ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

A Novel Approach by Collaborative Filtering Based On Similarity among Users

Khushee Singh[1], Saroj Hiranwal[2]

M.Tech Scholar, Department of CSE, Rajasthan Institute of Engineering & Technology, Jaipur , Rajasthan, India[1]

Professor, Department of CSE, Rajasthan Institute of Engineering & Technology, Jaipur , Rajasthan, India[2]

ABSTRACT:Recommendation systems are widely used by e-commerce websites to recommend items to users. Recommender systems apply knowledge discovery techniques to the problem of making personalized recommendations for information, products or services during a live interaction. Item based techniques first analyse the user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users. In this paper a new algorithm for item based recommendation system using collaborative filtering is proposed. The algorithm is used to recommend items to a user. The algorithm considers previous patterns of other users which are having same taste. The system predicts ratings for given items for given users. The algorithm is implemented for a standard data set and results conclude that the proposed algorithm performs better than the other existing algorithms.

KEYWORDS:Recommendation Systems, Web Usage Mining, Collaborative Filtering

I. INTRODUCTION

Recommender System (RS) objective is to give a meaningful suggestion of items or products to the group of users that might interest them. RS is a tool that helps predict what a user likes among the given list of items or products. The amount of information in the world is increasing far more quickly than our ability to process it. All of us have known the feeling of being overwhelmed by the number of new books, journal articles, and conference proceedings coming out each year. Technology has dramatically reduced the barriers to publishing and distributing information. Now it is time to create the technologies that can help us sift through all the available information to find that which is most valuable to us. One of the most promising such technologies is collaborative filtering [1]. Collaborative filtering works by building a database of preferences for items by users. A new user, Neo, is matched against the database to discover neighbours, which are other users who have historically had similar taste to Neo. Items that the neighbors like are then recommended to Neo, as he will probably also like them. Collaborative filtering has been very successful in both research and practice, and in both information filtering applications and E-commerce applications. However, there remain important research questions in overcoming two fundamental challenges for collaborative filtering recommender systems. The first challenge is to improve the scalability of the collaborative filtering algorithms. These algorithms are able to search tens of thousands of potential neighbors in real-time, but the demands of modern systems are to search tens of millions of potential neighbors. Further, existing algorithms have performance problems with individual users for whom the site has large amounts of information. For instance, if a site is using browsing patterns as indications of content preference, it may have thousands of data points for its most frequent visitors. These “long user rows” slow down the number of neighbours that can be searched per second, further reducing scalability. The second challenge is to improve the quality of the recommendations for the users. Users need recommendations they can trust to help them find items they will like. Users will”vote with their feet” by refusing to use recommender systems that are not consistently accurate for them. In some ways these two challenges are in conflict, since the less time an algorithm spends searching for neighbors, the more scalable it will be, and the worse its quality. For this reason, it is important to treat the two challenges simultaneously so the solutions discovered are both useful and practical. In this paper, we address these issues of recommender systems by applying a different approach–item-based algorithms. The bottleneck

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

in conventional collaborative filtering algorithms is the search for neighbors among a large user population of potential neighbors [2]. Item-based algorithms avoid this bottleneck by exploring the relationships between items first, rather than the relationships between users. Recommendations for users are computed by finding items that are similar to other items the user has liked. Because the relationships between items are relatively static, item-based algorithms may be able to provide the same quality as the user-based algorithms with less online computation.

