ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

A measure of Temporal Contextual Information on Trust based Recommender Systems

[1] Ankur Chaturevdi, [2] Dilip Kumar Sharma
[1] CEA Department, GLA University, Mathura

Abstract - **In an era of information age, recommender system helps users to make an effective decision. Collaborative filtering is one of the techniques to provide a personalized recommendation to users. Collaborative filtering based recommender technique provides the recommendation by aggregating ratings from similar users to predict ratings for an active user (who wants a recommendation). The similarity has a greater impact because it acts as a criterion to identify a group of similar users whose ratings will be merged to generate a recommendation for the new item for an active user. However, there are a lot of issues in Collaborative filtering for e.g. data sparsity and cold start, which can be removed by incorporating trust information. We propose a methodology to include temporal context information in providing accurate rating prediction along with Trust matrix and also propose a framework to analyze the performance of Trust-based recommender algorithms on Film Trust dataset which includes temporal context information.**

Keywords: Collaborative filtering, Recommender Systems.

1. INTRODUCTION

Recommender Systems is considered as an application of Machine learning and Big data. Collaborative filtering is one of the most prominent techniques in recommender systems. According to Collaborative filtering users having similar taste in the past are likely to favor the same items in future. Rating information are very sparse in nature. Including trust value in recommender systems gives a direction to provide users with recommendation which is based on past behavior and social trust values. It is

noticed that people get influenced easily by what their friends recommend. The approaches for Collaborative filtering are classified into two categories [1] [2].

I. Memory based Approaches: Algorithms based on it try to find similar users by looking into an entire user space which is not good in practice as well as time taking activity. Every user is considered as a part of a group of people having same interest. These Algorithms compute user Similarity using PCC.

II. Model based Approaches: It gives an approach for the system to learn from training data, and then make intelligent predictions for the test data. Usually SVD method and regression models can be used for numerical ratings[6][8].

Merging trust in Recommender Algorithm remove two drawbacks of Collaborative filtering [1] [2] [4].

I. Rating metrics are sparse means that very few ratings are available. Also Data sparsity means that there is a problem in finding similar users whose past behavior is same as an active user.

II. Cold start deals with a problem of generating accurate recommendation to those users who are inactive in system or those users who generally rate less than 4 or 5 items.

There are two ways of including trust in Recommender System is achieved by two ways: first is explicit trust (values specified by users) and second is the trust value calculated implicitly or called Implicit Trust[7][9]. Explicit trust means that the trust information is explicitly provided by users. However, several points have been taken into consideration for explicit trust. One of the issue is trust values can be specific in many system and second issue is that trust values can generate inaccurate results[11][13] For example two friends having good trust values can have different taste for a particular movie.

Similarity computation to compute similar users have significant influence on the performance of Collaborative filtering. It is applied in both memory-based and model-based approaches [14]. The methods adopted for calculating user similarity in Collaborative filtering are Cosine similarity (COS) and Pearson correlation coefficient. Cosine similarity (COS) defines similarity between two users as cosine value of the angle between

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

two rating vectors; Pearson correlation coefficient (PCC) defines user similarity as linear correlation between two rating vectors [2].

2. COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM

Collaborative filtering is a technique to generate recommendation for an active user by aggregating the rating for those users whose past behavior is same as the active user [5]. Collaborative filtering information domain represents users who are responsible to provide preferences to most of the items. There are various techniques to generate recommendation which can be summarized in the Figure 2.1. Collaborative filtering techniques can be classified into three categories [1].

I. Baseline prediction methods check the performance of personalized recommendation technique with non-personalized technique (Baseline).
II. User based Collaborative filtering generate the recommendation for an active user by finding the similar users whose past behavior is same as the active user.
III. Item based Collaborative filtering generate the recommendation on the basis of item similarity.

