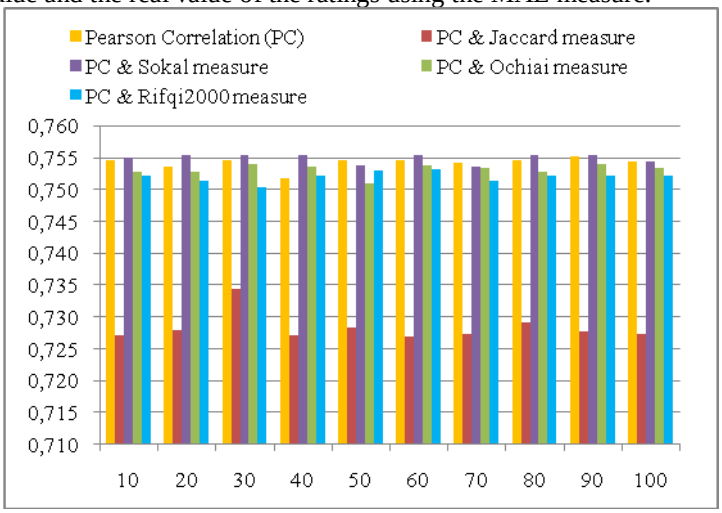

Figure 1 UBCF with reformed similarity measures

As presented in figure 2, we conclude that combining Pearson correlation similarity with Jaccard measure as a correction coefficient gives a better result than the other tested similarity measures

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have studied many combination of reformed similarity measures using a set-similarity as correction factor. This combination take into account many useful parameters associated with the users that can help computing similarity more accurately. As presented before, the combination of Pearson correlation and Jaccard measure provide the best result of the overall tested combinations. As future work, we plan to reduce the effect of scalability and sparsity problems by improving the selection neighborhood phase. We propose to use heuristic methods and include the social network information of users. In fact the focus is to select neighbors who are likely to be reliable to the active user before starting similarities computation phase which is time-consuming if we compute the similarity for all system users. In addition, social networks offer many opportunities for recommendations since people generally use their social networks to obtain reliable and useful information.

[1]https://www.cs.umd.edu/~golbeck/

REFERENCES

1. *K. Joseph, "Introduction to recommender systems: Algorithms and Evaluation," ACM Transactions on Information Systems, vol. 22, no. 1, pp. 1-4, 2004.*
2. *L. Greg, S. Brent and Y. Jeremy, "Amazon.com Recommendations: item-to-item collaborative filtering," Internet Computing, IEEE, vol. 7, no. 1, pp. 76 - 80, 2003.*
3. *M. Bradley N., A. Istvan, L. Shyong K., K. Joseph A. and R. John, "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System," in the 8th international conference on Intelligent user interfaces, Miami, Florida, 2003.*
4. *A. Gediminas and T. Alexander, "Towards the Next Generation of Recommender Systems:A Survey of the State-of-the-Art and Possible Extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, p. 734–749, 2005.*
5. *D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using collaborative fil- tering to weave an information tapestry," Communications of the ACM, vol. 35, no. 12, pp. 61-70, 1992.*
6. *M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," ACM Transactions on Information Systems (TOIS), vol. 22, no. 1, pp. 143 - 177, 2004.*
7. *J. S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in the Fourteenth conference on Uncertainty in artificial intelligence, San Francisco, USA, 1998.*
8. *P. Melville and V. Sindhwani, "Recommender Systems," in Encyclopedia of machine learning, Springer US, 2010, pp. 829-838.*
9. *B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in the 10th international conference on World Wide Web, New York, USA, 2001.*
10. *B. Sameer, "Recommender System Algorithms," Toronto, Canada, 2008.*
11. *E. Vozalis and K. Margaritis, "Analysis of Recommender Systems' Algorithms," in Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA-2003), Athens, Greece, 2003.*
12. *S. Gong, "A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering," Journal of Software, vol. 5, no. 7, pp. 745-752, 2010.*
13. *M. Rifqi, "Mesures de similarité, raisonnement et modélisation de l'utilisateur," Paris, France, 2010.*

[Online].Available: http://trust.mindswap.org/FilmTrust/. [Accessed 1 September 2013].