Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 68 – 77

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Controlled Expansion of Neighborhood in Trust Based Recommender Systems

Deepali Jain[a], Harmeet Kaur[a,]*

[a]HansRaj College, University of Delhi,Delhi -110007,India

Abstract

A technique to identify good recommenders and make them trustworthy neighbours is presented. By having good recommenders in direct association, an agent will have an improved set of recommendations in lesser amount of time and computations as compared to getting recommendations transitively from neighbour of a neighbour and so on. While estimating trust on an unknown agent, the upper limit on the number of hops one needs to explore to reach that agent is also proposed, hence further minimizing time and computation complexity. Results of experiments conducted on a real dataset illustrate the efficiency and effectiveness of the proposed method.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

*Keywords:*Recommender systems; Web of trust; Trust transitivity; Neighbourhood expansion; Trust decay

1. Introduction

With the rapidly growing amount of information available on the WWW, it becomes necessary to have tools to help users to select the relevant part of online information[8]. To satisfy this need, Recommender Systems have proven to be an effective solution to the information overload problem by providing users with information and services specific to their needs, rather than an undifferentiated mass of information[19].

* Corresponding author. Tel.: 9811503958.
E-mail address: hkaur@hrc.du.ac.in

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.01.057

Recommender system is emerging as a powerful and popular tool for online information relevant to a given user [9].Most prevalent recommender systems augment trust into the recommendation process. Trust-enhanced recommender systems mine the trust network referred to as WoT among its users. Trust is a vital ingredient of any successful interaction between individuals, among organizations and/or in society at large. The agent uses its level of trust on the recommenders to decide which recommendation to accept[18].

In most of the existing trust based recommender systems, the recommendations are generated transitively by propagating the query through the chain of connections towards user's neighbours of neighbours and so on. Consequently every time in such a situation, this mechanism involves lot of time in query propagation and response accumulation as well as computations in inferring transitive trust on distant agents to account for their recommendations. A solution to reduce time and computations required in getting suggestions from a far away agent is to include such distant agents in the neighbourhood. Thus this paper presents a technique of enlarging neighbourhood using which an agent can enlarge its neighbourhood by including some far away reachable good recommender agents in its direct association. By adopting the process of expanding neighbourhood an agent enriches and enhances its set of suggestions without travelling through series of acquaintances in WoT and saves its time and effort. An agent called *source* initiates the process of expansion of neighbourhood where it tries to find good recommender agents with whom it is not directly connected, to incorporate them in its neighbourhood. The process of expansion of neighbourhood can be invoked when either sufficient number of recommendations is not provided by user's immediate neighbours or when the user is not directly connected to enough number of other agents in the web of trust. The procedure of searching a suitable agent to be included in the neighbourhood should be bounded by the distance from the source. Since after a number of hops in WoT, trust decreases so much that it is futile to search for good recommenders. Hence the proposed method of expanding neighbourhood is limited by the number of hops from source agent to the target one which further helps in reducing time and computation complexity.

While estimating trustworthiness of a distant agent, employment of the process of determining transitive trust between two unknown users is inevitable. Trust transitivity is defined as the possibility to use trust information from other entities in order to infer a trust evaluation to a given entity[1]. Although an agent might be a stranger to the evaluating agent, the evaluating agent can attempt to estimate the stranger's reputation based on information garnered from others in the environment[20]. Establishing trust on a far away agent will be handled by exploiting the trust ratings already present in web of trust to find trust scores for unknown agents.

Adopting the simplest policy of trust propagation, all those people who are trusted by persons we trust are considered likewise trustworthy[2]. Trust would propagate through the network whenever two individuals can reach each other via at least one trust path[2]. The general question is: how to propagate trust? On one path or combining multiple paths? Considering just one path may cause losing a lot of information. So, try to consider multiple paths leading to the target agent. But, then the question is how to combine them?[8]. Since trust is fuzzy, a solution to this problem lies in using composition of fuzzy relations in order to calculate transitive trust.

