Trust Inference using Implicit Influence for Item Recommendation

Bithika Pal
Department of Industrial and Systems Engineering
Indian Institute of Technology
Kharagpur, India
Email: bithikapal@iitkgp.ac.in

Mamata Jenamani
Department of Industrial and Systems Engineering
Indian Institute of Technology
Kharagpur, India
Email: mj@iem.iitkgp.ernet.in

***Abstract*—Trust plays a very important role in many existing e-commerce recommendation applications. Social or trust network among users provides an additional information along with the ratings for improving the user reliability on the recommendation. However, in real world, trust data is sparse in nature. So, many algorithms are built for inferring trust. In this paper, we propose a new trust inference method based on the implicit influence information available in the existing trust network. This approach uses the transitivity property of the trust for trust propagation and scale-free complex network property to limit the propagation length in the network. In this regard, we define a new terminology, *degree of trustworthiness* for a user, which adds the global influence in the inferred trust. This process improves the recommendation accuracy from the existing trust-based recommendation and neighborhood based collaborative filtering. Due to the availability of users preference from trust network which is absent in rating data, it also alleviates the very well-known cold start users problem of a recommender system. We evaluate the proposed approach on two established real world datasets and report the obtained results.**

***Index Terms*—Recommender system; Collaborative Filtering; Trust Network**

I. Introduction

In today's world of internet, mobile, and online activity, personalization comes in a very big picture when it is talked about recommendation and searching. Recommender system (RS) has a broad area of application such as, movie recommendation in *netflix*[1], music recommendation in *lastfm*[2], book recommendation in *goodreads*[3], product recommendation in *amazon*[4], friend recommendation in *facebook* or other social networking sites, news recommendation, etc. The goal of a recommender system is to choose the right item for the right person at the right time in a personalized way. In RS, Collaborative Filtering (CF) is an elementary technique which returns very promising result by only considering user-item rating or purchase history. Several algorithms in this regard [1] has been developed including neighborhood based approaches and model-based approaches. In case of neighborhood based approaches [2], new recommendations are determined by considering the similarity of users or items and then taking nearest neighbors from them, whereas, model based method (latent-factor model) [3, 4] maps users and items into the same low dimensional latent space and predicts the recommendation. The big challenge of RS is sparsity of rating data [5]. Due to this reason, recent advances of recommender system work by not only taking rating data rather it incorporates more information, if available, to boost up the power of recommender system [5, 6]. Each kind of addition like social information, trust relation, the knowledge base of items or users helps to alleviate different type of problems that a recommender system faces [7, 8]. In this paper, we focus on a familiar problem of the recommender system, i.e. the existence of cold start users. To resolve that we have taken trust relation among users as extra information along with the rating or purchase history.

In case of RS, trust is identified in two ways either at individual level [7, 9] or trust on the whole system [10]. Individual level trust comes from when one user relies on other explicitly and how much this trust is quantified with some real value as the weight of trust. This framework generates directed weighted graph named "web-of-trust", where users are the vertices and the directed edges signify which user trusts on whom with weights as trust values like in *epinions*[5], *flimtrust*, skiing recommendation, etc [11, 12]. The number of users trusting to a particular user with their trust weight leads to the reputation of the user in a trust-based system. Sometimes social network information is also used as trust data like in *flixster*[6]. Explicit commence of trust to a particular user opens the door of recommending new items from its trusted user's purchase list; this enhances profitability and diversity by recommending less frequently sold items. The benefits of RS to the seller side come from this "long-tail" of item distribution. Also, it helps to know the taste of new user in the system. However, due to the sparse nature of trust data, many trust inference approaches have been come up such as graph theoretic models [9, 11, 13]–[15], algebraic methods [16] and machine learning approaches [17, 18]. Graph theoretic model has a good performance in terms of reasoning and efficiency. Discovering new edges in trust graph enhances the scope of recommending more diverse items to the user. On the other

[1] www.netflix.com
[2] www.last.fm
[3] www.goodreads.com
[4] www.amazon.com
[5] www.epinions.com
[6] www.flixster.com

978-1-5386-4283-2/17 $31.00 © 2017 IEEE
DOI 10.1109/SITIS.2017.15

hand, machine learning model gives better prediction level with the limitation of slower computational speed and a weak interpretation of inference. Here we have enhanced the graph-theoretic model of trust inference [7, 19, 20]. For inferring trust from one user to other, existing path based model considers the propagation property of trust through its connecting path and linear decaying effect of it. Also, this method uses trust inference at an individual level and has not considered the of the users in the whole system. To address this issue we have defined a new factor *degree-of-trustworthiness* which will capture the reputation of the user in the system. The basic idea is to add this information which is coming implicitly from the topological structure of trust network as an influence to the identified path based inference.

The contribution of our work is three-fold:

1) we introduce a new trust inference mechanism considering the influence of users.
2) we evaluate our method using two real-world datasets, for trust-aware recommender system and compare the result with existing methods.
3) we analyze our proposed method in depth for cold start users problem.

Our paper is organized as follows. In Sec. II and III, we discuss about notations, definitions, preliminaries and related work. Then, we introduce, our proposed method and algorithms in Sec. IV. We explore the dataset in Sec. V, consecutively, evaluate our proposed approach in Sec. VI, and conclude our work in Sec. VII.

