Community-based k-shell decomposition for identifying influential spreaders

Peng Gang Sun[a], Qiguang Miao[a], Steffen Staab[b,c]

[a] *School of Computer Science and Technology, Xidian University, Xi'an, 710071, China*
[b] *Institute for Web Science and Technologies (WeST), University of Koblenz-Landau, Koblenz, 56070, Germany*
[c] *Web and Internet Science (WAIS) Research Group, University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom*

Abstract

How to identify the most influential nodes in a network for the maximization of influence spread is a great challenge. Known methods like k-shell decomposition determine core nodes who individually might be the most influential spreaders for the spreading originating in a single origin. However, these techniques are not suitable for determining multiple origins that together lead to the most effective spreading. The reason is that core nodes are often found to be located closely to each other, which results in large overlapping regions rather than spreading far across the network. In this paper, we propose a new algorithm, called *community-based k-shell decomposition*, by which a network can be viewed as multiple hierarchically ordered structures each branching off from the innermost shell to the periphery shell. To alleviate the overlap problem, our algorithm pursues a greedy strategy that preferably selects core nodes from different communities in the network, thus maximizing the joint influence of multiple origins. We systematically evaluate our algorithm against competing algorithms on multiple networks with varying network characteristics, and find that our algorithm outperforms other algorithms on networks that exhibit community structures, and the stronger communities, the better performance.

Keywords: influential spreader, community-based k-shell decomposition, linear threshold model

1. Introduction

The phenomenon of spreading in complex networks has attracted more attention of researchers from a variety of fields [1, 2] because it can be used to describe many important processes including the spreading of epidemics [3], technique innovations [4], product promotion [5], and behavior adoption [6], which may help us to understand the mechanisms underlying complex phenomena and guide human productions and livings [3].

Real-world networks exhibit a rich set of features that determine which nodes are located in the most vital positions and hence more influential and more capable of triggering information diffusion at a large scale [3]. How to identify the most influential nodes in a network is very important [7, 8], as it lets us proactively choose an influencing strategy, e.g., in viral marketing, or it lets us retroactively analyse the most likely origins, e.g., in analyses of an outbreak of an epidemic [3] or a rumour [9].

Finding the most influential spreaders can be distinguished into (1) finding a single origin to achieve *individual influence maximization*, or (2) finding multiple origins to achieve *collective influence maximization* [3]. For identifying a single origin, a simple way is first to rank all the nodes by degree (k) [7], Betweenness [10, 11], Eigencentrality [10, 11], k-shell decomposition (ks) [12-14], PageRank [15] or SpringRank [16], and then choose the nodes in the top rank. For identifying multiple origins, we usually need to find a seed set with a given size so that the influence aggregated by activating the nodes within the set is maximized [16]. Kempe et al. [17] proposed two models for the influence maximization problem, i.e., the independent cascade (IC) model and the linear threshold (LT) model. The IC model assumes independence of spreading activation: each active node may probabilistically activate its inactive neighbors independently of the influence of other active nodes [17]. The LT model considers joint activation of neighbors: an inactive node is activated if the sum of weights of its active neighbors exceeds a threshold [17].

In the past several years, some traditional greedy algorithms [18], machine learning algorithms [19] and heuristic algorithms were developed such as CELF++ [20], the degree discount heuristic algorithm [21] and the maximum influence arborescence (MIA) heuristic algorithm [22]. In particular, Chen et al. [23] proposed the first scalable heuristic algorithm under the LT model, called LDAG algorithm. The algorithm first constructs the local directed acyclic graphs (DAGs), and then selects seeds with the maximum increment for influence spread by a greedy approach [22]. For each node, the algorithm updates the incremental influence spread with a fast strategy, which always makes the LDAG algorithm perform well among the best algorithms [23]. Morone and Makse [24] used the optimal percolation theory to identify the optimal set of influential nodes for influence maximization, and determined the minimal set of nodes in a network that could keep the global connectivity of the network, i.e., if these nodes are activated, information percolates through the whole network, and if removed, the whole network is broken down into many disconnected pieces. Morone and Makse [24] further developed a scalable algorithm, Collective Influence (CI), to solve the hard optimization problem. However, Hu et al. [25] showed that the actual influence of each node can be quantified from its local information, and demonstrated that the best nodes for breaking down the networks are not necessarily the best spreaders, which finally leads to the lower performance of the CI algorithm. Erkol et al. [26] performed a systematic test of the performance of the algorithms for identifying influential spreaders.

Actually, searching for the optimal seed set with a given size is more complicated. A simple way to determine the set is the same as that of individual spreaders, i.e., rank all the nodes firstly, and then choose the nodes in the top with a given size as the best seeds. However, Kitsak et al. [13] showed that multiple origins determined by the highest-k or the highest-ks are not efficient due to large overlapping regions. Communities as an important pattern characterized by more links between nodes in same communities than those of different communities are widely observed in real-world networks [27], as they are formed by the co-constitution of structures and communication. Nodes in the same community communicate more probably, and communication leads to the establishment of links [28]. Thus, communities greatly influence the spreading in networks, which motivates us to investigate the selection of nodes from different communities as origins [29]. In addition, the hierarchical structure of a network determined by k-shell decomposition is also of great importance for influence spread [13]. By considering the community structure as well as the hierarchical structure, we propose a new algorithm, called community-based k-shell decomposition (CKS), by which a network can be viewed as multiple hierarchically ordered structures each branching off from the innermost shell to the periphery shell. To alleviate the overlap problem, our algorithm pursues a greedy strategy that preferably selects core nodes from different communities in the network, thus maximizing the joint influence of multiple origins. We systematically evaluate our algorithm against competing algorithms on artificial networks with varying network characteristics as well as real-world networks.

