International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 01 | Jan-2018 www.irjet.net p-ISSN: 2395-0072

Engendering the reference links for the preference elicitation problem in social networks using recommender systems techniques

GowriThangam J[1], Sankar A[2]

[1]*Research Scholar, Department of Computer Applications, PSG College of Technology, Coimbatore, Tamil Nadu, India*
[2]*Associate Professor, Department of Computer Applications, PSG College of Technology, Coimbatore, Tamil Nadu, India*

Abstract -*The analysis on Social Networks converges on revealing the unknown patterns of people collaboration with respect to several domains. This paper targets to predict the missing data in a Social Network by employing the techniques available in Recommender systems. It provides a list of recommendations through a mix of latent factor model and neighborhood model. The latent factor model is used to discover the various reference links that is built on user's collaboration. The most collaborative reference links are further verified by means of neighborhood model. The experiments are conducted on UCI network data repository and the results are found to be convincing. The proposed approach progresses meaningfully which concerns about the performance of link prediction in social networks.*

Key Words: Collaborative filtering, Link Prediction, Non negative factorization, Recommender systems, Singular Value Decomposition, Social Networks.

1. INTRODUCTION

Social networks are graph structures whose vertices or nodes represent people or other entities embedded in a social context, and whose edges represent collaboration between these entities. Lots of research has been done recently to study different properties of the networks and the challenge is to analyze the dynamic property of social networks. Such complex analysis of large, multi-relational social networks has led to an interesting field of study known as Social Network Analysis (SNA).Link prediction is the only sub-field of SNA which has focus on links between objects rather than objects themselves. This makes link prediction interesting and different from traditional data mining areas which focus on objects.

Recommender systems are widely used in many applications that suggest products and items to potential users. Recommender systems or recommendation system is a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. Recommender systems shown in Figure 1 have become more common in recent years and the most popular areas are probably music, movies, news, books, research articles, social tags, financial services, life insurance, Twitter followers and products in general.

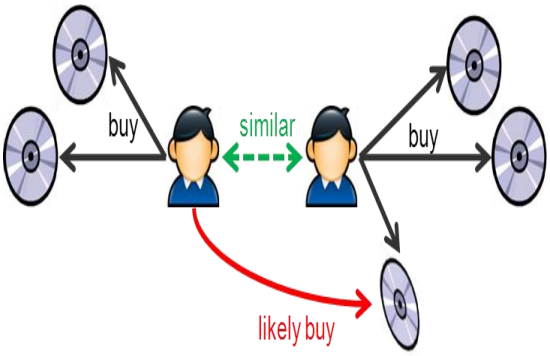

Figure 1. Recommender systems

The collaborative filtering is the most successful recommendation algorithm which helps to recommend a missing data or items. It has been applied to many different kinds of data like sensing and monitoring data. It is proved that memory based collaborative filtering is best to predict missing data, since it utilize the entire user-item database for prediction.

The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with similar tastes people. The collaborative filtering explores techniques for matching people with similar interests and making recommendations on this basis. The collaborative filtering algorithms often require customer's active participation and algorithms which is used to match people with similar interests.

The content-based filtering is a technique where an user recommend an item based upon a description of the item and a profile of the user's interests. In a content-based filtering, keywords are used to describe the items and a user profile is built to indicate the type of the item this user likes. The advantage of content-based filtering is user independency and transparency.In many recommender systems a number of techniques have been proposed including content based, collaborative filtering which suffers from three problems sparsity, scalability and cold-start.

Sparsity: The sparsity problem where the users or consumers are typically represented by the items they have purchased or rated. Because of sparsity, it is highly probable that the similarity (or correlation) between two given users is zero, rendering collaborative filtering useless. Even for pairs of users that are positively correlated, such correlation measures may not be reliable.

Scalability: In many of the environments, these systems make recommendations, there are millions of users and products. Thus, a large amount of computation power is often necessary to calculate recommendations.

