Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 54(4), 427–437(2013)

Weiwei Yuan, Donghai Guan, Lei Shu, Jianwei Niu

Recommender Searching Mechanism for Trust-Aware Recommender Systems in Internet of Things

DOI 10.7305/automatika.54-4.416
UDK 004.896:519.254
IFAC 2.8.4; 1.1

Original scientific paper

Intelligent things are widely connected in Internet of Things (IoT) to enable ubiquitous service access. This may cause heavy service redundant. The trust-aware recommender system (TARS) is therefore proposed for IoT to help users finding reliable services. One fundamental requirement of TARS is to efficiently find as many recommenders as possible for the active users. To achieve this, existing approaches of TARS choose to search the entire trust network, which have very high computational cost. Though the trust network is the scale-free network, we show via experiments that TARS cannot find satisfactory number of recommenders by directly applying the classical searching mechanism. In this paper, we propose an efficient searching mechanism, named S_Searching: based on the scale-freeness of trust networks, choosing the global highest-degree nodes to construct a Skeleton, and searching the recommenders via this Skeleton. Benefiting from the superior outdegrees of the nodes in the Skeleton, S_Searching can find the recommenders very efficiently. Experimental results show that S_Searching can find almost the same number of recommenders as that of conducting full search, which is much more than that of applying the classical searching mechanism in the scale-free network, while the computational complexity and cost is much less.

Key words: Searching Mechanism, Trust Network, Recommender System, Scale-freeness

Mehanizam pretraživanja preporučitelja za sustave sigurnih preporučitelja u Internetu stvari. Inteligentni objekti su naširoko povezani u Internet stvari kako bi se omogućio sveprisutni pristup uslugama. To može imati za posljedicu veliku redundanciju usluga. Stoga je za pronalaženje pouzdane usluge u radu predložen vjerodostojan sustav preporučitelja (VSP). Temeljni zahtjev VSP-a je učinkovito pretraživanje maksimalnog mogućeg broja preporučtelja za aktivnog korisnika. Kako bi se to postiglo, postojeći pristupi VSP-a u potpunosti pretražuju sigurnu mrežu što ima za posljedicu velike računske zahtjeve. Iako je sigurna mreža mreža bez skale, eksperimentima je pokazano kako VSP ne može naći zadovoljavajući broj preporučitelja direktnom primjenom klasičnog algoritma pretraživanja. U ovom radu je predložen učinkovit algoritam pretraživanja, nazvan S_Searching: temeljen na sigurnim mrežama bez skale koji koristi čvorove globalno najvećeg stupnja za izgradnju Skeleton-a i pretražuje preporučitelja pomoću *Skeleton*-a. Iskorištavanjem nadređenih izlaznih stupnjeva čvorova *Skeleton*-a S_Searching može s visokom učinkovitošću pronaći preporučitelje. Eksperimentalni rezultati pokazuju kako S_Searching može naći gotovo jednak broj preporučitelja koji bi se pronašli potpunom pretragom, što je mnogo više od onoga što se postiže primjenom klasičnog algoritma pretrage na mreži bez skale, uz znatno smanjenje računske kompleksnosti i zahtjeva.

Ključne riječi: algoritam pretraživanja, sigurna mreža, sustav preporučitelja, mreže bez skale

1 INTRODUCTION

Internet of Things (IoT) is a dynamic global network infrastructure with self configuring capabilities based on standard and interoperable communication protocols where virtual “things” are seamlessly integrated [1-3]. The things or objects have identities, physical attributes, and virtual personalities. They could be Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc [2-5]. These things are able to interact with each other and cooperate with their neighbors to reach common goals [4-7]. IoT greatly facilities its users by enabling them access services provide by various things anywhere anytime- ubiquitously.

Things are widely connected in IoT to provide various services. It is important to help users find reliable services. This not only helps to attract more users to IoT, but also helps to improve the overall performances of IoT.

The recommender system is a subclass of information filtering system which can recommend services that a user would like. However, as the most popular recommendation mechanism, collaborative filtering (CF) suffers from the well-known data sparseness problem and the cold start problem [8-10, 12-13]. The trust-aware recommender system (TARS) improves CF by suggesting the worthwhile information to the users on the basis of user trust: trust is the measure of willingness to believe in a user based on its competence and behavior within a specific context at a given time [8]. Its performances are mainly measured in two aspects: 1) the prediction accuracy and 2) the prediction coverage [8]. TARS can achieve better prediction coverage than CF with similar prediction accuracy, especially when the data is sparse.

The recommender searching mechanism is a fundamental research issue of TARS: TARS is requested to efficiently find sufficient number of recommenders for the active users. Recommendations, especially those who are different from the majority, given by various recommenders are the most important information for TARS to predict ratings on the target items, The system may lose valuable information by involving only partial recommenders, so the recommender searching mechanism should involve as many recommenders as possible for TARS. At the same time, the efficiency of the recommender searching mechanism directly affects the response time of TARS. Since users always tend to choose the system providing rapid personalized services, it is essential to reduce the computational complexity of the recommender searching mechanism to attract more users for TARS applications. In this paper, we mainly focus on the recommender searching mechanism and discuss the prediction coverage, which can be measured by both the rating coverage and the recommender coverage[1].

To the best of our knowledge, no existing work has systematically focused on the recommender searching mechanism of TARS. Most researches [12-16, 21-25] did not provide any information how they find the recommenders. A few other works [8-10] briefly mentioned that they search the entire trust network to find the recommenders: since it does not miss any node reachable by the trust propagations, the TARS can achieve high prediction coverage. However, it is computational very expensive, especially when the TARS has large scaled trust networks.

We propose a recommender searching mechanism for TARS, named S_Searching, based on the scale-freeness of the trust network, which is able to efficiently achieve high prediction coverage for TARS. The trust network has been verified to be the *scale-free* network [8-10], whose degree distribution follows a power law, i.e.,$P(\kappa) \sim \kappa^{-\gamma}$, where $P(\kappa)$ is the probability that a randomly selected node has κ connections, and γ is the power of the degree distribution [18-20]. The comparison between the structure of the random network and the structure of the scale-free network is given in Fig. 1. The most notable characteristic in the scale-free network is the existence of nodes with degrees greatly exceeds the average. These highest-degree nodes are often called "hubs". Though the number of hubs is limited, they dominate the connectivity of the scale-free network. Our proposed S_Searching chooses the hubs to construct a skeleton for the trust propagation. S_Searching finds the recommenders for the active users via the skeleton: it first propagates the active users' trusts to the skeleton, and then finds the recommenders via the trust propagations from the skeleton. Benefiting from the superior degrees of the hubs in the skeleton, S_Searching can find the recommenders efficiently. Experimental results show that the prediction coverage by applying S_Searching is almost the same as that of fully searching the trust network for TARS, while the computational complexity is much less expensive.

