A New Trust Evaluation for Trust-based RS

Sajjawat Charoenrien
Department of Mathematics
Chulalongkorn University
Bangkok, Thailand

Saranya Maneeroj
Department of Mathematics
Chulalongkorn University
Bangkok, Thailand

***Abstract*—Trust-based recommender systems provide the recommendations on the most suitable items for the individual users by using the trust values from their trusted friends. Usually, the trust values are obtained directly from the users, or by calculated using the similarity values between the pair of users. However, the current trust value evaluation can cause the following three problems. First, it is difficult to identify the co-rated items for calculating the similarity values between the users. Second, the current trust value evaluation still has symmetry property which makes the same trust value on both directions (trustor and trustee). Finally, the current trust value evaluation does not focus on how to adjust the trust values for the remote user. To eliminate all of these problems, our purposed method consists of three new factors. First, the similarity values between the users are calculated using a latent factor model instead of the co-rated items. Second, in order to identify the trustworthiness for every user in trust network, the degrees of reliability are calculated. Finally, we use the number of hops for adjusting the trust value for the remote users who are expected to be low trust as shown in the real-world application concept. This trust evaluation leads to better predicted rating and getting more predictable ratings. Consequently, from our experiment, the more efficiency trust-based recommender system is obtained, comparing with the classical method on both accuracy and coverage.**

Keywords—trust-based recommender systems; trust values; similarity values

I. INTRODUCTION

Recommender systems (RS) act as the tools that help selecting the most relevant items to the target users. First, the users' preference ratings on items are collected. After that, the similarities between users are calculated by using the ratings on their co-rated items. These similarities are then applied to select the nearest neighbors for each user. Finally, recommender systems predict the ratings for a target user by using the ratings from his neighbors. However, usually most users tend to provide the small fraction of all possible ratings. This leads to a sparsity problem which the system cannot provide the accurate prediction and, for some items, unpredictable.

In order to solve this problem, Trust-Awared Recommender systems (TARS) [3] have been implemented. The system uses not only the users' ratings data, but also the trust information for prediction. These trust values are usually collects directly from users or calculated as the same way as similarity values. However, calculating trust values this way has cause many problems. First, the similarity value has symmetry property which makes the trust value on both directions of the pair of users to be the same.

Actually, the trust value between two persons might not be the same. For example, user A might trust user B but user B might not trust user A. Therefore, the trust between these two should not be the same. Second, the systems may not be able to effectively find the trust values due to the sparsity problem. For the last, it cannot use the transitive property of graph to calculate the trust value between the friends of friends for a user. TidalTrust [1][4] and MoleTrust [5] were proposed to solve this problem. Both of them use the propagation technique to propagate the trust values for the raters (who rated score on the target item). The propagated trust values are calculated every time when the friends of friends of the target users are visited. However, finding the propagation in a very large trust network is time comsuming. Thus, another model has been proposed by Y. Guo. [9] This model tries to find the trust factor of the raters toward the ratings.

The trust factor is used to calculate the trust value as the important factor. The trust factor calculates from the number of friends and number of evaluated items belonging to that rater and the experience of rater on the past rated item, with the rater. That means the trust values are calculated from the relations between the raters and target users. However, this method does not concern the number of the hops, which might reduce the trust values of the remoteness users. While exploring the trust network, a target user has to visit the friends of his friends until he retrieves the wanted rating. Every time of the visiting, the number of hops is increased. If the number of hops is large, the two users are far from each other and they are less related. Therefore, the number of hops should be included in the model to suit the real world applications.

In this work, a new trust evaluation method is proposed. This method has three factors for calculating the trust value of each rater, e.g. similarity value between a pair of user, trustworthiness and number of hops. By using the latent feature model for calculating similarity value for each pair of users, the sparsity and symmetry problems can be solved. The second factor, trust worthiness is the extended idea from Y. GUO, which tries to find the degree of reliability of each rater in the trust.

