Community Based Information Dissemination

Zhengwei Yang[1], Ada Wai-Chee Fu[1], Yanyan Xu[1], Silu Huang[2], Ho Fung Leung[1]

[1]Chinese University of Hong Kong
{zwyang,adafu,hfl}@cse.cuhk.edu.hk
[2]University of Illinois at Urbana-Champaign
slhuang@illinois.edu

Abstract. Given a social network, we are interested to determine k seeds that maximize the dissemination of information. Based on the principle of homophily, communities play an important role since information can be disseminated to communities via the seeds. We introduce a new mechanism for detecting communities satisfying the pertinent criteria for communities and information dissemination. We demonstrate the effectiveness of our approach by an application of the results for influence maximization.

Keywords: information dissemination, community detection

1 Introduction

With the growth in social networks and other massive networks, network analysis has emerged as an important research topic. The detection of communities or the listing of *cohesive subgraphs* for a given graph has been of great interest. From studies in sociology, communities are a powerful channel for the dissemination of information [14, 16]. Our problem can be described as follows. In a social network, each vertex corresponds to an individual. We are given a limited amount of resources to inform a *seed* set of vertices (individuals) of size k and the problem is how to choose the seed set to maximize the spread from this seed set to other individuals in the network. We call this the IDM problem problem (IDM stands for Information Dissemination Maximization).

Inspired by the study in sociology about the role of homophily in information dispersal, we propose a model on the IDM problem based on communities. As in previous works on community discovery or cohesive subgraphs, we assume that we are given a simple undirected unweighted graph [9]. Our community definition is related to two basic criteria for cohesive subgraphs, namely, the concept of a clique (i.e. a set of vertices that induces a complete subgraph), and the density of the subgraph. In addition, we consider the distances of vertices from the seed. A key idea in the community search is that we look for the seeds in the process. Each community search begins with a potential seed vertex.

We show that the related optimization problems are NP-hard. We propose efficient algorithms for finding good core-based communities. We then apply the

solution to the problem of influence maximization [11], which has important applications in viral marketing, and the results on a real dataset show that our solution outperforms other state-of-the-art methods.

2 Problem Definition

We study the problem of information dissemination maximization (IDM) in an undirected simple graph, $G = (V, E)$, where V is the set of vertices and E is the set of edges of G. An edge in E between vertices u, v in V is denoted by (u, v) or (v, u), u is a neighbor of v, and vice versa. $adj(u)$ is the set of neighbors of u. Degree $d(u) = |adj(u)|$. A subgraph of G induced by vertex set V' is denoted by $G(V')$. We state the general problem definition as follows.

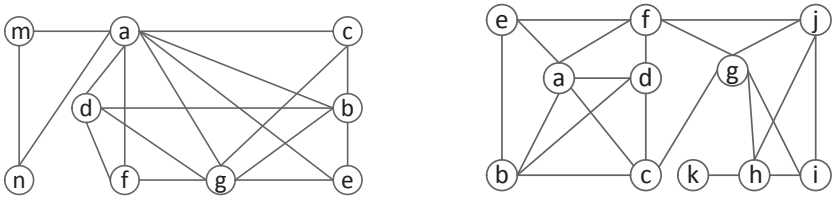
Problem Definition (IDM): Given a graph $G = (V, E)$, and a positive integer k, information dissemination maximization (IDM) aims to find a set S of k seeds, where $S \subseteq V$, that maximizes the number of vertices that are informed by the seed vertices according to some information dissemination model.

A complete undirected simple graph $G = (V, E)$ is a graph such that every pair of vertices u, v in V is linked by an edge (u, v) in E. A subset of vertices, $C \in V$, is called a **clique** if the subgraph of G induced by C is a complete subgraph. The size of C is given by the number of vertices in C. The edges $(v_0, v_1), (v_1, v_2)...(v_{\ell-1}, v_\ell)$ in G forms a path. The length of the path $v_0, v_1, ..., v_\ell$ is given by ℓ.

Definition 1. ***Density****: Given a set of vertices $S \subseteq V$ and the induced subgraph $G_S = (V_S, E_S)$, the density of S is denoted as $den(S)$ and $den(S) = \frac{2|E_S|}{|V_S|*(|V_S|-1)}$.*

Definition 2. ***Radius****: Given a core set C and a set S, where $C \subseteq S \subseteq V$, the radius of S regarding C is defined as the maximum shortest path from $u \in S$ to core C, denoted as $R_C(S)$ and $R_C(S) = max_{u \in S}\{min_{v \in C}|SP(u, v)|\}$, where $|SP(u, v)|$ is the length of a shortest path from u to v in $G(S)$.*

We say that $S \subseteq V$ satisfies a density constraint of γ if $den(S) \geq \gamma$. Let $C \subseteq S \subseteq V$, we say that S and C satisfy a radius constraint of r if $R_C(S) \leq r$.



