

Hybrid Trust-Driven Recommendation System for E-commerce Networks

Pavan Kumar K. N¹, Samhita S Balekai¹, Sanjana P Suryavamshi¹, Sneha Sriram¹, R. Bhakthavathsalam²

¹Department of Information Science and Engineering, Sir M. Visvesvaraya Institute of Technology
Bangalore, Karnataka, India

¹pavan1011@gmail.com, ¹samhita13@gmail.com, ¹sanjanap37@gmail.com, ¹sneh2093@gmail.com

²Super computer Education and Research Centre, Indian Institute of Science
Bangalore, Karnataka, India

²bhaktha@serc.iisc.ernet.in

Abstract

In traditional recommendation systems, the challenging issues in adopting similarity-based approaches are sparsity, cold-start users and trustworthiness. We present a new paradigm of recommendation system which can utilize information from social networks including user preferences, item's general acceptance, and influence from friends. A probabilistic model, particularly for e-commerce networks, is developed in this paper to make personalized recommendations from such information. Our analysis reveals that similar friends have a tendency to select the same items and give similar ratings. We propose a trust-driven recommendation method known as HybridTrustWalker. First, a matrix factorization method is utilized to assess the degree of trust between users. Next, an extended random walk algorithm is proposed to obtain recommendation results. Experimental results show that our proposed system improves the prediction accuracy of recommendation systems, remedying the issues inherent in collaborative filtering to lower the user's search effort by listing items of highest utility.

Keywords: Recommendations system, Trust-Driven, Social Network, e-commerce, HybridTrustWalker.

1. Introduction

Recommendation systems (RS) (sometimes replacing "system" with a synonym such as platform or engine) are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that user would give to an item. RSs have changed the way people find products, information, and even other people. They study patterns of behaviour to know what someone will prefer from among a collection of things he has never experienced. RSs are primarily directed towards individuals who lack sufficient personal experience or competence to evaluate the potentially overwhelming number of alternative items that a Web site, for example, may offer. A case in point is a book recommendation system that assists users to select a book to read. In the popular Website, Amazon.com, the site employs a RS to personalize the online store for each customer. Since recommendations are usually

personalized, different users or user groups receive diverse suggestions. In addition there are also non-personalized recommendations. These are much simpler to generate and are normally featured in magazines or newspapers. Typical examples include the top ten selections of books, CDs etc. While they may be useful and effective in certain situations, these types of non-personalized recommendations are not typically addressed by RS research.

1.1 Recommendation System Functions

First, we must distinguish between the roles played by the RS on behalf of the service provider from that of the user of the RS. For instance, a travel recommendation system is typically introduced by a travel intermediary (e.g., Expedia.com) or a destination management organization (e.g., Visitfinland.com) to increase its turnover (Expedia), i.e. sell more hotel rooms, or to increase the number of tourists to the destination. Whereas, the user's primary motivations for accessing the two systems is to find a suitable hotel and interesting events/attractions when visiting a destination [1]. In fact, there are various reasons as to why service providers may want to exploit this technology:

Increase in sales: This goal is achieved because the recommended items are likely to satisfy users' functional preferences. Presumably the user will recognize this after having tried several recommendations. From the service providers' point of view, the primary goal of introducing a RS is to increase the conversion rate, i.e. the number of users that accept the recommendation and consume an item compared to the number of visitors browsing through for information.

Exposure to a wider product range: Another major function of a RS is to enable the user to select items that might be hard to find without a precise recommendation. For instance, in a movie RS such as Netflix, the service

provider is interested in renting all the DVDs in the catalogue, not just the most popular ones.

Consolidating user satisfaction and fidelity: The user will find the recommendations interesting, relevant, and accurate, and when combined with a properly designed human-computer interaction she will also enjoy using the system. Personalization of recommendations improves user loyalty. Consequently, the longer the user interacts with the site, the more refined her user model becomes, i.e., the system representation effectively customizing recommendations to match the user's preferences.

