An Algorithm for Predicting Local Trust based on Trust Propagation in Online Social Networks

Munmun Bhattacharya
Department of Information Technology
Jadavpur University, Kolkata

Nashreen Nesa
Department of Information Technology
Jadavpur University, Kolkata

ABSTRACT

The objective of online social networking sites is to make it possible to connect people who share common interests and pursuits across different geographical locations. With this, the concept of trust also comes into perspective as the participants reveal a great quantity of personal information in the Web environment. This work adopts web-based Social Networks as the principle means for studying trust. Goal of this work is to find ways to utilize the structure of social graph and the trust relationships between them to accurately deduce how much two individuals that are not directly connected might trust one another. This paper presents an algorithm for inferring trust propagation between indirectly connected individuals in the network by the use of weighted trust ratings along the shortest and the most trusted path. The accuracy of this algorithm in predicting propagated trust is calculated and compared with that of simple average strategy and the multiplicative strategy algorithm [17]. This algorithm is tested with five real-world trust datasets and tried to discover that there exists a significant strong positive correlation between direct trusts and the corresponding propagated trusts obtained through this approach.

General Terms

Computer Science, Information Technology, Online Social Network.

Keywords

Social network, local trust, global trust, propagation, trusted path, trustworthy.

1. INTRODUCTION

With the unprecedented expansion of social network services, the need for recognizing trustworthy people has become a chief concern to protect the participants' huge amounts of personal information from being tainted by unpredictable users. Trust, is a very important part of the daily human life. It is used every day in one form or another. For instance, when one person walk on the road and trust cars not to crash him by accident, or sometimes buy eateries from roadside vendors and trust them not to poison the food that they serve, and even trust that the televisions will work every time when it will be switch on. Moreover, one person sometimes conducts business with unknown people and faced the difficult task of making judgments that involves risk in an online environment. For example, customers in any e-commerce site read reviews of items they intend to buy, and are often faced with the situation of deciding whether the reviews are posted by website representatives pretending to be customers or indeed are real buyers. As a result, the topic of trust in Internet is receiving much attention in social networks. Functioning societies rely heavily on trust among members and it is natural to expect the same to be true in online communities.

Communications on the web are complex, as they involve interactions with people who may even be strangers. As a result, a person on the web is often in a dilemma about how much to trust another person either for personal or for professional reasons. In that case, there is no personal history, on which to make an assumption or to provide a rating. In real life, people may ask their friends or friends of friends for information regarding the trustworthiness of a stranger, but online, a stranger may be very socially distant and finding the people to ask about trustworthiness can be a lot of work. Thus, a method that can accurately infer how much one person will trust another will be very useful. This paper adopts web-based social networks as the foundation for studying trust. We look at instances where trust is integrated into a social network. The goal of this work is to find the ways to utilize the structure of social networks and the trust relationships within them. If two individuals are not directly connected, a trust inference mechanism makes use of the paths that connect them in the social network, and the trust values along those paths, to come up with a recommendation about how much two persons who are not directly connected might trust one another.

This paper presents an algorithm for inferring trust to determine local trust value for the above said situation. The main aim of this paper is to illustrate how a study of the trust associations in online social networks can assist in developing methods for inferring trust values. These estimated values can then be integrated into applications, which can significantly enhance the user experience. The driving reason for choosing to work with web-based social networks is motivated by the fact that they form a large, publicly available dataset with remarkable interest from the general public. Then this algorithm is tested with the publicly available datasets for trust calculation. For proving more accuracy of our algorithm, we do a real life survey in the department of Information Technology, Jadavpur University, where 2nd year Under Graduate students participate. More information will available later.

The rest of the paper is organized as follows: The next section outlines the literature survey of previous works. Section 3 describes the problem formulation. In section 4, contribution of the authors is discussed. The experimental design is described in section 5. The details experimental results and analysis is given in section 6. Conclusion and future works are presented in section 7 and 8.

2. LITERATURE REVIEW

Trust in online social networks is an important issue and thus is widely studied and implemented. Several researchers have devised algorithms for successful trust calculation based on their own theories. A number of algorithms [12, 13, 33, 17, 21, 20, 15] deals with trust propagation between the source and the sink in the network to infer trust based on the user's perspective. These algorithms are called local algorithms and are personalized for each user. TidalTrust [12] and MoleTrust [13] are perhaps the most widely cited works in this domain. Both use a weighted average strategy to compute an inferred trust value for the sink. The TidalTrust algorithm searches for the shortest paths from the source to the sink. There could be several shortest paths with the same path length hence only the strongest paths among the shortest paths are considered in trust computation. A major drawback of this algorithm is that it is not always efficient as the longer chain path may also contain valuable information which should not be neglected. MoleTrust algorithm [13] takes care of this drawback with a slight variation. Although the MoleTrust algorithm is also based on shortest-path distance from the source user, but here, only those users with a propagative distance of less or equal to the trust propagation horizon are considered. Horizon is the maximum distance from the source user to which trust can be expected to propagate and is independent of any specific user and item. Although a disadvantage of this approach is its high time complexity that comes with it.