(a) (b)

Fig. 1. Two Example Graphs

Definition 3. ***Core-based Local Community****: Given a graph $G = (V, E)$. Given a clique size threshold of K, a density constraint γ, and a radius constraint r, a candidate core-based local community of a vertex $u \in V$ is a vertex set V', where $V' \subseteq V$, such that (1) there exists a clique (**core**) $c(u)$ of size at least K, where $u \in c(u) \subseteq V'$, (2) $R_{c(u)}(V') \leq r$ and (3) $den(V') \geq \gamma$. There can be more than one candidate core-based local community for a vertex u, one of them is assigned to u and we refer to it as the core-based local community of u, denoted by $LC(u)$.*

For simplicity we may refer to a core-based local community as local community or simply LC. The computation of a core-based local community for a vertex u can be broken down into 2 steps, the core $c(u)$ is first located, followed by an extension to a neighborhood of $c(u)$ within the radius constraint and the density constraint.

LC based Information Dissemination Model: Given a seed set S where each vertex w in S is assigned a local community LC_w under the constraints of K, γ, and r, LC_w forms a base for information dissemination by w. The spread base by seed set S is denoted as $I_S = \bigcup_{w \in S} LC_w$. The size of the spread base is given by $g(S) = |I_S|$.

Example 1. Figure 1 (a) shows a graph with 9 vertices. Let $K = 3$, the density constraint, γ, be 1, and the radius constraint, r, be 1. Suppose the seed set $S = \{g, m\}$ and $LC_g = LC(g) = \{a, b, d, g\}$, $LC_m = LC(m) = \{a, m, n\}$, then the spread base size is $g(S) = |\{a, b, d, g, m, n\}| = 6$. Note that there are other possible candidates for $LC(g)$, such as $\{a, b, g, c\}$ and $\{a, d, f, g\}$. One such candidate is set as $LC(g)$.

Example 2. Consider the graph in Figure 1 (a) again, let $K = 3$, and the radius constraint, r, be 1. If the density constraint $\gamma = 0.6$, then we may set $LC(g) = \{g, a, b, d, f, c, e\}$ since there are 15 edges in the induced graph. If $\gamma = 0.8$, then we may set $LC(g) = \{g, a, b, d, f, c\}$, since there are 12 edges in the induced subgraph.

IDM-LC Maximization Problem: The IDM problem under the LC based information dissemination model is to select k local communities so as to give the maximum value of $g(S)$, where S is the set of k seeds to which the k local communities are assigned.

It is easy to show that this problem is NP-hard since the classical maximum clique problem can be reduced to this problem by setting $k = 1$, $\gamma = 1$. In the next sections we shall examine the sub-problems involved.

3 Core-based Local Community

From previous discussions, computing a core-based local community for u consists of two steps: finding the core and extending the core. In the following, we show that these sub-problems are hard, we propose greedy algorithms for getting feasible solutions and analyze the corresponding complexity for each of the two sub-problems.

3.1 Finding the Cores

We are given a threshold of the core size K, a radius constraint r, and a density constraint of γ. First we consider the problem of finding the core for vertex u with a size of K or above. We show that this is NP-hard by showing that the decision problem of whether there exists a clique of size K is NP-hard. The proof is by a reduction from the classical maximum clique problem. A clique of size k exists in a graph G if and only if there exists in G a vertex v such that the maximum clique containing v has size k.

The maximum clique containing a vertex v is desirable for the core of v because a clique is the most densely connected subgraph with the smallest diameter. This is related to the NP-hard problem of computing the maximum clique of a graph. However, known algorithms for maximum clique cannot be adopted for two reasons. Firstly, existing heuristic algorithms are not scalable to very large graphs [3, 18]. Typically they deal with denser graphs, and their targeted graphs are small, e.g. $|V| \leq 1000$, whereas social networks have very low average degrees but $|V|$ is very large. Secondly, our problem is to find the maximum clique containing a vertex for each vertex in the graph, which is different from finding a single maximum clique for the entire graph. Here, we deal with this problem with an efficient greedy algorithm as shown in Algorithm 1.

In Algorithm 1, we maintain a clique $c(u)$ which contains u for each vertex u. Initially $c(u)$ contains only u, and more vertices are added to $c(u)$ iteratively. A vertex v is a candidate to be added to $c(u)$ if and only if v is a neighbor to every vertex in the current $c(u)$. Thus, for each vertex u, initially $c(u) = \{u\}$ and the initial candidate set *cand* consists of the neighbors of u (*line 4*). After the initialization, we iteratively select the vertex u' such that the candidate set in the next iteration is maximized (*line 6*) and update the candidate set by intersecting with the neighbors of u'(*line 7*). This maintains the invariant that each candidate is a neighbor to every vertex in $c(u)$. We repeat until no more candidate remains. Algorithm 1 selects a clique in a way to maximize the potential clique size at each vertex selection in *line 7*. Though there can be more than one eligible LC for a vertex, only one of them will be selected. In Figure 2 (a), $\{a, b, c, d\}$ will be returned as the core $c(a)$. We show that this algorithm has a scalable time complexity.