II. Notations and Definitions

a) Recommender system: A recommender system is a tool or technique for predicting the rating of unknown items and offers top rated item list for a user according to their choice. The inputs for this system are set of n users $\mathcal{U}$, set of m items $\mathcal{I}$ and rating matrix $\mathbf{R}_{n\times m} \in \mathbb{R}^{n\times m}$. The ith row $\mathbf{R}$, denoted by $\mathbf{r}_{u_i}$, is a row vector and represents the rating provided by user u_i. Each element of $\mathbf{R}$, $r_{i,j}$, can be defined as follows:

$$r_{i,j} = \begin{cases} x, & \text{if rating available} \\ \text{blank}, & \text{otherwise} \end{cases} \tag{1}$$

where x is real value in some ordinal scale. Binary rating matrix gives the purchase history where $r_{i,j} = 1$ if user u_i has purchased item i_j otherwise zero.

b) Trust network: Trust network is a directed graph $\mathcal{G}_t = (\mathcal{U}_t, \mathcal{E}_t, \omega)$ where vertex set $\mathcal{U}_t$ is set of users in trust network and edge set $\mathcal{E}_t(G_t) = \{< u_i, u_j > \mid u_i, u_j \in \mathcal{U}_t, u_i \text{ trust on } u_j\}$, and $\omega :< u_i, u_j > \rightarrow$ *trust value* as edge weight. For simplicity, we consider $\mathcal{U}_t(G_t) = \mathcal{U}$. Trust relation between two users is determined by the directed edge in $\mathcal{G}_t$, where for any edge $< u_i, u_j > \in \mathcal{E}_t$, u_i is called *truster* and u_j is called *trustee*. The weight matrix of the trust graph is also termed as trust-matrix $\mathbf{T}_{n\times n}$. Each element of $\mathbf{T}$, $t_{i,j}$ can be defined as follows:

$$t_{i,j} = \begin{cases} x & \text{if } u_i \text{ trust on } u_j \text{ and } x \in \mathbb{R} \cap (0, 1] \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

All the symbols and notations used in this paper are described in below Table I.

TABLE I: Sympbols and Definitions used

Symbols	Definition
$\mathcal{U}$, $\mathcal{I}$	Set of Users, Set of Items
n, m	Number of Users i.e. $\lvert\mathcal{U}\rvert$, Number of Items i.e. $\lvert\mathcal{I}\rvert$
u_i, i_j	particular user, particular item
$\mathcal{I}_{u_i}$	Set of items rated by user u_i
$\mathcal{U}_{i_j}$	Set of users who have rated item i_j
$\mathbf{R}_{n\times m}$	Rating matrix
$\mathbf{S}_{n\times n}$	Similarity Matrix from rating data
$\mathbf{T}_{n\times n}$	Weight matrix of Trust Network
$r_{i,j}$	Rating provided by user u_i to item i_j
$\hat{r}_{i,j}$	Predicted rating for user u_i to item i_j
$s_{i,j}$	Similarity of user u_i with user u_j
$t_{i,j}$	Trust from user u_i to user u_j
$\mathcal{G}_t$	Trust Network
$\mathcal{N}(u_i)$	Set of nearest neighbors of user u_i
$\mathcal{N}_{i_j}(u_i)$	Neighbors of user u_i who all have rated item i_j
$\mathcal{N}_{\mathcal{G}_t}(u_i)$	Open neighborhood of user u_i in trust network $\mathcal{G}_t$
$\bar{r}_i$	Mean rating of users u_i
$d_{i,j}$	Length of shortest path from u_i to u_j in $\mathcal{G}_t$
d_{max} or L	Maximum allowable trust propagation length
l_k	Length of the k^{th} path from u_i to u_j in $\mathcal{G}_t$

III. Preliminaries

A. Neighborhood Based Recommendation

Two major divisions in Neighborhood Based methods are *User Based Collaborative Filtering* (UBCF) and *Item Based Collaborative Filtering* (IBCF). As the name suggests, UBCF finds similarity among users for a user, say u_i, chooses nearest neighbors $\mathcal{N}(u_i)$ among similar users and predict rating $r_{i,j}$ for unrated items i_j where $i_j \in \mathcal{I} \setminus \mathcal{I}_{u_i}$. Whereas, in case of IBCF, rating $r_{i,j}$ is predicted by calculating similar items $\mathcal{N}(i_j)$ of item i_j where $i_j \in \mathcal{I} \setminus \mathcal{I}_{u_i}$ and rated by user u_i. Here instead of going into their merits and demerits, we focus on UBCF as it is comparable with trust between users. For more details it can be referred to [1, 2].

1) Similarity Calculation: User-user similarity can be calculated as *Pearson correlation coefficient* between rating vector of users from their co-rated items. Similarity among all users form the similarity matrix $\mathbf{S}_{n\times n}$, where each element $s_{i,j}$ similarity between user u_i and u_j can be defined by the following Equation 3.

$$s_{i,j} = \frac{\sum_{i_k \in \lvert\mathcal{I}_{u_i} \cap \mathcal{I}_{u_j}\rvert} (r_{i,k} - \bar{r}_i) \cdot (r_{j,k} - \bar{r}_j)}{\sqrt{\sum_{i_k \in \lvert\mathcal{I}_{u_i} \cap \mathcal{I}_{u_j}\rvert} (r_{i,k} - \bar{r}_i)^2} \cdot \sqrt{\sum_{i_k \in \lvert\mathcal{I}_{u_i} \cap \mathcal{I}_{u_j}\rvert} (r_{j,k} - \bar{r}_j)^2}} \tag{3}$$

where $\bar{r}_i$, $\bar{r}_j$ denotes mean rating of user u_i and u_j respectively. Pearson similarity lies within [-1,1] and positive values denotes similarity whereas negative value implies dissimilarity. Another correlation based mechanisms is *Cosine similarity*.

Other than vector consideration, similarity can be computed as *Jaccard similarity* by considering item set of users.

