Trust-aware Social Recommender System Design

Peixin Gao[1], John S. Baras[1] and Jennifer Golbeck[2]
[1]*Institute for System Research, and Department of Electrical & Computer Engineering, University of Maryland, College Park, MD, U.S.A.*
[2]*College of Information Studies, and Department of Computer Science, University of Maryland, College Park, MD, U.S.A.*
{gaopei, baras, jgolbeck}@umd.edu

1 STAGE OF THE RESEARCH

With the explosive growth of information and the development of the Internet, we are overwhelmed with an extreme load of information. As a result, it is difficult to make decisions in front of immense variety of options. Recommender systems are thus designed to overcome the problem of information overload created by the Internet. However, current approaches for recommender system still suffer from the problems such as sparse information, cold start, and adversary attacks. On the other hand, social network sites (SNS), like Facebook and Epinions, offer a good source of knowledge for recommendation. The idea of integrating signals from social network to improve the performance of the recommendation algorithm has been well accepted and has attracted an increasing amount of research in both academia and industry, and many social network-based recommender system designs have been proposed and evaluated.

In this work, we develop a trust-aware recommender system. We interpret connections in associated social graph as trust relationships among users in the recommender system, and establish a trust network accordingly. Within the trust network, we propose models for trust propagation and aggregation, and design trust-aware recommendation algorithms to predict the preferences of users over items (services).

Specially, we handle indirect trust in our model, which could enlarge the information source to a large amount. We also discuss the issue of distrust (i.e. negative trust, negative opinion) and propose a way to consider both trust (positive opinion) and distrust in our model. We also consider integrating our trust-aware recommendation framework with classic collaborative filtering to take advantage of both approaches and further improve the performance in rating estimation and item recommendation. As an example of application scenario, such framework of trust-aware recommender system design can be applied for directed SNS like Epinions and Delicious.

Currently we are at the stage of evaluating the accuracy and efficiency of the system with both synthetic and real data. Meanwhile, we are trying to extend our model to further improve its performance.

Peixin Gao is the PhD student working on the topic under the supervision of Dr. John S. Baras and Dr. Jennifer Golbeck.

2 OUTLINE OF OBJECTIVES

The objectives of our work on trust-aware recommender system design are five-fold:

1. We discuss the limitations of current recommender systems and the possibility of introducing information from aligned social networks to improve the performance of recommender systems.
2. We interpret connections (links) in social networks as trust relationships among users, and propose a trust network comprised by users in the social graph and trust relationships among them, and develop an appropriate model for trust propagation and aggregation as well as trust value update within the trust network.
3. Based on the trust network and trust model, we design the recommendation algorithms to predict item ratings of users with information from trusted neighbors in a collaborative filtering fashion. Specially, we take care of both indirected trust and distrust (negative trust).
4. we also propose ways to integrate our trust-aware recommendation algorithms with classic user-based collaborative filtering method to make the novel recommendation algorithm capable of exploiting information from both rating matrix and trust network to provide better recommendations.
5. We explore directions to further extend our model for better performance and wider application scenarios.

3 RESEARCH PROBLEM

3.1 Recommender Systems

Recommender system (RS) can be defined as an information filtering system that is used to predict the rating or preference that a user would give to an item in the system (Adomavicius and Tuzhilin, 2005), and offer different users diverse suggestions. The goal of recommender systems is to provide personalized recommendations of items that suit a user's taste, in order to increase the number and variety of items sold, meanwhile improve user satisfaction and fidelity. Recommender systems usually have two applications; one is to predict the extent of interest the user has in an item which she has not yet rated, the other is to help the user find items that she is interested in but have not previously encountered. With decades of development, recommender systems have become a fundamental and important technology to help people make decisions and selections that fit their preference from an extreme overload of information, maintaining the loyalty of the customers and increasing the sales. Electronic retailers and content providers like Amazon and Netflix are increasingly adopting recommender systems in their platforms to improve user experience as well as their own profits.

Referring to the taxonomy of (Burke, 2007; Zanker et al., 2011), the classic approaches in RS can be classified into four groups, namely *Collaborative Filtering*, *Content-based*, *Knowledge-based* and *Hybrid Recommendation*. While collaborative filtering exploits item ratings to derive recommendations, content-based approaches rely on product features and textual descriptions, and knowledge-based algorithms reason on explicit knowledge models from the domain. Hybrid ones integrate different approaches together. Among all these approaches, Collaborative Filtering (CF) is currently the most successful and widely implemented. For example, user-based collaborative filtering approach (Herlocker et al., 1999; Koren, 2008) uses Eq. 1 to predict r_{ik}, i.e. user u_i's rating about item o_k:

$$\hat{r_{ik}} = b_{ik} + \frac{\sum_{u_j \in S(k;i)} s_{ij} \cdot (r_{jk} - b_{jk})}{\sum_{u_j \in S(k;i)} s_{ij}} \quad (1)$$

where $S(k;i)$ is the neighbor set of u_i about o_k, i.e. the set of users that are most similar to user u_i in terms of rating history, with s_{ij} is the similarity between user u_i and u_j, $\forall u_j \in S(k;i)$. b_{ik} and b_{jk} are the baseline estimates for r_{ik} and r_{jk} respectively, and r_{jk} is the rating of user u_j about item o_k.