Cold-start problem: The cold-start problem refers to the lack of information which is not sufficient for generating recommendations for a new user to purchase an item.

To solve a cold-start problem, a hybrid approach is proposed. In this paper, latent factor model and neighborhood model are employed for predicting the user preferences (links) accurately.

The paper is organized as follows: the related work is presented in next section, while the preliminaries, the problem and proposed work is described in section 3, 4 and 5 . In section 6, an experimental study is conducted on real data sets. The complexity analysis is represented in section 7 and 8. Finally, section 9 is concluded and the future work is provided.

2. RELATED WORKS

In [1], the authors have proposed the boosted collaborative filtering for improving the recommendation of users for an item and this paper have tackled the sparsity problem through content based filtering by means of pseudo user ratings vector. There by the performance have been improved by collaborative filtering. In [2], the authors have addressed on scalability and sparsity problem in the collaborative filtering and a personalized recommendation have been proposed. It combines the user clustering technology and item clustering technology.

In [3], the author has factored the neighborhood model, by applying both item-item and user-user implementation to scale linearly with the size of the data. In [4], the authors have proposed a novel approach for predicting student performance that uses the recommender systems techniques and compared with the logistic/linear regression and found that result was convincing. In [5], the authors have extended the idea of analyzing user-item interactions as graph and have employed link prediction approach for collaborative filtering recommendations.

In [6], the authors have proposed network topology and preference correlation for finding the people with similar interests and examined the use of preference correlation in social network and an online community. In [7], the authors have rectified the drawbacks of traditional collaborative filtering by making the use of belief distribution algorithm which uses the rating values rather than a point rating value for predictions. In [8], the authors have considered the bipartite graph, matrix and tensor based methods for predicting the future links. In [9], the authors have made an attempt to analyze how users/agents collaborate in a collaborative recommender systems and the ideal collaboration model optimize the performance of these systems.

In [10], the authors have devised a novel framework that considers the heterogeneous and reciprocal knowledge from collaborative information and finally its impact on link prediction have been demonstrated. In [11], the authors have employed singular value decomposition for trust modeling in social network, that estimate the trust using real valued matrix of the reputation ratings of the agents in the network. In [12], the authors have proposed raking factor graph model for predicting links in social networks, which effectively improves the predictive performance.

In [13], the authors have studied the robustness of four link prediction algorithms based on local information similarity. In [14], the latent factor and neighborhood model for providing a top-k recommendation list, thereby increasing the accuracy. In [15], the authors have applied the non-negative matrix factorization to the statistical analysis of a multivariate data. In [16], the authors have shown that matrix decomposition could be applied for analyzing the structure of social networks.

3. PRELIMINARIES

Link prediction is major task in Social Network Analysis. It has a wide application in recommender systems. In recent era, recommender systems techniques are used in ecommerce sites where a customer can buy or view the ratings given by the users. Based on this ratings, the effectiveness of the marketing strategies can be improved.

3.1 PEARSON CORRELATION COEFFICIENT

One of the most often used similarity metrics in collaborative-based systems is Pearson's correlation coefficients. Pearson's correlation reflects the degree of linear relationship between two variables, i.e. the extent to which the variables are related, and ranges from $+1$ to -1. A correlation of $+1$ means that there is a perfect positive linear relationship between variables or in other words two users have very similar tastes, whereas a negative correlation indicates that the users have dissimilar tastes. Pearson's correlation coefficients are used to determine the degree of correlation between an active user and another user.

