Expert Systems with Applications 161 (2020) 113681

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Review

A systematic survey on influential spreaders identification in complex networks with a focus on K-shell based techniques

Giridhar Maji [a,*], Sharmistha Mandal [b], Soumya Sen [c]

[a] Department of Electrical Engineering, Asansol Polytechnic, Asansol, India
[b] Department of Computer Science and Technology, Kanyapur Polytechnic, Asansol, India
[c] A. K. Choudhury School of Information Technology, University of Calcutta, Kolkata, India

ARTICLE INFO

Article history:
Received 8 February 2020
Revised 30 May 2020
Accepted 18 June 2020
Available online 2 July 2020

Keywords:
K-shell decomposition
Systematic review
Node centrality
Influential spreader identification
Node ranking heuristics
Kendall's rank correlation
Influence maximization

ABSTRACT

Almost all the complex interactions between humans, animals, biological cells, neurons, or any other objects are now modeled as a graph with the nodes as the *objects of interest* and interactions as the edges. The identification of the most central or influential node in such a complex network has many practical applications in diverse domains such as *viral marketing*, infectious disease spreading, rumor spreading in a social network, virus/worm spreading in computer networks, etc. Many centrality measures using the position/location of a node and network structure have been proposed in the literature. The *node degree*, *shortest paths*(closeness), and *betweenness* are used since long with degree capturing local effect and others global effect. The k-shell considers the coreness of the nodes by dividing the network into layers or shells. Many variations of k-shell proposed in recent years, as well as many researchers, use k-shell as a building block in their heuristic technique to alleviate the problems of classical k-shell and to identify influential spreaders more elegantly. The main objective of this paper is to analyze and compare the major variations of the k-shell based methods along with representative *network topology* based hybrid techniques by considering a toy network with detailed computations. A discussion on different performance evaluation metrics and, simulation models such as the SIR epidemic model, has been undertaken with a comparative analysis between different state-of-the-art on a few standard real networks.

1. Introduction

Online social networking sites like *Facebook*, *Twitter*, etc. have attracted people of all ages from around the globe. User counts on such sites are far more than the total population of some countries. In these kinds of networks, users are connected to each other by different types of a relationship like *friendship* on Facebook, *follower-followee* on Twitter, etc. Whenever a user publishes or shares something on their profile, it appears in the notification of all his/her friend's profile. This social medium has become very important for quick marketing of some new product or concept, reaching out to a greater audience for some contemporary event, creating and spreading awareness on social injustice (Zareie et al., 2018), etc. On the other side, rumors and misinformation spread through the social networks very rapidly. An epidemic outbreak of contagion spreads much quickly with some infected super-spreaders than others. Computer virus/worm infection in a large network spreads much faster if not immunized properly.

Hence, the problem boils down to identifying those user(s) in a complex network who can be hired/inspired to promote something, and then the cascading effect (word of mouth) reaches to the furthest users in the shortest possible time. When such a complex network is modeled as a graph, users are represented by the nodes, and the relationship among them is represented by the edges. There are two different ways to assess the key nodes in such a network (Zareie & Sheikhahmadi, 2018). First, these are the nodes upon removal of which the network loses its integrity and degenerate into smaller subgraphs, and secondly, these are the nodes when infected or made aware of some information, the maximum spreading happens in minimum time.

In an information diffusion process, the problem is modeled as an optimization problem to identify the minimal set of influential nodes that would maximize the total information spread. This is commonly known as the *Influence Maximization* problem (Zhang et al., 2019). Some product companies or event organizers may have a limited budget and would like to hire only the most influential user for the marketing purpose. It is very important to have some technique to rank the nodes based on their spreading capability.

* Corresponding author.
E-mail address: Giridhar.Maji@gmail.com (G. Maji).

https://doi.org/10.1016/j.eswa.2020.113681
0957-4174/

Influential node identification in a complex network has many other important applications in real life. Designing a viral marketing plan (Sheikhahmadi & Nematbakhsh, 2017; Sela et al., 2018), optimal resource allocation (Ren et al., 2014), selecting most influential opinion leaders in social networks (Lu et al., 2011; Chen et al., 2012; Ma & Liu, 2014; Bamakan et al., 2019), formulating infectious disease outbreak immunization strategy (Keeling & Rohani, 2011; Kang & Fu, 2015), virus/worm attack propagation in affected computer networks (Li et al., 2018), preventing power network failure (Albert et al., 2004), understanding the key nodes in traffic network (Jia et al., 2014), are some of the mostly used applications.

The problem can be categorized into two types. The objective in the first category of problem is to identify a single spreader i.e. the *best spreader* or the *super spreader* in a network. Rumor spreading generally starts from a single person and belongs to the first type. Another type of problem is concerned with a set of influential nodes also, called *seed nodes*, that can maximize the total spreading in a network quickly. The viral marketing of some products normally starts with a chosen set of users, and selecting them belongs to the second type (Hu et al., 2015; Li & Du, 2017). The common approach towards the single spreader identification problem is to rank/index all nodes in a network by their spreading influence and selecting the top one. A trivial way to select seed nodeset can be choosing the top-k nodes from the same node ranking list generated by the single node identification method. One major drawback to such a method is the spreading influence overlap (Zhou et al., 2018).

The network topology plays an important role in network behavior and functioning. The position/location of a node in the network makes it important. Hence, ranking the nodes based on their capability of spreading is not a trivial task. The most simple measure of a node's influence can be the number of neighbors it has, i.e., the degree of the node. But it is observed that all nodes with the same node degree may not have the same spreading capacity, especially when high degree nodes are positioned around the periphery.

Ranking the nodes by actually simulating spreading nature and size of spread for each node in a brute force manner is not a feasible solution as it would be computation heavy and would not produce results in a reasonable time. Selecting the most influential node as an optimization problem is NP hard (Kempe et al., 2015). Researchers have started to use different heuristic measures (of importance) mainly based on *structural information* (Lü et al., 2016) of the network to rank the nodes within feasible time and moderate computational complexity.

There exist many different information spreading or diffusion models that are used in simulating the actual information propagation process. In general, these models simulate the real spreading by repeating the process thousands of times for each node, which is a very time consuming process to be used in large networks for ranking nodes. But these models are often used to rank the nodes based on their spreading influence and then evaluate different node ranking heuristics as a benchmark. There are mainly three types of models that prevail in literature, namely, *linear thresholding methods* (Granovetter, 1978; Borodin et al., 2010; Kempe et al., 2003), *independent cascading models* (Goldenberg et al., 2001; Kleinberg, 2007), and epidemic models (Moreno et al., 2002a). Among all these, the SIR epidemic model (Newman, 2002) is the most widely used. In the SIR model, each node is in any one of the following three states: *Susceptible* (S), *Infectious* (I), and *Recovered* (R). Any I-node spreads the infection to its neighbors with probability β during each time step. Only S-nodes could be infected and, during the next time step, an I-node becomes an R-node. R-nodes become immune to the infection and could not be infected again. SIR model computes a node's spreadability by initially infecting the concerned node, and then, computing the number of recovered nodes in the network when the infection spreading process stops on its own due to the non-availability of S-nodes after some time step. We discuss the SIR epidemic spreading model in Section 7.1.

The core decomposition technique, also known as *k-shell decomposition* (Pittel et al., 1996; Carmi et al., 2007), has been used to rank the nodes based on network coreness of the nodes. It identifies influential nodes better than other centrality measures such as *degree*. Some studies (Gonzalez-Bailon et al., 2011; Pei et al., 2014) have validated on twitter data that the k-shell identifies important nodes well in an information diffusion process. It assigns each node a k-shell index (k_s) value. The more the k_s value, the more central the node is. Nodes with the highest k_s value are known as the core nodes and constitute the core shell of the network. k-shell has some serious limitations such as (1) it fails to rank nodes as per there spreadability when many core-like groups are present in the network with higher k-shell value but lower effective spreading influence; (2) it assigns many nodes to same k_s index; (3) It is observed that in some plausible circumstances nodes with lesser k_s values are found to be more important than core nodes with highest k_s index (Kitsak et al., 2010). To overcome the above, as well as coup with different application objectives, many variations of k-shell have been proposed (Zeng & Zhang, 2013; Wei et al., 2015; Zareie & Sheikhahmadi, 2018; Liu et al., 2015b). Many authors have also proposed different node ranking measures using k_s as the main component along with other centrality, and network properties (Liu et al., 2016; Bae & Kim, 2014; Liu et al., 2013; Hou et al., 2012; Chen et al., 2012; Ma et al., 2016; Li et al., 2019; Namtirtha et al., 2020; Maji, 2020; Namtirtha et al., 2018; Maji et al., 2020). In this paper, we shall review major such representative network topology based methods that are either evolved directly from the classical k-shell method or uses k_s value along with other well-known centrality measures to design their heuristic. The main contributions of the present work are-.

- a brief overview of the problem domain and applications that require identification of influential spreaders when modeled as a network graph.
- a detailed analysis of different network topology based influential spreaders identification techniques
- a discussion on commonly used performance evaluation techniques and measures such as Kendall's rank correlation, monotonicity, total spreading influence etc.
- comparative evaluation of some well-known structural neighborhood methods with benchmark SIR epidemic model on real networks.

The rest of the paper is structured as follows. Many related studies with different approaches, including topological and, structural techniques are briefly discussed in Section 2. Motivation to the study with a hypothetical problem formulation is presented in Section 3, followed by detail description of classical k-shell decomposition in Section 4. Section 5 contains the representative techniques that are directly evolved by improving the k-shell method. Next, heuristics that use the k-shell index along with other centralities to formulate their centrality measure are considered in Section 6. In Section 7, a comparative evaluation of different heuristics discussed in the above sections is performed using standard metrics, and finally, the study is concluded in Section 8.

2. Related work

In this section, For the sake of completeness of the present work, we briefly discuss many other related aspects used by

researchers to tackle the influential node ranking problem in a multitude of ways, including our primary focus of this paper, i.e., topological or structural methods.

There are mainly two different types of approaches utilized to estimate the influence capacity of a node in a network: neighborhood based approaches and path-based approaches. In the first kind of approach, most centrality measures quantify how centrally a node is placed in the network using the topological location of nodes. The most common centrality measures in this category are the *degree centrality* (Bonacich, 1972), the *H-index centrality* (Lu et al., 2016), extended H-index (Zareie & Sheikhahmadi, 2019), the k-shell decomposition (Kitsak et al., 2010), *neighborhood diversity* (Zareie et al., 2019), *clustered local-degree* centrality, which is a combination of neighbors' degree with clustering coefficient (Li et al., 2018), the *eigenvector centrality* (Bonacich, 2007), etc. The *betweenness centrality* (Newman, 2005) and the *closeness centrality* (Sabidussi, 1966) are the well-known path-based centralities.

Information diffusion techniques rank the nodes based on the global network properties. PageRank (Page et al., 1999), improved PageRank (Alp & Öğüdücü, 2018), Katz centrality (Katz, 1953), *random walk with restart* on a signed social network (Jung et al., 2019), LeaderRank (Lü et al., 2011), HITS (Kleinberg, 1998), etc. are some of the iterative neighborhood-based ranking schemes that use network-based diffusion algorithms with a random walk and works well in directed networks but degenerates to degree centrality for undirected networks.

The *degree centrality* (k) assumes that a node with a large number of links (i.e., higher node degree) to other nodes will be more important. So, node degree is used as a measure of node's importance. The *closeness centrality* (cc) uses the average geodesic (shortest path) distance between the concerned node and all other nodes in the network. The *betweenness centrality* (bc) of a node is measured with the number of shortest paths between any two nodes in the network that passes through it. Both cc and bc use complete network information while computing centrality measures, i.e., they are global measures and hence time-consuming. Therefore, bc and cc are not much useful in real-world large networks with a huge number of nodes and edges.

Salavati et al. (2019) have improved upon the closeness centrality by employing *community detection* techniques as a preprocessing and then using local structural properties of the communities to reduce the computational complexity.

The main underlying assumption in *eigenvector centrality* is that a node connected to more number of influential nodes is itself influential. The *PageRank*, the *LeaderRank, HITS*, etc. work on the assumption that if a node is visited many times by the diffusion process, then it is more central. Recently some more heuristics proposed based on similar assumptions such as VoteRank (Zhang et al., 2016; Freitas et al., 2018) HybridRank (Ahajjam & Badir, 2018), ClusterRank (Rizos et al., 2017), CumulativeRank (Zhang et al., 2019), etc. Researchers later observed that nodes with high connections (hubs) or highest betweenness are not always the most influential one (Liu et al., 2011; Kempe et al., 2015).

On a different track, many researchers have focused on applying different bio-inspired evolutionary techniques to identify the minimal nodeset with an aim to maximize the total spreading influence. Optimal percolation (Morone & Makse, 2015; Radicchi & Castellano, 2016; Del Ferraro et al., 2018), collective influence theory (Teng et al., 2016; Morone et al., 2016; Pei et al., 2017), the *structural hole theory* (Zhu et al., 2017) have recently been applied for influence maximization with the smallest initial nodes.

Zhang et al. (2012) propose an *amoeboid algorithm* and have named as *Physarum centrality* for weighted networks. Later, Gao et al. (2013) combined that with k-shell to propose another bio-inspired node ranking heuristic. Authors in (Tang et al., 2019a; Tang et al., 2019b) use *discrete particle swarm optimization* to achieve minimal nodeset. A *discrete shuffled frog-leaping algorithm* has been used in (Tang et al., 2020) for ranking the nodes. Zareie, Sheikhahmadi, and Jalili (2020) have used a *gray-wolf optimization technique* to derive the seed nodes. A *hill-climbing strategy* has been used in (Kempe et al., 2003) for the same purpose of achieving maximal influence with minimal initial nodes.

3. Motivation and problem formulation

Let us consider a hypothetical scenario of a contagious disease outbreak in a region. Initially, some people get affected and then they become infectious. It is not feasible to immunize the whole population of that region due to cost, time, and infrastructure constraints. Suppose, only a limited number of people could be immunized within the permissible time and cost. Our objective will be to immunize those people who, after being immunized, would stop the spreading. The problem reduces to finding those persons who are well-connected and have greater influence over their neighborhood. This problem is popularly known as identifying influential users in a network. A network administrator also encounters a similar situation during a virus or worm attack. She/He has limited time to act as well as a limited number of resources (anti-virus) to contain the spreading. She/He does not have the luxury to shut down the whole network. She/He is again required to select those systems (nodes) that are central to the network and after being disinfected will contain the attack. In online social networks too, rumor spreading, has similar dynamics. All the above scenarios when modeled as a network, show similar structural and topological properties and can be tackled in a generic way. However, as the networks become huge with a large number of nodes and complex relationships, the solutions need to be efficient and scalable too. Any of the above scenarios can be modeled as a network graph, where nodes can be the people or systems, and their relationships or communications or connections become the edges. In a simple unweighted model, if there is any relation/connection/communication/interaction among any two nodes, then an edge is considered between them. Mathematically, a network graph is represented as $G < V, E >$ where $V = \{v_i | i = 0 \ldots N\}$and $E = \{e_{ij} = e_{ji} = \{v_i, v_j\} | v_i, v_j \in V\}$, N is the network size = $|V|$, edge e_{ij} is the arc connecting vertices v_i and v_j; In an undirected graph $e_{ij} = e_{ji}$. Edge weights are all taken as 1. G is stored by the adjacency matrix $A = [a_{ij}]_{N \times N}$,

where $a_{ij} = 1$, if there is an edge between nodes v_i and v_j
otherwise $a_{ij} = 0$.