Finally, the number of hops is used to adjust the trust values of the raters based on the distance from the target users in the trust network. These three factors are combined in this work to make a new trust value calculation. After that, the new trust value is used in rating prediction. It improves the quality of the prediction than the classical trust-based recommender systems on both accuracy and coverage as shown in the experiment in the fourth section of this paper.

II. Related Works

Recommender system involves three major steps. First, the system collects rating data representing the user's preference towards the different items. Second, it generates the user's pattern based on his past experience towards those items. Finally, RS makes the prediction for the new items based on the user's preference pattern. However, usually, the numbers of ratings are not large enough to give an effective prediction. This problem is called sparsity problem, which leads to inaccurate rating prediction or unpredictable for some items. After that, the study of Swearingen & Sinha [7] on the usability of three book RS and three movie RS has found that, by integrating trust information into the RS, the prediction is improved. Moreover, Ziegler & Golbeck [8] investigated the correlations between trust and similarity definition. They found that the trust values between each pair of user in the RS were be calculated by using their similarity values on the co-rated items (the overlap items that have been rated from both users). However, it still leads to two problems. First, the trust value from similarity value is symmetry on the both side of the users, such as A → B equals to B → A. In fact, two users who trust each other might not have the same trust value because it opposes to the fact which two persons not necessary to trust each other. One person can trust the other by one side. Another problem is that sometimes, the system cannot calculate the similarity values for some pairs of users because they do not have the co-rated items. To solve this, J. O' Donovan [6] proposed the work called Trust in Recommender Systems. In their work, the trust values can be found from the reliabilities of the raters which are indicated the amount of corrected ratings they have made. The trust value is the ratio between the number of correctly predicted ratings and the whole number of the predicted ratings as (1).

$$Trust^P = \frac{|CorrectSet(p)|}{|RecSet(p)|} \qquad (1)$$

Where *p* is rater, *CorrectSet(p)* is the set of correctly predicted items , and *RecSet(p)* is the set of all predicted items. Although the trust values calculated by J. O' Donovan can solved the symmetry problem, it did not use information in trust network which contains the relationships among the users to make more accurate prediction. Trust network can be represented as a graph that consists of nodes (as the user) and edges (as the relationship between the users).

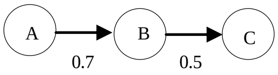

Fig. 1. The sample of Trust network

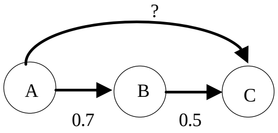

Fig. 2. Example of a figure caption. (*figure caption*)

The weight on each edge shows the trust value from source user to destination users in as Fig. 1. Sometimes, when the friends of the target user cannot provide the prediction for him, the system might use the opinions of the friends of that user's friends instead. The opinions from the target user's friends of friends can be transitively obtained from their directed friends. However, the target user might not have directed trust to these friends of friends. The example of this is shown in Fig. 2, if user A is a friend with user B and user B is a friend with user C, then how much user A trusts user C? Messa [3] proposed TAR architecture to provide trust value calculation for friends of friends.

From the TAR architecture, the predicted ratings are generated by rating predictor module using output from both User Similarity and Estimated Trust modules. The User Similarity is the process which calculates the similarity value between a pair of users.

While the Estimated Trust finds the trust values on friends of friends, which can solve the previous problem. TidalTrust and MoleTrust are the traditional methods that used to find the trust values from friends of friends for the target users. Both of them use transitive property of the graph to propagate the trust value from friends to friends. They use depth first search to propagate trust value from the target user to the rater. In the propagation process, the trust values are calculated by using the trust values of the previous friends as shown in (2).

$$trust(u) = \frac{\sum_{i \in predecessors}(trust(i) * trust\_edge(i,u))}{\sum_{i \in predecessors}(trust(i))} \qquad (2)$$

Where *u* is target user, *i* is the user whom the target user trusted, *trust(i)* is the trust value of user *i*, *trust_edge(i,u)* is the trust between user *i* and user *u* and predecessors are the previous friends of the user.

The different between their methods is that, in MoleTrust the cycle from the trust network is removed to reduce the distance of the trust propagation, in order to improve the performance. In contrast, TidalTrust propagates to every node in the trust network that is connected.