It uses user rating to compute similarity between the users or items which is used for making recommendations. The similarity between the items a and u is given by

$$sim_{a,u} = \frac{\sum_{i\in I}(r_{ai} - \bar{r}_a)(r_{ui} - \bar{r}_u)}{\sqrt{\sum_{i\in I}(r_{ai} - \bar{r}_a)^2}\sqrt{\sum_{u\in U}(r_{ui} - \bar{r}_u)^2}} \quad (1)$$

where $sim_{a,u}$ is the similarity between the users a and u, i is the item, I is a set of items, r_{ui} is the rating given by the user u for the item i, r_{ai} is the rating given by the active user a for the item i ,$\bar{r}_u$ is the mean rating given by the user u, $\bar{r}_a$ is the mean rating given by the active user a. Once the similarity measure is calculated the rating for any user-item pair can be predicted using the idea of weighted sum. Finally, the prediction is given by

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u\in K}(r_{ui} - \bar{r}_u) * sim_{a,u}}{\sum_{u\in K} sim_{a,u}} \quad (2)$$

where $p_{a,i}$ is the prediction for the active user a for item i.

3.2 SINGULAR VALUE DECOMPOSITION

The singular value decomposition of a matrix is usually referred to as the SVD. This is the final and best factorization of a matrix:

$$A = U\sum V^T \quad (3)$$

where U is a user feature matrix and V^T is the movie feature matrix and both are orthogonal matrix, $\sum$ is diagonal matrices. In the decomposition $A = U\sum V^T$, A can be any matrix.

The SVD models are so popular, because of its attractive accuracy and scalability. In this model, each user u is associated with a user factors vectors $p_u \in \mathbb{R}^f$ and each item i with an item-factors vector $q_i \in \mathbb{R}^f$. The prediction is calculated by using

$$r_{ui} = b_{ui} + q_i^T p_u \quad (4)$$

where r_{ui} is the rating of the user u for an item i, b_{ui} is the baseline estimate for an unknown ratingr_{ui} and it is calculated using

$$b_{ui} = \mu + b_u + b_i \quad (5)$$

where μ is the overall average rating and b_u and b_i are the observed deviations of user u and item i respectively from the average. For example, suppose that the baseline estimate for the rating of the movie "Harry potter" by the user Bob is to be calculated. Assume that the overall average rating μ is 3.6 stars and it is also believed that the movie is better than an average movie and b_i could be rated as 0.6 stars which is above the average. On the other hand, Jack tends to rate 0.4 stars lower than the average. Therefore, baseline estimate for b_{ui} Harray potter rating by Jack would be 3.8 stars by using the formula 3.6 - 0.4 + 0.6

In order to estimate b_u and b_i, one can solve the least square problem:

$$\min_{b*}\sum_{(u,i)\in\kappa}(r_{ui} - \mu - b_u - b_i)^2 + \lambda_1(\sum_u b_u^2 + \sum_i b_i^2) \quad (6)$$

In this the first term $\sum_{(u,i)\in\kappa}(r_{ui} - \mu - b_u - b_i)^2$ is used to find the b_u and the second term $\lambda_1(\sum_u b_u^2 + \sum_i b_i^2)$ is used to find b_i that fit the given ratings. It acts as a regularizing term that avoids over fitting problem.

3.3 NON-NEGATIVE FACTORIZATION MODEL

The Non-negative Matrix Factorization (NMF) model is similar to SVD. Here the collaborative filtering is based on non-negative factorization model and it is given by the formula:

$$r_{ui} = p_u . q_i \quad (7)$$

where r_{ui} is the estimated rating for a given user u and item i is given by the scalar product, p_u is the vector associated with each user u and q_i is the vector associated with each item i. The optimization procedure is a stochastic gradient descent with a specific choice of step size that ensures non-negativity of factors, provided that their initial values are also positive.

$$p_{uf} \leftarrow p_{uf} . \frac{\sum_{i\in I_u} q_{if} . r_{ui}}{\sum_{i\in I_u} q_{if} . r_{ui} + \lambda_u |I_u| p_{uf}} \quad (8)$$

where p_{uf} is the initial value of the user factors, I_u is the set of all items rated by the user u, q_{if} is the initial value of the item factors, r_{ui} is the true rating of user u for the item i, λ_u is the regularization parameter of the user u.

$$q_{if} \leftarrow q_{if} . \frac{\sum_{u\in U_i} p_{uf} . r_{ui}}{\sum_{u\in U_i} p_{uf} . r_{ui} + \lambda_i |U_i| q_{if}} \quad (9)$$

where U_i is the set of all users that have rated item i, λ_i is the regularization parameter of the user i. This algorithm is highly dependent on initial values and the user factors and item factors are uniformly initialized between initial low and initial high values. The prediction is given by

$$r_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (10)$$

It still ensures the positive factors. It yields better accuracy, but the biased version seems highly to prove the over fitting problem.