We have considered a small toy network with 16 nodes and 21 edges for illustrating different solution methods for the above discussed problem. We have simulated SIR model on this network and calculated spreadability of each node which is the number of recovered nodes after starting an infection from the concerned node considering infection probability as 0.4. We have simulated for a thousand times and considered the average number of *recovered* (R) nodes as spreadability of the node. Also, k-shell index values are shown using different colors to the nodes. Nodes with k-shell value = 3, 2 and 1 are colored in blue, red and black respectively. Epidemic threshold β_{th} ($=\frac{<k>}{<k^2>-<k>}$) is computed for the toy network as 0.313.

4. Classical k-shell decomposition method

The k-shell or k-core decomposition works by iterative decomposition of the network into different layers or cores. It first removes all 1-degree nodes from G and assigns each such node the k-shell index (k_s) value of 1. Then it recursively checks if some more nodes become of degree 1 due to the removal of nodes. If it finds such nodes, then removes them too and assign $k_s = 1$to them as well. The process is repeated recursively for nodes with degree

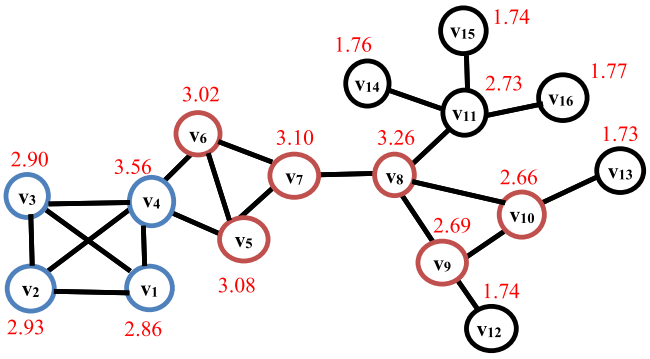


Fig. 1. Example network with 16 nodes and 21 edges. Spreadability as calculated using SIR model for each node is shown above the node labels. Also nodes are colored as per their coreness. Nodes with k-shell value = 3, 2 and 1 are colored in blue, red and black respectively. Epidemic threshold value(β_{th}) for the example network is 0.313, and the β value used during SIR simulation is 0.4. The spreadability values are obtained through 10000 simulation runs.

= 2,3. . .*k* until there are no more nodes left in *G*. Algorithm 1 outlines the process.

Let us now apply the core decomposition process step by step to the toy network shown in Fig. 1 to find the k_s values for all the nodes iteratively using Algorithm 1. In the first step, we find all nodes with *node degree* 1 and assign them a k_s value of 1. We observe that 1- degree nodes are the following. $v_{12}, v_{13}, v_{14}, v_{15}$, and v_{16}. We assign a k_s value of 1 to all those nodes. Next, we remove all the above 1-degree nodes to get the pruned graph. Now, we check again for nodes with degree 1, and node v_{11} has become a 1-degree node after pruning. So, we also assign v_{11} to the $k_s = 1$ group of nodes. Next, we remove the node v_{11} and all its edges from the graph and find that there are no other nodes in the pruned graph with degree 1. Hence, we complete finding all outer k-shell nodes with $k_s = 1$. Finally, outer k-shell nodes are $v_{12}, v_{13}, v_{14}, v_{15}, v_{16}$ and, v_{11}. Now we increment $k = 2$ and repeat the process to find nodes with a degree less than 2 in the pruned graph. We now see that v_9 and v_{10} both have degree = 2. So, we remove them both and assign $k_s = 2$ for them. Once both the above nodes are removed, node v_8 becomes a 1-degree node. Hence, we remove v_8 and assign $k_s = 2$ to it. Again, after removing v_8, node v_7 becomes a 2-degree node so, we assign $k_s = 2$ to v_7 too and remove it from the graph. Now, this makes v_6 and v_7 as 2-degree nodes. Hence, we remove them too and assign $k_s = 2$ to both of them. Next, we see there are no more nodes with degree less or equal to 2. So, the second outermost shell with $k_s = 2$ contains v_5, v_6, v_7, v_8, v_9 and, v_{10}. Once again, we increment $k = 3$ and repeat the process on the reduced graph. This time, we find all existing nodes are having a degree of 3. So, we assign $k_s = 3$ to v_1, v_2, v_3 and, v_4 and remove them from the graph. Finally, there are no more nodes in the graph. So, we halt the process. Effectively, the innermost core shell with $k_s = 3$ contains v_1, v_2, v_3 and, v_4. We have also, marked the different core shell nodes with different color in Fig. 1. Innermost core shell nodes are marked in *blue*, $k_s = 2$ nodes are marked in *red* and outermost shell nodes are colored in *black*.

There are implementations of the above method with low computational complexity of O($|V| + |E|$) (Batagelj & Zaversnik, 2003). Nodes with highest k_s belongs to the innermost core-shell and assumed to be highly influential than the outer core nodes. k-shell method produces good results when evaluated with SIR epidemic model (Newman, 2002) for influential spreader ranking. It is also observed that k-shell ranking is quite robust when used with incomplete networks. The k_s values remain almost same even when 50% of the network edges becomes missing (Kitsak et al., 2010).

One important drawback of this method is that it assigns a large number of nodes to each shell i.e. there are many number of nodes with same rank (k_s value) (Zeng et al., 2009; Barabasi & Albert, 1999). Hence it fails to rank the nodes of the same shell. Zeng and Zhang (2013) have crafted two good illustrative examples to demonstrate the limitations of k-shell. Suppose, a hub *h* is linked to a large number of spike nodes with degree *one*, then k-shell will assign $k_s = 1$ to the hub node *h*. Again, when only *three* nodes create a triangle then all of them will get $k_s = 2$. But, in reality it is apparent that the hub node *h* will be a better spreader than the nodes forming only a single triangle. Also not always a node with highest k_s value is the most influential (Kitsak et al., 2010). These limitations of k-shell necessitate the improved methods to accurately rank the nodes.

Algorithm 1: Classical k-shell Decomposition

```
input      : Network graph G < V, E >
parameters: degree k
output     : Rank[v, k_s]
where v ∈ V and k_s is the k-shell index of node v
The list Rank contains the nodes with corresponding k_s value assigned
Initially Rank contains all nodes of G with default k_s value of 1
k = 1
while !isEmpty(V) do
    repeat
        Find all nodes in G with degree = k
        foreach such node v with degree k do
            assign k_s = k
            update k_s in Rank list
            Delete node v from G
    until All remaining nodes in G have degree > k
    k = k + 1
```

Though we are focusing on node ranking methods for unweighted networks, but many of them have been extended to work with weighted networks quite easily and with little modification to the method. In a similar fashion, Garas, Schweitzer, and Havlin (2012) extended k-shell decomposition to consider network edge weights by proposing a *weighted degree* as shown in Eq. (1). The weighted k-shell decomposition uses this weighted degree instead of node degree keeping rest of the decomposition steps same. Edge weights of all neighbors are summed up and then multiplied with the node degree of the concerned node and finally, square rooted to obtain the weighted degree of a node. For illustration, let us compute the weighted degree for the node v_7 in our toy network in Fig. 1. The concerned node has three neighbors namely v_5, v_6 and v_8. Assume, each of the connecting edges have same edge-weight and the value is 1. The node degree of node v_7 is $k(v_7) = 3$. Hence, *weighted degree* of node v_7 becomes, $k\prime(v_7) = \sqrt{3 * (w(v_7, v_5) + w(v_7, v_6) + w(v_7, v_8))} = 3$.

$$k\prime(v_i) = \sqrt{k(v_i) \sum_{v_j}^{k(v_i)} w(v_i, v_j)} \tag{1}$$

5. Modified k-shell methods

In this section, discussion will be limited to only those k-shell variations that directly modify the original k-shell decomposition technique for improvement. All these methods borrow the basic iterative decomposition technique from the k-shell decomposition and attempt to tweak with the other parameters to improve upon

the classical k-shell. Only selected representative and well known modified techniques are considered in the following subsections.

5.1. Mixed degree decomposition (mdd) method

Zeng and Zhang (2013) have found some limitations of k-shell method when it is used to rank all nodes based on the spreading capability. They observed that the k-shell method ignores the removed nodes (they named it as the 'exhausted degree') and links between them after each iteration while calculating k-shell index. Only the remaining nodes (they named it as the 'residual degree') after first iteration contributes to the second round. Hence, they identified the suppression of influence of the *exhausted degree* as main reason behind such a limitation. To mitigate the above problem they have proposed a modified k-shell decomposition technique called *Mixed Degree Decomposition* (*mdd*) that takes into account the contribution of both the residual and the exhausted nodes. For each of the remaining nodes, residual degree is calculated by counting the number of edges between the concerned nodes and all other remaining nodes in the pruned network. Similarly, *exhausted degree* is calculated as the number of edges between the concerned node to the other removed nodes. The underlying assumption in k-shell decomposition is that all the remaining nodes after some iteration are homogeneously connected to the removed nodes, i.e. all remaining nodes are able to spread something to the lower layers in a similar fashion. But in reality, this assumption does not hold. As an example, consider a node in a lower layer that is connected to a large branch of removed nodes. In such a scenario this node should be ranked higher than the other nodes in the same layer. Let us consider k^r and k^e are the residual degree and exhausted degree of node v_i. The mixed degree is then defined as,

$$k^m(v_i) = k^r(v_i) + \lambda * k^e(v_i) \tag{2}$$

The *mixed degree decomposition* process is detailed step-wise in Algorithm 2.

Algorithm 2: Mixed Degree Decomposition

input : Network graph $G < V, E >$
parameters: mixed degree k^m
output : $Rank[v, k_m]$
where $v \in V$ and k_m is the m-shell index of node v
The list $Rank$ contains the nodes with corresponding k_m value assigned
Initially $Rank$ contains all nodes of G with default k_m value set to 1
Let k^r = Residual degree of a node = the number of links connected to the remaining nodes in the network k^e = exhausted degree of a node = the number of links to the removed nodes from the network.
Initially, there are no removed nodes so, $k^m = k^r = k$ and $k^e = 0$
$k^m = 1$ * We start with minimum mixed degree = 1 *\
while *!isEmpty(V)* **do**
  repeat
    Compute mixed degree of all nodes as per Equation 2
    Find all nodes in G with mixed degree $\leq k^m$
    foreach *such node v with mixed degree* k^m **do**
      assign $k_m = k^m$
      update k_m in $Rank$ list
      Delete node v from G
    Increment k^m to the next highest value
  until *All remaining nodes in G have mixed degree* $> k^m$

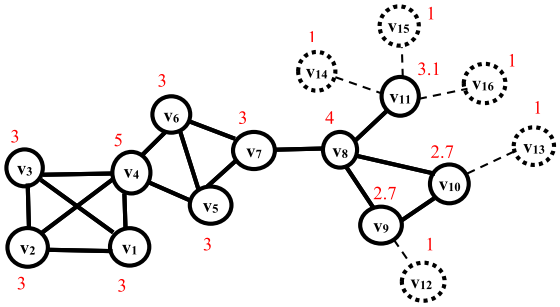
Let us go through the *mdd* decomposition process for one complete m-shell. Initially, there are no removed nodes hence, we have $k^e = 0$ and mixed degree $k^m = k^r$for all nodes. For all 1-degree nodes we obtain $k^m = 1$ using Eq. (2). So, we add $\{v_{12}, v_{13}, v_{14}, v_{15}, v_{16}\}$ to first m-shell and assign $k_m = 1$to all those nodes. Now, we remove all those nodes and recompute mixed degree of all remaining nodes using Eq. (2). For the node v_{11}, number of removed links = $k^e = 3$ and only a single link to the remaining graph (to node v_8), $k^r = 1$. Assuming $\lambda = 0.7$, we have $k^m(V_{11}) = 1 + 0.7 * 3 = 3.1$. In a similar fashion, we compute, $k^m(v_9) = 2 + 0.7 * 1 = 2.7; k^m(v_{10}) = 2 + 0.7 * 1 = 2.7$. For the time being no other node's mixed degree changes and remain same as k^r. The above step is visually depicted in Fig. 2a. Now, we observe that there are no more nodes with $k^m = 1$ so, outermost m-shell nodes are identified. Then, we consider the next highest value of mixed degree i.e. $k^m = 2.7$ and assign nodes with mixed degree less than equal to 2.7 to second outermost m-shell with the m-shell index $k_m = 2.7$. So, we assign v_9 and v_{10} to $k_m = 2.7$ shell. Now, we again prune these two nodes and recompute mixed degree of the remaining nodes. This scenario is shown in Fig. 2b.

Once, v_9 and v_{10} removed, we need to update mixed degree of v_8 only as $k^m(v_8) = 2 + 0.7 * 2 = 3.4$. At this instant the mixed degree of nodes $v_7, v_6, v_5, v_1, v_2, v_3$ is 3 (same as k^r as no links to removed nodes) and node v_4 has mixed degree value = 5 along with mixed degree of v_{11} as 3.1. Going by the rule, we have no more nodes with mixed degree value less than or equal to 2.7, so the second outermost m-shell with m-shell index $k_m = 2$ is complete with nodes v_9 and v_{10}.

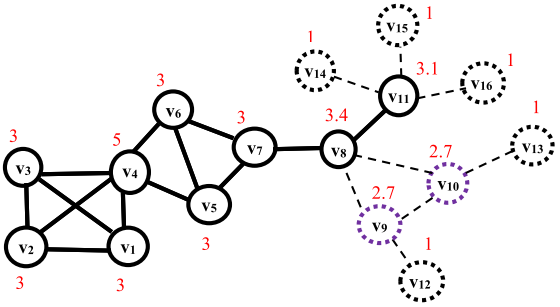
Next highest mixed degree value in the pruned graph is 3 so, we remove all nodes with mixed degree = 3 from the network and assign them to an m-shell with an m-shell index $k_m = 3$. So, we have third outermost m-shell with nodes v_1, v_2, v_3, v_5, v_6 and v_7. Next, we need to update the mixed degree value of v_4 as $k^m(v_4) = 0 + 0.7 * 5 = 3.5$ and v_8 as $k^m(v_8) = 1 + 0.7 * 3 = 3.1$. Also, note that mixed degree of v_{11} remains same as 3.1. This is depicted in Fig. 2c.