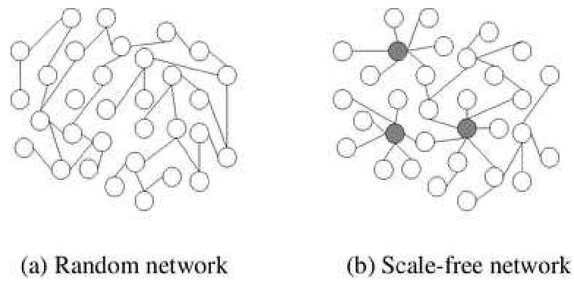


Fig. 1. Comparison between the structure of (a) random network and (b) scale-free network. In the scale-free network, the hubs are highlighted with black nodes

The rest of the paper is organized as follows: Section 2 presents the related works and the improved searching mechanisms based on the classical searching mechanism of the scale-free network; Section 3 gives our proposed recommender searching mechanism in details and gives the experimental results; Sections 4 concludes this paper and points out the future work.

2 RELATED WORKS

2.1 F_Searching

Despite those who did not mention their recommender searching mechanisms, existing models of TARS [8-10] choose to fully search the entire trust network to find recommenders. We call this recommender searching mechanism F_Searching for convenience in this paper. In

[1] The rating coverage is the portion of items that TARS is able to predict, i.e., the portion of items that the active users can get at least one recommendation. The recommender coverage is the portion of recommenders that could be involved in TARS.

F_Searching, both the selected nodes for the trust propagation and the trusted nodes are able to be the recommenders. TARS can achieve high prediction coverage by using F_Searching. However, its computational complexity is high [8]: $O(k^d)$, where k is the average degree of the trust network and d is the trust propagation distance.

2.2 C_Searching

The most classical and influential searching mechanism of scale-free networks is the one proposed by Pastor-Satorras and Vespignani [11]. We call it C_Searching in this paper. The key idea of C_Searching is: choosing the highest-degree node at each step of the trust propagation, and then choosing the nodes connected to the selected node at the next step. Two examples of C_Searching are given in Fig. 2. It has been verified that C_Searching can achieve high coverage for most scale-free networks [11]. Its computational complexity is $O(d)$, where d is the trust propagation distance. The computational complexity of C_Searching is much less expensive than that of F_Searching.

We use the system model mentioned in [8] and the public released TARS dataset Epinions, which is available at trustlet.org[2], to compare the performances of F_Searching and C_Searching. Epinions consists of 49288 users and 487183 trust statements. 15328 users of Epinions do not have outdegree, which means they do not trust any other user. It is impossible for TARS to predict ratings for them. In addition, 1543 users consist very small sized subnetworks (maximum size: 13, average size: 1.95). It is meaningless to predict ratings for them from the statistical point of view. We therefore focus on the other 32417 users of Epinions dataset. There are totally 461757 trust relationships between them. We call the selected data Epinions+ dataset. The indegree and outdegree distribution of the trust networks used in Epinions and Epinions+ dataset are given in Fig. 3: it clearly shows that both trust networks are the scale-free networks. Since the scale of Epinions+ dataset is large, we randomly choose 50 users from Epinions+ as the active users, predicting ratings for them on 706 items. The recommendations are totally from 14334 recommenders. The performances of all recommender searching mechanisms mentioned in this paper are verified on these data by using matlab.

The performances of C_Searching are given in Table 1. Comparing with F_Searching, it clearly shows that C_Searching does not perform well on finding recommenders for TARS. This is because C_Searching has a basic assumption [11]:$2 < \gamma < 3$. Though the authors of [11] expected that most scale-free networks fulfill this assumption, we have shown in [9] that for the public released trust networks, $\gamma < 2$. The experimental results shown in Fig. 3 also verified the small values of γ in our experimental data. Comparing the two examples given in Fig. 2, it clearly shows that the coverage of C_Searching is strongly related to the outdegree of the selected node at each step. If γ is small, the selected highest-degree node at each step of the trust propagation cannot cover superior number of nodes. This leads to the limited coverage of C_Searching. Moreover, the smaller γ is, the less the coverage C_Searching has.

[2] $http://www.trustlet.org/wiki/Datasets$

Table 1. Performance comparison between C_Searching and F_Searching in TARS

	Rating Coverage	Recommender Coverage
F_Searching	100.00%	98.14%
C_Searching	86.53%	19.51%

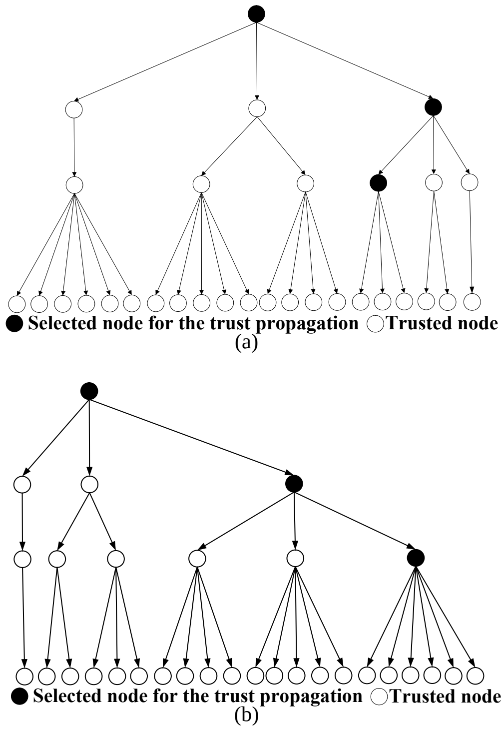


Fig. 2. Two examples of C_Searching: the highest-degree node is chosen at each step of the trust propagation