II. RELATED WORK

In this section we briefly present some of the research literature related to collaborative filtering, recommender systems, data mining and personalization. Tapestry [3] is one of the earliest implementations of collaborative filtering-based recommender systems. This system relied on the explicit opinions of people from a close-knit community, such as an office workgroup. However, recommender system for large communities cannot depend on each person knowing the others. Later, several ratings-based automated recommender systems were developed. The Group Lens research system provides a pseudonymous collaborative filtering solution for Use net news and movies. Ringo [4] and Video Recommender are email and web-based systems that generate recommendations on music and movies respectively. A special issue of Communications of the ACM [5] presents a number of different recommender systems. Other technologies have also been applied to recommender systems, including Bayesian networks, clustering, and Horting. Bayesian networks create a model based on a training set with a decision tree at each node and edges representing user information. The model can be built off-line over a matter of hours or days. The resulting model is very small, very fast, and essentially as accurate as nearest neighbor methods [6]. Bayesian networks may prove practical for environments in which knowledge of user preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which user preference models must be updated rapidly or frequently. Clustering techniques work by identifying groups of users who appear to have similar preferences. Once the clusters are created, predictions for an individual can be made by averaging the opinions of the other users in that cluster. Some clustering techniques represent each users with partial participation in several clusters. The prediction is then an average across the clusters, weighted by degree of participation. Clustering techniques usually produce less-personal recommendations than other methods, and in some cases, the clusters have worse accuracy than nearest neighbor algorithms [6]. Once the clustering is complete, however, performance can be very good, since the size of the group that must be analyzed is much smaller. Clustering techniques can also be applied as a ”first step” for shrinking the candidate set in a nearest neighbor algorithm or for distributing nearest-neighbor computation across several recommender engines. While dividing the population into clusters may hurt the accuracy or recommendations to users near the fringes of their assigned cluster, pre-clustering may be a worthwhile trade-off between accuracy and throughput. Horting [7] proposed is a graph-based technique in which nodes are users, and edges between nodes indicate degree of similarity between two users. Predictions are produced by walking the graph to nearby nodes and combining the opinions of the nearby users. Horting differs from nearest neighbor as the graph may be walked through other users who have not rated the item in question, thus exploring transitive relationships that nearest neighbor algorithms do not consider. In one study using synthetic data, Horting produced better predictions than a nearest neighbor algorithm [7]. Schafer et al., [8] present a detailed taxonomy and examples of recommender systems used in E-commerce and how they can provide one-to-one personalization and at the same can capture customer loyalty. Although these systems have been successful in the past, their widespread use has exposed some of their limitations such as the problems of Sparsity in the data set, problems associated with high dimensionality and so on. Sparsity problem in recommender system has been addressed in [9]. The problems associated with high dimensionality in recommender systems have been discussed in [10], and application of dimensionality reduction techniques to address these issues has been investigated in [11].

III. PROPOSED WORK

In this paper a new algorithm for item based recommendation system. The proposed algorithm uses a data set to recommend items for a given user. It works on the following data set. The proposed algorithm, predict ratings for given items for a given set of users.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Data Set - In this paper, Author have evaluated the system with the data from Movie Lens which contains 100,000 rating from 943 users and 1682 movies. The rating is on the scale of 1-5 where 1 means 'poor' and 5 means 'awesome'. Each user in the data set has at least rated 20 movies. It has additional demographic information like age, gender, occupation and zip code etc. In this paper a sample of that data set is considered for the implementation. From the given data set the user-item ratings for first 14 users are considered for implementation.
A sample of user and items is taken as a sample list of items. The proposed system predict the ratings for given items for given users. The proposed algorithm first create a user-user similarity matrix. This matrix is created by using ratings given for same item by different users. The proposed algorithm then create the list of related users for every user. It then uses the average of ratings given by all related users for a given item to predict ratings for given set of user-items.
The process use different algorithms to complete the task. The algorithms works as follows:

Algorithm-1 is used to find user-user similarity matrix. This matrix has number of rows and columns equal to number of user. It stores a similarity value at the cell (i,j) for evry $User_i$ and $User_j$. So it is a symmetric matrix. The process to calculate the similarity matrix is given in Algorithm-1.