On the very large trust network, the propagation technique is not good because the exploration on every node on the graph takes a large amount of time, depending on the complexity of that the trust network.

Y. Guo [9] proposed trust value calculation without propagation. This method finds the trust factor which is the main factor of the trust factor. It calculates from the combination of the number of friends and number of evaluated items. If a user has the number of friends more than the others, he is more reliable.

$$T_i = \frac{2(1-1/\ln f_i)*(1-1/\ln(q_i+3))}{2-1/\ln f_i - 1/\ln(q_i+3)} \qquad (3)$$

Where *i* is the user whom the target user trusted, T_i is the trust factor of user *i*, f_i is the number of user *i*'s evaluation on each item and q_i is the number of recommendations user *i* has made for the others.

In the real world application, when considering the transitive property, a friend who is far away from target user should have lesser trust value than close friend. While exploring the trust network, the target user will visit the friend of friend until retrieve the rating. Every time of the visiting, the number of hops is increased every time. If the number of hops is large, the two users are far from each other and they are less related. Therefore, the number of hops should be included into the model to suit the real world applications.

In this work, a new trust value calculation that solves the problems mentioned above is proposed. First, this method uses the latent features of users for finding similarity instead of using co-rated items. Second, the symmetry problem is eliminated by exploiting the degree of reliability of each user. And, finally, this method considers the distance among friends by using the number of hops.

Therefore, we can summarize pros and cons of related trust methods comparing with our proposed method on three attributes: symmetry or asymmetry of initial trust value, transitive opinion from the directed friends to the friend of the directed friends and concerning the number of hops of remoteness users as shown in the following Table I.

TABLE I. PROS AND CONS COMPARISON OF EACH METHOD

Method	Initial Trust Value	Transitive Opinion	Concern the number of hops
J. O'Donovan's Method	Asymmetry	No	No
TidalTrust	Symmetry	Yes	No
MoleTrust	Symmetry	Yes	No
Y Guo's Method	Asymmetry	Yes	No
Proposed Method	Asymmetry	Yes	Yes

III. PROPOSE METHOD

The new trust value calculation proposed in this work consists of three main steps. First, the similarity values for all pairs of users are calculated. Then, then reliability concept is combined into trust value calculation in order to get rid of symmetry problem. Finally, the number of hops is included in the calculation to reduce the trust of remote friends.

To guarantee that the trust values can be calculated for every pair of users, the singular value decomposition (SVD) [2] is applied in this work instead of relying on the co-rated items. First, the latent features of the user are extracted by the following.

$$R=USV^t \tag{4}$$

Where R is the user-rating matrix, U is the user matrix, S is the reduced matrix and V^t is the transpose matrix of the item matrix. From matrix U, each row is represented as the user feature vector. Each feature vector of user contains the latent features representing the user's characteristic. The cosine similarity is then applied on these feature vectors to find the similarity for every pair of users by the following.

$$sim_{A,B} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{5}$$

Where A_i is the i^{th} latent feature of user A (target user), B_i is the i^{th} latent feature of user B and n is the number of latent features.

To prevent the symmetry problem of the trust values on both directions of the pair of users, the system uses not only the similarity value between a pair of users (sim (A,B)) but also concerns the reliability of rater(B) in the network. That is, reliability concept by Y. Guo [9] is applied in this work by using the number of in-degree which is represented the number of friends who trust rater (B).

$$trustworthiness_B =$$

$$\frac{ln\,(n_{in,B}+e)}{ln\,(\max(\{n_{in,C} | C \in \{user\ in\ Trust\ Network\}\})+e)} \tag{6}$$

Where e is a natural number, $n_{in,B}$ is the number of the in-degree edges of the user B and

$\max(\{n_{in,C} | C \in \{user\ in\ Trust\ Network\}\}$ is the maximum number of the in-degree edge of the user in the trust network. Also, the confidence of B towards A is calculated by merging $sim(A,B)$ and $trustworthiness_B$ using harmonic mean as shown in (7).