The rest of the paper is organized as follows. In Section 2, we describe related works for identifying influential spreaders. In Section 3, we discuss k-shell decomposition. In Section 4, we introduce our algorithm, called community-based k-shell decomposition. In Section 5, we evaluate and compare our algorithm with other algorithms on artificial networks as well as real-world networks. The conclusion is provided in Section 6.

2. Related works

Many works demonstrated that communities in a network are of great importance for influence spread [29]. Cao et al. [29] proposed the first community-based algorithm for influence maximization, called OASNET (Optimal Allocation in a Social NETwork). The algorithm first partitions a network into m communities, and then selects s nodes from each community by traditional greedy algorithm. Lastly, the algorithm uses dynamic programming to determine s nodes from $m \times s$ candidates as the best spreaders. Zhang et al. [30] used k-medoid algorithm to identify influential nodes on the networks with communities by constructing an information transfer matrix. Chen et al. [31] developed a community-based algorithm for influence maximization, called CIM. The algorithm first uses hierarchical clustering to partition a network into communities. Then, it determines nodes from significant communities based on position score and hub purity. Lastly, to determine the best seeds, it swaps a new candidate with a seed node if it potentially increases the influence spread. Shang et al. [32] explored a community-based framework for influence maximization, called CoFIM. The algorithm first expands seeds among different communities, and then studies influence spread within communities. Finally, the hill-climbing greedy algorithm is adopted to determine the best seeds.

3. *k*-shell decomposition

The k-shell decomposition [12] is used to analyze the hierarchical structure of a network by assigning each node with a k-shell index.

Definition 1 (k-shell index)

Given a graph, $G(V, E)$, where $V = \{1, 2, ..., n\}$ is the node set, and $E = \{(i, j)|i, j \in V\}$ is the edge set. $G' = G(V - V')$, where G' is an induced subgraph from G by $V - V'$, and k'_i denotes the degree of node i in G'; $\mathbf{KS} = (ks_1, ks_2, ..., ks_n)$, where $ks_i = l$ indicates that node i are located in the l-shell of G;

The k-shell index ks_i for node i can be defined recursively by:

1: $l = 0$, $V' = \oslash$;

2: $G' = G(V - V')$;

3: For all $i \in V - V'$; If $k'_i = l$; Then $ks_i = l$, $V' = V' \cup \{i\}$; GOTO (2); Else GOTO (4);

4: If $V' == V$, Then stop ;

5: $l = l + 1$;

6: GOTO (2);

In Fig. 1, we illustrate the k-shell decomposition in a toy network. From the figure, we can see that the nodes with the largest k-shell index tend to be located in the innermost shell, called core nodes, and the nodes with the lowest k-shell index are located in the periphery shell, called periphery nodes.

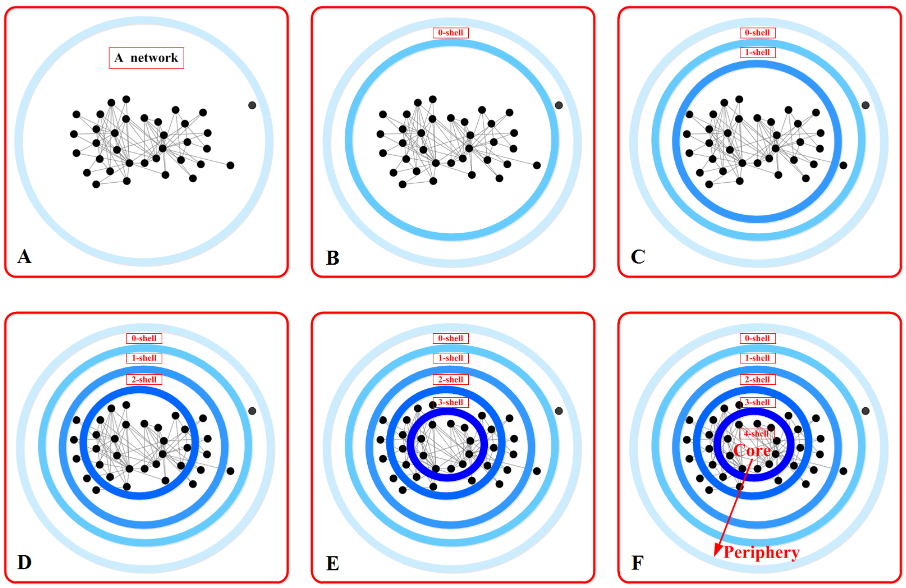

Figure 1: Illustration of the k-shell decomposition in a toy network. (A) A toy network used in this paper. (B)-(F) correspond to the 0-shell, 1-shell, 2-shell, 3-shell, 4-shell respectively. The network can be viewed as a hierarchically ordered structure consisting of many sub-structures from the innermost shell to the periphery shell.

