OPEN ACCESS Freely available online

PLOS ONE

Finding Near-Optimal Groups of Epidemic Spreaders in a Complex Network

Geoffrey Moores[1,3], Paulo Shakarian[1,3]*, Brian Macdonald[2,3], Nicholas Howard[2,3]

1 Electrical Engineering and Computer Science Department, United States Military Academy, West Point, New York, United States of America, **2** Mathematical Science Department, United States Military Academy, West Point, New York, United States of America, **3** Network Science Center, United States Military Academy, West Point, New York, United States of America

Abstract

In this paper, we present algorithms to find near-optimal sets of epidemic spreaders in complex networks. We extend the notion of local-centrality, a centrality measure previously shown to correspond with a node's ability to spread an epidemic, to sets of nodes by introducing combinatorial local centrality. Though we prove that finding a set of nodes that maximizes this new measure is NP-hard, good approximations are available. We show that a strictly greedy approach obtains the best approximation ratio unless P = NP and then formulate a modified version of this approach that leverages qualities of the network to achieve a faster runtime while maintaining this theoretical guarantee. We perform an experimental evaluation on samples from several different network structures which demonstrate that our algorithm maximizes combinatorial local centrality and consistently chooses the most effective set of nodes to spread infection under the SIR model, relative to selecting the top nodes using many common centrality measures. We also demonstrate that the optimized algorithm we develop scales effectively.

Citation: Moores G, Shakarian P, Macdonald B, Howard N (2014) Finding Near-Optimal Groups of Epidemic Spreaders in a Complex Network. PLoS ONE 9(4): e90303. doi:10.1371/journal.pone.0090303

Editor: Jesus Gomez-Gardenes, Universidad de Zarazoga, Spain

Received July 7, 2013; **Accepted** February 3, 2014; **Published** April 2, 2014

Funding: This work is funded by the Army Research Office. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: paulo@shakarian.net

Introduction

In this paper we look to find *optimal* sets of individuals in a complex network to initiate an epidemic. Addressing such a problem will have clear implication in seeding a social network to ensure a given phenomenon diffuses optimally and may also provide insight into mitigation strategies against an infection initiated by a group of individuals. Further, this problem is non-trivial. For instance, it has previously been noted in [1] that selecting a second influential node, or 'spreader,' does not always significantly increase the spread of the epidemic. In [2], the authors show that identifying an optimal set of spreaders under a more generalized epidemic model is NP-hard.

The susceptible-infected-recovered (SIR) model [3] is one of the most well-studied models of epidemic disease spread in a population. In this model, individuals in a population are in one of three states: *susceptible* individuals can acquire the disease from *infected* individuals who after a certain amount of time become *recovered* and can no longer transmit or acquire the disease. In recent studies, there has been much interest in studying this model on populations structured as a network [4–6].

Throughout this paper, we will assume a population structured as an undirected network $G=(V,E)$ where V is a set of individuals ("nodes") and $E \subseteq V \times V$ where $(v,v') \in E$ implies $(v',v) \in E$. The size of V and E are denoted n and m respectively. For other sets of elements, we shall use the notation $|\cdot|$ to denote the size of that set. For a given node $v \in V$, η_v is the set of neighbors, formally $\{v' \in V | (v,v') \in E\}$. We will extend this to sets: for set V', $\eta_{V'} = \bigcup_{v \in V'} \eta_v$. We will use k_v to denote the degree of v which is the cardinality of η_v. The we denote the average and maximum degree of any node in the graph as $\langle k \rangle$ and k^* respectively. We note that, in most real-world social networks, $k^* \ll n$. The quantity N_v is the number of neighbors and next-nearest neighbors of node v and is defined formally as follows:

$$N_v = |\eta_v \cup \{\cup_{i \in \eta_v} \eta_i\}| \quad (1)$$

In this paper, we use a the version of the SIR model specified by Chen et al. [7]. Nodes in the network under this version of the model are in one of three states: susceptible, infected, or recovered. Once a node is infected, one of its neighbors then becomes infected at random (by a uniform probability over the neighbors of the initially infected node). After infecting a neighbor, the node then recovers with a probability $\frac{1}{\langle k \rangle}$.

Chen et al. accurately identified individual spreaders with Local Centrality. For a given node v, its Local Centrality, $C_L(v)$ is defined as follows:

$$C_L(v) = \sum_{u \in \eta_v} \sum_{w \in \eta_u} N_w \quad (2)$$

We extend Local Centrality with a set centrality-based technique similar to that of [8,9]. We then frame an optimization problem that seeks to find k of nodes in the network that together optimize our extended version of local-centrality. For some set

$V' \subseteq V$ with *combinatorial local centrality*, denoted $C_{LC}(V')$, is defined as follows:

$$C_{LC}(V') = \sum_{u \in \eta_{V'}} \sum_{w \in \eta_u} N_w \quad (3)$$

Figure 1 demonstrates N_v, C_L, and C_{LC} on a small, arbitrary network.

Using this definition, we now present the problems we wish to study in this paper which deal with finding a set of nodes (of sized K) that optimizes the above function.

Definition 1. *Max Combinatorial Local Centrality Problem (Max C_{LC}):*

INPUT: $K < n$;

OUTPUT: $V' \subseteq V$, *s.t.* $|V'| \leq K$ *and* $= \nexists\ V'' \subseteq V$ *s.t.* $|V''| \leq K$ *where* $C_{LC}(V') < C_{LC}(V'')$

Definition 2. *Combinatorial Local Centrality Decision Problem (Dec C_{LC})*

INPUT: $K < N$; X

OUTPUT: *Yes if* $\exists V' \subseteq V$, *s.t.* $|V'| \leq K$ *and* $C_{LC}(V') \geq X$; *No otherwise.*

Unfortunately, the MAX C_{LC} problem is also NP-hard and difficult to approximate as well. However, we demonstrate certain mathematical properties of the problem (namely sub-modularity and monotonicity) that allow us to leverage the results of [10] to prove that a greedy approach achieves the best approximation ratio unless P = NP. We then create an algorithm that selects nodes in a manner equivalent to the greedy approach but does so more efficiently, hence running faster. This second algorithm maintains the theoretical guarantees of the greedy approach with respect to approximation and improves upon the theoretical guarantees of the greedy approach with respect to runtime.

Both algorithms are then experimentally evaluated to demonstrate a significant speedup of several orders of magnitude with the improved algorithm. We then analyze the experimental spreading potential of a set of vertices chosen with our algorithm against the top k nodes based on several common centrality measures from the literature. We found our GREEDY-C_{LC} algorithm identifies sets of nodes whose corresponding C_{LC} value is consistently greater than that found using centrality measures (average increase as compared to centrality measures was 7%). We also compare our approach to the centrality measures based on the expected number of infectees in the aforementioned SIR model. On average, GREEDY-C_{LC} outperforms the other centrality measures (average of 1%). Additionally, we also found that both in terms of optimizing C_{LC} and expected number of infectees, GREEDY-C_{LC} more consistently picked the well-performing sets of nodes than any single centrality measure.