$$confidence_B = \frac{2\ \times trustworthiness_B \times sim_{A,B}}{trustworthiness_B + sim_{A,B}} \tag{7}$$

Moreover, to make the method suitable for the real world situation, the confidence level of user who is far from target user should be adjusted by using number of hops as shown in (8).

$$trust_{A\to B} = confidence_B \times \frac{1}{d_B} \tag{8}$$

Where d is the number of hops from user A to B

After gathering the trust values of all raters from previous steps, the predicted rating can be calculated by using trust values of target user to all the raters who have rated the target item as weights.

$$predict\ rating_t = \frac{\sum_{i \in Rater} rating(i) * trust_{t\to i}}{\sum_{i \in Rater} trust_{t\to i}} \tag{9}$$

Where t is target user, $rating(i)$ is an actual rating of rater(i) on the target item and $trust_{t\to i}$ is trust value of target user(t) toward rater (i).

IV. EXPERIMENT

In order to evaluate the performance of our proposed method, we compares our work with 2 classical trusted based RS methods: TidalTrust and MoleTrust

A. Dataset

The Epinions dataset is used in this work. This dataset consists of user-rating data containing 664,824 reviews from 49,290 users on 139,738 items, and also the trust network data containing 487,181 issued trust statements.

B. Evaluation Metric

To evaluate the model, we use two measurements: RMSE and coverage.

1) RMSE (Root Mean Square Error): It represents the accuracy of the prediction. The lower the value means the method is more accurate that the method with higher RMSE. This value can be calculated by the following

$$RMSE=\sqrt{\frac{\sum_{(u,i)|R_{u,i}}\left(r_{u,i}\text{-}\hat{r}_{u,i}\right)^{2}}{|\{(u,i)|R_{u,i}\}|}} \qquad (10)$$

Where $r_{u,i}$ is an actual rating of user(u) on target item(i), $\hat{r}_{u,i}$ is a predicted rating of user(u) on target item(i) and $\{(u,i)|R_{u,i}\}$ is the set of user who rated on item(i).

2) Prediction Coverage: This value indicates the fraction of the ratings that can be predicted from all of the rating available.

$$coverage=\frac{|\{(u,i)|\hat{r}_{u,i}\}|}{|\{(u,i)|r_{u,i}\}|} \qquad (11)$$

Where $|\{(u,i)|\hat{r}_{u,i}\}|$ is the number of ratings that can be predictable and $|\{(u,i)|r_{u,i}\}|$ is number of all ratings. The higher coverage means the system provides more predictable rating.

C. Experimental Results

In the experiment, we compare TidalTrust and MoleTrust with the proposed method by using the dataset mentioned above. To predict the rating for the target item, we use the leave one out technique (hide only the actual rating on the target item of the target user in the dataset, and uses the rest for prediction) . Instead of using all ratings in the dataset, we randomly select one target item per target user and only the first 5,000 users are use as the test set. To avoid the bias, we use the same random data on TidalTrust, MoleTrust and proposed methods. The results of the experiment are shown as Fig. 3.

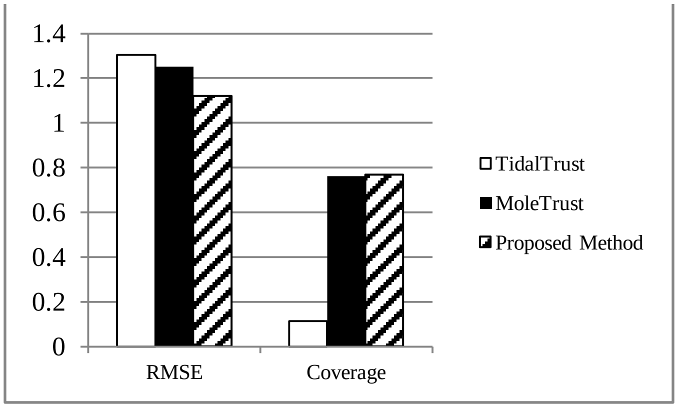

Fig. 3. Comparison of RMSE and Coverage between the proposed method and others

From the results shown above, RMSE of our proposed method is lower than TidalTrust and MoleTrust. It can be concluded that proposed method provides better accuracy. In the aspect of prediction coverage, our proposed method provides higher coverage values than both TidalTrust and MoleTrust. It can be concluded that proposed method provides more predictable items.

