International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

An effect of temporal information for Trust aware Recommender System

Ankur Chaturvedi, Aprna Tripathi, Rahul Pradhan, Dilip Kumar Sharma

Abstract*: Recommender systems are commonly used by many platforms online from movie renting website to movie streaming sites, from grocery store online portal to Amazon. It makes user to choose better and easily among the wide variety of products. Personalized recommendations are most effective, Collaborative filtering is best known for this. This technique aggregates the liking and ratings of various users and prepare recommendations. Similarity have a greater impact because it act as a criterion to Identify a group of similar users whose ratings will be merged to generate recommendation for 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 MovieTweetings dataset which include temporal context information.

***Index Terms*: Collaborative filtering, Recommender System.**

I. 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 For example 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 metrics suffer from various drawbacks. Implicit nature of trust is interpreted by past behavior of rating. The metrics which have proposed do not show asymmetric nature of trust because these metrics are based on similarity measures. These metrics are calculated based on the assumption that the two users are considered trustworthy if their ratings are similar or close to similar. so it is better to consider explicit trust in recommendation[11][13]. Ratings and trust values are dynamic in nature it can change with respect to time so it is important to consider the temporal information along with rating and trust information. 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 two rating vectors; Pearson correlation coefficient (PCC) defines user similarity as linear correlation between two rating vectors [2].

II. RELATED WORK

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][15].

Revised Manuscript Received on 30 May 2019.
* Correspondence Author
Aprna Tripathi*, Department of Computer Engineering and Applications, GLA University, Mathura, India.
Rahul Pradhan, Department of Computer Engineering and Applications, GLA University, Mathura, India.
Ankur Chaturvedi, Department of Computer Engineering and Applications, GLA University, Mathura, India.
Dilip Kumar Sharma, Department of Computer Engineering and Applications, GLA University, Mathura, India.

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 classified into three categories [1][18].

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.

2.1 Baseline prediction methods

These methods denote some non-personalized methods against which personalized methods or algorithms can be 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. It is found in [2] that and Baseline can be expressed using the following equation.User baseline predictor value (UBP), which indicates the base value (rating prediction cannot be less than this value) of corresponding user in rating prediction is represented by the equation (1).

$$UBP = \frac{\sum_{i\epsilon I(a)} r(a,i) - \mu}{|I(a)|} \quad \textbf{(1)}$$

Where r(a, i) is the rating prediction of user a for an ith item. I(a) is the set of all items rated by user a and μ represents mean average rating for a system.Similarly Item baseline predictor (IBP), which indicates the effect of item popularity is represented by the equation (2).

$$IBP = \frac{\sum_{a\epsilon A(i)} r(a,i) - UBP - \mu}{|A(i)|} \quad \textbf{(2)}$$

Where $A(i)$ is the set of users who have rated ith item.So the Baseline predictor value for a user a for an ith item is represented by using equation (3).

$$BP(a,i) = \mu + UBP + IBP \quad \textbf{(3)}$$

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.

To generate predictions for an active user let's say for user a, user-user Collaborative filtering first computes similar users corresponding to an active user using various similarity measures (PCC, COS). Once this is computed, system combines rating of these users to generate recommendation for an active user for a particular item. The rating prediction [3] [4] is given by using equation (1).

$$P(a,i) = r'(a) + k\sum_{b=1}^{n} w(a,b)(r(b,i) - r'(b)) \quad \textbf{(4)}$$

Where p (a, i) is the predicted value of a rating for user a. b denotes all the users who have provided rating for an ith item , w(a, b) is the weight function. It denotes correlation, similarity between user a and b. r'(b) denotes average ratings provided by user b. Pearson correlation coefficient was considered as a weighting scheme in Group Lens project [1] [2].The Pearson correlation coefficient (PCC) between two users m and n is given by using equation (5).

$$\boldsymbol{w(m,n)} = \frac{\sum_j \left(\boldsymbol{r(m,j) - r'(m)}\right)\left(\boldsymbol{r(n,j) - r'(n)}\right)}{\sqrt{\sum_j (\boldsymbol{r(m,j) - r'(m)})^2 \sum_j (\boldsymbol{r(n,j) - r'(n`)})^2}} \quad \textbf{(5)}$$

