2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)

An efficient and fast influence maximization algorithm based on community detection

Esmaeil Bagheri, Gholamhossein Dastghaibyfard, Ali Hamzeh
Department of Computer Science and Engineering and Information Technology
School of Electrical Computer Engineering, Shiraz University
Shiraz, Iran

Abstract— **Influence maximization specifies a set of nodes that maximizes the influences in social networks. The influence maximization problem due to its importance in targeted marketing has been explored by many researchers. All proposed algorithms are not scalable and are too time consuming for large-scale social network. In this paper, an efficient and fast algorithm called ComPath+ is proposed for influence maximization based on community detection. ComPath+ enhances ComPath algorithm investigate the small number of nodes and preserve quality of seeds. The results show that proposed algorithm is more efficient and much faster than current algorithms.**

Keywords-component; Social networks; Influence maximization; community detection;

I. Introduction

Social Networks allow people to establish social connections and facilitate interactions and share things between friends. Friendship among people of a social community with the intention to form an excellent advertising medium, as the scale of person bases in social networks fast grows through the years. Social networks play a key role in new world. People can have an impact on each other a good deal extra without problems. This easy influence causes many researchers find new and more efficient methods for finding influential people among social networks that maximize influence on other people. Maximizing influence or influence maximization has many usage in real and virtual world for example marketing, opinion propagation, and acceptance of new things.

Marketing on social networks has important issue that is finding influential individuals. For example, advertising a new product thru social networks or an organization may also need to goal a small quantity of users for a test of a product, hoping that those users could have an effect on their buddies to buy the product or grow to be acquainted with it. Via influential users the company ought to attain a big range of potential clients. This idea is described as the influence maximization problem. It selects the preliminary seeds who might also impact a maximal wide variety of people to include an advertised product [1].

Our method selects the most important nodes that maximize influence based on the linear threshold (LT) model. Experiments show that sour proposed algorithm to find the influential nodes is more efficient and much faster than the state of the art algorithms.

The rest of the paper is prepared as follows: Section 2 reviews related works. In section 3 the ComPath algorithm [2] is presented. The proposed algorithm is discussed and is compared with other algorithms in Section 4. In section 5 experimental results are shown and the paper conclusion is presented in Section 6.

II. Related Works

In this section, several related works are reviewed and classified as Linear threshold model (LT), community detection, and influence maximization algorithms.

Kempe et al. [3] 2003 introduced several information diffusion models, where two famous ones are: the linear threshold (LT) model and the independent cascade (IC) model. In LT model a threshold value θ_v in [0, 1] is assigned to every node, indicating how much a node tends to be influenced by its active neighbours. Where there is a lack of sufficient information about the social network, this value is selected for all of the nodes uniformly at random. A node v is influenced by each of its neighbours w by a weight $b_{v,w}$, where $\Sigma_{w \text{ neighbours of } v}\ b_{v,w} \leqslant 1$ is held. Assuming A_0 to be the initial set of active nodes, the activation process proceeds in discrete steps as follows: in step t in addition to the active nodes of step t − 1, every node v is subject to activation formula $\Sigma_{w \text{ neighbours of } v}\ b_{v,w} \geqslant \theta_v$ will become active. This process continues until no other activation is possible. In this model the influence function is monotone and submodular [3].

A community is defined as a subset of users who interact with each other greater frequently than other users outside the community [4]. Two classifies of community detection algorithms are overlapping and non-overlapping. Here we review overlapping community detection algorithms:

- Clique percolation algorithms: These algorithms have most efficient to locate powerfully linked sub-graphs to estimate the real communities among overlapping community detection algorithms. Clique percolation algorithms are popular approachs for analyzing the overlapping community structure of networks[5].
- Label propagation algorithms: These algorithms distribute specific labels at the input graph primarily based on some predefined roles and eventually a

978-1-5090-4093-3/16/$31.00 ©2016 IEEE

community is specified from the nodes with the same label[6].