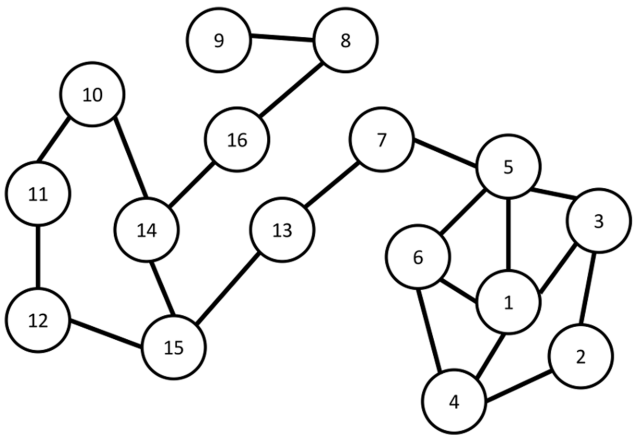

Figure 1. An Example Network. $N_{12} = 5$, $N_{11} = 4$, and $N_{15} = 7$. $C_L(1) = 77, C_L(2) = 35$, and $C_L(10) = 27$. $C_{LC}(1,2) = 77, C_{LC}(1,10) = 104$. Note that although the local-centrality for node 2 is higher than that of 10, the combinatorial local centrality is higher when nodes 1 and 10 are paired, rather than 1 and 2.
doi:10.1371/journal.pone.0090303.g001

After we review related work, the rest of the paper is outlined as follows. In Algorithms and Analysis, we present the complexity and approximation results for the MAX-C_{LC} and DEC-C_{LC} problems. We then show that the a greedy approach obtains the best possible approximation ratio under currently-accepted theoretical assumptions. We then refine our greedy algorithm and produce the theoretical speed-up. Next, our data sets and experimental set-up are provided, followed by our experimental results. We conclude with a brief discussion including directions for future research. Full proofs are contained in the supplemental information section.

Related Work

Identifying epidemic spreaders in a social network is a very active area of research. For instance, identifying a single node with the ability to spread an epidemic effectively has been previously studied in [1,5–7,11]. This paper focuses on a different problem: identifying a set of nodes that can optimally spread an epidemic. We build on the centrality measure and epidemic model of [7]. In that work, the authors introduce Local Centrality as a centrality measure which is a trade off between computational complexity and influence prediction, finding a middle ground between measures such as betweenness and degree (respectively too computationally expensive and of little relevance on large networks).

Identifying sets of epidemic spreaders from a combinatorial centrality measurement (similar to what is done in this paper) is discussed in [8] where the authors elegantly discuss the issues with choosing a set of nodes which either promote or disrupt spreading (KPP-POS and KPP-NEG). They also find that off the shelf centrality measures are not well suited to finding such sets. They describe their own greedy algorithm to find sets of nodes using their proposed group centrality measures [9]. approaches KPP-POS and KPP-NEG with an information theory entropy measure and demonstrate positive results in their simulating environment, however the authors note the entropy calculation is too computationally expensive for large networks.

A more generalized epidemic-like model, the *independent cascade* (IC) model was introduced in classic work of [2] and later improved upon in [12] in terms of efficiency (the original work of [2] had scalability issues due to its dependence on simulation runs). However, this framework is somewhat different from the epidemic model introduced in [7] as under the IC model, an infected node has only one chance to spread a contagion before recovering where here the infected node recovers probabilistically. Further, we note that [12] uses a path-based approach where here we use a neighborhood-based approach (which in our tests outperforms the related path-based approach of closeness). Developing a combinatorial path-based heuristic for the model of [7] and comparing it to the algorithm presented in this paper is an important direction for future work.

In [13] they instead focus on a rumor spreading model for social contagion and information propagation, which is similar to the SIR Model but includes a dampening effect where nodes are more likely to become a stifler (similar to recovered nodes) if they are in

Table 1. Algorithm: GREEDY-C_{LC}.

Algorithm: GREEDY-C_{LC}

INPUT: $K<n$

OUTPUT: $V' \subseteq V$, s.t. $|V'| \leq K$ and V' is the greedy solution to Max C_{LC}.

```
1:  V' = ∅
2:  bestVal,curVal = 0
3:  bestInd = null
4:  while |V'|<K do
5:    for i∈U−V' do
6:      curVal = C_LC(V' ∪ {i}) − C_LC(V')
7:      if curVal>bestVal then
8:        bestVal = curVal
9:        bestInd = i
10:     end if
11:   end for
12:   V' = V' ∪ {bestInd}
13: end while
14: RETURN V'
```

contact with other spreaders (infectees) or stiflers. They find that k-core index does not determine the spreading capabilities of the nodes but rather whether or not a given node prevents the diffusion of a rumor to a system-wide scale. Additionally [14], and [2] investigate spreading conditions under a linear threshold model, where the activity of neighbor nodes activates currently inactive nodes [14]. finds a formula for the average size of activated nodes given the size of the seed set and note that the existence of cascades are extremely sensitive to small initial sets of active nodes. The dynamics of these models provide rich new testing grounds for our algorithm in future work. We believe the linear threshold model could be particularly conducive to C_{LC} because it tends to avoid clustering in lieu of a more even spread which may result in more areas with inactive nodes surrounded by active nodes. The rumor dynamics model also is disadvantageous for highly clustered infected sets so we may also see positive results under that model.

Materials and Methods

Algorithms and Analysis

Here, we present theoretical results on the C_{LC} problems defined in the introduction as well as establish algorithms that obtain certain guarantees. First, we examine the computational complexity of the optimization and decision problems associated with maximizing combinatorial local centrality. Unfortunately, these problems are intractable by an embedding on the Max-K-Cover problem of [15] which has previously been proved to be NP-hard.

Theorem 1. *The Max C_{LC} Problem is NP-Hard.*

Theorem 2. *The Dec C_{LC} Problem is NP-Complete*

We will use the notion of approximation introduced in [16] to analyze the performance of our algorithms. Specifically, we define an α-approximate algorithm as follows. Let U be a universe of elements and f be a function that maps subsets of U to real numbers. Let S,S^* be subsets of U and $f(S^*)$ obtains an optimal value and S be a subset returned by approximation algorithm A. We say that A is an α-approximate algorithm if $f(S) \geq \alpha f(S^*)$. Based on this notion, we are able to leverage another result of [15] to make the following statement on the limit of our ability to approximate Max C_{LC} (in polynomial time) under accepted theoretical assumptions.

Theorem 3. *Max C_{LC} cannot be approximated in polynomial time within $1-\frac{1}{e}+\epsilon$ for $\epsilon>0$ unless $P=NP$.*

Knowing this limit, it is desirable to seek an algorithm that obtains a matching approximation ratio. Clearly, such an algorithm would then obtain the best provable approximation unless P = NP, a currently-accepted assumption in computer science. In order to provide such a result, we prove a few important lemmas that we shall require that deal with properties of the function C_{LC}. First, we show that it is monotonic. Given set U, a function f is monotonic iff for any pair of subsets $S,S' \subseteq U$ where $S \subseteq S'$, we have $f(S) \leq f(S')$.