Here j denotes the set of items for which both users m and n have provided ratings. It has been investigated that Pearson correlation coefficient (PCC) suffer from various drawbacks [3] e.g. In constant rating problem: if all rating values in user-item rating matrix are constant, e.g., (2, 2, 2), PCC is not computable because denominator part becomes 0. Guo.et.al. [3] proposed a probabilistic similarity measureby looking into direction (rating distances) and length (rating amount) of rating vectors. Rating Distances are represented using Dirichlet distribution, which is calculated on the basis of observed evidences, each of which is represented as a pair of ratings (from two rating vectors) towards commonly rated items. Then the user similarity is calculated as a weighted average of rating distances, respective of the amount of new evidences falling in the distance. An advantage of this approach is one can infer in the same manner from a small sample size as from a large sample size.

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

2.4 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*Data sparsity* and *Cold start* of Collaborative filtering [5],[9],[10].Properties of trust can be described as follows[16],[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. MovieTweetings dataset where information is represented in quadruple user, item, rating, timestamp information [19].
IV. ***ContextDependence*** 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.

III. PROPOSED METHODOLOGY

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 information .Temporal Context Information factor C_k can be added by modifying the above rating prediction for function f(i, j) as follows.

$$f(i,j,C_k) =< U_i, M_j, C_k > \quad \textbf{(6)}$$

Where $f(i,j,C_k)$ represents rating prediction of a user i for j item during timestamp C_k. $U_i$represents user feature matrix, $M_j$represents item feature matrix.Similarly Trust Prediction at the context of C_k is given as using equation (12).

$$T'(r,e,C_k) =< T_r, T_e, C_K > \quad \textbf{(7)}$$

Where $T'(r,e,C_k)$ denotes trust prediction of truster r for a trustee e at timestamp C_k. Where T_r and T_e represents Truster feature matrix and Trustee feature matrix respectively. Using Factorization Approach, it can be represented as.

$$f(i,j,C_k) =< U_i, M_j >< M_j, C_k >< U_i, C_k > \quad \textbf{(8)}$$

$$T'(r,e,C_k) =< T_r, T_e >< T_e, C_k >< T_r, C_k > \quad \textbf{(9)}$$

Loss function in rating matrix L_r and trust matrix L_t is modified as.

$$L_t = \frac{1}{2}\sum_u \sum_{v\epsilon T_u}(w_v{}^T p_u - t_{u,v})^2 + \frac{1}{2}\sum_v \sum_{n\epsilon C_v}(s_n{}^T w_v - t_{v,n})^2 + \frac{1}{2}\sum_u \sum_{o\epsilon C_u}(S_o{}^T p_u - t_{o,u})^2 \quad \textbf{(11)}$$

$s_l{}^T q_j$ represents j^{th} item rating at timestamp s_l. $r_{j,l}$ represents ground truth value of rating, similarly $s_m{}^T p_u$ represents prediction of rating of a user p_u at timestamp s_m.
Total Loss function (L) using equation (15) and (16) is represented as.

$$L=\text{Min}\{L_r+L_t\} \quad \textbf{(12)}$$

Learning of Model

The proposed model describes the objective function in terms of loss which has to be minimized considering the effect of temporal information. Rating and Trust information can be changed with respect to time. Including timestamp information along with rating and trust increase the dimension of the data which is difficult to process in matrix form. Dimensionality reduction approach is used to process the information in 2D form. When one coordinate is not considered then loss of information is generated. Loss function L is represented as a sum of loss occurred in rating matrix as well as loss occurred in trust matrix. Learning of a model on the training dataset in the proposed methodology can be done using following steps. Timestamp feature vector is also updated in the proposed Model.

I. **Input** : Rating Matrix (R), Trust Matrix (T), Dimension of feature vectors (d), λ is Learning rate.
Output: Prediction of Rating.
II. Initialize user matrix and item matrix with some values (0,1).

