Combining Trust in Collaborative Filtering to Mitigate Data Sparsity and Cold-Start Problems

Vahid Faridani
Department of Computer Engineering
Mashhad Branch, Islamic Azad University
Mashhad, Iran
vahid.faridani@mshdiau.ac.ir

Majid Vafaei Jahan
Mashhad Branch, Islamic Azad University
Mashhad, Iran
vafaeijahan@mshdiau.ac.ir

Mehrdad Jalali
Mashhad Branch, Islamic Azad University
Mashhad, Iran
jalali@mshdiau.ac.ir

Abstract—**Collaborative filtering (CF) is the most popular approach to build recommender systems and has been successfully employed in many applications. However, it suffers from several inherent deficiencies such as *data sparsity* and *cold start*. To better show user preferences for the cold users additional information (e.g., trust) is often applied. We describe the stages based on which the ratings of an active user's trusted neighbors are incorporated to complement and represent the preferences of the active user. First, by discriminating between different users, we calculate the significance of each user to make recommendations. Then the trusted neighbors of the active user are identified and aggregated. Hence, a new rating profile can be formed to represent the preferences of the active user. In the next stage, similar users probed based on the new rating profile. Finally, recommendations are generated in the same way as the conventional CF with the difference that if a similar neighbor had not rated the target item, we will predict the value of the target item for this similar neighbor by using the ratings of her directly trusted neighbors and applying MoleTrust algorithm, so as to incorporate more similar users to generate prediction for this target item. Experimental results demonstrate that our method outperforms other counterparts both in terms of accuracy and coverage.**

Keywords-recommender systems; collaborative filtering; cold start; data sparsity; trusted neighbors

I. Introduction

One of the key innovations in on-line marketing is the creation of recommender systems for suggesting new interesting items to users (or buyers). In essence a recommender system (RS) tries to predict the ratings that users would give to different items, e.g. there are popular recommenders for movies[1], books[2], music[3], etc.

Collaborative filtering is the most popular approach to create recommender systems and has been successfully employed in many applications. However, it suffers from several inherent deficiencies such as *data sparsity* and *cold start*. Additional information from other sources have been studied and incorporated into CF including friendship [1], membership [2] and social trust [3] in order to solve these deficiencies and model user preferences more accurately. In this paper, trust is defined as one's belief toward others in providing accurate ratings relative to the preferences of the active user [4]. Both implicit trust (e.g., [5, 6]) and explicit trust (e.g., [7-10]) have been investigated in the literature. The former trust is inferred from user behaviors such as ratings whereas the latter is directly specified by users. By definition, the explicit trust tends to be more accurate and reliable than the implicit one. We focus on the explicit trust in this paper.

In this paper, we propose a novel trust-based approach called ''Combine'' by incorporating the trusted neighbors that are explicitly specified by the active users in the systems, aiming to improve the overall performance of recommendations and to mitigate the data sparsity and cold-start problems of CF. Specifically, we calculate the significance of each user to make recommendations. Then, by averaging the ratings on the commonly rated items according to the extent to which the trusted neighbors are similar to the active user and also according to the extent to which the trusted neighbors are significant to make recommendations, we incorporate the ratings of trusted neighbors of an active user. The set of new ratings is then used to represent the active user's preferences and to find similar users based on user similarity. Finally, recommendations are generated in the same way as the conventional CF with the difference that if a similar neighbor had not rated the target item, we will predict the value of the target item for this similar neighbor by using the ratings of her directly trusted neighbors and applying MoleTrust algorithm, so as to incorporate more similar users to generate prediction for this target item.

The rest of this paper is organized as follows. Section **II** gives a brief overview of related research on trust-based CF from which the research gap is identified and motivating our present work. The proposed approach is then elaborated in Section **III** where we also highlight the advantages of our method in principle. Experiments on Flixster data set are conducted in Section **IV** to verify the

[1] http://www.netflix.com.
[2] http://www.amazon.com.
[3] http://www.last.fm.

effectiveness of our method in predicting items' ratings, especially for the cold users. Finally, Section **V** concludes our work and outlines potential future research.