CF has several advantages in application. First of all, CF techniques don't require domain knowledge or extensive data collection (Koren, 2008), and could scale well to large item bases. In addition, relying directly on user behavior allows uncovering complex and unexpected patterns that would be difficult or impossible to profile using known data attributes. Another advantage of using such an approach is that it is adaptive, i.e., the quality of the system improves over time. The more available ratings, the more accurate recommendations can be generated. Different from content-based recommendation, CF can add a serendipitous factor into the recommendation process. CF attracted much of attention in the past decade, resulting in significant progress and being adopted by some successful commercial systems, including Amazon and Netflix.

3.2 Open Issues and Challenges

However, as the most commonly applied approach, collaborative filtering still provide less than accurate rating prediction due to several open issues (Zanker et al., 2011). In this study, we focus on the following four problems, which we perceive to be the most significant ones.

1. **Data Sparsity:** Due to the fact that users typically rate or experience only a small fraction of available items, the rating matrix for a RS with millions of items is very sparse, which makes it hard to find users who have a similar rating behavior. Consequently, the quality of the generated recommendations might suffer from this.
2. **Cold Start:** The term "cold start" in RS design (Schein et al., 2002) basically describes two situations: (a) making recommendations to new users that have not rated any item, and (b) dealing with items that have not been rated or bought yet. RSs are usually unable to make good quality recommendations in such situations, since these newly joined users cannot be linked with similar users, so as the items. However, these are the users who need good quality recommendations the most as an incentive to continue using the system.
3. **Rating Integrity:** The integrity of ratings is another problem in RS, due to lack of incentives for a customer to rate products and the existence of the malicious users. RSs are facing latent attacks that aim to influence the functioning of the system, including random attack, average attack, bandwagon attack, segment attack love/hate attack and so on (Mobasher et al., 2007; O'Mahony et al., 2005; Hurley et al., 2007). Incentive design for RS and robustness of the system under attackers is currently an open issue that attracts more

and more research interest, especially when RSs start to become more decentralized.

4. **Scalability:** Scalability of recommendation algorithms with large and real-world datasets is also a practical and important consideration in recommender system design.

3.3 Trust-aware Social Recommender System as a Possible Approach

Social network sites (SNS), like Facebook and Epinions, provide as a complementary source of information for recommendation. Thus the idea of integrating information from SNS into RS to improve the performance of the recommendation algorithm (Guy et al., 2009) has been commonly accepted and has attracted an increasing amount of research (Palau et al., 2004; He and Chu, 2010; Tekin et al., 2013). Many social network-based recommender system designs have been proposed and evaluated (Yang et al., 2014).

In this work, we are trying to develop a trust-aware social recommender system, a social network-based RS based on CF techniques, which takes the behaviors and preferences of neighbors in SNS into consideration and interprets the social connections among users as trust relationships. A trust network is set up accordingly. The information extracted from SNS about the users in the recommender system could largely fill the “white space” and tackle the problem of cold start and data sparsity, as well as improve the relevance of recommendations and user satisfaction. We mainly focus on a static network structure while dynamic networks will be considered more thoroughly in our further work.

4 STATE OF THE ART

4.1 Social Recommender Systems

Since RS users share more and more information in associated social network sites (SNS), it makes SNS an important source of information about users and their preferences. SNS users have social profile, which can be used to calculate high-dimensional social and interest similarity between users. Meanwhile, due to the inter-connection of users in social networks, the effect of social influence from neighbor or community can be exploited in recommendation algorithms. Several issues of RSs mentioned above, for instance cold start, can be addressed with the help of this additional source of information. A social recommender system (SRS) can be built via integrating SNS with the recommender system and using information about users social profiles and/or relationships to suggest items that might be of interest to them.

The concept of social recommender system has been proposed for about a decade and a series of approaches have been proposed concerning combining information from social network into RS for better performance. However, the topic is still far from well-development, waiting for better solutions. A collective knowledge systems is proposed in (Gruber, 2008), in order to leverage collective intelligence in semantic web. Bonhard et al. (Bonhard and Sasse, 2006) studied the factors that drive people’s decision-making and advice-seeking through empirical studies, and found out that the profile similarity and rating overlap of a recommender have a significant impact on a person’s decision. Carmagnola et al. (Carmagnola et al., 2009) proposed to evaluate the value of a user’s interest in an item (”score”) using both the strength of the relationship between the user and its neighbors and the level of interest the item has for each neighbor of the target user. The RS algorithm designed in (He and Chu, 2010) integrates user’s own preference, item’s general acceptance and influence from friends. While the introduction of distant friends improves the coverage of the recommendation algorithm, it affects the accuracy of the approach.