III. While Loss function L is not converged do
(a) Update user feature vector using gradient descent as

$$p_u \leftarrow p_u - \lambda\frac{\partial L}{\partial p_u} \qquad u=1.....m \text{ (No. of users)} \quad \textbf{(13)}$$

(b) Update item feature vector as

$$q_j \leftarrow q_j - \lambda\frac{\partial L}{\partial q_j} \qquad j=1.....n \text{ (No. of items)} \quad \textbf{(14)}$$

(c) Update truster feature vector as

$$w_v \leftarrow w_v - \lambda\frac{\partial L}{\partial w_v} \qquad v=1.....m \text{ (No. of users)} \quad \textbf{(15)}$$

(d) Update timestamp feature vector as

$$s_l \leftarrow s_l - \lambda \frac{\partial L}{\partial s_l} \qquad l=1.....m(\text{No. of timestamp}) \qquad (16)$$

Similarly update s_m, s_n, s_o feature vector.

IV. Return user feature vector, item feature vector.

This approach is continued until loss gets stable in terms of minimization.

IV. Proposed Framework

In this section a framework which calculates the performance of trust based algorithm considering temporal context information is proposed. User similarity is computed using Pearson correlation coefficient (PCC). The Proposed framework consist various sections e.g., Data file format, Data Convertor, Data splitter, recommender Algorithms, Data filter and evaluator. Fig.1. depicts the proposed framework.

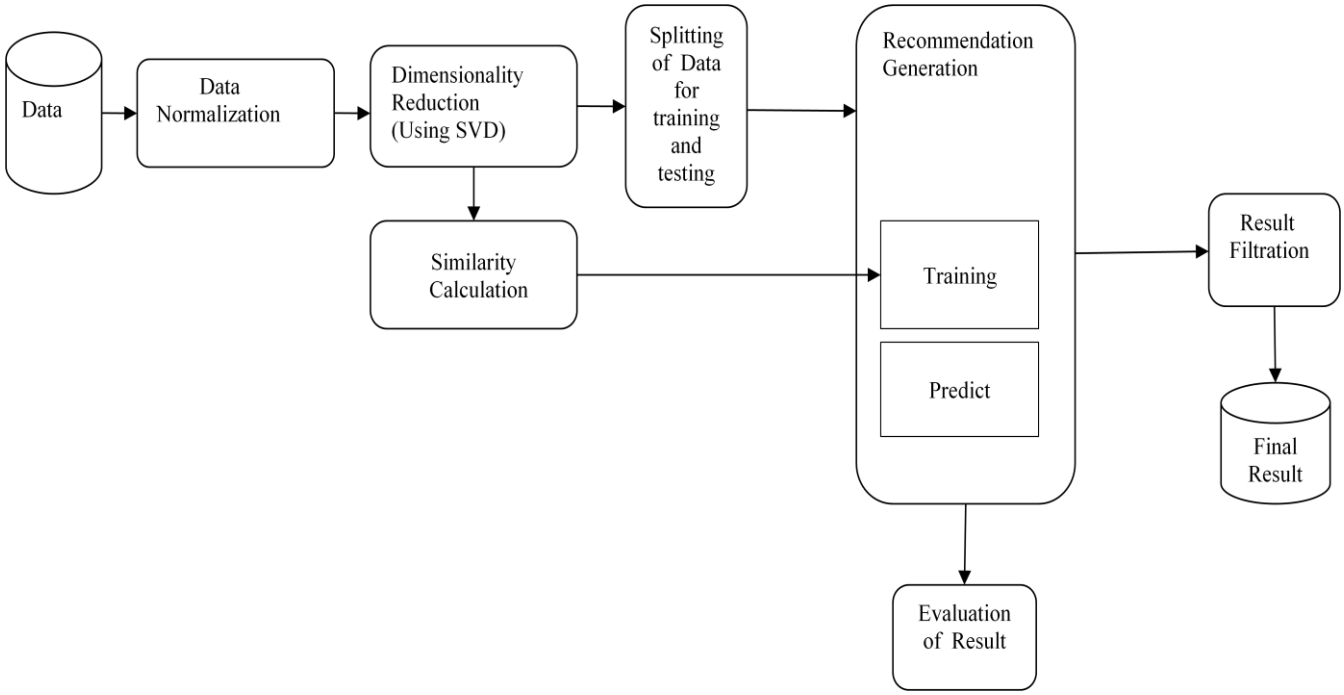

Fig. 1.Proposed Framework to compute rating prediction in Recommender System.

Various sections of proposed framework are described below.