Improve QoS through customer feedback: Another important function of a RS, which can be leveraged to many other applications, is the description of the user's preferences, either collected explicitly or predicted by the system. The service provider may then decide to reuse this knowledge for a number of other goals such as improving the management of the item's stock or production.

1.2 Common Recommendation Techniques

In order to implement its core function, identifying the useful items for the user, an RS must predict that an item is worth recommending. In order to do this, the system must be able to predict the utility of some of them, or at least compare the utility of some items, and then decide what items to recommend based on this comparison. The prediction step may not be explicit in the recommendation algorithm but we can still apply this unifying model to describe the general role of a RS. Some of the recommendation techniques are given below:

Collaborative filtering: The simplest and original implementation of this approach recommends the items that other users with similar tastes liked, to the target user. The similarity of taste between two users is calculated based on the rating history of the users. Collaborative filtering is considered to be the most popular and widely implemented technique in RS. Neighbourhood methods focus on relationships between items or, alternatively, between users. An item-item approach models the preference of a user to an item based on ratings of similar items by the same user. Nearest-neighbours methods enjoy considerable popularity due to their simplicity, efficiency, and their ability to produce accurate and personalized recommendations. The authors will address the essential decisions that are required when implementing a neighbourhood based recommender system and provide practical information on how to make such decisions [2].

Content-based: The system learns to recommend items that are similar to the ones that the user liked in the past. The similarity of items is calculated based on the features associated with the compared items. For example, if a user has positively rated a movie that belongs to the

horror genre, then the system can learn to recommend other movies from this genre.

Demographic: This type of system recommends items based on the demographic profile of the user. The assumption is that different recommendations should be generated for different demographic niches. Many Web sites adopt simple and effective personalization solutions based on demographics. For example, users are dispatched to particular Web sites based on their language or country. Or suggestions may be customized according to the age of the user. While these approaches have been quite popular in the marketing literature, there has been relatively little proper RS research into demographic systems.

Knowledge-based: Recommendation based on specific domain knowledge about how certain item features meet users' needs and preferences and, ultimately, how the item is useful for the user. In these systems a similarity function estimates how well the user needs match the recommendation. The similarity score can be directly interpreted as the utility of the recommendation for the user. Content-based systems are another type of knowledge-based RSs. In terms of used knowledge, both systems are similar: user requirements are collected; repairs for inconsistent requirements are automatically proposed in situations where no solutions could be found; and recommendation results are explained. The major difference lies in the way solutions are calculated. Knowledge-based systems tend to work better than others at the beginning of their deployment but if they are not equipped with learning components they may be surpassed by other shallow methods that can exploit the logs of the human/computer interaction (as in CF).

1.3 Problems in Existing Recommendation Systems

Sparsity problem: In addition to the extremely large volume of user-item rating data, only a certain amount of users usually rates a small fraction of the whole available items. As a result, the density of the available user feedback data is often less than 1%. Due to this data sparsity, collaborative filtering approaches suffer significant difficulties in identifying similar users or items via common similarity measures, e.g., cosine measure, in turn, deteriorating the recommendation performance.

Cold-start problem: Apart from sparsity, cold-start problem, e.g., users who have provided only little feedback or items that have been rated less frequently or even new users or new items, is a more serious challenge in recommendation research. Because of the lack of user feedback, any similarity-based approaches cannot handle such cold-start problem.

Trustworthiness problem: Prediction accuracy in recommendation systems requires a great deal of

consideration as it has such a strong impact on customer experience. Noisy information and spurious feedback with malicious intent must be disregarded in recommendation considerations. Trust-driven recommendation methods refer to a selective group of users that the target user trusts and uses their ratings while making recommendations. Employing 0/1 trust relationships, where each trusted user is treated as an equal neighbour of the target user, proves to be rudimentary as it does not encapsulate the underlying level of trust between users.

