TrusTem: Recommendation in Disjoint Trust Networks

Syed Tanveer Jishan
Presentation ID 9
Student ID 301279717
sjishan@sfu.ca

Khaled Diab
Presentation ID 5
Student ID 301235653
kdiab@sfu.ca

Abstract

Collaborative filtering (CF) based recommender systems suggest items to users by employing a similarity metric. With the introduction of online social networks, graph-based approach of recommendation has emerged. This approach assumes trust among users and recommends items based on trust. Although it solves cold start problem in recommendation system, user-item coverage decreases. We argue that one of the causes of coverage issue is the disjointedness of trust network, therefore, an item not rated by anyone in a subgraph cannot be recommended to any user in that subgraph. To solve this problem, we propose TrusTem: a recommendation algorithm that merges user-item and trust domains. We have conducted experiments on real-life dataset and compared TrusTem against state-of-art algorithms. Our experiments demonstrate that TrusTem covers 99.8% of user-item pairs, at cost of reducing accuracy up to 8%. However, we show numerically that TrusTem does better coverage-accuracy trade-off than these algorithms.

I. Introduction

Personalized recommender systems predict items ratings to users based on their items preferences. With the increase in popularity of online social networks, the trust-based approach to recommendation has emerged[?,?,?,?]. In such approach, the friendship/trust network among users is taken into consideration to make recommendations. Trust information can be exploited to get rid of cold start problem, since it is assumed by the sociologists that friends (or people you trust) will have similar attributes, hence similar preferences [1]. However, recommending items that are only rated by direct trusted users can lead to item coverage issue. To put it simply, the items a target user may like but not rated by his friends will not be recommended. Therefore, recommender systems take advantage of the transitive property of the trust [2], and use trust propagation metrics to estimate trust between indirect neighbors so that more items can be covered for recommendation.

We argue that despite propagating trust in the trust network, there are still hindrance in the coverage of the items to be recommended, since trust networks are usually a set of *disjoint* subgraphs rather than one large fully connected graph. To verify this claim, we looked into several real life datasets. For instance, we have collected trust information among $7,073$ users in Epinions.com and found that it consists of 17 disjoint subgraphs. Clearly, trust cannot be propagated from one such subgraph to another, since there exist no edges to connect them. In such cases, if an item is not rated by any of the user within a subgraph, then it cannot be recommended to the target user belonging to that subgraph. Therefore, the issue of item coverage will persist.

In order to propagate trust among these disjoint subgraphs, we propose a new recommender algorithm: TrusTem. TrusTem takes advantage of the transitive property of matrix factorization technique [3]. It considers trust to be in the same domain as the item ratings which results in a hybrid trust based recommender system. In such hybrid recommender system, trust propagation and ratings prediction happen simultaneously. We list the contributions of this report as follows:

- We define a new research problem in trust-based recommendation system. That is, recommendation in disjoint trust networks.
- We propose a new solution (TrusTem). Our solution model is based on the idea of merging the rating and trust domains.
- We evaluate TrusTem using real dataset against stat-of-art trust-based recommendation algorithms. We show that TrusTem covers 99.8% of user-item pairs, at cost of reducing accuracy up to 8%.

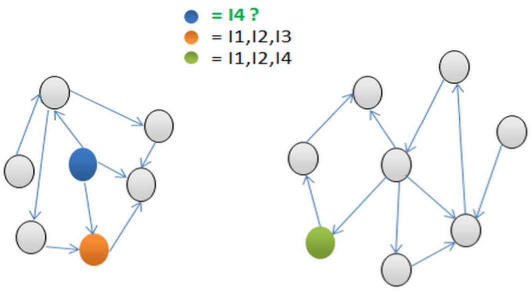


Fig. 1: Example of trust network disjointedness. The trust network consists of two subgraphs, item I4 can not be recommended to the blue node user.

The report is structured as follows. Section 2 surveys trust-based recommenders systems while Section 3 defines the problem of recommendation in disjoint trust networks. Section 4 describes our solution model. We evaluate TruTem in Section 5, before concluding the report in Section 6.

II. Related Work

MoleTrust [?] considers rating information up to certain depth threshold. In order to propagate trust, a backward exploration is performed. It means that the trust value from user A to user B is the aggregation of trust values between the user A and every other users who directly trust B. Similar to MoleTrust, TidalTrust [?] performs graph search in the trust network to estimate trust value between two indirect neighbours. To infer the trust between two indirect neighbours, TidalTrust first finds the shortest path in the graph from a user A to every other user, then picks a user B before aggregating the trust value from every users to the user B who are direct neighbour of A. Jamali and Ester proposed TrustWalker [?], a random walk model that combines trust-based approach and item-based CF approach. Random walks are made on the trust network to infer items ratings. If the item is not found within a certain depth threshold, item rating is estimated based on similar items. SocialMF [?] is a variant of trust-based recommender system which uses probabilistic matrix factorization approach. The latent features of this model is based on the trust information. The latent feature vector of a particular user depends on the latent feature vectors of all their direct neighbours, which propagates trust recursively.