- Link partitioning algorithms: these algorithms partition iteratively the input graph into several graphs that the final graphs are considered as the graph communities[7].
- Local expansion and optimization algorithms: These algorithms extend natural communities or a partial community. Also those algorithms focus on a local profit feature that attempts to discover the nodes which are densely linked to every other [8].

We use very fast algorithm called SLPA[9] presented by Xie et al., 2011 for community detection in this paper. SLPA (Speaker-listener Label Propagation Algorithm) uses listener-speaker interaction to simulates information diffusion process. It gives a label to each node and then spreads labels among nodes based totally on some regulations. every node keep the labels that it gets in a memory in different steps of the set of rules. A node is considered as a member of a community based totally on opportunity of observing a community label in a node's memory.

The influence maximization problem was introduced by Domingson and Richardson [10] in 2001 and the preliminaries of this problem that considers information diffusion models were first presented by Kempe et al. [3]. The influence maximization problem is NP-hard problem under the context of various diffusion models [3]. Most of the proposed algorithms try to obtain approximate solutions. An influential person is the one with many friends. Selecting seeds based on their degree (degree centrality) is an extensively followed heuristic to cope with the influence maximization problem. Large communities have more nodes and then have higher degree centrality, therefore in small communities degree centrality is low. As a result, degree centrality may also without problems bring about seeds within the equal big network. The variety of influenced nodes by seeds in the same network generally tend to overlap. Distance centrality is any other heuristic algorithm which selects seeds in order of increasing average distance to other nodes. Normally nodes have a small average distance in big communities, so distance centrality additionally effects in seeds in the identical massive network. Briefly degree centrality and distance centrality result degeneration in influence spread singly. Some well-known algorithms of influence maximization are:

- Page Rank: Brin and Page [11] in 1998 proposed for ranking webpages based on their significance within the web graphs. Page Rank is used where top k nodes are selected.
- Maximum Degree: Kempe et al. [3] in 2003 proposed an algorithm that selects top k nodes with maximum degree as seeds. Their method use the degree centrality measure.
- MC-CELF: Leskovec et al. [12] in 2007 proposed a greedy algorithm for influence maximization that is genuinely an optimized version of CELF. They use Monte Carlo simulation that it run for 10,000 times to gain the final seeds. Therefore, it is not suitable for big graphs because this algorithm is slow.
- LDAG: Chen et al. [13] in 2010 proposed this algorithm that runs on the input graph to check creation of local DAG (directed acyclic graphs) or LDAG for every node. They assumed every node could only influence a limited number of its neighbors. Therefore, a LDAG used for every node. Their experimental results proved that this heuristic algorithm is more efficient and faster than other greedy algorithm.
- CELF++: This algorithm that presented by Goyal et al. [14] in 2011 is more efficient and faster version of CELF [12]. CELF is a greedy algorithm and sub-modularity agent proposed by Leskovec et al., in 2007.
- CIM: Chen et al. [1] in 2014 considered influence maximization under the heat diffusion model and developed a hierarchical community detection algorithm named *H-clustering* to detect the communities of the social network effectively. Next, by considering the size of the communities and the connection between them, candidate set generation is conducted through specifying some significant communities and selecting candidate nodes from them. Finally, the target set is selected heuristically where target nodes are identified based on their position in the communities.
- PaS: Ok et al. [15] in 2014 investigated maximizing diffusion speed of a new innovation. Their method use a noisy game-based model. This is done by seeding a subset of users. They obtained new topological insights for influence maximization by analyzing three representative graph (Erdos-Renyi, planted partition and geometrically structured graphs). Their results show that a careful seeding is not necessary for globally well-connected graphs but a good seeding for locally well-connected graphs is needed.
- CGA: Song et al. [16] in 2015 proposed an algorithm for finding most influential nodes. First CGA (Community-based Greedy algorithm) by information diffusion divides the large-scale mobile social network into several communities and then by use of a dynamic programming it selects communities for finding influential nodes. They parallelized their method to further improve the performance.

ComPath [2] is a new algorithm presented by Rahimkhani et al., 2015 that has better performance than above algorithms. This algorithm will be explained in next section.