In a recent work, Kitsak et al. [13] demonstrated that high-degree nodes located in the periphery of a network tend to be less efficient than those located in the core of the network. Considering the spreading dynamics, Liu et al. [14] further tried to improve the accuracy of the k-shell method by removing redundant links. Actually, in some cases, we need to find a small set of nodes that are the most influential, as multiple origins can accelerate the spreading to the largest scale. However, Kitsak et al. [13] showed that high-degree nodes as well as core nodes tend to be less efficient in the spreading process due to the large overlap of infected areas. In Fig. 2, we illustrate the overlap problem in a toy network, and Fig. 2 (A) corresponds to the spreading originating from two nodes with the highest-ks, and Fig. 2 (B) corresponds to the spreading origi-

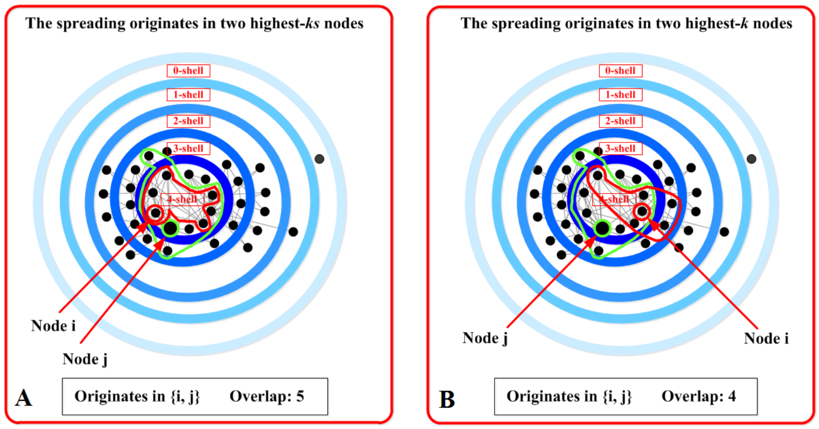

Figure 2: Illustration of the overlap problem. (A) The spreading originates from two nodes with the highest-ks ($ks_i = ks_j = 4$). (B) The spreading originates from two nodes with the highest-k ($k_i = 17, k_j = 16$). The propagation range of a node can be determined by the activated nodes with high frequency in the realizations of the spreading process.

nating from two nodes with the highest-k. The infected areas are highlighted by color curves, and the propagation range of a node can be determined by the activated nodes with high frequency in the realizations of the spreading process.

4. Community-based k-shell decomposition

Here, we propose a new algorithm, called community-based k-shell decomposition (CKS), by which a network can be viewed as multiple hierarchically ordered structures each from the innermost shell to the periphery shell. To alleviate the overlap problem, we preferentially choose the core nodes from different communities as origins for influence maximization.

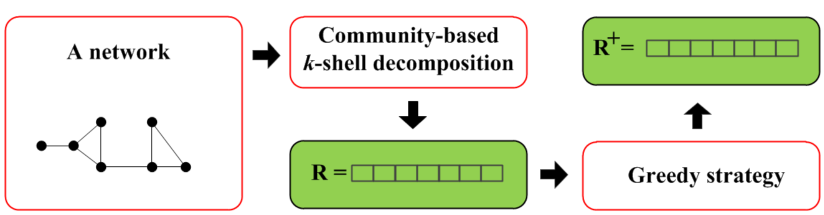

Figure 3: Illustration of our framework for finding the most influential spreaders.

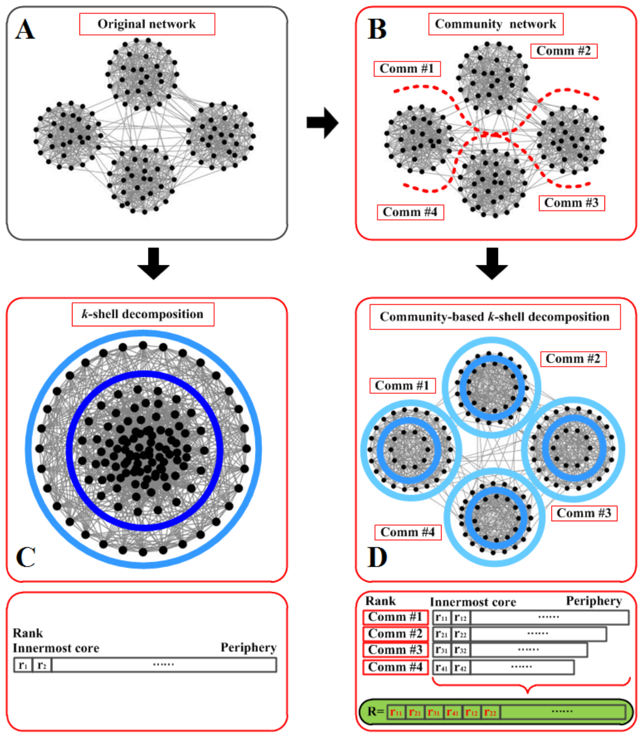

Figure 4: Illustration of the CKS in a toy network. (A) A toy network used in this paper. (B) we partition the network into four communities, i.e., the community network. (C) and (D) The hierarchical structures of the network from the perspectives of the k-shell decomposition and the CKS respectively. In the bottom, we show the ways to rank all the nodes for the k-shell decomposition and the CKS respectively.

