Locating influential agents in social networks: Budget-constrained seed set selection

Rishav Agarwal[1], Robin Cohen[1], Lukasz Golab[1], and Alan Tsang[2]

[1] University of Waterloo, Canada {rragarwal, rcohen, lgolab}@uwaterloo.ca
[2] National University of Singapore, Singapore, akhtsang@gmail.com

Abstract. The study of information spread in social networks has applications in viral marketing, rumour modelling, and opinion dynamics. Often, it is crucial to identify a small set of influential agents that maximize the spread of information (cases which we refer to as being budget-constrained). These nodes are believed to have special topological properties and reside in the core of a network. We introduce the concept of nucleus decomposition, a clique based extension of core decomposition of graphs, as a new method to locate influential nodes. Our analysis shows that influential nodes lie in the k-nucleus subgraphs and that these nodes outperform lower-order decomposition techniques such as truss and core, while simultaneously focusing on a smaller set of seed nodes. Examining different diffusion models on real-world networks, we provide insights as well into the value of the degree centrality heuristic.

1 Introduction

With the rise of big data tools and platforms, it has become easier to mine social networks. One topic of particular interest is the study of information spread through a network. Finding influential agents is often key, either to stem the spread of harmful content or to facilitate influence maximization for such positive aims as spreading HIV awareness among homeless youths [24] or increasing revenue with viral marketing [4].

Many approaches have been developed to locate influencers and track the spread of their communications to peers. The NP-hard optimization problem known as *influence maximization* [8] looks to find a set of n nodes that, when "activated", can spread information maximally throughout a given network under a given information diffusion model (see Li et al. for a survey of approximate algorithms for influence maximization [12]). Other heuristics consider properties of a given node such as its degree, as well as information about its local graph structure (for example, avoid nodes at the fringes of a graph that have a high degree but weakly connected neighbours [14]).

Two classes of topology-based heuristics to locate influential nodes are centrality based methods and subgraph decomposition methods. Centrality methods consider the degree of a node (degree centrality), the length of shortest paths from a node to all other nodes (closeness centrality), or the number of times a node occurs in the shortest paths (betweenness centrality). By contrast, [10] and [13] argue that less connected but strategically placed nodes may be better candidates for disseminating information. They turn to k-core [20] and k-truss decompositions [5], which identify subgraphs having high degree or many triangles, respectively (details in the next section). Simulation studies have found that k-core methods outperform some centrality based measures [10] and that k-truss methods, in turn, outperform k-core methods [13].

In this paper, we focus on scenarios where organizations have a limited budget to expend when engaging with potential influencers, and thus locating a small seed set of agents is paramount. Our main contribution is the evaluation of a new method for budget-constrained seed set selection based on nucleus decomposition [19]. A k-nucleus is a generalization of graph decomposition methods, and it has been observed that k-nuclei often overlap with the densest parts of k-cores and k-trusses. Using four real datasets, we compare the effectiveness of topology based methods – k-nucleus decomposition, k-truss, k-core and degree centrality – under three popular information diffusion models: Independent Cascade [8], Linear Threshold [7], and Susceptible-Infectious-Recovered (SIR) [15]. We further show that degree centrality, an often ignored heuristic, can perform as well as the nucleus in some cases, as long as sufficiently many high-degree nodes (e.g., as many as there are in a maximal k-nucleus) are selected. This observation is in contrast to prior work that only used the nodes with the *highest* degree in the network as influencers, which was not as effective as using core or truss decomposition [13]. Finally, we show that topology based methods often perform on par with an approximation algorithm that solves the underlying influence maximization problem (IMM [22]). Our analysis enables practitioners to better choose heuristics according to their choice of information diffusion model and to consider k-nucleus decomposition and degree centrality as important algorithms in their arsenal.

2 Methods

We start by describing the methods included in our study, followed by a discussion of the diffusion models. Let $G(V, E)$ be an undirected graph that models the underlying social network with $|V|$ nodes and $|E|$ edges. Let $v \in V$ be a node in G and let $e \in E$ be an edge in G. Finally, let k be a positive integer.