2) Rating Prediction: In neighborhood method for rating prediction, an important part is neighbors selection after calculating similar users. Based on requirement, $\mathcal{N}(u_i)$ selection is done by threshold filtering on similarity or selecting *k*-nearest neighbors where k varies in different applications. Rating for a user u_i to an item i_j is predicted using the following Equation 4.

$$\hat{r}_{i,j} = \bar{r}_i + \frac{\sum_{u_k \in \mathcal{N}_{i_j}(u_i)} s_{i,k} \cdot (r_{k,j} - \bar{r}_k)}{\sum_{u_k \in \mathcal{N}_{i_j}(u_i)} s_{i,k}} \tag{4}$$

It is the weighted (weight as similarity) average of the rating provided by similar users of u_i who all have rated item i_j, i.e. $\mathcal{N}_{i_j}(u_i)$. Prediction is made unbiased from the neighbors by subtracting their mean rating and bias of u_i is imposed by adding $\bar{r}_i$.

B. Trust Based Recommendation

According to [15, 21] , from the sociological point of view, trust to someone is subjective, topic dependent, asymmetric and time sensitive. In case of any system, trust can be viewed as local or global. Trust also poses its propagative property in terms of transitivity i.e. if a user u_i trusts u_j and u_j trusts u_k then u_i trusts u_k. Due to this reason, we are able to infer the trust when it is not present explicitly.

1) Trust Inference: Earlier, different types of inference techniques are mentioned in Section I. Out of which we consider the path based approach where trust from user u_i to u_j depends on the shortest path between them and trust propagates along that path. Now, path based or distance based inference of trust has a major challenge of deciding how long this propagation will continue and how the decaying of trust will be. In [19, 20] trust from user u_i to user u_j is calculated as in the following Equation 5.

$$t_{i,j} = \begin{cases} \frac{d_{max} - d_{i,j} + 1}{d_{max}} & \text{if } d_{i,j} \leq d_{max} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $d_{i,j}$ is the length of the shortest path from u_i to u_j, and d_{max} is maximum allowable propagation distance between the users which is the average path length of the trust network $\mathcal{G}_t$ and can be given by Equation 6

$$d_{max} = \left\lceil \frac{\ln(|\mathcal{U}_t(\mathcal{G}_t)|)}{\ln(AverageDegree(\mathcal{G}_t))} \right\rceil \tag{6}$$

Here, linear decaying of trust is considered from [7] and decision for propagation length is taken by considering scale-free and small-world complex network property of $\mathcal{G}_t$ from [19].

2) Rating Prediction: Similar to the UBCF, the rating for a user u_i to an item i_j is predicted as the weighted average of ratings from the neighbors, $\mathcal{N}_{\mathcal{G}_t}(u_i)$, to whom u_i has shown trust and who all have also rated i_j. Here, the weight is inferred trust value from Equation 5. From [7, 19], the formula for rating prediction in trust-aware recommender system is given by the following Equation 7.

$$\hat{r}_{i,j} = \bar{r}_i + \frac{\sum_{u_k \in \mathcal{N}_{\mathcal{G}_t}(u_i)} t_{i,k} \cdot (r_{k,j} - \bar{r}_k)}{\sum_{u_k \in \mathcal{N}_{\mathcal{G}_t}(u_i)} t_{i,k}} \tag{7}$$

There are several other techniques of prediction, like weight can be the harmonic mean of trust and similarity, threshold based filtering where the threshold is on trust as proposed in [10, 20, 22]. In [20], the authors use reliability on top of trust. Here, we will limit our scope to the prediction using trust only as in Equation 7.

The existing method of trust inference has the following limitations:

- It has not considered the influence of the connecting users within a path when $d_{i,j} > 1$.
- As it is concerned only about $d_{i,j}$, it has not taken all possible path from u_i to u_j.
- Inferred trust $t_{i,j}$ in Equation 5 is discrete and the number of discrete trust value varies based on d_{max}.

To address the above-mentioned limitations we propose a new path based trust inference method.

IV. Proposed Method

We describe the proposed methods in two parts i) Inferred trust network construction, ii) Algorithms for trust computation. It takes user rating data and trust matrix as input and evaluates the method in terms rating prediction accuracy as output.

A. Inferred Trust Network Construction

As defined in Sec. II, in the trust network we can categorize two types of nodes *Truster* and *Trustee* for a trust relation i.e. the directed edge from former to later. For any edge $< u_i, u_j > \in \mathcal{E}_t(\mathcal{G}_t)$: $u_i \rightarrow u_j$ in trust network $\mathcal{G}_t$, node u_i is *Truster* and u_j is *Trustee*. Physically, if a user is trusted by many other users that indicate trustee user as more trustworthy in a global frame. In graph theoretic way, in a network, the node having greater in-degree is considered as *influential node*. To add this global trust influence of a particular user in the local trust inferencing method (refer Sec. III-B1) we define *degree of trustworthiness* for each user.

Definition (*Degree of Trustworthiness*)**.** *The degree of trustworthiness δ_i of a particular user u_i is proportional to the ratio of in-degree of that node with the maximum in-degree of the trust network $\mathcal{G}_t$. Mathematically,*

$$\delta_i = c \cdot \frac{indeg(u_i)}{(maxindeg(\mathcal{G}_t) + \epsilon)}, \epsilon \geq zero \tag{8}$$

where c is constant and ϵ denotes the factor of how much the maximum inferred trust will shift from the maximum trust value in original network. Higher the *Degree of Trustworthiness*, more *influential node* it will be. Global reputation of a user node in $\mathcal{G}_t$ is captured by the *Degree of Trustworthiness* and be used along with the local trust inference.