Data Format- A standard publically available dataset MovieTweetings [19] is related to movies rating given by users along with timestamp information. Text data of a dataset is taken into consideration to implement the work, where the data is stored in four columns. Every row in rating matrix is a user-item-rating-timestamp information. The columns are separated by spaces in these datasets.

Table 1. Format of MovieTweetings[19] data set.

User	Item	Rating	Timestamp
1010	210	2	1387645427139852
1010	215	3	2987632421298713
1010	220	2.5	6543289763213871

Ratings are represented on a scale of [1, 10] with step 1.This dataset consist two files rating.txt and trust.txt. rating.txt consist 65,115 records. These records have four attributes-user-id, movie-id and movie-Rating and timestamp information.

Data Normalization- Data normalization means that baseline value corresponding to user and item is subtracted from rating matrix. These user baseline and item baseline gets updated in learning of a recommender system. The overall baseline of rating matrix can be calculated by considering the effect of mean rating, user bias and item bias.

Dimensionality Reduction-Singular value decomposition (SVD) approach is used to reduce the dimension of user and item specific latent feature vector. SVD is used on normalized rating matrix for better learning of model. An advantage of dimensionality reduction is that several similar group of users can be merged in user vector space to reduce the dimension of user feature vector.

Splitting of data for training and testing-Data can be split into training data, test data in a certain ratio. Data can be split on the basis of user, item, rating. There are various options available to split the data. loocv picks up randomly one user or item as train data and the rest as test data. givenn selects N items or users as test data and the rest as train data. The path of test data should be in the same directory as of train data so that when reading all the data, test data can also be read.

Similarity Computation-User similarity is calculated using Pearson correlation coefficient (PCC). Pearson correlation coefficient finds similarity using correlation between two users' rating on a common rated item.

Recommender-The task of Recommender can be carried out by calculating similarity between user to user or item to item using similarity matrixes, which consist the distance between users or items in the data set.

In our work we have used similarity class which represents distance function as a function of distance of a specified user pair.

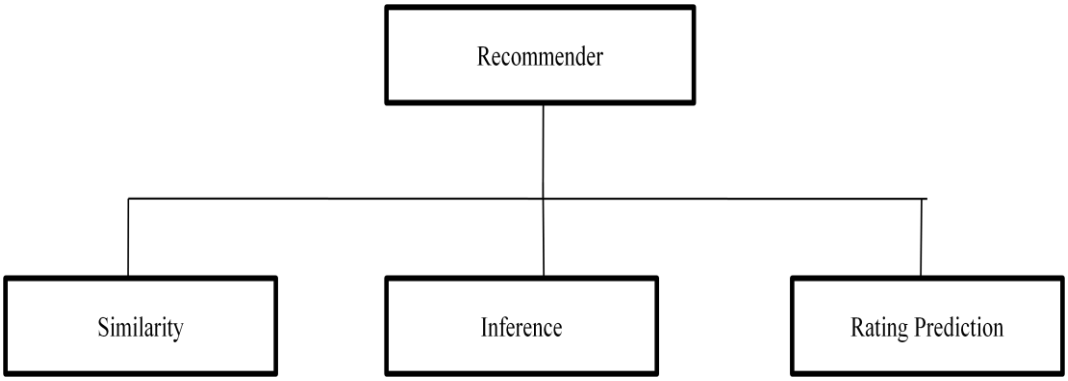

Fig. 2.Task of a Recommender

Evaluation of result- To check the performance of rating prediction various evaluation matrix have been proposed e.g. Mean absolute error (MAE) and Root mean square error (RMSE)[20][21].

Result filtration-Filter is used to filter the data, for this purpose filter class is used which is available in java. The purpose of this filter is used to perform filtering operations on the result provided by recommender. The recommended list is represented consist three attributes (user-id, item-id, rating value). A specific user-id or item-id can be given as an input to filter class to carry out filtering operation for a particular user or for a particular item.

V. EXPERIMENTAL RESULTS

In this section experiments performed using the proposed framework on MovieTweetings data set is discussed. Furthermore, the results of different iterations considering temporal information are calculated and are compared with existing Trust based approaches [1].