III. ComPath Algorithm

ComPath is an influence maximization algorithm based on the LT model. Their algorithm first detects the communities and then to lessen the execution time it investigates a limited number of communities. To investigate less nodes from input graph, ComPath can suggest a limited number of communities.

They introduced two modifications to computing the influence spread of nodes in LT model formula. First they use betweeness centrality measure, community detection and selecting the candidate nodes to limit the number of nodes to be investigated. Then they limit input network by limit the scope to find seeds. The size of scope depends on the candidate nodes set and the selected path length. Figure 1 shows the steps of the ComPath algorithm. First, communities of the input network are extracted by SLPA community detection algorithm. Then detected communities form the new network that each community show a node in new network. Then betweenness centrality measure is used to detect the most central nodes of the new network. A set of nodes as candidate nodes are selected from important community based on degree centrality based on the size of the community. Then top-k most influential nodes are chosen from the candidate set. The paths in the graph are used to this selection. Details of this algorithm are showed in Algorithm 1.

Algorithm 1: ComPath(G,L,k)

```
Input: Graph G(V,E) and k number of needed seeds
Output: k seed set
Step 1: Compute New_Network
N = Read the input graph G(V,E);
Node = nodes of N;
CN = Detect communities of N;
Labels(Node(i)) = index of the CN(i) thet encompasses Node(i);
New_Network = NewNetwork(N,CN,Labels);
Step 2: Compute Selected_Nodes
CN = Detect communities of N Compute New_Network;
Compute betweenness of nodes in New_Network;
y = Sort nodes of New_Network in ascending order based on their betweenness;
Noc(i) = number of nodes in CN(i) that encompasses y(i);
Nmin = Size of the smallest community in CN;
Nmax = Size of the largest community in CN;
CandidatesCN(i) = [ ( Noc(i) / Nmax – Nmin ) * $\beta$ ] + $\alpha$ ;
Selected_Nodes = select CandidatesCN(i) number of nodes with the highest degree within the community CN(i) that encompasses y(i);
Step 3: Compute Seeds Set
u = Select a node from Selected_Nodes;
Seeds = {};
V' = (Selected_Nodes) $\cup$ {neighbors (Selected_Nodes,L)};
$\delta$(u) = BackTrack(u,V'-Seeds,L);
Seeds = Select k nodes with maximum $\delta$(u);
Output k seeds;
```

IV. The Proposed Algorithm

Our algorithm steps is almost similar to ComPath except in step 2. In step 2, ComPath only use betweenness centrality measure to detect the most important nodes of the new network. This method is time consuming for big networks and is not accurate. Because it doesn't consider power of connections that exists between nodes of two communities. For example in Figure 2, ComPath uses one edge from community1 to community2 for creating new network but there are 3 edges from nodes of community1 to nodes of community2 in input graph. In other word influence of nodes of community1 on nodes of community2 is more than influence of nodes of community2 on nodes of community1. Our algorithm named ComPath+ computes number of connections between nodes of two communities and assigns this number as weight to the edge that create in the new network in step 2. Figure 3 shows this result in contrast to Figure 2.

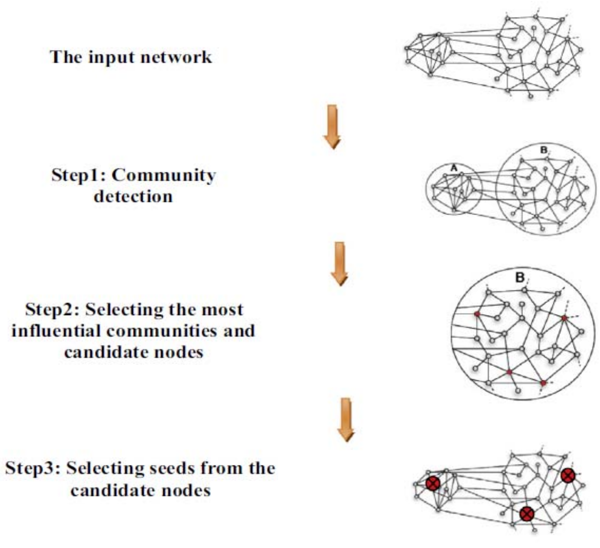

The input network
Step1: Community detection
A
B
Step2: Selecting the most influential communities and candidate nodes
B
Step3: Selecting seeds from the candidate nodes