V. Disscusion

A. Accuracy

The reasons that proposed method has better accuracy than both TidalTrust and MoleTrust are following by this:

Both current trust-based RSs calculate the trust values by using the co-rated items which is hard to be identified because of the sparsity problem. The small number of co-rated items leads to the low quality neighbor which causes the low accuracy in prediction. On the other hand, the purpose method uses latent features of user instead of co-rated items. About 5,000 user features are extracted from user-rating matrix, so, it can represent user characteristic more correctly.

The current trust-based RSs do not consider the degree of reliability for each user in trust network. In the proposed method, the ratio between the numbers of friends on each specified user to the maximum number of friends from all users in the trust network is used to determine the reliability.

The current trust based RSs do not concern the remoteness of the friends of the target users. This type of friends should have the lesser trust value comparing to the close friends. The proposed method deals with this by adjusting the trust values using the number of hops.

B. Coverage

The reason that proposed method can provide more prediction coverage than the other two methods is that the proposed uses the latent features of user for calculating the similarity values without using co-rated items. Therefore, it can calculate similarity values for all pairs of users. However, it cannot provide 100% coverage because the target item obtains the rating only from the target user, not from other users. Therefore, there is no rating from rater for the prediction step. However, this case occurs not only in our proposed method, but also in current trust based RS such as MoleTrust and TidalTrust as well.

VI. CONCLUSION

A new trust evaluation calculation is proposed in this work. It consists of three new factors. First, the similarity values between the users are calculated using a latent factor model instead of the co-rated items. Second, in order to identify the trustworthiness for every user in trust network, the degrees of reliability are calculated. Finally, we use the number of hops for adjusting the trust value for the remote users who are expected to be low trust as shown in the real-world application concept. From the experiment results, our proposed method is more efficiency than the classical trust based RS (MoleTrust and TidalTrust) on both accuracy and coverage.

VII. FUTURE WORK

Usually, Recommender System calculates a prediction by using opinion of the directed friends who rated the target item. However, in the trust-based recommender system, the rater who rated the target item may not be the directed friend of the target user. Therefore, there should be the way to translate the rater’s opinion into directed friend’s opinion. This is an objective of our near-future work.

REFERENCES

[1] J. Golbeck, “Computing and Applying Trust in Web-based Social Networks”, PhD thesis, University of Maryland College Park, 2005.

[2] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Application of Dimensionality Reduction in Recommender Systems: a Case Study”, ACM WebKDD 2000 Workshop, 2000.

[3] P. Massa and P. Avesani, “Trust-aware recommender systems”, In ACM Recommender Systems Conference (RecSys), USA, 2007.

[4] J. Golbeck, “Personalizing Applications through Integration of Inferred Trust Values in Semantic Web-Based Social Networks”, Proceedings of Semantic Network Analysis Workshop, Galway, Ireland, 2005.

[5] P. Massa and P. Avesani, “Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers”, International Journal on Semantic Web and Information Systems (IJSWIS), 3(1), 2007.

[6] J. O'Donovan , B. Smyth, “Trust in recommender systems”, Proceedings of the 10th international conference on Intelligent user interfaces, San Diego, California, USA , January 10-13, 2005.

[7] R. Sinha and K. Swearingen, “Comparing recommendations made by online systems and friends.”, In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries. Dublin, Ireland, 2001

[8] C. Ziegler and J. Golbeck, "Investigating interactions of trust and interest similarity", Decision Support Systems, v.43 n.2, p.460-475, March, 2007

[9] Y. Guo, X. Cheng, D. Dong and C. L. Rishuang Wang, "An Improved Collaborative Filtering Algorithm Based on Trust in E-Commerce Recommendation Systems," research supported by a grant from the Chinese National Science Foundation Key Project, IEEE 2010, pp 1-4