Paper's main contributions are as follows:

- Expanding neighbourhood: Process adopted by an agent to expand its neighbourhood by including some more good recommenders in its direct approach to get more and better suggestions in reduced amount of time and computations
- Limiting number of hops required from source to sink: The proposed process of computing degree of trust for an unknown agent limits the number of hops from source to sink beyond which possibility of finding trustworthy agent is negligible and consequently further saves time and computations.

Organization of this paper is as follows: related work is discussed in section 2. The proposed model of neighbourhood expansion is explained in section 3, 4 and 5. Experimentation and results hence obtained are reported in section 6. Finally, Section 7 concludes the paper and presents some directions for future work.

2. Related Study

In the literature, various models have been suggested to infer the transitive trust.

The work presented in[4, 11] researchers have adopted the confidence based mechanism in which the confidence between participants is considered in trust transitivity.

The model of trust proposed by Golbeck explains an algorithm named TidalTrust to infer trust to be placed by the source on the sink[7]. The source infers trust rating for the sink by using a weighted average over all neighbours. This approach has the problem of time complexity as request from source need to travel lot of the WoT to locate the node having the rating. The reason is that by using this algorithm one need to look among all the users till a certain depth.

The Appleseed model presented by Ziegler is based on spreading activation energy[2]. In the AppleSeed algorithm, the decay factor d is also considered. Major weakness in this approach is that, this model assume the trust to be additive. While computing the trust from source to sink there are many weakly trusted paths to sink, this according to Appleseed algorithm sums up to high trust value which may not be the case, thus the assumption of additive trust is not accurate.

In the work carried out by Massa and Avesani[15, 16, 17], two matrices are provided, one is rating matrix [N ×M] giving ratings to M items by N agents, and trust matrix [N ×N] depicting value of trust among agents. He uses a trust propagation algorithm MoleTrust to find the transitive trust values, and the trust matrix will be updated with inferred transitive trust values. The main weak point in this approach is the time complexity of the algorithm. N and M are large values for large networks and calculating the trust and similarity values are time consuming.

The model presented by Walter et al.[21] consists of agents, objects, and agent's profiles. In this model whenever a source agent wants to rate a particular item it asked its neighbours and its neighbours in turn pass on a query to their neighbours if they cannot provide a rating themselves. In order to generate the transitive trust from source agent to sink agent they have used the multiplicative approach and multiply the trust values along the path between the source and sink agent. Problem associated with their approach is that as the length of the path between the source and sink increases no discount in trust value is taken and no solution in case of multiple paths between source and target is given.

In the work presented by Alcalde and Mauw[1], they have developed an abstract algebra expressing the basic properties of trust dilution and trust fusion where dilution was used to calculate the trust along trust chains and fusion is used to compute the overall trust if there are different sources of information. In this case no trust decay has been incorporated in finding trust transitivity.

In the work done by Liu et al. [12, 13, 14] the concept of Quality of Trust Transitivity (QoTT) has been proposed. They have included the impact of social relationship, recommendation roles and preference similarity into account to compute transitive trust but they haven't discussed the limit on the number of hops from source to sink while estimating indirect trust from source to sink. Major weakness of this method is the need to aggregate values for every QoTT characteristic in every social network trust path from the source to the sink.

In order to overcome the above mentioned weaknesses, this paper proposes a method of expansion of neighbourhood where

- it provides an upper limit on the number of hops to be taken from source to sink thereby increasing the efficiency of the proposed model.
- as the length of the path between source and sink increases, the decay of trust is taken into account.
- source driven parameters like trust_threshold_neighbour (minimum value of trust in an immediate neighbour so that the request can be propagated) and trust_threshold_sink (lowest accepted degree of trust for sink) are also included.