Figure 1. The steps of the ComPath algorithm [2]

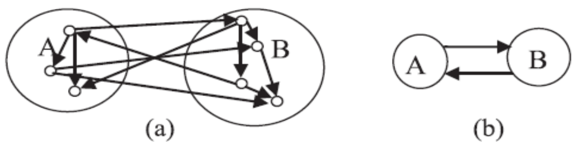


Figure 2. (a) Two sample communities and (b) its reduced graph in ComPath[2]

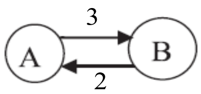


Figure 3. Corresponding reduced graph in ComPath+ with weighted edges.

Our algorithm will be explained in the below.

A. Step1: Community Detection with SLPA

In the first step, communities of the input graph are detected by SLPA algorithm [9]. SLPA uses listener-speaker interaction to simulates information diffusion process. It gives a label to each node and then spreads labels among nodes. It use some rules to spread labels. Every node keep the labels that it gets in a memory in different steps of the set of rules. A node is considered as a member of a community based totally on opportunity of observing a community label in a node's memory.

B. Step2: Creating New Network

Each community from previous step is use as a node to construct new graph G = (V, E). Where the set of nodes (communities) is shown by V and the weighted directed edges by E. A weighted directed edge (A, B, W) shows that W number of members of the community A are linked to members of community B in the initial input graph. It must be noted that ComPath algorithm doesn't consider W weights.

C. Step3: Selecting Important Communities

Degree centrality is efficient measure for selecting important communities in new network. High degree centrality value for a node (community) means high weight to the other nodes (communities). Our algorithm considers only out degree centrality measure, but current algorithms especially ComPath don't have this step and use only centrality measures instead and also don't consider power of connections between nodes (communities).

D. Step4: Selecting Important Nodes

Similar to previous step, in this step important nodes are selected among important communities by out degree centrality measure based on community quota. If a community has more nodes it has more quota. A node with high out degree centrality value means it has more communication links to its neighbors in its community.

E. Step5: Selecting Seed Nodes

This step select seed nodes set and similar to ComPath. It selects k nodes from the output of previous stage such that enabled them to maximum influence in the initial input graph. To find the final seed nodes set, any routes from each node to other nodes in the graph is selected and nodes with maximum influence is achieved. First, the route is considered equal to 2 and the seed nodes are selected with the greatest efficiency. Then the route is increased and seeds will be recalculated. If the average of seeds is increased compared with previous length of the route, the route length will be increased and the seeds will be recalculated.

Details of our algorithm are depicted in Algorithm 2.

Algorithm 2: ComPath+(G,L,k)

```
Input: Graph G(V,E) and k number of needed seeds
Output: k seed set
Step 1: Community Detection with SLPA
N = Read the input graph G(V,E);
Node = nodes of N;
CN = Detect communities of N with SLPA;
Step 2: Creating New Network
New_Network = NewNetwork(N,CN)
Compute W = number of members of the community A that have a link to members of community B in initial input graph
Update New_Network with W weights
Step 3: Selecting  Important Communities
Compute degree centrality of nodes in New_Network based on W weights;
y = Sort nodes of New_Network in ascending order based on their degree centrality;
Step 4: Selecting  Important Nodes
Community quata = k * (community nodes / all nodes)
Important_Nodes = Select important nodes from each community based on community quata
Step 5: Selecting  Seed Nodes
u = Select a node from Important _Nodes;
Seeds = {};
V' = (Important _Nodes) ∪ {neighbors (Important _Nodes,L)};
δ(u) = BackTrack(u,V'-Seeds,L);
Seeds = Select k nodes with maximum δ(u);
Output k seeds;
```

Steps 2,3,4 of Algorithm2 (ComPath+) is equal to step2 of algorithm1(ComPath). In fact in terms of implementation the difference between ComPath+ and ComPath is steps 2,3,4 that was explained already.