2.1 Graph Decomposition Methods

k-core Decomposition [20]: A $k-$core is a largest connected subgraph of G where each node has degree at least k. Each node $v \in V$ can be assigned a core value $c(v)$ that equals k if v belongs to a $k-$core but not a $(k+1)-$core. Using this concept, the influential nodes are those with the largest value of $c(v)$. To find a k-core subgraph, we repeatedly remove nodes with a degree of less than k and their adjacent edges. Since removing edges reduces the degree of some of the remaining nodes, whenever a node is removed, we decrement the degree of the affected nodes, and we continue until all the remaining nodes have a degree at least k. The time complexity of this method is $O(|V| + |E|)$ since a node or an edge can be removed at most once.

k-truss Decomposition [5]: This method expands on k-core decomposition by considering triangles, i.e., cycles of length 3. A k-truss is a largest subgraph of G where each edge is contained in at least $k-2$ triangles within the subgraph. Each edge e can be assigned a truss number t_e that equals k if e belongs to a k-truss but not a $(k+1)$ truss. Furthermore, the truss number t_v for a node v is equal to the maximum edge truss of the edges adjacent to v. Using this concept, the influential nodes are those with the largest value of t_v. To find a k-truss, we follow a similar methodology as that for k-core decomposition. However, instead of removing nodes directly, we repeatedly remove edges that are not part of at least $k-2$ triangles, and we output the connected components that remain at the end (time complexity $O(|E|^{1.5})$).

k-nucleus Decomposition [19]: This method generalizes k-truss and k-core decomposition by finding subgraphs of *cliques*. Let r and s be two positive integers such that $r < s$. Let K_r be an r-clique, i.e., a clique with r nodes. Intuitively, a k-(r,s)-nucleus is a maximal subset of smaller r-cliques, each of which is part of many larger s-cliques. Formally, let χ be a set of s-cliques K_s in G. Let $K_r(\chi)$ be a set of smaller r-cliques K_r in some $S \in \chi$. The χ**-degree** of an r-clique $u \in K_r(\chi)$ is the number of larger s-cliques in χ that contain u. χ**-connected:** Two K_r, call them u and u', are χ-connected if there exists a sequence of r-cliques $u = u_1, u_2, ..., u_k = u'$ in $K_r(\chi)$ such that for each i, some s-clique $S \in \chi$ contains $u_i \cup u_{i+1}$. Finally, we define a k-(r,s) nucleus as a maximal union χ of s-cliques K_s such that the χ-degree of any r-clique $u \in K_r(\chi)$ is at least k and any r-clique pair $u, u' \in K_r(\chi)$ is χ-connected.

Setting $r = 1$ and $s = 2$ allows us to recover the definition of k-core from k-(r,s) nucleus. To see this, observe that any node is a 1-clique, and any edge is a 2-clique. Thus, the χ degree of a 1-clique is the degree of the node, and, by the χ-connected property, we simply get a set of edges connecting nodes of degree at least k. Similarly, setting $r = 2$ and $s = 3$ reduces to k-truss decomposition. Triangles are 3-cliques, and we get a set χ that is part of at least k triangles.

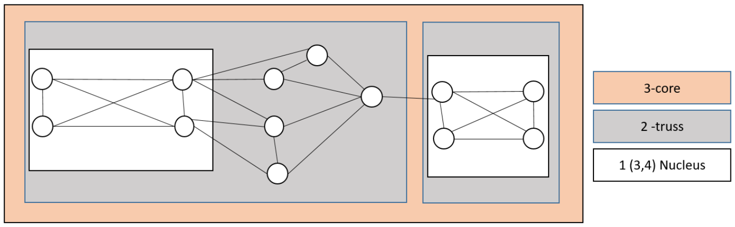
Let $RT(K_r)$ and $RT(K_s)$ be the time complexity of enumerating all $K_r \in G$ and all $K_s \in G$, respectively. The complexity of nucleus decomposition was shown to be bounded by $O(RT(K_s) + RT(K_r))$ [19]. In this paper, we consider k-$(3,4)$ nucleus decomposition as complexity grows rapidly for K_4 and above. From now on, we refer to a k-$(3,4)$ nucleus as k-nucleus for simplicity[3]. As in k-core and k-truss decomposition, each node can be assigned a nucleus value $n(v)$ that equals k if v belongs to a k-nucleus but not a $(k+1)$-nucleus. The influential nodes are those with the largest value of $n(v)$. Some nodes of a $k+1$ core are part of a k-truss and some nodes of a $k+1$-truss are part of a nucleus. Figure 1 illustrates a graph and the corresponding 3-core, 2-trusses and 1-nuclei. The entire graph is a maximal 3-core (as each node has at least three edges). In the 3-core, there are two 2-trusses and two 1-nuclei. Note that the nuclei and trusses are smaller and identify denser subgraphs than the core.

