Exploring Community Structures for Influence Maximization in Social Networks

Yi-Cheng Chen, Su-Hua Chang, Chien-Li Chou, Wen-Chih Peng and Suh-Yin Lee

Department of Computer Science

National Chiao Tung University

Hsinchu, Taiwan 300

ejen.cs95g @nctu.edu.tw {wasiseal, fallwind, wcpeng}@cs.nctu.edu.tw sylee@csie.nctu.edu.tw

ABSTRACT

Since the surge of the popularity of social network, recently, there has been a tremendous wave of interest in the investigation of influence maximization problem. Given a social network structure, the problem of influence maximization is to determine a minimum set of nodes that could maximize the spread of influences. Nowadays, due to the dramatic size growing of social network, the efficiency and scalability of algorithms for influence maximization become more and more crucial. Although many recent studies have focused on the problem of influence maximization, these works, in general, are time consuming when a large-scale social network is given. In this paper, by utilizing community structures, we develop two efficient algorithms *CDH-Kcut* and *CDH-SHRINK* (standing for *Community and Degree Heuristic with Kcut / SHRINK*) that significantly decrease the number of candidate influential nodes. The experimental results on real datasets indicate that our algorithms not only significantly outperform state-of-the-art algorithms in efficiency but also possess graceful scalability.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and behavioral sciences; H.2.8 [**Data Management**]: Database Applications - Data Mining

General Terms

Algorithms

Keywords

community discovery, diffusion models, influence maximization, social network

1. INTRODUCTION

In more recent years, social network analysis has drawn much attention due to its widespread applicability. A social network is a social structure made up of individuals who are tied by one or more specific types of relationship or interdependency, such as friendship, co-authorship, common interest or financial exchange, to name a few. Nowadays, many worldwide social-networking web sites, such as Facebook and Twitter, are very popular since users can share their thoughts and comments with their friends and also bring small and disconnected social networks together. Hence, marketing on online social networks shows great potential to be much more successful than traditional marketing techniques. In many enterprises, the budget of advertisement spending on worldwide social networking sites is almost the same or even in excess of that spent in traditional ways.

For example, a company has developed a mobile application and advertised it on a social network. The company gave the trial to a small number of users and hoped that these users could influence their friends to buy the application; their friends could influence their friends' friends. Through the word-of-mouth effect, the company could make a large number of users buy the application. The ***influence maximization*** problem [7] aims to select initial users (referred to as *seeds*) so that the number of users that adopt the product or innovation is maximized. That is, the problem is how to find the influential individuals in a social network.

Since many social networks become large-scale, developing efficient algorithms for influence maximization is more and more critical. Kempe et al. [13] proved that the influence maximization problem is *NP-hard*. If it takes a week for companies to decide which set of individuals should be given free samples to promote their products, they may lose their superiority because of non-timeliness. Moreover, the selected set of individuals will not be useful since the network may change significantly during this week. As such, timeliness is an important issue for the influence maximization problem.

Some efficient approximate algorithms have been proposed [4, 10]; however, although efficient, they are only applicable to some diffusion models which are not realistic for modeling influences in social networks. Different social networks may have different types of influences. For example, sometimes we want to know which set of individuals could trigger more adoptions of products after 3 days, 7 days, or a month. How to model the influences in social networks is very important. Hence, the diffusion model of influence is another issue for the influence maximization problem. A realistic diffusion model is essential for making correct predictions of the future behavior of the network.

The contributions of this paper are as follows,

- By our observation, a common property of social network is that nodes tend to cluster together. We claim to utilize

The 6th SNA-KDD Workshop'12 (SNA-KDD'12), August 12, 2012, Beijing, China. Copyright 2012 ACM 978-1-4503-1544-9…$15.00.

the community information for the selection of seed nodes of influence spread.

- Based on realistic model, two novel algorithms, *CDH-Kcut* and *CDH-SHRINK* (*Community and Degree Heuristic on Kcut / SHRINK*), are proposed to solve the problem of influence maximization efficiently.
- We consider and utilize the overlapping community characteristics (hubs and outlier) to deal with the influence maximization problem.
- Although, based on a realistic model, the mechanism used by CDH can totally avoid the greedy selection of seed nodes which consumes heavy computation. Hence, we tackle the issues of efficiency and scalability for influence maximization problem.

Since most social networks are larger and larger, an efficient and scalable algorithm becomes very critical. The experimental studies on several real datasets indicate that CDH is not only efficient but also have good influence spreads.