4.2 Trust Evaluation in Social Network

When integrating a social network of directed links to improve the performance of the recommender system, these directed links can be interpreted as trust relationships among users, which reflect the preference (fondness), social closeness (subjective similarity) and integrity of users in the network. These trust relationships can be exploited to provide extra source of information for recommendation hence improve the performance of the RS. A lot of research has been conducted in inferring and evaluating trust/distrust relationships in SNS, as well as establishing trust network based on SNS structure.

On calculating trust values, Richardson et al. proposed to deploy a web of trust (Richardson et al., 2003), where each user only maintains trusts of a small number of other users (i.e. neighbors) and trust values for all other users can be calculated via the trust network. Zaihrayeu et al. presented an trust inference infrastructure called IWtrust for calculating trust values for answers from the web (Zaihrayeu et al., 2005). Golbeck proposed TidalTrust (Golbeck, 2005), which aggregates the weighted trust values between neighbors with direct trust. TidalTrust only takes into account the shortest, strongest paths thus may loose some ratings from distant users in the network. MoleTrust (Avesani et al., 2005) proposed by Avesani et al. is similar to TidalTrust but considers all raters up to a maximum-depth given as an input. Weng et al. used trust for the cold start problem by automatic trust generation even for users who have rated few items in common (Weng et al., 2006). Jøsang et al. (Jøsang et al., 2006) treat trust and distrust as two separate concepts and proposed three probabilistic aggregation operators called consensus operators for the fusion of dependent, independent and partially dependent opinions respectively. However, these operators assume that users have equal weights (equally importance), and hence lack flexibility. DuBois et al. designed a probabilistic approach to infer the trust relationship between users in social networks under axioms of inferred trust and applied it in network clustering (DuBois et al., 2009b). In their following work (DuBois et al., 2011), they further considered distrust in the network and introduced a modified spring-embedded algorithm for trust inference. Leskovec et al. conducted research on signed edge (link) prediction in online social networks using a machine learning framework (Leskovec et al., 2010). They showed that the information about negative relationships is useful in edge prediction.

4.3 Trust-aware Recommender System

Introducing a trust network into RS could improve not only the coverage but also the accuracy of the recommendation and the satisfaction of users. Also, since malicious nodes can be recognized and excluded from trust network, many attacks towards recommender system can be effectively resisted. Currently RS are only being used in low risk domains, due to lack of transparency in RS. According to (Sinha and Swearingen, 2001), users prefer more transparent systems, and tend to rely more on recommendations from people they trust than on those from anonymous people “similar”to them. With the trust network, the RS can be more transparent and easier to be accepted, opening space for more application scenarios for RS.

In (Massa and Avesani, 2004), a trust-aware method for recommender system is proposed, where the collaborative filtering process is informed by the reputation of users computed via propagating trust. The design was showed to increases the coverage while not reducing the accuracy. Bedi et al. (Bedi et al., 2007) proposed a trust-based recommender system for the semantic web with the knowledge distributed over the network in the form of ontologies, where a trust network is deployed to generate recommendations. Andersen et al. (Andersen et al., 2008) developed an natural set of five axioms, namely symmetry, positive response, independence of irrelevant stuff, neighborhood consensus and transitivity desired for designing trust-aware recommender systems, and proposed a recommendation algorithm based on random walks by weakening the axioms. In order to tackle the problem of opinion ignorance and inconsistency which may affect trust estimations, Victor et al. proposed a new framework called “Trust Score”(Victor et al., 2011), where the trust relation is a fuzzy mapping (Griffiths, 2006) from agent pairs to trust vectors in $[0,1]^2$, containing trust degree and distrust degree. Additional dimensionality of trust value introduced in (Victor et al., 2011) can differentiate partial trust, partial distrust, partial ignorance and partial inconsistency. In (DuBois et al., 2009a) the author used an probabilistic trust inference algorithm developed in (DuBois et al., 2009b) to set up trust metrics and conduct trust-based clustering on users to further improve recommendation accuracy.

Some approaches also tried to combine classic collaborative filtering with trust-based recommendation algorithms. O’Donovan et al.(O’Donovan and Smyth, 2005) focused on trust-based adaptations of collaborative filtering, which they called trust-based filtering, where only the most trustworthy neighbors participate in the recommendation process. Two trust-aware methods at profile-level and profile-item-level respectively are introduced to improve standard collaborative filtering methods and showed that trust information can help increase recommendation accuracy. This algorithm does not involve trust propagation or aggregation thus has the disadvantage of sparser rating matrix. Ma et al. demonstrated that a factor analysis method (Ma et al., 2008), which fuses the user-item matrix with the users’ social trust networks, generates better recommendations than the traditional collaborative filtering algorithms. This approach, however, does not reflect the real world recommendation process. Later in (Ma et al., 2009), a user’s final rating decision was interpreted as the balance between this user’s own taste and her trusted users’ favors, and an ensemble probabilistic matrix factorization method is proposed to implement the idea. Jamali et al. (Jamali and Ester, 2009) developed a random walk model combining the trust-based and the collaborative filtering approach for recommendation. The random walk model allows to define and to measure the confidence of a recommendation. Wei et al. proposed a multi-collaborative filtering trust network algorithm (MCFTN) for recommendation under Web 2.0 circumstance, with the assumption that all the necessary information is available (Wei et al., 2013). In their approach, different sources of information are combined in a collaborative filtering fashion. However, the trust propagation does not consider aggregation of different opinions.