Lemma 1. *Given $G = (V, E)$, the time complexity of Algorithm 1 is given by $O(|c_{max}||V|d^2_{max})$, where $|c_{max}|$ is the maximum core size, and d_{max} is the maximum degree of a vertex.*

Proof. In each while loop, *line 6* is the most costly operation compared to *lines 7,8*. Hence we calculate the complexity of *line 6*, which involves the intersection of two sorted sets, for each while loop of $\forall u \in V$. Note that $d(u) = |adj(u)|$.

$$\Sigma_{v \in cand}(d(u) + d(v)) < \Sigma_{v \in adj(u)}(d(u) + d(v)) = d^2(u) + \Sigma_{v \in adj(u)} d(v)$$

The time complexity is analyzed as follows: $\Sigma_{u \in V}|c_{max}|*(d^2(u)+\Sigma_{v \in adj(u)}d(v)) = |c_{max}|(\Sigma_{u \in V}d^2(u) + \Sigma_{u \in V}\Sigma_{v \in adj(u)}d(v) = |c_{max}|(\Sigma_{u \in V}d^2(u) + \Sigma_{u \in V}d^2(u)) = 2|c_{max}|(\Sigma_{u \in V}d^2(u))$. Thus, the complexity is $O(|c_{max}||V|d^2_{max})$. □

Algorithm 1: SelectMC(G,K)

Input : A graph $G = (V, E)$, parameter K
Output: $C = \{c(u)$: $c(u)$ is an approximate maximum clique containing $u \in V\}$

```
1  begin
2      C ← ∅;
3      foreach vertex u ∈ V do
4          cand ← adj(u); c(u) ← {u};
5          while cand ≠ ∅ do
6              u' ← argmax_{v∈cand}{|cand ∩ adj(v)|};
7              cand ← cand ∩ adj(u');
8              c(u) ← c(u) ∪ {u'};
9          if |c(u)| < K then
10             c(u) ← ∅;
11         C ← C ∪ {c(u)};
12     return C;
13 end
```

Most existing social networks have been found to be scale-free [1, 8] and they have a highly scalable time complexity as shown below.

Lemma 2. *For a scale free network G with a parameter of γ, $2 < \gamma < 3$, the time complexity of Algorithm 1 is $O(|c_{max}||V|d_{max})$.*

Proof. From the proof of Theorem 1, the time complexity of SelectMC(G,K) is $O(2|c_{max}|(\Sigma_{u\in V} d^2(u))$. For a scale-free network, the degree distribution follows a power law. The fraction $p(k)$ of vertices in the network having degree k is given by $p(k) \approx k^{-\gamma}$, where typically $2 < \gamma < 3$.

$\Sigma_{u\in V} d^2(u) \approx \sum_{k=1}^{d_{max}} p(k)|V|k^2 \approx |V| \sum_{k=1}^{d_{max}} k^{2-\gamma} = |V| \sum_{k=1}^{d_{max}} k^{-\alpha}$, where $\alpha = \gamma - 2$ and $0 < \alpha < 1$. Since the summation in the above expression is a monotonically increasing function of k, we can bound it by $\int_{k=0}^{d_{max}} k^{-\alpha} \leq \sum_{k=1}^{d_{max}} k^{-\alpha} \leq \int_{k=1}^{d_{max}+1} k^{-\alpha} \leq \frac{1}{1-\alpha}(d_{max}+1)^{1-\alpha} = O(d_{max})$. Thus, $\Sigma_{u\in V} d^2(u) = O(|V|d_{max})$, and the complexity of $O(|c_{max}||V|d_{max})$ follows. □

Many social networks have a d_{max} much smaller than $|V|$, typically less than $\sqrt{V}$. The core size c_{max} is also very small. As reported in [19], the maximum clique sizes in their experiments are below 100. The core sizes in our experiments with real datasets are similarly small.