II. RELATED WORK

Many CF approaches have been proposed in the literature to solve the data sparsity and cold start problems. Generally, they can be classified into two categories: memory-based and model-based. The most well-known model is matrix factorization (MF) based approaches, such as SVD [11], NNMF [12] and tensor factorization [13]. Model-based approaches usually can achieve better accuracy and coverage than memory-based approaches. This is because the former ones will train a prediction model using global rating data whereas the latter concentrate on local rating information. However, model-based approaches cannot properly explain how the recommendations are generated and effectively adopt new ratings due to trained static models.

The closest approaches to ours are as follows. Reference [9] analyzes the drawbacks of conventional CF-based recommender systems, and elaborates the rationale why incorporating trust can mitigate those problems. They propose the MoleTrust algorithm, which performs depth-first search, to propagate and infer trust in the trust networks. Empirical results show that the coverage is significantly widened but the accuracy remains comparable when propagating trust. Similarly, [8] propose a breadth-first search method called TidalTrust to infer and compute trust value. Both approaches substitute similarity with trust to predict item ratings, and the performance of the two algorithms is close [14]. Hence, we will only compare our method with one of them, namely MoleTrust in this paper. In addition, [7] propose to enhance CF by predicting the ratings of similar users who did not rate the target items according to the ratings of their trusted neighbors, so as to incorporate more similar users for recommendation. However, it performs badly in cold conditions where only few ratings are available, which is the main concern of the present work. Another recent work using the explicit trust network is proposed by [10]. They improve the prediction accuracy by reconstructing the trust networks. More specifically, the trust links between two users will be removed if their similarity is lower than a threshold. Empirical results show that good performance is achieved at the cost of poor coverage, and it fails to function in cold conditions where user similarity may not be computable.

The purpose of our work is to take a step further in addressing the cold start and sparsity problems by proposing a novel approach to incorporate trusted neighbors in CF.

III. THE COMBINE METHOD

In this section, we will present the proposed Combine method the basic principle of which is to incorporate the ratings of trusted neighbors to complement and represent the preferences of active users.

A. Combining Process

For clarity, we introduce a number of notations to model the recommendation problem. Specifically, we denote the sets of all users, all items and all ratings as U; I and R, respectively. We keep the symbols u, v for the users and i, j for the items. Then $r_{u,i}$ represents a rating given by user u on item i, and takes a value in a certain rating scope, predefined by a recommender system. The predicted rating is denoted as $\hat{r}_{u,j}$. In a trust-aware recommender system, the active user u may have identified a set of trusted neighbors TN_u. For each trusted neighbor $v \in TN_u$, user u also specifies a trust value $t_{u,v} \in [0,1]$ indicating the extent to which user u believes in user v's ability in giving accurate ratings. For simplicity, the set of items rated by user u is denoted by $I_u = \{i | r_{u,i} \in R, i \in I\}$ and the set of users who rated item i is denoted by $U_i = \{u | r_{u,i} \in R, u \in U\}$. Hence, the recommendation problem can be re-described as: given a set of user ratings $(u, i, r_{u,i})$ and a set of user trust $(u, v, t_{u,v})$, predict a best prediction $(u, j, \hat{u}_j)$ for an active user u on a target item j.

1) Calculating the Significance of Each User

Reference [15] has shown that some users are more significant than other users to make recommendations. Suppose $W = \{1, \ldots, 5\}$ be the set of possible values for ratings. Let $V = \{4, 5\}$ be the subset of W with elements that are regarded as relevant ratings. Let $V^c = \{1, 2, 3\}$ be the subset of W with elements that are regarded as non-relevant ratings. Let $D_u = \{i \in I | r_{u,i} \in V\}$ be the set of items that user u has rated with a relevant value. Let $E_u = \{i \in I | r_{u,i} \in V^c\}$ be the set of items that user u has rated with a non-relevant value. To weight the importance of a rating, we use the following factors:

$$f_1 = 1 - \frac{\#D_u}{\#D_u + \#E_u}, \tag{1}$$

$$f_2 = \frac{\#D_u + \#E_u}{\#I_u}. \tag{2}$$

According to the factor f_1, the lower the number of relevant ratings made by user u, the higher the significance of this user's u relevant ratings. According to the factor f_2, the higher the number of ratings made by user u, the higher the significance of user u to make recommendations. Finally, we define the significance of a user to make recommendations as:

$$s_u = \left(1 - \frac{\#D_u}{\#D_u + \#E_u}\right)\left(\frac{\#D_u + \#E_u}{\#I_u}\right), \tag{3}$$

where $s_u \in [0,1]$ is the significance of user u to make recommendations.