3. Utilizing Trust Transitivity in Expansion Process

The process of expansion of neighbourhood is called by the source in the case where either he is not directly connected to adequate number of other agents in the WoT or satisfactory recommendations are not provided by source's immediate neighbours. Thus in such a scenario agent will have to propagate its request towards its neighbours of neighbours and so on until its query is satisfied. This results in involvement of time and computations in calculation of trust on distant agents and fetching results from those agents. In order to avoid the additional load of computing trust transitively each time the query propagates, an agent would like to increase members in its neighbourhood by having good recommenders as neighbours. This way one would have to calculate trustworthiness of newly added agents only once and in future their recommendations could be taken without wasting much time and effort. Our process of expanding neighbourhood works on similar lines, where calculation of degree of trust for a distant agent happens once and later on source can fetch its advice directly. Moreover the process of searching a new neighbour is bounded by the path length from source to the sink which saves time and computations. Thus the proposed procedure of expanding neighbourhood saves time and computation complexity twice:

1. Since the method provides upper limit on the number of hops need to be taken by the request from the source, hence source will save time and computations while searching suitable new neighbour,
2. Once a new agent has been added into the neighbourhood of the source, source can obtain its advice directly and save time and effort required in getting recommendations transitively.

4. Finding Trust Score for the Sink

When source agent say a_i wishes to ascertain degree of trust on an unknown agent called sink say a_{sink}, it carries out the process of finding trust rating for the sink transitively which involves two processes: 1) Request Propagation and 2) Response Accumulation. Sections 4.1 and 4.2 explain these processes in detail.

4.1. Request Propagation

As a part of finding the trustworthiness of the sink, the source prepares a request as the following 7-tuple <request_id, sink, level (d), dampening_factor (df), traversed_nodes, trust_threshold_neighbour >
where
request _id is the unique identification number of the request,
sink provides the name of the sink,
Level (d) signifies the distance of a node in WoT from source with level 0 assigned to the source, its immediate neighbour is at level 1, and neighbour of neighbour is at level 2 and so on and so forth. It is the path length from source to the current node.

In trust transitivity, trust decays with the increase of transitivity hops along a social trust path. Trust decay is commonly agreed upon, for people tend to trust individuals trusted by immediate friends more than individuals trusted only by friends of friends[3, 10]. The path with trust information linking the source node and the target one is called a social trust path[5]. In addition, the general decay is non-linear[10]. Thus a parameter called dampening_factor is included to take account of the trust decay as query moves away from the source. dampening_factor denoted by df ($df < 1$), furnishes the information about how much the trust values should be lowered at each level as the request moves away from the source.

traversed_nodes is the list of all nodes visited by the request in its propagation from the source. Every agent adds its id in the list of traversed_nodes before propagating the request further. The path traversed by the recommendation about the sink from the recommender to the source is same as of the request from source to recommender in reverse direction and this information about the traversed path is stored in the list of traversed_nodes in the request itself.

trust_threshold_neighbour (ttn) defines the minimum value of trust in an immediate neighbour so that the request can be propagated to that neighbour.

Source agent a_i prepares the request and finds the trust t_{ij} on all its neighbouring agents a_j. For all the neighbouring agents a_j such that $t_{ij} > ttn$, send a request to find degree of trust for sink. When a neighbouring agent a_j of source agent a_i receives a request in the form of a 7-tuple from the source, where sink is a_{sink}, it undertakes steps as outlined in Algorithm 1.

Algorithm 1: Request Propagation

```
1.  If the agent $a_j$ is directly connected to the sink then it will compute
        $(\text{Trust\_rating}_{jsink})_p = t_{jsink} * (1/(d)^{df})$        (1)
        $t_{jsink}$ is the degree of trust assigned to agent $a_{sink}$ by $a_j$ in WOT
        df is dampening_factor,
        d denotes the level,
        p the path is the list of traversed nodes from $a_j$ to $a_{sink}$
    1.1    Send response as < $a_j$, level, p, $(\text{Trust\_rating}_{jsink})_p$ > to the sender of the request.
```

```
2.   Else for all the neighbours $a_x$ of agent $a_j$ such that $t_{jx} > ttn$
     2.1   $level = level + 1$                                                          (2)
     2.2   Include $a_j$ in the list of traversed nodes for the path p.
     2.3   $a_j$ will transfer the request further to $a_x$
3.   For all those neighbours $a_x$ of agent $a_j$ which receives the request from $a_j$, repeat steps 1 to 2.
```

To calculate $(Trust\_rating_{jsink})_p$, which is the rating given to the agent a_{sink} by the agent a_j along the path p, trust value from agent a_j to the a_{sink} is reduced by multiplying it with $(1/(d)^{df})$.The value gets reduced by the virtue of the fact that *dampening_factor* <1, and d= the distance of the current node from the source, thus as d increases 1/d further reduces the trust value. The overall effect of the reduction in trust value can be controlled by dampening_factor. This diminishing of trust value is done to take the length of the path into account. This reduction of trust values on the basis of length of the path also restricts the propagation of request up to some manageable length as the distance d increases every time the request is propagated which in turn reduces the trust and finally after some transfers, it will become less than the acceptable value. In this manner the dampening_factor provided by the source determines the number of hops the request can take from source to sink.