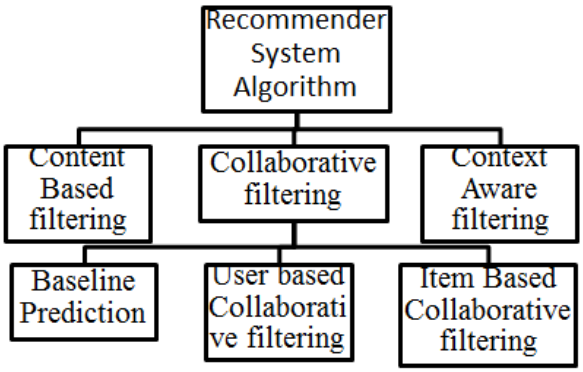

Figure 1: Classification of Recommender System techniques [1].

Rating information is represented using rating matrix. The rating preference given by a user is expressed in the form of triplet given by three parameters which are user, item and rating. Some system uses integer or real values of rating given on a scale of [0, 5] e.g. Film Trust while some uses binary (likes or dislikes) scale [5] . The rating triple forms a rating matrix. Table 2.1 shows a rating matrix of three users and three movies in a movie recommender system. There are some missing values which indicate that the user has not given any rating to that movie. Collaborative filtering focuses on two tasks. One is calculating rating prediction it means that what can be the user's preference for an item which is equivalent to finding missing values in a matrix.

Table 1: User-Item Rating matrix [5]

User\Movie	Harry Potter	The Conjuring 3	The Decent
User P	3	?	4
User Q	?	2	3
User R	4	5	?

Second task is for a particular user; generate the best ranked list of items. A recommended list is not fixed to generate items based on the highest predicted values because predicted preference is not the only criteria to generate the recommendation list.

To represent various elements of a recommender system, let U consist a collection of users, collection of items are represented by I. U_i represents all those users who have provided rating for an item i. similarly I_u depicts a set of all those items for which user u have given a rating. Further, The rating matrix R denotes tuple in the form of $r_{u,i}$ which indicates rating value. The rating vector for all items rated by user u is r_u and r_i is the rating vector for all ratings enlisted for an item i. r_u' with r_i' be the mean of user's rating and item's rating respectively. The prediction by a recommender system is given by $p_{u,i}$ [1] [5]. There are some other methods for generating recommendation, such as content-based filtering, context aware recommendation [18]. Content based filtering finds the item which is similar to an item using textual similarity, it uses the log history to generate recommendation. Context aware recommendation uses context information e.g time information which is available in most of the data set [19].

2.1 BASELINE PREDICTION

These methods denote some non-personalized methods against which personalized methods or algorithms can be

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

evaluated [1]. In addition to it all baseline strategies which do not depend on user's rating can be used to provide recommendation for new users. Simplest baseline to predict rating can be μ (where μ is the average rating).This concept is enhanced by predicting the average rating in terms of user's average rating or item's average rating. Let $b_{u,\,i}$ denotes baseline value of a user u for an item i then it can be expressed as $b_{u,\,i}=r_u'$ or $b_{u,\,i}=r_i'$. In Exhibit [1] it is found that and Baseline can be expressed using the following equation also.

Baseline Predictor can be given as ($b_{u\,i}$) =Average overall rating + User baseline predictor (b_u) + Item baseline predictor (b_i). (2.1)

User baseline predictor (b_u) = (rating value by a user for a particular item- average overall rating)/Total number of items for which rating is provided by user u. (2.2)

Similarly an Item baseline predictor (b_i) = (user's rating for a particular item-user baseline predictor-average overall rating)/no. of users rated an item i. (2.3)

Advantage of baseline is it can capture user bias, item bias. Disadvantage of baseline is coverage is low as soon as the size of data set starts increasing baseline methods can be inferior to generate predictions.

2.2. USER BASED COLLABORATIVE FILTERING

User based Collaborative filtering was the first technique to automate Collaborative filtering. It was introduced in an article recommender called Group Lens [1] [2] . It is based on the principle that identify those users whose past behavior is similar to the current user and use their ratings on other items to predict the rating preference for an active user. The rating of these users is weighted by an agreement with active user's rating to predict his rating. In addition to rating matrix user-based Collaborative filtering requires similarity function which calculates the similarity between two users.