2.2 Information Diffusion Models

Independent Cascade (IC) Model [8]: In this model, nodes that are *activated* can influence their neighbours. Activation proceeds one step at a time. Each *directed* edge $(v, v') : v \to v'$ in the underlying graph has a threshold value $p_{v,v'} \in [0,1]$ denoting the propagation probability of information from v to v'. We begin with a set of nodes that are initially assumed to be active. The information then flows as follows. At time t, any active node $v \in V$ has a chance to activate an inactive child node v' with probability

[3] [19] showed that $(3,4)$-nucleus provides high-quality outputs in terms of density and network hierarchy; e.g., it finds both small sets of high density and large sets of low density.

Fig. 1. Comparison of subgraph decompositions



$p_{v,v'}$. If v succeeds then v' becomes active in step $t+1$. If multiple parents of v' are active at the same time, their activation attempts are arbitrarily sequenced at time t. v only gets one chance to activate v' and cannot activate v' in subsequent rounds. The process terminates when no more activations are possible.

Linear Threshold Model (LT) [7]: In this model, a node is influenced by every incoming neighbour v' with a weight $b_{v',v} \in [0,1]$. Each $v \in V$ also has a threshold $\theta_v \in [0,1]$, which represents the minimum pressure that has to be exerted on v to activate it. v is activated iff the sum of the weights of the active neighbours of v is greater than a threshold θ_v: $\sum_{v' \rightarrow v, v' \text{active}} b_{v',v} \geq \theta_v$. The information flow proceeds in discrete steps (from $t=0$), with a seed set of active nodes S. For each neighbour v of $v' \in S \subseteq V$, we check the threshold condition. If a node satisfies its condition, it is activated in the next step. The algorithm continues until no more activations occur.

Susceptible Infected Recovered (SIR) Model [9]: In this model, a node can be in one of three states: Susceptible (S): not yet infected; Infected (I): can spread information to the rest of the population; Recovered (R): after a node has been infected for some period of time, it is considered to be immune and cannot further spread the information. To examine the spreading power of a set of nodes, we initially set these nodes as infected, and we set all other nodes as susceptible. Then, at each time step t of the process, every infected node can infect its susceptible neighbours with probability β (called infection rate), and afterwards, it can recover with probability γ (called recovery rate). A node cannot directly pass from state I to state R during the same time step. The process ends when no more nodes can be infected.

3 Results

We now explore the performance of k-core, k-truss, k-nucleus and degree centrality in locating influential nodes using four real-world social networks. We start by showing that k-nucleus decomposition identifies fewer nodes as being influential. We then show that despite being lower in number, the nucleus nodes have similar or better information spreading power than those identified by other methods. We also show that the nodes selected by nucleus decomposition are robust to low diffusion rates under the IC and SIR models. Finally, we show that choosing a sufficient number of high degree nodes can work well as the nucleus, and these topology based methods often perform on par with an approximation algorithm that solves the underlying influence maximization problem.

Table 1. Properties of datasets and their subgraph decompositions.

Dataset	Nodes	Edges	τ	k_{max}			v_{max}		
				Nucleus	**Truss**	**Core**	**Nucleus**	**Truss**	**Core**
WikiVote	7,115	88,750	0.00720	15	19	49	37	80	332
Slashdot	81,871	545,671	0.00074	26	34	54	65	77	118
Epinions	131,828	841,372	0.00540	97	105	121	112	135	149
EuEmail	265,214	420,045	0.00970	13	18	37	56	62	292

3.1 Datasets

We use four publicly available datasets [11]. **Slashdot** is a technology news website whose users form a signed social network as they can tag each other as “friend” or “foe”. **Epinions** is a trust-based (who-trusts-whom) network between members of the Epinions.com product review website. **WikiVote** is a dataset showing who voted for whom in the Wikipedia election for membership. **EuEmail** is a “who talked to whom” network with an edge between two nodes, A and B, meaning A sent an email to B.