A. *Dataset*

MovieTweetings data set is used to consider an effect of temporal contextual information [19]. This data set is publicly available data set which consist both item ratings and trust values specified by users along with timestamp value. It extracts rating information from latest tweets available by the user using tweeter search API. Table 2. shows the features of MovieTweetings data set.

Table 2. Features of MovieTweetings[19] data set

Feature	MovieTweetings
Users	12420
Items	8468
Ratings	65115
Trusters	6235
Trustees	5890
Trust	4678

This data sets consist movie rating data. In MovieTweetings rating data is represented on a scale of [1, 10] with step 1. Trust values are binary in nature in this data set.

B. *Experimental Setup*

To evaluate the performance of trust based algorithm on MovieTweetings dataset using proposed approach, First Data normalization is done on the sparse matrix. After that dimensionality reduction of matrix using singular value decomposition (SVD) is performed to process the vectors in an effective manner. User similarity is calculated by applying Pearson correlation coefficient (PCC) measure on this matrix. 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.

C. *Results and Discussion*

The performance of the trust based algorithm on MovieTweetings is verified on different parameters to evaluate accuracy and the results are compared with the some of the previous approaches [1] which do not take temporal information into consideration.

D. *Quantitative Analysis*

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 [1] [20] [21].

$$MAE = \frac{\sum_u \sum_i |P_{u,i} - B_{u,i}|}{N} \quad \textbf{(17)}$$

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. A temporal version of Mean absolute error is calculated using the following formula [20] [21].

$$TA\text{-}MAE = \frac{1}{n_t} \sum_{p_{u,i}:t_{u,i} \le t} (P_{u,i} - B_{u,i}) \quad \textbf{(18)}$$

Where n_t is the number of ratings given up to time t and $t_{u,i}$is the time of ground truth value of rating which is dented as$B_{u,i}$. 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 [1] [20] [21].

$$RMSE=\sqrt{\frac{1}{n}\sum_{u,i}(p_{u,i}-B_{u,i})^2} \quad \textbf{(19)}$$

A temporal version of Root mean square error (RMSE) gives for all rating prediction generated up to time t. It is represented by the following formula [20] [21].

$$TA\text{-}RMSE=\sqrt{\frac{1}{n_t}\sum_{p_{u,i}:t_{u,i}\le t}(p_{u,i}-B_{u,i})^2} \quad \textbf{(20)}$$

Where n_t is the number of ratings given up to time t and $t_{u,i}$ is the time of rating $B_{u,i}$.

E. *Comparison*

`Two evaluation matrix Mean absolute error (MAE) and Root mean square error (RMSE) are taken into consideration to calculate the accuracy in rating prediction. Quantitative comparison in terms of MAE is shown in Tables 3 and 4.

Table 3. Effect of temporal contextual information in terms of MAE and its time variant in trust based recommender system.

Data Set	Evaluation Criteria	Size	Rating Prediction (Without Temporal Context)	Evaluation Criteria	Rating Prediction (including Temporal Context)
Movie Tweetings	MAE	10	0.624	TA-MAE	0.607
		20	0.622		0.605
		30	0.621		0.604
		40	0.621		0.602
		50	0.621		0.602
		60	0.621		0.602

From Table 3. It is clear that the comparison of trust based system against temporal context in terms of MAE on MovieTweetings dataset [19] achieves good accuracy in rating prediction. Fig. 4. shows the effect of temporal context in trust based system using line graph.