4.2. Response Accumulation

A response is a tuple of the form < sender, level, path, $(Trust\_ rating\_ sink)_{path}$> where
sender is the one who is sending the response towards the source,
level(d)same as in algorithm1
path is the list of traversed _nodes navigated from source to recommender in order to reach sink and
$(Trust\_ rating\_ sink)_{path}$ is the trust value for the sink along the *path*.

Algorithm 2 outlines the steps taken when agent a_j receives a response from its neighbour a_x, which is of the form <a_x, level, p, $(Trust\_rating\_sink)_p$>. Now the agent a_j computes the minimum of $(Trust\_rating_{jx})_p$ and $(Trust\_rating\_sink)_p$, which then becomes $(Trust\_rating\_sink)_p$. This is done by considering the composability of trust which describes that a user should combine the different trust values received from different paths. If agent a_j is the source agent then it accepts the $(Trust\_rating\_sink)_p$ as the final trust value that the source can place on the sink via path p. Else agent a_j transports the response up in the ladder i.e. to the agent from which it received the request and decrement level by one. This can be easily done with the help of path p. This transportation process will go on until source is reached. There can be n number of such paths and their associated n trust values for sink. The source agent finds out the maximum of all these n trust values and calls it as Trust_ sink_final**.**

If this value comes out be greater than tts (lowest accepted degree of trust for sink) as set by source, then source can include sink in its neighbourhood and in future can take recommendations from sink.

```
Algorithm 2: Response Accumulation
1.   Agent $a_j$ calculates
       1.1 $(Trust\_rating_{jx})_p = t_{jx} * (1/(d)^{df})$                                        (3)
       1.2 $(Trust\_rating\_sink)_p = min((Trust\_rating\_sink)_p, (Trust\_rating_{jx})_p)$       (4)
2.   If $a_j$ = source then it computes
       2.1 $Trust\_sink\_final = max\ of\ all\ (Trust\_rating\_sink)_p$                           (5)
           for all paths from which it received the rating of sink
3.   Else
       3.1  $level = level - 1$                                                                   (6)
       3.2  Send response to $a_i$ which had sent the request to $a_j$ as < $a_j$, level, p, $(Trust\_rating\_sink)_p$>
```

Algorithm 1 and Algorithm 2 are used to ascertain degree of trust between two unknown entities.

5. Expansion of Neighbourhood

The process of expansion of the neighbourhood is initiated according to Algorithm 3 which is to expand neighbourhood.

Algorithm 3: Expand neighbourhood

1. Agent a_i asks its *most trusted direct neighbour* about the probable candidate to be included in the direct vicinity of the agent a_i. This neighbour in turn will find out its *most trusted direct neighbour* (known as sink) and present it as the likely contender for inclusion in agent a_i's neighbourhood
2. Source launches the procedure of finding the trustworthiness of the sink using Algorithm 1 and Algorithm 2 by seeking recommendations for sink from all its neighbouring agents except from the most trusted agent in the application domain.

This *most trusted direct neighbour* of the agent a_i is the one who is enjoying the highest trust rating by the agent a_i in its neighbourhood. The exclusion of *most trusted direct neighbour* of the source from the procedure of ascertaining trust value from source to sink comes from the fact that it has already claimed the sink as the best candidate to become a new neighbour of the source. Thus source wants to find the trustworthiness of the sink via other contacts in order to avoid any biasness. If the degree of trust for sink comes out to be greater that minimum acceptable value then source will includes sink into its neighbours list and in future seeks its recommendations when finding some relevant product, thereby expanding its neighbourhood.