As a solution, the concept of Trust Relevancy [3] is introduced first, which measures the trustworthiness factor between neighbours, defining the extent to which the trusted user's rating affects the target user's predicted rating of the item. Next, the algorithm HybridTrustWalker performs a random walk on the weighted network. The result of each iteration is polymerised to predict the rating that a target user might award to an item to be recommended. Finally, we conduct experiments with a real-world dataset to evaluate the accuracy and efficiency of the proposed method.

2. Related Work

Since the first paper published in 1998, research in recommendation systems has greatly improved reliability of the recommendation which has been attributed to several factors. Paolo Massa and Bobby Bhattacharjee in their paper Using Trust in Recommendation System: An Experimental Analysis (2004) show that any two users have usually few items rated in common. For this reason, the classic RS technique is often ineffective and is not able to compute a user similarity weight for many of the users. In 2005, John O'Donovan and Barry Smyth described a number of ways to establish profile-level and item-level trust metrics, which could be incorporated into standard collaborative filtering methods.

Shao et al (2007) proposed a user-based CF algorithm using Pearson Correlation Coefficient (PCC) to compute user similarities. PCC measures the strength of the association between two variables. It uses historical item ratings to classify similar users and predicts the missing QoS values of a web service by considering QoS value of service used by users similar to her [4].

Zheng et al furthered the collaborative filtering dimension of recommendation systems for web service QoS prediction by systemically combining both item-based PCC (IPCC) and user-based PCC (UPCC). However, the correlation methods face challenges in providing recommendations for cold-start users as these methods consider users with similar QoS experiences for same services to be similar [3].

The most common trust-driven recommendation approaches make users explicitly issue trust statements for other users. Golbeck proposed an extended-breadth first-search method in the trust network for prediction called TidalTrust [5]. TidalTrust finds all neighbours who have rated the to-be recommended service/item with the shortest path distance from the given user and then aggregates their ratings, with trust values between the given user and these neighbours as weights. Mole Trust [6] is similar to TidalTrust but only considers the raters within the limit of a given maximum-depth. The maximum-depth is independent of any specific user and item.

3. Proposed System

In a trust-driven recommendation [7] paradigm, the trust relations among users form a social network. Each user invokes several web services and rates them according to the interaction experiences. When a user needs recommendations, it predicts the ratings that the user might provide and then recommends services with high predicted ratings. Hence, the target of the recommendation system predicts users' ratings on services by analysing the social network and user-service rating records.

There is a set of users $U = \{u_1, u_2, \dots, u_m\}$ and a set of services $S = \{s_1, s_2, \dots, s_n\}$ in a trust driven recommendation system. The ratings expressed by users on services/items are given in a rating matrix $R = [R_{u,s}]_{m \times n}$. In this matrix, $R_{u,s}$ denotes the rating of user u on service (or item) s . $R_{u,s}$ can be any real number, but often ratings are integers in the range of [3]. In this paper, without loss of generality, we map the ratings 1, ..., 5 to the interval [0,1] by normalizing the ratings. In a social rating network, each user u has a set S_u of direct neighbours, and $t_{u,v}$ denotes the value of social trust u has on v as a real number in [0, 1]. Zero means no trust, and one means full trust. Binary trust networks are the most common trust networks (Amazon, eBay, etc.). The trust values are given in a matrix $T = [T_{u,v}]_{m \times m}$. Non-zero elements $T_{u,v}$ in T denote the existence of a social relation from u to v . Note that T is asymmetric in general [8].

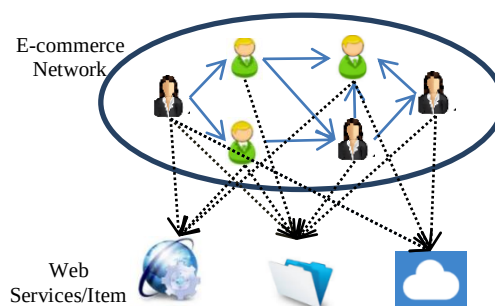


Fig. 1. Illustration of trust-driven recommendation approach

Thus, the task of a trust-driven service recommendation system is as follows: Given a user u_0 belonging to U and a service s belonging to S for which $R_{u_0,s}$ is unknown, predict the rating for u_0 on service s using R and T . This is done by first determining a degree of trust between users in the social network to obtain a weighted social network from the Epinions data, using 0/1 trust relation from the input dataset and cosine similarity measures of user and service latent features. Then, a random walk performed on this weighted network yields a resultant predicted rating. Ratings over multiple iterations are polymerized to obtain the final predicted ratings.