2) Aggregating Trusted Neighbors

The cold users are generally defined as the users who have rated less than five items [9]. Since cold users usually are less active in the systems, they may not have a large number of trusted neighbors. We conduct experiments to show the statistics for cold users in Flixster data set, the specifications of which will be presented in Section **IV.A**. **Fig. 1** shows the distribution of trusted neighbors for the cold users in Flixster data set. Specifically, most cold users have only few trusted neighbors and only few cold users have identified many trusted neighbors.

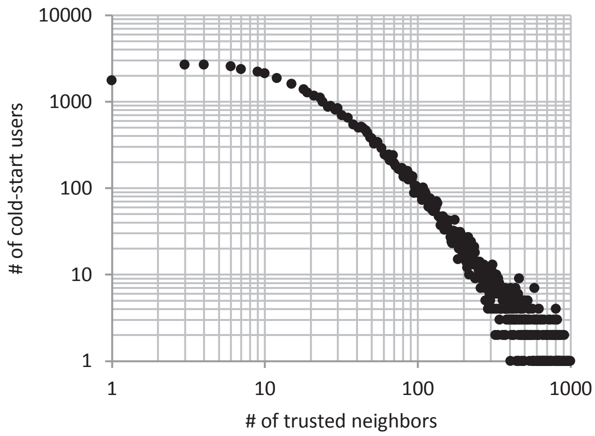

Figure 1. The distributions of trusted neighbors for the cold users on Flixster data set

Fortunately, trust can be propagated along with the web -of-trust. MoleTrust [9] and TidalTrust [8] are two typical algorithms to infer trust value. In this work, we adopt the MoleTrust to infer the trust value of indirectly connected users. Note that the trust value in the data set is binary, i.e., 0 or 1, where 0 means no direct trust connections whereas 1 indicates that a user directly connects with and trusts another user. As a result, the inferred trust value by the MoleTrust will be also binary, and thus we cannot distinguish trusted neighbors in a shorter distance with those in a longer distance. Hence, like [4] we adopt a weighting factor to devalue trust in a long distance:

$$t_{u,v} = \frac{1}{d} * t'_{u,v}, \tag{4}$$

where $t'_{u,v}$ denotes the inferred trust value by the MoleTrust algorithm, d is the shortest distance between users u and v determined by a breath first search algorithm, and $t_{u,v} \in [0,1]$ is the trust value that user u has towards another user v. In this work, we restrict $d \leq 3$ to prevent meaningless searching and save computational cost for large-scale data sets.

Hence, a set of users can be identified as trusted neighborhood for user u if the trust value of a user v is greater than a trust threshold:

$$TN_u = \{v | t_{u,v} > \theta_t, v \in U\}, \tag{5}$$

where θ_t is the trust threshold. Since the distance is restricted by $d \leq 3$, we presume that the all connected trusted neighbors are useful and hence set $\theta_t = 0$ for simplicity. In addition, we presume that user u will always believe in her own ratings as they are accurately reflecting her real preferences.

3) *Incorporating the Ratings of Trusted Neighbors*

After determining the trust neighborhood, a set of items can be identified as the candidate items for the combining process:

$$\tilde{I}_u = \{j | r_{v,j} \in R, v \in TN_u, j \in I\}. \tag{6}$$

That is, $\tilde{I}_u$ consists of items that have been rated by at least one trusted neighbor from the trust neighborhood. Then all the ratings of trusted neighbors on each item $j \in \tilde{I}_u$ will be incorporate into a single rating based on the weights of trusted neighbors:

$$\tilde{r}_{u,j} = \frac{\sum_{v \in TN_u} w_{u,v} r_{v,j}}{\sum_{v \in TN_u} |w_{u,v}|}, \tag{7}$$