Algorithm-1: Algorithm for finding user-user similarity matrix

Input: Ratings for different items given by different users in data set.
Output: user-user similarity matrix

1. Read ratings for different items for different users given in data set.
2. Do steps 2.1 to 2.6 for all users.
   - 2.1 Select a user user-1 from list of users.
   - 2.2 Select another user user-2 from list of users.
   - 2.3 Find absolute difference between ratings for different items by user-1 and user-2 and add these absolute differences.
   - 2.4 Set the sum calculated in step 2.3 as a similarity value between user-1 and user-2 in user-user similarity matrix.
   - 2.5 Check if all users are processed
     thenGoto step-3
     otherwiseGoto step-2.
   - 2.6 Otherwise select next user as user-1 from list of users. And goto step-2.
3. Return user-user similarity matrix.

Algorithm-2 is used to find a list of related users for every user. These algorithms traverse the user-item-ratings given as a input to recommendation system (data-set). The algorithm checks what the ratings given for different items for are given two users. If two users give similar ratings to similar items then these users are related. The process to find list of related users for every user is explained in Algorithm-2.

Algorithm-2 : Finding list of related users for all available users .

Input : Ratings of various items given by various users. Similarity threshold.
Output: List of related users for all available users

1. Find user-user similarity matrix using Algorithm-1.
2. For all users given in list of users select a user-1 and do following steps
   - 2.1 Select next user user-2 from list of users.
   - 2.2 Find the similarity-value between user-1 and user-2 from user -user similarity matrix find in step-2.
   - 2.3 If similarity-value >= similarity-threshold then add this user-2 in list of related users for user-1.
   - 2.4 goto step3 until some users left in list of users.
3. Exit.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Algorithm-3 is used to predict the rating for a given set of user-item pairs. The algorithm entertains the rating given by similar users to the same item. It calculates the average of ratings of all similar users given to that item. It predicts that average value as a rating for given user-item pair. The process is explained in Algorithm-3.

Algorithm-3 : Predicting ratings for a sample of user-item pairs.

Input : A sample of user-item pairs
Output: Predicted ratings for given user-item pairs

1. Read a sample of user-item-pairs
2. Select a user-item pair from the input data
   - 2.1 Set sum=0, set related_user_count=0
   - 2.2 Find *rating* by the given user for given item from ratings table
   - 2.3 Add ratings into sum i.e. sum=sum + *rating*
   - 2.4 Find the list of related users for given user
   - 2.5 Select a related user from list of related users
   - 2.6 Find ratings(related-user-rating) of that user for given item
   - 2.7 Set sum = sum + related-user-rating and related_user_count = related_user_count + 1
   - 2.8 If(no user left in related user list) then goto step 3
     Else Select next related user from list of related users and goto step 2.6.
   - 2.9 Find avg-rating = sum/ related_user_count
   - 2.10 Set sum as predicted-rating for this user-item pair.
   - 2.11 If all user-item pairs from input are processed then Goto step-3.
     Otherwise Select next user-item pair from input data and Goto step 2.1
3. Exit.

IV. RESULTS AND ANALYSIS

Proposed algorithms i.e. Algorithm-1, Algorithm-2 and Algorithm-3 are implemented in JAVA using JDK1.7 and Netbeans IDE 8.0.2. A sample of 10 user-item ratings is taken as a input and system predict the ratings for these user-item pairs using Algorithm-3. These predicted ratings are compared with actual ratings given in data set for same user-item pairs and the value of metric Mean Absolute Error MAE is calculated.

Mean Absolute Error (MAE)

MAE is defined as the average absolute difference between predicted rating and actual ratings.

$$\text{MAE} = \left(\sum_{i=1}^{n} (|pui - rui|)\right)/N$$

Here MAE- Mean Absolute Error
pui – Predicted rating by user u on item i
rui – Actual rating by user u on item i
N – Total number of ratings

Root Mean Square Error (RMSE)

It's the square root of the average of squared differences between prediction and actual observation.

$$\text{RMSE} = \sqrt{\left(\sum_{i=1}^{n} (|pui - rui|)2\right)/N}$$

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Here RMSE is Root Mean Square Error Mean Absolute Error
pui – Predicted rating by user u on item i
rui – Actual rating by user u on item i
N – Total number of ratings

Table-1 is showing the predicted rating and actual rating of 10 sample items. It also calculate the value of MAE. The value of MAE comes out 0.38 which is better than the value of MAE found by [30]. Here columns are showing the value of UserId, MovieId, and ratings given in MovieLens dataset. Column 4 is showing the ratings of the item predicted using proposed algorithm. Column 5 is showing the difference between the given rating and predicted rating.