Our algorithm is illustrated in Fig. 3. For finding a seed set, $\mathbf{B}$ with s seeds in a network, the CKS first determines the communities in a network by the modularity optimization method proposed by Newman and Girvan [27]. Blondel et al. [34] developed a heuristic method based on the modularity optimization, which is very fast for unfolding communities in large size networks. Then, the CKS sorts nodes in each community by k-shell index, and alternately choose core nodes from different communities to construct a ranking, $\mathbf{R}$, which is illustrated in the bottom of Fig. 4. Lastly, the CKS determines the seed set, $\mathbf{B}$ that

Algorithm 1 Pseudocode for the CKS.

Input: A network, $G(V, E)$, **A vector**, $\mathbf{X} = (x_1, x_2, ..., x_n)$, where $x_i = b$ indicates that node i belongs to community b;

Output: A vector, $\mathbf{Y} = (y_1, y_2, ..., y_n)$, where $y_i = l$ indicates that node i belongs to the l-shell; **A List, R** indicates the ranking for all the nodes;

Initialization
For all $i \in V$;
    $x_i = i$;
$\mathbf{X} = (1, 2, ..., n)$;
Determine $\mathbf{X} = (x_1, x_2, ..., x_n)$ by maximizing modularity;
Determine a partition, P by **X**;
$\mathbf{P} = \{G^1(V^1, E^1), G^2(V^2, E^2), ...\}$
For all $G^t(V^t, E^t) \in \mathbf{P}$;
    For all $i \in V^t$;
        Determine the location of node i, l by the k-shell decomposition;
        $y_i = l$;
    End for
End for
$\mathbf{Y} = (y_1, y_2, ..., y_n)$
For all $G^t(V^t, E^t) \in \mathbf{P}$;
    Rank all $i \in V^t$ by **Y**;
    $D^t = V^t$;
    For $i, j, ... \in D^t$;
        If $y_i = y_j = ...$;
        Then re-rank $i, j, ...$ in D^t by degree;
        $\mathbf{D} \leftarrow D^t$;
    End for
End for
$\mathbf{D} = \{D^1, D^2, ...\}$;
Rank all $D^t \in \mathbf{D}$ by $|D^t|$;
For $s = 0, 1, 2, ...$;
    For $t = 0, 1, 2, ...$;
        $D^t = \mathbf{D}[t]$;
        $\mathbf{R} \leftarrow D^t[s]$;
    End for
End for
Output Y, R

consists of the top s nodes in $\mathbf{R}$. The pseudocodes for the CKS can be seen in **Algorithm 1**. Zhao et al. [33] tried to identify multiple influential nodes by coloring a network. This algorithm colors each node in a network by one kind of color, and the nodes with the same color constitutes an independent set. It chooses the nodes with highest centrality index such as degree in an independent set as seeds. In contrast to our algorithm, communities consist of nodes that are densely connected, while independent sets consist of nodes that are often not directly linked.

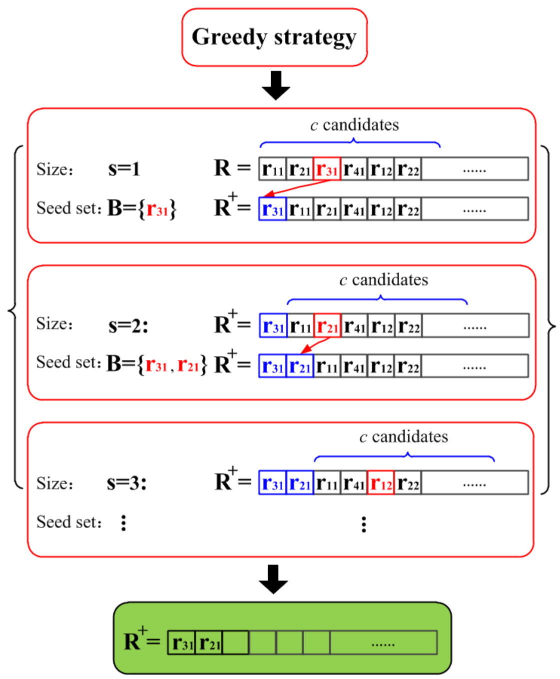

Figure 5: Illustration of the greedy strategy for CKS$^+$.

For the CKS$^+$, a greedy strategy is adopted to determine the best seeds (see Fig. 5). In the CKS$^+$, we consider $\mathbf{R}$ as an initial ranking, and a new ranking, $\mathbf{R}^+$ can be determined by a fast greedy strategy, which selects the best node from c candidates (c is the number of candidates) in $\mathbf{R}$. For example, in Fig. 5, (1) we first find the best spreader, r_{31} with the maximal individual spreading ability, and $\mathbf{B} = \{r_{31}\}$. $\mathbf{R}^+$ can be determined by updating r_{31} in

$\mathbf{R}$; (2) then fix r_{31}, and find the second best spreader r_{21} with the maximal collective spreading ability in $\mathbf{B} = \{r_{31}, r_{21}\}$, and update $\mathbf{R}^+$; (3) repeat this process until s influential spreaders are determined, which correspond to the top s nodes in $\mathbf{R}^+$. The pseudocodes for the greedy strategy of the CKS$^+$ can be seen in **Algorithm 2**.

Algorithm 2 Pseudocode for the greedy strategy of CKS$^+$.