To compute subgraph decompositions, we use the code from [19], available at `http://sariyuce.com/nucleus-master.zip`. Table 1 summarizes the datasets and the properties of the corresponding k-core, truss and nucleus decompositions (v_{max} is the number of nodes in the maximal subgraph of order k_{max}). The dataset statistics we report are the number of nodes and edges, and the inverse of the largest eigenvalue of the corresponding adjacency matrix, τ. It is known that epidemic spreading can be achieved by setting the propagation probabilities to be at least τ [3]. Below this threshold, the number of affected nodes decreases exponentially.

3.2 Subgraph Decomposition Properties

For the subgraph decompositions, Table 1 shows the largest values of k, denoted as k_{max}, that gave a non-empty core, truss and nucleus, and the number of nodes that were identified as influential, i.e., the number of nodes belonging to the k_{max}-cores, trusses and nuclei, denoted by v_{max}. As shown in Table 1, the number of maximal nucleus nodes (v_{max}) is smaller than the number of maximal truss nodes, which is smaller than the number of maximal core nodes. This is expected as truss decomposition relies on triangles and nucleus decomposition relies on cliques, which are increasingly stricter criteria. Furthermore, Table 2 reports the overlap between influential nodes identified by the different decompositions. Many nodes are common among the three decompositions. In fact, the entire k_{max}-nucleus is often a subset of a k_{max}-truss or k_{max}-core subgraph. This was also seen in our illustrative example in Figure 1.

3.3 Analyzing Trust

Two datasets, Epinions and Slashdot, contain ground truth about who trusts whom in the network. This allows us to explore the contextual properties of our subgraphs. These two graphs have directed edges with binary edge weights: An edge from A to B has a weight of one if A marks B as a “friend” and zero if A marks B as a “foe”. Only

Table 2. Overlap among various selected sets: N is the maximal set of Nucleus nodes, T is the maximal set of Truss nodes, C is the maximal set of Core nodes, D is the set of top 100 Degree centrality nodes, and $I_{IC,\tau}$ is the set of nodes (number of nodes equal to the size of N) found by IMM under the IC model at τ. The left table is the percentage overlap with N and the right is the percentage overlap with D.

Dataset	$N \cap T$	$N \cap C$	$N \cap T \cap C$
WikiVote	100%	94.6%	94.6%
Slashdot	96.9%	96.9%	96.9%
Epinions	100%	100%	100%
EuEmail	71.4%	100%	71.4%

Dataset	$C \cap D$	$T \cap D$	$N \cap D$	$I_{IC,\tau} \cap D$
Wikivote	19%	15%	38.8%	5.4%
Slashdot	5.9%	3.9%	24.6%	23.1%
Epinions	0%	0%	0%	29.5%
Euemail	6.8%	3.2%	33.9%	53.57%

Table 3. Average trust metrics for Slashdot and Epinions.

Dataset	**Subgraph**	**Trusted by**	**In Degree**	**Out Degree**	**Reputability (in %)**
Slashdot	Whole	5.159	6.665	6.665	77.4
	core	176.0	186.4	180.9	95.0
	truss	180.7	191.2	185.5	96.2
	nucleus	183.3	191.1	194.1	96.8
Epinions	Whole	5.444	6.382	6.392	49.0
	core	177.4	183.1	239.3	96.9
	truss	182.3	188.5	245.3	96.8
	nucleus	191.5	197.4	254.3	96.9

15 percent of edges in Epinions are foe edges, and 23 percent of edges in Slashdot are foe edges. We assume that individuals trust their friends but not their foes. This is important in the context of influence maximization because, in practice, influential people are generally those who are trusted by others.

Table 3 presents the following statistics for the entire graphs and for their respective maximal core, truss and nuclei: the average number of nodes that trust a given node, the average node in and out degrees, and the average node *reputability*, defined as the percentage of nodes who trust the given node v and the node v's in-degree. We see that higher-order decompositions are more densely connected and have higher reputability (on average), reinforcing our belief that subgraph decomposition identifies topologically and contextually essential nodes. As users often rate things they like or not rate at all [17], being connected to more people makes one more likely to be positively rated and may have a cascading effect on reputability. The influential nodes identified by nucleus decomposition have the highest reputability in Slashdot, whereas in Epinions, all three tested methods have similar reputability scores.

3.4 Evaluating Spreading Performance

We now evaluate spreading effectiveness using the three information diffusion models. We test k-core, truss and nucleus decomposition as well as the *Degree Centrality* method for selecting the seed set. For degree centrality, we take the top-n highest degree centrality nodes, where n is the number of nodes in a maximal nucleus. While [13] use only the nodes having the highest degree as the seed set, we found that the number of highest-degree nodes can often be too small to be of any practical significance.