2.3 Our Improvement on C_Searching

Since C_Searching is not effective in TARS, we first try to improve it to achieve better performances. The heuristic improvement of C_Searching is to choose more nodes at each step of the trust propagation. We propose two searching mechanisms: C+_Searching and C++_Searching. The key idea of C+_Searching is: choosing the top N highest-

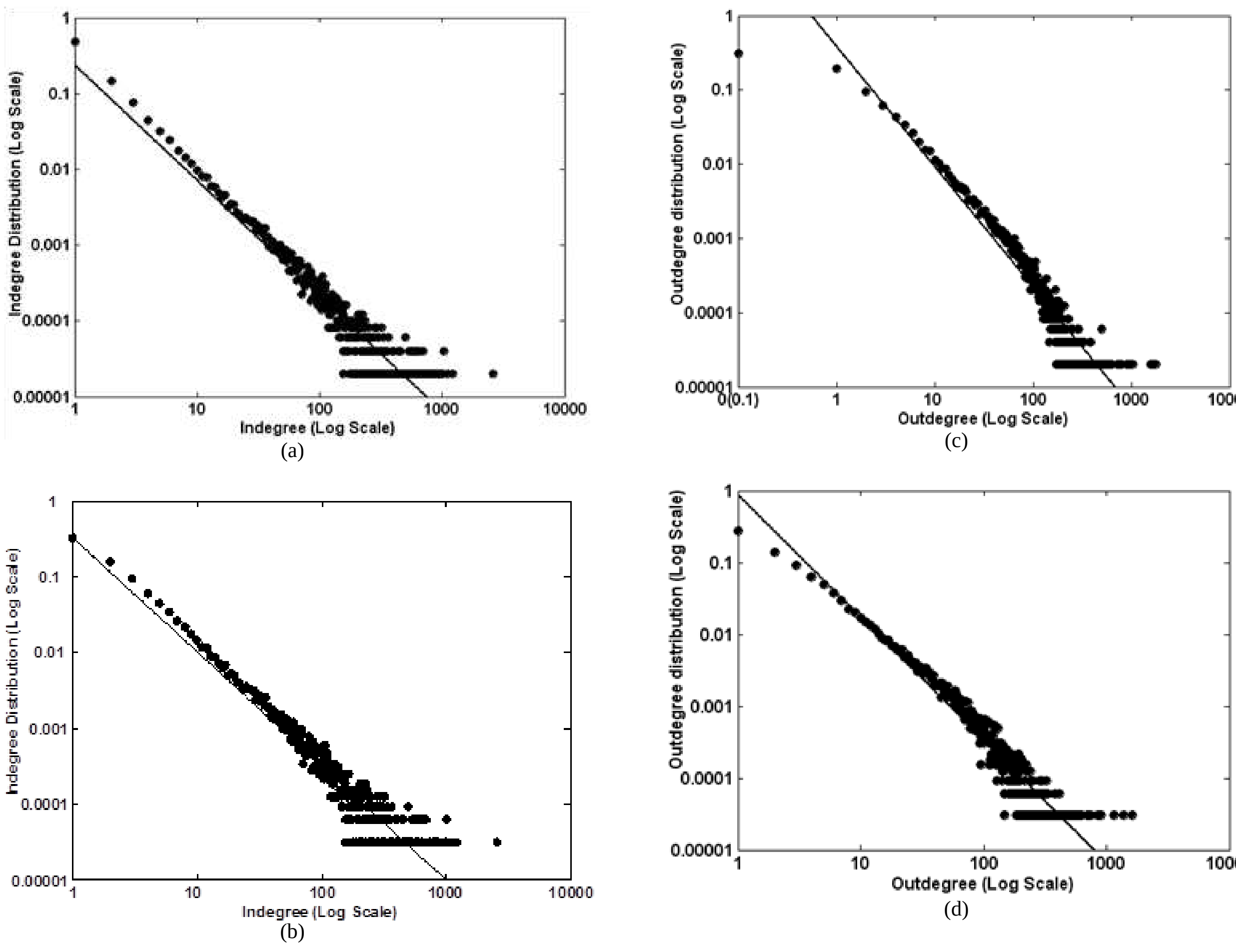


Fig. 3. The indegree distribution of (a) Epinions, γ=1.53, and (b) Epinions+, γ=1.51. The outdegree distribution of (c) Epinions, γ=1.62, and (d) Epinions+, γ=1.71

degree nodes at each step of the trust propagation, and then choosing the nodes connected to the selected nodes at the next step. The key idea of C++_Searching is: choose the top N highest-degree nodes for each selected node at each step, and then choose the nodes connected to the selected nodes at the next step. Examples of C+_Searching and C++_Searching are given in Fig. 4, in which two highest-degree nodes are selected at each step. It clearly shows that C++_Searching can cover more nodes than C+_Searching. The computational complexity of C+_Searching is $O(Nd)$, while the computational complexity of C++_Searching is $O(N^d)$, where N is the number of selected nodes at each step of trust propagation and d is the trust propagation distance.

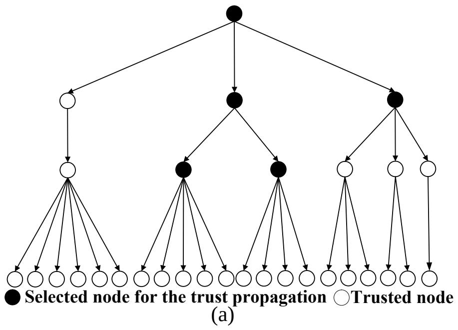
Experiments are held on the data mentioned in Section 2.2 to verify the performances C+_Searching and C++_Searching. The experimental results are given in Fig. 5.

Comparing with C_Searching:

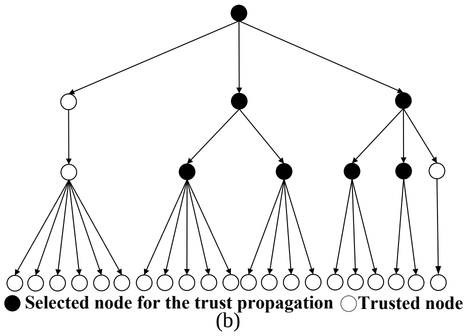
1. The advantages of C+_Searching and C++_Searching lie in
   (a) Both the rating coverage and the recommender coverage are improved, especially the recommender coverage; the prediction coverage of C++_Searching is better than that of C+_Searching;
   (b) Both the rating coverage and the recommender coverage of TARS increases as N increases;
2. The limitation of C+_Searching and C++_Searching is that they are computational more expensive.