Hasan, Brunie and Pierson [17] evaluated the efficiency of iterative multiplication strategy for trust propagation where trust of a path is calculated by multiplying the direct trust values along the path provided the values are in the range [0-1]. A strong positive linear correlation was shown to exist between the direct and propagated trust values. A disadvantage here is that this strategy is not feasible for other range of trust values (eg. [1-5]).Chakraborty et al. [33] introduced a decay factor with increase in path length for trust propagation based on simple multiplicative strategy. His work integrates the path length and decay of direct trust values along the trust path into trust propagation algorithms. Kim et al. [21] compared and estimated how the length of trusted chain and aggregation techniques affect the accuracy of the calculated trust. They proposed four strategies to calculate this. Two of those strategies use shortest paths to calculate the predicted trust. The other two makes use of all paths for inferring trust. The two aggregation strategies implemented are the weighted mean aggregation and the min–max aggregation. Among those four, they discovered that the weighted mean aggregation strategy that gives the most accurate result and hence were chosen as the optimal strategy. Cho et al. [20] defined the trust availability and path reliability terminologies to determine the optimal length of a trust path that could generate the most accurate trust. One recent work includes that of Hamdi et al. [15]. In this paper, they introduce TISoN(Trust Inference for Social Networks) to infer trust among users in OSNs. In addition they also propose a trust path searching algorithm TPS and a trust inference method TIM. TPS involves defining neighbours with priority based on their direct trust degrees and then selecting trusted paths while controlling the path length.

Several approaches have been proposed by numerous authors for evaluating reputation for P2P networks. One extensively cited work for P2P systems is EigenTrust [10]. The EigenTrust algorithm works with a design that is similar to the PageRank algorithm [8], implemented by Google for ranking the relevance of web pages in response to a Google search. A direct trust rating is created between two peers based on their historical performance. The algorithm creates a matrix representation of the trust values within the system and over a string of iterations it converges to a globally accepted trust rating of each peer. EigenTrust has been shown to be highly resistant to attack. Aringhieri et al. [6] have proposed another trust model for P2P networks where they make use of weighted mean technique to aggregate reputations obtained from other peers. They used the fuzzy logics to calculate the trust value and the trust value is considered in the range of [0,1].

Other related works deals with reputation and global trust calculations. A model proposed by Abdul-Rahman et al. [9] identified the concept of trust in virtual society, where trust is measured based on observed experiences and reputation. However, their model is insufficient to be considered applicable in broader scope. Mui et al. [4] proposed a computational model based on sociological understanding that can be used to measure agent's trust and their reputation scores. Pujol [27] proposed a method for calculating the reputation of a given member in a society or in a social network by making use of PageRank™ algorithm. Wang Y et al [18] employs the reputation and trust value to classify the user into different classes and then recommend different services to them. Caverleea et al. [24] presented the SocialTrust framework which gives a network-wide perspective of all users in OSNs that is based on their trust. Liu et al. in their work [19], calculates trust value between the users and clusters by predicting the information propagation area of a specific message. Initially the message is converted to an eigenvector, and then the similarity with user as well as with the cluster is calculated. Finally, the number of users that receive the message is estimated, together with the user's trust value in the cluster. However, its drawback is that this model needs a lot of calculation, which is a waste of time. Advogato's reputation [14] and Appleseed algorithm [29] are popular algorithms that assign a global trust score to each member of a given community.

Researches on recommendation try to predict a user's opinion on a specific item, so as to recommend proper item to the user. De Meo et al. [22] proposed a general approach which operates in a social internetworking framework instead of on a single social network. The work considers both explicit and implicit relationships and takes into account both local and global information to recommend similar users, resources, and social networks to a user. Kim et al. [23] proposed a framework for trust prediction in social networks that is based on rating based experience sharing platform. A degree of trust is calculated based on users' topic-based proficiency and preferences with the help of users' feedback rating data.

Works in [26, 28, 34] utilizes the concept of ant colony for trust calculation. In [26] Bedi and Sharma proposed Trust based Ant Recommender System (TARS) which effectively produces recommendations by incorporating a notion of dynamic trust between users and choosing a small and best neighborhood based on biological metaphor of ant colonies. In [28] the authors proposed an algorithm (Trust-ACO) to calculate trust in online social network where the trust path and trust cycle is predicted with the help of ant colony optimization (ACO). Web-based system user interface hybrid recommendation method is presented in [34] where ant colony metaphor is used for selecting the most optimal path in the user interface graph.