where $\tilde{r}_{u,j}$ is the incorporated value for user u on item $j \in \tilde{I}_u$ based on the ratings of all the trusted neighbors, and $w_{u,v}$ denotes the importance weight of user v's ratings relative to the active user u. We claim that the importance weight $w_{u,v}$ is composed of four parts: significance value s_v, trust value $t_{u,v}$, rating similarity $sim_{u,v}$ and social similarity $j_{u,v}$. Hence, $w_{u,v}$ is computed as a linear combination of the four parts:

$$w_{u,v} = \alpha sim_{u,v} + \beta t_{u,v} + \gamma s_v + (1 - \alpha - \beta - \gamma) j_{u,v}, \tag{8}$$

where parameters α, β and γ indicate the extent to which the combination relies on rating similarity, trust value and significance value, respectively. The rationale behind this computation, i.e., incorporating four parts rather than trust value only, is that people trusting each other may not share similar preferences [16]. Specifically, it is possible that trusted neighbors have low similarity. In addition, all users do not have equal significance to make recommendations [15]. Reference [15] also noted that making recommendations based on more significant users will result in good accuracy and good coverage. Therefore, it is necessary to consider significance value, rating similarity and trust value.

Pearson correlation coefficient is often used to compute user similarity based on ratings:

$$sim_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \bar{r}_v)^2}}, \tag{9}$$

where $sim_{u,v} \in [0,1]$ is the similarity between two users u and v, and $I_{u,v} = I_u \cap I_v$ denotes the set of items rated by both users u and v. Considering the cases with positive trust, significance and social similarity but negative similarity may not make sense or be expected. Hence, we only consider the positively correlated users in this regard, i.e., $sim_{u,v} > 0$. Another reason is to be consistent with the value range of significance, trust and social similarity in **(3), (4) and (10)**.

The fourth component is the ratio of commonly trusted neighbors between two users u and v. The intuition is that two users are socially close if they share a number of trusted neighbors. The social similarity is defined as the ratio of shared trusted neighbors over all the trusted neighbors, and computed by the Jaccard Index:

$$j_{u,v} = \frac{|TN_u \cap TN_v|}{|TN_u \cup TN_v|}, \tag{10}$$

where $j_{u,v} \in [0, 1]$ indicates the social similarity of two users u and v based on their trusted neighbors. Hence, the importance weight $w_{u,v}$ can be computed using **(8)** since the four components are derived by **(3), (4), (9) and (10)**, respectively. In this way, all the ratings of trusted neighbors on a certain item can be incorporated into a single value by **(7)**.

To put it simply, the active user will keep all her own ratings, and the ratings of trusted neighbors will be used to complement her own preferences so that a new more complete and accurate rating profile can be formed and used to represent the preferences of the active user.

B. Incorporating with Collaborative Filtering

Given the new rating profile on the item set $\tilde{I}_u$ after the combining process in Section **III.A**, we then apply a conventional CF technique to predict the rating of a target item j that has not been rated by user u. More specifically, we first probe a set NN_u of similar users (i.e., nearest neighbors) for user u based on the similarity between user u and other users who either have rated item j or if They have not rated the target item j we can predict the rating of the target item j for them by using the ratings of their directly trusted neighbors and applying MoleTrust algorithm, so as to incorporate more similar users to generate prediction for this target item j.

In general, Pearson correlation coefficient (PCC) is often adopted to measure the similarity between two users u and v as $sim'_{u,v}$ according to their ratings on the items that they commonly rated (see **(9)**):

$$sim'_{u,v} = \frac{\sum_{i\in I_{u,v}}(\tilde{r}_{u,i}-\bar{r}_u)(r_{v,i}-\bar{r}_v)}{\sqrt{\sum_{i\in I_{u,v}}(\tilde{r}_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{i\in I_{u,v}}(r_{v,i}-\bar{r}_v)^2}}, \quad (11)$$

where $I_{u,v} = \tilde{I}_u \cap I_v$ denotes the set of items rated by both users u and v after combining process, $\bar{r}_u$ and $\bar{r}_v$ are the average ratings for users u and v respectively.