4. PROBLEM DESCRIPTION:

Given an undirected graph $G = (V, E)$ where V represents either an item or an user and E represents the relationship between an item and an item that occur at time t whether the user will purchase the particular item at time t_1 predict t_0

5. PROPOSED WORK

This paper provide an effective way to overcome the cold-start problem and improve predictions of an item for the user by recommender systems techniques. The collaborative filtering is a most common method in recommendation system. In a collaborative filtering SVD and NMF both are used as a recommender systems techniques for providing the necessary recommendations. In recent decades, most of the collaborative filtering methods are based on user-item ratings which is in a matrix form. In user-item ratings matrix, the row represents an user and column represents an item. The collaborative filtering method focuses on predicting the unknown rating for an item by a particular user. Based on the history of ratings given by the user to an item, it is easy to predict the unknown ratings.

The user-item ratings matrix is found to be very sparse. In order to reduce the sparsity, SVD and NMF are used. It is a matrix factorization technique which is employed on the user-item ratings matrix. Hence, the aforementioned sparsity problem is encountered. The proposed method is twofold, one is to find the top 10 ratings of an item by all the users through SVD and NMF model and to compare which model produces minimum error rate. The second is to verify whether the rating given by users for the item is correct or not through Euclidean measure and the overall flow is shown in Figure 2.

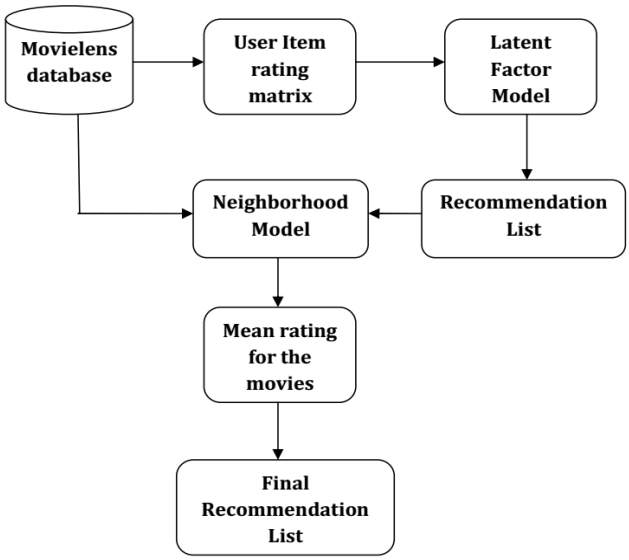

Figure 2. Proposed Work

Algorithm

Compute_TopN(predictions, n=10)
$top_n \leftarrow 0$
for all $U_{id} \in Movielens$ $top_n \leftarrow 0$
for all $U_{id} \in Movielens$
    $Map\ predictions\ to\ each\ U_{id}$
    $top_n \leftarrow predictions$
    Sort top_n
Return top_n
Train:
for all $i \in R(u)$ do
    d$\leftarrow q_i^T p_u$
    $r_{ui} \leftarrow b_{ui}$ + d
    $b_{ui} \leftarrow \mu + b_u + b_i$
To estimate b_u and b_i solve least squares problem
$\min_{b*} \sum_{(u,i)\in\kappa}(r_{ui} - \mu - b_u - b_i)^2 + \lambda_1(\sum_u b_u^2 + \sum_i b_i^2)$
Prediction:
for all $(u, i) \notin training$
    predictions$\leftarrow testing\ set$
Compute_TopN(predictions, n=10)
Print the recommended items for each user
Evaluate the performance of the algorithm using

6. EXPERIMENTAL SETUP AND RESULTS

The experiments are conducted on a 2.50GHz Intel Dual core PC with 4GB RAM running Microsoft 7 ultimate. The SVD and NMF algorithms are implemented using Python 2.7.8. An analysis is made to evaluate the performance of the proposed technique.