1) Inferred Trust Along a Path: Two users u_i and u_j where $u_i, u_j \in \mathcal{U}_t(\mathcal{G}_t)$ are connected by a path of length (say, l) and $2 \leq l \leq d_{max}$ and path is $< u_i, u_1, u_2, \ldots, u_{l-1}, u_j >$, $u_1, u_2, \ldots, u_{l-1} \in \mathcal{U}_t(\mathcal{G}_t) \setminus \{u_i, u_j\}$. The *influence* in inferred trust is captured by the *degree of trustworthiness* of the corresponding intermediate nodes in that path i.e. $\delta_1, \delta_2, \ldots, \delta_{l-1}$. The *influence* information is *implicit* here, as δ comes from the trust network itself and not from any additional source. As per our method, for above scenario the inferred trust is calculated by the following equation:

$$t_{i,j} = \underbrace{\left(\frac{d_{max} - l + 1}{d_{max}}\right)}_{\text{A}} + \underbrace{(\delta_1 + \delta_2 + \cdots + \delta_{l-1})}_{\text{B}} \quad (9)$$

where part A is coming from local trust and part B is influence due to intermediate nodes in the path from u_i and u_j.

Lemma. *The value of c will always be less than equal to $\frac{1}{d_{max}}$ and greater than 0 (i.e. $0 < c \leq \frac{1}{d_{max}}$).*

Proof. As per Equation 9, it can be easily said that part A is constant for some l, $2 \leq l \leq d_{max}$ and part B varies according to δ value of the intermediate nodes. Now value of $t_{i,j}$ will be maximum when $\forall u_k \in \{u_1, u_2, \ldots, u_{l-1}\}$, $indeg(u_k) = maxindeg(\mathcal{G}_t)$. Then for $\epsilon = 0$, Equation 9 will be

$$\begin{aligned}
[t_{i,j}]_{max} &= \underset{\delta_k}{argmax} \left[\left(\frac{d_{max} - l + 1}{d_{max}}\right) + (l-1) \cdot (\delta_k)\right] \\
&= \left(\frac{d_{max} - l + 1}{d_{max}}\right) + (l-1) \cdot \left(c \cdot \frac{indeg(u_k)}{maxindeg(\mathcal{G}_t)}\right) \\
&= 1 - \frac{(l-1)}{d_{max}} + (l-1) \cdot c \\
&\qquad [\because indeg(u_k) = maxindeg(\mathcal{G}_t)]
\end{aligned}$$

Now, as per Equation 2,

$$\begin{aligned}
[t_{i,j}]_{max} \leq 1 &\Rightarrow 1 - \frac{(l-1)}{d_{max}} + (l-1) \cdot c \leq 1 \\
&\Rightarrow (l-1) \cdot c \leq \frac{(l-1)}{d_{max}} \\
&\Rightarrow c \leq \frac{1}{d_{max}}
\end{aligned}$$

Again value of $t_{i,j}$ will be minimum when $\forall u_k \in \{u_1, u_2, \ldots, u_{l-1}\}$, $indeg(u_k) = 1$ [as, u_k lies within the path from u_i to u_j, so u_k must have at least 1 incoming edge]. Then for $\epsilon = 0$ minimum vale of $t_{i,j}$ will be,

$$\begin{aligned}
[t_{i,j}]_{min} &= \underset{\delta_k}{argmin} \left[\left(\frac{d_{max} - l + 1}{d_{max}}\right) + (l-1) \cdot (\delta_k)\right] \\
&= \left(\frac{d_{max} - l + 1}{d_{max}}\right) + (l-1) \cdot c \cdot \frac{1}{maxindeg(\mathcal{G}_t)}
\end{aligned}$$

Now, from the above equation, it is trivial that,

$$\begin{aligned}
&[t_{i,j}]_{min} > \left(\frac{d_{max} - l + 1}{d_{max}}\right) \\
\Rightarrow (l-1) \cdot c \cdot \frac{1}{maxindeg(\mathcal{G}_t)} > 0 &\quad \Rightarrow c > 0
\end{aligned}$$

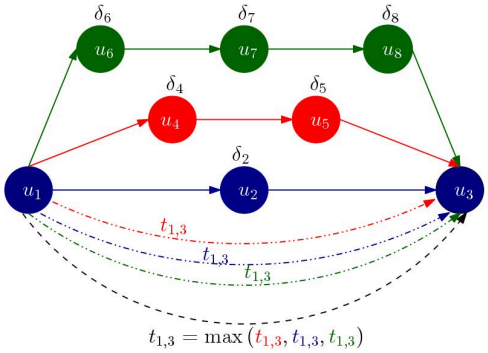


Fig. 1: Example of Trust Inference from various paths

Hence, it is proved that for calculating $t_{i,j}$ by Equation 9 c will always lie within $(0, \frac{1}{d_{max}}]$. □

If c is chosen towards upper bound then for the cases where a majority of the nodes are having lower in-degree, achieve significant δ sum in the inferred trust. Whereas, if c is chosen towards lower bound then for the cases where most of the smaller in-degree users are connected to the higher in-degree users, the inferred trust will have variation according to in-degree of the intermediate users. Along with this, for the case $c = \frac{1}{d_{max}}$, choosing ϵ as zero maps the maximum inferred trust to 1. Now, to distinguish from the user provided explicit trust, the ϵ value can be set as per designers choice. Choosing a small value of ϵ serves the purpose here in both the cases.

2) Inferred Trust Between Two User Nodes: Now consider the situation, there exist multiple paths (say K) from u_i to u_j with corresponding length $l_1, l_2, \ldots, l_K$ ($\forall l_k \in \{l_1, l_2, \ldots, l_K\}, l_k \leq d_{max}$). Then by intuition, we are supposed to trust most trusted user so we take trust coming from the most influential path as inferred trust $t_{i,j}$. Mathematically, it is the maximum trust coming from Equation 9 for all K possible path from u_i to u_j and it is given by the following Equation 10,

$$t_{i,j} = \underset{l_k \in \{l_1, \ldots, l_K\}}{\text{MAX}} \left[\frac{d_{max} - l_k + 1}{d_{max}} + \delta_1 + \delta_2 + \cdots + \delta_{l_k - 1}\right] \quad (10)$$

One example scenario for Equation 9 and 10 is demonstrated by Figure 1, where from u_1 to u_3 can be reached by three possible paths shown in different colors and $d_{max} = 4$. By Equation 5, shortest path is blue one via u_2 and $t_{1,3} = 0.75$. Now by Equation 9, $t_{1,3}$ inferred trust from blue path will be greater than 0.75 i.e. $0.75 + \delta_2$. Similarly, $t_{1,3} = 0.5 + \delta_4 + \delta_5$ and $0.25 + \delta_6 + \delta_7 + \delta_8$ by red and green path respectively. So, from Equation 10 our proposed trust is the maximum $t_{1,3}$ coming from blue, red and green path. In extreme case, it may choose green path if u_6, u_7, u_8 all are much more influential than u_2, u_4, u_5, i.e. $(\delta_6 + \delta_7 + \delta_8) \to 0.75$.