Finally, in the next shell nodes v_8 and v_{11} get added with $k_m = 4$. The last remaining node v_4 is not connected to the removed nodes, hence its mixed degree value remains at 3.5. The innermost M-shell with M shell index $k_m = 5$is assigned to node v_4. As there are no more nodes in the pruned graph the decomposition process halts. The above two steps are shown in Fig. 2d.

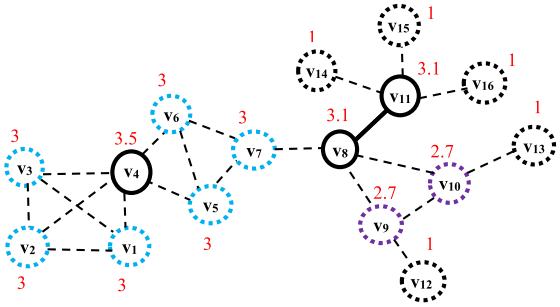
It is observed that with λ = 0 the process becomes same as the k-shell and while $\lambda = 1$, it matches degree centrality. Also unlike k-shell index, m-shell values are not always integers. To illustrate the calculations we have applied *mdd* over the toy network shown in Fig. 1 with $\lambda = 0.7$and final *M* values assigned to different nodes are shown in Table 2.



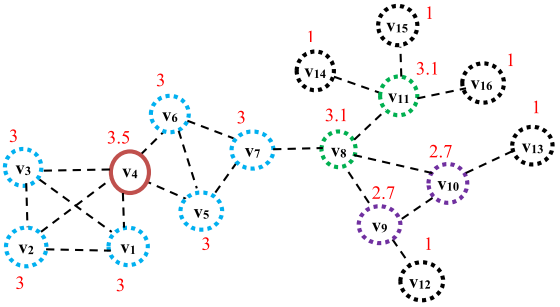
(a) Nodes with mixed degree =1 are removed. Initially, all nodes have mixed degree same as their node degree as there are no links to removed nodes.All dotted nodes are added to the outermost m-shell with m-shell index k_m=1



(b) In the next step nodes v_9 and v_{10} with mixed degree $k^m = 2.7$ are removed. All the removed nodes and links to them are shown in dotted lines.



(c) Nodes with $k^m = 3$ are removed in next step. Corresponding m-shell contains nodes v_1, v_2, v_3, v_5, v_6 and v_7.



(d) In this step nodes v_8 and v_{11} with updated mixed degree $k^m = 3.1$ are removed and assigned to next higher m-shell. Only remaining node with $k^m = 3.5$ gets assigned to the last highest m-shell. Hence, the innermost m-shell contain only a single node v_4.

Fig. 2. Mixed degree decomposition steps for toy network (Refer Fig. 1) shown step by step in (a), (b), (c) and (d). Removed nodes and removed links are shown with dotted lines and different colors are used to indicate the nodes assigned to different m-shells. Mixed degree values of each node as computed in each step are shown above each node.

5.2. K-shell decomposition based on potential edge weights

Wei et al. (2015) have proposed another solution to the monotonicity problem of k-shell by introducing the concept of *potential edge weight* for unweighted networks. It assigns the potential edge weights as the sum of the degrees of all connected nodes to the concerned node. Underlying assumption here is that an edge's importance increases as it's end nodes' degree increases. Now for any node it's degree and edge weights (in terms of edge's importance) are considered while each shells are identified. As per the definition potential edge weight of an edge $e(v_i, v_j)$ is $w(v_i, v_j) = k_{v_i} + k_{v_j}$, hence, weighted degree of a node v_i is given by Eq. (2)

$$k^w_{v_i} = \lambda k_{v_i} + (1-\lambda) \sum_{v_j \in \Gamma_{v_i}} w(v_i, v_j) \tag{3}$$

where Γ_i is the set of neighboring nodes and λ is again a tunable free parameter whose value lies between 0 and 1. Authors have considered $\lambda = 0.5$in their experiments to give equal importance to both degree and edge weight. Here the network pruning steps are same as that of classical k-shell, only difference is that it is now based on both the degree of the node as well as the potential edge weight of the incident edges (using Eq. (3)) instead of only degree in k-shell. Derived weighted degree may not be always integer as it depends on the value of the free parameter λ so authors have considered the value rounded down to the nearest integer.

We have applied their method on our toy network and weighted k-shell index (W_{KS}) of the nodes are shown in Table 2. Comparison of node ranks generated by this technique with others has also been shown in Table 3.

5.3. Hierarchical k-shell method

Zareie and Sheikhahmadi (2018) has proposed an improved version of the k-shell decomposition method named as the *hierarchical k-shell* where they have first categorized the nodes in a network with respect to a node v_i into three groups.

- *Pred*(v_i): it is the set of nodes whose shortest path to the core passes through the node v_i.
- *Succ*(v_i): it is the set of nodes that are present in the shortest path to the core from the node v_i.
- *Sibl*(v_i): it is the set of nodes in the neighborhood of v_i in which they do not visit each other on their shortest path to the network core.

For example, in our toy network the *core* consists of nodes v_1, v_2, v_3, v_4 with highest k-shell index. So, *pred*(v_8) = v_4, v_5, v_6, v_7; *Succ*(v_8)=$v_{11}, v_{14}, v_{15}, v_{16}$; *Sibl*($v_8$) = ϕ. Again, for node v_9, we have *pred*(v_{10})= v_4, v_5, v_6, v_7, v_8; *Succ*(v_{10}) = v_{13} and *Sibl*(v_{10}) = v_9. Authors have now computed two integer measures termed as b and F for each node while doing classical k-shell decomposition. Parameter b is the measure of the distance from the periphery whereas F measures the extent of closeness from the core. Finally, a measure $S(v_i)$ is formulated as

$$S(v_i) = \sum_{v_j \in \Gamma(v_i)} k(v_j) * (b(v_j) + \mathrm{F}(v_j))$$

And then hierarchical k-shell measure of a node v_i is defined iteratively over its neighborhood as depicted in Eq. (4). Authors claim its time complexity to be $O(N)$, though it requires many shortest path calculations and iterations. Its performance on very large scale networks yet to be evaluated.

$$hks(v_i) = \sum_{v_j \in \Gamma(v_i)} S(v_j) \tag{4}$$

5.4. Improved k-shell with removed redundant links

Many recent studies have identified that due to the presence of local core-like groups with very few out-ward edges in a network, k-shell accuracy reduces dramatically. For example, in our toy network, nodes v_1, v_2, v_3 and v_4 are very densely connected and form a local core-like group with only two out-ward edges from the core-like group connected to the rest of the network. We also observe that all of the four nodes are assigned same ($k_s = 3$) k-shell index value, though there spreading efficiencies are not same. Clearly, v_4 has not been given proper importance with respect to k-shell measure. Liu et al. (2015b) improve the classical k-shell decomposition by first removing redundant links based on diffusion importance of the edges and then applying classical k-shell on the residual network. A diffusion importance is assigned to the edges based on the number of out-ward edges. The diffusion importance of an edge (e_{ij}) is defined in Eq. (5).

$$\delta(v_i, v_j) = \frac{(n_{i\to j} + n_{j\to i})}{2} \tag{5}$$

where $n_{i\to j}$ and $n_{j\to i}$ are the number of out-ward edges connecting the nodes v_i and v_j to the outside of first order neighborhood nodes correspondingly. This measure provides an average diffusion importance of the network edges where nodes in a core-like group with limited out-ward links tend to get smaller values. Next, authors have defined a diffusion threshold $\delta_{th} = 2$ and removed all edges with edge importance less than δ_{th} to obtain a new graph $G\prime < V\prime, E\prime >$ from the original network $G < V, E >$ with many edges removed ($E\prime \subseteq E$ and $V\prime = V$) mostly from core-like groups. Authors have now applied classical k-shell on $G\prime$ to obtain an improved coreness value (k_s^r) for all the nodes.

6. K-shell-based methods

There are many node ranking methods that uses k-shell index as a constituent component in their heuristics. Most of the researches have used different combinations of *degree*, *closeness* or, *betweenness* along with k-Shell index to design their heuristics. Different levels of neighborhood has been applied in different studies. In the following subsection a discussion on basic concept of Neighborhood centrality is done followed by gravity and potential edge weight based strategies and finally a discussion on some recent hybrid node ranking techniques.

6.1. Based on considered neighborhood

Liu et al. (2016) propose a generic template structure where any benchmark centrality measures can be fitted in so that it may achieve an improvement over that benchmark centrality. Authors propose a node influence measure based on the centrality of a node and its neighbors' centrality. Most general formula is given by the Eq. (6) where θ denotes any one of the basic benchmark centrality measures v_i is the considered node and n is the levels of neighborhood. a is a free weight parameter with value lies between 0 and 1. Γ_i is the set of neighboring nodes of v_i. Also s is the direct (1-level) neighbor of a node v_o which is the $(s-1)$-level neighbor of v_i, and the slashed node v_x is the direct neighbor of node v_o but is the $(s-2)$-level neighbor of v_i. Here main underlying assumption is that the importance of a node depends not only on it's direct neighbors but also on distant neighbors up to n-step.

$$C_i^n(\theta) = \theta_i + a\sum_{j\in\Gamma_i}\theta_j + a^2\sum_{l\in\Gamma_i\setminus i}\theta_l + a^3\sum_{m\in\Gamma_i\setminus j}\theta_m + \ldots + a^n\sum_{s\in\Gamma_{s-1}\setminus x}\theta_s \tag{6}$$

But, one important observation made by the authors is that increasing the *neighborhood step* (i.e. n) does not improve results after a certain level of neighborhood. So, it (improvement gained with increased neighborhood) saturates quickly and does not improve indefinitely with larger range of neighborhood.

The computation cost increases with the increase of the neighborhood range so authors considered 2-step neighborhood for a good trade-off of cost and performance. They concluded that when further distant neighborhood is considered, then there is no obvious improvement and may even decrease the ranking. As the values of $a \leqslant 1$ so, the effect of neighbors' diminishes with the increase of neighborhood levels.

6.2. Neighborhood coreness method

Bae and Kim (2014) overcome monotonicity problem with a balanced mix of degree and coreness of neighbors in calculating node's ranking. Underlying assumption considered by the authors is that a node with more links to the neighbors positioned in the core of network is more influential. Based on the said assumption, the neighborhood coreness C_{nc} of node v can be defined as shown in Eq. (6)

$$c_{nc}(v_i) = \sum_{v_j\in\Gamma_{v_i}} k_s(v_j) \tag{7}$$

where Γ_{v_i} is the neighbors set of node v_i and $k_s(v_j)$ is the k-shell index of it's neighboring node v_j. Hence the extended neighborhood coreness C_{nc+} of node v_i is defined recursively as shown in Eq. (6)

$$c_{nc+}(v_i) = \sum_{v_j\in\Gamma_{v_i}} c_{nc}(v_j) \tag{8}$$

For our example toy network coreness values calculated using Eq. (7) for different nodes are as follows:

$$c_{nc}(v_1) = 9; c_{nc}(v_2) = 9; c_{nc}(v_3) = 9; c_{nc}(v_4) = 13;$$
$$c_{nc}(v_5) = 7; c_{nc}(v_6) = 7; c_{nc}(v_7) = 6; c_{nc}(v_8) = 7;$$
$$c_{nc}(v_9) = 5; c_{nc}(v_{10}) = 5; c_{nc}(v_{11}) = 5; c_{nc}(v_{12}) = 2;$$
$$c_{nc}(v_{13}) = 2; c_{nc}(v_{14}) = 1; c_{nc}(v_{15}) = 1; c_{nc}(v_{16}) = 1.$$

Extended values i.e. c_{nc+} values of nodes and ranking based on that are shown in Table 2 and Table 3.

6.3. Ranking based on distance to the network core

Liu et al. (2013) employ the idea that if a node is connected to more number of core-shell nodes (i.e. nodes with highest k_s value) then it will be more important in terms of spreading ability. As k-shell assigns a large number of nodes to same shell now nodes with same k_s value can be ranked based on their shortest distance from the highest k_s value(k_s^{Max}) core-shells. Eq. (9) is designed for the purpose.

$$\theta(v_i|k_s) = (k_s^{Max} - k_s + 1)\sum_{j\in\Gamma_{k_s^{Max}}} d_{ij},\ v_i \in S_{k_s} \tag{9}$$

where k_s^{Max} is the highest k_s value in the network, Γ_{k_s} is the core node set with $k_s = k_s^{Max}$; d_{ij} is the shortest distance between node

v_i to the node v_j, and S_{k_s} is denoted as the node set whose k-core values equal to k_s. Our example network has v_1, v_2, v_3 and v_4 nodes in core-shell with $k_s^{Max} = 3$. Again nodes $v_5 - v_{10}$ all have $k_s = 2$and rest all has $k_s = 1$. So using the above Equation nodes with same k_s value can be ranked based on the geodesic from core nodes (here v_1, v_2, v_3 and v_4). To distances from each of $k_s = 2$nodes to the core nodes are used as per Eq. (9) to arrive at a ranking among nodes v_5 to v_{10}. All calculations and ranking comparison are shown in Table 2 and Table 3 respectively.

6.4. All around distance method

Hou et al. (2012) have observed that the node ranking achieved using many different centrality measures to real networks are correlated. They exploited the fact to propose a combined metric called all-around distance that combines degree, betweenness and k-shell to quantify a node's influence. They normalize each of the measures without loosing their original distribution in rank list before computing the combined metric. Authors reasons that degree provides local influence, whereas betweenness gives a global perspective with positional importance imposed by k-shell value of a node. Hence combining them into a single measure will incorporate all three different perspectives to node ranking as shown in Eq. (10).

$$S_a(v_i) = \sqrt{\|d(v_i)\|^2 + \|C_b(v_i)\|^2 + \|k_s(v_i)\|^2} \tag{10}$$

where $S_a(v_i)$ is the all around distance of node v_i and $d(v_i), C_b(v_i), k_s(v_i)$ represents the degree, betweenness centrality and k-shell index value of node v_i respectively. One advantage of this measure is that, it is more stable to the changes in spreading probability or ranking technique, as only those nodes tops the list that are scored good in all individual ranking.

6.5. Semi local centrality method

Chen et al. (2012) used degree centrality along with other global centralities and considers only two step neighborhood as a trade off between performance and computation. Local centrality (C_L) has been proposed by the authors and defined as shown in Eq. (11).

$$C_L(v_i) = \sum_{v_j \in \Gamma(v_i)} \sum_{v_k \in \Gamma(v_j)} N(v_k) \tag{11}$$

where $\Gamma(v_i)$ contains the neighboring nodes of v_i and $N(v_k)$ is the number of nodes in 2-step nearest neighborhood of node v_k. Lets consider our toy network as an example to understand the calculations. Node v_1 has 3 neighbors, $\Gamma(v_1) = v_2, v_3, v_4$. Now, $N(v_2)$ is the number of 2-step neighbors of v_2, = 5. Similarly, $N(v_3) = 5, N(v_4) = 6$. Summing them up for node v_1 gives, 16. For nodes v_2, v_3 both give 16 and v_4 gives a value of 29. Now we add them up to get $C_L(v_1) = 16 + 16 + 16 + 29 = 61$. C_L values of all the nodes have been calculated and tabulated in Table 2. Calculating N(w) requires to visit w's neighborhood within 2 hop, with complexity of $O(n < k^2 >)$ and it linearly grows with the size of the sparse network.