The efficiency of these algorithms are evaluated using the Movielens dataset. The dataset is collected and maintained by Grouplens Research project at the University of Minnesota. It consists of 100000 ratings with the scale ranging from 1 to 5. The ratings are given by 943 users. Each user has rated at least 20 movies among 1682 movies.

The users and movies are numbered consecutively from 1 and these data are randomly ordered. The data set is divided into training and testing set with 80% and 20% of the original data using 5-fold cross validation.

The SVD algorithm predicts the top 10 ratings of the movies for each and every user. For example, the userid *u315* has given higher ratings for 10 movies and are listed as follows: [*m408, m357, m89, m114, m427, m483, m258, m480, m474, m169*]. The ratings for the 10 movies given by the user *u315* is shown in Figure 3.

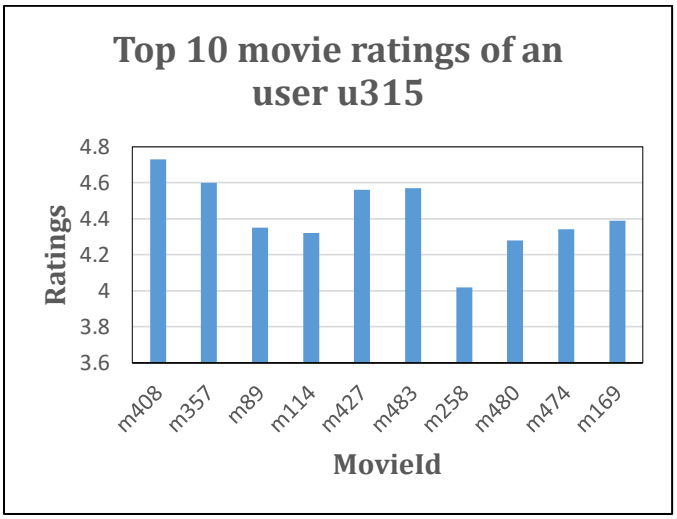

Figure 3. Top 10 movie ratings of an user u315

It is observed that all the movies have the ratings greater than 4, so there is a higher probability for recommending these movies to others in future.

In connection to this, the movie id *m315* have been rated by number of users with the rating scale is shown in Figure 4.

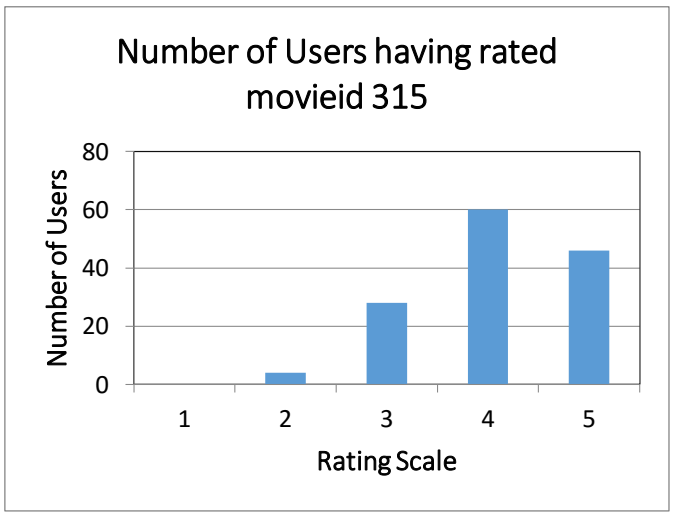

Figure 4. Number of users having rated movieid 315

It is observed that a rating scale of 4 is given by 60 users. The movieid *m315* have been rated higher among the other movies and is shown in Figure 5.