3. PROBLEM FORMULATION

- User: we define U as the set of all the users associated with a given web environment.

- Relation: we define a relation, T = {(i,j): i,j ϵ U}.The relation T represents the trust relation between two users in the given web environment.

- Trust Graph: a Trust Graph is defined as a weighted directed graph, $G = (U,)$. The users form the vertices of the graph. The trust connections between the members of set U given as ordered pairs in the set T form the edges of the graph. A directed edge from i to j, implies (i,j) or i trusts j.

- Weight: a weight is associated with every edge (i,j) in the graph, which represents the quantity of trust that entity i holds for entity j.The weight associated with an edge (i, j) is given as the function t(i, j); t: T → X.
- Trust ratings: the set X is the range of possible trust ratings. The range X is any real number in the range [1,n] where the lowest trust value is indicated as 1 and the highest trust value is indicated as n.
- Path: a path P = <$u1$,$u2$,$u3$,…um,ud> from a user $u1$ to a user ud is said to exist if $u1$,$u2$,$u3$ …um,ud∈U and ($u1$,$u2$), ($u2$,$u3$) ,......, (um,ud) ∈ T.

Now, if there are two users A and B, and user B is a stranger to A, but user A wants to know how much reliable user B is. This algorithm try to find the recommended (estimated) rating of the user A for the user B.

Trust metrics can be categorized into global and local metrics from a personalization perspective. Global trust metrics predict the same trust of a given user for all users, i.e.; predicting a global trust value for each user. In some applications, global trust value is also referred to as reputation scores or status scores. To calculate global trust scores, global trust metrics usually need to access the whole trust networks. Formally, a global trust metric aims to compute a global trust score pi for each user ui in a given trust network. In reality, it may be difficult to reach an agreement among the users regarding another user. Users may have completely different opinions about the same users. i.e., users' trust opinion may be personalized. Hence, local trust metrics provide a trust measure based on the point of view of the evaluating user and they calculate a trust value Ti,j for each pair of users (ui,uj) without any explicit trust relations between them.

- Global trust metrics focus on the nodes (or the users) of the network and compute a trust value for each node
- Local trust metrics focus on pairs of nodes (or users) without explicit trust relations in the network. A local trust metric suggests some pairs of users with trust relations such as trust relation from u4 to ui.

Given a social network, information about trust can be provided to users in many ways. The goal is generally the same: recommend to one node how much to trust another node in the network. The way this is done varies. The goal of this paper is to recommend what trust rating one person might want to give another, unknown person if there were a connection. The trust recommendations are very much like predictive recommendations made by a recommender system. Designing an algorithm for the task of predicting trust values must be guided by the properties of trust.

As trust is personal and judgments vary between two people; with the help of a local trust calculation algorithm, we can improve the correctness of the results. For instance, if a person wants a recommendation about how much to trust another person, an algorithm that simply aggregates all the values in the system can be expected to give an insufficient result. This is because the result reflects the opinion of the population as a whole, and is not a recommendation to the particular individual. Since many people may have contrasting opinions about the trustworthiness of the same person.

3.1 Multiplication Strategy for Trust Propagation

Consider a trust path, P in a social network as <u_1 , u_2 , u_3 ,...,u_n> where u_1 ,u_2 , u_3,…u_n are the users along in the path. The simple multiplication strategy for trust prediction measures trust Ti along the path as

$$Ti= t(u_1,u_2) \times t(u_2,u_3)\times\ldots..\times t(u_{n-1},u_n) \qquad (1)$$
$$\exists 0 < t(u_i,u_j)<=1$$

Where $t(u_i,u_j)$ is the direct trust value between users u_i and u_j respectively.

The multiplicative strategy, despite being incredibly simple, has some interesting characteristics. First, if all the trust values along the trust chain have value 1, then the propagated trust between the source and destination node is calculated to be also 1. Secondly propagated trust value will decrease with the increase in the number of users along the trust path. Thirdly if the source node poorly trust the next node in the chain, the propagated trust value of the path will drop even if the direct trust values between the next nodes in the path is high. For finding path from source to sink, algorithm used in [19] is Dijkstra's shortest-path algorithm. When the path is found between Source and destination using Dijkstra's shortest-path algorithm, trust is then calculated using the trust propagation function said in equation 1.

3.2 Simple Average Strategy for Trust Propagation

For a trust path, P in a social network as <u1, u2, u3,...,un> where u1,u2, u3,…un are the users along in the path. The simple average strategy of trust propagation calculates trust Ti along the path as:

$$\text{Ti} = \frac{\text{t}\ (\text{u}_1,\text{u}_2)+\text{t}(\text{u}_2,\text{u}_3)\ldots\ldots+\text{t}(\text{u}_{n-1},\text{u}_n)}{n} \qquad (2)$$

where t(ui,uj) is the direct trust value between users ui and uj respectively.