3.2 Extending the Cores

After getting the core for each vertex u, we next extend the core to get a core-based local community that is within the density and radius constraints. Since the goal is to maximize the spread, in this step we consider to maximize the size of the local communities. However, we show that the maximum local community

Algorithm 2: SelectLC($G, c(), \gamma, r$)

Input : A graph $G = (V, E)$, $c(u)$ for $\forall u \in V$,γ and r
Output: Set LC of core-base local communities $LC(u)$ $\forall u \in V$

```
1  begin
2      LC ← ∅;
3      foreach vertex u ∈ V do
4          cand ← {v|R_{c(u)}({v}) ≤ r, ∀v ∈ V − c(u)}; LC(u) ← c(u);
5          while cand ≠ ∅ do
6              u' ← argmax_{v∈cand}{|LC(u) ∩ adj(v)|};
7              if den(LC(u) ∪ u') < γ then
8                  break;
9              cand ← cand − u'; LC(u) ← LC(u) ∪ u';
10         LC ← LC ∪ {LC(u)};
11     return LC;
12 end
```

problem is NP-hard by a reduction from the maximum quasi-clique problem which is NP-complete [17]. Given a simple undirected graph $G = (V, E)$ and a constant $\gamma' = (0, 1)$, a subset of V is called a γ'-quasi-clique if it induces a subgraph with a density of at least γ'. We skip the proof for interest of space, the proof can be found in [23].

Theorem 1. *Given a graph $G = (V, E)$, computing the maximum local community $LC(u)$ for a vertex $u \in V$, given a core of C and a density constraint of γ, a radius constraint of r, with the clique threshold $K = 1$, is NP-hard.*

The above shows that the problem of maximizing the local community under the special condition of $K = 1$ is NP-hard, hence, the general problem where $K \geq 1$ is also NP-hard. We propose a heuristic algorithm (Algorithm 2) to extend the core for each vertex. First, find the candidate set from the radius constraint (*line 3*). Next, iteratively select the vertex u' such that u' has the largest neighborhood size with respect to the current $LC(u)$ (*line 6*). If the density is still above the threshold after adding u' (*line 7*), we include u' into the local community of u $LC(u)$, and update the candidate set (*lines 9, 10*). Note that if two vertices have the same core, we will not do the extension redundantly.

Example 3. Consider Figure 1 (b). We extend the core of $c(a) = \{a, b, c, d\}$ to get $LC(a)$. Let $\gamma = 0.7$ and $r = 1$. Initially $LC(a) = c(a)$ and $cand = \{e, f, g\}$. Pick vertex e to extend the core, since $|LC(a) \cap adj(e)| = 2$ and $den(\{a, b, c, d\} + \{e\}) = 2 * (6 + 2)/(5 * 4) = 0.8 > \gamma$. Then, $cand = \{f, g\}$. Next, pick vertex f to extend the core since $|LC(a) \cap adj(f)| = 3$ and $den(\{a, b, c, d, e\} + \{f\}) \approx 0.73 > \gamma$. $cand$ is now $\{g\}$. Finally, $den(\{a, b, c, d, e, f\} + \{g\}) =< 0.7$ if g is added. Hence, $LC(a) = \{a, b, c, d, e, f\}$. Similarly, we can get $LC(g) = \{f, g, h, i, j\}$. Note that b, c, d have the same LCs as a, and h, i, j have the same LCs as g.

Algorithm 3: SelectSeedSet($LC(u)$)

Input : $LC(u) = (V_{LC(u)}, E_{LC(u)})$ $\forall u \in V$ and k
Output: Top-k seed set S

```
1 begin
2     cand = V, S = ∅, I_S = ∅, g(S) = |I_S|, i = 0;
3     while i < k do
4         u ← argmax_{v∈cand}{|V_{LC(v)} ∪ I_S| − g(S)}; cand = cand − u;
5         w ← highest degree vertex in LC(u);
6         if w ∉ S then
7             S = S ∪ w; I_S = V_{LC(u)} ∪ I_S; g(S) = |I_S|; i++;
8     return S;
9 end
```

Lemma 3. *The time complexity of Algorithm 2 is given by $O(|V||cand_{max}|(|c_{max}|+d_{max}))$, where $|cand_{max}|$ is the maximum candidate set size, c_{max} is the maximum core size, and d_{max} is the maximum degree of a vertex, lc_{max} is the maximum size of a LC.*

PROOF: Let $cand(u)$ be the initial *cand* set. For each vertex u, the initialization at *line 4* costs $O(\sum_{v \in cand(u)}(|c(u)| + d(v)))$ time. The set *cand* can be computed by a breadth first search from u. After the initialization, The intersection size at *line 6* is computed by adding u' (in $O(1)$ time) to the intersecting set $LC(u) \cap adj(v)$ of the neighbors v of the selected u' in the previous iteration. The maximum size is obtained by a scan of the candidate set. Let $M = \min(|LC(u)|, |cand(u)|)$, processing u takes $O(\sum_{v \in cand(u)} d(v) + \sum_{j=1}^{M}(|cand(u)| - j + 1))$ time. Note that $M \leq lc_{max}$ and $c_{max} \leq lc_{max}$. Hence, summing up the above processing time for all $u \in V$ gives $O(\sum_{u \in V}\sum_{v \in cand(u)}(|c(u)| + d(v)))$ $= O(\sum_{u \in V}|c(u)||cand(u)| + \sum_{u \in V}\sum_{v \in cand(u)} d(v)) = O(|V||cand_{max}|(lc_{max} + d_{max}))$. The time complexity is $O(|V||cand_{max}|(lc_{max} + d_{max}))$. □