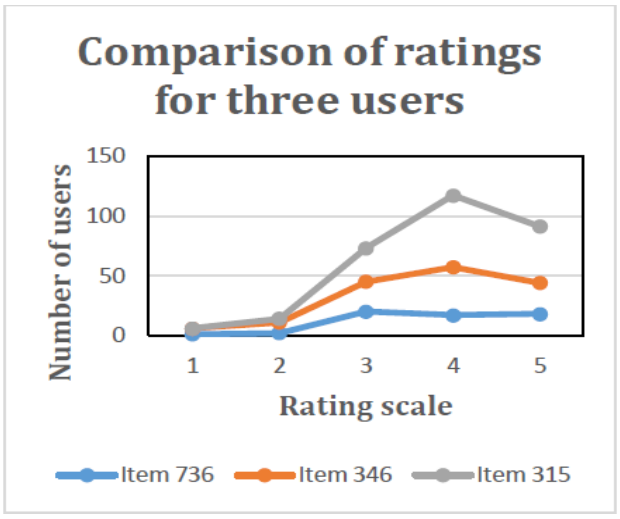

Figure 5. Comparison of ratings for three users

Among all the three movieids, more number of users have rated the movieid *m315*. So it is observed that there is a higher probability for the movieid *m315* to be rated more in future.

In order to verify that the movieid 315 is rated higher, the neighborhood algorithm is used and it is confirmed that the mean rating of the movieid 315 is also higher. So it is strongly proved that the probability of the movieid 315 to be rated higher by other users in future. The movies having higher mean rating are shown in Figure 6.

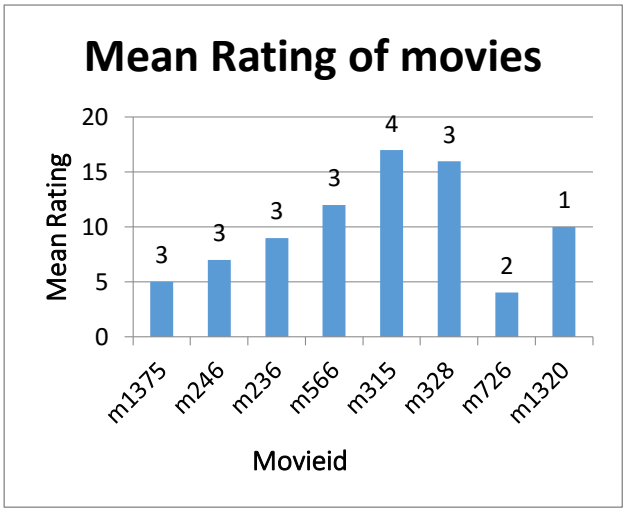

Figure 6. Mean Rating of movies

Hence, the proposed algorithm first identifies the top 10 ratings for any item of all the users. From this, the particular movieid is found that has been rated higher by most of the users. Secondly, the movieid has been verified by taking the mean rating of the movies and it is concluded that the concern user is the higher probability for rating the movie in future.

7. THEORETICAL ANALYSIS

The proposed approach includes two major tasks, the first task is to find the top 10 ratings of all the movies for all the users. The time complexity would be $O(u * i)$, where u is the user and i is the item. The second task is to find mean rating of the movies using the Euclidean measure between the two users and the time complexity would be $O(u_j * u_k)$, where u_j is the j^{th} user and u_k is the k^{th}user. Hence, the total time complexity of the proposed approach would be $O(\max(u * i, u_j * u_k))$, since the number of users is less than the number of movies, the time complexity would be $O(u * i)$. The space complexity would be the user item ratings which is $O(u * i)$.