3.1 Trust-Driven Recommendation Approach

Incorporating trust metrics in a social network does not absolutely affect the target user's ratings because the target user and trusted users might differ in interests, preferences and perception. The concept of trust relevancy considers both the trust relations between users together with the similarities between users. This section presents our approach in detail for trust-driven service recommendations. First, we define the concept of trust relevancy, on which our recommendation algorithm is based. Then, we introduce the algorithm HybridTrustWalker by extending the random walk algorithm in [7]. Lastly, the predicted ratings are returned. The methodology is summarized as shown in Fig. 2.

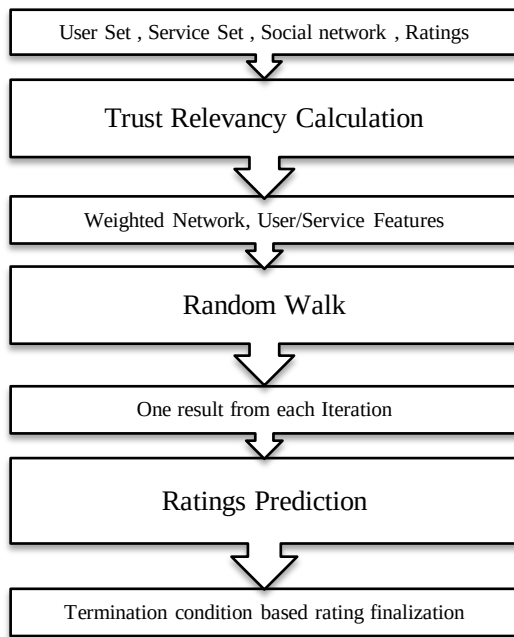


Fig. 2. Proposed Methodology

Given user u and v , the trust relevancy between u and v is as follows:

$$tr(u,v) = simU(u,v) * t(u,v) \quad (1)$$

Here, $simU(u,v)$ is the similarity of users u and v , and

$t(u,v)$ is the degree of trust of u towards v . By computing the trust relevancy between all connected users in a social network, we can obtain a weighted trust network (SN^+) , where the weight of each edge is the value of trust relevancy. The aim of calculating trust relevancy is to determine the degree of association between trusted neighbours.

In RSs, the user-item/service rating matrix is usually very large in terms of dimensionality but most of the score data is missing. Therefore, matrix factorization (MF) has been widely utilized in recommendation research to improve efficiency by dimension reduction [9]. For an $m * n$ user-service rating matrix R , the purpose of matrix factorization is to decompose R into two latent feature matrices of users and items with a lower dimensionality d such that,

$$R \approx PQ^T \quad (2)$$

where $P \in R^{m \times d}$ and $Q \in R^{n \times d}$ represent the user and item latent feature matrices, respectively. Each line of the respective matrix represents a user or service latent feature vector. After decomposing the matrix, we use the cosine similarity measure to calculate the similarity between two users. Given the latent feature vectors of two users, u and v , their similarity calculation is as follows:

$$simU(u,v) = \cos(u,v) = \frac{u \cdot v}{|u| \cdot |v|} \quad (3)$$

where u and v are latent feature vectors of users u and v .

3.2 Recommendation Algorithm

The HybridTrustWalker algorithm attains a final result through multiple iterations. For each iteration, the random walk starts from the target user u_0 in the weighted trust network SN^+ . In the k^{th} step of the random walk in the trust network, the process will reach a certain node u . If user u has rated the to-be-recommended service s , then the rating of s from user u is directly used as the result for the iteration. Otherwise, the process has two options, one of which is:

- The random walk will stop at the current node u with a certain probability $\phi_{u,s,k}$. Then, the service s_i is selected from RS_u based on the probability $F_u(s_i)$. The rating of s_i from u is the result for the iteration.