4 Seed Selection

After calculating $LC(u)$ for each vertex u, we aim to select a seed set S to maximize the information spread base, i.e., $g(S) = |I_S| = \bigcup_{u \in S} LC(u)$. This problem can be shown to be NP-hard by a transformation from the *Maximum coverage problem*. Here we use a greedy algorithm to select the top k seeds. At each iteration, we choose a vertex u where $LC(u)$ contains the largest number of uncovered vertices. We examine $LC(u)$ and choose the highest degree vertex, w, as the next seed, provided that w has not been chosen before. The corresponding pseudocode is shown in Algorithm 3. Algorithm 3, adding vertex u to S that maximize $g(S + u) - g(S)$ in each iteration, can be shown to attain $(1 - 1/e)$ approximation.This is because g is monotone ($g(S + v) \geq g(S)$) and submodular(diminishing return: $g(S + v) - g(S) \geq g(T + v)g(T)$, $\forall S \subseteq T$) [15].

Finally, we show that our seed selection algorithm is also efficient and can be scalable to large graphs.

Lemma 4. *Assume a new seed is picked in each while iteration, the time complexity of Algorithm 3 is given by* $O(k|V|(|LC(avg)|+|I_S|))$*, where* $|LC(avg)|$ *is the average LC size for the seeds in* S*.*

PROOF: Each seed is selected by scanning the candidate set of LC's and the current spread base. $I_S(i)$ and $cand(i)$ below refer to the spread base and candidate set before the i-th iteration, respectively. Since

$$\Sigma_{i=1}^{i=k}\Sigma_{v\in cand(i)}(|LC(v)|+|I_S(i)|) < k|V|(|LC(avg)|+|I_S|)$$

thus, the time complexity is $O(k|V|(|LC(avg)|+|I_S|))$. □

5 Application in Influence Maximization

We consider the application of information dissemination for the problem of influence maximization. One important use of influence maximization is viral marketing [7, 21]. The problem of influence maximization (IM) can be defined as follows: Given a graph $G = (V, E)$, with vertex set V and edge set E. Given a model for quantifying the influence of a vertex set. The problem is to choose a set $S \subseteq V$ of k' vertices, or seeds, to target so that the influence of S is maximized.

In Section 7, we describe the issues found with the prevalent models based on probabilistic propagations. We propose to model influence from the perspective of communities instead. Our assumption is that influence is related to information dissemination. We apply the seed set solution of the IDM problem to the IM problem. The rationale is that influence increases with information spread. Unlike previous models, we do not predict the influence since it is application dependent, e.g. the kind of products, or how contagious a disease is, etc. Instead, we aim to maximize the base for the influence.

6 Experimental Results

Our experiments are conducted on a computer with an Intel i7 CPU, 16GB RAM with Ubuntu 12.04 and implemented in C++. For the algorithms that involve randomization we repeat each experiment 1000 times and report the average result. For the Corebased method, the minimum core size K is set to 4, and the radius r is set to the default value of 1. The value of r can be set to 1 because setting r to 2 or more has little effect on the results. This is because the diameter of the LC becomes at most 5 for $r = 2$, and the diameter of a social network is typically small. Hence, when $r = 2$, the reaches become too far and the sparsity of the graph will lead to violation of the density constraint. K can be set to 4 without affecting the results. This is because 2-cliques and 3-cliques (triangles) are numerous in our datasets but for size 4 or above the clique number decreases and these cliques become significant as cohesive components.

6.1 Results on Running Time

In our first set of experiments, we verify the efficiency of our method as predicted by the runtime complexity analysis. We have tested on real graphs from Koblenz Large Network Collection (KONECT), Stanford Large Network Dataset Collection (SNAP), and Max Planck Institute collection. For our method, we convert directed graphs into undirected graphs by making each edge undirected.

	Amazon	DBLP	Epinions	Facebook	Arxiv
$\|V\|$	403394	317080	75879	63731	34546
$\|E\|$	3387388	1049866	405740	1545686	421578
$\|d_{max}\|$	2752	343	3044	1098	864

The runtime results are shown in Figure 2. The time to compute cores and extend cores are dominating. With higher degree vertices in Epinion, more vertices in the candidate set of the core can be used to extend the core, hence, more set intersection operations in Algorithm 2, and bigger LC sizes. For seed selection, we scan the communities to determine the next seed, the time complexity grows linearly with the total community size. Overall, the results show that our algorithm can efficiently handle large graphs.