UserId	MovieId	Given Rating	Predicted Rating	Error
10	1200	4	4	0
10	1197	4	3	1
4	2987	5	5	0
6	2072	4	4	0
2	497	3	3.5	.5
4	1265	5	3.5	1.5
6	7361	4	4	0
4	153	4	4	0
6	7090	3	3	0
8	2762	4.5	3.58	0.92
8	2791	4.5	4.75	0.25
4	1334	5	5	0
11	1201	5	5	0
5	344	3.5	3.25	0.25
8	1754	3	3.0	0
13	2571	3	3.80	0.80
4	1616	1	1	0
5	1923	4.5	4.75	0.25
8	2858	4.5	4.25	0.25
4	364	5	4	1

Table-1 Comparison of Actual rating and predicted rating

Table-2 is showing the difference between the value of MAE and RMSE using proposed algorithm and [30]. It also shows the percentage of improvements in results using proposed algorithm. Column1 is showing the name of the metric. Column 2 is showing the value of quality metrics MAE and RMSE given in reference [12]. Column 3 is showing the value of MAE and RMSE calculated using proposed algorithm. Column 4 is showing the %age improvement in the value of MAE and RMSE using proposed algorithm.

Metric	Value using existing algorithm[12]	Value using proposed algorithm	%age of improvement
MAE	0.55	0.335	64.17%
RMSE	0.69	0.558	23.6%

Table-2 Comparison of value of MAE for proposed algorithm and existing algorithm [12]

Figure-1 is showing a snapshot of actual implementation result for the sample data shown in Table-1. From the Figure 1 it is clear that the value of MAE is 0.335 and value of RMSE is 0.558.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