Furthermore, we compare our methods to the IMM algorithm [22], which is an approximation algorithm to solve the underlying NP-hard influence maximization problem.

Given the desired number of seed nodes, IMM identifies the (approximately) best such nodes given the underlying diffusion model. As we did in the degree centrality method, we set the desired number of seed nodes to be the number of nodes in a maximal nucleus.

We also note that previous work often used undirected versions of datasets. However, an undirected edge means that "if A trusts B, B also trusts A." This reciprocal behaviour may not always be true, which may affect the efficacy of the diffusion process. Thus, we use directed graphs in our simulations. The experimental setup for the three diffusion models is given below:

- **Independent Cascade Model:** We draw propagation probabilities from a uniform distribution. We set the propagation probability, or **activation rate**, of an edge (v, v') to $p_{v,v'} = u(0, t)$, where $u(0, t)$ is a uniform function between 0 and t for some $t \in (0, 1]$. We choose the uniform distribution as we do not have complete knowledge about users' propagation probabilities [4]. We limit the propagation probabilities to t and use this as a parameter for our experiments. For instance, a low value of t means that nodes are not easily influenced and thus can be thought of as low-trust networks. We start with $t = \tau$ as per Table 1.
- **Linear Threshold Model:** We set the activation thresholds based on a uniform distribution as we do not know the real thresholds for the nodes. Thus, $\theta_v = u(0, 1)$ where $u(0, 1)$ is a uniform function.
- **SIR Model:** We set the infection rate to be the threshold τ (see Table 1) and the recovery rate to be 0.08 as suggested in [13].

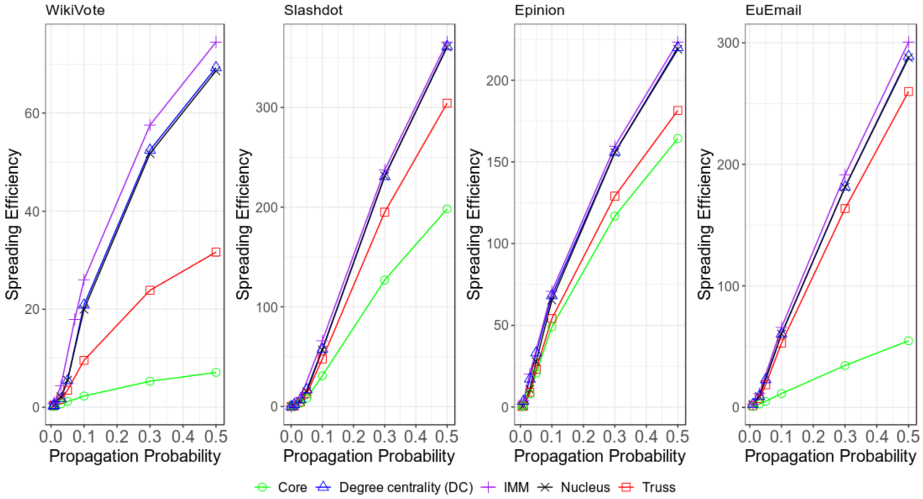
Table 4. Average Spreading performance (number of nodes activated per seed set node). Note: for SIR and IC, we use threshold τ.

Dataset	Subgraph	LT	SIR	IC
WikiVote	Core	7.85	1.15	0.16
	Truss	14.99	1.15	0.30
	Nucleus	17.65	2.00	0.35
	DC	28.44	0.24	0.25
	IMM	28.66	3.34	0.72
Slashdot	Core	47.6	0.68	0.04
	Truss	71	0.24	0.05
	Nucleus	87.46	1.12	0.05
	DC	79.42	1.08	0.05
	IMM	85.2	1.27	0.06

Dataset	Subgraph	LT	SIR	IC
Epinions	Core	8.23	6.01	0.32
	Truss	9.79	6.50	0.36
	Nucleus	11.01	8.47	0.43
	DC	38.2	14.41	1.13
	IMM	68.39	7.16	1.76
EuEmail	Core	9.45	2.82	0.79
	Truss	58.21	3.15	1.39
	Nucleus	78.27	6.96	2.48
	DC	74.71	4.02	2.46
	IMM	76.67	9.48	2.52

For each trial, we activate all the nodes that were reported by a given method as being influential. We then run the diffusion process until no new nodes can be activated.