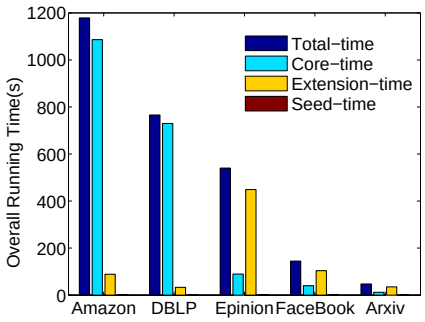


Fig. 2. Runtime of CoreBased for 5 datasets with $\gamma = 0.4$ and 50 seeds

6.2 Results on application: Influence Maximization

For the experiment on influence maximization, we follow the methodology in [10]. It is a known issue in the study of IM that it is difficult to obtain ground truth. To the best of our knowledge, [10] provides the only method for this issue. It takes a dataset with a social graph recording the friendship among a set of users and also an action log, which consists of triples (u, a, t) recording an action a by user u at time t. The action log concerning an action a is called the

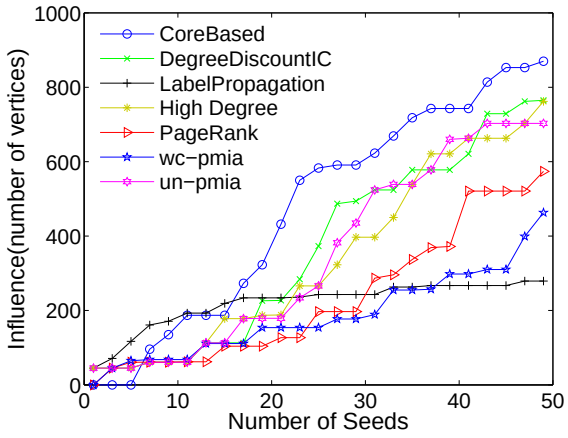


Fig. 3. Comparison of influence for different algorithms

propagation trace associated with a. E.g., an action is the rating of a movie. If user v rates a movie, and at a later time a friend u of v also rates the movie, there is a propagation from v to u. Hence, we would find (v, a, t) and (u, a, t') in the action log, where $t' > t$. In [10], such propagation from a user to the friends of the user is considered the ground truth.

For the Facebook dataset of New Orleans from socialnetworks.mpi-sws.org, there are two components, one contains a list of user-to-user links and the second is from a list of wall postings, both lists contain a UNIX timestamp for each link. The second component is thus the action log. We consider each wall posting an action. The friendships among users are obtained from the user-to-user links. Suppose u is in the seed set, then on the wall of u, we take the posting by u to be the initiating action. The influence of u is the number of friends of u that have posted on u's wall after u's own posting.

Since our solution is related to community search, we compare with existing community search methods. We consider the highest degree vertex as a potential seed in each computed community. In the recent survey in [6], 40 community discovery methods are listed, however, very few of these methods have scalable complexity. We have selected the only two scalable algorithms reported to handle large graphs under the category of "Diffusion", which is related to information dissemination. The two methods are Label Propagation [20] and DegreeDiscountIC [5]. Our method is tailored for maximizing the spread base, it is not fair to compare the spread base. Instead, we compare by the measure of influence.

We also compare with the following methods: High Degree, PageRank, un-pmia [4], and wc-pmia [4]. High Degree is a baseline approach by select k vertices with the highest degrees as the seeds. pmia is a scalable algorithm that is shown to be effective for the IC model for the IM problem [4], wc-pmia adopts the WC model while un-pmia is pmia with uniform probability of 0.01 at each edge. We adopt the settings for PageRank and DegreeDiscountIC from [4].

The results are shown in Figure 3. Our method consistently produces the highest influence among all the other methods in the experiment. Although it is difficult to locate more datasets for similar comparison, the results here provide evidence that our approach based on the principles of homophily has great potential to outperform the existing models and methods.

7 Related Work

For LCs, we compute a clique for each vertex, this is related to the maximum clique problem, since the maximum clique must be the maximum clique for one of the vertices. The maximum density-based quasi clique problem is proved to be NP-complete by reducing from the classical clique problem in [17]. [2] utilizes a existing framework known as a Greedy Randomized Adaptive Search Procedure (GRASP), which consists of initial construction and local optimization in each iteration. However, these algorithm only return one quasi-clique, while our problem requires one clique for each vertex. Another related work [22] is to find a subgraph of G that contains a given set of query nodes and which is densely connected. Unlike conventional approaches, [22] seeks the dense subgraph containing the query nodes. The problem is to maximize the minimum degree with size restriction. The authors propose a greedy algorithm to solve their problem in $O(|V| + |E|)$ time.