Ma and Ma (2017) have observed that, in real networks, degree centrality performs better with small spreading probabilities and *LC* is more suitable in larger cases. To overcome these sensitivity to the spreading probability, they modified *LC* and then integrated with the degree centrality by considering the spreading probability to propose *hybrid degree centrality* (*HC*). It takes the advantages of *degree centrality* or *local centrality* depending on the given spreading probability. Authors have divided nodes' influence on spreadability into near-source and distal where, near-source influence is the ability of an infected node to influence its one-hop neighbors and distal influence is the ability to influence the nodes with more than one hop distance. To incorporate this characteristics they had modified *LC* to calculate distal influence (and named *MLC*) in such a way to remove the one-hop neighbors from *LC* to get only the 2-hop neighborhood as shown in Eq. (11)

$$C_M(v_i) = \sum_{v_j \in \Gamma(v_i)} \sum_{v_k \in \Gamma(v_j)} N(v_k) - 2 \sum_{v_j \in \Gamma(v_i)} \Gamma(v_j). \tag{12}$$

They have employed the basic assumption that put more weight on node degree (K) with low spreading probability (β) and make *MLC* more important with higher β. Thus *HC* value of a node v_i is defined as,

$$C_H(v_i) = (\alpha_1 - \beta) * \alpha_2 * d(v_i) + \beta * C_M(v_i) \tag{13}$$

where α_1, α_2 are two free parameters. α_1 adjusts the K/MLC ratio to improve the *HC* performance and α_2 is used to compensate the magnitude difference in K and *MLC* value, and bring them to a close level.

Pei et al. (2014) also propose a similar local heuristic to estimate the spreading influence of a node by using the sum of degrees of the nearest neighbors as shown in Eq. (14). This method is also known as the *Neighbor's degree* method. The most important advantage of any such node degree based method lies in the fact that it does not require complete network topology for computation. The same fact also makes computation easy for large networks.

$$k_{sum}(v_i) = \sum_{v_j \in \phi(v_i)} k(v_j) \tag{14}$$

Let us calculate k_{sum} value for node v_7 for the toy network. The neighborhood of the node v_7 = v_5, v_6, v_8. So, $k_{sum}(v_7) = k(v_5) + k(v_6) + k(v_8) = 3 + 3 + 4 = 10$.

It is the right place to mention a recent hybrid centrality measure named as *clustered local degree* (Li et al., 2018) that uses sum of degree of nearest neighbors with the clustering coefficient to index the network nodes.

6.6. Ranking based on gravity formula

Isaac Newton's famous gravity formula states that, it is proportional to the mass of the bodies and inversely proportional to the distance between them. Ma et al. (2016) have applied the same fact to identify a node's influence in spreading activity. They considered k-shell value of a node as *mass* and the shortest path distance between any two nodes in the network as the *distance* to propose their gravity measure formula as shown in Eq. (15).

$$G(v_i) = \sum_{v_j \in \Psi(v_i)} \frac{k_s(v_i) * k_s(v_j)}{s^2(v_i, v_j)} \tag{15}$$

where $s(v_i, v_j)$ is the shortest distance between node v_i and v_j. $\Psi(v_i)$ is the set of neighborhood nodes of v_i up to the level of a given value, r. Authors have considered $r = 3$, so, in that case $\Psi(v_i)$ consists of 3 levels of neighboring nodes. We shall also consider $r = 3$ while apply this to our toy network. Authors have also proposed an extended gravity formula as shown in Eq. (15)

$$G_+(v_i) = \sum_{v_j \in \Gamma(v_i)} G(v_j) \tag{16}$$

where $\Gamma(v_j)$ is the set of nearest neighbors of v_i.

We now calculate the value of $G(v_7)$ for node v_7 on the toy network to elaborate the computation steps. The (r =) 3 step neighborhood of the node v_7 = $\{(v_5, v_6, v_8), (v_4, v_9, v_{10}, v_{11}), (v_1, v_2, v_3, v_{12}, v_{13}, v_{14}, v_{15}, v_{16})\}$. We keep nodes for each values of r within a bracket just for understanding. So,

$$G(v_7) = k_s(v_7) * \left\{\frac{k_s(v_5)+k_s(v_6)+k_s(v_8)}{1^2} + \frac{k_s(v_4)+k_s(v_9)+k_s(v_{10})+k_s(v_{11})}{2^2} + \frac{k_s(v_1)+k_s(v_2)+k_s(v_3)+k_s(v_{12})+k_s(v_{13})+k_s(v_{14})+k_s(v_{15})+k_s(v_{16})}{3^2}\right\}$$
$$= 2 * \left\{\frac{2+2+2}{1} + \frac{3+2+2+1}{4} + \frac{3+3+3+1+1+1+1+1}{9}\right\} = 19.11.$$

We compare the ranking in Table 3 with calculated measures depicted in Table 2.

Recently Li et al. (2019) have proposed another similar node indexing heuristic based on the *Gravity formula* but used node degree as the proxy for mass instead of the k-shell index as above. They proposed two variants of the node influence estimation technique. In the more general formulation as depicted in Eq. (17), authors have used the shortest distance between all pairs of nodes. The problem with the above model is that when the network is large then calculating the shortest distance between all pairs of nodes becomes prohibitive. To avoid such huge computational complexity without much compromise on final outcome authors have proposed *Local Gravity Model* that considers only r level of nodes' influence instead of all nodes in the network. They have experimented with different values of r and when its value is near half of the network diameter ($r \approx D/2$), it yields almost equivalent spreading influence estimate to the more exhaustive general method.

$$S(v_i) = \sum_{v_j \neq v_i} \frac{k(v_i) * k(v_j)}{s^2(v_i, v_j)} \tag{17}$$

6.7. Methods assuming edge weights

Namtirtha et al. (2020) have proposed a *weighted k-shell degree neighborhood method* by assigning weights to the edges using node degree and k-shell index of end nodes. The ksd^w method computes it's measure by adding up all such potential edge weights of all edges connected to the concerned node. It is designed as per Eq. (19).

$$w_{ij} = (c_1 * k(v_i) + c_2 * k_s(v_i)) * (c_1 * k(v_j) + c_2 * k_s(v_j)) \tag{18}$$

$$ksd^w(v_i) = \sum_{v_j \in \Gamma(v_i)} w_{ij} \tag{19}$$

Where $\Gamma(V_i)$ is the set of neighbors of node v_i and c_1, c_2 are two tunable free parameters. Authors have used a set of free parameters c_1 and c_2 whose values lies between $[0, 1]$. Proper tuning of these values yield good results in ranking the nodes based on their influence.

A similar approach, but without the free parameters is proposed in Maji (2020). Author have assigned potential edge weights using a combination of the k-shell index and the node degree values of end nodes of an edge as per Eq. (20) where the parameter c is computed using some well-known network parameters and it is derived as

$$c = \frac{< k_s >}{< k >}$$

, where $< k >$ and $< k_s >$ are the average degree and average k-shell index value of the network. Finally, the iterative computation remains same as shown in Eq. (21).

$$w^p_{ij} = (k_s(v_i) + k_s(v_j)) + c * (k(v_i) + k(v_j)) \tag{20}$$

$$ksd^w_e(v_i) = \sum_{v_j \in \Gamma(v_i)} w^p_{ij} \tag{21}$$

As an example, we shall now compute both the above measures on our toy network for node v_7 to understand the process step by step. For the illustrative purpose let us consider the free parameter values for the ksd^w method as $c_1 = 0.1, c_2 = 0.4$ and average degree $< k >= 2.625$, average k-shell value $< k_s >= 1.875$, hence parameter $c = \frac{< k_s >}{< k >} = 0.714$. The neighboring nodes of v_7 are v_5, v_6 and v_8. $ksd^w(v_7) = w_{75} + w_{76} + w_{78}$. Now, $w_{75} = (c_1 * k(v_7) + c_2 * k_s(v_7)) * (c_1 * k(v_5) + c_2 * k_s(v_5)) = (0.1 * 3 + 0.4 * 2) * (0.1 * 3 + 0.4 * 2) = 0.0121$. Similarly, $w_{76} = 0.0121$, and $w_{78} = 0.0132$. Hence, $ksd^w = 0.0121 + 0.0121 + 0.0132 = 0.0374$.

Let us now compute the ksd^w_e measure for the same node v_7 as per Eqs. (21) and (20). $ksd^w_e(v_7) = w^p_{75} + w^p_{76} + w^p_{78}$. Now, $w^p_{75} = (k_s(v_7) + k_s(v_5)) + c * (k(v_7) + k(v_5)) = (2 + 2) + 0.714 * (3 + 3) = 8.284$. Similarly, $w^p_{76} = 8.284$ and, $w^p_{78} = 8.998$. Hence, $ksd^w_e(v_7) = 8.284 + 8.284 + 8.998 = 25.566$.

Shao, Liu, Zhao, and Liu (2019) propose a similar method that considers edge importance along with second-degree neighbors impact to index the nodes. The *weight neighborhood centrality* (Wang, Hou, Li, & Ding, 2017) has also been proposed considering the diffusion importance of the edges in a complex network. The diffusion importance of an edge $e(v_i, v_j)$ is defined as

$$w^d_{ij} = k(v_i) * k(v_j).$$

The weight neighborhood centrality of node v_i assuming a benchmark centrality as ϕ is then formulated as depicted in Eq. 22.

$$c_{v_i}(\phi) = \phi(v_i) + \sum_{v_j \in \Gamma(v_i)} \frac{w^d_{ij}}{< w^d >} \phi(v_j) \tag{22}$$

where $\Gamma(v_i)$ is the set of neighbors of node $v_i, < w^d >$ is the average diffusion importance of all edges and ϕ could be any benchmark centrality such as degree or k-shell etc. This technique uses edges' importance derived from connecting node degrees and final measure is derived with a combination of nodes' centrality and weighted neighbors' centralities. Though this method yields good ranking but it is not computationally competitive as it requires to compute the benchmark centrality measures of each nodes and then compute the diffusion importance of all edges and finally, iterates over all neighbors of the concerned node.

6.8. K-shell hybrid methods

Namtirtha et al. (2018) have combined the node degree and k-shell index to propose a k-shell hybrid method as shown in Eq. (23). They have derived the benefits of a local measure (k) and a global measure (k-shell) and combined them using a free parameter μ. Authors have first computed a *k-shell power* as

$$\alpha(v_i, v_j) = \sqrt{k_s(v_i) + k_s(v_j)}$$

and then combine it with node degree. They obtain good performance with carefully chosen values of the free parameter γ for different networks. They also propose an iterative version of their method which is a cumulative measure of all neighbors *ksh* values.

$$ksh(v_i) = \sum_{v_j \in \phi(v_i)} \frac{\alpha(v_i, v_j) + \gamma * k(v_j)}{s^2(v_i, v_j)} \tag{23}$$

Let us now compute the *ksh* value of a node v_7 of our toy network (Fig. 1). Also, assume considered level of neighborhood $r = 2$ and the free parameter $\gamma = 0.4$. The 2-level neighborhood nodes of v_7 are $v_5, v_6, v_8, v_4, v_9, v_{10}, v_{11}$.

$$ksh(v_7) = \frac{\alpha(v_7,v_5)+\gamma * k(v_5)}{s^2(v_7,v_5)} + \frac{\alpha(v_7,v_6)+\gamma * k(v_6)}{s^2(v_7,v_6)} + \frac{\alpha(v_7,v_8)+\gamma * k(v_8)}{s^2(v_7,v_8)} + \frac{\alpha(v_7,v_4)+\gamma * k(v_4)}{s^2(v_7,v_4)} + \frac{\alpha(v_7,v_9)+\gamma * k(v_9)}{s^2(v_7,v_9)} + \frac{\alpha(v_7,v_{10})+\gamma * k(v_{10})}{s^2(v_7,v_{10})} + \frac{\alpha(v_7,v_{11})+\gamma * k(v_{11})}{s^2(v_7,v_{11})}$$
$$= \frac{\sqrt{2+2}+0.4 * 3}{1^2} + \frac{\sqrt{2+2}+0.4 * 3}{1^2} + \frac{\sqrt{2+2}+0.4 * 4}{1^2} + \frac{\sqrt{2+3}+0.4 * 5}{2^2} + \frac{\sqrt{2+2}+0.4 * 3}{2^2} + \frac{\sqrt{2+2}+0.4 * 3}{2^2} + \frac{\sqrt{2+1}+0.4 * 4}{2^2} = 12.692.$$

Recently, Maji et al. (2020) improve the above mentioned *ksh* method by providing a mathematical formulation for the free parameter γ instead of guessing it's value through trial and error. They have also updated the calculation of *k-shell power* as

$$\alpha_l(v_i,v_j) = \sqrt{k_s(v_i) * k_s(v_j)}$$

and the free parameter γ has been defined as a function of the level of neighborhood (l) considered. It is defined as

$$\gamma(l) = \frac{2 * (r-l+1)}{r*(r+1)},$$

where r is the considered maximum neighborhood. Now, the improved k-shell hybrid method (*kshi*) is formulated as in Eq. 24.

$$kshi(v_i) = \sum_{l=1}^{r} \sum_{v_j \in \phi(v_i,l)} \frac{\alpha_l(v_i,v_j)+\gamma(l)*k(v_j)}{s^2(v_i,v_j)} \tag{24}$$

where node $\phi(v_j,l)$ is the l-level neighborhood of node v_i. To clarify the computation, we shall calculate the *kshi* measure of a selected node (v_7) from the toy network. We consider ($r=2$) level of neighborhood while calculating the measure. The 2-level neighborhood nodes of v_7 are as follows. In the first level v_5, v_6, v_8 and, in the second level v_4, v_9, v_{10}, v_{11}.

$$kshi(v_7) = \frac{\alpha_l(v_7,v_5)+\gamma(1) * k(v_5)}{s^2(v_7,v_5)} + \frac{\alpha_l(v_7,v_6)+\gamma(1) * k(v_6)}{s^2(v_7,v_6)} + \frac{\alpha_l(v_7,v_8)+\gamma(1) * k(v_8)}{s^2(v_7,v_8)} + \frac{\alpha_l(v_7,v_4)+\gamma(2) * k(v_4)}{s^2(v_7,v_4)} + \frac{\alpha_l(v_7,v_9)+\gamma(2) * k(v_9)}{s^2(v_7,v_9)} + \frac{\alpha_l(v_7,v_{10})+\gamma(2) * k(v_{10})}{s^2(v_7,v_{10})} + \frac{\alpha_l(v_7,v_{11})+\gamma(2) * k(v_{11})}{s^2(v_7,v_{11})}$$
$$= \frac{\sqrt{2 * 2}+0.67 * 3}{1^2} + \frac{\sqrt{2 * 2}+0.67 * 3}{1^2} + \frac{\sqrt{2 * 2}+0.67 * 4}{1^2} + \frac{\sqrt{2 * 3}+0.33 * 5}{2^2} + \frac{\sqrt{2 * 2}+0.33 * 3}{2^2} + \frac{\sqrt{2 * 2}+0.33 * 3}{2^2} + \frac{\sqrt{2 * 1}+0.33 * 4}{2^2} = 14.56.$$

In summary, we have depicted node-wise centrality metrics assigned to the toy network (see Fig. 1) nodes by some selected representative techniques discussed in Section 5 and 6 in Table 2. The computational steps are illustrated for a single node during the discussion of the methodology of the corresponding techniques. Here we have consolidated such measure values for all the nodes of the toy network so that one can easily verify their understanding by cross checking these metrics for individual nodes. In Table 3 we have shown the rank-wise grouping of the nodes as generated by different techniques on the toy network. Such kind of representation also provides us an analytical view on *ranking monotonicity* (more on this metric discussed in Section 7.3) i.e. number of nodes having unique rank. We observe that k-shell has divided the nodes into only three ranks, whereas degree centrality grouped into four ranks. Again, gravity plus has highest number of unique ranks. It is apparent that Gravity plus has more uniquely identified the relative positions of the nodes based on their influentiality whereas, k-shell or degree has assigned many nodes together even though they differ greatly in their spreading capacity. Actual spreadability of the nodes are estimated using the SIR model simulation and second column of Table 3 shows the actual spreadability of the nodes as measured by SIR model simulation over 10000 runs. We consider the ranking generated by the SIR model spreadability measure as benchmark and evaluate all other heuristic methods by comparing their node ranking with respect to the SIR ranks.