6. Experimental Setup

Experiments were carried out to study the effect of expansion of neighbourhood on a distant agent given in section 4 and 5. The dataset for experiments was derived from web community of Apartmentratings.com[6]. The data set rates thousands of apartments in USA on the seven criteria viz. Parking, Maintenance, Construction, Noise, Grounds, Safety and Office Staff. The above set of parameters describes basic features of an apartment, according to which recommender will describe the apartment and probable user will choose the apartment to live in. For experiments the data has been collected directly from the Apartmentratings Web site. The dataset consists of approximately 500,000 raters who rated a total of almost 1000 different apartments at least once. The total numbers of reviews are around 1,000,000. Out of 500,000 raters, 10 different sets of 25 raters were chosen as a sample to study algorithms. Thus in total 250 raters were chosen. For each set of 25 raters their corresponding 25 agents were created using JADE and profile of each user was placed in its agent. The system is implemented using Java and JADE platforms. The algorithms of finding transitive trust and expansion of neighbourhood are developed and implemented as Java classes and are integrated with the JADE platform. The interaction among different agents for developing trust relationships were implemented as agent behaviours. In the initial phase of the experiment for each of its 25 users their list of acquaintances along with the degree of trust that they can place on each other were generated randomly. According to these lists initial web of trust was spawned which is similar to web of trust in Fig.1. but with 25 agents. Web of trust thus contains a directed edge from an agent to all the agents in its list of acquaintances weighted by the degree of trust as reported in the randomly generated list and hence become its neighbours. This was done for each set of 25 agents. The algorithm of finding trust score for an unknown agent (algorithm 1 and 2) was run on the initial web of trust. In order to start the algorithm out of 25 agents one agent was chosen as source agent. The goal of the experiment was to find out how much trust can the source place on sink via some intermediate links and to find out some relationship between numbers of hops required to reach the sink. Once this trust on a distant node has been established by the source, it can include the sink into its direct neighbourhood so that subsequently source can seek this newly added agent's advice also.

6.1. Discussion

To study the algorithm of evaluating trustworthiness of an unknown agent, simulations were carried out with 25 request propagations for each set of 25 agents by making each agent as source. In each simulation source agent initiated the process of ascertaining trust value for sink agent. As a second step of the process source asked its most trusted neighbour to provide the information about sink, after which source prepared the request to gather the information regarding trustworthiness of the sink. This request prepared by the source was then broadcasted among its neighbours who can further propagate the request according to the ttn as mentioned by the source in the request itself. This propagation of request and response accumulation is according to algorithms 1 and 2 given in section 4. Once the source received the trust_values for the sink from various paths it uses composition of fuzzy relations which computes the maximum of various minimums of all the trust values to discover the final trust valve that the source can place on the sink.

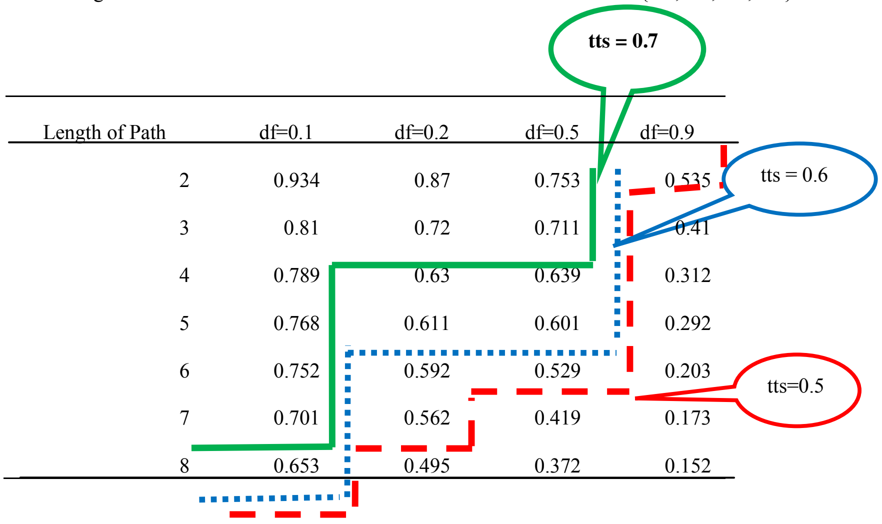
For the purpose of evaluation of algorithms and to determine the upper limit on the length of the path from source to sink, simulations were carried out for different values of parameters used in algorithms 1 and 2 given in section 4 which includes: dampening_factor (df), trust_threshold_neighbour (ttn), trust_threshold_sink (tts).