IM has been studied under the IC and LT models [11]. [4] proposes a scalable algorithm for the IC model with a parameter on the influence probability for early stopping. Graph sparsification is introduced in [13] with edge pruning during influence propagation. While the two models are widely adopted and initiated a lot of interesting works, they are not validated with ground truth. Some recent studies have found some critical issues with these models. In the viral marketing study in [12], it is shown that the real world does not match these models. An underlying assumption of the models is that no distance, time, or capacity limit is attached to the influence. However, in the viral marketing study in [12], it is found that a high-degree vertex has a limit in its influence. They conclude that individuals tend to "have influence over a few of their friends, but not everybody they know". Consider a star graph with a center vertex v_0 being linked to n other vertices by edges $(v_0 \rightarrow v_i)$, $1 \leq i \leq n$. If v_0 is chosen as a seed, then all n vertices are influenced with probability 1. If n is large, such influence would not be possible according to the results in [12].

Another issue shown in the study in [12] is a general tendency of influence to terminate after just a short number of steps. Consider a line graph, where a vertex v_0 is linked to n other vertices $v_1, ..., v_n$ via edges $(v_i \rightarrow v_{i+1})$. With the WC model all the vertices will be influenced by v_0 with a probability of 1, which is also not realistic if n is large. The detailed study in [12] also finds that the probability of infection decreases with repeated interactions, which in this model corresponds to time. Also it is found that the probability of being influence will increase with the number of active neighbors initially but will saturate after a certain point and more active neighbors will not have any further

effect. These contradict the assumption in the LT model that each vertex has the same threshold, hence, the probability of being infected remains an invariant.

Similar findings are reported in [10], a scatter plot between influence predicted by the models and the actual influence on a real dataset shows a big discrepancy where the predicted influence is many times higher than the actual value.

8 Conclusion

For future work, we may consider the issue of overlapping communities. The *LC* model can be extended to include multiple communities for each seed. Another extension is to consider directed graphs in community search, which is an interesting problem in general. To our knowledge the IDM problem is a new problem and has important applications in IM. Most existing IM studies rely on propagation models shown in some recent works such as [12, 10] to be problematic. Most works assumed that such a model is the ground truth and results were compared only based on measurements defined by such models. Our study deviates from this trend and shows that the homophily based IDM model for IM can produce better results in a case study. More study will be needed for this approach, but the methodology is sound and we believe that this can be a promising new approach.

References

1. http://knoect.uni-koblenz.de/networks
2. Abello, J., Resende, M.G., Sudarsky, S.: Massive quasi-clique detection. In: LATIN 2002: Theoretical Informatics, pp. 598–612. Springer (2002)
3. Bomze, I., Budinich, M., Pardalos, P., Pelilo, M.: The maximum clique problem. Handbook of Combinatorial Optimization A, 1–74 (1999)
4. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: 16th SIGKDD. pp. 1029–1038 (2010)
5. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: 15th SIGKDD. pp. 199–208. ACM (2009)
6. Coscia, M., Giannoti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. Journal of Statistical Analysis and Data Mining 47(11), 41–45 (Nov 1997)
7. Domingos, P., M.Richardson: Mining the network value of customers. In: 7th SIGKDD (2001)
8. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: SIGCOMM (1999)
9. Fortunato, S.: Community detection in graphs. Physical Reports 486, 75–174 (2010)
10. Goyal, A., f. Bonchi, Lakshmanan, L.: A data-based approach to social influence maximization. In: VLDB (2011)
11. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: ninth SIGKDD. pp. 137–146. ACM (2003)

12. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Transactions on the Web 1(1), 1–39 (2007)
13. Mathioudakis, M., Bonchi, F., Castillo, C., Gionis, A., Ukkonen, A.: Sparsification of influence networks. In: 17th SIGKDD. pp. 529–537. ACM (2011)
14. Michael, J.: Labor dispute reconciliation in a forest products manufacturing facility. Forest Products Journal 47(11), 41–45 (Nov 1997)
15. Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of the approximations for maximizing submodular set functions. Mathematical Programming 14, 265–294 (1978)
16. de Nooy, W., Mrvar, A., Batagelj, V.: Exploratory Social Network Analysis with Pajek. Cambridge University Press, Cambridge, UK (2005)
17. Pattillo, J., Veremyev, A., Butenko, S., Boginski, V.: On the maximum quasi-clique problem. Discrete Applied Mathematics 161(1), 244–257 (2013)
18. Pullan, W., Franco, M., Mauro, B.: Cooperating local search for the maximum clique problem. J. Heuristics 17, 181–199 (2011)
19. Pullan, W., Hoos, H.H.: Dynamic local search for the maximum clique problem. Journal of Artificial Intelligence Research 25, 159–185 (2006)
20. Raghavan, U., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical Review E 76 (2007)
21. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: KDD (2002)
22. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: 16th SIGKDD. pp. 939–948. ACM (2010)
23. Yang, Z., Fu, A., Xu, Y., Huang, S., Leung, H.: Community based information dissemination. Technical Report, CSE, CUHK, http://www.cse.cuhk.edu.hk/ adafu/paper/cbid.pdf (2015)

9 Appendix

The following lemma is used in the proof of Theorem 1.