7. Evaluation and discussion

In influential node identification studies many different variations of epidemic spreading models are used as a benchmark to evaluate and compare other ranking techniques. Most popular models are linear threshold model (Watts, 2002), independent cascade model (Saito et al., 2008), epidemic models such as SI, SIS (Nåsell, 1996) and SIR model (Newman, 2002). The *SIR* model is the simplest but the most effective among them. Different ranking techniques discussed in the paper use their own set of measures and assign values to the nodes for ranking. So it is not possible to compare them based on the absolute values assigned to nodes by different methods. Node's relative position in the list is important, not their absolute values. Kendall's τ correlation has been used in such a scenario to compare two ranked list. We have used our toy network in Fig. 1 to demonstrate the methodology as well as results arrived by different studies. Table 1 summarizes the different symbols used in different studies along with a brief description. Many different parameters has been used in representative studies and we have tried to set the values as per original studies wherever possible. Overall calculations of some of the heuristic measures for node ranking are done in Table 2 and the final ranking comparison of the toy network nodes based on the above calculation are done in Table 3. We aim at reviewing only k-shell based node ranking techniques with their merits and demerits with a small example so that a novice to this area can quickly gain understanding of the basic tools and techniques employed by the popular studies. We discuss the SIR model and Kendall's τ rank correlation along with monotonicity and average spreading influence in the following subsections.

7.1. SIR epidemic model

Susceptible-Infectious-Recovered (SIR) epidemic model (Pastor-Satorras & Vespignani, 2001; Newman, 2002; Moreno et al., 2002b) has long been used to model disease spreading within a population. Later, it has been used to model rumor spreading, virus propagation, information diffusion in a network etc. This model initially considers all nodes as susceptible (S) except the one whose spreading capacity is to be measured. That single node is considered as infected (I) and it can infect its neighbors with an infection spreading probability β. At each time step, initial source node infects its

Table 1
Symbols used with description.

Symbol	Description	Symbol	Description
V	Set of nodes in the network	E	Set of edges
$G < V, E >$	Network graph with V and E	N	Network size
$e(v_i, v_j)$	Edge between the nodes v_i and v_j	$w(v_i, v_j)$	Weight of the edge $e(v_i, v_j)$
$k_s(v_i)$	k-shell index of node v_i	k	Node degree
k_i^r	Residual degree of node v_i	k_i^e	Exhausted degree of node v_i
k_i^m	Mixed degree of node v_i	λ	Tunable free parameter between 0,1
C_i^n	Neighborhood centrality of node v_i	a	Neighborhood weight parameter between 0,1
k_s^{Max}	Highest k_s index assigned to a network node v_i	θ	Node value using geodesic distance from core nodes
β	Spreading probability of a node	$s(v_i, v_j)$	shortest path distance between nodes v_i and v_j
μ	Recover probability of a node	$< k >$	Average node degree of a network
β_{th}	SIR epidemic threshold probability	D	Network Diameter
$p < mean >$	Average shortest path length	C	Average clustering coefficient

Table 2
Values assigned to toy network nodes by different k-shell variants such as degree centrality (k), k-shell (k_s), mixed degree decomposition (mdd), weighted k-shell (W_{KS}), neighborhood coreness (nc), coreness plus (c_{nc+}), improved method (θ), number of 2-step neighbors (N_V) required in computations for hybrid centrality (C_H), local centrality (C_L), Gravity (G) and Gravity Plus (G_+), Neighbors' degree (k_{sum})).

Node	Degree	k_s	mdd	W_{KS}	c_{nc}	c_{nc+}	θ	$N(v)$	$C_L(v)$	G	G_+	k_{sum}
v_1	3	3	3	10	9	31	3	5	61	30.67	102.5	11
v_2	3	3	3	10	9	31	3	5	61	30.67	102.5	11
v_3	3	3	3	10	9	31	3	5	61	30.67	102.5	11
v_4	5	3	3.5	10	13	41	3	6	88	41.17	133.22	15
v_5	3	2	3	5	7	26	14	7	74	20.61	80.89	11
v_6	3	2	3	5	7	26	14	7	74	20.61	80.89	11
v_7	3	2	3	6	6	21	22	7	66	19.11	60.39	10
v_8	4	2	3.1	2	7	21	30	11	88	19.11	53.39	13
v_9	3	2	2.7	5	5	14	38	6	52	13.56	36.17	8
v_{10}	3	2	2.7	5	5	14	38	6	52	13.56	36.17	8
v_{11}	4	1	3.1	2	5	10	57	7	47	7.17	27.17	7
v_{12}	1	1	1	2	2	5	69	3	20	3.44	13.56	3
v_{13}	1	1	1	2	2	5	69	3	20	3.44	13.56	3
v_{14}	1	1	1	3	1	5	69	4	23	2.67	7.17	4
v_{15}	1	1	1	3	1	5	69	4	23	2.67	7.17	4
v_{16}	1	1	1	3	1	5	69	4	23	2.67	7.17	4

Table 3
Ranking of toy network nodes by selected variants of k-shell method such as degree (k), k-shell, mixed degree decomposition (mdd), potential edge weight based k-shell index (W_{KS}), neighborhood coreness plus (c_{nc+}), improved method(θ), local centrality (C_L) and Gravity plus (G_+). In the columns we shown only the vertex indices i.e. in "Degree(k)" column, first row with "Rank = 1" has a value of "4", it signifies vertex v_4, similarly for all other columns except the first two, a numerical value "x" signifies "vertex v_x". We skip many other methods due to space constraints.

Rank	SIR	Degree(k)	k-shell	mdd	W_{KS}	c_{nc+}	θ	C_L	G_+
1	4	4	1,2,3,4	4	1,2,3,4	4	1,2,3,4	4,8	4
2	8	11,8	10,5,6,7,8,9	11,8	5,6,7	1,2,3	5,6	5,6	1,2,3
3	7	others	others	1,2,3,5,6,7	8,9,10	5,6	7	7	5,6
4	6	12,13,14,15,16	–	9,10	others	7,8	8	1,2,3	7
5	5	–	–	others	12,13	9,10	9,10	9,10	8
6	3	–	–	–	–	11	11	11	9,10
7	1	–	–	–	–	others	others	14,15,16	11
8	2	–	–	–	–	–	–	12,13	12,13
9	11	–	–	–	–	–	–	–	14,15,16
10	10	–	–	–	–	–	–	–	–
11	9	–	–	–	–	–	–	–	–
12	15	–	–	–	–	–	–	–	–
13	14	–	–	–	–	–	–	–	–
14	12	–	–	–	–	–	–	–	–
15	16	–	–	–	–	–	–	–	–
16	13	–	–	–	–	–	–	–	–

susceptible neighbors with probability β and those nodes who get infected change their status from 'S' to 'I'. At the same time if incubation period is 1 unit of time step then, infected nodes start infecting its neighbors. Also, 'I' nodes recover with a rate of μ. Once recovered, a node's status changes to 'R' and it can not be infected again. For simplicity, μ is taken as 1 in most of the cases. This process continues until the infection dies out. The final count of the recovered (R) nodes represents the spreading capability of the source node. One important point to keep in mind is that β value must be kept small (just little more than $\beta_{th} = \frac{<k>}{(<k^2>-<k>)}$(Castellano

& Pastor-Satorras, 2010)), otherwise whole network will eventually be infected independent of the source node and we shall be unable to measure the individual node's influence. It is observed that when β is kept small, spreading reaches up to a finite portion of the network for most of the seed nodes (Kitsak et al., 2010).

7.2. Kendall's τ correlation

To compare two ranked list Kendall's τ correlation (Kendall, 1945; Knight, 1966) is used. It compares items of two lists for their relative positions. Its value lies between -1 to 1. It is calculated based on the number of concordant and discordant pairs. Let us consider two ranked list $L1$ and $L2$. Let (p1,q1) and (p2,q2) be a set of a joint rank. If $p1 > p2$ and $q1 > q2$ or $p1 < p2$ and $q1 < q2$, then rank (p1,q1) and (p2, q2) are concordant. If $p1 > p2$ and $q1 < q2$ or $p1 < p2$ and $q1 > q2$, they are discordant. If $p1 = p2$ or $q1 = q2$, the rank is neither concordant nor discordant. The Kendall's rank correlation τ is defined as,

$$\tau = \frac{2 * (N1 - N2)}{N * (N - 1)}, \tag{25}$$

where $N1$ and $N2$ are the number of concordant and discordant pairs and N is the network size. A high τ value signifies more similarity between the given ranked lists while a lower τ indicates dissimilarity. If both the lists are same then $\tau = 1$. When both the lists are exactly opposite to one another (i.e. 2^{nd} list is reverse-ordered version of the 1st one) then τ value becomes -1. Average Kendall's rank correlation can be calculated by using different infection probabilities in increasing steps and then taking average.

$$\tau_A = \frac{1}{n}\sum_{j=1}^{n}\tau(\beta = \beta_{th} + \delta * j) \tag{26}$$

where n is the total number of results taken and δ is the increase in infection probability in each step.

7.3. Monotonicity (M)

Monotonicity (M_r) of ranking measures the uniqueness of the ranked nodes. Its value lies between 0 and 1. The monotonicity of a ranking method increases when it assigns distinct ranks to all individual nodes while it decreases with a lot of nodes put into same rank position in the list. Bae and Kim (2014) proposed this metric to measure the uniqueness of the elements in a ranking list and it is computed following Eq. (27).

$$M_r(L) = \left[1 - \frac{\sum_{r \in L} N_t(r)(N_t(r) - 1)}{N(N-1)}\right]^2 \tag{27}$$

where L is the ranking list and $N_t(r)$ denotes the number of ties with the same rank r. N is the number of elements in the ranked list L. We have calculated the monotonicity score for some of the ranking schemes discussed in this study. All are calculated on the ranking of our toy network.

7.4. Average spreading influence (SI)

As we have seen SIR spreading model is probabilistic in nature and evaluating a node's spreading influence on such models may

Table 4
Network parameter of some benchmark dataset. All measures calculated using Gephi Bastian et al., 2009 with network representation as un-directed, unweighted, without self-loop and duplicate edges.

Network	V	E	β_{th}	β_{exp}	D	$< k >$	$p < mean >$	C
NS (Newman, 2003)	1589	2742	0.144	0.15	17	3.415	5.823	0.878
C.elegans (Duch & Arenas, 2005)	306	2148	0.038	0.06	5	14.039	2.455	0.308
PB (Adamic & Glance, 2005)	1490	16718	0.012	0.15	8	22.44	2.738	0.361
Advogato (Massa et al., 2009)	6541	51127	0.012	0.05	9	7.819	3.275	0.408
Zachary (Zachary, 1977)	34	78	0.129	0.15	5	2.294	2.408	0.588
PG (Watts & Strogatz, 1998)	4941	6594	0.258	0.28	46	2.669	18.989	0.107
Hamsterster (Konect, 2017)	1858	12534	0.022	0.07	15	8.104	5.104	0.136
USAirLine (Batagelj & Mrvar, 1998)	332	2126	0.023	0.05	6	6.4	2.738	0.749
Jazz (Gleiser & Danon, 2003)	198	2742	0.026	0.06	6	13.848	2.235	0.633
Dolphins (Lusseau et al., 2003)	62	159	0.147	0.15	8	5.129	3.357	0.303
Citation (Ley, 2002)	12591	49743	0.023	0.05	10	3.951	4.423	0.192
Euroroads (Šubelj & Bajec, 2011)	1174	1417	0.333	0.35	62	2.414	18.371	0.02
High School (Coleman, 1964)	70	182	0.107	0.15	6	5.2	2.942	0.258
Foodweb (Martinez et al., 1991)	183	2494	0.023	0.05	4	13.626	2.147	0.325
Macaques (Takahata, 1991)	62	1187	0.026	0.05	2	19.145	1.383	0.667
Football (Girvan & Newman, 2002)	115	613	0.093	0.35	4	10.661	2.508	0.403
ODLIS (Joan, 2002)	2909	18241	0.014	0.05	8	6.290	3.17	0.351

Table 5
Kendall's τ rank correlation ($\tau(.)$) of different variants of the k-shell methods with the SIR model simulated ranking for varying infection probability ranging from epidemic threshold β_{th} to twice of that on our toy network. We have employed the following methods in the study: degree(k), k-shell(k_s), mixed degree decomposition (mdd), potential edge weight based k-shell index (W_{KS}),coreness plus (c_{nc_+}), improved method (θ), local centrality (C_L), gravity plus (G_+), local gravity model(S), and neighbors' degree method (k_{sum}). Results are averaged over 100 independent runs.