8. EMPIRICAL ANALYSIS

The prediction accuracy is verified considering the error rate and it is computed by using Mean Average Error (MAE) and Root Mean Square Error (RMSE). The MAE is defined by

$$MAE = \frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui} \in \hat{R}} |r_{ui} - \hat{r}_{ui}| \qquad (11)$$

where$\hat{R}$ is the overall ratings of all the items, r_{ui} is the observed ratings, $\hat{r}_{ui}$ is the estimated ratings.

$$RMSE = \sqrt{\frac{1}{|\hat{R}|}\sum_{\hat{r}_{ui} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2} \qquad (12)$$

The performance of the SVD algorithm is verified using MAE and RMSE metrics for 5-fold cross validation. It is shown in Figure 7.

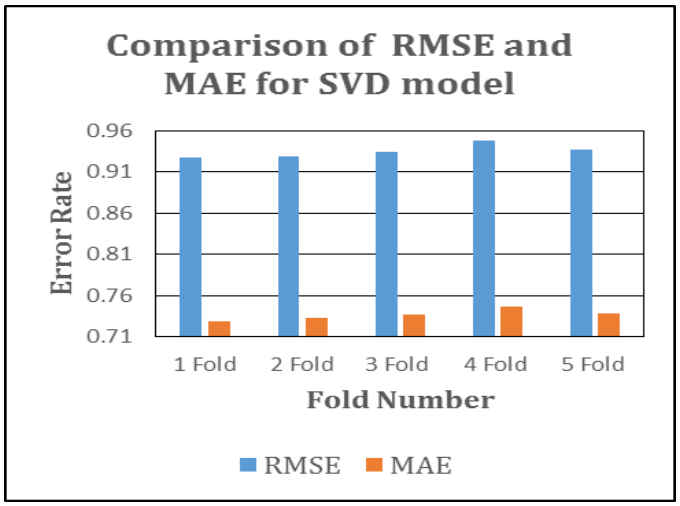

Figure 7. Comparison of RMSE and MAE for SVD model

Similarly, the performance of the NMF algorithm is verified using MAE and RMSE metrics for 5-fold cross validation. It is shown in Figure 8.

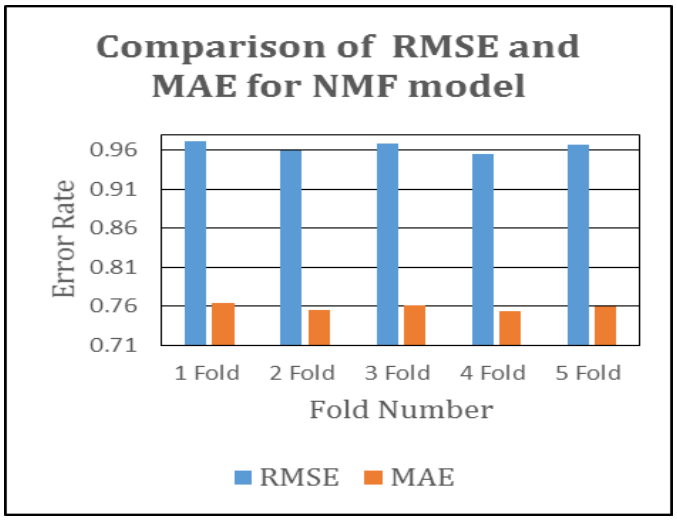

Figure 8. Comparison of RMSE and MAE for NMF model

It is observed that RMSE and MAE for SVD is better than NMF model. Since, the error rate of SVD is low, the average of 5-folds are computed and it is shown in Figure 9.