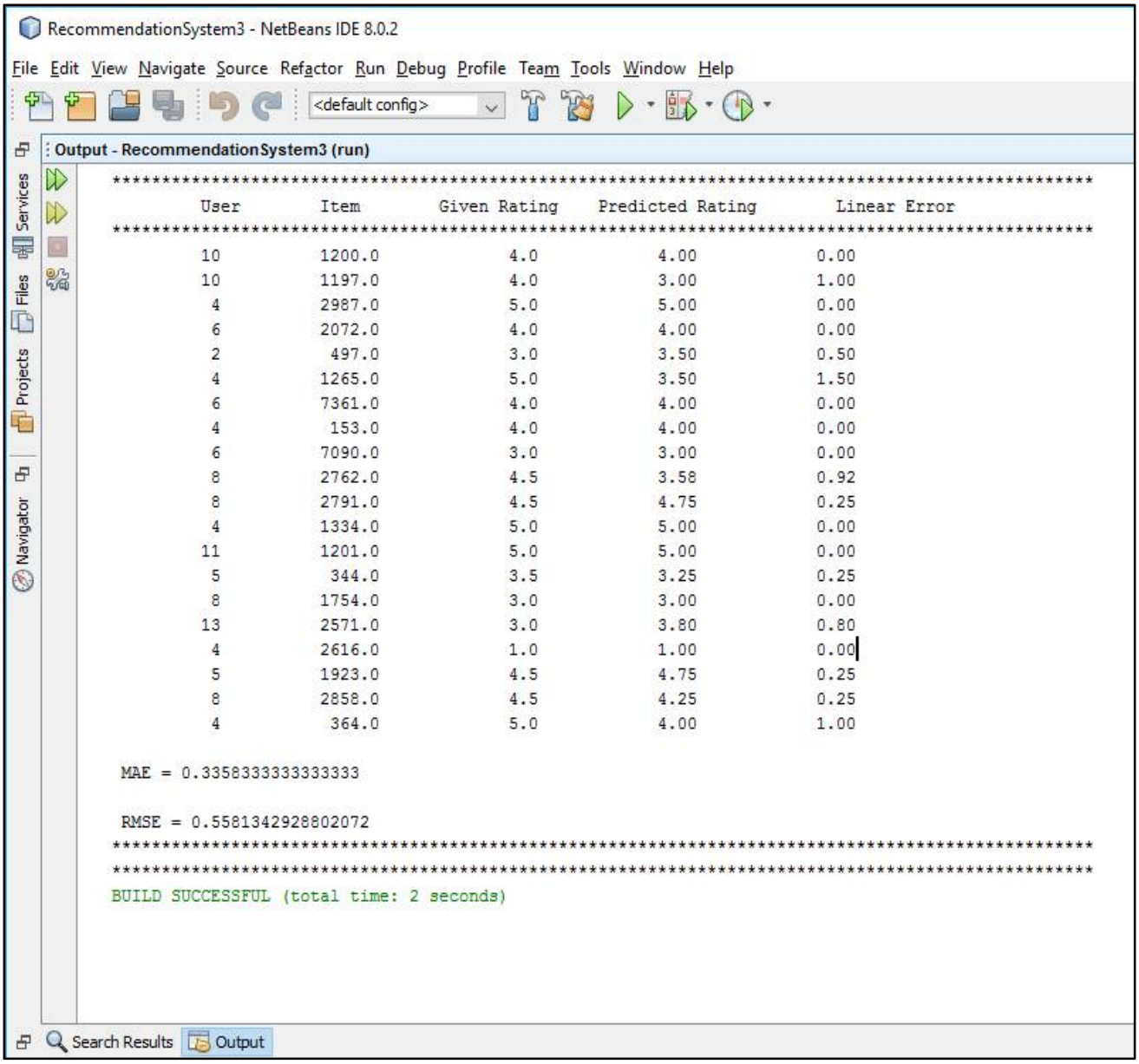

Fig1 A snapshot of actual implementation result for the sample data shown in Table-1.

Figure-2 is showing the graphical comparison between the values of MAE calculated using proposed algorithm and existing algorithm. From the graph chart it can be easily concluded that the value of MAE and RMSE using proposed algorithm is less as compared to existing algorithm.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