V. Experimental Results

We evaluated the efficiency of the proposed ComPath+ algorithm based on two well-known datasets (NetHEPT and Epinion) and is compared with several algorithms. These algorithms are implemented by MATLABR2015b on a PC with 3.30 GHz Intel Core i3 CPU and 2G memory running Windows10 Operating system. Datasets are:

- NetHEPT: a well-known dataset with 15233 nodes and 58891 edges that has been used in many studies [3,13,14]. High Energy Physics (Theory) part of arXiv3 created NetHEPT which is the contribution network of paper authors. In this network A(u, v) is the number of papers that u and v have published together. Users' actions for publishing papers are collected in NetHEPT.
- Epinion [17]: This network has 75879 nodes and 508837 edges, nodes are members of a site and an edge from v to u, means v trusts u (i.e. u influences v).

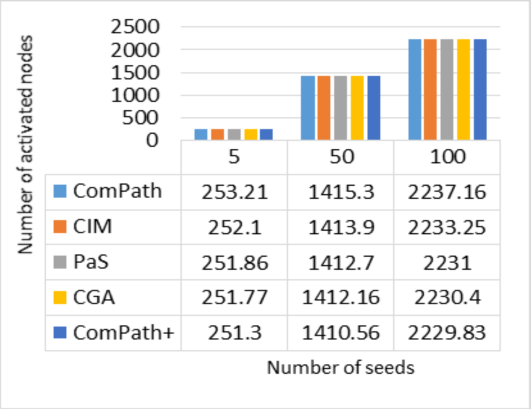
Rahimkhani et al., 2015 [2] compared their algorithm (ComPath) with some well-known algorithms. These algorithms are Maximum Degree, Page Rank, LDAG, MC-CELF and CELF++. They showed their algorithm is better than other algorithm both in the execution time and selecting better initial seeds. Also their experimental results show ComPath is much faster than the other algorithms.

In this paper we compare our algorithm (ComPath+) with new algorithms including ComPath [2], CIM [1], PaS [15] and CGA [16]. Figs. 4 and 5, compare ComPath+ with other algorithms based on the two datasets for different size of seeds (5, 50, 100 seeds). It can be observed that Number of activated nodes in algorithms is similar. The difference between the values is negligible.

Figs. 6 and 7, compare running time of the algorithms. Running time of ComPath+ for different size of seeds set are less than running time of other algorithms for both datasets.

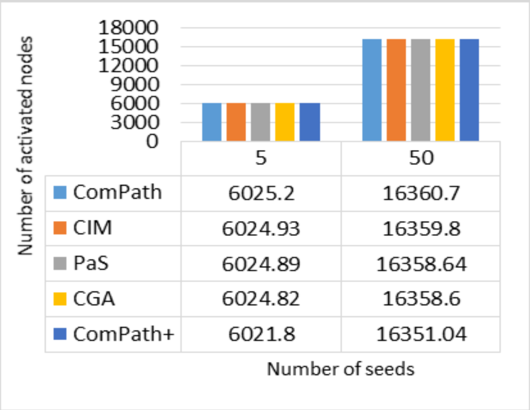
Figures 8 and 9 are performance comparison of algorithms. These Figures show that the proposed method performed better than others.

The experiments show that ComPath+ not only actives similar count nodes but also it is much faster than other algorithms. ComPath+ is comparable with other algorithms in terms of activated nodes but other algorithms are not fast to select seeds where high quality seeds is needed. Also if big number of seeds are requested, ComPath+ is more efficient and much faster than others. Then performance of ComPath+ is better than other algorithms.