Lemma 1. *$C_{LC}(V')$ is monotonic.*

The next important property we prove about C_{LC} is that it is *sub-modular*. We say a function f is sub-modular iff for $i \notin S'$ and $S \subseteq S'$, we have $f(S \cup \{i\}) - f(S) \geq f(S' \cup \{i\}) - f(S')$.

Theorem 4. *C_{LC} is sub-modular.*

Using the properties of monotonicity, we are able to show that a greedy algorithm for approximating C_{LC} obtains the best approximation ratio unless P = NP. This follows directly from the results of [23]. We include a basic greedy algorithm (GREEDY-C_{LC}, show in in Table 1) and a theorem showing it can run in polynomial time below.

Theorem 5. *GREEDY-C_{LC} takes $O(K^2nk^{*4})$ time.*

The following theorem leverages our two previously described lemmas as well as the construction used in the proof of Theorem 1 to show that the algorithm obtains the best approximation ratio unless P = NP.

Theorem 6. *GREEDY-C_{LC} obtains the best possible approximation ratio in polynomial time unless $P=NP$*

Though polynomial, the result of Theorem 5 is likely problematic for larger networks. As such is the case we sought to improve upon this run-time with an improved algorithm - GREEDY-C_{LC}2 (pseudo-code provided in Table 2). We prove the following guarantees for this algorithm.

Theorem 7. *Any solution produced by algorithm 2 could also be produced by algorithm 1.*

Theorem 8. *GREEDY-C_{LC} 2 takes $O(K^2m)$ time.*

In this improved approach, our first intuition was to pre-compute the quantity $\sum_{w \in \eta_v} N_w$ for each node v and store it in a data-structure. Next we decided to keep track of all the first neighbors of the set we are building, which allows the algorithm to avoid recalculating that set each loop. This yields a provable improvement in time complexity by a factor of k^{*3}. Additionally, we added a practical improvement as well. In a related submodular problem, Leskovec [17] obtained a 700 percent increase by "lazy" evaluation of the submodular function (over the basic greedy approach, based on experiments). We include that in this approach by altering line 7, correctly avoiding unnecessary calculations of centrality for poorly-performing nodes. We present experimental evaluations of how this modification affected our problem in the next section.

Example 1. *Table 3 features the improved algorithm selecting a set of three vertices from a small network of 35 primates' relationships. Each column contains a vertex followed by how much that vertex would increase the C_{LC} of the set if it were added to the set. For example, as the algorithm runs through each vertex seeking the first to add to the set, the first vertex is automatically the first greatest increase found, until the fourth vertex is found to generate a higher C_{LC} value, and last in the column is vertex 16, which is then becomes first vertex in the set. In the second and third columns the practical improvement of*

Table 2. Algorithm: GREEDY-C_{LC}2.

Algorithm: GREEDY-C_{LC}2

INPUT: K<N; $G = (V, E)$

OUTPUT: V′ ⊆ V, s.t. $|V'| \leq K$ and V' is the greedy solution to Max C_{LC}.

```
V′ = ∅
bestVal,curVal = 0
bestInd = null
firstNeighbors = ∅
for i∈V do
    for j∈Neighbors(i) do
        NW = NW+j
        for k∈Neighbors(j) do
            if k ∉ NW then
                NW = NW+k
            end if
        end for
    end for
    N_v[i] = |NW|
end for
for i∈V do
    for j∈Neighbors(i) do
        N2_v[i] = N2_v[i]+N_v[j]
    end for
end for
lastVal[0..n] = ∞
while |V′|<K do
    firstNeighbors = firstNeighbors+newNeighbors(firstNeighbors,bestInd)
    (see note[1])
    C_LC(V′) = C_LC(firstNeighbors)
    for i∈U−V′ do
        if lastVal[i]>bestVal then
            lastVal[i], curVal = C_LC(V′ ∪ {i}) − C_LC(V′)
            curVal = C_LC(V′ ∪ {i}) − C_LC(V′)
            if curVal>bestVal then
                bestVal = curVal
                bestInd = i
            end if
        end if
    end for
    V′ = V′ ∪ {bestInd}
end while
RETURN V′
```

[1]The function *newNeighbors(V; v)* takes a set of nodes and a new node and adds any neighbors of the new node that are not already in the set.

doi:10.1371/journal.pone.0090303.t002

GREEDY-C_{LC}2 is visible. Each time a $X > Y$ appears it signifies that a vertex was skipped because in the last iteration it increased C_{LC} by less than whatever is the current best increase for this iteration.

Datasets

We examined five different networks in our analysis. They include an a sexual interaction network [18], email network [19], an academic collaboration network [20], a protein interaction network [21], and a social network [22]. Each network is both unweighted and undirected. Our intuition was to utilize networks from a variety of domains in our evaluations.

The sexual interaction, email, academic collaboration, and protein interaction networks are denoted A, B, C, and D (respectively) in Figures 2 and 3. We provide some details on these networks in Table 4. The social network was primarily used for run-time analysis (Table 5). These networks are described in more detail below.

The sexual interaction network is an online sex community in Brazil in which a link represents that one of the individuals posted online about a sexual experience with the other individual, resulting in a bipartite graph. The data was extracted from September of 2002 to October of 2008 Luis E. C. Rocha & Holme [18].

The email network is derived from the communications of members of the University Rovira i Virgili. It was extracted in 2003 [19].

The academic collaboration network is derived from the arXiv pre-print server and covers scientific collaborations between authors papers submitted to the General Relativity and Quantum Cosmology category from Jan. 1993–Apr. 2003 [20].

The protein interaction network is a network consisting of protein-protein interactions in yeast [21].

The social network is derived from YouTube, the video-sharing website that allows users to establish friendship links [22]. The sample was extracted in Dec. 2008. Links represent two individuals sharing one or more subscriptions to channels on YouTube.

The Douban network was mined from Douban.com, launched on March 6, 2005, which is a Chinese Web 2.0 website providing user review and recommendation services for movies, books, and music. It is also the largest online Chinese language book, movie and music database and one of the largest online communities in China [23].

Experimental Set-Up

The runtime experiments on the Douban social media network were conducted on a platform with an Intel X5677 Xeon Processor operating at 3.46 GHz with a 12 MB Cache and 288 GB of physical memory. The machine was running Red Hat Enterprise Linux version 6.1. Only one core was used for experiments. All other experiments were run on a computer equipped with an Intel Core i7 M620 equipped with two cores at 2.67 GHz with 4.00 GB of RAM (only one core was utilized). The machine was running Windows 7. GREEDY-C_{LC} and GREEDY-C_{LC}2 were written using Python 2.7.3 in 75 and 80 lines of code, respectively, that leveraged the NetworkX library available from http://networkx.lanl.gov/. The SciPy library from http://www.scipy.org/ was also used for the experimental setup.