Fig. 2. Impact of propagation probabilities on spreading performance (number of nodes activated per seed set node) for IC model. Note: the standard errors were too small to report (standard error < 0.15).



We repeat the simulation $1,000$ times and report the average number of nodes activated divided by the seed set size. This metric is our **spreading efficiency**.

Table 4 shows the results. We present the spreading performance for IC and SIR at the minimum threshold chosen (τ). We see that the absolute value of the final spreading efficiency is low for SIR and IC. This is because the reported values are at the lowest threshold we tested. Furthermore, we see that both Degree Centrality (DC) and nucleus decomposition performs better than core and truss decompositions, even at low thresholds. In some cases, DC outperforms the tested graph decomposition methods.

Impact of Propagation Probabilities We now illustrate the impact of propagation probabilities in the Independent Cascade (IC) Model. We focus on the IC model as SIR has been discussed in depth in [13], and SIR reduces to IC when the propagation probability is the same for all nodes (called infection rate for SIR).

For each method and dataset, Figure 2 shows the number of activated nodes for the following propagation probabilities, shown from left to right: τ, 0.01, 0.03, 0.05, 0.1 and 0.5 (recall that τ is the inverse of the largest eigenvalue of the corresponding adjacency matrix and gives a reasonable lower bound for the propagation probability threshold). We stop at 0.5 as information spread tends to saturate at some propagation probability threshold, typically ≈ 0.5. As expected, the number of activated nodes increases for all methods as the propagation probability increases. Second, as nucleus decomposition starts with a smaller seed set, nucleus nodes have better per-node spreading efficiency on average. The degree centrality nodes also perform on par with the nucleus nodes.

Comparison with IMM Algorithm The methods we considered so far select influential nodes based on graph properties. In contrast, IMM selects influential nodes by solving the underlying influence maximization problem approximately to $1 - 1/e - \epsilon$, where ϵ controls the approximation (a higher value trades solution quality for runtime). Since we randomly assign propagation probabilities in our simulations, we found that different experimental runs of IMM on the same graph gave different influential nodes. To account for this, for each experiment, we run IMM 100 times and output the nodes that were most frequently identified as influential over the 100 runs. (Again, as mentioned earlier, the total number of influential nodes we select is equal to the number of nodes in a maximal nucleus.) IMM requires the user to set ϵ, which we set to 0.1 following prior work [22].

From Figure 2 and Table 4, we conclude that IMM has the best average spreading performance in many situations, and is nearly as good as nucleus decomposition for Slashdot and EuEmail. Our results align with those from [1], which shows that there is no single state of the art technique in Influence Maximization. We also note that IMM took about 1500 seconds per iteration for the Epinions dataset (IC model, 0.5 threshold), while nucleus decomposition took 126 seconds, truss took 5.2 seconds, and core took 0.2 seconds. The runtimes on all the datasets are shown in Table 5. As we explained above, we ran IMM 100 times to arrive at a "stable" set of seed nodes, giving a total runtime of over 41 hours. Moreover, the memory footprint of IMM is high ($>$ 30GB of RAM for the EuEmail dataset). Thus, nucleus decomposition may be the algorithm of choice for practitioners willing to sacrifice some effectiveness for much faster runtime.

Comparison with Degree Centrality According to Figure 2 and Table 4, DC performs better than nucleus decomposition in some cases. In Table 2, we see that the top 100 nodes ranked by degree centrality contain $\sim 20\%$ of nucleus nodes. Interestingly, there is no overlap between the top degree centrality nodes and subgraphs for Epinions. One possible reason for this could be the sparse nature of the Epinions graph, as seen in Table 1. We also see that DC has a high overlap with the nodes found by IMM, indicating that the optimal nodes chosen by IMM often have high degree as well.

Table 5. Runtimes in seconds. For IMM, we report the time at threshold 0.5 for IC

Dataset	Core	Truss	Nucleus	IMM
WikiVote	0.16	0.35	4.1	83.13
Slashdot	0.10	0.81	5.2	373.8
Epinions	0.19	4.00	126.2	1275.2
EuEmail	0.11	0.38	2.0	786.7

4 Discussion

Identifying influential nodes that can disseminate information to a large part of a network is of particular interest in social network research. k-core, a subgraph decomposition based on maximal node degrees, has mainly been studied in this context and found to be effective [10] [15]. However, even k-core can overlook critical features in the graph, motivating the use of a higher-order decomposition called k-truss [13].