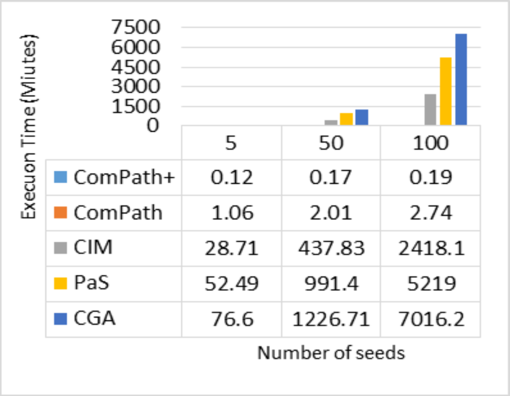
	5	50	100
🟦 ComPath	253.21	1415.3	2237.16
🟧 CIM	252.1	1413.9	2233.25
◼ PaS	251.86	1412.7	2231
🟨 CGA	251.77	1412.16	2230.4
🟦 ComPath+	251.3	1410.56	2229.83

Figure 4. Number of activated nodes on NetHEPT for different seeds



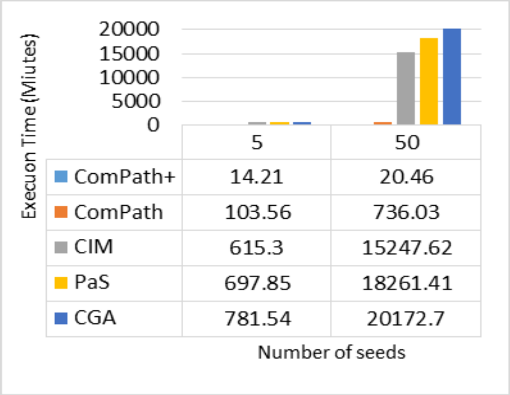
	5	50
🟦 ComPath	6025.2	16360.7
🟧 CIM	6024.93	16359.8
◼ PaS	6024.89	16358.64
🟨 CGA	6024.82	16358.6
🟦 ComPath+	6021.8	16351.04

Figure 5. Number of activated nodes on Epinion for different seeds



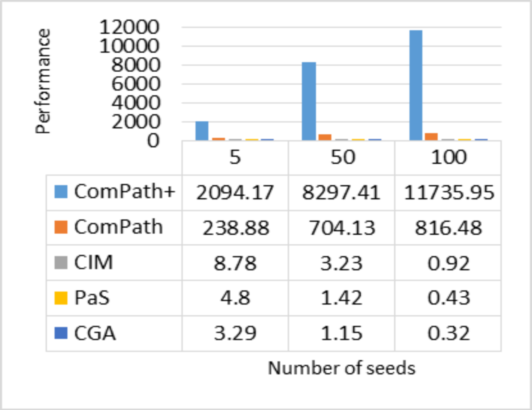
	5	50	100
🟦 ComPath+	0.12	0.17	0.19
🟧 ComPath	1.06	2.01	2.74
◼ CIM	28.71	437.83	2418.1
🟨 PaS	52.49	991.4	5219
🟦 CGA	76.6	1226.71	7016.2

Figure 6. Running time on the NetHEPT



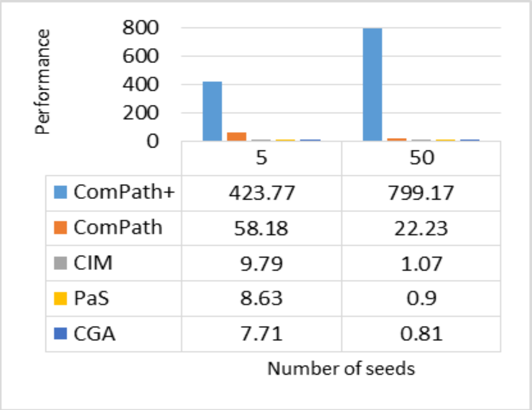
	5	50
🟦 ComPath+	14.21	20.46
🟧 ComPath	103.56	736.03
◼ CIM	615.3	15247.62
🟨 PaS	697.85	18261.41
🟦 CGA	781.54	20172.7

Figure 7. Running time on the Epinion



	5	50	100
🟦 ComPath+	2094.17	8297.41	11735.95
🟧 ComPath	238.88	704.13	816.48
⬜ CIM	8.78	3.23	0.92
🟨 PaS	4.8	1.42	0.43
🟦 CGA	3.29	1.15	0.32

Figure 8. Performance on the NetHEPT



	5	50
🟦 ComPath+	423.77	799.17
🟧 ComPath	58.18	22.23
⬜ CIM	9.79	1.07
🟨 PaS	8.63	0.9
🟦 CGA	7.71	0.81

Figure 9. Performance on the Epinion

Conclusion

In this paper, an efficient and fast Community-based influence maximization algorithm called ComPath+ is proposed for influence maximization based which enhances ComPath algorithm investigate the small number of nodes and preserve quality of seeds and then it can influence on more nodes in less time.