Values assigned to df were {0.1, 0.2, 0.5, 0.9}, ttn were {0.7, 0.8, 0.9} and tts were {0.5, 0.6, 0.7}.

On the basis of the data collected, the appropriate level up to which the request can be propagated was determined as there is no use of unguarded propagation which results in trust value for the sink lesser than the tts. For each set of 25 agents experiments were carried out and their results were documented independently as well as their average result was also recorded.

Table 1, illustrates average of 25 runs of first set of 25 recommenders where ttn = 0.9 for different values of df. It records Trust_ sink_final obtained at path length listed in first column against various values of df provided in first row. For example, Trust_ sink_final coming out to be 0.64 at three hops away from source for df = 0.2.

Table 1. Average of 25 runs of a set of 25 recommenders with ttn = 0.9 and df = {0.1, 0.2, 0.5, 0.9}


tts = 0.7
tts = 0.6
tts=0.5

Length of Path	df=0.1	df=0.2	df=0.5	df=0.9
2	0.934	0.87	0.753	0.535
3	0.81	0.72	0.711	0.41
4	0.789	0.63	0.639	0.312
5	0.768	0.611	0.601	0.292
6	0.752	0.592	0.529	0.203
7	0.701	0.562	0.419	0.173
8	0.653	0.495	0.372	0.152

In Table 1, no values were recorded for path length = 1 as it stands for already connected agents. All the values of degree of trust that lies above the green line will be accepted by the source if tts = 0.7, values above blue dotted line will be accepted if tts is 0.6 and if tts is 0.5 then source will include sink as its neighbour in cases where degree of trust lies above dashed red line.

Fig. 1. depicts various trust values of sink given in table 1 as calculated by the source against the length of the path from the source to sink for df = {0.1, 0.2, 0.5, 0.9}, tts = (0.5, 0.6, 0.7) and ttn = 0.9.

It clearly demonstrates that as the length of the path from source to the sink increases, the final degree of trust for the sink as calculated by the source decreases. This is actually close to real life situation where one tends to take advice from closer associates as compared to a person that can be reached by a long chain of intermediate contacts.

Similarly other runs were also carried out for other sets of 25 agents and their readings were recorded. For ttn = 0.9, the upper limit on the amount of distance traversed from source to sink can be calculated from Fig. 1. and other simulations for ttn = 0.9, Fig. 2. summarizes for different values of df and sink thresholds where ttn = 0.9.