Prediction Computation

To generate predictions for an active user let's say for user u, user-based Collaborative filtering first computes similar users corresponding to an active user. Once this is computed, system combines rating of these users to generate recommendation for an active user for a particular item. The Rating prediction is given by using the following formula [1].

$$p(A,i) = r'(A) + \frac{\sum_{B\epsilon N} s(A,B)(r(B,i) - r'(B))}{\sum_{B\epsilon N} |s(A,B)|} \qquad \textbf{(1)}$$

Where p(A, i) is the predicted value of a rating. N denotes the neighborhood set corresponding to user A, s(A, B) is the similarity function between user A and B. r'(B) is the average rating of a user B. By subtracting user's mean tells that there will not be dramatic change in using the rating scale because some user may tend to give higher rating than others.

Exhibit [5] used an unweighted average on ratings with similar users. Here an important question comes that how many neighbors can be selected because It is found that limiting the size of neighbors can improve the performance of rating prediction [10].

Computing User-User Similarity

Several similarity functions have been proposed to calculate user-user similarity.

Pearson correlation coefficient finds similarity between two users using correlation between two users' rating on a common rated item. Advantage of Pearson correlation coefficient is that correlation between two users can be found using statistical approach. The drawback of this method is, it computes high similarity between two users with number of ratings between them is few [3].

Cosine similarity is a vector approach. According to this method users are represented by using a vector of length |I|, where I is the number of items rated by a user. Then similarity of two users is given by taking cosine distance between two vectors. It can be represented using the following formula [1].

$$S(A,B) = \frac{\sum_i r_1(A,i) r_2(B,i)}{\sqrt{\sum_i r_1{}^2}\,(A,i)\sqrt{\sum_i r_2{}^2}\,(B,i)} \qquad \textbf{(2)}$$

Where S(A, B) represents similarity between two user, r_1(A, i) denotes rating value given by a user A for i^{th} item while r_2(B, i) depicts rating value provided by a user B for i^{th} item.

Advantage of Cosine similarity is that it is a vector approach users are represented as a |I| dimensional vector. Two users are considered more similar if their rating distances are minimum. Drawback of this method is that in case of few ratings Cosine similarity generate high similarity.

2.3. ITEM BASED COLLABORATIVE FILTERING

Item-based Collaborative filtering uses similarities between items by considering their rating patterns. According to it two items are said to be similar if same

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

users like or dislike them [7]. In this type of Collaborative filtering recommendation is generated by selecting candidate items having good number of predictions.

3. MERGING TRUST IN COLLABORATIVE FILTERING

Including trust value in recommender systems gives a direction to provide users with recommendation which is based on past behavior and social trust values. Some system uses deep learning to initialize trust network [24]. It is noted that people get influenced easily by what their friends recommend. Merging trust remove two drawbacks of Collaborative filtering [5] [9] [10].

a. Rating matrix are sparse means that very few ratings are available. Also data sparsity means that because the rating matrix are sparse in nature so there is a problem in finding similar users whose past behavior is same as an active user.
b. Cold start deals with a problem of generating accurate recommendation to those users who are inactive in system or those users who generally rate less than 4 or 5 items. In table 2.2 user Y is considered cold user.

Table 2: Rating matrix to show cold user

User\Item	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User X	3	?	4	5	3	2
User Y	?	2	3	?	?	?
User Z	4	5	?	3	2	1

Classification of trust in recommender system is achieved by two ways: first is explicit trust (values specified by users) and second is the trust value calculated implicitly or called implicit trust. Explicit trust means that the trust information is explicitly provided by users [5] [10]. However, several points have been taken into consideration for explicit trust. One of the issue is that trust values can be specific in many system e.g. Film Trust have binary trust information and second issue is that trust values can generate inaccurate results e.g. two friends having good trust values can have different taste for a particular movie.