We compared our improved algorithm to choosing the top K vertices from many common centrality measures. Top-LC refers to choosing the top K vertices using Local Centrality, rather than trying to optimize Combinatorial Local Centrality. Degree is simply the number of edges a node has. Shell number refers to the greatest core to which a node belongs (see [1] for details). Betweenness measures how many shortest paths, of all vertex pairs in the network, run through a vertex. Closeness is defined as the inverse of farness, where a node's farness is the sum of distances to every other node along shortest paths. Eigenvector centrality and PageRank are recursive measures which take into account both how many neighbors a vertex has and the Eigenvector centrality/Pagerank of those neighbors.

Table 3. Example GREEDY-$C_{LC}2$ Run.

1) 481	1) 160	1) 160
4) 1441	4) 685	2) 160>160
5) 1592	5) 826	3) 160>160
10) 1885	7) 826>703	4) 240
12) 2259	8) 826>279	11) 240>0
13) 2298	9) 826>279	19) 240>69
16) 2727	12) 987	20) 240>0
	18) 987>606	34) 240>80
	23) 987>899	35) 240>80
	24) 987>533	
	25) 987>306	
	26) 987>445	
	27) 987>759	
	28) 987>690	
	29) 987>690	
	30) 987>445	
	31) 987>708	
	32) 987>250	
	33) 987>245	
	34) 987>80	
	35) 987>80	

Each column represents the algorithm choosing a vertex to add to the set; vertices 16, 12, and 4 were chosen and in that order. Vertices only appear if they are the maximum addition when considered or if they are ignored (represented by the inequalities). The format is as follows: the vertex considered appears first, followed by a parenthesis, and then either a value or an inequality. The inequality represents that the considered node had a lower addition to the C_{LC} of the set last iteration than the current best addition now, and therefore does not need to be computed this round. A single value represents the addition to C_{LC} that vertex would contribute.

doi:10.1371/journal.pone.0090303.t003

Results

Runtime

We first examined the run time of our improved algorithm as opposed to the simple greedy algorithm. Using small subsets of the email network, we prompted each algorithm to select 5% of the subgraph. Table 5 displays the speed-up of the improved algorithm over the simple greedy algorithm even on these very small graphs. The difference is multiple orders of magnitude, aligning with our theoretical results.

Next we wanted to demonstrate that our improved algorithm also performs well with respect to computing other common centrality measures. Taking four of the datasets, the email, sexual interaction, social network, and the Douban network, we generated initial seed sets with GREEDY-$C_{LC}2$ and compared this time to how long it took for the NetworkX built in functions for Closeness and Betweenness dictionaries to be calculated, shown in Table 6. Our improved algorithm relies on pre-computation of the value N_w, the number of first and second neighbors of each vertex in the graph, so the time it takes to calculate N_w is also included in Table 6. Once the dictionaries for Closeness and Betweenness are found, they must be sorted to deliver the top K nodes, but that time is negligible next to the time required to build the dictionaries and therefore is not included. The NetworkX implementations for both Closeness and Betweenness are of complexity $O(nm)$ [24,25]. Recall that the time complexity of GREEDY-$C_{LC}2$ is $O(K^2m)$, therefore when K is relatively small compared to n we should expect GREEDY-$C_{LC}2$ to outperform Closeness and Betweenness.

Finally, we demonstrated that our GREEDY-$C_{LC}2$ algorithm could also deliver results on a larger datasets - which is a more typical need in practical applications dealing with social media site. Here we used a social network extracted from the Douban social media site [23], which consisted of 154,907 nodes and 654,188 edges. For this experiment, we evaluated the runtime of our algorithm as a function of the cardinality of the solution (Figure 4). We found that a quadratic relationship was maintained ($R^2 = 0.99$) which reflects our complexity result of Theorem 8. Finding a set of 4% of the population (6200 nodes) took 18.25 hours, which significantly outperformed other measures. Currently, we are exploring means to further scale this approach, including additional heuristic approximations and parallelization.

C_{LC} Optimization

To test the efficacy of GREEDY-$C_{LC}2$, we examined five different 500 node subgraphs of four separate networks. On each subgraph, we chose the top 1, 3, 5, and 8 percent of vertices based on several common centrality measures and using GREEDY-$C_{LC}2$. First we needed to demonstrate that GREEDY-$C_{LC}2$ does in fact optimize C_{LC} better than other measures. This is difficult to show definitively, because we do not have other algorithms which aim to maximize C_{LC} to use as a comparison, but the contrast with common centrality measures is still helpful. In Figure 2 we present the averages of the C_{LC} value over those five subgraphs for the subsets chosen by GREEDY-$C_{LC}2$ versus each of the subsets chosen by selecting the top X percent of nodes using other centrality measures. Figure 2 shows both that sets that have a high C_{LC} are in practice very different from other measures (i.e. we did not develop a trivially new definition), and then that seeking sets with other centrality measures is not good shortcut to finding sets that have a high C_{LC}. In all cases, GREEDY-$C_{LC}2$ chose the set with the highest C_{LC}, and was an average of 7% greater than the top performer for each percent and data set pair. On every dataset an analysis of variance (ANOVA) reveals that there is a significant difference in the performance among our algorithm and the centrality measures with respect to increase or decrease in C_{LC} (p-value less than 0.04556 calculated with R version 3.01) except academic collaboration network, which had a p-value between 0.8949 and 0.9977 for each percentage trial. Some of the uncertainty in the statistical analysis is attributable to the variance between the random subgraphs, as in many cases average C_{LC} values across all centrality measures differed between two subgraphs as much as 20%.

In some trials, particularly in sexual interaction and academic collaboration (A and C in Figure 2), GREEDY-$C_{LC}2$ reached a maximum C_{LC} value before selecting 8% of the graph, at which point the averages of other centrality measures begin to approach GREEDY-$C_{LC}2$. However, as C_{LC} has already been maximized in this case (because the first neighbors of the seed set cover the entire graph), they will never surpass the C_{LC} of the smaller set. In a real world scenario, this may be taken advantage of as a way to save advertising costs or focus on a smaller set of the population for epidemic evaluation.