The promising results using k-truss decomposition motivated us to consider even more dense substructures, and we arrived at a generalized notion of nucleus decomposition [19]. By imposing more restrictions on nodes in terms of topological and positional factors, our experiments reveal that nucleus decomposition significantly reduces the number of candidates for influential spreaders. This means that in practice, fewer nodes need to be engaged to obtain similar spreading performance. Nucleus decomposition can be used in conjunction with other influence maximization algorithms to reduce their search space for even better results. For example, one may start with a set of nucleus nodes and choose a subset of them with the highest degree. However, as seen in the experimental results using the Epinions dataset, nucleus decomposition may not always produce the most influential nodes.

We explored three models for information spread in order to gauge the performance of topology-based IM methods. The Linear Threshold (LT) model is sensitive to the number of neighbours that can influence a node: a large number of neighbours make it more likely for information to spread. Thus, Degree Centrality works well in the LT model. The SIR and Independent Cascade (IC) models use propagation probabilities that may be different for different node pairs. In these cases, nucleus decomposition performs better as it tends to identify strategically placed nodes. We notice that Degree Centrality often performs on par with nucleus decomposition, especially for higher propagation probabilities. This may be due to the high overlap between the nucleus and high degree-centrality nodes, as seen in Table 2. Furthermore, we found that in many situations, DC and nucleus decomposition performs similar to IMM in terms of average spreading performance. However, they take much less time to be computed. This suggests the benefits of topology-based methods such as nucleus decomposition compared to approaches that solve the underlying influence maximization problem.

Modeling and optimizing influence spread has garnered much interest from multiagent systems researchers. Some researchers in this field contend that graph properties may be prone to an error in predictions of information spread, and that actual behaviour in certain networks, especially ones of more modest sizes (e.g., for homeless youth HIV prevention [23]) may play out differently, integrating more connection to those currently outside the network. These authors also advocate considering the set of seed nodes as a multiagent team, with inter-connections. We view the work in this paper as complementary to these research threads in the multiagent systems domain. For one, if it is indeed critical to be examining relationships between the nodes in the seed set, this can be done all the more effectively if operating with a smaller set of nodes, the behaviour of which can be examined in detail. It is also possible to use nucleus decomposition together with other models of diffusion, which are more generous to the integration of external nodes.

5 Conclusions and Future Extensions

This work provides vital new insights into how to track influence within social networks when operating with constrained resources, revealing the effectiveness of smaller seed sets, of use for a host of applications. Calibrating the value of k-nucleus is an important part of our effort. Recall that truss and core can be thought of as $(2, 3)$ and $(1, 2)$-nuclei, respectively. A $(3,4)$-nucleus is an even denser subgraph with fewer nodes that have the potential to exhibit good spreading power. While there is a marked improvement, there will be a diminishing return on computation time investment on successively mining denser subgraphs (as noted by [19]). This opens several avenues for future work.

A limitation of subgraph mining methods is that they usually consider undirected graphs, and thus some information may be lost. A potential solution is to identify d-cores [6], which separately consider the in-degree and out-degree of nodes and thus may be more suitable for directed graphs. Furthermore, other graph decomposition based approaches have been proposed, such as k-meanoid [25] and modified k-shell [2]. A comparison of k-nucleus against these methods would be an ideal next step.

Various empirical studies to date have provided insights into the theoretical advantages of different algorithms (e.g. [8]); for future work, it would be valuable to expand these kinds of discussions to nucleus decomposition. Additionally, since nucleus decomposition gives a verifiably smaller set of nodes with better-spreading properties than other methods such as core and truss, it can also be used as a preprocessing step for optimal algorithms. Moreover, there are now parallelized algorithms available for nucleus decomposition, which can improve the efficiency of our approach [18].

Interestingly, the work of [15] on tracking real-world information flow found that the most influential nodes lie in the k-core subgraph, and it would be valuable to show that they lie in the nucleus or truss subgraphs as well. It would also be interesting to empirically compare subgraph based methods with the greedy algorithm of [8]. [16] assigns a “Klout Score” of influence to 750 million users by extracting features from user interactions in multiple social networks and then aggregating them into a hierarchical scoring structure. Combining these content-based methods with k-nucleus decomposition is another potential direction for future work.