B. Algorithm for Trust Computation

We propose here an algorithm for computing all possible path of length $2, \ldots, L$ (or d_{max}) in a graph for our problem. We take trust network as its adjacency list or dictionary based structure where for each user node (i.e. dictionary item) it has a list of its out-degree nodes or *trustee* in a sorted fashion according to their index. Now length 1 path is very trivial from its out-degree list of the graph.

1) Data Structure for Path Computation: We use a data structure to store and compute all possible path list from each vertex up to length L as a list of list format, shown in Figure 2.

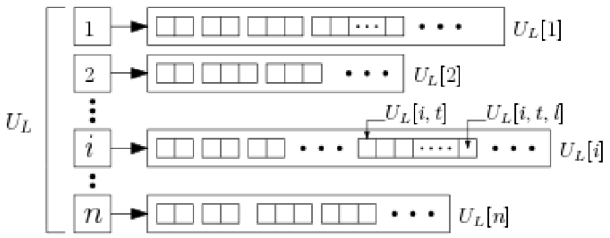


Fig. 2: List of path list of each node in the graph $\mathcal{G}_t$

Operations can be performed on U_L are,

(a) `APPEND`: performs on a path object $U_L[i,t]$, takes node number(j) as input parameter and returns a path by appending j to the path t starting from node i.

(b) `ADD`: performs on a list item $U_L[i]$ and adds a path (list of nodes) in the path-list of node i

Algorithm 1: Algorithm for calculating all possible paths up to length L in trust network $\mathcal{G}_t$

Data: $U[1 \ldots n]$ *Trustee* user list per user (Out-degree vertex list of $\mathcal{G}_t$), integer L
Result: $U_L[1 \ldots n]$ List of all possible path in $\mathcal{G}_t$ up to length L per user

1 $U_L \leftarrow U$;
2 **for** *each node* $i \in \{1 \ldots n\}$ **do**
3 **for** $l \in \{2 \ldots L\}$ **do**
4 **for** *each list item* t *in* $U_L[i]$ **do**
5 $x = U_L[i, t, l-1]$;
6 **for** $j \in U[x]$ **do**
7 *newpath* $\leftarrow NULL$;
8 **if** $j \in U_L[i,t]$ *or* $j == i$ *or* $j \in U[i]$ **then**
9 **continue**;
10 **else**
11 *newpath* = $U_L[i,t]$·`APPEND` (j) ;
12 $U_L[i]$·`ADD` (*newpath*) ;

2) Algorithm for Path Finding: In Algorithm 1, the **for loop** at line no. 2 can be parallelized by processing each node at the same time in multiple processes. In every step l, the algorithm uses $(l-1)$ level state of the data structure U_L i.e. for a particular node after computing path of length $(l-1)$ it tries to extend that to length l by looking its out-degree vertices. The cycle is removed by the **if** condition at line no. 8. The time complexity of the algorithm can be analyzed by assuming $\mathcal{G}_t$ as regular graph of degree d, $d << n$, to have loose upper bound. For a particular node at $l = 2$, **for loop** of line no. 4 runs at most d^2 times. Similarly, at $l = 3$ for d^3 times and so on if at every step the extended path does not form cycle. Hence, **for loop** of line no. 3 runs at most total $\mathcal{O}(d^L)$ times asymptotically $[d^2 + d^3 + \cdots + d^L = \sum_{l=1}^{L-1} d^2 \cdot d^{(l-1)} = \frac{d(d^L - d)}{d-1} = \mathcal{O}(d^L)]$. So, running time of Algorithm 1 is $\mathcal{O}(nd^L)$.

Algorithm 2: Algorithm for Trust Computation

Data: $U_{in}[1 \ldots n]$ *Truster* user list per user (In-degree vertex list of $\mathcal{G}_t$), $U_L[1 \ldots n]$, c, ϵ, d_{max}
Result: Weight Matrix **T** of inferred trust network

1 T $= AdjacencyMatrix(\mathcal{G}_t)$; `// T[1 ...n][1 ...n]`
2 $\delta \leftarrow 0$;
`/* δ[1...n] stores degree of trustworthiness per node */`
3 $maxDegree =$ `MAX` (`LENGTH` $(U_{in}[1 \ldots n])$) ;
4 **for** *each node* $i \in \{1 \ldots n\}$ **do**
5 $\delta[i] = c$·`LENGTH` $(U_{in}[i])$ /$(maxDegree + \epsilon)$;
6 **for** *each node* $i \in \{1 \ldots n\}$ **do**
7 **for** *each path* k *in* $U_L[i]$ **do**
8 $L =$ `LENGTH` $(U_L[i,k])$;
9 $j = U_L[i,k,L]$;
10 $prevTrust = (d_{max} - L + 1)/d_{max}$;
11 $deltaSum = 0$;
`/* for each intermediate node U_L[i,k,l] */`
12 **for** $l \in \{1 \ldots L-1\}$ **do**
13 $deltaSum = deltaSum + \delta[U_L[i,k,l]]$;
14 $newTrust = prevTrust + deltaSum$;
15 **if** *newTrust* $>$ *T[i][j]* **then**
16 $T[i][j] = newTrust$;