Input: **A List,** $\mathbf{R} = (R[1], R[2], ..., R[n])$, s and c indicate the seed set size and the number of candidates respectively;

Output: **A List**, $\mathbf{R}^+ = (R^+[1], R^+[2], ..., R^+[n])$, $\mathbf{B}$ is the seed set;

```
Initialization
B = ∅;
For i = 1, 2, ..., s;
    For all j = {i, i + 1, ..., i − 1 + c};
        B + {R[j]} = Max Influence Spread (B + {R[j]});
        B ← R[j];
        Determine R+ by inserting R[j] in i^th of R;
    End for
End for
Output R+, B;
```

5. Results and discussions

5.1. Experimental Set-up

In the experiments, we choose the most influential nodes identified by our algorithm as seeds to trigger a spreading process under the LT model. In the LT model, we use the uniform method mentioned by Chen et al. [23] to generate the weights for all the links in the networks, and for the threshold parameter, θ, we choose θ =1/320 [18, 24]. The number of nodes activated in the process by seeds on average over 2000 simulations is used to evaluate the performance for influence spread.

Here, we compare our algorithm with several other algorithms that determine the most influential spreaders by the highest-k, the highest-ks, the highest-eigencentrality, PageRank [15], SpringRank [16] and the LDAG algorithm [23] on artificial networks as well as real-world networks.

5.2. Experimental results

5.2.1. Test on artificial networks with various network characteristics

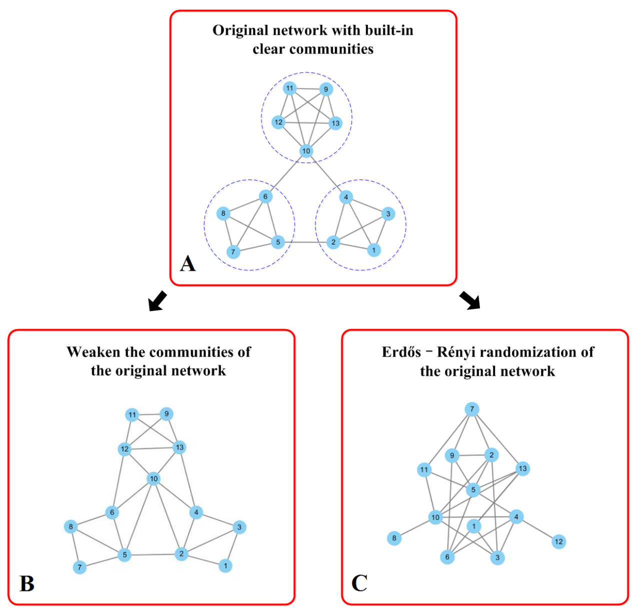

Figure 6: Illustration of artificial networks with different network structures. (A) Original networks with built-in clear communities. (B) The weakened versions of the original networks. (C) The reconfigured networks by the ER randomization.

In this subsection, we test our algorithm on multiple artificial networks with various network characteristics (see Fig. 6): (1) we generate artificial networks with built-in clear communities by the Lancichinetti, Fortunato and Radicchi (LFR) benchmark [35]. The networks contain 1.0×10^3 nodes with average degree, $\langle k \rangle = 16$, maximum degree, $maxk = 50$, the maximum community sizes, $maxc = 50$, and the minimum community sizes, $minc = 10$. The mixing parameter, $mu = 0.1$, where $mu \in [0, 1]$, i.e., the communities in the networks tend to be clear as mu decreases [33] (see Fig. 6 (A)). (2) we weaken the communities in the networks, and here $mu = 0.4$ (see Fig. 6 (B)); (3) we reconfigure the networks by the Erdős - Rényi (ER) randomization [36], i.e., the nodes in the reconfigured networks are completely connected at random (see Fig. 6 (C)).