On the other side, trust values which are implicitly calculated by using different matrix suffer from various drawbacks [5] [7][9]. Implicit nature of trust is interpreted by past behavior of rating or click-through analysis. The matrix which have proposed does not show asymmetric nature of trust because these matrix are based on similarity measures. These matrix are calculated based on the assumption that the two users are considered trustworthy if their ratings are similar or close to similar.
Exhibit [8] stress that a user is considered more trust worthy if he/she provides even opposite rating than the one who is not interested to share opinions. But none of these trust matrix depicts dynamism (trust values may change over time) and context dependency (trust values may change from context to context) properties of trust so explicit trust is weighted higher than implicit trust. Properties of trust can be described as follows [9] [10] [17].

I. Asymmetry refers that two users may express different opinions. It can be possible because trust vales can change from one context to other. It means that if user A trust to his friend B with some value then it is not necessary that B express the same trust value for A.

II. Transitivity refers that if user A trust to his friend B with some value and B trust to his another friend C with some value. Then it can be inferred that C is trustworthy to A for some extent. This property is very useful while extending trust network.

III. Dynamicity refers that trust values are changed over time. It means that time is an important factor to change the trust value. Some data sets include time as an attribute e.g. FilmTrust dataset where information is represented in quadruple user, item, rating, timestamp information [19].

IV. Context Dependence says that trust values are context dependent. It means that trust can change from context to context. A friend can be trustworthy in one context while not in others e.g. friend can give good recommendation for movie but not for book recommendation.

4. PROPOSED METHODOLOGY TO MEASURE TEMPORAL CONTEXT

Trust based Model represented in the related work does not consist the effect of temporal context information (user's preferences may change over time).Rating prediction can be enhanced by including this additional

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Temporal Context Information factor Ck can be added by modifying the rating prediction for function f(i,j) as follows:

$$f(i,j,C_k) =< U_i, M_j, C_k > \quad (3)$$

Similarly Trust Prediction at the context of Ck is given as:

$$T'(r,e,C_k) =< T_r, T_e, C_K > \quad (4)$$

5. RESULTS

To evaluate the performance of trust based algorithm on FilmTrust dataset using our proposed approach, First we normalize the data of sparse matrix by including 0 for each blank entry in the matrix. After that we reduce the dimensionality of matrix using SVD to compute it in an effective manner. We divide the whole data set into 5 sets. While in each iteration, four sets are used as training sets while the last set is used as a test set. This scheme is called cross validation scheme. MAE (Mean absolute error) is taken as evaluation criteria to check the performance on real world data set.

Mean Absolute Error (MAE) calculate the degree to which a rating prediction is close to the ground truth. It can be calculated using the following formula.

$$MAE = \frac{\sum_u \sum_i |P_{u,i} - B_{u,i}|}{N} \quad (5)$$

Where P_u ,_i is the predicted value of rating and B_u ,_i is the ground truth value. N represents number of Test set. MAE is calculated in the same scale as of the ratings in a particular dataset. Another Evaluation Measure which we have taken into consideration is Root Mean Square Error (RMSE). It emphasis on large errors. It is computed by using the following formula.

$$\sqrt{\frac{1}{n}\sum_{u,i}(p_{u,i} - B_{u,i})^2} \quad (6)$$

The performance of trust based algorithm is computed on FilmTrust using our proposed approach.