Epidemic Evaluation

Next the same sets as chosen in the previous section were the initial infectees for 1000 simulation runs over the SIR model. In this paper, to remain consistent with the work of [7], we mimicked their experimental model. After setting our initial infectees to the infected state, we run the SIR model for ten time steps and then

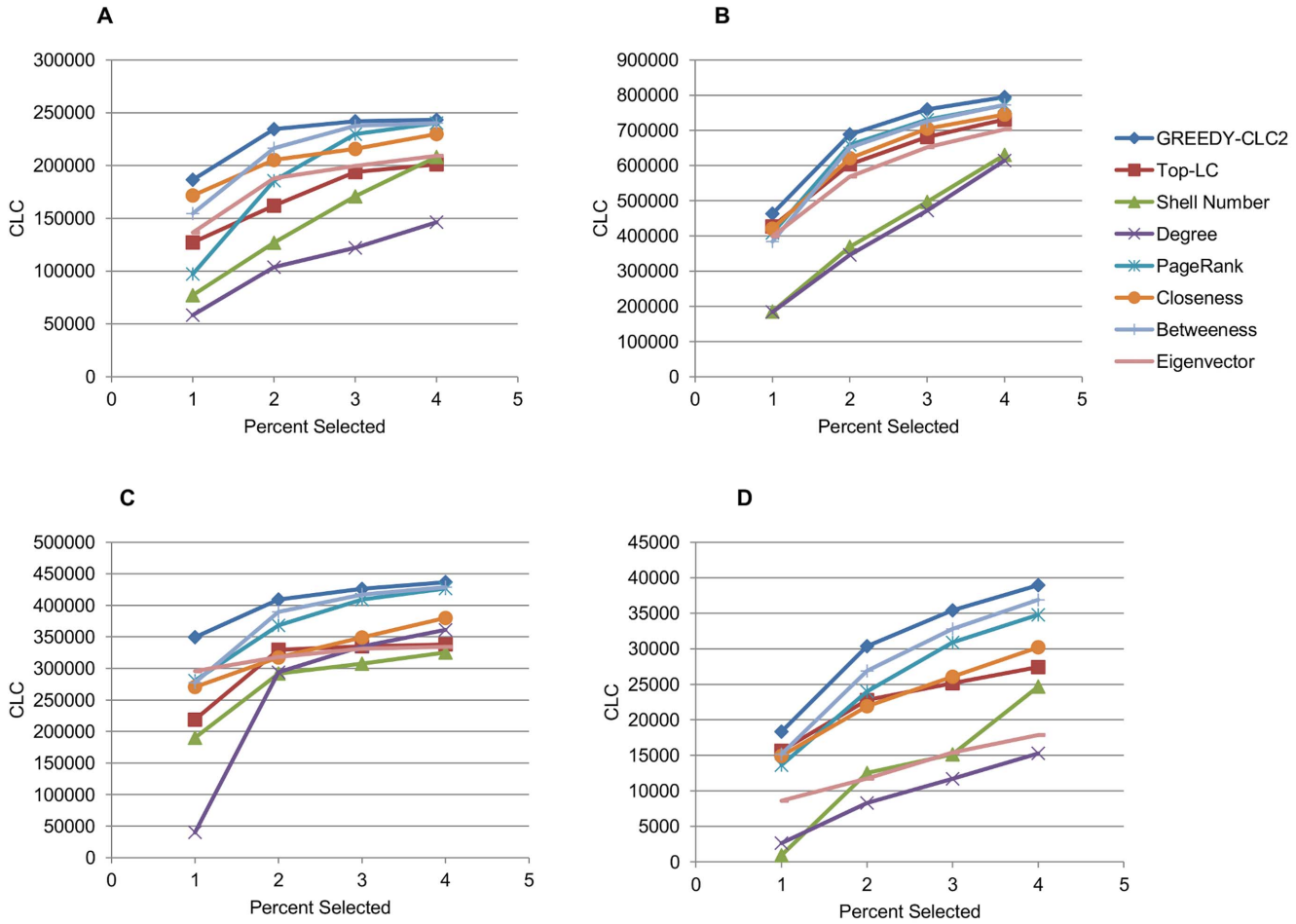

Figure 2. C_{LC} of Sets Chosen with Various Centrality Measures. The C_{LC} of sets chosen by various centrality measures. Sexual interaction, email, collaboration, and protein interaction networks are respectively graphs A, B, C, and D.
doi:10.1371/journal.pone.0090303.g002

sum the recovered and infected vertices to determine the total number of infected vertices. The results, again averaged over the five subgraphs from each network, are shown in Figure 3. The sets chosen by GREEDY-$C_{LC}2$ spread on average to 1% more vertices than the maximum spreader from the rest of the centrality measures over each percent and dataset pair. Furthermore, although occasionally another centrality measure will outperform GREEDY-$C_{LC}2$ on a single cardinality and dataset pair, which measure does so is highly inconsistent. Particularly visible in the sexual interaction network (panel A of Figure 3), GREEDY-$C_{LC}2$ did not produce a set as big as 5% or 8% of the graph on every subgraph, so other centrality measures gained an advantage in that they began with more infectees. Interestingly though, C_{LC} still remained in the top half of the centrality measures, suggesting again a certain threshold after which it is inefficient to continue seeding a graph and a way to conserve real world resources. An analysis of variance (ANOVA) on every dataset reveals that there is a significant difference in the performance among sets chosen by our algorithm and the other centrality measures with respect to increase or decrease in total vertices infected (p-value less than 0.0003426 calculated with R version 3.01), except the sexual interaction which had a p-value between 0.9572 and 0.9985 for each percentage trial. However, we also note that this may be a somewhat degenerate case as this particular sexual interaction network consisted of only heterosexual interactions - which leads to a bipartite structure. This may account for the C_{LC} measure covering the entire network without using all of the resources - which in turn led to inconsistent performance against the centrality measures in the simulation trials.

Discussion

In this paper, we explored the problem of identifying a set of nodes that will cause an epidemic to spread under the SIR model of [7]. To do so, we extended the centrality measure of [7] for sets rather than individual nodes. Though we found that finding a set of nodes that maximizes this combinatorial centrality measurement is NP-hard, we develop a polynomial-time heuristic that we prove to provide the best approximation ratio unless P = NP. We then further improve the performance, both theoretically and practically in a modified version of the algorithm that provides the same theoretical guarantee. We implemented our algorithms and evaluated them on real-world datasets in terms of runtime, ability to maximize the combinatorial centrality measure, and the ability to find sets of nodes that encourage spreading in the SIR model. We found our algorithms to outperform standard approaches in all of these evaluations. Further, we show our approach to scale to networks of 10^5 nodes.