β	$\tau(k)$	$\tau(k_s)$	$\tau(mdd)$	$\tau(W_{KS})$	$\tau(c_{nc_+})$	$\tau(\theta)$	$\tau(C_L)$	$\tau(G_+)$	$\tau(S)$	$\tau(k_{sum})$
0.341	0.699868	0.557773	0.766748	0.608588	0.608698	0.578481	0.887796	0.528540	0.857794	0.792875
0.372	0.699868	0.557773	0.766748	0.554088	0.644503	0.614636	0.870388	0.615186	0.840465	0.775058
0.403	0.659296	0.597614	0.728884	0.626755	0.680309	0.650791	0.940019	0.615186	0.783083	0.787253
0.434	0.639010	0.577694	0.709952	0.554088	0.662406	0.632714	0.887796	0.649844	0.771148	0.775058
0.465	0.639010	0.577694	0.709952	0.608588	0.644503	0.632714	0.818165	0.632515	0.788477	0.757240
0.496	0.618724	0.597614	0.691020	0.626755	0.662406	0.650791	0.835573	0.649844	0.719161	0.685971
0.527	0.618724	0.597614	0.691020	0.590421	0.662406	0.650791	0.800757	0.684502	0.753819	0.757240
0.558	0.662060	0.560112	0.731940	0.510803	0.611250	0.580907	0.812855	0.626466	0.684502	0.685971
0.589	0.618724	0.517932	0.653156	0.463254	0.572892	0.542326	0.765942	0.597857	0.684502	0.685971
0.620	0.598438	0.498012	0.634224	0.445087	0.554989	0.524249	0.748534	0.580527	0.719161	0.721605

yield different results. It is not wise to rank a node on top of the list based on a single measure of influence. Also, it may occur that for a particular node ranking method, a node's estimated spreading influence is very high while for other nodes it fails. To avoid any such partiality in comparing different node indexing techniques average spreading influence is useful. Pei et al. (2014) have used this measure using varying fraction of nodes. Suppose, a network has N nodes and we want to consider $f\%$ nodes, then we first select top $f * N$ nodes from the ranking list and compute spreading influence of each node separately. Finally, we sum up all individual influences and divide by the number of fraction nodes i.e. $f * N$ to get the average influence of each node with respect to top $f\%$ nodes. Average influence with top f fraction of nodes $SI(f)$ is calculated as shown in Eq. (28).

$$SI(f) = \frac{1}{f * N} \sum_{v_k \in V_L} \sigma(v_k) \tag{28}$$

where $\sigma(v_k)$ is the spreading influence of node v_k and, V_L is the set of top $f * N$ nodes in the rank list generated by any node ranking technique. Also, this measure is used in comparing total influence in *influence maximization* problems where a seed nodeset is identified to maximize the collective influence.

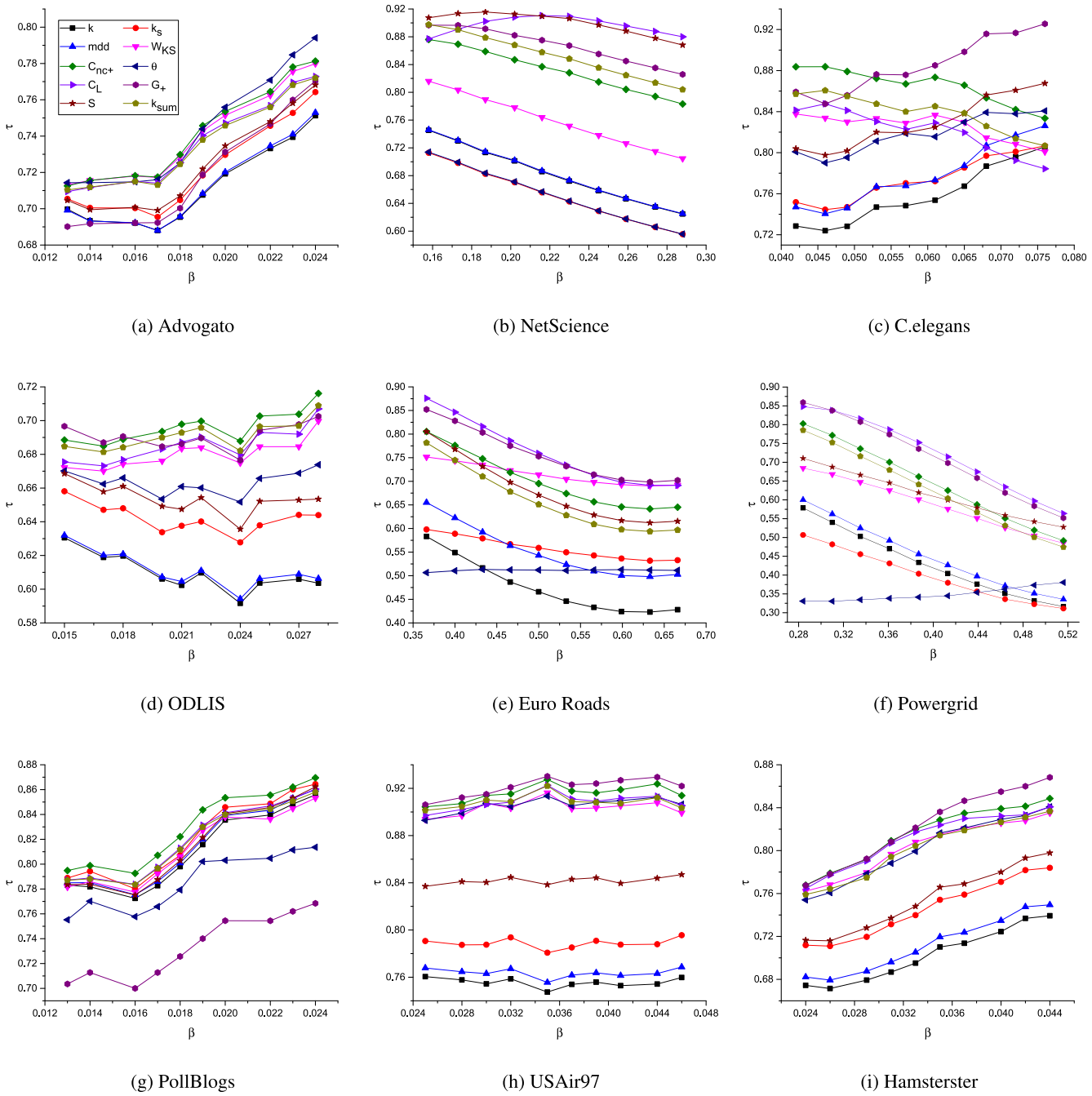


Fig. 3. Kendall's rank correlation (τ) is plotted against different values of infection probability (β) for different node ranking heuristics such as degree (k), k-shell (k_s), mixed degree decomposition (*mdd*), coreness plus (c_{nc+}), the *improved method*(θ), semi local centrality (C_L), gravity plus (G_+), neighbors' degree method (k_{sum}), local gravity model (S) and, weighted k-shell (W_{KS}). Nine different real networks have been used. We have considered β values starting from just above epidemic threshold (β_{th}) to $2 * \beta_{th}$. The results are averaged over 100 independent runs. The objective of this plots is to corroborate the fact that ranking performance of different node ranking heuristics (in terms of Kendall's τ) varies sufficiently with change in infection probability (β) considered.

7.5. Imprecision function

Another commonly used node spreading performance evaluation metric is the *imprecision function*. It is first proposed by Kitsak et al. (2010), further modified by Liu et al. (2015a) and employed in (Wang et al., 2017; Liu et al., 2015b). It's definition is presented in Eq. (29).

$$\epsilon(f) = 1 - \frac{SI(f)}{SI_{eff}(f)} \tag{29}$$

where $SI(f)$ is the average spread with top $f * N$ nodes of any centrality method and $SI_{eff}(f)$ is the average spread with the top $f * N$ nodes with highest spreading efficiency. It measures the difference between the average spreading influence of f fraction of total nodes in a network with the highest centrality and the average spreading influence of the $f * N$ most efficient spreaders (as obtained by SIR simulation). It's ideal value is zero and normally smaller this metric better the performance of the node ranking method.

7.6. Discussion

The SIR model has been used for benchmarking the spreading capabilities of the nodes in a network. A 1000 simulation run has been conducted to arrive at the spreadability of the nodes for ranking. In most of the studies, the same has been followed as a standard practice.

A standard approach to validate a newly proposed method is to employ the proposed heuristic measure to arrive at the ranking of the nodes of some test network, and then, compare that ranking with the SIR model generated ranking. For that purpose, Kendall's rank correlation (τ) is calculated between the ranks generated by the proposed technique and SIR model simulation. Average τ value is checked for all such techniques to evaluate them. The highest correlation with the SIR ranking generally wins.

As SIR model node ranking depends on the value of spreading probability, β, so, as a demonstration of variability, we have varied β from an initial value of β_{th} up to $2 \times \beta_{th}$ and calculated Kendall's τ between different ranking methods and SIR benchmark. The results for our toy network are shown in Table 5. We observe that

Table 6
Comparison of average rank correlation ($\tau_A(.)$) of different variants of k-shell method (such as degree (k), k-shell (k_s), mixed degree decomposition (mdd), potential edge weight based k-shell index (W_{KS}), coreness plus (c_{nc+}), improved method (θ), local centrality (C_L), gravity plus (G_+), neighbors' degree (k_{sum}) and, local gravity model (S)) generated node ranking with SIR model generated ranking of nodes on some real networks. Infection probability β value varying from β_{th} to $2 \times \beta_{th}$. Results are averaged over 100 independent runs.

Network	$\tau_A(k)$	$\tau_A(k_s)$	$\tau_A(mdd)$	$\tau_A(W_{KS})$	$\tau_A(c_{nc+})$	$\tau_A(\theta)$	$\tau_A(C_L)$	$\tau_A(G_+)$	$\tau_A(S)$	$\tau_A(k_{sum})$
NS	0.6812349	0.6510366	0.6820479	0.758613	0.8312353	0.6516207	0.8964502	0.867016	**0.900**	0.8518605
C.elegans	0.7585463	0.7741578	0.7778694	0.8253752	0.865409	0.8179698	0.8213031	**0.886**	0.8290008	0.8390457
PB	0.8112936	0.8213658	0.8148035	0.8142458	**0.830**	0.7863199	0.8203002	0.7334024	0.8156254	0.8190258
Advogato	0.7119119	0.7217211	0.7125507	0.7404688	0.7416785	**0.743**	0.7362418	0.7193697	0.7241935	0.735489
Zachary	0.7077541	0.6410737	0.7216474	0.7800407	0.8864995	0.6678638	0.8762189	**0.891**	0.7712193	0.8778044
PG	0.4304416	0.3984131	0.4516791	0.5867844	0.6446624	0.3489318	**0.723**	0.7124731	0.6134946	0.6252577
Hamsterster	0.7031458	0.7463221	0.7125544	0.8038784	0.8159852	0.8020979	0.8116606	**0.823**	0.7551324	0.8024478
USAirLine	0.7555064	0.7887088	0.7636665	0.9035818	0.9158976	0.9059728	0.9086781	**0.921**	0.8419048	0.9086506
Jazz	0.8425452	0.8097056	0.8667867	0.8203237	0.9293798	0.7858264	**0.936**	0.910608	0.8800549	0.9254286
Dolphins	0.7561803	0.7212152	0.7728876	0.8425284	0.8351992	0.7466427	0.8926943	**0.900**	0.8369008	0.8175672
Euroroad	0.4755009	0.5584701	0.5510954	0.714447	0.700615	0.5113969	**0.761**	0.7561428	0.6793629	0.6591435
Highschool	0.7834875	0.798159	0.815323	0.759741	**0.928**	0.7912368	0.9271262	0.9236217	0.8303686	0.9092211
Foodweb	0.8335075	0.8349139	0.8640829	0.7675559	0.903095	0.8538835	0.9067816	**0.932**	0.8946357	0.9097882
Macaques	0.850797	0.4538755	**0.855**	0.4974019	0.8428174	0.8339237	0.8430404	0.8453558	0.8451557	0.8430404
Football	**0.638**	0.1314262	0.516958	0.2878465	0.6298946	0.3208221	0.6352494	0.6302821	0.6166113	0.6354645
ODLIS	0.6091914	0.6418392	0.611131	0.6803429	**0.696**	0.6633086	0.6857445	0.6905637	0.6532564	0.6913386

Important and maximum values in each row are marked with bold face.

Table 7
Monotonicity (M(.)) of node ranking list generated by different variants of k-shell methods (such as degree (k), k-shell (k_s), mixed degree decomposition (mdd), potential edge weight based k-shell index (W_{KS}), coreness plus (c_{nc+}), improved method (θ), local centrality (C_L), gravity plus (G_+), neighbors' degree (k_{sum}) and, local gravity model (S)) on the toy network and some other selected benchmark real networks.

Network	M (k)	M (k_s)	M (mdd)	M (W_{KS})	M (c_{nc+})	M (θ)	M (C_L)	M (G_+)	$M(S)$	$M(k_{sum})$
NS	0.7069	0.6634	0.7397	0.8570	0.9125	0.6638	0.9147	**0.9167**	0.9166	0.8966
C.elegans	0.9217	0.6094	0.9687	0.7816	0.9975	0.9892	**0.9977**	**0.9977**	**0.9977**	0.9949
PB	0.9324	0.9060	0.9443	0.9655	0.9992	0.9964	**0.9993**	0.5463	0.9987	0.9986
Advogato	0.8445	0.8197	0.8630	0.9813	0.9985	0.9980	**0.9986**	0.8213	0.9977	0.9969
Zachary	0.7079	0.4958	0.7536	0.6870	0.9472	0.8791	**0.9542**	**0.9542**	**0.9542**	0.940
PG	0.5927	0.2460	0.6928	0.6548	0.9419	0.9604	0.9911	0.9991	**0.9999**	0.8866
Hamsterster	0.8860	0.8489	0.9110	0.9799	0.9984	0.9960	0.9987	0.9922	**0.9989**	0.9957
US-Airline	0.8586	0.8114	0.8871	0.9614	0.9945	0.9640	0.9949	**0.9951**	**0.9951**	0.9924
Jazz	0.9659	0.7944	0.9882	0.8258	**0.9993**	0.9345	**0.9993**	0.9971	**0.9993**	0.9981
ODLIS	0.8728	0.8151	0.9152	0.8526	0.9996	0.9969	**0.9999**	0.9995	**0.9999**	0.9968
Dolphins	0.8312	0.3769	0.9041	0.6589	0.9873	0.9737	0.9968	**0.9979**	**0.9979**	0.9550
EURO-Road	0.4442	0.2129	0.6498	0.4970	0.9175	0.9882	0.9840	**0.9962**	0.9947	0.8400
High-School	0.8231	0.5469	0.9110	0.7094	0.9983	0.9712	0.9992	**1.0000**	**1.0000**	0.9819
Foodweb	0.9083	0.6256	0.9092	0.3910	**0.9370**	0.9194	**0.9370**	0.9358	**0.9370**	0.9369
Macaques	0.9324	0.0447	0.8851	0.0716	**1.0000**	0.9263	0.9989	**1.0000**	0.9989	0.9989
Football	0.3637	0.0003	0.6089	0.0101	0.9604	0.9539	0.9991	**1.0000**	**1.0000**	0.9243
Toy	0.4556	0.4900	0.6006	0.7084	0.7511	0.7225	0.8403	**0.8556**	**0.8556**	0.7656

Important and maximum values in each row are marked with bold face.