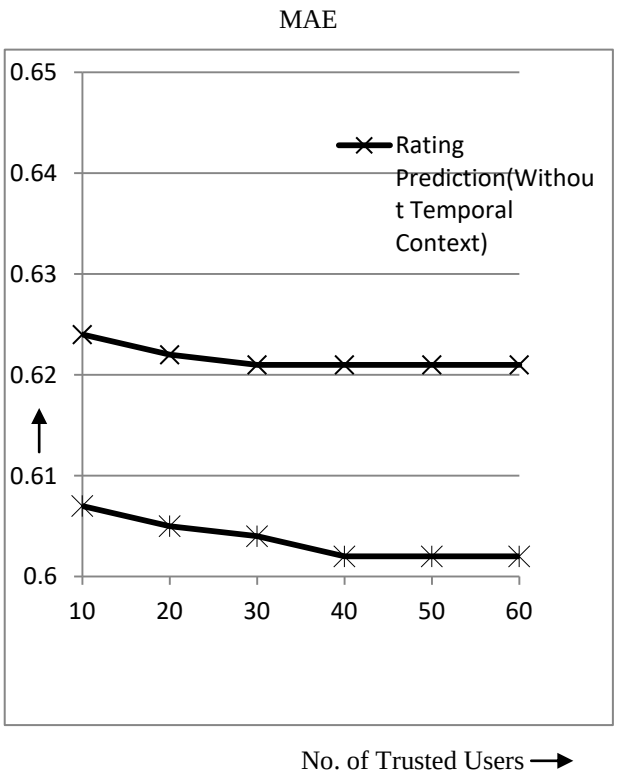

Fig. 3. Effect of temporal context in trust based system in terms of MAE.

Another evaluation criteria Root mean square error (RMSE) is taken to compute the effect of temporal contextual information on trust based recommender system. Table 4. describes this effect.

Table 4. Effect of temporal contextual information in terms of RMSE and its time variant in trust based recommender system.

Data Set	Evaluation Criteria	Size	Rating Prediction (Without Temporal Context)	Evaluation Criteria	Rating Prediction (including Temporal Context)
Movie Tweetings	RMSE	10	0.852	TA-RMSE	0.823
		20	0.847		0.822
		30	0.845		0.820
		40	0.845		0.818
		50	0.845		0.818
		60	0.845		0.818

From table 4. It is clear that the comparison of trust based system against temporal context in terms of RMSE on MovieTweetings dataset [19] achieves good accuracy in rating prediction. Fig. 5. shows the effect of temporal context in trust based system in terms of Root mean absolute error (RMSE).

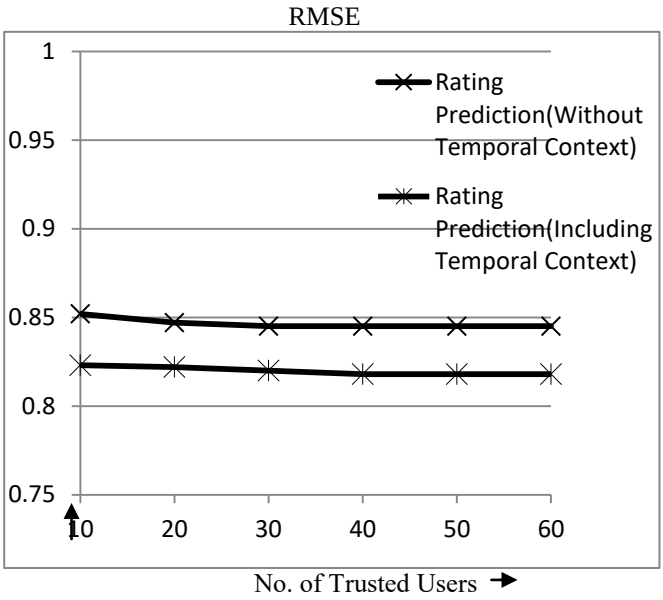

Fig. 4. Effect of temporal context in trust based system in terms of RMSE.

From Fig. 4. and Fig. 5. it is noticed that dimensionality of user feature vectors and item feature vectors are initialized as 10. As soon as more similar trust users are merged up this dimension starts increasing. Value of each evaluation matrix are being measured with respect to this new sample size. An approach which does not take time factor into consideration gets converge after 30 no. of users while taking time into consideration it gets converge after 40 no. of users. It indicates that time to train a model along with temporal contextual information is more as compared to without considering temporal information.

VI. CONCLUSION AND FUTURE DIRECTION

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 sparsityand Cold start problem when predicting ratings of unknown items using similarity measure Pearson correlation coefficient (PCC). 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.Also inclusion of this temporal information increases the training time because of later convergence. For future work some other parameters e.g. sentiments needs to be investigated in addition to time for better prediction of ratings