5 METHODOLOGY

In order to tackle the issues mentioned in Sec. 3.2, we propose a trust-aware recommender system which integrates information from directed social networks.

Trust is an umbrella term for a wide range of meanings. In the research area of RS, trust can be interpreted in several different ways:

- Trust can be defined as a measure of confidence that the user's rating reflects her real opinion (i.e. integrity), which is evaluated to discover and avoid attacks on the system. This is similar to the definition of trust in control and operation research (Theodorakopoulos and Baras, 2006; Jiang and Baras, 2009).
- Trust can also be used to describe the quality of the recommendations made by the system.
- When integrating SNS into RS, trust is directed relationship between users and is a compound of integrity, preference and social closeness. In such circumstance, trust among users forms a network called trust network (Victor et al., 2011).

In this literature, we use the third definition of trust in our trust-aware recommender system.

5.1 Trust Model

The main aim in setting up trust networks is to allow agents to form trust opinions on unknown agents or sources by asking for a trust opinion from acquainted agents. While trust is increasingly involved, the use and modeling of distrust remains relatively unexplored. Although recent research (Golbeck, 2008; Victor et al., 2011) show an emerging interest in modeling the notion of distrust, models that take into account both trust and distrust are still scarce. Most approaches completely ignore distrust, or consider trust and distrust as opposite ends of the same continuous scale. However, there is a growing body of opinion that distrust cannot be seen as the equivalent of lack of trust (Gans et al., 2001; Guha et al., 2004).

Referring to previous work (Massa and Bhattacharjee, 2004; Guha et al., 2004; Golbeck, 2005; Massa and Avesani, 2007; Golbeck, 2008; Walter et al., 2008; Victor et al., 2011), we define the concept of trust network used in our trust-aware recommender system as follows.

Definition. *Trust Network:* The directed trust network $T(V,E)$ is established via the social graph associated with the recommender system, where V is the set of nodes (i.e. users) with $|V| = N$, E is the set of directed edges (i.e. trust links). $\forall$ directed edge $e_{ij} = (v_i, v_j) \in E, v_i, v_j \in V$, is a directed trust link from node v_i towards v_j, with value (weight) $w_{ij} \in [-1,1]$ indicating the extent of trust that node v_i put on v_j. These links are not necessarily symmetric. A weight of 1 means "totally agree"or "like", while weight of -1 means "totally disagree"or "dislike". $N_i = \{v_j | e_{ij} \in E\}$ is the neighbor set of node v_i.

As is in classic collaborative filtering, there exist a set of items $O = \{o_1, o_2, \cdots, o_M\}$. The rating of user v_i on item o_k is r_{ik}, which is the element in ith column and kth row of the rating matrix R. The ratings are integers and the rating range is set to be $[-5,5]$, both configurable to different application scenarios.

Trust networks are typically challenged by two important problems that influence trust opinions. Firstly, in large networks it is likely that many agents do not know each other, hence there is an abundance of ignorance. Secondly, because of the lack of a central authority, different agents might provide different and even contradictory information, thus inconsistency may occur. In our trust model, we propose to apply trust propagation to eliminate the ignorance on users due to no direct connections. Regarding the second problem, we design a distributed trust aggregation rule that can handle both trust and distrust.

Definition. *Trust Propagation:* In our trust-aware system, trust between two nodes of no direct connections can be estimated using trust values of edges along the path between the two nodes. A maximum path length λ and trust threshold τ are set up to save computation resource and avoid infinite loop. If the path length exceeds λ or the trust value decreases below τ along the path, then there's no trust (or distrust) relation between the two nodes.

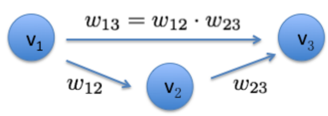

Figure 1: Trust propagation.

Fig. 1 illustrates how the trust propagation works in our system. In the example, v_1 can reach indirect trust (w_{13}) about v_3 via its direct trust (w_{12}) about v_2 and v_2's direct trust about v_3 (i.e. w_{23}). If the maximum length $\lambda \geq 2$, and the threshold $\tau \leq w_{12} \cdot w_{23}$, then v_1's indirect trust about v_3 exists and can be calculated as Eq. 2 (otherwise w_{13} doesn't exist).

$$w_{13} = w_{12} \cdot w_{23} \tag{2}$$

When there are multiple paths between two nodes, the indirect trust value calculated along different paths are combined using trust aggregation. There are two major ways to conduct trust aggregation, namely *First Aggregate Then Propagate* (FATP) and *First Propagate Then Aggregate* (FPTA). We apply the first approach (i.e. FATP) in our system.