Comparing with F_Searching:

1. The advantage of C+_Searching is that it is computational much less expensive;
2. The advantage of C++_Searching is that if N is smaller than k, it is computational less expensive;



(a)



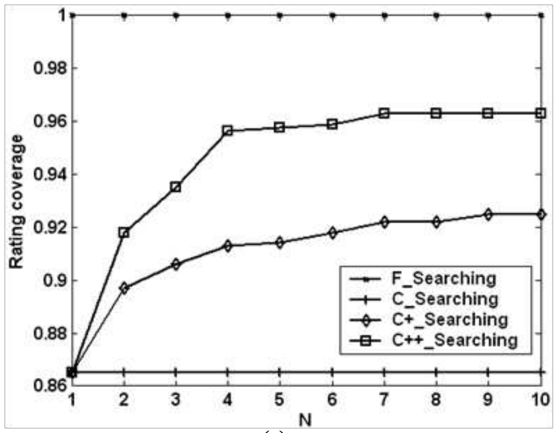
(b)

Fig. 4. An example of (a) C+_Searching, in which two highest-degree nodes are selected at each step of the trust propagation, and (b) C++_Searching, in which two highest-degree nodes are selected for each selected nodes at each step of the trust propagation

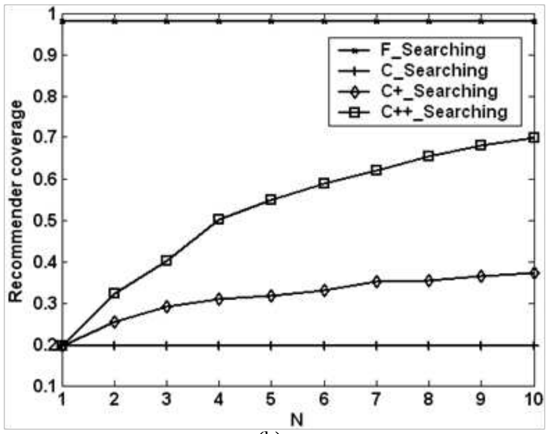
3. The limitation of C+_Searching and C++_Searching is that their coverage, especially the recommender coverage, is still very limited; if N is no less than k, the computational complexity of C++_Searching is no less expensive than that of F_Searching.

3 OUR PROPOSAL: S_SEARCHING

We have shown in Section 2 that the searching mechanisms can effectively improve their performances by choosing more nodes at each step of the trust propagation. However, their prediction coverage, especially the recommender coverage, is still limited. This is because: though C+_Searching and C++_Searching choose highest-degree nodes at each step, since γ is small, these nodes' outdegrees may not be very superior in terms of the entire network, i.e., the number of nodes covered by these selected nodes may be limited. This indicates that TARS cannot achieve satisfactory prediction coverage by depending on the local highest-degree nodes at each step of the trust propagation. We therefore propose an efficient recommender searching mechanism, named S_Searching, in this section.



(a)



(b)

Fig. 5. Comparison of (a) rating coverage and (b) recommender coverage by using C++_Searching, C+_Searching, C_Searching and F_Searching in TARS

3.1 Principal of S_Searching

Though γ is small, since the trust network is the scale-free network, there always exist a number of hubs. It is more efficient for the recommender searching mechanism to find the recommenders by benefiting from the superior outdegrees of these hubs. Based on the scale-freeness of the trust network, our proposed S_Searching aims at achieving high prediction coverage for TARS. The key idea is: instead of choosing one or more local highest-degree nodes at each step of the trust propagation, we select a number of hubs in the trust network to construct a skeleton, and then choose the nodes connected to the skeleton to find the recommenders. The relationships between n, n_S and k, k_S are:

$$n >> n_S, \tag{1}$$

$$k << k_S, \tag{2}$$

where n is the size of the trust network, n_S is the size of the skeleton, k is the average degree of the trust network and k_S is the average degree of the skeleton.

We regard the skeleton as one super node in S_Searching: the node trusted by any node of the skeleton is regarded as the node trusted by the skeleton, and if a node trusts any node of the skeleton, it is regarded as trust the skeleton. To find a recommender for an active user, S_Searching first connects the active user to the skeleton with the shortest path of trust propagation, and then find the recommender from the skeleton with the shortest path of trust propagations. An example of S_Searching is given in Fig. 6, in which the active user is connected to the skeleton with two hops of trust propagation and the skeleton is connected to the recommender with one hop of the trust propagation.

The computational complexity of S_Searching consists of three parts: 1) connecting the active users to the skeleton, 2) searching inside the skeleton, and 3) connecting the skeleton to the recommenders. Since $n_S << n$, the computational complexity of searching inside the skeleton is much less expensive than the other two operations, so the computational complexity of S_Searching is $O(k^{d_{AS}} + k^{d_{SR}})$:

$$O(k^{d_{AS}} + k^{d_{SR}}) = O(k^{\max(d_{AS},d_{SR})}), \qquad (3)$$

where d_{AS} is the trust propagation distance from the active user to the skeleton, d_{SR} is the trust propagation distance from the skeleton to the recommender and k is the average degree of the trust network

The computational complexity of the above five mentioned recommender searching mechanisms is summarized in Table 2. Since $\max(d_{AS}, d_{SR}) < d$, the computational complexity of F_Searching and is much less than that of S_Searching

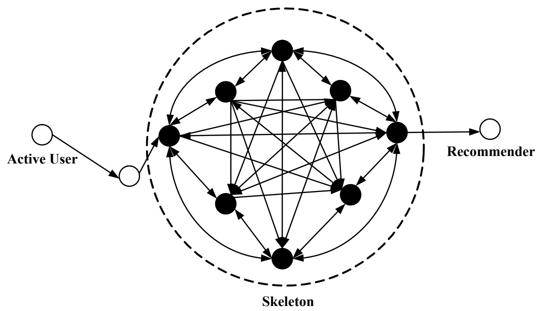


Fig. 6. An example of S_Searching, in which the Skeleton is composed of a number of hubs in the trust network. The active user is first connected to the Skeleton, and then S_Searching finds the recommenders from the Skeleton for the active user