The probability that the random walk stops at user u in the k -th step is affected by the similarity of the items that u has rated and the to-be-recommended service s . The more similar the rated items and s , the greater the probability is to stop. Furthermore, a larger distance between the user u and the target user u_0 can introduce more noise into the prediction. Therefore, the value of probability $\phi_{u,s,k}$ should increase when k increases [10].

Thus, the calculation for $\varphi_{u,s,k}$ is as follows:

$$\varphi_{u,s,k} = \max_{s_i \in RS_u} \text{SimS}(s_i, s) * \frac{1}{1 + e^{-\frac{k}{2}}} \quad (4)$$

where $\text{simS}(s_i, s)$ is the similarity between the services s_i and s . The sigmoid function of k can provides value 1 for big values of k , and a small value for small values of k . In contrast to collaborative filtering techniques [2], this method can cope with services that do not have ratings from common users. Service similarities are calculated using Matrix Factorization [8]:

$$\text{SimS}(s_i, s_j) = \cos(s_i, s_j) = \frac{s_i \cdot s_j}{|s_i| \cdot |s_j|} \quad (5)$$

When it is determined that user u is the terminating point of the walk, the method will need to select one service from RS_u . The rating of s_i from u is the outcome for the iteration. The probability of the chosen service $F_u(s_i)$ is calculated according to the following formula:

$$F_u(s_i) = \frac{\text{simS}(s, s_i)}{\sum_{s_j \in RS_u} \text{simS}(s, s_j)} \quad (6)$$

Services are selected $F_u(s_i)$ through a roulette-wheel selection [11], that is, services with higher values of $F_u(s_i)$ are more possible to be selected. Also, adopting the "six degrees of separation" [12], by setting the maximum step of each walk to 6, prevents infinite looping of the random walk.

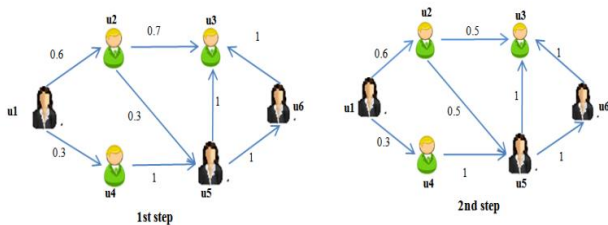


Fig. 3. Example of HybridTrustWalker

The other alternate option during the walk if the user u has not rated the to-be recommended service s is:

- The walk can continue with a probability of $1 - \varphi_{u,s,k}$. In which case, a target node for the next step is selected from the set of trusted neighbours of the user u .

To distinguish different users' contribution to the recommendation prediction, we propose that the target node v for the next step from the current user u is selected according to the following probability:

$$E_u(v) = \frac{\text{tr}(u, v)}{\sum_{x \in TU_u} \text{tr}(u, x)} \quad (7)$$

where $\text{tr}(u, v)$ is the trust relevancy introduced earlier. The trust relevancy guarantees that each step of the walk will choose the user that is more similar to the current user, making the recommendation more accurate and thus enhancing productivity and user acceptance.

3.3 HybridTrustWalker Algorithm

Input: U (user set), S (service set), R (rating matrix), SN^+ (weighted social network), u_0 (the target user), s (to-be-recommended service).

Output: r (predicted rating).