Definition. *Trust Aggregation:* Indirect trust values towards node v_t calculated from different paths can be combined in a recursive way: for nodes that have direct trust values about v_t, the aggregated trust values are their direct trust values; for each node v_i along the paths who has no direct trust value, she learns her neighbors' trust values about v_t and combine them according to her trust towards her neighbors:

$$w_{it} = \frac{\sum_{v_j \in N_i, w_{ij} \geq \sigma} w_{ij} \cdot w_{jt}}{\sum_{v_j \in N_i, w_{ij} \geq \sigma} w_{ij}} \tag{3}$$

The opinion of neighbor $v_j \in N_i$ of v_i will not be take into consideration if v_i has low trust about v_j (i.e. $w_{ij} < \sigma$ the threshold).

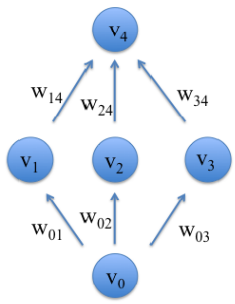

Figure 2: Trust aggregation.

Fig. 2 illustrates how the trust aggregation is conducted in our system. If w_{01}, w_{02} and w_{03} are all greater than the threshold σ, they will be used for calculating the trust of v_0 about v_4 using Eq. 4

$$w_{04} = \frac{w_{01} \cdot w_{14} + w_{02} \cdot w_{24} + w_{03} \cdot w_{34}}{w_{01} + w_{02} + w_{03}} \tag{4}$$

5.2 Trust-aware Recommendation Algorithms

Based on our trust model, we propose several candidate trust-aware recommendation algorithms which takes both trust and distrust into consideration.

5.2.1 Distrust Filtering

For RS of adequate rating information, we propose to integrate trust network via *Distrust-based Filtering*. This approach is intuitive and easy to apply to collaborative filtering RS. The idea is that with the trust network, we could filter out users of low trust even they are "similar" to the target user.

There are two possible approaches:

- If the target user's trust value about a node is lower than the pre-configured threshold, the node's rating will not be considered in rating prediction.
- When using neighborhood-based CF method, the calculated similarity will be weighted by the trust about the node.

When applying the first approach, $S(k;i)$ should be updated as $\{v_j | v_j \in S(k;i), w_{ij} \geq \eta\}$. When using the second approach, Eq. 1 should be modified as:

$$\hat{r_{ik}} = b_{ik} + \frac{\sum_{v_j \in S(k;i), w_{ij} \geq 0} w_{ij} s_{ij} \cdot (r_{jk} - b_{jk})}{\sum_{v_j \in S(k;i), w_{ij} \geq 0} w_{ij} s_{ij}} \tag{5}$$

In this way, the security of the RS can be improved. However, the coverage of the RS has no improvement and the problem of data sparsity and cold start are not addressed via this approach.

5.2.2 Trust-weighted Recommendation

Similar to previous work (Massa and Avesani, 2004; Golbeck, 2005; Massa and Avesani, 2007; Golbeck, 2008), this method uses trust relationships for recommendations. When introducing distrust (negative opinion), the equation for rating prediction is modified accordingly. Here we propose a 2-step approach.

1. In the first step, only trusted neighbors are considered in rating prediction.
2. Then the opinions of distrusted nodes are mirrored about the rating reached in the first step. Then calculate the weighted average rating with absolute value of trust as weights.

Based on our trust propagation and aggregation rule, this recommendation algorithm has a greedy implementation. For node v_i, if she doesn't have direct rating about item o_k, then in step 1, her preliminary rating $\bar{r_{ik}}$ about item o_k can be calculated using Eq. 6:

$$\bar{r_{ik}} = \frac{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} \cdot r_{jk}}{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij}} \tag{6}$$

With the preliminary rating $\bar{r_{ik}}$, a rating r_{nk} about o_k from a distrusted node v_n (with negative trust value) is adjusted as $\tilde{r_{nk}}$ according to Eq. 7.

$$\tilde{r_{nk}} = \min\{\max\{2\bar{r_{ik}} - r_{nk}, r_{\min}\}, r_{\max}\} \tag{7}$$

where the min and max bounds are used to avoid exceeding the rating range. With the adjusted rating of distrusted nodes, in the second step, the predicted rating about r_{ik} can be calculated with Eq. 8.

$$\hat{r_{ik}} = \frac{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} \cdot r_{jk} + \sum_{v_n \in N_i, w_{in} < 0} |w_{in}| \cdot \tilde{r_{nk}}}{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} + \sum_{v_n \in N_i, w_{in} \geq 0} |w_{in}|} \quad (8)$$

In order to avoid infinite loops and save computation resources, the length of the trust propagation path is upper-bounded by K.

Note that in this algorithm, only information of neighbors in the trust network is needed to reach the prediction on ratings, thus the algorithm has the advantage of good scalability in large scale systems. The number of trusted neighbors used in calculation, similar to user-based CF, is configurable and can be tuned to reach better performance.