Table 2. Performance comparison between C_Searching and F_Searching in TARS

	Computational Complexity
F_Searching	$O(k^d)$
C_Searching	$O(d)$
C+_Searching	$O(Nd)$
C++_Searching	$O(N^d)$
S_Searching	$O(k^{\max(d_{AS},d_{SR})})$

Table 3. Detailed information of the five Skeletons used in this work

	Outdegree	**Num of node**	**Num of trust relationships**
Skeleton1	>=100	829	49856
Skeleton2	>=200	190	6031
Skeleton3	>=300	74	1357
Skeleton4	>=400	33	314
Skeleton5	>=500	16	78

3.2 Experimental Verification on the Effectiveness of S_Searching in TARS

Experiments are held on the data shown in Section 2.2. Based on the outdegree distribution of the trust network in Epinions+, as shown in Fig. 3, we choose five skeletons for the experiments held in this section. Their detailed information is given in Table 3.

The distributions of d_S (the trust propagation distance in the skeleton), d_{AS} (the trust propagation distance from the active user to the skeleton) and d_{SR} (the trust propagation distance from the skeleton to the recommender) are given in Fig. 7. For the selected five skeletons:

(1) Users inside the skeletons can connect to each other within 4 hops of trust propagations, in which most users can connect to others within 3 hops;

(2) Users can connect to the skeleton within 7 hops of trust propagations, in which most users can connect to the skeleton within 3 hops;

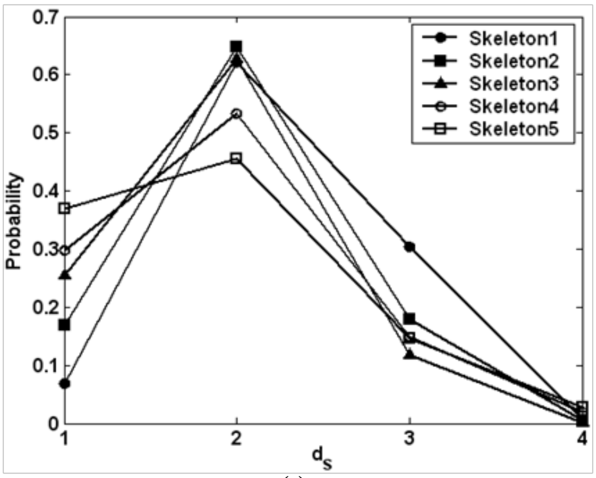
(3) Skeletons can connect to other users within 7 hops of trust propagations, and the other four skeletons can connect to other users within 8 hops of trust propagations; specifically, the skeletons can connect to most users within 3 hops;

(4) The larger the scale of the skeleton is, the shorter d_{AS} and d_{SR} are. This is because if the skeleton consists of more nodes, it is easier for a user to build its trust relationship to some nodes of the skeleton, and it is also easier for the skeleton to cover more nodes. The average path lengths of d_S, d_{AS} and d_{SR} are given in Table 4.

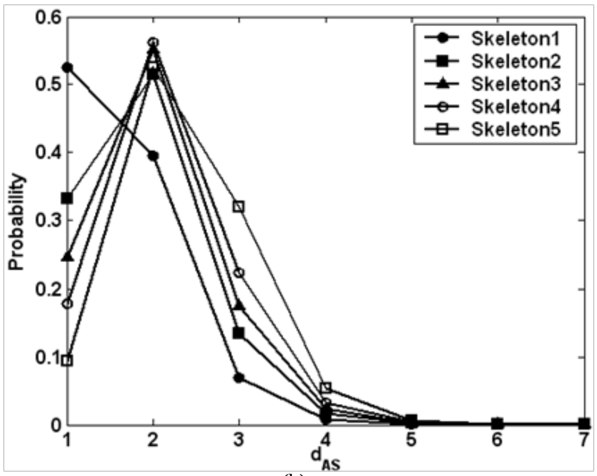
Using the selected five skeletons in S_Searching, the recommender coverage and the rating coverage of TARS

Table 4. Average path length of the trust propagation distance

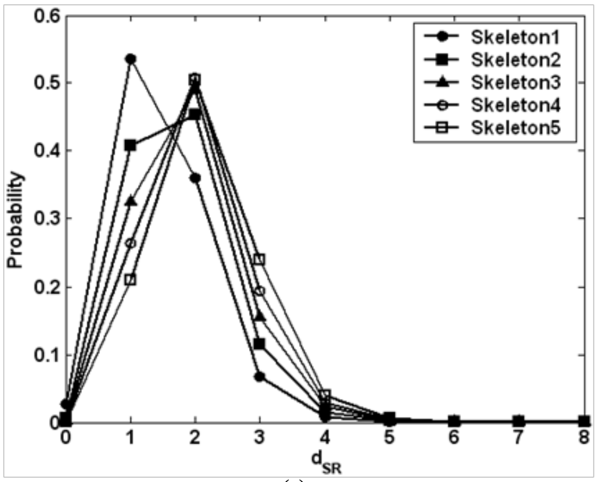
	$\overline{d_{AS}}$	$\overline{d_S}$	$\overline{d_{SR}}$
Skeleton1	1.56	2.25	1.54
Skeleton2	1.84	1.84	1.74
Skeleton3	1.99	1.99	1.89
Skeleton4	2.12	2.12	2.00
Skeleton5	2.36	2.36	2.13



(a)



(b)



(c)

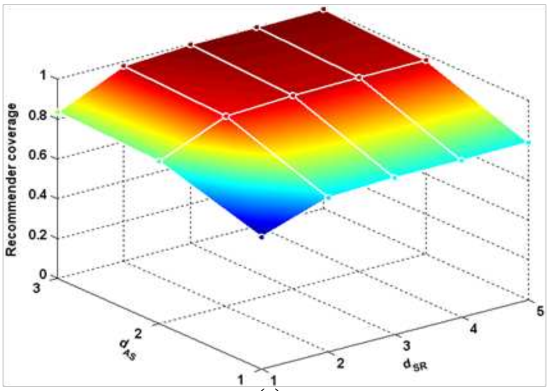
Fig. 7. The distribution of (a) d_S, (b) d_{AS}, and (c) d_{SR}

are given in Fig. 8 and Fig. 9 respectively. The experimental results show that:

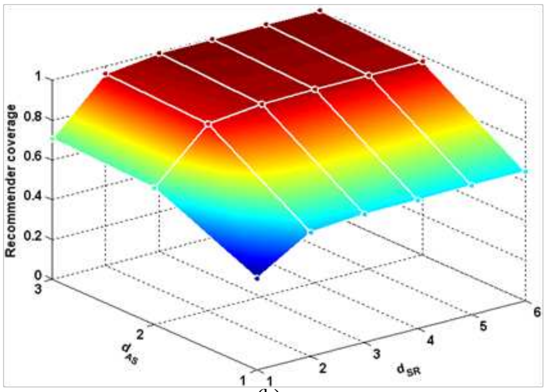
(1) TARS can achieve better prediction coverage with larger d_{AS}, especially in case the scale of the skeleton is getting smaller. And this is more obvious for the rating coverage. This is because if d_{AS} is larger, more users can connect to the skeleton, TARS can therefore predict ratings for more users. In addition,d_{AS} is more influential to the rating coverage than the recommender coverage due to its relationship to the active users. With the decreasing of the skeleton's scale, the number of the nodes in the skeleton is getting less; it is less probable to build up the trust relationships between the active users to some nodes in the skeleton within limited hops of trust propagations. So d_{AS} is more influential to the small scaled skeletons.

(2) TARS can achieve better prediction coverage with larger d_{SR}, however, when d_{SR} is bigger than some value, the changing of the prediction coverage is getting very slightly. This is more obvious for the recommender coverage. It is shown that: for the recommender coverage, it is much better if d_{SR} is set to be 2 than 1, it is better if d_{SR} is set to be 3 than 2, and there is no big difference if d_{SR} is set to be bigger than 3; for the rating coverage, it is better if d_{SR} is set to be 2 than 1, and there is no big difference if d_{SR} is set to be bigger than 2. This is because if d_{SR} is set to be 3, S_Searching can find most nodes in the trust network, as shown in Fig. 7, which is the basis for achieving high recommender coverage. In addition, since to enlarge d_{SR} is able to evolve more recommenders in TARS, the value of d_{SR} is more influential to the recommender coverage than the rating coverage.

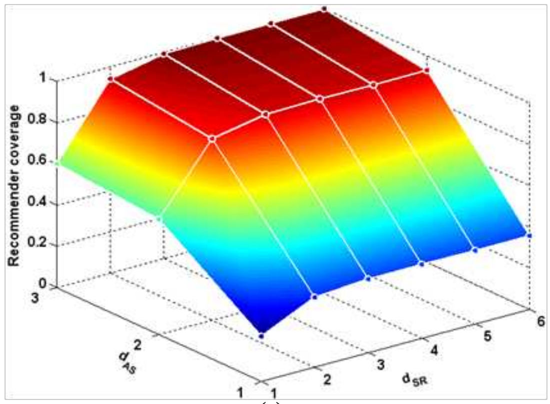
We further compare the performances of S_Searching with other recommender searching mechanisms mentioned in this paper. Comparing Fig. 5 with Fig. 8 and Fig. 9, it is shown that: in case d_{AS} is set to be maximum, i.e., set $d_{AS} = 3$ in our experiments, and d_{SR} is set to be a suitable value, i.e., set $d_{SR} = 3$ in our experiments, the recommender coverage and the rating coverage of S_Searching are almost the same as those of F_Searching, which are much better than C_Searching, C+_Searching and C++_Searching; if we set $d_{SR} = 2$ in our experiments, the rating coverage of S_Searching



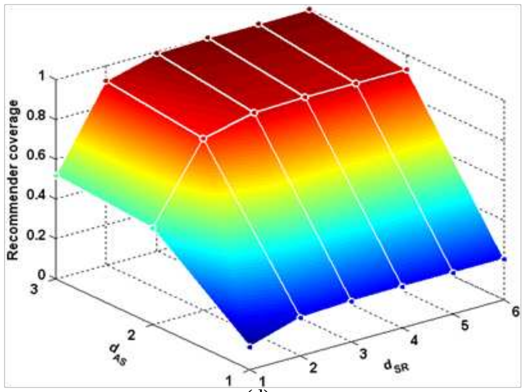
(a)



(b)



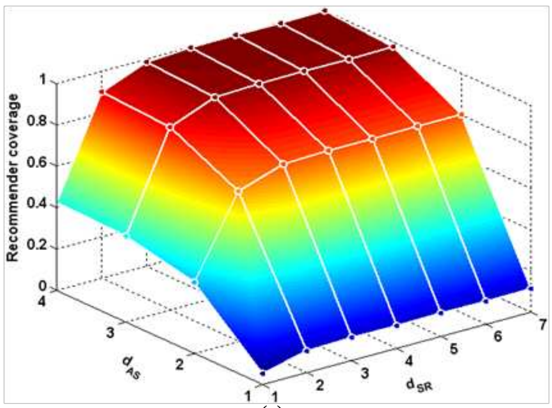
(c)



(d)

is almost the same as that of F_Searching, the recommender coverage of S_Searching is worse than that of F_Searching, while both coverages are much better than C_Searching, C+_Searching and C++_Searching. In our experiments, $k \approx 10$, $d = 5$, if we set $d_{SR} = 3$, considering the outdegree distribution of the trust network in Epinions+, as shown in Fig. 3, and the selection of the skeletons, as shown in Table 3, S_Searching is computational less expensive than F_Searching with the similar prediction coverage; if we set $d_{SR} = 2$, S_Searching is computational much less expensive than F_Searching with the similar rating coverage and slightly worse recommender coverage.

To sum up, by setting a reasonable value for d_{SR} (2 or 3 in our experiments) and maximizing d_{AS}, S_Searching can achieve much better prediction coverage than C_Searching, C+_Searching and C++_Searching. Moreover, its prediction coverage is almost the same as that of F_Searching, while the computational complexity is much less expensive. This means S_Searching can efficiently find satisfactory number of recommenders for TARS.