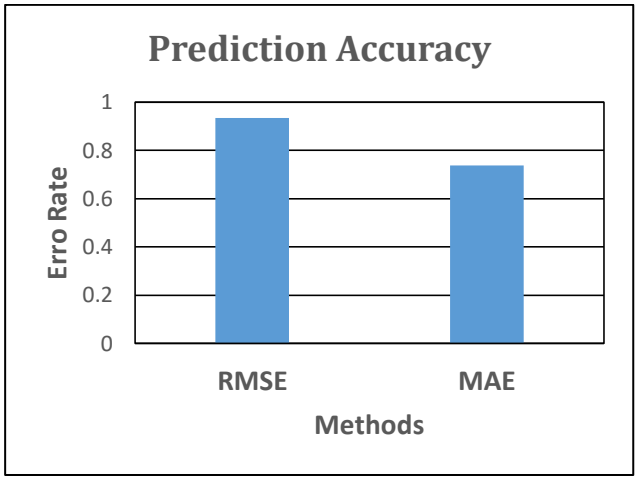

Figure 9. Prediction Accuracy

So, it is concluded that MAE of SVD provides low error rate. It is seen that the items are recommended for the users by considering the SVD model.

9. CONCLUSION

This paper predicts missing data in a social network using recommender systems techniques. There is a lack of work in finding the missing data by means of recommender systems. It is very powerful for extracting additional value for business from the user databases. It help us to find the items that the user wants to buy or would like to view. It benefits the users by helping them to find the item they like. They are increasingly used as a tool for E- commerce on the web. This paper allows scaling large datasets as well as giving accurate recommendations. The experimental results show that this approach significantly

works better. The future research direction must include evolutionary algorithm as a part of link prediction using recommender systems and also be applied to different domains like fraud detection, terrorism and website navigation.

ACKNOWLEDGEMENT:

We thank the anonymous reviewers for their valuable comments, which help to enhance our paper in a better way.

REFERENCES:

[1] Melville, Prem, Raymond J. Mooney, and RamadassNagarajan. "Content-boosted collaborative filtering for improved recommendations." Aaai/iaai. 2002.

[2] Gong, Songjie. "A collaborative filtering recommendation algorithm based on user clustering and item clustering." JSW5.7 (2010): 745-752.

[3] Yehuda Koren. "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering". Yahoo! Research.

[4] Thai-Nghe, Nguyen, et al. "Recommender systems for predicting student performance." Procedia Computer Science1.2 (2010): 2811-2819.

[5] Chen, Hsinchun, Xin Li, and Zan Huang. "Link prediction approach to collaborative filtering." Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on. IEEE, 2005.

[6] Hogg, Tad. "Inferring preference correlations from social networks." Electronic Commerce Research and Applications9.1 (2010): 29-37.

[7] McLaughlin, Matthew R., and Jonathan L. Herlocker. "A collaborative filtering algorithm and evaluation metric that accurately model the user experience." Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004.

[8] Dunlavy, Daniel M., Tamara G. Kolda, and EvrimAcar. "Temporal link prediction using matrix and tensor factorizations." ACM Transactions on Knowledge Discovery from Data (TKDD) 5.2 (2011): 10.

[9] Palau, Jordi, et al. "Collaboration analysis in Recommender systems using social networks." International Workshop on Cooperative Information Agents. Springer, Berlin, Heidelberg, 2004.

[10] Cai, Xiongcai, et al. "Reciprocal and heterogeneous link prediction in social networks." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, 2012.

[11] BundiNtwiga, Davis, Patrick Weke, and Michael Kiura Kirumbu. "Trust model for social network using singular value decomposition." Interdisciplinary Description of Complex Systems: INDECS 14.3 (2016): 296-302.

[12] Dong, Yuxiao, et al. "Link prediction and recommendation across heterogeneous social networks." Data Mining (ICDM), 2012 IEEE 12th International Conference on. IEEE, 2012.

[13] Wang, Liang, Ke Hu, and Yi Tang. "Robustness of Link-Prediction Algorithm Based on Similarity and Application to Biological Networks." Current Bioinformatics 9.3 (2014): 246-252.

[14] Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008.

[15] Lee, Daniel D., and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." Advances in neural information processing systems. 2001.

[16] Skillicorn, David B. "Social Network Analysis Via Matrix Decompositions." (2006): 367-391.