Lemma 5. *Given integers $n \geq 2$, $K \leq n$ and $x \geq K'$, then*

$$\frac{2(2n^2+n^2)}{(2n^2+n-1)(2n^2+n-2)} \leq \frac{2(n^2+x)}{(n^2+K')(n^2+K'-1)}$$

Proof. We need only show that

$$(n^2+x)(2n^2+n-1)(2n^2+n-2) \geq (3n^2)(n^2+K')(n^2+K'-1)$$

Since $x \geq K$, the above inequality holds if for $b \geq 0$,

$$(n^2+K'+b)(4n^4+4n^3-5n^2-3n+2) \geq (n^2+K')(3n^2)(n^2+K'-1)$$

Since $K' \leq n$, the above inequality holds for $b \geq 0$ if

$$(n^2+K'+b)(4n^4+4n^3-5n^2-3n+2) \geq$$
$$(n^2+K')(3n^2)(n^2+n-1) = (n^2+K')(3n^4+3n^3-3n^2)$$

It remains to show that

$$\alpha = (4n^4+4n^3-5n^2-3n+2) - (3n^4+3n^3-3n^2) \geq 0$$
$$\alpha = n^4+n^3-2n^2-3n+2 = n^2(n^2-2)+n(n^2-3)+2$$

Since $n \geq 2$, $\alpha \geq 0$, which completes the proof. □

Theorem 1: *Given a graph $G = (V, E)$, computing the maximum local community $LC(u)$ for a vertex $u \in V$, given a core of C and a density constraint of γ, a radius constraint of r, with the clique threshold $K = 1$, is NP-hard.*

Proof. Since $K = 1$, the core C must contain a single vertex u. The proof is by reduction from the maximum quasi-clique decision problem to the decision problem for the maximum local community. The maximum quasi-clique decision problem is: Given a simple undirected graph $G = (V, E)$, a positive real number γ' satisfying $0 < \gamma' < 1$, and an integer K', does there exist a γ'-quasi-clique of size at least K' in G ?

Given an instance of the maximum quasi-clique decision problem with parameters γ' and K'. Let x be the smallest integer satisfying $2x/(K'(K'-1)) \geq \gamma'$. If $x < K'$, the quasi-clique is a tree and it is trivial. Hence, in the following, we only consider instances where $x \geq K'$.

We transform the given instance into a LC instance as follows: Let $n = |V|$. Create a vertex a, and for each vertex $v_i \in V$, create $n^2 - 1$ vertices $b_{i1}, b_{i2}, ..., b_{i(n^2-1)}$, two edges (a, b_{i1}) and $(b_{i(n^2-1)}, v_i)$ and also $n^2 - 2$ edges of the form $(b_{ij}, b_{i(j+1)})$, $1 \leq j \leq n^2 - 2$. A new graph $G' = (V', E')$ is thus created containing vertices in V and all the above new vertices, and edges in E and all new edges created. Hence a is linked to each vertex v in V via a path with n^2 edges. Let us call this path the ***a*-path for** v.

The problem is to find $LC(a)$, with a density threshold of

$$\gamma = \frac{2(n^2 + x)}{(n^2 + K')(n^2 + K' - 1)}$$

Also, $r = n$, and the size threshold is $K'' = (n^2 + K')$.

Suppose there exists a γ'-quasi-clique C in G of size at least K' and density at least γ'. It is easy to see that we can select any vertex v in C, and C together with the a-path will be a $LC(a)$ in G' satisfying all the threshold requirements.

Conversely, given a solution S to the $LC(a)$ problem. Firstly, S must contain at least one a-path because of the size threshold K'' and the graph construct. We show that S contains at most one a-path for some vertex v. If S contains 2 or more a-paths then the density D_S of S is given by

$$D_S \leq \frac{2(2n^2 + n^2)}{(2n^2 + n - 1)(2n^2 + n - 2)}$$

From Lemma 5, D_S is smaller than the threshold γ when $n \geq 2$. This contradicts the fact that S is a solution for $LC(a)$. If S contains at most one a-path then we can form a solution C for the γ'-quasi-clique problem by computing $S \cap V$.

□