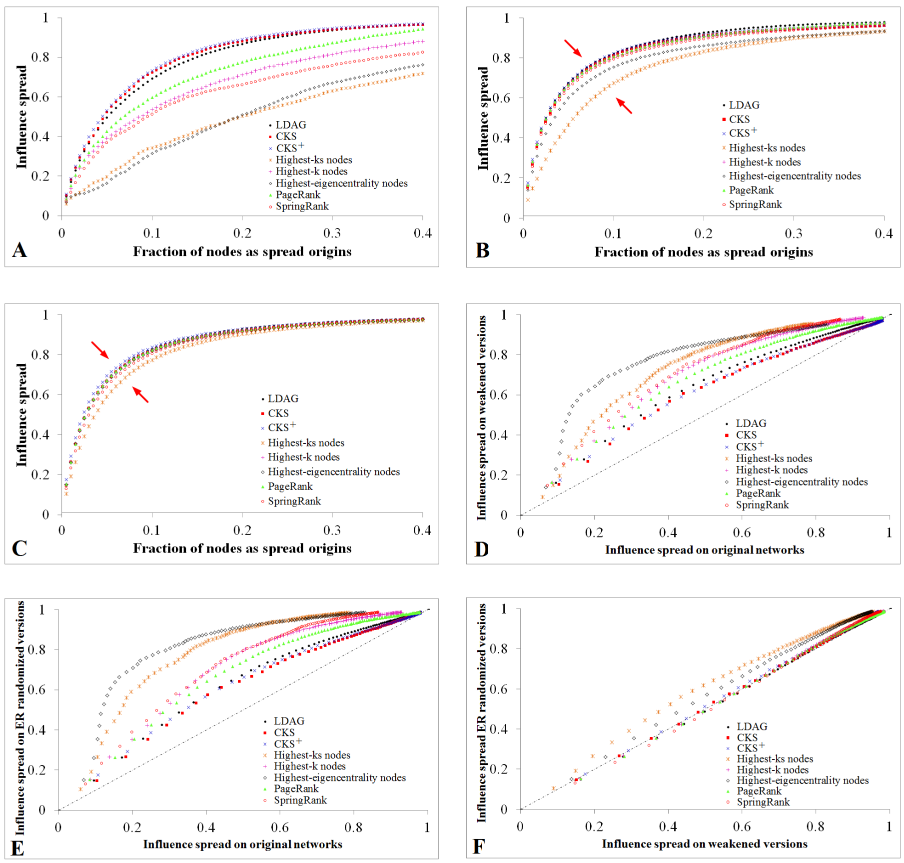

Figure 7: Influence spread on artificial networks with different network structures. (A) Original networks with built-in clear communities. (B) The weakened versions of the original networks. (C) The reconfigured networks by the ER randomization. (D) Comparison between the original networks and the weakened versions. (E) Comparison between the original networks and the ER randomized versions. (F) Comparison between the weakened versions and the ER randomized versions. For the CKS^{+}, we only consider $c = 50$ candidates.

For the spreading originating from multiple origins, we study the number of nodes activated in the spreading process as a function of the number of origins (i.e., the size of seed set). The results on artificial networks with various network characteristics are shown in Fig. 7. Fig. 7 (A) corresponds to the results on the artificial networks with built-in clear communities, and we can observe that both the CKS and the CKS^{+} outperform other algorithms, i.e., the cores nodes selected from different communities tend to be the best spreaders for the maximal collective influence spread. Fig. 7 (B)-(C) correspond to the results on the weakened versions and the ER randomized versions respectively. In Fig. 7 (B), we can see that our algorithm tends to achieve better performance on the networks with weak communities when the size of seed set is small. We can also see that the curves of influence spread for all the algorithms start to converge when we weaken the communities (see Fig. 7 (B)), and this convergence is more clearly observed on the ER randomization versions (see Fig. 7 (C)). In Fig. 7 (D), we compare the results between the original networks and the weakened versions, and we can see that it is more efficient for influence spread if we weaken the internal links and strength the external links of communities in the original networks. Similarly, the reconfigured networks by the ER randomization are also more efficient for influence spread (Fig. 7 (E) and (F)).

5.2.2. Test on real-world networks with well-known communities

In this subsection, we try to test our algorithm on four real-world networks with well-known communities including, (1) Zachary karate club network [37]. This network (34 nodes and 78 links) was constructed by Wayne Zachary after he studied the members of a university karate club. In the network, each node represents a member, and each edge represents a tie between two members in the club. During the study, he observed a split in the club, i.e., the club's instructor took away a half of the members in the club and built a new club due to a disagreement between the club's instructor and administrator. Therefore, this network contains two well-known communities, which are centred with the club's instructor and administrator respectively. (2) Co-appearance network [38]. In this network (75 nodes and 254 links), a node represents a character

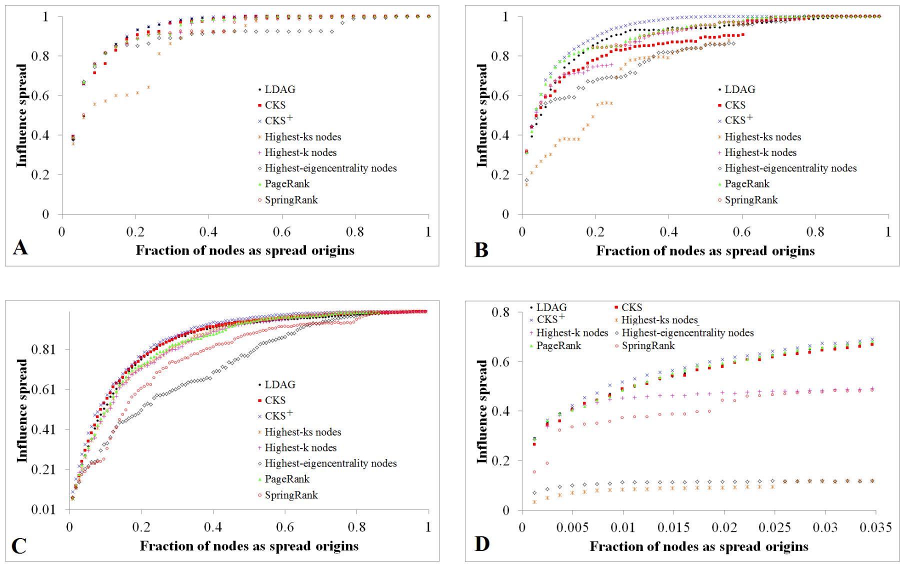

Figure 8: Influence spread on real-world networks with well-known communities. (A) Karate club network. (B) Co-appearance network. (C) US college football netowrk. (D) Facebook network. For the CKS$^+$, we only consider $c = 50$ candidates.

in Victor Hugo's novel (Les Miserables), and a link connecting two nodes if the two characters appeared in the same chapter of the book. Communities in this network often reveal the relationships of characters in the novel such as kinships, street gangs and friendships. (3) US college football network [27]. This network contains 115 nodes and 616 links. One node represents a team, and one link represents a game between two teams. The 115 teams can be roughly divided into 12 conferences, and teams within the same conference meet more frequently in a game than those of different conferences. (4) Facebook network [39]. This network contains 4039 nodes and 88234 links, and is collected from http://snap.stanford.edu/. The network consists of plenty of social circles or friends lists, and communities always correspond to social circles with same features such as political affiliations, hometown and education.