Pseudocode:

```

1  set k = 1 ; //the step of the walk
2  set u = u0 ; //set the start point of the walk as u0
3  set max-depth = 6 ; //the max step of the walk
4  set r = 0 ;
5  while (k <= max-depth) {
6      u = selectUser(u) ; //select v from TUu
        as the target of the next step based on
        the probability Eu(v).
7      if (u has rated s) {
8          r = ru,s ;
9          return r ;
10     }
11     else {
12         if (random (0,1) < φu,s,k || k == max-depth) {
13             //stop at the current node
14             si = selectService(u) ; //service
15             si is selected from RSu based
16             on the probability Fu(si).
17             r = ru,si ;
18             return r ;
19         }
20     }
21     k++ ;
22 }
23 return r ;
    
```

Fig.3 shows an example to illustrate the algorithm clearly. The weight of each edge represents the probability $E_u(v)$. Suppose the service s_3 is to be recommended for the user u_1 . For the first step of the walk, u_2 is more likely to be selected as the target node since the value of $E_u(u_2)$ is larger. If u_2 has rated s_3 with the rating r , r will be returned as the result of this walk (Line.7–9). Otherwise, if the termination condition (Line.12) is not reached, the walk would continue. For the second step, u_5 is selected. It should also check whether u_5 has rated s_3 . If u_5 has not rated s_3 but the termination condition is reached, it will select the most similar service to s_3 from the items u_5 has rated (Line.13). Then, the rating of the selected service by u_5 is returned as the result of this walk.

3.4 Ratings Prediction

The HybridTrustWalker algorithm attains a final result through multiple iterations. The final predicted rating is obtained by polymerizing the results returned from every iteration:

$$p_{uo,s} = \frac{1}{n} \sum_{i=1}^n r_i \quad (8)$$

where r_i is the result of each iteration, n is the number of iterations.

To obtain a stable predict result, the algorithm needs to perform an adequate number of random walks. We can decide the termination condition of the algorithm through the calculation of the variance of the prediction values. The variance of the prediction results after a random walk is denoted and calculated as:

$$\sigma_i^2 = \frac{1}{i} \sum_{j=1}^i (r_j - \bar{r})^2 \quad (9)$$

where r_j is the result of every iteration, i is the total number of iterations until the current walk, and σ_i^2 the variance obtained from the last i iterations, which will eventually tend to a stable value. When $|\sigma_{i+1}^2 - \sigma_i^2| \leq \varepsilon$, the algorithm terminates ($\varepsilon = 0.0001$).

4. Results and Discussion

We use the dataset of Epinions published by the authors of [11]. The large size and characteristically sparse user-item rating matrix makes it suitable for our study. This contains data of 49,290 users who have rated 139,738 items. There are a total of 664,824 ratings with 487,181 trust relations within the network.

We adopt the Root Mean Squared Error (RMSE), which is widely used in recommendation research, to measure the error in recommendations:

$$RMSE = \sqrt{\frac{\sum_{u,s} (R_{u,s} - \hat{R}_{u,s})^2}{N}} \quad (10)$$

where $R_{u,s}$ is the actual rating the user u gave to the service s and $\hat{R}_{u,s}$ which is the predicted rating the user u gave to the service s . N denotes the number of tested ratings. The smaller the value of $RMSE$ is, the more precisely the recommendation algorithm performs. We use the coverage metric to measure the percentage of pairs of $\langle \text{user}, \text{service} \rangle$, for which a predicted value can be generated:

$$\text{Coverage} = \frac{S}{N} \quad (11)$$

where, S denotes the number of predicted ratings and N denotes the number of tested ratings. We have to convert

RMSE into a precision metric in the range of $[0, 1]$. The precision is denoted as follows:

$$\text{precision} = 1 - \frac{RMSE}{4} \quad (12)$$

To combine RMSE and coverage into a single evaluation metric, we compute the F-Measure as follows :

$$\text{F-Measure} = \frac{2 \times \text{precision} \times \text{coverage}}{\text{precision} + \text{coverage}} \quad (13)$$

Comparison analysis of performance measure for various RS paradigms including collaborative filtering approaches:

Table 1: Comparing results for all users

Algorithms	RMSE	Coverage (%)	F-measure
Item based CF	1.345	67.58	0.6697
User based CF	1.141	70.43	0.7095
Tidal trust	1.127	84.15	0.7750
Mole trust	1.164	86.47	0.7791
Trust Walker	1.089	95.13	0.8246
HybridTrustWalker	1.012	98.21	0.8486