τ values for the same method on the same network vary widely with changes in β value. For example, with gravity plus centrality (G_+), τ value increased from 0.529 to 0.581, with an increase in β value from 0.341 to 0.62. Again, in the case of coreness plus (c_{nc+}), the correlation τ drops from 0.9125 to a value of 0.7511. Next, we have plotted Kendall's rank correlation of different heuristics with different values of infection probability β for 9 real world networks as shown in Fig. 3. We observe from the plots that for some node ranking method τ values vary significantly with change in β values. Different methods get impacted differently with changing infection probability. It may happen, for some β value method-1 outperforms method-2 on some network, and again, when some other β value is considered method-2 becomes the winner. There are many overlaps in the plots to indicate the above fact. So, we conclude that the choice of infection probability (β) may impact the comparative performance of different methods.

To overcome the above problem, we have used the average Kendall's rank correlation (τ_A) to compare the average performance of different ranking algorithms. The spreading process, as well as node ranking performances, are highly dependent on infection probability. Most of the node ranking heuristics are sensitive to infection probability. Running experiments on a single infection probability may give advantage to some heuristic whereas, may penalize others. A little change in β may yield different results. So, to introduce some robustness to the performance comparison, many authors Wang et al., 2017; Namtirtha et al., 2018; Maji et al., 2020 have employed an average correlation over an important region of infection probability value space for performance comparison. Wang et al. (2017) have used β values almost twice or more than that of epidemic threshold value β_{th}. Liu et al. (2015b) have used β values up to 3 times of β_{th} during experiments. Li et al. (2019) have simulated Kendall's correlation with β varying till $1.4 * \beta_{th}$. Inspired from the above studies, we vary β starting from the epidemic threshold β_{th} to twice of that in ten equal steps and compute average rank correlation over all such β values as defined in Eq. (26). We have taken $\delta = 0.1 * \beta_{th}$ in our experiments for performance comparison. The comparison results are presented in Table 6. We observe from the results that average rank correlation is more with the hybrid methods that use a combination of degree (k), k-shell, and closeness in terms of geodesic distances. Whenever a method uses more than one level of the neighborhood, its performance starts improving. It is prominent

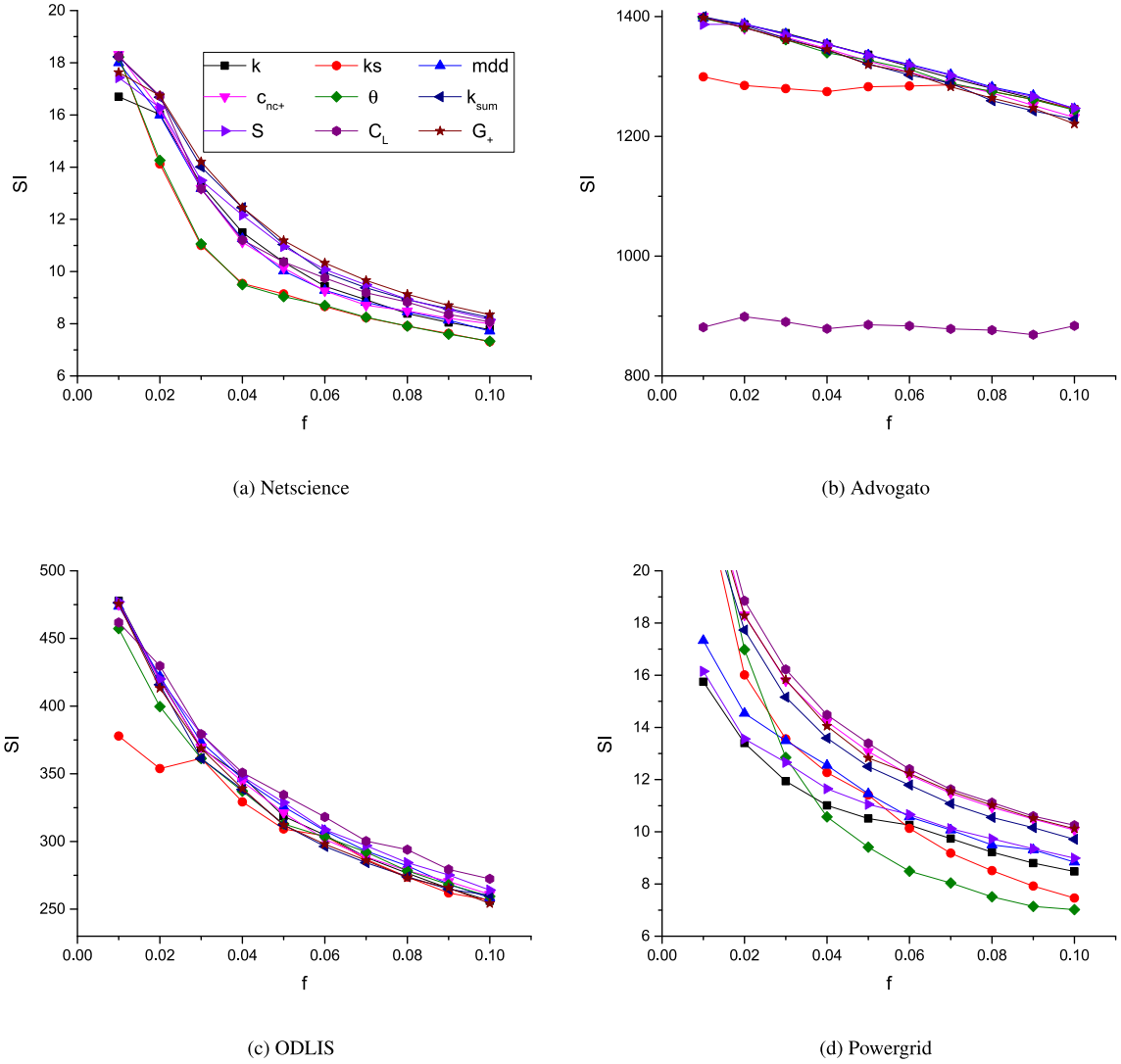


Fig. 4. Average spreading influence (SI) is plotted against different fraction (f) of total nodes taken from the rank list generated by different node ranking heuristics such as degree (k), k-shell (k_s), mixed degree decomposition (mdd), coreness plus (c_{nc+}), the *improved method*(θ), semi local centrality (C_L), gravity plus (G_+), neighbors' degree method (k_{sum}), local gravity model (S) and, weighted k-shell (W_{KS}). Four different real networks have been used. The general trend shows that average influence decreases with the increase of the fraction of nodes. It signifies that top nodes (smaller f) have more spreading influence than lower nodes that are considered with higher fractions.

that more the global information put into the formulation, better the results are but with an associated computational cost. Here, G_+, S, k_{sum} and C_L yields better correlation in comparison to degree centrality alone (k).

Next, we compare the clustering of nodes into a small number of shells, i.e., the uniqueness of ranked items. For that purpose, monotonicity value is calculated for each ranking generated by a heuristic technique. A higher value wins the race. We have calculated this metric as per Eq. (27) and record in Table 7 for our small toy network and some other real networks. Important values are marked with bold face. We like to stress on the point that these calculations are only shown to let the reader know about the used evaluation measures and how to arrive at them. These results (quantitatively) do not convey any indication to compare the different techniques. Actual evaluation and comparison should always be done on many different benchmark real network data set. No single method gives a good result for all the real networks. We have listed some of the most commonly used real networks, that are used for comparison and benchmarking new measures in Table 4 along with some of their standard properties. From the results in Table 7, we observe that the simplest degree centrality (k) yields an almost unique ranking of nodes on some of the networks (eg. Pollblogs, Macaques, celegans), but fails for the others. On the other hand, heuristics that combines closeness or distance from the core along with k-shell index or degree performs much better with consistent higher monotonicity value. The general trend shows a monotonicity value of more than 90% with most of the networks is achieved with hybrid methods such as coreness plus (c_{nc+}), improved method (θ), local centrality (C_L), local gravity model (S) etc. We may conclude that techniques that use neighboring information (geodesic, n-step neighborhood, etc.) along with local information (node degree) are likely to perform better in terms of the uniqueness of the ranked nodes.

Finally, we have analyzed different heuristics based on the average influence of the top nodes by considering the different fraction of top-ranked nodes as an initial seed node. We have presented the average spreading influence (SI) versus the fraction of nodes used as *seed* (f) for four of the selected real networks in Fig. 4. The general trend we observe is that when a small fraction of nodes are used ($f = 0.01$ signifies that only top 1% nodes are used; if network size is N, then top $f * N$ nodes are used to compute average spreading influence by simulating each of them separately using SIR epidemic model), the spreading influence is highest, and it decreases as we consider lower ranked nodes. As a general note, most of the heuristics identify important nodes correctly on average. But, when we consider only the most important node in a network, then their results are not very accurate. Here also, hybrid methods that employ more than a single centrality measure provides a better result.

8. Conclusion

Influential node identification in real networks is an active research area with wide and rich applications. Viral marketing, virus spreading, rumor containment are among some of the most common applications. The k-shell is one of the most important node ranking method used extensively by social network analysts. Many modifications, as well as improvements, have been proposed recently over the classical k-shell method.

In this study, we have aimed at providing a brief guide to new researchers in this domain to quickly grasp the basic essence of the node ranking techniques along with standard evaluation measures and methodology followed. A small set of representative techniques have been discussed to understand the evolution of the classical k-shell method over time to cater to different real network scenarios and to overcome certain limitations. We have kept our discussion limited to only unweighted, undirected network representations and solution techniques though, many of the heuristics have been extended for weighted networks. This study shows the detailed steps and calculations, with the help of a small toy network for most of the discussed techniques to clarify understanding, and also, researchers able to manually verify them with ease. We have considered some standard real networks and applied different discussed techniques on them to rank the nodes based on their influentiality.

We have then utilized some standard evaluation metrics such as Kendall's τ, *monotonicity*, average spreading influence, etc. (discussed in Section 7) for comparing their usefulness with respect to SIR ranking. We observe that heuristics that utilize mainly local structure such as the *degree* or k_{sum} allow for quick and easier computation but offer less competitive outcome. K-shell considers global structure, and identify important nodes well but puts many nodes in a shell hence rank's monotonicity deteriorates. Recent hybrid techniques that utilize more than a single centrality measure, and also incorporate global topology using some kind of "distance/closeness" measure performs much better than others in most of the cases. Hence, we conclude that hybrid methods such as *gravity formula* (G_+), *local gravity model* (S), *coreness plus* (c_{nc+}) are more time consuming and computation heavy but performs better than degree centrality or k-shell. Again, we should acknowledge the fact that there is not a single node ranking heuristic that works well with all networks. Depending on the network structure, topology, connectedness, and many other network properties, some heuristics works better in some network while others are not. The search for an ideal heuristic that will be easy to compute and performs well with most of the networks is still on. There are a lot more studies that are not covered in this study as well as most methods, related to weighted k-shell have also been kept out of scope. All discussions are for static networks, that do not change its topology with time, but most of the real-world networks are dynamic and also, weighted. We plan to focus on weighted k-shell and its derivatives on static as well as dynamically evolving networks in a future study.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Giridhar Maji: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Sharmistha Mandal:** Software, Methodology, Investigation, Writing - review & editing. **Soumya Sen:** Supervision, Resources, Writing - review & editing.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.eswa.2020.113681.

References

Adamic, L. A., & Glance, N. (2005). The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery* (pp. 36–43). ACM.

Ahajjam, S., & Badir, H. (2018). Identification of influential spreaders in complex networks using hybridrank algorithm. *Scientific Reports, 8*, 11932.

Albert, R., Albert, I., & Nakarado, G. L. (2004). Structural vulnerability of the north american power grid. *Physical Review E, 69* 025103.

Alp, Z. Z., & Öğüdücü, Ş. G. (2018). Identifying topical influencers on twitter based on user behavior and network topology. *Knowledge-Based Systems, 141*, 211–221.

Bae, J., & Kim, S. (2014). Identifying and ranking influential spreaders in complex networks by neighborhood coreness. *Physica A: Statistical Mechanics and its Applications, 395*, 549–559.

Bamakan, S. M. H., Nurgaliev, I., & Qu, Q. (2019). Opinion leader detection: A methodological review. *Expert Systems with Applications, 115*, 200–222.

Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*, 509–512.

Bastian, M., Heymann, S., Jacomy, M., et al. (2009). Gephi: an open source software for exploring and manipulating networks. *Icwsm, 8*, 361–362.

Batagelj, V., & Zaversnik, M. (2003). *An o (m) algorithm for cores decomposition of networks*. arXiv preprint cs/0310049.

Batagelj, V., & Mrvar, A. (1998). Pajek-program for large network analysis. *Connections, 21*, 47–57.

Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology, 2*, 113–120.

Bonacich, P. (2007). Some unique properties of eigenvector centrality. *Social Networks, 29*, 555–564.

Borodin, A., Filmus, Y., & Oren, J. (2010). Threshold models for competitive influence in social networks. In *International workshop on internet and network economics* (pp. 539–550). Springer.

Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y., & Shir, E. (2007). A model of internet topology using k-shell decomposition. *Proceedings of the National Academy of Sciences, 104*, 11150–11154.

Castellano, C., & Pastor-Satorras, R. (2010). Thresholds for epidemic spreading in networks. *Physical Review Letters, 105* 218701.

Chen, D., Lü, L., Shang, M. S., Zhang, Y. C., & Zhou, T. (2012). Identifying influential nodes in complex networks. *Physica a: Statistical Mechanics and Its Applications, 391*, 1777–1787.

Coleman, J. S. (1964). *Introduction to mathematical sociology*. London: Free Press Glencoe.

Del Ferraro, G., Moreno, A., Min, B., Morone, F., Pérez-Ramírez, Ú., Pérez-Cervera, L., Parra, L. C., Holodny, A., Canals, S., & Makse, H. A. (2018). Finding influential nodes for integration in brain networks using optimal percolation theory. *Nature Communications, 9*, 2274.

Duch, J., & Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E, 72* 027104.

Freitas, J., Barbastefano, R., & Carvalho, D. (2018). *Identifying influential patents in citation networks using enhanced voterank centrality*. arXiv preprint arXiv:1811.01638.

Gao, C., Lan, X., Zhang, X., & Deng, Y. (2013). A bio-inspired methodology of identifying influential nodes in complex networks. *PloS One, 8* e66732.