This strategy also share same property as multiplicative strategy that if all the direct trust values along the trust path have trust value 1, then propagated trust between the source to destination node is also 1. On the other hand, if all the direct trust values along the trust path have the same trust value$x$, then the propagated trust between the source and destination node is also the same trust value x.

4. OUR CONTRIBUTION

In a trust rating based system in any online social network, if any user want to know the reliability of other user, our target is to find the most trusted path in as minimum time as possible than other and at the time of trust calculation, instead of giving equal weightage to all user in the path, decrement the weightage of the user on the basis of their distance from the sink to source.

Input

n← number of users in the trust network

G ← (n×n) trust matrix with trust rating $trust_{i,j}$

where $i,j$are directly connected neighbours such that

$$G_{i,j} = \begin{cases} trust_{i,j}\, if i != j \\ 0 \quad otherwise \end{cases}$$

$G^{'}$ ← (n×n) trust matrix with the value $G'_{i,j}$for every directly connected

neighbours (i,j) in the matrix G such that

$$G'_{i,j} = \begin{cases} \frac{1}{trust_{i,j}} if trust_{i,j} != 0 \\ 999 \quad otherwise \end{cases}$$

s ←source node

t ← target node

$direct_{trust}$ ← a list for storing path trust

4.1 Path finding algorithm

For finding trust path in our algorithm, we incorporated a methodology that at each level we choose the edge with the highest trust rating instead of choosing the lowest as in Dijkstra's algorithm. In order to achieve this, all the trust rates in the trust dataset are replaced by their reciprocals. Then we use the Dijkstra's algorithm for finding shortest path, where we actually get the most trusted path.

For a given graph $G_{i,j}$, where i and j are directly connected neighbours and $t_{i,j}$ is the direct trust value; another graph $G'_{i,j}$ is constructed as

$$G'_{i,j} = \begin{cases} \frac{1}{t_{i,j}} \quad if \ \ t_{i,j} != 0 \\ 999 \quad otherwise \end{cases}$$

The idea is to make the smaller trust value larger and larger value smaller. To understand the design, suppose for a given set of three ratings: 2, 4, and 5.

Ratings: 2, 4, 5 (Ascending order)

Reciprocals: $\frac{1}{2}, \frac{1}{4}, \frac{1}{5}$(Descending order ~ 0.5, 0.25, 0.2)

Thus we see how the highest rating 5 became the smallest trust rating 0.2 and the smallest value 2 becomes the largest element 0.5.Dijkstra's Shortest Path Algorithm can easily be applied to the resultant matrix. The path selected in this case will not be the minimum weight path; rather it will be the most weighted path or in other words, the most trusted path.

Algorithm:

```
$if(s != t)\{$
$direct_{trust}$ ← $G(s,t)$
remove the edge $(s,t)$ from $G'$
$PathP$ ←Dijkstra $(G^{'}, s, t)$
$propagated_{trust} = \frac{\sum_{i,j\in P}^{i=s}(d_j \times trust_{i,j})}{\sum d_j}$
Compare$direct_{trust}$ and $propagated_{trust}$
Restore the edge $(s,t)$ in $G'$
}
```

4.2 Trust propagation function

For the set of selected nodes in the shortest and most trusted path P, the propagated trust $propagated_{trust}$ from source node s to target node t is the average of the trust ratings from each nodes in P weighted by the propagative distance d_i from the source node to each node i in the path P:

$$\boldsymbol{propagated_{trust}} = \frac{\sum_{i,j\in P}^{i=s}(d_j \times trust_{i,j})}{\sum d_j}$$

5. EXPERIMENTAL DESIGN

5.1 Data Preprocessing

To apply the above said algorithm in this study, we design a simulation environment in Windows 7 and design a software using java, where we store the online social graphs information in the database and the graphs will reform in our software at the time of execution as an adjacency matrix representation. As some of the real datasets were available in adjacency list representation; first a conversion was done to change the representation of the graph. Then the algorithm was applied to this graph to approximate local trust of unknown nodes from the all source nodes.

Another issue faced while executing the code, that the Java Virtual Machine generate an out of memory errors when tried to apply large network dataset to this algorithm. Hence, the memory heap size was manually increased to cope with the situation.

5.2 Experiment

In this study, we have compared our algorithm with multiplicative strategy and simple average strategy using the same dataset. In comparing the algorithms we have followed the common approach namely "leave-one-out" that is for every direct trust link in the Trust Graph, we first remove the link from the trust graph, next we calculate the propagated trust value through our approach between the corresponding nodes and finally restore the connecting link.