To best of our knowledge, *trust network disjointedness* is not handled previously by any recommendation algorithm. This report defines this problem and proposes a new solution model and algorithm.

III. Problem Definition

In order to understand the coverage issue and motivate the reader, we consider the following example. Consider a trust information of 15 users in an online social network. Using the trust information we construct a trust network shown in Figure 1. User with orange node has rated items I1, I2, I3 and user with green node has rated items I1, I2, I4. Now if none of the users in the left subgraph rated item I4, then we are unable to recommend that particular item (I4) to the blue node. This is an example of item coverage issue due to disjoint subgraphs. On the other hand, we observe that the trust network disjointedness exists in many real datasets. For example, the trust network of Epinions.com dataset has 17 subgraphs. Also, FilmTrust dataset consists of five subgraphs. Now, we define the disjointedness problem formally.

Consider a recommender system with users $U = \{u_1, u_2, \ldots, u_N\}$, and items $I = \{i_1, i_2, \ldots, i_M\}$, where N and M are the users and items counts respectively. Each user $u \in U$ rated an item $i \in I$ with value $r_{u,i}$. We consider ratings with discrete values belonging to a finite set (e.g. $\{1, 2, \ldots, 5\}$). Also, each user u trusts a set of users $T(u) \subseteq U$. This trust information can be acquired explicitly (e.g. a user declares their trust), or implicitly (e.g. based on mutual activities). In this report, we consider a general trust values (e.g. not necessarily a binary). Given the trust information, we construct a directed trust network T, where nodes are the users, and weighted edges represent the trust information. Let the trust network T consists of multiple subgraphs $T_1, T_2, \ldots, T_d$ such that: $T = \cup_i^d \ T_i$ and $\cap_i^d \ T_i = \phi$. That is, the original trust network T consists of disjoint trust networks, where trust information is not propagated across them. This *typical* trust network structure reduces the recommendation

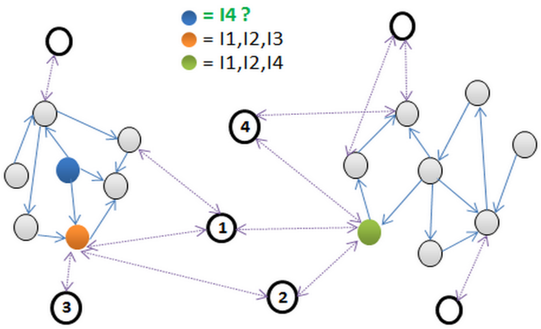
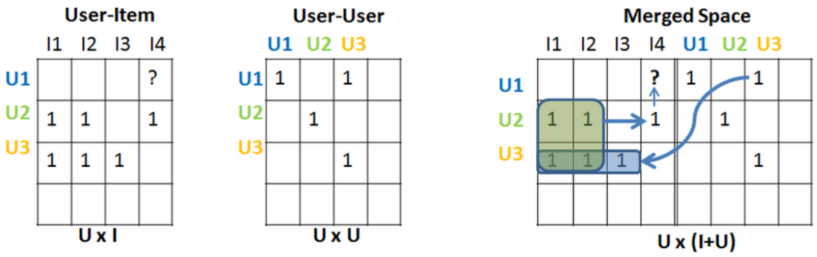


Fig. 2: Bridging the disjoint trust networks by using items nodes between the subgraphs.



User-Item

	I1	I2	I3	I4
U1				?
U2	1	1		1
U3	1	1	1	

U x I

User-User

	U1	U2	U3	
U1	1		1	
U2		1		
U3			1	

U x U

Merged Space

	I1	I2	I3	I4	U1	U2	U3	
U1				?	1		1	
U2	1	1		1		1		
U3	1	1	1				1	

U x (I+U)

Fig. 3: Merging the rating and trust domains in TrusTem.

coverage. Also, there is an inherent trade-off between the recommendation coverage and accuracy. For example, recommender systems can favor accuracy for coverage or vice versa, and such disjoint trust network will reduce coverage even more. Our *goal* is to predict unknown item ratings $\widehat{r_{u,i}}$ to users in such a disjoint trust network, such that the coverage is maximized without significantly reducing the accuracy.

IV. TrusTem: Recommender System in Disjoint Trust Networks

A. Graph-based Model

As depicted in Figure 1, orange and blue nodes users have similar item preferences even though they belong to different subgraphs. One possible interpretation is that these two users have similar item preferences but they do not know each other. Therefore, if we can establish a link between these two users, not only we recommend trust but importantly we get rid of the coverage problem stated in the previous section. This link will work as the bridge between the two subgraphs.

To create such link between subgraphs, we introduce items as nodes within the original trust network. The network will no longer be trust network since there will be two different kinds of edges, user to user trust edges and user to item edges if the user rated the item. In Figure 2, it is evident that item I4 can be recommended to the blue node user, since it trusts the orange node user which is now linked to the green node user through the item nodes.

B. Matrix Representation and Factorization

We address the problem of *recommendation in disjoint trust networks* by considering a user-item and user-user matrices. This is different than some state-of-art algorithms, which adopted graph traversal techniques (such as TrustWalker[?]). We first build the two matrices using the user-item ratings and user trust information. Our model is based on merging both of the matrices as shown in Figure 3. This indicates that trust information is considered as a rating as well.