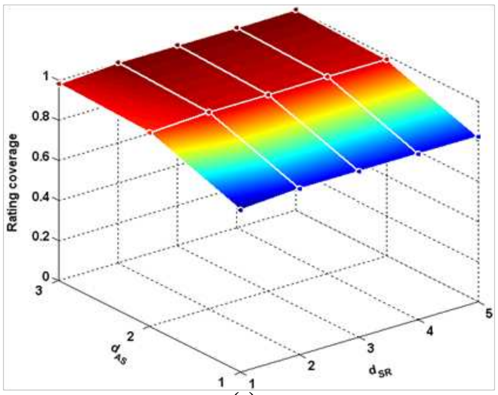


(e)

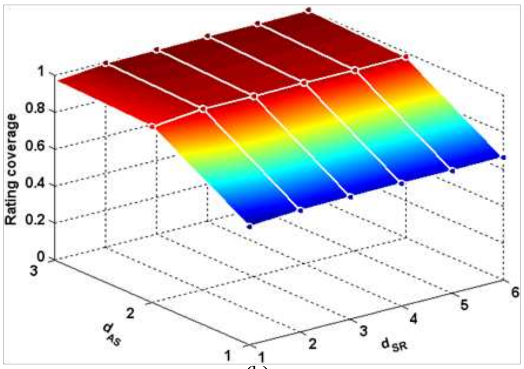
Fig. 8. Recommender coverage of TARS by using (a) Skeleton1, (b) Skeleton2, (c) Skeleton3, (d) Skeleton4, and (e) Skeleton5 in S_Searching

4 CONCLUSION

To improve the performance of TARS, it is essential to find satisfactory number of recommenders for the users efficiently. Existing works use F_Searching, which fully

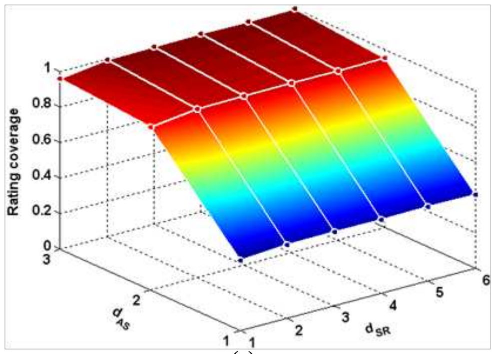


(a)

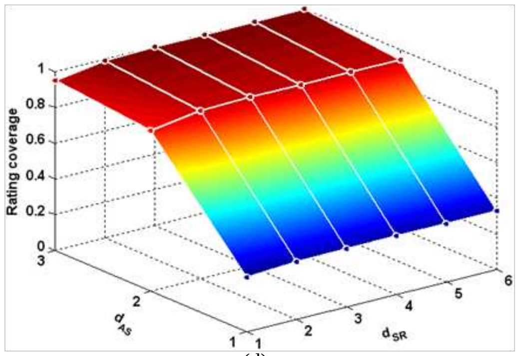


(b)

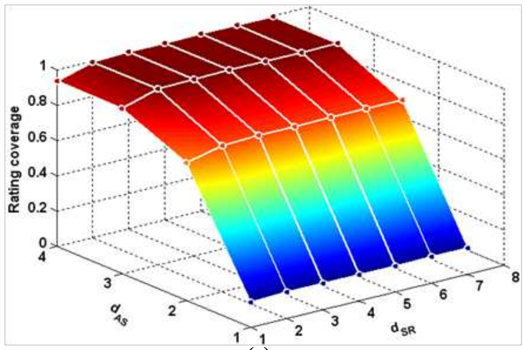
searches the trust network. It has high prediction coverage while it is computational very expensive. Though the trust network is the scale-free network [9], experimental results show that TARS cannot achieve high prediction coverage by directly applying C_Searching. This is because γ of the trust network is usually smaller than the requirements of C_Searching. We further verify that this problem cannot be fundamentally solved by increasing the number of nodes selected by C_Searching at each step of the trust propagation, i.e., to apply C+_Searching and C++_Searching. We propose a recommender searching mechanism on finding recommender efficiently for TARS, named S_Searching, based on the scale-freeness of the trust network. Different from other recommender searching mechanisms, S_Searching chooses a number of hubs to build up a skeleton, and finds the recommenders for the active users via the skeleton: the active users are connected to the skeleton, and the skeleton is responsible on finding the recommenders. Benefiting from the hubs' dominating power on connecting to other users, it is much easier for S_Searching to find the recommenders. Experimental results show that S_Searching has similar prediction coverage as F_Searching, which is much better than that of C_Searching, C+_Searching and C++_Searching, while it is computational much less expensive.



(c)



(d)



(e)

Fig. 9. Rating coverage of TARS by using (a) Skeleton1, (b) Skeleton2, (c) Skeleton3, (d) Skeleton4, and (e) Skeleton5 in S_Searching

In the future, we plan to focus on more details of the recommender searching mechanism used in TARS, such as finding the most reliable recommenders for the active users, in which a heuristic method is to make full use of the nodes with the high indegrees. Though the research on the recommender searching mechanism of TARS is still at the beginning stage, we do believe that it presents a promising path for the future research.

ACKNOWLEDGMENT

This research was supported by National Natural Science Foundation of China (Grant No. 61100007, 61100081, 61100006), and the collaborative research project under NSFC-NRF cooperative program (Grant No. 613111015). Dr. Donghai Guan is the corresponding author of this paper.

REFERENCES

[1] L. Atzori, and A. Iera, "The Internet of Things: A Survey", *Computer Networks*, vol. 54, pp. 2787-2895, 2010.

[2] N. Gershenfeld, R. Krikorian, and D. Cohen, "The Internet of Things", *Scientific American*, vol. 291, pp. 76-81, 2004

[3] G. Kortuem, "Smart Objects as Building Blocks for the Internet of Things", *Internet Computing*, vol. 14, No. 1, pp. 44-51, 2010.

[4] N. Wang, and W. Wu, "The Architecture Analysis of Internet of Things", In *Proceedings book of 5th IFIP TC 5/SIG 5.1 Conference*, pp. 193-198, CCTA 2011.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", CoRR abs/1207.0203, 2012.

[6] D. Giusto, A. Iera, G. Morabito, and L. Atzori (Eds.), "The Internet of Things", *Springer*, *ISBN: 978-1-4419-1673-0*, 2010.

[7] H. Xiaobin, "Application and Practice on the Internet of Things", *Information Construction*, No.11, pp. 21-22, 2009.