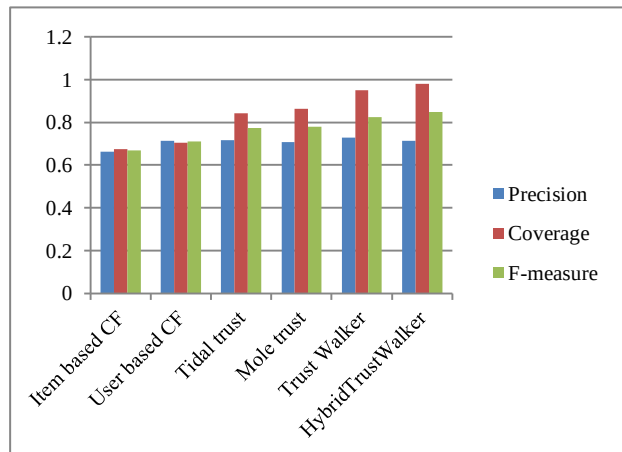


Fig.4. Comparing results of all users.

The reduction of precision of the proposed model is compensated by the increased coverage and F-measure as shown in Table 1 and Table 2 (in the case of cold-start users).

Table 2: Comparing results of cold-start users

Algorithms	RMSE	Coverage (%)	F-measure
Item based CF	1.537	23.14	0.3362
User based CF	1.485	18.93	0.2910
Tidal trust	1.237	60.75	0.6463
Mole trust	1.397	58.29	0.6150
Trust Walker	1.212	74.36	0.7195
HybridTrustWalker	1.143	79.64	0.7531

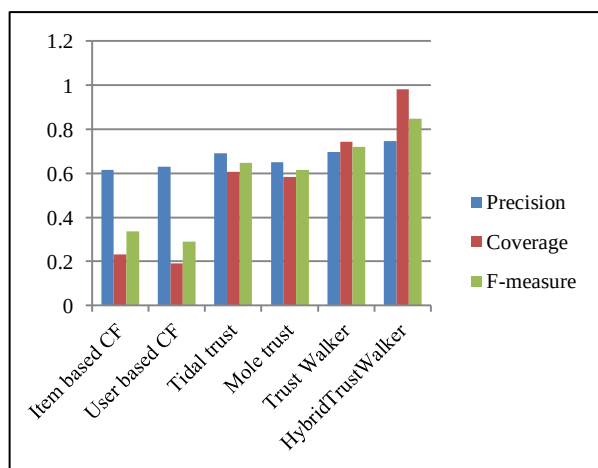


Fig. 5 Comparing results of cold-start users

This means the ratings from most number of relevant users is considered during the rating prediction in each step of the walk in HybridTrustWalker. Due to cold-start users (Fig 5), item-based and user-based CF performs poorly. They have highest RMSE and lowest coverage than all the other algorithms considered during analysis. Due to the introduction of trust factor, TidalTrust, MoleTrust and TrustWalker have improved coverage compared to CF whereas precision does not change much.

5. Conclusion

The proposed recommendation system has three main objectives: (1) Tackling the problem of recommendations with cold-start users; (2) Address the problem of recommendations with a large and sparse user-service rating matrix and (3) Solve the problem with trust relations in a recommendation system. Thus, the main contributions of HybridTrustWalker presented in this paper, include, introducing the concept of trust relevancy, which is used to obtain a weighted social network.

Furthermore, for this model, we develop a hybrid random walk algorithm. Existing methods usually randomly select the target node for each step when choosing to walk. By contrast, the proposed approach selects the target node based on trust and similarity. Thus, the recommendation contribution from trusted users is more accurate. We also utilize large-scale real data sets to evaluate the accuracy of the algorithm. The experimental results show that the proposed method can be directly applied in existing e-commerce networks with improved accuracy. Personalized service recommendation systems have been heavily researched in recent years and the proposed model provides an effective solution for the same. We believe that there is scope for improvement. For example, here, the trust relationships between users in the social trust network are considered to be invariant. But in reality, the trust relationship between users can change over time. In addition, the user ratings are also time sensitive. As a result, ratings that are not up-to-date may become noise information for recommendations. In large user communities, it is only natural that besides trust also distrust starts to emerge. Hence, the more users issuing distrust statements, the more interesting it becomes to also incorporate this new information. Therefore, we plan to include time sensitivity and the distrust factor in our future work.