The karate network, the co-appearance network, the US college football network and the facebook network are often used as a benchmark to evaluate the performance of community detection methods. Fig. 8 (A)-(D) correspond to the influence spread on the four networks respectively, and we can see that the CKS$^+$ obtains the best performance on these networks. In addition, the CKS outperforms other algorithms on the co-appearance network.

5.2.3. *Test on large real-world networks*

In this subsection, we especially choose two large real-world networks to test the performance of our algorithm including, (1) Cond-mat collaboration network. This network is relevant to the collaborations between scientists who post preprints on the Condensed Matter E-Print Archive from 1995 to 2005. In the network, a node represents an author, and two authors are linked if they have co-authored one paper at least. The network contains 39577 nodes and 175693 links [40]. (2) DBLP, a co-authorship network where two authors are connected if they publish at least one paper together in computer science. This network contains 317080 nodes and 1049866 links [41].

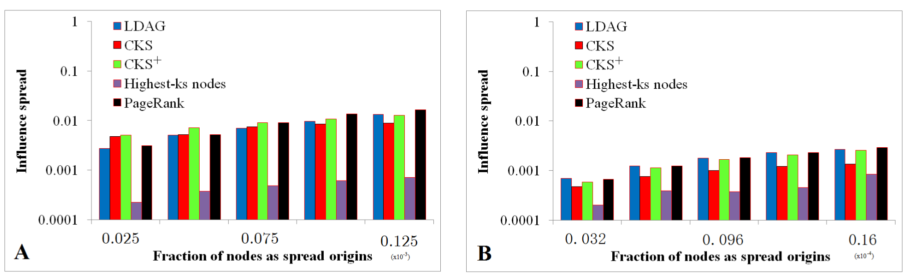

Figure 9: Influence spread on real-world networks. (A) Cond-mat collaboration network. (B) DBLP. For the CKS$^+$, we only consider $c = 50$ candidates.

Here, we only compare our algorithm with the LDAG algorithm, PageRank and the k-shell decomposition. The results in Fig. 9 show that our algorithm tends to achieve better performance when the size of seed set is small size, which is similar to the finding on the artificial networks with weak communities in Fig. 7 (B).

6. Conclusions

For the influence maximization problem, identifying influential nodes is a hot topic in complex networks. By taking the topological structure of a network into account, community-based algorithms play an important role in identifying influential nodes. Actually, the hierarchical structure of a network determined by k-shell decomposition is also of great importance for identifying influential nodes. However, for the spreading originating from multiple origins, the spreading efficiency of core nodes is very low due to the overlap problem.

In this paper, we propose a new algorithm, called community-based k-shell decomposition. Our algorithm first partitions a network into communities, and the communities are ranked by their sizes in descending order. Then, we sort nodes in each community by k-shell index, and alternately choose core nodes from the communities to construct a ranking for all the nodes. For finding a seed set with s nodes, the top s nodes in the ranking can be considered as initial candidates, and a greedy strategy is adopted to determine the best seeds. The results indicate that the influential nodes identified by our algorithm as origins are more efficient for influence spread on artificial networks and real-world networks.

For the weaknesses of our algorithm, a community is more likely to contain nodes with the same ks index. How to rank these nodes in a community is a challenging work. On the one hand, we can improve the accuracy of k-shell decomposition. On the other hand, we can rank nodes by considering other centralities if these nodes have the same ks index. Therefore, in the future work, we will try to improve the performance of our algorithm and test it on a large number of real-world networks.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61872432), the Fundamental Research Funds for the Central Universities (Grant No. JB210303).

References

[1] S. H. Strogatz, Exploring complex networks, Nature, 410, 268–276, 2001.

[2] R. Albert, A.-L. Barabsi, Statistical mechanics of complex networks, Rev. Mod. Phys., 74, 47–97, 2002.

[3] S. Pei, H. A. Makse, Spreading dynamics in complex networks, Journal of Statistical Mechanics: Theory and Experiment, 12, P12002, 2013.

[4] E. M. Rogers, Diffusion of innovation 4th edn, (Free Press) 1995.

[5] P. Domingos, M. Richardson, Mining the network value of customers, Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 57–66, San Francisco, California, August 26–29, 2001.

[6] D. Centola, The spread of behavior in an online social network experiment, Science, 329(5996), 1194–1197, 2010.

[7] F. Radicchi, C. Castellano, Leveraging percolation theory to single out influential spreaders in networks, Phys. Rev. E 93, 062314, 2016.

[8] L. Lu, D. Chen, X.L. Ren, Q.M. Zhang, Y.C. Zhang, T. Zhou, Vital nodes identification in complex networks. Phys. Rep., 650, 1–63, 2016.