The testing algorithm is as follows-

1. For every edge, e in Trust matrix, G between source user, S and destination user, D
2. Direct trust= weight of matrix G(S,D)
3. Remove edge, e from G and build modified graph G'.
4. Get the shortest and most trusted path between S and D using a modification of Dijkstra algorithm and store in an arraylist, PATH.
5. Calculate Propagated Trust between S and D from our

Trust propagation Algorithm

6. Propagated_Trust(S,D)= Trust_propagation_Algorithm(PATH, S, D)

7. Compare Direct trust and Propagated Trust between S and T.

8. Finally, the results of the execution of the trust network with estimated local trust were stored in Microsoft Excel files for further analysis.

5.3 Datasets

Our algorithm makes use of explicit trust ratings to infer propagated trust in social networks. For this, we have used one real life survey for real dataset and four online available real world dataset for our experiments.

5.3.1 Real world survey dataset

With the objective of gathering real-world trust dataset to test the accuracy of our algorithm, we have conducted a survey.

A questionnaire was first designed and tested in an undergraduate engineering class of Information Technology, in Jadavpur University where a group of 65 students took part in the survey.

Students were asked to rate their close friends on a scale of 1-5 (least trusted-most trusted).

The data collected from this survey was used for analysis of the algorithm. The result found suggests the fact that the person who is most popular may not be the most trusted person on class and vice versa. Indeed, a student i might nominate j as a trustworthy friend without being nominated as a trusted friend by j. Hence, it is immaterial to calculate trust on the basis of number of trust ratings because trust is personal and depends on an individual’s perspective.

5.3.2 Online available Real world dataset

Advogato: The primary data set that we used for our experiment is Advogato[39], an online community for open source software developers. The users of the site rate each other on a level of trust. The preferences of trust values are master, journeyer and apprentice, with master being the maximum level in that category. There exists self-loops in the dataset as it is possible to trust oneself on Advogato, but we exclude them out as they do not conform to our model. We substitute its three trust values as follows: master = 10, journeyer = 6.6, and apprentice = 3.3. The result of these ratings among members is a rich web of trust, and after removing the self-loops the resulted datasets comprises of 6539 users and 32608 trust ratings. The distribution of trust values in the Advogato web of trust is as follows: master: 13840; journeyer: 14883, and apprentice: 4245.

EIES: This dataset is Freeman’s EIES network [41] and was collected in 1978 and contains associations of researchers working on social network analysis. The dataset comprises of personal associations among 48 of the researchers during their time of study. All relations in the network have a weight between 0 and 4. A close personal friend of the researcher’s is presented by a rating 4; 3 symbolizes a friend; 2 represents a casual acquaintance; 1 represents a person the researcher has only heard of but never met; and 0 corresponds to a complete stranger.

Wolf: The dataset Wolf is a 20×20 matrix; referred to as wolf in this work. This dataset was collected by observing a troop of monkeys in Ocala, Florida for a period of 3 months, by Linda Wolfe [42]. Joint presence at the river was coded as an interaction.

Zachary: The last dataset we use, is called the Zachary karate club is a 34×34 matrix; this dataset, referred to as Zachary; was gathered from the participants of a karate club in a university by Wayne Zachary. The matrix indicates the comparative strength of the relationships i.e., number of situations in and outside the club in which they have interacted.

6. RESULT AND ANALYSIS

To provide sufficient and fair analysis we compared our algorithm with the iterative multiplicative strategy presented by Hasan and Brunie [17] and simple average strategy in our study. Our experiments demonstrate that our algorithm outperforms both these algorithms. In our algorithm we take up the most trusted and shortest path for trust calculation whereas in [19] the lowest trusted part albeit the shortest path is chosen for trust calculation. This is the major advantage of our algorithm over the iterative multiplicative strategy presented by Hasan and Brunei as we know that a chain of highly trusted individual is more reliable than a chain of low trusted individual. Moreover, our experiment supplies evidence that a significantly strong positive linear correlation of 0.71(rounded up to two decimal places) exists between direct and propagated trust obtained through our algorithm while that of the iterative multiplicative strategy is 0.48 for the Advogato dataset [19]. A scatter plot of the direct trust values and the corresponding propagated trust values is given in Figure 1.