After computing user similarity, a group of similar users are then selected into the nearest neighborhood NN_u of the active user u. Herein we use the thresholding method, i.e., adopting the users whose similarity with the active user u is greater than a predefined threshold:

$$NN_u = \{v | sim'_{u,v} > \theta_s, v \in U\}, \quad (12)$$

where θ_s is a predefined similarity threshold.

Finally, all the ratings of nearest neighbors are aggregated to produce a prediction on a target item j that the active user u has not rated. We use the simple weighted average method, i.e., to compute the average value of all ratings provided by the nearest neighbors v weighted by their similarity $sim'_{u,v}$ with the active user u. Formally, the prediction is computed by:

$$\hat{r}_{u,j} = \frac{\sum_{v\in NN_u} sim'_{u,v} r_{v,j}}{\sum_{v\in NN_u} |sim'_{u,v}|}, \quad (13)$$

where $\hat{r}_{u,j}$ represents the predicted value on item j. We adopt the weighted average because the two most related works [9, 10] also take the same equation.

IV. EVALUATION

In order to verify the effectiveness of the Combine method, we conduct experiments on one real-world data set.

A. Data Acquisition

One real-world data set is used in our experiments, namely Flixster that contains both the data of explicit trust statements and user-item ratings. The specifications of this data set are summarized in **Table I**.

We sample a subset by randomly choosing 2108 users who issued 50044 item ratings and 415883 trust ratings. The rating sparsity is computed by:

$$Sparsity = \left(1 - \frac{\#Ratings}{\#Users \times \#Items}\right) \times 100\% \quad (14)$$

It is noted that the selected data set is highly sparse.

TABLE I. THE SPECIFICATIONS OF FLIXSTER DATA SET

Dataset	# Users	# Items	# Rates	# Trust	Sparsity
Flixster	2108	6356	50044	415883	99.63%

B. Experimental Settings

In the experiments, we compare the performance of our method Combine with a number of trust-based state-of-the-art methods as well as a conventional user-based CF method.

- **CF** computes user similarity using the PCC measure, selects the users whose similarity is above the predefined similarity thresholds for **(12)**, and uses their ratings to generate item predictions by **(13)**. In this work, the threshold θ_S is set 0 for all methods.
- **MT**x(x = 1, 2, 3) is the implementation of the MoleTrust algorithm [9] in which trust is propagated in the trust network with the length x. Only trusted neighbors are used to predict item ratings.
- **RN** denotes the approach proposed by [10] that predicts item ratings by reconstructing the trust networks. We adopt their best performance settings where the correlation threshold is 0.5, propagation length is 1, and the top 5 users with highest correlations are selected for rating predictions.
- **TCF**x(x = 1, 2) denotes the approach proposed by [7] that enhances CF by predicting the ratings of the similar users who did not rate the items according to the ratings of the similar users' trusted neighbors, so as to incorporate more users for recommendation. The best performance that they report is achieved when the prediction iteration x over trust network is 2. We adopt the same settings in our experiments.
- **Combine**x(x = 1, 2, 3) is our method with the trust propagation length x, aiming to investigate the

impact of trust propagation on the Combine method.

In addition, we split each data set into two different views as defined in [9]: the view of **All Users** represents that all users and their ratings will be tested whereas the view of **Cold Users** denotes that only the cold users who have rated less than five items, and their ratings will be tested in the experiments.

C. Evaluation Metrics

In each data view, users' ratings are hidden one by one in each iteration and then their values will be predicted by applying a certain method until all the testing ratings are covered. The errors between the predicated ratings and the ground truth are accumulated. The evaluation metrics are described as follows.

- Mean Absolute Error, or *MAE*, measures the degree to which a prediction is close to the ground truth:

$$MAE = \frac{\sum_u \sum_i |\hat{r}_{u,i} - r_{u,i}|}{N}, \quad (15)$$

where N is the number of testing ratings. Inspired by [17] who define a measure *precision* based on root mean square error (RMSE), we define the inverse MAE, or *iMAE* as the predictive accuracy normalized by the range of rating scales:

$$iMAE = 1 - \frac{MAE}{r_{max} - r_{min}}, \quad (16)$$

where r_{max} and r_{min} are the maximum and minimum rating scale defined by a recommender systems, respectively.