Garas, A., Schweitzer, F., & Havlin, S. (2012). A k-shell decomposition method for weighted networks. *New Journal of Physics, 14* 083030.

Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences, 99*, 7821–7826.

Gleiser, P. M., & Danon, L. (2003). Community structure in jazz. *Advances in Complex Systems, 6*, 565–573.

Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters, 12*, 211–223.

González-Bailón, S., Borge-Holthoefer, J., Rivero, A., & Moreno, Y. (2011). The dynamics of protest recruitment through an online network. *Scientific Reports, 1*, 197.

Granovetter, M. (1978). Threshold models of collective behavior. *American Journal of Sociology, 83*, 1420–1443.

Hou, B., Yao, Y., & Liao, D. (2012). Identifying all-around nodes for spreading dynamics in complex networks. *Physica A: Statistical Mechanics and its Applications, 391*, 4012–4017.

Hu, Y., Ji, S., Feng, L., & Jin, Y. (2015). *Quantify and maximise global viral influence through local network information*. arXiv preprint.

Jia, T., Qin, K., & Shan, J. (2014). An exploratory analysis on the evolution of the us airport network. *Physica A: Statistical Mechanics and its Applications, 413*, 266–279.

Joan, M. (2002). *Odlis: Online dictionary of library and information science reitz*. Western Connecticut State University Libraries.

Jung, J., Jin, W., & Kang, U. (2019). Random walk-based ranking in signed social networks: Model and algorithms. *Knowledge and Information Systems*, 1–40.

Kang, H., & Fu, X. (2015). Epidemic spreading and global stability of an sis model with an infective vector on complex networks. *Communications in Nonlinear Science and Numerical Simulation, 27*, 30–39.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika, 18*, 39–43.

Keeling, M. J., & Rohani, P. (2011). *Modeling infectious diseases in humans and animals*. Princeton University Press.

Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 137–146). ACM.

Kempe, D., Kleinberg, J., & Tardos, É. (2015). Maximizing the spread of influence through a social network. *Theory of Computing, 11*, 105–147.

Kendall, M. G. (1945). The treatment of ties in ranking problems. *Biometrika, 33*, 239–251.

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., & Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics, 6*, 888.

Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. In *Proceedings of the ninth annual ACM-SIAM symposium on discrete algorithms, SODA '98* (pp. 668–677). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Kleinberg, J. (2007). Cascading behavior in networks: Algorithmic and economic issues. *Algorithmic Game Theory, 24*, 613–632.

Knight, W. R. (1966). A computer method for calculating kendall's tau with ungrouped data. *Journal of the American Statistical Association, 61*, 436–439.

Konect. (2017). *Hamsterster friendships network dataset – KONECT*. URL:http://konect.uni-koblenz.de/networks/petster-friendships-hamster.

Ley, M. (2002). The dblp computer science bibliography: Evolution, research issues, perspectives. In *International symposium on string processing and information retrieval* (pp. 1–10). Springer.

Li, F., & Du, T. C. (2017). Maximizing micro-blog influence in online promotion. *Expert Systems with Applications, 70*, 52–66.

Li, X., Guo, J., Gao, C., Zhang, L., & Zhang, Z. (2018). A hybrid strategy for network immunization. *Chaos, Solitons & Fractals, 106*, 214–219.

Li, Z., Ren, T., Ma, X., Liu, S., Zhang, Y., & Zhou, T. (2019). Identifying influential spreaders by gravity model. *Scientific Reports, 9*, 8387.

Liu, J. G., Ren, Z. M., & Guo, Q. (2013). Ranking the spreading influence in complex networks. *Physica A: Statistical Mechanics and its Applications, 392*, 4154–4159.

Liu, Y. Y., Slotine, J. J., & Barabási, A. L. (2011). Controllability of complex networks. *Nature, 473*, 167.

Liu, Y., Tang, M., Zhou, T., & Do, Y. (2015a). Core-like groups result in invalidation of identifying super-spreader by k-shell decomposition. *Scientific Reports, 5*, 9602.

Liu, Y., Tang, M., Zhou, T., & Do, Y. (2015b). Improving the accuracy of the k-shell method by removing redundant links: From a perspective of spreading dynamics. *Scientific Reports, 5*, 13172.

Liu, Y., Tang, M., Zhou, T., & Do, Y. (2016). Identify influential spreaders in complex networks, the role of neighborhood. *Physica A: Statistical Mechanics and its Applications, 452*, 289–298.

Li, M., Zhang, R., Hu, R., Yang, F., Yao, Y., & Yuan, Y. (2018). Identifying and ranking influential spreaders in complex networks by combining a local-degree sum and the clustering coefficient. *International Journal of Modern Physics B, 32*, 1850118.

Lü, L., Chen, D., Ren, X. L., Zhang, Q. M., Zhang, Y. C., & Zhou, T. (2016). Vital nodes identification in complex networks. *Physics Reports, 650*, 1–63.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., & Dawson, S. M. (2003). The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology, 54*, 396–405.

Lü, L., Zhang, Y. C., Yeung, C. H., & Zhou, T. (2011). Leaders in social networks, the delicious case. *PloS One, 6* e21202.

Lü, L., Zhou, T., Zhang, Q. M., & Stanley, H. E. (2016). The h-index of a network node and its relation to degree and coreness. *Nature Communications, 7*, 10168.

Maji, G. (2020). Influential spreaders identification in complex networks with potential edge weight based k-shell degree neighborhood method. *Journal of Computational Science, 39* 101055.

Maji, G., Namtirtha, A., Dutta, A., & Malta, M. C. (2020). Influential spreaders identification in complex networks with improved k-shell hybrid method. *Expert Systems with Applications, 144* 113092.

Ma, N., & Liu, Y. (2014). Superedgerank algorithm and its application in identifying opinion leader of online public opinion supernetwork. *Expert Systems with Applications, 41*, 1357–1368.

Ma, Q., & Ma, J. (2017). Identifying and ranking influential spreaders in complex networks with consideration of spreading probability. *Physica A: Statistical Mechanics and its Applications, 465*, 312–330.

Ma, L., Ma, C., Zhang, H. F., & Wang, B. H. (2016). Identifying influential spreaders in complex networks based on gravity formula. *Physica A: Statistical Mechanics and its Applications, 451*, 205–212.

Martinez, N. D., Magnuson, J. J., Kratz, T., & Sierszen, M. (1991). Artifacts or attributes? Effects of resolution on the Little Rock Lake food web. *Ecological Monographs, 61*, 367–392.

Massa, P., Salvetti, M., & Tomasoni, D. (2009). Bowling alone and trust decline in social network sites. In *Proc. int. conf. dependable, autonomic and secure computing* (pp. 658–663).

Moreno, Y., Pastor-Satorras, R., & Vespignani, A. (2002a). Epidemic outbreaks in complex heterogeneous networks. *The European Physical Journal B-Condensed Matter and Complex Systems, 26*, 521–529.

Moreno, Y., Pastor-Satorras, R., & Vespignani, A. (2002b). Epidemic outbreaks in complex heterogeneous networks. *The European Physical Journal B-Condensed Matter and Complex Systems, 26*, 521–529.

Morone, F., & Makse, H. A. (2015). Influence maximization in complex networks through optimal percolation. *Nature, 524*, 65.

Morone, F., Min, B., Bo, L., Mari, R., & Makse, H. A. (2016). Collective influence algorithm to find influencers via optimal percolation in massively large social media. *Scientific Reports, 6*, 30062.
Nåsell, I. (1996). The quasi-stationary distribution of the closed endemic sis model. *Advances in Applied Probability, 28*, 895–932.
Namtirtha, A., Dutta, A., & Dutta, B. (2018). Identifying influential spreaders in complex networks based on kshell hybrid method. *Physica A: Statistical Mechanics and its Applications, 499*, 310–324.
Namtirtha, A., Dutta, A., & Dutta, B. (2020). Weighted kshell degree neighborhood: A new method for identifying the influential spreaders from a variety of complex network connectivity structures. *Expert Systems with Applications, 139* 112859.
Newman, M. E. (2002). Spread of epidemic disease on networks. *Physical Review E, 66* 016128.
Newman, M. E. (2003). The structure and function of complex networks. *SIAM Review, 45*, 167–256.
Newman, M. E. (2005). A measure of betweenness centrality based on random walks. *Social Networks, 27*, 39–54.
Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web*. Technical Report. Stanford InfoLab.
Pastor-Satorras, R., & Vespignani, A. (2001). Epidemic spreading in scale-free networks. *Physical Review Letters, 86*, 3200.
Pei, S., Muchnik, L., Andrade, J. S., Jr, Zheng, Z., & Makse, H. A. (2014). Searching for superspreaders of information in real-world social media. *Scientific Reports, 4*, 5547.
Pei, S., Teng, X., Shaman, J., Morone, F., & Makse, H. A. (2017). Efficient collective influence maximization in cascading processes with first-order transitions. *Scientific Reports, 7*, 45240.
Pittel, B., Spencer, J., & Wormald, N. (1996). Sudden emergence of a giantk-core in a random graph. *Journal of Combinatorial Theory, Series B, 67*, 111–151.
Radicchi, F., & Castellano, C. (2016). Leveraging percolation theory to single out influential spreaders in networks. *Physical Review E, 93* 062314.
Ren, Z. M., Zeng, A., Chen, D. B., Liao, H., & Liu, J. G. (2014). Iterative resource allocation for ranking spreaders in complex networks. *EPL (Europhysics Letters), 106*, 48005.
Rizos, G., Papadopoulos, S., & Kompatsiaris, Y. (2017). Multilabel user classification using the community structure of online networks. *PloS One, 12* e0173347.
Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika, 31*, 581–603.
Saito, K., Nakano, R., & Kimura, M. (2008). Prediction of information diffusion probabilities for independent cascade model. In *International conference on knowledge-based and intelligent information and engineering systems* (pp. 67–75). Springer.
Salavati, C., Abdollahpouri, A., & Manbari, Z. (2019). Ranking nodes in complex networks based on local structure and improving closeness centrality. *Neurocomputing, 336*, 36–45.
Sela, A., Goldenberg, D., Ben-Gal, I., & Shmueli, E. (2018). Active viral marketing: Incorporating continuous active seeding efforts into the diffusion model. *Expert Systems with Applications, 107*, 45–60.
Shao, Z., Liu, S., Zhao, Y., & Liu, Y. (2019). Identifying influential nodes in complex networks based on neighbours and edges. *Peer-to-Peer Networking and Applications, 12*, 1528–1537.
Sheikhahmadi, A., & Nematbakhsh, M. A. (2017). Identification of multi-spreader users in social networks for viral marketing. *Journal of Information Science, 43*, 412–423.
Šubelj, L., & Bajec, M. (2011). Robust network community detection using balanced propagation. *European Physical Journal B, 81*, 353–362.
Takahata, Y. (1991). *Diachronic changes in the dominance relations of adult female Japanese monkeys of the Arashiyama B group. The Monkeys of Arashiyama* (pp. 123–139). Albany: State University of New York Press.
Tang, J., Zhang, R., Wang, P., Zhao, Z., Fan, L., & Liu, X. (2020). A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. *Knowledge-Based Systems, 187* 104833.
Tang, J., Zhang, R., Yao, Y., Yang, F., Zhao, Z., Hu, R., & Yuan, Y. (2019a). Identification of top-k influential nodes based on enhanced discrete particle swarm optimization for influence maximization. *Physica A: Statistical Mechanics and its Applications, 513*, 477–496.
Tang, J., Zhang, R., Yao, Y., Zhao, Z., Chai, B., Li, H., et al. (2019b). An adaptive discrete particle swarm optimization for influence maximization based on network community structure. *International Journal of Modern Physics C (IJMPC), 30*, 1–21.
Teng, X., Pei, S., Morone, F., & Makse, H. A. (2016). Collective influence of multiple spreaders evaluated by tracing real information flow in large-scale social networks. *Scientific Reports, 6*, 36043.
Wang, J., Hou, X., Li, K., & Ding, Y. (2017). A novel weight neighborhood centrality algorithm for identifying influential spreaders in complex networks. *Physica A: Statistical Mechanics and its Applications, 475*, 88–105.
Watts, D. J. (2002). A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences, 99*, 5766–5771.
Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*, 440–442.
Wei, B., Liu, J., Wei, D., Gao, C., & Deng, Y. (2015). Weighted k-shell decomposition for complex networks based on potential edge weights. *Physica A: Statistical Mechanics and its Applications, 420*, 277–283.
Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research, 33*, 452–473.
Zareie, A., & Sheikhahmadi, A. (2018). A hierarchical approach for influential node ranking in complex social networks. *Expert Systems with Applications, 93*, 200–211.
Zareie, A., & Sheikhahmadi, A. (2019). Ehc: Extended h-index centrality measure for identification of users' spreading influence in complex networks. *Physica A: Statistical Mechanics and its Applications, 514*, 141–155.
Zareie, A., Sheikhahmadi, A., & Jalili, M. (2019). Influential node ranking in social networks based on neighborhood diversity. *Future Generation Computer Systems, 94*, 120–129.
Zareie, A., Sheikhahmadi, A., & Jalili, M. (2020). Identification of influential users in social network using gray wolf optimization algorithm. *Expert Systems with Applications, 142* 112971.
Zareie, A., Sheikhahmadi, A., & Khamforoosh, K. (2018). Influence maximization in social networks based on topsis. *Expert Systems with Applications, 108*, 96–107.
Zeng, A., Hu, Y., & Di, Z. (2009). Optimal tree for both synchronizability and converging time. *EPL (Europhysics Letters), 87*, 48002.
Zeng, A., & Zhang, C. J. (2013). Ranking spreaders by decomposing complex networks. *Physics Letters A, 377*, 1031–1035.
Zhang, J. X., Chen, D. B., Dong, Q., & Zhao, Z. D. (2016). Identifying a set of influential spreaders in complex networks. *Scientific Reports, 6*, 27823.
Zhang, D., Wang, Y., & Zhang, Z. (2019). Identifying and quantifying potential super-spreaders in social networks. *Scientific Reports, 9*, 1–11.
Zhang, B., Zhang, L., Mu, C., Zhao, Q., Song, Q., & Hong, X. (2019). A most influential node group discovery method for influence maximization in social networks: A trust-based perspective. *Data & Knowledge Engineering, 121*, 71–87.
Zhang, Y., Zhang, Z., Wei, D., & Deng, Y. (2012). Centrality measure in weighted networks based on an amoeboid algorithm. *Journal of Information & Computational Science, 9*, 369–376.
Zhou, M. Y., Xiong, W. M., Wu, X. Y., Zhang, Y. X., & Liao, H. (2018). Overlapping influence inspires the selection of multiple spreaders in complex networks. *Physica A: Statistical Mechanics and its Applications, 508*, 76–83.
Zhu, C., Wang, X., & Zhu, L. (2017). A novel method of evaluating key nodes in complex networks. *Chaos, Solitons & Fractals, 96*, 43–50.