The remainder of this paper is organized as follows. Section 2 present related works. Section 3 details the framework of CDH and two proposed algorithms, CDH-Kcut and CDH-SHRINK. Section 4 presents the experiments on real datasets. Finally, we conclude this paper in Section 5.

2. RELATED WORKS

Formally speaking, a social network is generally modeled as an undirected graph $G(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the vertex set and $E = \{(v_i, v_j) \mid$ there is an edge from v_i to $v_j\}$ is the set of edges. A node represents an individual, and an edge between two nodes represents some kind of relationship (friendship or co-authorship, etc.). A node is marked as active if it has adopted an idea or an innovation, or as inactive if it has not. Thus, the problem of influence maximization is given below:

(Influence Maximization) Given a social network $G = (V, E)$, the output is to determine a set of seeds (i.e., nodes) such that these seeds could spread their influence to other nodes with the purpose of maximizing the number of nodes affected by the seeds.

2.1 Diffusion model

Rogers [16] theorizes that diffusion is the process by which an innovation is communicated through certain channels over time among the members of a social system. Diffusion is a type of communication concerned with the spread of messages that are perceived as new ideas.

The naïve diffusion model is Linear Threshold Model (LTM). For an undirected graph $G(V, E)$, $N(v) = \{u \mid (u, v) \in E\}$ is defined as the neighbor set of node v and b_{uv} is defined as the influence of active node u on its inactive neighbor v. We define $A(v)$ as the set of active nodes in $N(v)$, ($A(v) \subseteq N(v)$). Given an activation threshold θ, for a node v, if $\sum_{u \subseteq A(v)} b_{uv} \geq \theta$, node v becomes active. Intuitive meaning is that for an inactive node v, if the total influence exerted by all its active neighbors exceeds a pre-defined activation threshold θ, node v becomes active.

Another fundamental diffusion model is independent cascading model (ICM) [11]. If a node u is activated at step t, it will try to activate its inactive neighbor v with success probability p. If it is successful, then v will be active in step t+1, else u is failed and will no longer have chance to activate v. In addition, each active node has only one chance to activate its neighbor.

Heat diffusion model (HDM) [14] is a realistic model to simulate the social behavior. The heat flow is actually similar to the process of people influencing others. The innovators and early adopters of a product or innovation act as heat sources, and process a very high amount of heat. These people start to influence others.

In HDM, the value $f_i(t)$ describes the heat at node v_i at time t, beginning from an initial distribution of heat given at time zero. Supposed at time t, each node v_i receives an amount of heat from its neighbor v_j during a period Δt. The heat should be proportional to the time period Δt and the heat difference $f_j(t) - f_i(t)$. As a result, the heat difference at node v_i between time t and $t + \Delta t$ will be equal to the sum of the heat that it receives from all its neighbors. This is formulated as, $\frac{f_i(t+\Delta t) - f_i(t)}{\Delta t} = \alpha \sum_{j:(vj, vi) \in E} (f_j(t) - f_i(t))$ $= \alpha H f(t)$, where H is a matrix and α is the heat diffusion coefficient. Heat diffusion model can easily simulate time effect in information and different types of information flow since non-activated nodes still can spread information. If the amount of heat of node v exceeds the activation threshold θ, we think node v purchases a product or adopt an innovation.

In reality, it is not reasonable that only activated nodes could spread information. ICM and LTM are built at a very coarse level, typically with only a few global parameters, and are not useful for making correct predictions of the future behavior of the network [6, 8]. HDM provides more parameters to simulate the conditions of real world, such as time and thermal conductivity. Some other models [10, 19] have been proposed, but all of them are the variations of two core models, LTM and ICM.

2.2 Influence Maximization Algorithm

Some previous works have been proposed to achieve approximate solutions. The intuitive strategy, in general, is selecting seeds based on their degrees, called *degree centrality*. Nevertheless, the members of large communities often have larger degree than other members of smaller communities. Consequently, degree centrality can easily select seeds in the same large community. Influence spreads (the number of influenced nodes) of each seed in the same community tend to be overlapped. As a result, degree centrality does not have good performance on influence spread.

The set cover greedy algorithm [10] was developed based on ICM model. It keeps selecting node with highest "uncover degrees". Once a node is selected, all its neighbors as well as itself are labeled as "covered". This procedure continues until k seeds are selected. However, it has good influence spread only in high success probability. The climbing-up greedy algorithm [13] was proposed based on ICM and LTM models with approximation guarantees for influence spread. It selects most "influential" node on the condition of considering all the seeds selected before. For selecting the most influential node, we have to compute each node's influence until required k seeds are selected. Due to the heavy computing load, climbing-up greedy algorithm is not appropriate for large social networks.