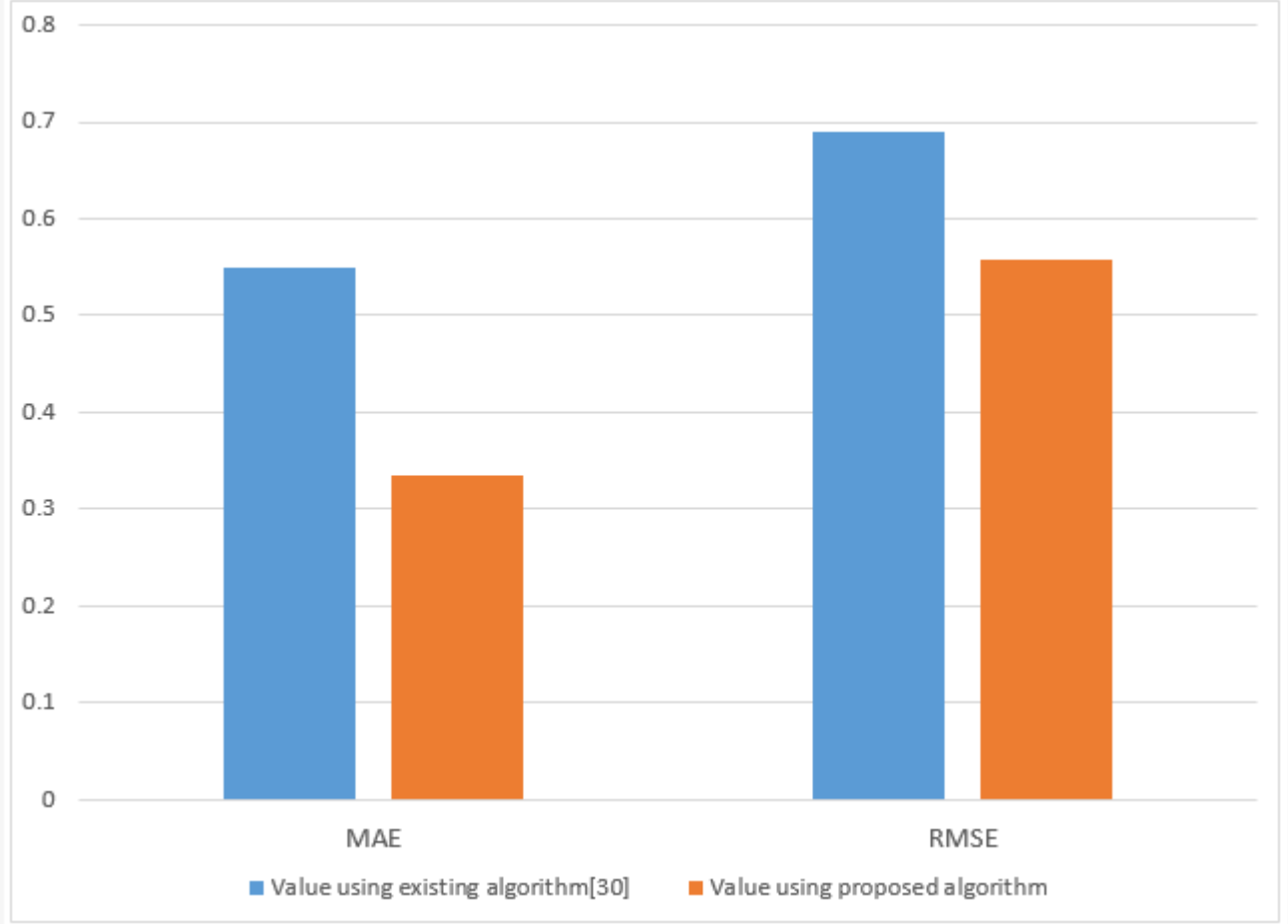

Fig2 Comparison of MAE and RMSE using proposed algorithm and existing algorithm

V. CONCLUSION AND FUTURE WORK

Recommendation systems are widely used by ecommerce companies to recommend items to its users. In this paper a recommendation system using user item similarity is proposed. The proposed system first finds the similarity between the users based on their past history available in data set. It then find the ratings for a sample of user-item pairs and compare the value of actual rating and predicted rating. Mean absolute Error MAE and Root Mean Square Error RMSE is calculated for the out coming results. Experimental results validate that the proposed algorithm perform 64.17% better for the value of MAE and 23.6% better for the value of the metric RMSE as compared to the existing algorithm. However this work also open path for some other researchers working in this area. In future the system can be extended to recommend some items for given user. Also the work can be done to overcome the problem of sparsity and scalability. Further the algorithm can be tested on some other data sets.

REFERENCES

[1] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J.GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, 40(3), pp. 77-87, 1997.
[2] Herlocker, J., Konstan, J., Borchers, A., and Riedl, J. ,An Algorithmic Framework for Performing Collaborative Filtering. In Proceedings of ACM SIGIR'99. ACM press, 1999.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