TABLE 3: EFFECT OF TEMPORAL CONTEXTUAL INFORMATION IN TERMS OF MAE & RMSE

Data Set	Evaluation Criteria	Size	Rating Prediction (Without Temporal Context)	Rating Prediction (Including Temporal Context)
FilmTrust	MAE	10	0.630	0.612
		20	0.629	0.611
		30	0.628	0.611
		40	0.628	0.610
		50	0.628	0.610
		60	0.628	0.610
	RMSE	10	0.890	0.794
		20	0.882	0.786
		30	0.880	0.785
		40	0.880	0.785
		50	0.880	0.785
		60	0.880	0.785

6. CONCLUSION

The Proposed work describes the effect of dynamic preferences of user in rating prediction. The Accuracy in rating prediction is described in terms of Mean absolute error (MAE) and Root mean square error (RMSE). It is noticed that incorporating temporal information reduces the prediction error. The proposed work represents an approach to consider an effect of temporal contextual information in trust based recommender systems. Trust based algorithms, removes the inherent issues e.g. Data sparsity and Cold start problem when predicting ratings of unknown items

7. REFERENCES

[1] G. Guo, "A novel recommendation model regularized with user trust and item ratings," IEEE Transaction on Knowledge and Data Engineering, 2016.

[2] M. Ekstrand, M. Konsten, "Collaborative filtering recommender systems," in Foundations and Trends® in human-computer interaction, vol. 4, pp. 81—173, 2011.

[3] G. Guo., J. Zhang. "A novel Bayesian similarity measure for recommender system," in proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp.1264—1269, 2013.

[4] B. Yang, Y. Lei, D. Liu, "Social collaborative filtering by trust," in proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 2747—2753, 2013.

[5] G. Guo, "Merging trust in recommender systems to alleviate data sparsity and cold start for recommender systems," in proceedings of the 7th ACM Conference on Recommender Systems (RecSys), 2015.

[6] M. Jamali, M. Easter, "A Matrix factorization technique with trust propagation in recommender system for social networks," IEEE Transaction on Knowledge and data Engineering, 2013.

[7] Y. Korean, "Factor in the neighbors: Scalable and accurate collaborative filtering," ACM Transactions on Knowledge Discovery from Data (TKDD), 2010.

[8] C.S. Hwang, Y.P Chen, "Using trust in collaborative filtering recommendation," in New Trends in Applied Artificial Intelligence, pp. 1052—1060, 2007.

[9] J. O'Donovan, B. Smyth, "Trust in recommender systems," in: Proceedings of the 10th International Conference on Intelligent User Interfaces, pp.167–174, 2005.

[10] I. Konstas, V. Stathopoulos, J. Jose, "On social networks and collaborative recommendation," in proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 195—-202, 2009.

[11] J. Audun, W. Quattrochicchi, "Taste and trust," in Trust Management V, pp. 312—322, 2011.

[12] B. Knijnenburg, J. O'Donovan, S. Bostandjiev, A. Kobsa, "Inspectability and control in social recommenders," in proceedings of the 6th ACM Conference on Recommender Systems, pp. 43—50 2012.

[13] H. Ma, D. Zhou, C. Liu, " Recommender systems with social regularization," in proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM'11), pp. 287—296, 2011.

[14] G. Adomavicius, A. Tuzhilin, "Toward the next generation of recommender systems, A survey of the state-of-the-art and possible extensions," in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734—749, 2005.

[15] P. Avesani, P. Massa, R. Tiella, " A Trust-enhanced recommender system Application," in ACM SAC'05, pp. 1589—-1593, 2005.

[16] G. Adomavicius, A. Tuzhilin, "Context-Aware Recommender Systems," in Recommender Systems Handbook, pp. 217—256, 2011.

[17] S. Deng, X. Wu, "On deep learning for trust aware recommendations," in social networks IEEE transactions on Neural Networks and Learning Systems, vol. 28, pp. 1164—1177, 2016.

[18] B. Li, X. Zhu, "Cross-domain Collaborative filtering over time," in proceedings of 22nd International Joint Conference on Artificial Intelligence, vol. 3, pp.2293—2298, 2011.

[19] S. Dooms, L. Martens, "MovieTweetings – A movie Rating dataset collected from Twitter," in CrowdRec, 55th International Symposium, pp. 49--54, IEEE, 2013.