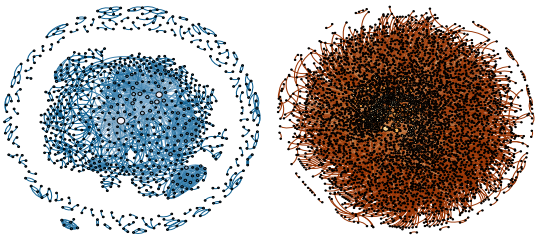
3) Algorithm for Inferred Trust Calculation: Algorithm 2 has two parts, *degree of trustworthiness* calculation for each node from line no. 2 to 5 by the Equation 8 and trust computation from line no. 6 to 16 by the Equation 10. At line no. 3, it computes the maximum in-degree by computing the length of in-degree node list for each node. *Degree of trustworthiness* is calculated in $\theta(n)$ time. For the next part there can be at most d^L number of paths which leads to that the **for loop** at line no 7 will run with the upper bound $\mathcal{O}(Ld^L)$ $[1d^2 + 2d^3 + \cdots + (L-1)d^L = \sum_{l=2}^{L} (l-1)d^l \approxeq Ld^L = \mathcal{O}(Ld^L)]$. So, running time of Algorithm 2 is $\mathcal{O}(nLd^L)$. Similar to Algorithm 1, Algorithm 2 can be parallelized by computing **for loop** at line no. 6 for each node in parallel.

Using this inferred trust network for recommendation, rating $r_{i,j}$ for user u_i to item i_j is predicted using the method

discussed in Sec. III-B2.

V. Dataset Exploration

We evaluated the performance of our proposed method on two real datasets FilmTrust and CiaoDVD. Both datasets are taken from https://www.librec.net/datasets.html. The statistics of the original dataset is given in Table II and the dataset collection description can be found in [23, 24] for FilmTrust and in [25] for CiaoDVD. The visualization of Trust Network is shown in Figure 3 .



(a) FilmTrust (b) CiaoDVD

Fig. 3: Visualization of Original Trust Network

From Figure 3, it can be easily seen that there are several components in the graph mainly consisting two or three vertices. Using inference method of the equation 10 those cannot be connected to the main giant component so only the main component of trust graph is considered for trust propagation. FilmTrust dataset consists of 610 users with 1604 trust statements whereas CiaoDVD dataset holds 4562 users with 40073 trust statements in its main component of the trust graph.

For the analysis and comparison with existing methods, we have selected the common users present in both rating and trust data. The characteristics of the datasets are given in Table III. Before applying the trust propagation length which is based on the scale-free network property as per [19, 20] we have shown the Power-Law degree distribution of the trust network generated by common users described in the Table III.

VI. Experimental Evaluation

To evaluate the recommendation accuracy we have conducted an off-line experiment using *leave-on-out* mechanism. Each rating value of the datasets is predicted by removing that entry. The accuracy of prediction is measured in terms of *Mean Absolute Error* (MAE) and *Mean Absolute User Error* (MAUE) value. MAE is an average of the absolute difference for all the predicted ratings. It can be described by the Equation 11.

$$\text{MAE} = \frac{\sum abs(\hat{r}_{i,j} - r_{i,j})}{\text{no. of ratings}} \tag{11}$$

MAUE can be calculated for each user from the absolute difference of predicted rating with its actual for the items purchased by that user only. MAUE for user u_i can be described by the Equation 12.

$$\text{MAUE}_{u_i} = \frac{\sum_{i_j \in \mathcal{I}_{u_i}} abs(\hat{r}_{i,j} - r_{i,j})}{|\mathcal{I}_{u_i}|} \tag{12}$$

The actual MAUE for the recommender system is an average of all the MAUE per user and defined by the Equation 13.

$$\text{MAUE} = \frac{\sum_{u_i \in \mathcal{U}} \text{MAUE}_{u_i}}{|\mathcal{U}|} \tag{13}$$

We have compared our proposed methods with User Based Collaborative Filtering (UBCF) and existing trust based Collaborative Filtering (Trust-1). Our proposed inferred trust network based on *degree of trustworthiness* is referred as Trust-2. For UBCF, we have chosen *Pearson Correlation Coefficient* as similarity metric described in Sec. III-A1 and used all positive similar user as nearest neighbors. On the other hand, here trust Base CF will depend on the maximum trust propagation path length d_{max} which is average path length in the network and equivalent to the Equation 6. For Filmtrust dataset, average-degree is 5.4642, hence $d_{max} = \lceil ln(530)/ln(5.4642) \rceil \approx 3.6934 = 4$ and for CiaoDVD dataset, average-degree is 16.7212, hence $d_{max} = \lceil ln(2687)/ln(16.7212) \rceil \approx 2.8034 = 3$. To compute *degree of trustworthiness*, we have taken ϵ value as 1 and c as $1/d_{max}$. The density of the inferred trust network becomes 21.5829% and 13.8014% for FilmTrust and CiaoDVD datasets respectively.

The results of the experiments are shown in the Table IV.

For CiaoDVD dataset, both MAE and MAUE values have improved from all the other methods in Trust-2. In case of CiaoDVD, the gradual improvement from UBCF to Trust-1 and then Trust-2 is achieved due to the construct of the datasets. The density of rating dataset is very low and the having heavy tailed power-law in-degree distribution for trust network. Whereas, for FilmTrust dataset, the MAE value is not improved for trust based method from UBCF but Trust-2 achieves lower MAE than Trust-1. The reason for this is that many of the cold start users in rating data are zero out-degree users in trust data. There are 258 users covering 51% of the total users in the system of who has rated less than 20 items and out of these 70 users (27%) are zero out-degree. Neighbors for zero out-degree users is not recoverable by this inference. The MAUE for users of different category based on a number of items purchased is discussed further focusing mainly on the cold start users.