ComPath+ discovers the communities by SLPA in the input graph and then creates new network that is a weighted directed graph. This new network eliminates finding final seeds problem and surveys a small set of nodes for reducing the running time. Important communities are selected from new network based on power of connections and then important nodes are selected among communities. Final seeds are chosen from important nodes. This method is applicable for different applications such as targeted marketing.

Our experimental results show that the proposed algorithm has properly overall performance and affords an excellent stability between running time and efficiency. Also if a large number of initial seeds are needed, proposed algorithm has better performance than others.

References

[1] Y-C. Chen, W-Y. Zhu, W-C. Peng, W-C. Lee and S-Y. Lee, “CIM: Community-based influence maximization in social networks,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 5, no. 2, 2014.

[2] K. Rahimkhani, A. Aleahmad, M. Rahgozar and A. Moeini, “A fast algorithm for finding most influential people based on the linear threshold model,” Expert Syst. Appl., vol. 42, no. 3, pp. 1353-1361, 2015.

[3] D. Kempe, J. Kleinberg and E. Tardos, “Maximizing the spread of influence through a social network,” in the ninth ACM SIGKDD international conference on knowledge discovery and data mining KDD 03, New York, 2003.

[4] S. Wasserman and K. Faust, “Social network analysis: Methods and applications,” Cambridge University Press, 1994.

[5] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” Nature 435, pp. 814–818, 2005.

[6] U. N. Raghavan, R. Albert and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Physics and Society, vol. 76, no. 3, 2007.

[7] T.S. Evans, R. Lambiotte, “Line Graphs, Link Partitions and Overlapping Communities,” eprint arXiv:0903.2181, 2009.

[8] A. Lancichinetti, S. Fortunato and J. Kertesz, “Detecting the overlapping and hierarchical community structure in complex networks,” New Journal of Physics, vol. 11, no. 3, 2009.

[9] J. Xie, B. K. Szymanski and X. Liu, “SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process,” in the IEEE 11th international conference on data mining workshops, 2011.

[10] P. Domingos and M. Richardson, “Mining the network value of customers,” in the seventh international conference on knowledge discovery and data mining KDD 01, New York, 2001.

[11] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” Computer Networks and ISDN Systems, vol. 30, no. 7, pp. 107–117, 1998.

[12] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen and N. Glance, “Cost-effective outbreak detection in networks,” in the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, California, USA, 2007.

[13] W. Chen, Y. Yuan and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in the 2010 IEEE international conference on data mining, Washington, DC, USA, 2010.

[14] A. Goyal, W. Lu and L. V. S. Lakshmanan, “CELF++: Optimizing the greedy algorithm for influence maximization in social networks,” in the 20th International conference companion on world wide web, Hyderabad, India, 2011.

[15] J. Ok, Y. Jin, J. Shin and Y. Yi, “On maximizing diffusion speed in social networks: impact of random seeding and clustering,” in the 2014 ACM international conference on Measurement and modeling of computer systems (SIGMETRICS '14), ACM, New York, NY, USA, pp. 301-313, 2014.

[16] G. Song, X. Zhou, Y. Wang and K. Xie, “Influence maximization on large-scale mobile social network: a divide-and-conquer method,” IEEE Trans. Parallel Distrib. Syst. 26, pp. 1379–1392, 2015.

[17] M. Richardson, R. Agrawal and P. Domingos, “Trust management for the semantic web,” in The 2nd international semantic web conference (ISWC2003), Berlin Heidelberg, 2003.