5.2.3 Recommendation with Random Trust Propagation

Even through coverage of the RS can be improved via trust propagation, the accuracy of recommendation algorithm may be affected. In order to balance the coverage and accuracy, the major concerns in RS, we follow the idea of using ratings of similar items with probabilistic switch between trust propagation and approximate rating discussed in (Jamali and Ester, 2009), and proposed our recommendation algorithm with probabilistic trust propagation based on the trust-aware recommendation algorithm discussed in Sec. 5.2.2.

At each node v_i, we don't always predict its rating about an item o_k via aggregating opinions from its neighbors in N_i. Instead, there exists probability $\theta_{ik,t}$ that the rating $\hat{r_{ik}}$ is calculated with the v_i's ratings about items that are similar to o_k. Here the similarity metric can be any form, e.g. cosine similarity used in item-based collaborative filtering.

As the probability to choose between trust propagation and rating prediction with similar items, $\theta_{ik,t}$ is related to node v_i, item o_k and the position of v_i along the trust propagation path (i.e. the step t). If v_i chooses to use her ratings on similar items to predict $\hat{r_{ik}}$, then

$$\hat{r_{ik}} = \frac{\sum_{o_t \in I(k;i)} s_{tk} \cdot r_{it}}{\sum_{o_t \in I(k;i)} s_{tk}} \quad (9)$$

where $I(k;i)$ is the set of items rated by v_i which are similar to o_k.

Apart from the upper-bound K for trust propagation, another condition is introduced to terminate the propagation process:

$$(1 - \theta_{ik,t}) \cdot \max\{w_{ij}, j \in N_i\} \geq \Gamma \quad (10)$$

where Γ is the threshold for the condition and is tunable. This condition indicates that if v_i has rated items very similar to o_k ($\theta_{ik,t}$ very large), or the neighbors of v_i are not trustworthy, then the trust propagation process will end. The performance of the system is determined by paths of high trust and items of high similarity with the target item, both related to Γ.

The switch probability $\theta_{ik,t}$ can also be interpreted as a weight term, in which case the equation for rating prediction is:

$$\begin{aligned}\hat{r_{ik}} =& (1-\theta_{ik,t}) \frac{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} \cdot r_{jk} + \sum_{v_n \in N_i, w_{in} < 0} |w_{in}| \cdot \tilde{r_{nk}}}{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} + \sum_{v_n \in N_i, w_{in} \geq 0} |w_{in}|} \\ &+ \theta_{ik,t} \cdot \frac{\sum_{o_t \in I(k;i)} s_{tk} \cdot r_{it}}{\sum_{o_t \in I(k;i)} s_{tk}}\end{aligned} \quad (11)$$

Similar to the trust-aware recommendation algorithm described in Eq. 8, this probabilistic trust propagation model can also be implemented in decentralized way and only local information is used to reach prediction, which makes it highly scalable in large scale systems.

5.2.4 Trust-aware Combinatorial CF

In this method, both similarity and trust values are normalized. A proportion α is calculated based on the total weight of similar nodes and trusted neighbors:

$$\alpha_{ik} = \frac{\sum_{v_l \in S(k;i), w_{il} \geq \eta} s_{il}}{\sum_{v_l \in S(k;i), w_{il} \geq \eta} s_{il} + \sum_{v_j \in T_i} w_{ij}} \quad (12)$$

where $S(k;i)$ is the neighbor set of v_i about item o_k, T_i is the set of nodes used for trust-aware recommendation for v_i, and η is the threshold used to exclude users who is distrusted by v_i.

The combinatorial CF is a mixture of user-based and trust-aware collaborative filtering. The proportion α_{ik} is the weight of result reached via user-based CF, and the estimated rating can be expressed as:

$$\begin{aligned}\hat{r_{ik}} =& (1-\alpha_{ik}) \cdot \hat{r_{ik}}^{\text{trust}} + \alpha_{ik} \cdot \hat{r_{ik}}^{\text{sim}} \\ =& (1-\alpha_{ik}) \frac{\sum_{j \in N_i, w_{ij} \geq 0} w_{ij} r_{jk} + \sum_{n \in N_i, w_{in} < 0} |w_{in}| \tilde{r_{nk}}}{\sum_{j \in N_i, w_{ij} \geq 0} w_{ij} + \sum_{n \in N_i, w_{in} < 0} |w_{in}|} \\ &+ \alpha_{ik} (b_{ik} + \frac{\sum_{l \in S(k;i), w_{il} \geq \eta} s_{il} (r_{lk} - b_{lk})}{\sum_{l \in S(k;i), w_{il} \geq \eta} s_{il}})\end{aligned} \quad (13)$$

where $\hat{r_{ik}}^{\text{sim}}$ is the prediction of user-based CF (Herlocker et al., 1999; Koren, 2008), and $\hat{r_{ik}}^{\text{trust}}$ is the result of trust-aware CF as mention in Sec. 5.2.2.