We use matrix factorization technique to propagate trust and recommend items. Matrix factorization approximates the underlying relationship between users, items and trust in TrusTem. That is, given K latent features (e.g. the underlying relationship), we can interpret matrix factorization results as follows. Each user rates each item through

	f1	f2
u1	0.8	0.2
u2	0.1	0.9

	i1	i2	u1	u2
f1	0.9	0.25	0.9	0.1
f2	0.15	1	0.1	0.8

Fig. 4: Illustration of matrix factorization interpretation in TrusTem.

Method	**RMSE**	**Coverage (%)**	**F-Measure**
TidalTrust[?]	1.096	84.9	0.65
MoleTrust[?]	1.093	82.1	0.608
TrustWalker[?]	1.077	95.4	0.827
SocialMF[?]	1.085	NA	NA
TrusTem	1.172	99.8	0.828

TABLE I: Experimental results using Epinions.com dataset.

these K features. On the other hand, the same user may trust other users through the same K features. For example, consider two users u_1 and u_2 and two items i_1 and i_2 and $K = 2$. u_1 likes i_1 but not i_2, u_2 dislikes i_1 but likes i_2. Figure 4 shows the merged matrix decomposition results.

V. Evaluation

We used the Epinions dataset, which includes $7,073$ users who rated a total of $139,738$ different items. In our experiments, we merged the ratings and trust matrices before running matrix factorization technique. We have set the latent features size to $K = 25$. In order to evaluate TrusTem, we use three performance metrics: (1) root mean squared-error (RMSE), (2) item coverage, (3) F-measure. To calculate $RMSE$, we use the dataset D that we already have ratings for.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in D}\left(r_{u,i} - \widehat{r_{u,i}}\right)^2}{|D|}} \tag{1}$$

F-measure combines both accuracy and coverage in one numeric metric. We employ the same formulas as TrustWalker[?] to calculate F-measure. First, we convert the $RMSE$ to precision using the following equation:

$$precision = 1 - \frac{RMSE}{4} \tag{2}$$

This represents the worst precision a recommender system can achieve, since the worst difference between actual and predicted ratings are $5 - 1 = 4$. Hence, F-measure can be calculated as follows:

$$Fmeasure = \frac{2 \times precision \times coverage}{precision + coverage} \tag{3}$$

These performance metrics applied on the user-item sub-matrix of the resultant matrix, since the user-user trust sub-matrix of the resultant matrix is usually considered as a secondary outcome of TrusTem.

We present our experiments results and comparison against other state-of-the-art algorithms. Table I shows RMSE, coverage and F-measure of these models when used on the Epinions.com dataset.

As shown in the table, TrusTem has better item coverage than the other models. TrustWalker model has one of the best item coverage which is around 95.4% whereas our model has around 99.8% coverage rate. However, we do not know the coverage rate of SocialMF which is a trust-based recommender system using matrix factorization technique. In SocialMF trust propagation happens through latent feature vector of every user and the latent feature of one user depends on their direct neighbor in the trust network, therefore, item coverage issue due to disjointedness will exist. Although TrusTem has improved item coverage rate compared to other methods but in terms of accuracy, it is 7% and 8% less than that of the SocialMF and TrustWalker methods respectively. However, TrusTem scores high F-measure compared to TidalTrust and MoleTrust. Also, it is as high as TrustWalker. It means that TrusTem almost does the same accuracy-coverage trade-off compared to TrustWalker.

VI. Conclusions

In this report, we defined a new problem: in trust-based recommender systems, trust disjointedness is one of the reasons behind item coverage reduction. In order to address this problem we devised a new solution model, based on merging user-item rating and user-user trust domains. Then, we employ matrix factorization to recommend items. We discussed how trust in the merged matrix propagates to recommend items to users belong to different subgraphs. Experiments on real life dataset from Epinions.com demonstrate that TrusTem outperforms existing methods for item coverage. However, it trades off rating accuracy to achieve that. This work suggests several directions for future work. Since items and users are represented in the same latent feature space, it implicitly describes the trust information based on several features. This gives us a notion that trust can be granularized into further details, which may improve the overall accuracy of the system.

1) N. E. Friedkin. (1998) A Structural Theory of Social Influence. Cambridge University Press.
2) Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., ... Tennenholtz, M. (2008). Trust-based recommendation systems: an axiomatic approach. In Proceedings of the 17th international conference on World Wide Web (pp. 199-208). ACM.
3) Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. IEEE Computer, 42:3037, 2009.
4) P. Massa and P. Avesani. Trust-aware recommender systems. In ACM Recommender Systems Conference (RecSys), USA, 2007.
5) J. Golbeck. Computing and Applying Trust in Web-based Social Networks. PhD thesis, University of Maryland College Park, 2005.
6) Jamali, M., Ester, M. (2009, June). 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.
7) Jamali, M., Ester, M. (2010, September). A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the fourth ACM conference on Recommender systems (pp. 135-142). ACM.
8) https://github.com/khaledmdiab/trustem