Future work could include a modified version of C_{LC} which produces a disease spread mitigation strategy. In such a scenario,

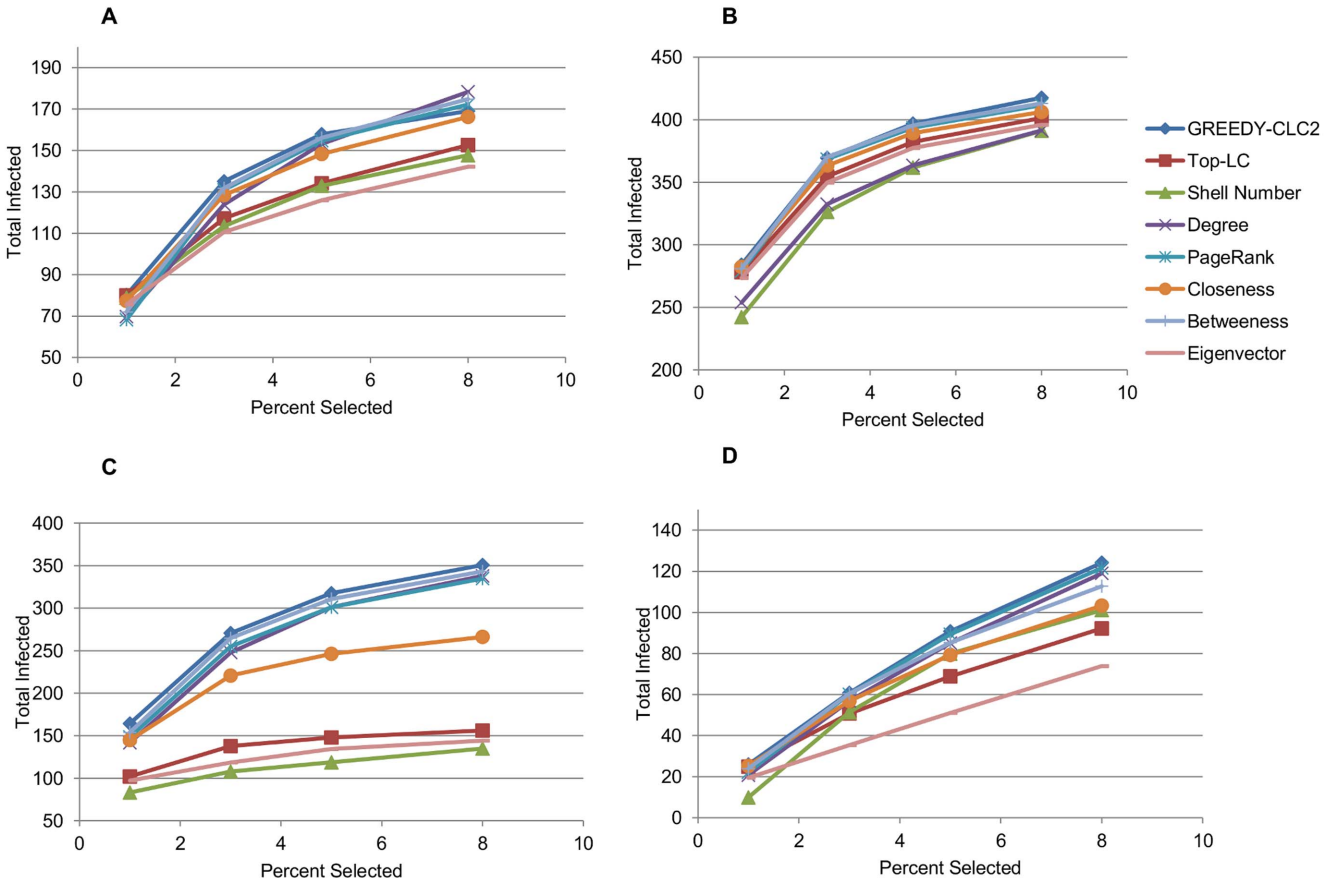

Figure 3. Spreading Impact of Sets Chosen with Various Centrality Measures. The number of vertices infected after 1000 simulation runs to 10 steps in the SIR model, averaged over five subgraphs for each datum. Sexual interaction, email, academic collaboration, and protein interaction networks are respectively graphs A, B, C, and D.
doi:10.1371/journal.pone.0090303.g003

we would attempt to find nodes that, if "inoculated" would minimize the maximum value for C_{LC} with respect to a given cardinality constraint. Additionally, further evaluation of C_{LC} based on different diffusion models, such as those raised in the related work section, is another important direction for further research. In particular, an evaluation of the metric under a classic SIR Model, rather than the variant described in this paper and in [9], would be a good first step.

Appendix

Proof of Theorem 1

The Max C_{LC} Problem is NP-Hard.

Proof.

Definition 3. *Max K-Cover* [15]
INPUT: *Universe* U, *a set of subsets* C, *and natural number* K'
OUTPUT: $C' \subseteq C$, $|C'| \leq K'$ *s.t.* $|\cup_i C_i|$ *is maximized.*

Embedding: Given Max K Cover as defined in definition 2.4, we create an instance of the Max C_{LC} as follows. Form a bipartite graph G by creating a vertex for each $c_i \in C$ and each $e_i \in U$. Create a directed edge from c_i to e_i if $e_i \in c_i$. For each vertex corresponding to an element e_i create two additional nodes, $e_{i,a}$ and $e_{i,b}$. Also add a directed edge from e_i to $e_{i,a}$ and from $e_{i,a}$ to $e_{i,b}$. Each node corresponding to a subset c_i now has a path length of three to some $e_{i,b}$.

Table 4. Dataset Information.

Dataset	Nodes	Edges
Brazil Sexual Interaction Network	16730	39044
University Rovira i Virgili Email Network	1133	5451
Academic Collaboration Network	5242	14484
Yeast Protein Interaction Network	1870	2203
Youtube Social Network	13723	76765
Douban Social Network	154907	654188

doi:10.1371/journal.pone.0090303.t004

Table 5. Runtime of GREEDY-C_{LC} vs GREEDY-C_{LC}2.

Graph Size	Simple Run Time	Fast Run Time
100	80.37	0.03
200	1502.95	0.19
300	9784.2	0.74
400	35610.68	1.90

Runtimes are in seconds. Each algorithm selected 5% of the given graphs, which are random samples of the email dataset.
doi:10.1371/journal.pone.0090303.t005

Table 6. Runtime of GREEDY-$C_{LC}2$, Closeness, and Betweenness.

Nodes	Edges	CLC Size	N_w Time	CLC Time	Betweenness	Closeness
1133	5451	249	0.61	2.39	8.64	2.60
16730	39044	1000	16.58	383.89	2307.14	561.54
13723	76765	1000	66.13	407.93	2525.92	693.55
154907	654188	2000	249.22	7129.11	>24 hrs	>24 hrs

Runtimes are in seconds. N_w must be precomputed and stored once before GREEDY-$C_{LC}2$ can be run.
doi:10.1371/journal.pone.0090303.t006

Claim 1. *Embedding of Max K-Cover into Max C_{LC} can be accomplished in polynomial time, as graph G has $|C|+3|U|$ vertices and $3|U|$ edges, whose creation takes constant time.*

Claim 2. *Given set V' returned by the an instance of Max C_{LC} with $K<|C|$, the set $C^*=\{c|v_c\in V'\}$ is the solution to the Max K-Cover problem.*

Suppose by way of contradiction that there exists some set $C^{**}\subseteq C$ such that $|C^{**}|\leq K$ and the number of elements covered by C^{**} is greater than the number of elements covered by C^*. Let $V''=\{v_c|c\in C^{**}\}$.