For FilmTrust dataset, the quartile of item count per user is [2, 23.25, 44.5, 85.75, 244]. Users are categorized on item range of step size 10 up to item count 100 and later step of size 50. The MAUE for each category of users is noted in the Table V by highlighting minimum error among all the methods. The Bar Plot for the same is shown in the Figure 5. It can be easily seen that Trust-2 gives the best result for the majority of the

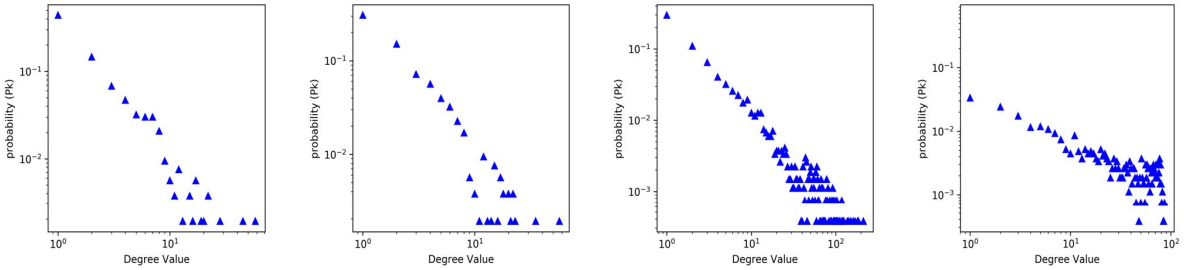
TABLE II: Statistics of Original Datasets

Dataset	#users	#items	#rating	rsize	csize	density	scale / step
FilmTrust	1508	2071	35497	23.54	17.14	0.014263	[0.5 ∼ 4] / 0.5
CiaoDVD	17615	16121	72665	4.12	4.50	0.000973	[1 ∼ 5] / 1

(a) Rating Dataset

Dataset	#users	#trust	density
FilmTrust	874	1853	0.002428
CiaoDVD	4658	40133	0.001850

(b) Trust Dataset



(a) (b) (c) (d)

Fig. 4: **Power-Law Degree Distribution on Log-Log Scale of the Trust Network formed by the common user present in both rating and trust dataset, (a)FilmTrust Dataset for In-Degree with $\gamma_{in} = 1.6347$, (b)FilmTrust Dataset for Out-Degree with $\gamma_{out} = 1.6849$, (c)CiaoDVD Dataset for In-Degree with $\gamma_{in} = 2.0672$, (d)CiaoDVD Dataset for Out-Degree with $\gamma_{out} = 2.9690$.**

TABLE III: Statistics of Datasets used for Experiments

Dataset	#users	#items	#rating	#trust	rating density	trust density
FilmTrust	530	1888	14272	1448	1.4263%	0.5164%
CiaoDVD	2687	13023	34075	22465	0.0974%	0.3113%

TABLE IV: Results of Experiment for Evaluation

	Dataset	UBCF	Trust-1	Trust-2
MAE	FilmTrust	0.6524	0.6674	0.6591
	CiaoDVD	0.7980	0.7906	0.7760
MAUE	FilmTrust	0.6673	0.6784	0.6671
	CiaoDVD	0.7956	0.8060	0.7808

cases when users have given few number of ratings. Whereas, UBCF performs better for the heavy rated users.

For CiaoDVD dataset, the quartile of item count per user is [2, 33, 67, 137, 1106]. Here, more than 50% of users are cold start users rated less than items 10 items. Similar to FilmTrust dataset, users are categorized had MAUE values for each category is listed in the Table VI. The Bar plot of the same is shown in the Figure. 6. Here, for all the cases MAUE values are less in trust based method from its counterpart. Also, Trust-2 outperforms in all the cases due to very low *rsize* of rating dataset [II].

VII. Conclusion and Future Scope

In this work, we have used the graph theoretical approach for trust inference between two users and considered the influence of a user by *degree-of-trustworthiness*. This initiates of choosing maximum trust gaining path from all the available paths between two nodes in the trust network. We have shown

TABLE V: MAUE per User Category of FilmTrust

Item Count(#users)	UBCF	Trust-1	Trust-2
< 10 (132)	0.754849	0.768931	**0.748121**
11 ∽ 20 (126)	0.629907	0.633991	**0.624155**
21 ∽ 30 (84)	**0.646646**	0.656887	0.649426
31 ∽ 40 (45)	0.615494	0.609827	**0.603429**
41 ∽ 50 (64)	**0.589303**	0.61468	0.610115
51 ∽ 60 (26)	**0.687864**	0.71394	0.705644
61 ∽ 70 (4)	**0.759323**	0.777824	0.77099
71 ∽ 80 (3)	0.632733	0.622812	**0.611919**
81 ∽ 90 (3)	0.802189	**0.751122**	0.757363
91 ∽ 100 (5)	0.685747	0.68699	**0.660804**
101 ∽ 150 (9)	**0.721646**	0.753772	0.742574
151 ∽ 200 (4)	**0.810605**	0.840092	0.830762
> 200 (2)	**0.582429**	0.622773	0.619766