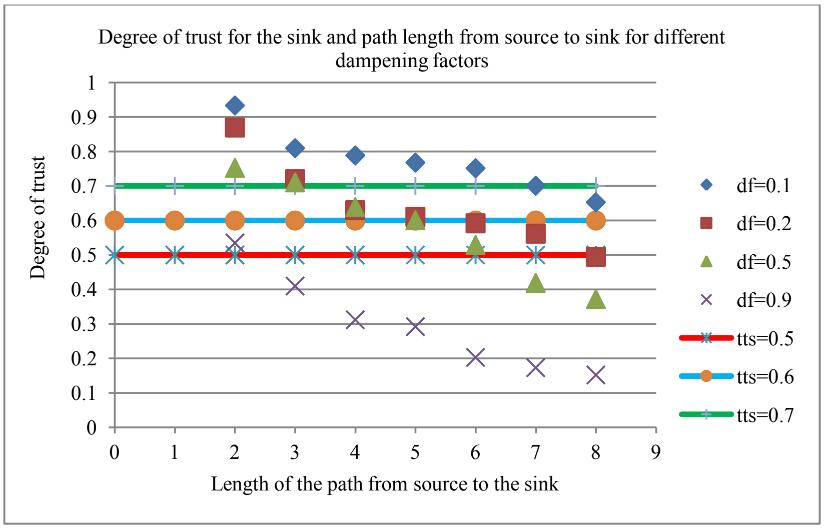


Fig. 1. Trust scores for sink computed for different df and tts, for ttn = 0.9

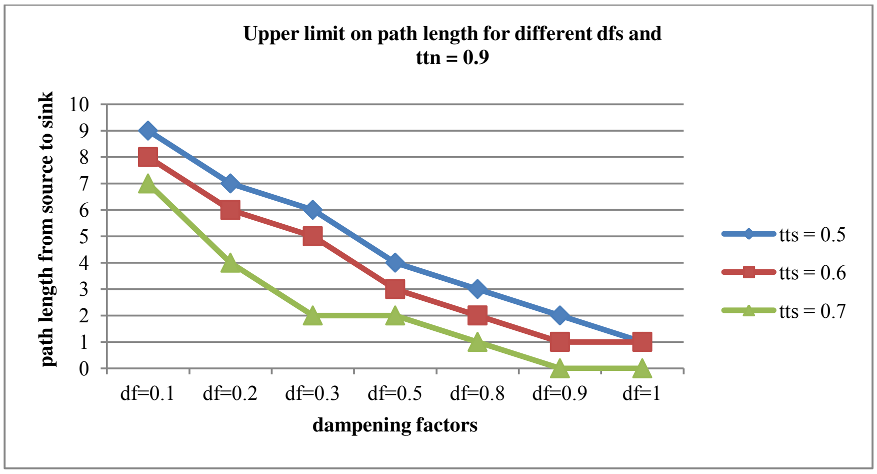


Fig. 2. Upper limit on the length of the path from source to sink based on df and ttn = 0.9

Similarly For ttn = 0.7, the upper limit on length of path from source to sink can be calculated from simulations for ttn = 0.7, Fig. 3. summarizes results for different values of df and sink thresholds where ttn = 0.7.

While comparing results for ttn = 0.7 and ttn = 0.9, it was discovered that path lengths are more relaxed for ttn = 0.9. That is in the process of searching sink, one can go farther for ttn = 0.9 as compared to the case where ttn = 0.7. This is due the fact that in case of ttn = 0.9 only highly trusted paths has been utilized in order to search the sink, that is higher the value of trust_ threshold_neighbour (ttn) more trusted neighbours will employed to search sink and then high trust value will placed on the sink through these routes. Thus source can explore further if highly trustworthy neighbours exist in each hop.

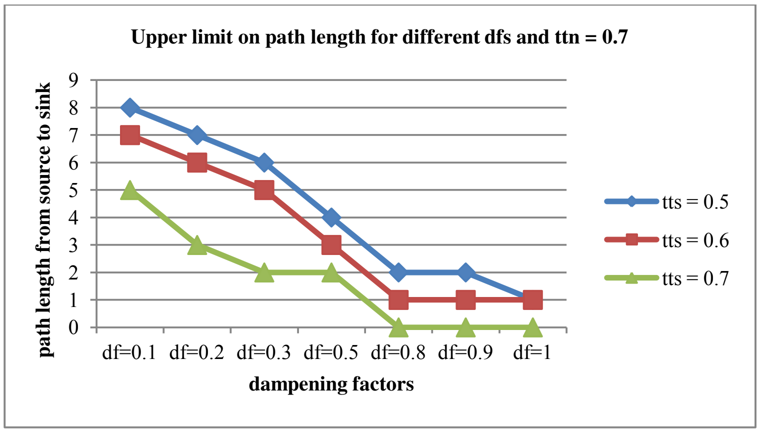


Fig. 3. Upper limit on the length of the path from source to sink based on df and ttn = 0.7

From the Fig. 2. and Fig. 3. it is clearly evident that maximum path length from source to sink is inversely proportinal to the trust_threshold_sink. As the value of tts increases from 0.5 to 0.7, upper limit on path length decreases, which reiterates the real life's state of affairs where more critical the situation is and more stricter the person is in getting recommendations from unknown parties, less will be the distance the query can travel. For df > 0.5 there is a rapid fall in the maximum path length that can be achieved.