The number of distinct nearest neighbors for C^{**} is greater than the number of distinct nearest neighbors of C^*. Note that for all vertices corresponding to elements i, $\sum_{w\in\eta_i}N_w=1$ by the construction, and $C_{LC}(V_1)$ is simply the count of distinct nearest neighbors of set V_1. Therefore $C_{LC}(V'')>C_{LC}(V')$, which is a contradiction.

Claim 3. *Given set C^* returned by Max K-Cover, the set $V'=\{v_c|c\in C^*\}$ is a solution to Max C_{LC}.*

Suppose by way of contradiction that there exists some V'' where $|V''|\leq|C|$ and $C_{LC}(V'')>C_{LC}(V')$. Let $C^{**}=\{c|v_c\in V''\}$.

$C_{LC}(V'')=\sum_{u\in\eta_{V''}}\sum_{w\in\eta_u}N_w$, which under the construction is $|\eta_{V''}|$. Similarly $C_{LC}(V')=|\eta_{V'}|<C_{LC}(V'')$. This is equivalent to saying that the number of nearest neighbors covered by set C^{**} is greater than that of C^*, which is a contradiction.

Proof of Theorem 2

The Dec C_{LC} Problem is NP-Complete.

Proof. Given an oracle that produces a solution V', we can clearly check if $C_{LC}(V')\geq X$ in polynomial time by Theorem 1.

Proof of Theorem 3

Max C_{LC} cannot be approximated within $\frac{e-1}{e}+\epsilon$ for $\epsilon>0$ unless P = NP.

Proof. Embedding: We use the same embedding as in Theorem 2.1 above.

Let $x=$ the number of sets covered by some set C^* of Max K-Cover.

Let $y=C_{LC}(V')$ where V' is the set of vertices for Max C_{LC}.

Claim 4. $x\geq y$.

Suppose by way of contradiction that $x<y$. If C^* covers fewer neighbors than $C_{LC}(V')$ then at least one of those neighbors u must have a $Q(u)>1$. However under the construction all vertices e_i associated with elements have $Q(i)=1$ as they each have only one next nearest neighbor e_{ib} and no neighbors to that vertex, and we have a contradiction.

Claim 5. $x\leq y$.

Suppose by way of contradiction that $x>y$. If C^* covers more neighbors than $C_{LC}(V')$ then at least one of those neighbors u must have a $Q(u)<1$. However under the construction all vertices e_i associated with elements have $Q(i)=1$ as they each have only one next nearest neighbor e_{ib} and no neighbors to that vertex, and we have a contradiction.

By the embedding, Claims 1.4 and 1.5, and Thm 4.4 of [15] concerning the limit of approximating set cover, the Max C_{LC} cannot be approximated within $\frac{e-1}{e}+\epsilon$ for $\epsilon>0$ unless P = NP.

Proof of Lemma 1

$C_{LC}(V')$ is monotonic.

Proof. Suppose by way of contradiction there exists $S\subseteq S'$ s.t. $C_{LC}(S)>C_{LC}(S')$. Then

$\sum_{u\in\eta_S}\sum_{w\in\eta_u}N_w>\sum_{x\in\eta_{S'}}\sum_{y\in\eta_x}N_y$ which implies $\sum_{w\in\eta_u}N_w>\sum_{y\in\eta_x}N_y$.

However, because $S\subseteq S'$ we know $\eta_S\subseteq\eta_{S'}$ and $\eta_u\subseteq\eta_x$.

Because the total neighbors of a subset is necessarily less than the total neighbors of its superset, we have a contradiction.

Proof of Theorem 4

C_{LC} is sub-modular.

Proof. Setup: $S\subseteq S'\subseteq V$; V is the set of vertices in graph G; vertex $i\notin S'$;

Suppose by way of contradiction $C_{LC}(S\cup\{i\})-C_{LC}(S)<C_{LC}(S'\cup\{i\})-C_{LC}(S')$.

Then

$\sum_{a\in\eta_{(S\cup\{i\})}}\sum_{w\in\eta_a}N_w-\sum_{a\in\eta_S}\sum_{w\in\eta_a}N_w<\sum_{b\in\eta_{(S'\cup\{i\})}}\sum_{w\in\eta_b}N_w-\sum_{b\in\eta_{S'}}\sum_{w\in\eta_b}N_w$

If we let $a'=\eta_{(S\cup\{i\})}-\eta_S$ and $b'=\eta_{(S'\cup\{i\})}-\eta_{S'}$ the inequality above becomes:

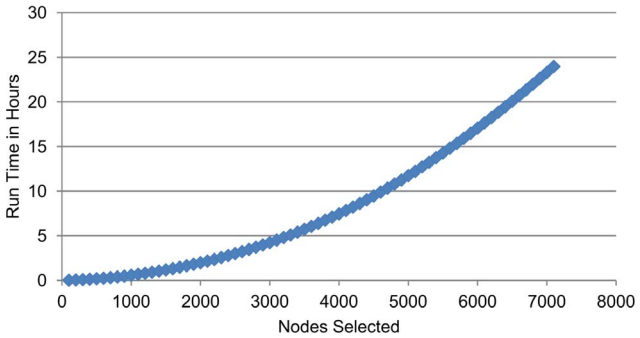

Figure 4. Run Time of GREEDY-$C_{LC}2$ on the Douban Social Network. The run time in hours for GREEDY-$C_{LC}2$ to build sets between 100 and 7100 nodes.
doi:10.1371/journal.pone.0090303.g004

$$\sum_{i\in a'}\sum_{w\in\eta_i} N_w < \sum_{i\in b'}\sum_{w\in\eta_i} N_w \quad (4)$$

Note that a' and b' are the sets of neighbors added to sets S and S', respectively, with the addition of vertex i.

Claim 6. $a' \subseteq b'$

$$a' = \eta_{S\cup i} - \eta_S = \eta_S \cup \eta_i - (\eta_S \cap \eta_i) - \eta_S = \eta_i - (\eta_S \cap \eta_i)$$

Similarly, $b' = \eta_i - (\eta_{S'} \cap \eta_i)$. Since $S \subseteq S'$, $(\eta_S \cap \eta_i) \subseteq (\eta_{S'} \cap \eta_i)$, therefore $a' \subseteq b'$.

However, with $a' \subseteq b'$, inequality 4 cannot be true, therefore C_{LC} is sub-modular.

Proof of Theorem 6

GREEDY-C_{LC} obtains the best approximation ratio unless $P = NP$.

Proof.

Claim 7. *GREEDY-C_{LC} is a Greedy Algorithm.*

We build set V' by adding one element at each iteration of the while loop. A new element is chosen by analyzing the increase C_{LC} for each node not in V' and picking the maximal node. Using a local heuristic to make each choice in a set of decisions is a greedy approach.