Acknowledgments

The authors sincerely thank the authorities of Supercomputer Education and Research Center, Indian Institute of Science for the encouragement and support during the entire course of this work.

References

- [1] Ricci, Francesco, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer US, 2011.
- [2] Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426-434. ACM, 2008.
- [3] Deng, Shuiguang, Longtao Huang, and Guandong Xu. "Social network-based service recommendation with trust enhancement." *Expert Systems with Applications* 41, no. 18 (2014): 8075-8084.
- [4] Shao, Lingshuang, Jing Zhang, Yong Wei, Junfeng Zhao, Bing Xie, and Hong Mei. "Personalized qos prediction forweb services via collaborative filtering." In *Web Services, 2007. ICWS 2007. IEEE International Conference on*, pp. 439-446. IEEE, 2007.
- [5] Golbeck, Jennifer Ann. "Computing and applying trust in web-based social networks." (2005).
- [6] Massa, Paolo, and Paolo Avesani. "Trust-aware recommender systems." In *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 17-24. ACM, 2007.
- [7] Jamali, Mohsen, and Martin Ester. "Trustwalker: a random walk model for combining trust-based and item-based

recommendation." In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 397-406. ACM, 2009.

- [8] Sarwar, Badrul, George Karypis, Joseph Konstan, and John Riedl. *Application of dimensionality reduction in recommender system-a case study*. No. TR-00-043. Minnesota Univ Minneapolis Dept of Computer Science, 2000.
- [9] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 8 (2009): 30-37.
- [10] Salakhutdinov, Ruslan, and Andriy Mnih. "Probabilistic Matrix Factorization Advances in Neural Information Processing Systems 21 (NIPS 21)." *Vancouver, Canada* (2008).
- [11] Lipowski, Adam, and Dorota Lipowska. "Roulette-wheel selection via stochastic acceptance." *Physica A: Statistical Mechanics and its Applications* 391, no. 6 (2012): 2193-2196.
- [12] Milgram, Stanley. "The small world problem." *Psychology today* 2, no. 1 (1967): 60-67.

Mr. Pavan Kumar K N obtained his B.E. degree with distinction in Information Science and Engineering from Visvesvaraya Technological University. Presently he is taking up the position of Trainee Decision Scientist in Mu Sigma, Bangalore, India. His areas of interests include Data Analytics and Cyber Security.

Ms. Samhita S Balekai received her B.E degree in Information Science and Engineering from Visvesvaraya Technological University. She secured an offer for the position of Software Engineer in Accenture, Bangalore, India. Her areas of interests are Data Analytics, Social Networks, Data Warehousing and Mining.

Ms. Sanjana P Suryavamshi was awarded her B.E. degree with distinction in Information Science and Engineering from Visvesvaraya Technological University. Presently she is employed as a Software Engineer in Tata Consultancy Services (TCS), Bangalore, India. Her areas of interests are Networks and Cyber Security.

Ms. Sneha Sriram earned her B.E. degree in Information Science and Engineering from Visvesvaraya Technological University. She is pursuing her M.S. degree in Information Technology Management from University of Texas, Dallas. Her areas of interests are Enterprise Systems and Information Technology.

Dr. R. Bhakthavathsalam is presently working as a Principal Research Scientist in SERC, Indian Institute of Science, Bangalore. His areas of interests are Pervasive Computing and Communication, Wireless Networks and Electromagnetics with a special reference to exterior differential forms. Author held the position of Fellow of Jawaharlal Nehru Centre for Advanced Scientific Research during 1993 - 1995. He is a Member of IEEE Communication Society, ACM and CSI.