- Ratings Coverage, or *RC*, measures the degree to which the testing ratings can be predicted and covered relative to the whole testing ratings:

$$RC = \frac{M}{N}, \quad (17)$$

where M and N are the number of predictable and all the testing ratings, respectively.

- F-measure, or *F1*, measures the overall performance in considering both rating accuracy and coverage. According to [17], the F-measure is computed by:

$$F1 = \frac{2.iMAE.RC}{iMAE + RC}. \quad (18)$$

Hence the F-measure reflects the balance between accuracy and coverage.

D. Results and Analysis

In this section, we conduct a series of experiments on one real-world data set to demonstrate the effectiveness of our approach relative to others. Both data set views, namely All Users and Cold Users are tested. The results are presented in **Tables II** corresponding to the predictive performance on the Flixster.

1) Importance Weights with Parameters α, β and γ

An important step for the Combine method is to compute the importance weights of trusted neighbors which is a linear combination of rating similarity, trust value, significance value and social similarity with parameters α, β and γ (see **(8)**). The experiments show that the settings of (α, β, γ) are (0.4, 0.3, 0.2) on Flixster.

2) Trust Propagation in Different Lengths

We investigate the influence of trust propagation on the performance of the Combine method. Compared with Combine1, Combine2 and Combine3 have a better accuracy and coverage. Our work does not improve the accuracy for $d > 3$, hence we can conclude the optimum value for *d* is 3.

3) Comparsion with Other Methods

For other methods, we obtain different results on Flixster as shown in **Table II**. More specifically, CF cannot achieve large portion of predictable items. It confirms that CF suffers from cold start severely. Unlike our imagination the RN method accomplishes bad accuracy, and also covers the smallest portion of items, since only the ratings of the users who have a large number of trusted neighbors and high rating correlations are possible to be predicted. Comparing with CF, all other methods except of MT1 and MT2 achieve better performance for cold users. When only direct trusted neighbors are used (MT1, Combine1), our method achieves better accuracy and coverage. When trust is propagated in longer length, both accuracy and coverage are increased. Nevertheless, our method outperforms MTx in all propagation lengths. TCF methods generally obtain better coverage in the view of All Users. However, for cold users, TCF functions badly due to the limitation that it relies on CF to find similar users before it can apply trust information on them. We further compute the percentage of improvements that each method obtains comparing with the CF in terms of F1. Formally, it is computed by[4]:

$$Improvment = \frac{Method.F1 - CF.F1}{CF.F1} \times 100\% \quad (19)$$

where Method refers to any one of the methods tested in our experiments except the CF approach, whose F1 performance is regarded as a reference. The results are shown in **Table III** both in the view of All Users and in the view of Cold Users. A conclusion that can be drawn from the results in **Table III** is that our method consistently outperforms the others, and significantly improves the performance of traditional collaborative filtering.

V. CONCLUSION AND FUTURE WORK

This paper proposed a novel method to incorporate trusted neighbors into traditional collaborative filtering techniques, aiming to resolve the data sparsity and cold start problems from which traditional recommender systems suffer. Specifically, the ratings of trusted neighbors were incorporated to complement and represent the preferences of the active users, based on which similar

---

[4] The formula can be referred to as the relative change defined in http://en.wikipedia.org/wiki/Relative_change_and_difference.

TABLE II. THE PREDICTIVE PERFORMANCE ON THE FLIXSTER DATA SET

Views	Approaches Measured by MAE, RC and F1									
	CF	***MT1***	***MT2***	***MT3***	***RN***	***TCF1***	***TCF2***	***Combine1***	***Combine2***	***Combine3***
All	0.786	1.079	1.008	0.940	0.820	0.795	0.781	0.744	0.735	0.711
Users	35.53%	2.77%	20.54%	48.71%	0.30%	51.29%	62.59%	56.35%	64.84%	69.60%
	0.4967	0.0535	0.3284	0.6029	0.0061	0.6321	0.7122	0.6728	0.7266	0.7621
Cold	0.813	1.162	0.998	0.964	0.890	0.822	0.829	0.815	0.813	0.808
Users	19.25%	2.29%	18.70%	46.59%	0.13%	29.08%	39.99%	33.67%	40.97%	47.63%
	0.3117	0.0445	0.3016	0.5850	0.0027	0.4290	0.5367	0.4772	0.5462	0.6027