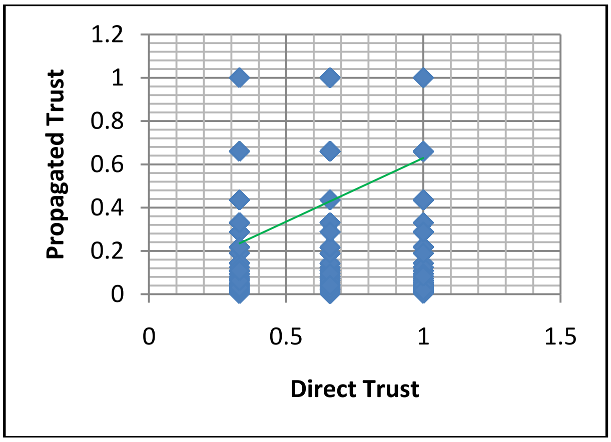

Figure 1: Correlation between Direct Trust and Propagated Trust for Advogato Dataset

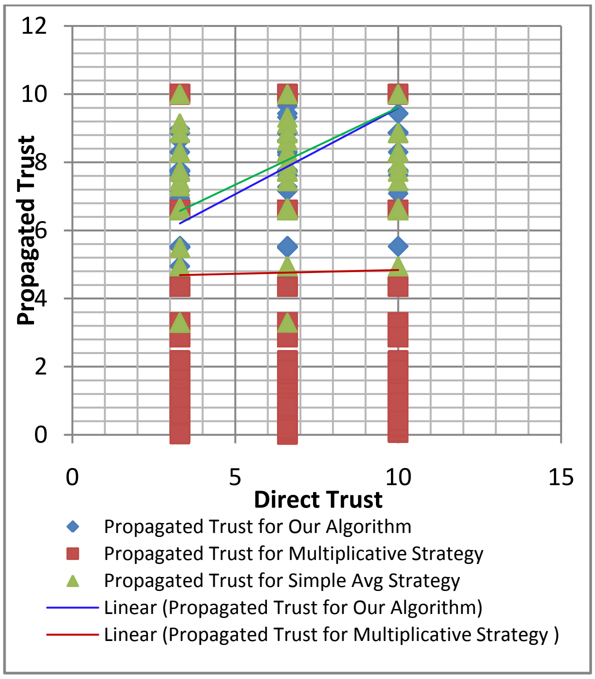

Figure 2: Comparison of our algorithm with Multiplicative Strategy and Simple Average

It is noted that the values of direct trust and propagated trust are acquired independently of each other in the experiment. This outcome is significant since the data set used is a real and large web of trust. A comparison of this algorithm with the iterative multiplicative strategy and the simple average strategy for the advogato dataset is given in Figure 2.

The number of cases when an alternate path was found between two vertices with a direct edge is 32968. A bar graph of the number of available paths for every hop lengths is depicted in Figure 3. As is shown in the figure, maximum number of paths are available for path length 2; thus making it one of the least time consuming algorithm. It is to be noted here, that the path is not only the shortest path but the most trusted path as well. The figure also shows that the vertex count is at most 3 for over 99% of the instances when a path is found from the source vertex to the target vertex. As we know, fewer edges on the path between two entities leads to more reliability of trust propagated over that path. The result thus implies that a very high percentage of the inferred trust values have high reliability.

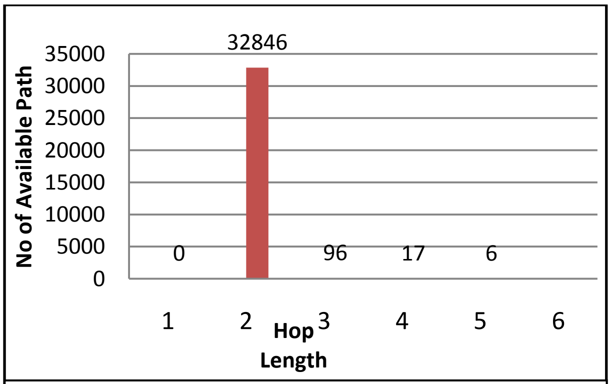

Figure 3: Availability of paths with their path length

A comparison of the average run time for all three strategies is shown in Table 1. The values acquired are an average of the time taken by the respective algorithms for five consecutive executions. Indeed, the execution time is directly proportional to the number of users in the trust graph. It is clear that our algorithm fairs better than the other algorithms in terms of the execution time in majority of cases followed closely by simple average strategy.

Table 1. Average running time calculated in milliseconds

Dataset	Number of users	Our Algorithm	Multiplicative Strategy	Simple Average
wolf	20	66	75	65
Zachary	34	37	34	31
JU IT	35	115	125	125
EIES	47	116	144	109
Advogato	6541	9101× 103	8875274	9808546

We have calculated Mean absolute percentage Error (MAPE) as our evaluation metric to assess the degree of deviation of the inferred trust values from the direct trust values for all the five datasets. The mean absolute percentage error (MAPE) is a measure of prediction accuracy in statistics. It usually expresses accuracy as a percentage, and is defined by the following formula:

$$MAPE = \frac{1}{n}\sum_{i,j=1|i!=j}^{n}\left|\frac{DT_{i,j} - PT_{i,j}}{DT_{i,j}}\right|$$

where, n is the total number of users in the trust graph, $DT_{i,j}$ is the direct trust and $PT_{i,j}$ indicates predicted trust between any two users i and j.