One final avenue for future work is to experiment with something other than a uniform propagation probability distribution, determining which scenarios benefit from lifting that assumption. Examining other methods for measuring influence would also be valuable (for example, [21] uses social media posts to create a content based metric).

References

1. Arora, A., Galhotra, S., Ranu, S.: Debunking the myths of influence maximization: An in-depth benchmarking study. In: ACM International Conference on Management of Data. pp. 651–666 (2017)
2. Brown, P.E., Feng, J.: Measuring user influence on twitter using modified k-shell decomposition. In: Fifth international AAAI conference on weblogs and social media (2011)
3. Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., Faloutsos, C.: Epidemic thresholds in real networks. ACM Transactions on Information and System Security **10**(4), 1 (2008)
4. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1029–1038. ACM (2010)
5. Cohen, J.: Trusses: Cohesive subgraphs for social network analysis. National Security Agency Technical Report **16** (2008)
6. Giatsidis, C., Thilikos, D.M., Vazirgiannis, M.: D-cores: Measuring collaboration of directed graphs based on degeneracy. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on. pp. 201–210. IEEE (2011)

7. Granovetter, M.: Threshold models of collective behavior. American journal of sociology **83**(6), 1420–1443 (1978)
8. Kempe, D., Kleinberg, J., Tardos, É.: 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 (2003)
9. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character **115**(772), 700–721 (1927)
10. Kitsak, M., Gallos, L.K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H.E., Makse, H.A.: Identification of influential spreaders in complex networks. Nature physics **6**(11), 888 (2010)
11. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data (2014)
12. Li, Y., Fan, J., Wang, Y., Tan, K.L.: Influence maximization on social graphs: A survey. IEEE Transactions on Knowledge and Data Engineering **30**(10), 1852–1872 (2018)
13. Malliaros, F.D., Rossi, M.E.G., Vazirgiannis, M.: Locating influential nodes in complex networks. Scientific reports **6**, 19307 (2016)
14. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. pp. 29–42. ACM (2007)
15. Pei, S., Muchnik, L., Andrade Jr, J.S., Zheng, Z., Makse, H.A.: Searching for superspreaders of information in real-world social media. Scientific reports **4**, 5547 (2014)
16. Rao, A., Spasojevic, N., Li, Z., Dsouza, T.: Klout score: Measuring influence across multiple social networks. In: Big Data, IEEE International Conference. pp. 2282–2289 (2015)
17. Sardana, N., Cohen, R., Zhang, J., Chen, S.: A bayesian multiagent trust model for social networks. IEEE Transactions on Computational Social Systems **5**(4), 995–1008 (2018)
18. Sariyüce, A.E., Seshadhri, C., Pinar, A.: Local algorithms for hierarchical dense subgraph discovery. Proceedings of the VLDB Endowment **12**(1), 43–56 (2018)
19. Sariyüce, A.E., Seshadhri, C., Pinar, A., Çatalyürek, Ü.V.: Nucleus decompositions for identifying hierarchy of dense subgraphs. ACM Transactions on the Web **11**(3), 16 (2017)
20. Seidman, S.B.: Network structure and minimum degree. Social networks **5**(3), 269–287 (1983)
21. Sun, B., Ng, V.T.: Identifying influential users by their postings in social networks. In: Ubiquitous social media analysis, pp. 128–151. Springer (2013)
22. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: A martingale approach. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. pp. 1539–1554. ACM (2015)
23. Wilder, B., Onasch-Vera, L., Hudson, J., Luna, J., Wilson, N., Petering, R., Woo, D., Tambe, M., Rice, E.: End-to-end influence maximization in the field. In: 17th International Conference on Autonomous Agents and MultiAgent Systems. pp. 1414–1422 (2018)
24. Yadav, A., Wilder, B., Rice, E., Petering, R., Craddock, J., Yoshioka-Maxwell, A., Hemler, M., Onasch-Vera, L., Tambe, M., Woo, D.: Bridging the gap between theory and practice in influence maximization: Raising awareness about hiv among homeless youth. In: IJCAI. pp. 5399–5403 (2018)
25. Zhang, X., Zhu, J., Wang, Q., Zhao, H.: Identifying influential nodes in complex networks with community structure. Knowledge-Based Systems **42**, 74–84 (2013)