REFERENCES

1. Guo, G., Zhang, j.: A novel recommendation model regularized with user trust and item ratings. In: IEEE Transaction on Knowledge and Data Engineering, vol. 28, pp. 1607—1620 (2016).
2. 2. Ekstrand, M., Konsten, M.: Collaborative filtering recommender systems. In: Foundations and Trends® in human-computer interaction, vol. 4, pp. 81—173 (2011).
3. 3. Guo, G., Zhang, J.: A novel Bayesian similarity measure for recommender system. In: proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp.1264—1269 (2013).
4. 4. Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), pp. 2747—2753 (2013).
5. 5. Guo, G.: 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), vol. 57, pp. 57—68 (2014).
6. 6. Jamali, M., Easter, M.: A Matrix factorization technique with trust propagation in recommender system for social networks. In: proceedings of the 4th ACM conference on Recommender Systems, pp. 135—142 (2010).
7. 7. Korean, Y.: Factor in the neighbors: Scalable and accurate collaborative filtering. In: ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 4, pp. 258—269 (2010).
8. 8. Hwang, C.S., Chen, Y.P.: Using trust in collaborative filtering recommendation. In: New Trends in Applied Artificial Intelligence, pp. 1052—1060 (2007).
9. 9. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: proceedings of the 10th International Conference on Intelligent User Interfaces, pp.167—174 (2005).
10. 10.Konstas, I., Stathopoulos, V., Jose, J.: 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. 11.Audun, J., Quattrochicchi, W.: Taste and trust. In: Trust Management V, pp. 312—322 (2011).
12. 12.Knijnenburg, B., O'Donovan, J., Bostandjiev, S., Kobsa, A.: Inspectability and control in social recommenders. In proceedings of the 6th ACM Conference on Recommender Systems, pp. 43—50 (2012).
13. 13.Ma, H., Zhou, D., Liu, C.: 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. 14.Adomavicius, G., Tuzhilin, A.: 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. 15.Avesani, P., Massa, P., Tiella, R.: A Trust-enhanced recommender system Application. In: ACM SAC '05, pp. 1589–-1593 ACM (2005).
16. 16.Adomavicius, G., Tuzhilin, A.: Context-Aware Recommender Systems. In: Recommender Systems Handbook, pp. 217—256 (2011).
17. 17.Deng, S., Wu, X.: 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. 18. Li, B., Zhu, X.: Cross-domain Collaborative filtering over time. In: proceedings of 22nd International Joint Conference on Artificial Intelligence, vol. 3, pp.2293-- 2298 (2011).
19. 19.Dooms, S., Martens, L.: Movie Tweetings – A movie Rating dataset collected from Twitter. In: CrowdRec, 55th International Symposium, pp. 49--54, IEEE, (2013).
20. 20.Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. In: Journal of Machine Learning Research, vol. 10, pp. 2935—2962 (2009).
21. 21.Herlocker, J. L., Konstan, A., Terveen, L. G., Riedl, J. T.: Evaluating collaborative filtering recommender systems. In: ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5—53 (2004).

AUTHORS PROFILE

Ankur Chaturvedi He had 8 years of teaching and research experience. He is pursuing PhD from Punjab Technical University, Jalandhar. He is member of CSI. He is M. Tech CSE from GLA University, Mathura.

Aprna Tripathi had 13 years Teaching/Research experience in the field of computer science, She is PhD in 2015 from Motilal Nehru National Institute of Technology, Allahabad, U.P., India and M.Tech. in 2009 from Banasthali University, Rajasthan,India. She had published her work in number of conferences and Journals. She is member of CSI.

Rahul Pradhan had 7 years Teaching/Research experience in the field of computer science. He is currently pursuing PhD in CSE from GLA University. He is member of CSI and IEEE. He had number

of publications in various per reviewed Journals and Conferences.

Dilip Kumar Sharma, He had 13 years of experience in teaching and research. He is Office Bearer- Vice Chairman of IEEE Uttar Pradesh Section since January 2019 to till now. He is Office Bearer- Vice Chairman of IEEE Computer Society Chapter of IEEE Uttar Pradesh Section since January 2017 to till now. He is Office Bearer of Chairman Computer Society of India Mathura Chapter since April 2018 to till now. He is Office Bearer of Secretary IEEE Uttar Pradesh Section since January 2016 to January 28, 2018. He is Convener/Chair of IEEE Special Interest Group on Humanitarian Technology (SIGHT), Uttar Pradesh Section since February 07, 2018 to January 2019. He had 81 publications in Journals and conferences.