Table 2. Mean Absolute Percentage Error

Dataset	Number of users	Our Algorithm	Multiplicative Strategy	Simple Average
wolf	20	2.06577	0.734957	2.063154
Zachary	34	0.569338	0.717802	0.557824
JU IT	35	0.664504	0.69073	0.665724
EIES	47	0.743257	0.833571	0.768774
Advogato	6541	0.217575	0.387433	0.242086

The descriptions of the five datasets have been specified in section 5.1. The result has been given in Table 2. The results reveal that our algorithm outperforms multiplicative strategy in case of three of the five datasets; and also simple average in three of the five datasets and performs almost equally in the other two. As is clearly shown, for our algorithm MAPE is calculated to be much less than the other two approaches for the same datasets. It is worth mentioning here that the simple average has been applied to the most trusted and shortest path extracted with the help of our algorithm. In other words, the selection of the path is done using our algorithm; the two approaches differ only in the method of trust calculation. However, for the multiplicative strategy path selection and trust inference both has done by the method mentioned in [19].

It is examined the correlation between direct trust and propagated trust of this algorithm. In this analysis, the popular Pearson correlation coefficient which gives a measure of the linear correlation between two variables A and B in the range of [-1,+1], where 1 indicates full positive correlation, 0 indicates no correlation, and −1 indicates full negative correlation is used. In order to make a comparison between these three algorithms the correlations are calculated as well for the same datasets. The result has been given in Table 3. In case of this algorithm, a strong positive correlation between direct and propagated trust values is seen for large graph; for very small graph this algorithm does not give satisfactory result. Results of the other two algorithms are not very effective for large graph as this algorithm do.

Table 3. Correlation between direct and propagated trust

Dataset	Number of users	Our Algorithm	Multiplicative Strategy	Simple Average
wolf	20	0.53794	0.13353	0.57542
Zachary	34	0.38190	0.19533	0.42512
JU IT	35	0.21044	0.14057	0.21044
EIES	47	0.35213	0.04058	0.40531
Advogato	6541	0.71058	0.48937	0.69029

7. CONCLUSION

Recognizing trustworthy people for developing relationships is a primary concern in online social networks these days. To address these issues, ‘‘trust’’ is an invaluable concept in social network services.

In this paper we have presented a trust propagation algorithm. Discovering optimal and reliable trust path is always challenging in large online social networks. Trust propagation prediction accuracy is affected by the length of trust paths and different measuring approaches which decide how to unite multiple information sources. This algorithm takes up the most trusted as well as the shortest path for trust inference. Then the average of trust values is weighted by the propagation distance to calculate trust for the chosen path. This strategy is evaluated and compared with that of the multiplicative strategy proposed in [17] and simple average strategy. With experimental evaluation; we demonstrate that this algorithm outperforms the other two approaches. The dataset that was primarily used was the Advogato dataset with over 30,000 trust links. Apart from this, four other smaller datasets were used for analysis. The statistical techniques used to analyze the data. Finally, the results of the experimental research were presented and interpreted.

In future work we would like to further improvise the implementation to gain better throughput. For finding shortest path dijkstra algorithm is used which will find only one shortest path at a time but there can be more than one. In future we want to solve this problem with some stochastic optimization techniques. Another idea would be to include content-related features for determining the conditions for trustworthy and untrustworthy behaviour. Although it would reach more to the domain of artificial intelligence, it would be an interesting extension nevertheless.

8. REFERENCES

[1] http://Advogato.org

[2] Grabner-Krauter et al. “Trust in online social networks: A multifaced perspective.” Pages 48-68, Forum for social Economics (2015).

[3] Golbeck ,Hendler, “Inferring binary trust relationships in Web-based social networks”, ACM Transactions on Internet Technology (TOIT), v.6 n.4, p.497-529, November, 2006.

[4] Mui, Lik, Mojdeh Mohtashemi, and Ari Halberstadt. "A computational model of trust and reputation." System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on. IEEE, 2002.

[5] Benantar, Messaoud. Access control systems: security, identity management and trust models. Springer Science & Business Media, 2006..

[6] Aringhieri, Roberto, et al. "Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems." Journal of the American Society for Information Science and Technology 57.4 (2006): 528-537.

[7] Sanaz Kavianpouret. Al, “Calculating trust value in information propagation for online social network sites.” International Journal of Engineering and Technical Research(IJETR), Vol-2, Issue-3 Mac (2014)

[8] Page, Lawrence, et al. "The PageRank citation ranking: bringing order to the web." (1999).

[9] Abdul-Rahman et al, S. Supporting Trust in Virtual Communities. Proceedings of the Hawaii International Conference on System Sciences, USA. (2000).