[8] W. Yuan, D. Guan, Y.K. Lee, S. Lee, and S. J. Hur, "Improved Trust-Aware Recommender System using Small-Worldness of Trust Networks", *Knowledge-Based Systems*, vol. 23, pp. 232-238, 2010.

[9] W. Yuan, D. Guan, Y.K. Lee, and S. Lee, "The Small-World Trust Network", *Applied Intelligence*, vol. 35 (**??**), pp. 399-410, 2011.

[10] W. Yuan, L. Shu, D. Guan, Y.K. Lee, and S. Lee, "iTARS: Trust-Aware Recommender System using Implicit Trust Networks", *IET Communications*, vol. 14, pp. 1709 – 1721, 2010.

[11] R. Pastor-satorras, and A. Vespignani, "Epidemic Spreading in Scale-Free Networks", Physics Review Letter, vol. 86, pp. 3200–3203, 2001.

[12] P. Massa, and P. Avesani, "Trust-Aware Collaborative Filtering for Recommender Systems", *In Proceedings book ff Federated International Conference on the Move to Meaningful Internet*, pp. 492-508, 2004.

[13] P. Massa, and P. Avesani, "Trust Metrics in Recommender Systems", In *Proceedings book of Computing with Social Trust*, pp. 259-285, 2009.

[14] Y. Li, and C. Kao, "TREPPS: A Trust-based Recommender System for Peer Production Services", *Expert Systems with Applications*, vol. 36, pp. 3263-3277, 2009.

[15] F. Walter, S. Battistion, and F.Schweitzer, "A Model of A Trust-Based Recommendation on A Social Network", *Autonomous Agents and Multi-Agent System*, vol.16, pp. 57-74, 2008.

[16] P. Massa, and P. Avesani, "Trust-Aware Recommender Systems", *In Proceedings book of the 2007 ACM Conference on Recommender Systems*, pp. 121-126, 2007.

[17] D. Watts, and S. Strogatz, "Collective Dynamics of 'Small-World' Networks", *Nature*, vol. 393, pp. 440-442, 1998.

[18] A.L. Barabási, "Scale-Free Networks: A Decade and Beyond", *Science*, vol. 325, pp. 412-413, 2009.

[19] A.L. Barabási, and E. Bonabeau, "Scale-Free Networks", *Scientific American*, 288 (**??**), pp. 60–69, 2003.

[20] A.L. Barabási, and R. Albert, "Emergence of Scaling in Random Networks", *Science*, 286 (5439), pp. 509–512, 1999.

[21] M. Kim, Y.S. Jeong, J.H. Park, and S.O. Park, "Group-Aware Social Trust Management for A Movie Recommender System", *In Proceedings book of Int'l Conference on Information Technology*, vol. 107, Part 5, pp. 495-501, 2011.

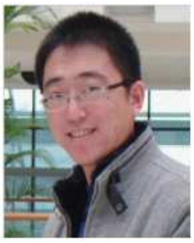
[22] J. Sun, and X. Yu, "Research on Trust Recommender Model based on Experts' Social Networks", *In Proceedings book of International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2503-2506, 2010.

[23] N. Sahli, and G. Lenzini, "Trust-Aware Recommender Systems for Open and Mobile Virtual Communities", *Collaborative Computer Security and Trust Management*, ch007, 2010

[24] S.C. Kim, J.W. Ko, J.S. Cho, and S.K. Kim, "Collaborative Filtering Recommender System based on Social Network", *In Proceedings book of Int'l Conference on Information Technology*, vol. 107, Part 5, pp. 503-510, 2011.

[25] S. Ray, and A. Mahanti, "Improving Prediction Accuracy in Trust-Aware Recommender Systems", *In Proceedings book of Hawaii International Conference on System Sciences*, pp. 1-9, 2010.

Weiwei Yuan received her B.S. and M.S. in Automation and Computer Engineering in 2002 and 2005 respectively from Harbin Engineering University, China. She got her Ph.D. degree in Computer Science from Kyung Hee University, South Korea in 2010. She is currently an assistant professor in Harbin Engineering University, China. Her research interests are trust-aware computing, social computing and machine learning.



Donghai Guan received his B.S. in College of Automation from Harbin Engineering University, China in 2002, and his M.S. degree in Computer Science from Kumoh National Institute of Technology, South Korea in 2004, and his Ph.D. degree in Computer Science from Kyung Hee University, South Korea in 2009. He is currently an assistant professor in Harbin Engineering University, China. His research interests are Machine Learning, Pattern Recognition, Data Mining, Activity Recognition, and Trust modeling.

Lei Shu received his B.Sc. degree in Computer Science from South Central University for Nationalities, China, 2002, and the M.Sc. degree in Computer Engineering from Kyung Hee University, Korea, 2005, and the PhD Degree in Digital Enterprise Research Institute, National Univeristy of Ireland, Galway, 2010. He is currently a full professor in Guangdong University of Petrochemical Technology, China. His research interests include wireless sensor networks and mobile multimedia sensor networks.

Jianwei Niu received his Ph.D. degrees in 2002 in computer science from Beijing University of Aeronautics and Astronautics (BUAA). He was a visiting scholar at School of Computer Science, Carnegie Mellon University, USA from Jan. 2010 to Feb. 2011. He is now a professor and Ph.D. advisor of Beihang University. His current research interests include mobile and pervasive computing, mobile video analysis.

AUTHORS' ADDRESSES

Asst. Prof. Weiwei Yuan, Ph.D.
Department of Computer Science and Technology,
Asst. Prof. Donghai Guan, Ph.D.
Department of Automation,
Harbin Engineering University,
Nantong Street 145, 150001, Harbin City, China
email: yuanweiwei@hrbeu.edu.cn, dhguan@gmail.com

Prof. Lei Shu, Ph.D.
Guangdong University of Petrochemical Technology,
Industrial Security and Wireless Sensor Networks Lab,
No. 139 Guandu Erlu, Maoming City, Guangdong, China
email: lei.shu@lab.gdupt.edu.cn
Prof. Jianwei Niu, Ph.D.
Beihang University University,
School of Computer Science and Engineering,
G1030, New Main Building, Beihang University, Beijing, China
email: niujianwei@buaa.edu.cn

Received: 2012-11-10
Accepted: 2013-03-12