[9] Y. Luo, J. Ma, C. K. Yeob, BCMM: A novel post-based augmentation representation for early rumour detection on social media, Pattern Recognition, 113, 107818, 2021.

[10] L. C. Freeman, Centrality in social networks: Conceptual clarification, Social Networks, 1, 215–239, 1979.

[11] N. E. Friedkin, Theoretical foundations for centrality measures, Am. J. Sociology, 96, 1478–1504, 1991.

[12] S. Carmi, S. Havlin, S. Kirkpatrick, Y. Shavitt, E. Shir, A model of Internet topology using k-shell decomposition, Proc. Natl. Acad. Sci., 104, 11150–11154, 2007.

[13] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E.Stanley, H. A. Makse, Identification of influential spreaders in complex networks, Nature Physics, 6, 888–893, 2010.

[14] Y. Liu, M. Tang, T. Zhou and Y. Do, Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics, Sci. Rep., 5, 13172, 2015.

[15] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, Computer Networks, 30(1) 107–117, 1998.

[16] C. D. Bacco, D. B. Larremore, C. Moore, A Physical Model for Efficient Ranking in Networks, Science Advances, 4(7), 8260, 2018.

[17] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the spread of influence through a social network, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 137–146, Washington, D.C. August 24-27, 2003.

[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Van-Briesen, and N. S. Glance, Cost-effective outbreak detection in networks, Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 420–429, San Jose, California, USA, August 12-15, 2007.

[19] J. Tang, J. Sun, C. Wang, and Z. Yang, Social influence analysis in large-scale networks, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 807–816, Paris, France, June 28 - July 01, 2009.

[20] Goyal, W. Lu, L.V. Lakshmanan, Celf++: optimizing the greedy algorithm for influence maximization in social networks, Proceedings of the 20th inter-

national conference companion on World Wide Web, pp. 47–48, Hyderabad, India, March 28-April 01, 2011.

[21] W. Chen, Y. Wang, and S. Yang, Efficient influence maximization in social networks, Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 199–208, Paris, France, June 28-July 01, 2009.

[22] W. Chen, C. Wang, and Y. Wang, Scalable influence maximization for prevalent viral marketing in large scale social networks, Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1029–1038, Washington, DC, USA, July 25-28, 2010.

[23] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, Proceedings of the 2010 IEEE International Conference on Data Mining, pp. 88–97, Sydney, NSW, Australia, December 13-17, 2010.

[24] F. Morone, H. A. Makse, Influence maximization in complex networks through optimal percolation, Nature, 524, 65–68, 2015.

[25] Y. Hu, S. Ji, Y. Jin, L. Feng, H. E. Stanley, and S. Havlin, Local structure can identify and quantify influential global spreaders in large scale social networks, Proc. Natl Acad. Sci., 115(29), 7468–7472, 2018.

[26] S. Erkol, C. Castellano, F. Radicchi, Systematic comparison between methods for the detection of influential spreaders in complex networks, Scientific Reports, 9, 15095, 2019.

[27] M. E. J. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci., 103, 8577–8582, 2006.

[28] Y. Xie, M. Gong, S. Wang, B. Yu, Community discovery in networks with deep sparse filtering, Pattern Recognition, 81, 50–59, 2018.

[29] T. Cao, X. Wu, S. Wang, X. Hu, Oasnet: an optimal allocation approach to influence maximization in modular social networks, ACM Symposium on Applied Computing, ACM, pp. 1088–1094,2010.

[30] X. Zhang, J. Zhu, Q. Wang, H. Zhao, Identifying influential nodes in complex networks with community structure, Knowledge-Based Systems ,42, 74C84, 2013.

[31] Y.-C. Chen, W.-Y. Zhu, W.-C. Peng, W.-C. Lee, S.-Y. Lee, CIM: community-based influence maximization in social networks, ACM Trans. Intell. Syst. Technol., 5 (2), 25, 2014.

[32] J. Shang, S. Zhou, X. Li, L. Liu d, H. Wu, CoFIM: A community-based framework for influence maximization on large-scale networks, Knowledge-Based Systems, 117, 88–100, 2017.

[33] X. Y. Zhao, B. Huang, M. Tang, et al. Identifying effective multiple spreaders by coloring complex networks, Europhysics Letters, 108(6): 68005, 2015.

[34] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment, 10, P10008, 2008.

[35] Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E, 78, 046110, 2008.

[36] S. Brohe, K. Faust, G. Lima-Mendez, G. Vanderstocken, J. van Helden, Network Analysis Tools: from biological networks to clusters and pathways, Nature protocols, 3(10), 1616–1629, 2008.

[37] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research, 33, 452–473, 1977.

[38] D. E. Knuth, The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA, 1993.

[39] J. McAuley and J. Leskovec, Learning to discover social circles in ego networks, Proceedings of the 25th International Conference on Neural Information Processing Systems, pp. 539-547, Lake Tahoe, Nevada, December 3–6, 2012.

[40] M. E. J. Newman, The structure of scientific collaboration networks, Proc. Natl. Acad. Sci., 98, 404–409, 2001.

[41] J. Yang and J. Leskovec, Defining and evaluating network communities based on ground-truth, Proceedings of the 12th IEEE International Conference on Data Mining, pp. 181–213, Brussels, Belgium, December 10-13, 2012.