[10] Kamvar, Schlosser, Garcia-Molina, “The Eigentrust algorithm for reputation management in P2P networks,” In International Conference on World Wide Web (WWW), 2003.

[11] Richardson, Matthew, Agrawal, Domingos. “Trust Management for the Semantic Web,” Proceedings of the Second International Semantic Web Conference. Sanibel Island, Florida. (2003)

[12] Golbeck, Computing and Applying Trust in Web-based Social Networks, Ph.D. Dissertation, University of Maryland, College Park, (2005)

[13] Avesani, Massa, et al. “A trust-aware recommender system for ski mountaineering”. International Journal for Infonomics. (2005)

[14] Levien, “Attack-resistant Trust Metrics”. Ph.D. thesis, University of California at Berkeley, USA, (2004).

[15] Sana Hamdi, Alda Lopes Gancarski, AmelBouzeghoub, and Sadok Ben Yahia. 2016. TISoN: Trust Inference in Trust-Oriented Social Networks. ACM Trans. Inf. Syst. 34, 3, Article 17 (April 2016), 32 pages. DOI=http://dx.doi.org/10.1145/2858791

[16] D. Gambetta. Can We Trust Trust?, chapter 13, pages 213{237. Basil Blackwell, 1988. Reprinted in electronic edition from Department of Sociology, University of Oxford.

[17] Omar Hasan O, Brunie L, Pierson J Evaluation of the Iterative Multiplication Strategy for Trust Propagation in Pervasive Environments, ICPS’09 17, July 13-17 2009, London, United Kingdom.

[18] Wang J. C, Chiu C. C. Recommending trusted online auction sellers using social network analysis. Expert Systems with Applications. 2008, 34(3) pp. 1666-1679

[19] Wenxue LIU, et al. "A Trust-based Information Propagation Model in Online Social Networks" The FARMS Review 8.8 (2013): pp-1767-1773

[20] J.-H. Cho, A. Swami, I.-R. Chen, Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks, Journal of Network and Computer Applications 35 (3) (2012), pp-1001–1012.

[21] Y.A. Kim, H.S. Song, Strategies for predicting local trust based on trust propagation in social networks, Knowledge-Based Systems 24 (8) (2011), pp-1360–1371.

[22] P. De Meo, A. Nocera, G. Terracina, D. Ursi, Recommendation of similar users, resources and social networks in a Social Internetworking Scenario, Information Sciences 181 (2011) , pp-1285–1305.

[23] Y.A. Kim, R. Phalak, A trust prediction framework in rating-based experience sharing social networks without a web of trust, Information Sciences 191(2012), pp-128–145.

[24] J. Caverleea, L. Liu, S. Web, The socialtrust framework for trusted social information management: architecture and algorithms, Information Sciences 180 (1) (2010) 95–112.

[25] T. Abdessalem and I. BenDhia. A reachability-based access control model for online social networks. In Proceedings of the First ACM SIGMOD Workshop on Databases and Social Networks, DBSocial' 11, pages 31–36, Athens, Greece, June 12-16, 2011.

[26] P. Bedi and R. Sharma, "Trust based recommender system using ant colony for trust computation," Expert SystAppl, vol. 39, no. 1, pp. 1183–1190, Jan. 2012.

[27] Pujol, J. M., Sanguesa, R., Delgado, J. (2002). Extracting reputation in multi-agent system by means of social network topology. The Proceedings of the first international joint conference on autonomous agents and multi-agent systems, Italy, pp. 467-474.

[28] Sanadhya, S., & Singh, S. (2015). Trust Calculation with Ant Colony Optimization in Online Social Networks. Procedia Computer Science, 54, 186-195.

[29] Ziegler, C.N. (2005). Towards Decentralized Recommender Systems. PhD Thesis, University of Freiburg, Germany.

[30] http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/

[31] LesaniMohsen, and NiloufarMontazeri. "Fuzzy trust aggregation and personalized trust inference in virtual social networks." Computational Intelligence 25.2 (2009): 51-83.

[32] A. Jøsang, A.R.Ismail, and C. Boyd. 2007. "A Survey of Trust and Reputation Systems for Online Service Provision". Decision Support Systems, 43(2):618–644.

[33] Partha Sarathi Chakraborty, Sunil Karform, "Designing Trust Propagation Algorithms based on Simple Multiplicative Strategy for Social Networks", Procedia Technology, Volume 6, 2012, Pages 534-539, ISSN 2212-0173.

[34] Sobecki, J. (2007). "Web-based system user interface hybrid recommendation using ant colony metaphor." LNCS 4694/2008. Berlin/Heidelberg: Springer, pp.1033–1040, ISBN: 978-3-540-74828-1