TABLE III. THE IMPROVEMENTS OF ALL METHODS COMPARING WITH CF IN F1

Dataset	Views	MTx (%)	RN (%)	TCFx (%)	Combinex (%)
Flixster	All Users	21.38	-98.77	43.39	53.43
	Cold Users	87.68	-99.13	72.18	93.36

users can be identified and recommendations are generated. The prediction of a given item is generated by averaging the ratings of similar users weighted by their importance. Experiments on one real-world data set were conducted and the results showed that significant improvements against other methods were obtained both in accuracy and coverage as well as the overall performance. Further, by propagating trust in the trust networks, even better predictive performance can be achieved. In conclusion, we proposed a new way to better apply trust, similarity and significance to improve the performance of collaborative filtering.

The present work depends on the explicit trust during the combining process. For future work, we intend to infer implicit trust from user behaviors, and enhance the generality of the Combine method.

REFERENCES

[1] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," presented at the Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, Boston, MA, USA, 2009.

[2] Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, and W. Zheng, "Augmenting collaborative recommender by fusing explicit social relationships," *Workshop on Recommender Systems and the Social Web, Recsys 2009,* 2009.

[3] A. Jøsang, W. Quattrociocchi, and D. Karabeg, "Taste and Trust," in *Trust Management V*. vol. 358, I. Wakeman, E. Gudes, C. Jensen, and J. Crampton, Eds., ed: Springer Berlin Heidelberg, 2011, pp. 312-322.

[4] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Know.-Based Syst.,* vol. 57, pp. 57-68, 2014.

[5] J. O'Donovan and B. Smyth, "Trust in recommender systems," presented at the Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, California, USA, 2005.

[6] A. Seth, J. Zhang, and R. Cohen, "Bayesian Credibility Modeling for Personalized Recommendation in Participatory Media," in *User Modeling, Adaptation, and Personalization*. vol. 6075, P. De Bra, A. Kobsa, and D. Chin, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 279-290.

[7] M. Chowdhury, A. Thomo, and W. W. Wadge, "Trust-Based Infinitesimals for Enhanced Collaborative Filtering," in *COMAD*, 2009.

[8] J. A. Golbeck, "Computing and applying trust in web-based social networks," University of Maryland at College Park, 2005.

[9] P. Massa and P. Avesani, "Trust-aware recommender systems," presented at the Proceedings of the 2007 ACM conference on Recommender systems, Minneapolis, MN, USA, 2007.

[10] S. Ray and A. Mahanti, "Improving Prediction Accuracy in Trust-Aware Recommender Systems," presented at the Proceedings of the 2010 43rd Hawaii International Conference on System Sciences, 2010.

[11] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer,* vol. 42, pp. 30-37, 2009.

[12] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from Incomplete Ratings Using Non-negative Matrix Factorization," in *SDM*, 2006.

[13] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver, "TFMAP: optimizing MAP for top-n context-aware recommendation," presented at the Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, Portland, Oregon, USA, 2012.

[14] P. Victor, C. Cornelis, M. D. Cock, and A. M. Teredesai, "Trust- and Distrust-Based Recommendations for Controversial Reviews," *Intelligent Systems, IEEE,* vol. 26, pp. 48-55, 2011.

[15] J. Bobadilla, A. Hernando, F. Ortega and A. Gutierrez, "Collaborative filtering based on significances," *Inf. Sci.,* vol. 185, pp. 1-17, 2012.

[16] P. Singla and M. Richardson, "Yes, there is a correlation: - from social networks to personal behavior on the web," presented at the Proceedings of the 17th international conference on World Wide Web, Beijing, China, 2008.

[17] M. Jamali and M. Ester, "TrustWalker: a random walk model for combining trust-based and item-based recommendation," presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France, 2009.