Claim 8. $C_{LC}(\emptyset) = 0$:

$$C_{LC}(\emptyset) = \sum_{u\in\eta\emptyset}\sum_{w\in\eta_u} N_w = \sum_{\emptyset}\sum_{w\in\eta_u} N_w = 0$$

Proof of Theorem: For any monotonic, sub-modular function $f(S)$ where $f(\emptyset) = 0$, a greedy algorithm guarantees an $\alpha = (1-1/e)$ approximation [23]. By Theorems 1 and 4, and Claims 7 and 8, GREEDY-C_{LC} gives an $\alpha = (1-1/e)$ approximation.

By Theorem 2.1 of [2] and the approximation ratio α above, α is the best approximation if $P \neq NP$.

Proof of Theorem 5

GREEDY-C_{LC} takes $O(K^2nk^{*4})$ time.

Proof.

Claim 9. C_{LC} *takes* $O(|V'|k^{*4})$

To compute $C_{LC}(V')$, first we iterate through each vertex in V'. For each vertex, we consider each neighbor, and barring repeated vertices in the set we add those neighbors to a set of first neighbors for set V', which takes $|V'|k^*$. For each vertex in the first neighbor set we count the first and second neighbors, which is no worse than k^{*4}. Therefore the time complexity is $O(|V'|k^{*4})$.

GREEDY-C_{LC} utilizes two looping control structures. The first is a while loop that runs K times, and the second is a nested for loop that runs for at most n times, for each vertex in the graph. Inside that loop the C_{LC} algorithm, $O(|V'|k^{*4})$, is called twice. The time complexity is then $O(K^2nk^{*4})$.

Proof of Theorem 7

Any solution produced by algorithm 2 could also be produced by algorithm 1.

Proof. Suppose by way of contradiction the condition that lastVal[i]>bestVal caused us to omit the maximal node, j, or that the maximal node's last recorded marginal increase in C_{LC} was lower than the current best value. As C_{LC} is sub-modular by Thm 2.5, an updated marginal increase of C_{LC} would have to be lower than lastVal[j]. However if the new marginal increase is lower than lastVal[j], it must also be lower than bestVal, and therefore j could not be optimal.

Proof of Theorem 8

Proof. GREEDY-$C_{LC}2$ takes $O(K^2m)$ time.

Given that we store N_w for all vertices and a list $N2_v$ which contains the sum of N_w for all neighbors w of a node v, and an alternate form of computing $C_{LC}(V')$ which takes the first neighbors of set V', $fn(V')$:

To compute $C_{LC}(V')$, now we simply iterate through the $fn(V')$ and sum $N2_v$ for each, which takes $O(|fn(V')|)$. Updating $fn(V')$ requires adding all new neighbors whenever a new vertex is appended to the set, which takes $O(k^*)$ ($fn(V')$ can take multiple vertices, but in the algorithm's implementation it only takes one).

The improved algorithm must also loop until it reaches K vertices, and considers each vertex in the graph when choosing a new vertex. To choose a new vertex, it must update $fn(V')$ with the potential new neighbors of a possible vertex and calculate $C_{LC}(V')$, so the complexity is $Kn|fn(V')|$. But $|fn(V')|$ is bound by Kk^* because it is the total number of neighbors of a set of at most K elements, so the complexity may be reduced to $O(K^2nk^*)$. Finally we simplify the factors nk^* to m.

Acknowledgments

The authors would like to thank Gerardo I. Simari for his support in this research. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders, the U.S. Military Academy, or the U.S. Army.

Author Contributions

Performed the experiments: GM. Analyzed the data: GM PS. Contributed reagents/materials/analysis tools: GM PS BM NH. Wrote the paper: GM PS. Conceived the experiments: GM PS. Designed the experiments: GM PS BM NH.

References

1. Kitsak M, Gallos LK, Havlin S, Liljeros F, Muchnik L, et al. (2010) Identification of inuential spreaders in complex networks. Nat Phys 6: 888–893.
2. Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of inuence through a social network. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, pp. 137–146. doi: http://doi.acm.org/10.1145/956750.956769
3. Anderson RM, May RM (1979) Population biology of infectious diseases: Part i. Nature 280: 361.
4. Easley D, Kleinberg J (2010) Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press. Available: http://www.cs.cornell.edu/home/kleinber/networks-book/.
5. Klemm K, Serrano MA, Eguiluz VM, San Miguel M (2012) A measure of individual role in collective dynamics: spreading at criticality. Scientific Reports 2.
6. Castellano, Pastor-Satorras R (2012) Competing activation mechanisms in epidemics on networks. Scientific Reports 2.
7. Chen D, Lü L, Shang MS, Zhang YC, Zhou T (2012) Identifying inuential nodes in complex networks. Physica A: Statistical Mechanics and its Applications 391: 1777–1787.
8. Borgatti SP (2006) Identifying sets of key players in a social network. Computational & Mathematical Organization Theory 12: 21–34.
9. Arroyo DO, Hussain DMA (2008) An information theory approach to identify sets of key players. In: EuroISI. pp. 15–26.

10. Nemhauser GL, Wolsey LA, Fisher M (1978) An analysis of approximations for maximizing submodular set functions. Mathematical Programming 14: 265–294.
11. Kang C, Molinaro C, Kraus S, Shavitt Y, Subrahmanian V (2012) Diffusion centrality in social networks. In: Proc. 2012 IEEE/ACM Intl. Conf. on Advances in Social Networks Analysis and Mining (ASONAM-12).
12. Chen W, Wang C, Wang Y (2010) Scalable inuence maximization for prevalent viral marketing in large-scale socials. In: Proc. of KDD '10. ACM, pp. 1029–1038.
13. Borge-Holthoefer J, Moreno Y (2011) Absence of inuential spreaders in rumor dynamics. Physics and Society.
14. Gleeson JP, Cahalane DJ (2007) Seed size strongly affects cascades on random networks. Physical Review E 75.
15. Feige U (1998) A threshold of ln n for approximating set cover. J ACM 45: 634–652.
16. Garey MR, Johnson DS (1979) Computers and Intractability; A Guide to the Theory of NPCompleteness. New York, NY, USA: W. H. Freeman & Co.
17. Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, et al. (2007) Cost-effective out-break detection in networks. In: KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, pp. 420–429. doi:http://doi.acm.org/10.1145/1281192.1281239
18. Rocha LEC, Fredrik L, Holme P (2010) Information dynamics shape the sexual networks of internet mediated prostitution. Proceedings of the national academy of sciences, March.
19. Arenas A (2012) Network data sets.
20. Leskovec J (2012) Stanford network analysis project. Available: http://snap.stanford.edu/.
21. Barabasi AL, Toraczkai Z (2007) Network databases. Available: http://www3.nd.edu/networks/resources.htm.
22. Zafarani R, Liu H (2009) Social computing repository at asu.
23. Zafarani R, Liu H (2009) Social computing data repository at ASU. Available: http://socialcomputing.asu.edu.
24. Hagberg A, Schult DA, Swart PJ (2013) Networkx. Available: https://networkx.lanl.gov/.
25. Brandes U (2001) A faster algorithm for betweenness centrality. Journal of Mathematical Sociology 25: 163–177.