5.2.5 Trust-aware Composite CF

With normalized similarity and trust values, we can apply the way *EnsembleTrustCF* (Victor et al., 2011) used to combine ratings of similar users and trusted neighbors:

$$\hat{r_{ik}} = \frac{\sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij}(r_{jk} - b_{jk}) + \sum_{v_n \in N_i, w_{in} < 0} |w_{in}|(\tilde{r_{nk}} - b_{nk})}{\sum_{v_j \in S(k;i)} s_{ij} + \sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} + \sum_{v_n \in N_i, w_{in} \geq 0} |w_{in}|} + \frac{\sum_{v_j \in S(k;i)} s_{ij}(r_{jk} - b_{jk})}{\sum_{v_j \in S(k;i)} s_{ij} + \sum_{v_j \in N_i, w_{ij} \geq 0} w_{ij} + \sum_{v_n \in N_i, w_{in} \geq 0} |w_{in}|} + b_{ik} \quad (14)$$

5.2.6 Trust Value Update

v_i's trust value about node v_j can be updated via comparing the predicted ratings $\hat{r}_i$, real ratings r_i and r_j, as shown in Fig. 3. When $\hat{r}_i$ locates between r_j and r_i, it means j's opinion makes negative contributions to recommendation, thus the trust value w_{ij} will decrease ($w_{ij} = w_{ij} - \delta$). When r_i lies in between r_j and $\hat{r}_i$, the weight w_{ij} should be increased ($w_{ij} = w_{ij} + \delta$), since j makes positive contribution in this case.

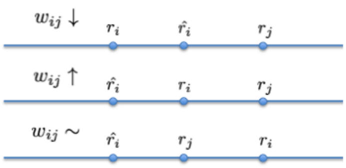

Figure 3: Update w_{ij} based on real and predicted ratings.

With such trust update scheme, the trust values are adjusted according to users' (nodes) behaviors and the prediction results using trust-aware recommendation algorithms be more accurate and the performance of the system can be further improved. This trust dynamics is distributed and restricted only to neighbors , instead of requiring users to maintain global knowledge. Such distributed trust update scheme makes the system efficient in memory and computation resources.

6 EXPECTED OUTCOME

Current recommender systems are confronting problems like data sparsity, cold start and adversary attacks. One promising approach is to establish a trust network from the social networks associated with the RS and conduct rating prediction with neighbors' ratings in the trust network. Such an RS design is called trust-aware social recommender system.

In this work, we propose several trust-aware recommendation algorithms, in order to increase knowledge base for better recommendations, by introducing trust signals from SNS. Specially, we introduce a novel way to handle negative trust (distrust). We also consider integrating the novel recommendation algorithms with classic collaborative filtering approach.

By introducing information from social network, as shown in Sec. 5, the knowledge base of the recommender system can be enlarged and the issue of data sparsity can be addressed. Thus we expect that the performance of the RS can be improved. Especially for the case of cold start, when the system has little information about the user's preference, it's hard for classic collaborate filtering methods to predict her ratings over items. However, with trust relationships extracted from social connections, the system is expected to be able to predict her preferences and offer proper suggestions on items she needs.

Among the five approaches in trust-aware RS design that we propose in this work, two are solely based on trust values and ratings of neighbors in trust network and can be implemented in a decentralized way, which makes the two approaches highly scalable. Here in our model, the trust propagation ends within a limited number of steps, which is inspired by the fact that social influence is shallow (Ugander et al., 2011). We predict that this setting still offers relatively precise trust estimation, meanwhile bringing a substantial save on computation resource and boost on algorithm efficiency. Besides, since trust value update doesn't require global information, it enables the prompt update on trust values, which could further improve the accuracy of the algorithms.

Since the trust relationship between users are exploited in the algorithm, it's much harder for the adversaries to attack or compromise the system with techniques used on current recommender systems. Thus we expect our system to be more secure.

We will conduct experiments on large scale datasets to test the performance of the trust-aware recommendation algorithms that are proposed in this work and compare them with other state-of-art methods, in order to verify our design and adjust the model to further improve the performance.

So far, we assume that the social network associated with the recommender system does not change over time, which in reality may not be valid. Typically, social networks evolve over time. Thus in our further work, we plan to investigate the influence of network dynamics to the system, and introduce probabilistic model to describe the system behaviors.

Trust is currently defined as a 1-dimensional value. In the future we plan to extend it to a multi-dimensional vector, with each element representing the trust relation in that dimension (e.g. category). Meanwhile, we will discuss the robustness of the different RS designs under adversary attacks. We are also interested in expanding the horizon of recommender systems and making them available in more application scenarios.

REFERENCES

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749.

Andersen, R., Borgs, C., Chayes, J., Feige, U., Flaxman, A., Kalai, A., Mirrokni, V., and Tennenholtz, M. (2008). Trust-based recommendation systems: an axiomatic approach. In *Proceedings of the 17th international conference on World Wide Web*, pages 199–208. ACM.

Avesani, P., Massa, P., and Tiella, R. (2005). A trust-enhanced recommender system application: Moleskiing. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1589–1593. ACM.

Bedi, P., Kaur, H., and Marwaha, S. (2007). Trust based recommender system for semantic web. In *IJCAI*, volume 7, pages 2677–2682.