7. Conclusion

In this paper a method of expansion of neighbourhood to assist an agent to enhance its set of recommendations by discovering good recommenders and make them its trustworthy neighbours is proposed. A for evaluating transitive trust between two unknown agents of trust based recommendation system has been discussed. Proposed technique involves trust propagation and response accumulation procedures to estimate the trust rating for the sink via a chain of acquaintances. Our process takes trust decay along the path into account while computing transitive trust. Experimental results have demonstrated that the proposed model follows the real life pattern of trust and provides an upper limit on the length of the path that needs to be explored to make a trustworthy neighbour out of an unknown agent.

More experiments are being conducted with some other real social network datasets to further validate the results. A feasible remedy for the situation where the source needs to enlarge its neighbourhood and the calculated trust for the sink comes out to be lower than the threshold is also under consideration.

References

1. Alcalde B, Mauw S. An algebra for trust dilution and trust fusion. In: *Formal Aspects in Security and Trust* 2009. p. 4-20.
2. Cai-Nicolas Ziegler. Towards Decentralized Recommender Systems. PhD Thesis, University of Freiburg 2005.
3. Gimpel J, Karnes K, Mctague J, Pearson- Merkowitz S. Distance-decay in the political geography of friends-and-neighbors voting. *Political Geography* 2008: 27(2). p. 231–252.
4. Guha R, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. In: *International Conference on World Wide Web* 2004. p. 403–412.
5. Hang C, Wang Y, Singh M. Operators for propagating trust and their evaluation in social networks. In: *The Eight International Conference on Autonomous Agents and Multiagent Systems* 2009. p. 1025–1032.
6. http://www. apartmentratings.com.
7. Golbeck J. Computing and Applying Trust in Web-Based Social Networks. PhD Thesis, University of Maryland 2005.
8. Jamali M. A Distributed Method for Trust-Aware Recommendation in Social Networks. *arXiv* preprint arXiv:1011.2245 2010.
9. Jinfeng Y, Li L. Recommendation Based on Trust Diffusion Model. *The Scientific World Journal* 2014.
10. Jøsang A, Gary E, Kinateder M. Analysing topologies of transitive trust. In: *International Workshop on Formal Aspects in Security and Trust* 2003.

11. Kuter U, Golbeck J. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence model. In: *Twenty-Second AAAI Conference on Artificial Intelligence* 2007. p. 1377–1382.
12. Liu G, Wang Y, Orgun MA. Optimal social trust path selection in complex social networks. In: *Twenty-Fourth AAAI Conference on Artificial Intelligence* 2010. p. 1397–1398.
13. Liu G, Wang Y, Orgun MA, Lim EP. A heuristic algorithm for service provider selection in complex social networks. In: *IEEE International Conference on Services Computing* 2010. p. 130–137.
14. Liu G, Wang Y, Orgun MA, Lim EP. Trust Transitivity in Complex Social Networks. In: *Twenty-Fifth AAAI Conference on Artificial Intelligence* 2011. p. 1222-1229.
15. Massa P, Avesani P. *Trust-aware Bootstrapping of Recommender Systems.* In: *Biennial European Conference on Artificial Intelligence Workshop on Recommender Systems* 2006. p. 29-33.
16. Massa P, Avesani P. Trust-aware recommender systems. In: *ACM conference on Recommender systems* 2007a. p. 17-24.
17. Massa P, Avesani P. Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems* 2007b: 3(1).
18. Sarda K, Gupta P, Mukherjee D, Padhy S, Saran H. A Distributed Trust-based Recommendation System on Social Networks. In: *Second IEEE workshop on Hot Topics in Web Systems and Technologies* 2008. p. 1-6.
19. Shambour Q, Lu J. Government-to-Business Personalized e-Services Using Semantic-Enhanced Recommender System. In: *Second International Conference on Electronic Government and the Information Systems Perspective* 2011. p. 197-211.
20. Shekarpour S, Katebi M, Katebi SD. A Trust Model For Semantic Web. *International Journal of Simulation Systems, Science and Technology* 2009: 10(2). p. 13-24.
21. Walter F, Battiston S, Schweitzer F. A model of a trust-based recommendation system on a social network. *Journal of Autonomous Agents and Multi-Agent Systems* 2008:16(1). p. 57-74.