TABLE VI: MAUE per User Category of CiaoDVD

Item Count(#users)	UBCF	Trust-1	Trust-2
< 10 (1340)	0.805566	0.82454	**0.793702**
11 ∽ 20 (264)	0.759168	0.748781	**0.734414**
21 ∽ 30 (109)	0.763291	0.758656	**0.748578**
31 ∽ 40 (54)	0.757964	0.751031	**0.741331**
41 ∽ 50 (32)	0.794759	0.784695	**0.779448**
51 ∽ 60 (22)	0.838443	0.826305	**0.820648**
61 ∽ 70 (16)	0.868752	0.843946	**0.836588**
71 ∽ 80 (9)	0.652386	0.650396	**0.61914**
81 ∽ 90 (10)	0.789419	0.7867	**0.773447**
91 ∽ 100 (8)	0.731232	0.740631	**0.709727**
101 ∽ 130 (10)	0.776505	0.766072	**0.757354**
131 ∽ 160 (8)	0.869902	0.840406	**0.836809**
161 ∽ 200 (6)	0.892034	0.867792	**0.863434**
201 ∽ 300 (11)	0.847066	0.814724	**0.799766**
> 300 (11)	0.806224	0.795353	**0.783272**

that this incorporation of influence improves the recommendation accuracy. Further, we have shown that the Trust-2 achieves better improvement among other two, mainly for cold start users. Here, we have not studied how hit-rate is behaving by

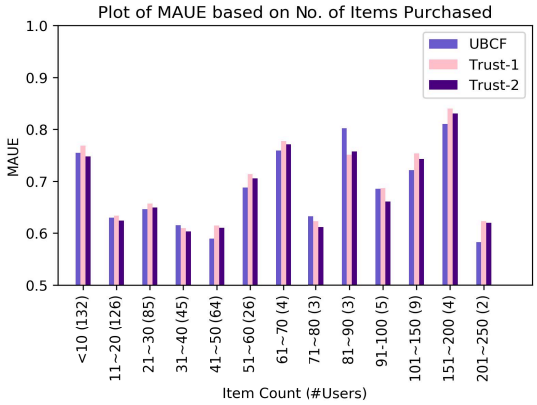


Fig. 5: FilmTrust

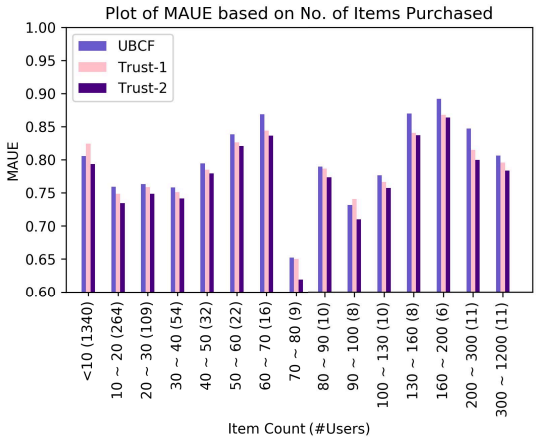


Fig. 6: CiaoDVD

trust incorporation. In future, that can be analyzed. Also, we can add distrust statement in this method and do the further treatment of it. There is another scope of improvement is to design better algorithm in this problem space for all possible path computation to minimize space and time complexity.

ACKNOWLEDGMENT

The work has been financially supported by the project *E-business Center of Excellence* funded by Ministry of Human Resource and Development (MHRD), Government of India under the scheme of *Center for Training and Research in Frontier Areas of Science and Technology (FAST)*, Grant No. F.No.5-5/2014-TS.VII. First author would also like to thanks Suman Banerjee for insightful comments.

REFERENCES

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[2] X. Ning, C. Desrosiers, and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender systems handbook*. Springer, 2015, pp. 37–76.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[4] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 880–887.

[5] S. Khusro, Z. Ali, and I. Ullah, *Recommender Systems: Issues, Challenges, and Research Opportunities*. Singapore: Springer Singapore, 2016, pp. 1179–1189. [Online]. Available: https://doi.org/10.1007/978-981-10-0557-2_112

[6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[7] P. Massa, P. Avesani *et al.*, "Trust-aware collaborative filtering for recommender systems," *CoopIS/DOA/ODBASE (1)*, vol. 3290, pp. 492–508, 2004.

[8] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 287–296.

[9] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," *Trust Management*, pp. 93–104, 2006.

[10] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th international conference on Intelligent user interfaces*. ACM, 2005, pp. 167–174.

[11] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 2007, pp. 17–24.

[12] J. Golbeck, B. Parsia, and J. Hendler, "Trust networks on the semantic web," *Cooperative information agents VII*, pp. 238–249, 2003.

[13] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *Proceedings of the 2005 ACM symposium on Applied computing*. ACM, 2005, pp. 1589–1593.

[14] J. A. Golbeck, "Computing and applying trust in web-based social networks," Ph.D. dissertation, 2005.

[15] W. Jiang, G. Wang, M. Z. A. Bhuiyan, and J. Wu, "Understanding graph-based trust evaluation in online social networks: Methodologies and challenges," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, p. 10, 2016.

[16] P. Gao, H. Miao, J. S. Baras, and J. Golbeck, "Star: Semiring trust inference for trust-aware social recommenders." in *RecSys*, 2016, pp. 301–308.

[17] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings." in *Aaai*, 2015, pp. 123–129.

[18] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[19] W. Yuan, D. Guan, Y.-K. Lee, and S. Lee, "The small-world trust network," *Applied Intelligence*, vol. 35, no. 3, pp. 399–410, 2011.

[20] P. Moradi and S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7386–7398, 2015.

[21] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision support systems*, vol. 43, no. 2, pp. 618–644, 2007.

[22] Y. Du, X. Du, and L. Huang, "Improve the collaborative filtering recommender system performance by trust network construction," *Chinese Journal of Electronics*, vol. 25, no. 3, pp. 418–423, 2016.

[23] J. Golbeck, J. Hendler *et al.*, "Filmtrust: Movie recommendations using trust in web-based social networks," in *Proceedings of the IEEE Consumer communications and networking conference*, vol. 96, no. 1, 2006, pp. 282–286.

[24] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, 2013, pp. 2619–2625.

[25] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Etaf: An extended trust antecedents framework for trust prediction," in *Proceedings of the 2014 International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2014, pp. 540–547.