Bonhard, P. and Sasse, M. (2006). knowing me, knowing youusing profiles and social networking to improve recommender systems. *BT Technology Journal*, 24(3):84–98.

Burke, R. (2007). Hybrid web recommender systems. In *The adaptive web*, pages 377–408. Springer.

Carmagnola, F., Vernero, F., and Grillo, P. (2009). Sonars: A social networks-based algorithm for social recommender systems. In *User Modeling, Adaptation, and Personalization*, pages 223–234. Springer.

DuBois, T., Golbeck, J., Kleint, J., and Srinivasan, A. (2009a). Improving recommendation accuracy by clustering social networks with trust. *Recommender Systems & the Social Web*, 532:1–8.

DuBois, T., Golbeck, J., and Srinivasan, A. (2009b). Rigorous probabilistic trust-inference with applications to clustering. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 655–658. IET.

DuBois, T., Golbeck, J., and Srinivasan, A. (2011). Predicting trust and distrust in social networks. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 418–424. IEEE.

Gans, G., Jarke, M., Kethers, S., and Lakemeyer, G. (2001). Modeling the impact of trust and distrust in agent networks. In *Proc. of AOIS01*, pages 45–58.

Golbeck, J. (2008). *Computing with social trust*. Springer.

Golbeck, J. A. (2005). Computing and applying trust in web-based social networks.

Griffiths, N. (2006). A fuzzy approach to reasoning with trust, distrust and insufficient trust. In *Cooperative Information Agents X*, pages 360–374. Springer.

Gruber, T. (2008). Collective knowledge systems: Where the social web meets the semantic web. *Web semantics: science, services and agents on the World Wide Web*, 6(1):4–13.

Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, pages 403–412. ACM.

Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., and Ofek-Koifman, S. (2009). Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM conference on Recommender systems*, pages 53–60. ACM.

He, J. and Chu, W. W. (2010). *A social network-based recommender system (SNRS)*. Springer.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM.

Hurley, N. J., O'Mahony, M. P., and Silvestre, G. C. (2007). Attacking recommender systems: A cost-benefit analysis. *Intelligent Systems, IEEE*, 22(3):64–68.

Jamali, M. and Ester, M. (2009). 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*, pages 397–406. ACM.

Jiang, T. and Baras, J. S. (2009). Graph algebraic interpretation of trust establishment in autonomic networks. *Preprint Wiley Journal of Networks*.

Jøsang, A., Marsh, S., and Pope, S. (2006). Exploring different types of trust propagation. In *Trust management*, pages 179–192. Springer.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM.

Ma, H., King, I., and Lyu, M. R. (2009). Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM.

Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM.

Massa, P. and Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 492–508. Springer.

Massa, P. and Avesani, P. (2007). Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM.

Massa, P. and Bhattacharjee, B. (2004). Using trust in recommender systems: an experimental analysis. In *Trust Management*, pages 221–235. Springer.

Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology (TOIT)*, 7(4):23.

O'Donovan, J. and Smyth, B. (2005). Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174. ACM.

O'Mahony, M. P., Hurley, N. J., and Silvestre, G. C. (2005). Recommender systems: Attack types and strategies. In *AAAI*, pages 334–339.

Palau, J., Montaner, M., López, B., and De La Rosa, J. L. (2004). Collaboration analysis in recommender systems using social networks. In *Cooperative Information Agents VIII*, pages 137–151. Springer.

Richardson, M., Agrawal, R., and Domingos, P. (2003). Trust management for the semantic web. In *The Semantic Web-ISWC 2003*, pages 351–368. Springer.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260. ACM.

Sinha, R. R. and Swearingen, K. (2001). Comparing recommendations made by online systems and friends. In *DELOS workshop: personalisation and recommender systems in digital libraries*, volume 106.

Tekin, C., Zhang, S., and van der Schaar, M. (2013). Distributed online learning in social recommender systems. *arXiv preprint arXiv:1309.6707*.

Theodorakopoulos, G. and Baras, J. S. (2006). On trust models and trust evaluation metrics for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):318–328.

Ugander, J., Karrer, B., Backstrom, L., and Marlow, C. (2011). The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*.

Victor, P., Cornelis, C., and De Cock, M. (2011). *Trust networks for recommender systems*, volume 4. Springer.

Walter, F. E., Battiston, S., and Schweitzer, F. (2008). A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74.

Wei, C., Khoury, R., and Fong, S. (2013). Web 2.0 recommendation service by multi-collaborative filtering trust network algorithm. *Information Systems Frontiers*, 15(4):533–551.

Weng, J., Miao, C., and Goh, A. (2006). Improving collaborative filtering with trust-based metrics. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1860–1864. ACM.

Yang, X., Guo, Y., Liu, Y., and Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications*, 41:1–10.

Zaihrayeu, I., Da Silva, P. P., and McGuinness, D. L. (2005). Iwtrust: Improving user trust in answers from the web. In *Trust Management*, pages 384–392. Springer.

Zanker, M., Felfernig, A., and Friedrich, G. (2011). *Recommender systems: an introduction*. Cambridge University Press.