[3] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D., Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM. December, 1992.
[4] Shardanand, U., and Maes, P., Social Information Filtering: Algorithms for Automating 'Word of Mouth'. In Proceedings of CHI '95. Denver, CO, 1995.
[5] Resnick, P., and Varian, H. R., Recommender Systems. Special issue of Communications of the ACM. 40(3), 1997.
[6] Breese, J. S., Heckerman, D., and Kadie, C., Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43-52, 1998.
[7] Aggarwal, C. C., Wolf, J. L., Wu K., and Yu, P. S., Horting Hatches an Egg: A New Graph-theoretic Approach to Collaborative Filtering. In Proceedings of the ACM KDD'99 Conference. San Diego, CA. pp. 201- 212, 1999.
[8] Schafer, J. B., Konstan, J., and Riedl, J., Recommender Systems in E-Commerce. In Proceedings of ACME-Commerce conference, 1999.
[9] Sarwar, B., M., Konstan, J. A., Borchers, A., Herlocker, J., Miller, B., and Riedl, J., Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In Proceedings of CSCW '98, Seattle, WA, 1998.
[10] Billsus, D., and Pazzani, M. J., Learning Collaborative Information Filters. In Proceedings of ICML '98. pp. 46-53, 1998.
[11] Sarwar, B. M., Karypis, G.,Konstan, J. A., and Riedl, J., Application of Dimensionality Reduction inRecommender System–A Case Study. In ACM WebKDD Workshop, 2000.
[12] Celine Michael Rodrigues, SheetalRathi and Ganesh Patil, "An Efficient System using Item & User-based CF Techniques to improve Recommendation", 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16, 978-1-5090-3257-0/16/$31.00 © IEEE, 2016
[13] Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as Classification: Using Social and Content-based Information in Recommendation. In Recommender System Workshop, pp. 11-15, 1998.
[14] Berry, M. W., Dumais, S. T., and O'Brian, G. W., Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 37(4), pp. 573-595, 1995.
[15] Brachman, R., J., Khabaza, T., Kloesgen, W., Piatetsky-Shapiro, G., and Simoudis, E., Mining Business Databases. Communications of the ACM, 39(11), pp. 42-48, November, 1996.
[16] Cureton, E. E., and D'Agostino, R. B., Factor Analysis: An Applied Approach. Lawrence Erlbaum associates pubs. Hillsdale, NJ., 1983.
[17] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R., Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), pp. 391-407,1990.
[18] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., Eds. Advances in Knowledge Discovery and Data Mining. AAAI press/MIT press, 1996.
[19] Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J., Combining Collaborative Filtering With Personal Agents for Better Recommendations. In Proceedings of the AAAI-'99 conference, pp 439-446, 1999.
[20] Herlocker, J., Understanding and Improving Automated Collaborative Filtering Systems. Ph.D. Thesis,Computer Science Dept., University of Minnesota., 2000.
[21] Hill, W., Stead, L., Rosenstein, M., and Furnas, G., Recommending and Evaluating Choices in a Virtual Community of Use. In Proceedings of CHI 1995.
[22] Karypis, G., Evaluation of Item-Based Top-N Recommendation Algorithms. Technical Report CS-TR-00- 46, Computer Science Dept., University of Minnesota, 2000.
[23] Ling, C. X., and Li C., Data Mining for Direct Marketing: Problems and Solutions. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 73-79, 1998.
[24] Peppers, D., and Rogers, M., The One to One Future : Building Relationships One Customer at a Time. Bantam Doubleday Dell Publishing, 1997.
[25] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of CSCW , Chapel Hill, NC, 1994.
[26] Reichheld, F. R., and Sasser Jr., W., Zero Defections: Quality Comes to Services. Harvard Business School Review, 1990(5): pp. 105-111, 1990.
[27] Reichheld, F. R., Loyalty-Based Management. Harvard Business School Review, 1993(2): pp. 64-73, 1993.
[28] Sarwar, B. M., Karypis, G.,Konstan, J. A., and Riedl, J., Analysis of Recommendation Algorithms forE-Commerce. In Proceedings of the ACM EC'00 Conference. Minneapolis, MN. pp. 158-167, 2000.
[29] Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J., PHOAKS: A System for Sharing Recommendations. Communications of the ACM, 40(3). pp. 59-62, 1997
[30] Ungar, L. H., and Foster, D. P., Clustering Methods for Collaborative Filtering. In Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence, 1998.