Enhanced greedy algorithm [14] was proposed based on heat diffusion model. Same as climbing-up greedy algorithm, we cannot solve influence maximization problem under heat

diffusion model in acceptable time. The potential-based node selection method [20] was proposed to select some inactive nodes that might not be optimal at starting phase but could trigger more nodes in later stage of diffusion. It can save half time of c*limbing-up greedy algorithm* and cause more adoptions than that in [13]. However, in practice it is still not efficient enough in on-line social networks.

Based on the variation of ICM, Saito et el. [18] constructed a layered graph approach and applied the bond percolation with two control strategies, pruning and burnout, to solve influence maximization problem. The extremely efficient algorithm, *degree discount heuristic,* was presented in [4]. It obtains the approximate solutions in large datasets for only a few seconds. However, both of [4, 20] are only under LTM or ICM, which are not very realistic diffusion models. In addition, degree discount heuristic is only for very low successful probability, i.e., people are extremely hard to be influenced in very low success probability.

2.3 Community Structure

A community is characterized as a subset of individuals who interact with each other more frequently than other individuals outside the community [21]. Community discovery is similar but not equivalent to the conventional graph partitioning problem. Both community discovery and conventional graph partitioning problem aim to cluster vertices into groups. A key challenge for the former, however, is that the algorithm has to decide what is the "the best", or in other words, the "most natural" partition of a network. "Most natural" partitioning method means that we need not give any heuristic information. Furthermore, if there is no good community structure, then there is no need to partition the network. That is why we use the community detection algorithm rather than conventional graph partitioning algorithm.

A quantitative measure, called modularity (Q), was to assess the quality of community structures, and the community discovery was formulated as an optimization problem. Since optimizing Q is an NP-problem, several heuristic methods have been proposed, as surveyed in [5]. Assume M is the number of edges and N is the number of nodes. The time complexity of most community detection algorithms [3, 5, 9, 12, 17, 22] are between $O(N \log N)$ and $O(N^3)$. In this paper, the efficiency of algorithms are most concerned, so we select Kcut [17] and SHRINK [12], which have low time complexity $O(M \log N)$ and good modularity, as our community detection algorithms. SHRINK algorithm also can detect hubs which are very useful for influence maximization problem.

3. ALGORITHMS of CDH-Kcut and CDH-SHRINK

In this section, we first present the design of algorithm CDH. Then, based on the community detection algorithms used (i.e., Kcut and SHRINK), we propose two algorithms, CDH-Kcut and CDH-SHRINK.

3.1 The Design of Algorithm CDH

Given a social network G with N individuals and a quota number k, based on the heat diffusion model, CDH (*Community and Degree Heuristic approach*) utilizes the community characteristics and then selects the initial k "influential" individuals to maximize the number of individuals who are influenced by the set of selected k individuals.

Initially, we select k individuals as seeds, denoted by the set $S = \{s_1, s_2, \ldots, s_k\}$, and the k seeds are given a certain amount of heat h_0, and then find the influence spreads, i.e., the number of influenced nodes, based on Heat Diffusion Model (abbreviated as HDM) with G and S. At time t_0 of the heat diffusion process, we set $f_i(t_0) = h_0$ for $v_i \in S$. As time elapses, the heat will diffuse through the whole social network. If the amount of heat of individual v_i at time t is larger than or equal to an activation threshold θ, this individual v_i will be considered as having been successfully influenced or activated by others. Denote the set of influenced nodes of S as $I_s(t)$, i.e., the expected number of individuals who will adopt the product at time t. Now the ***Influence Maximization Problem*** could be interpreted as: finding the most influential k-size set S to maximize the size of set $I_s(t)$ at time t, where $I_s(t) = \{ i \mid f_i(t) \geq \theta, i \leq N \}$. This problem is NP-hard, as already proven in [13]. We select the HDM to be our diffusion model since it can realistically simulate a real social network [17].

The proposed CDH is composed of two phases, the *partition phase* and the *selection phase*. The ***partition phase*** detects the communities of the social network, where a community is a subset of individuals who interact with each other more frequently than other individuals outside the community. Based on the communities discovered, in the ***selection phase*** phase, we propose mechanisms to select seed nodes. Explicitly, in real life, one's information often spread in his/her circle of friends. In other words, most of someone's influence clearly spread in his/her community. We also can find the same phenomenon in heat diffusion model. Fig. 1 illustrates the framework of CDH.

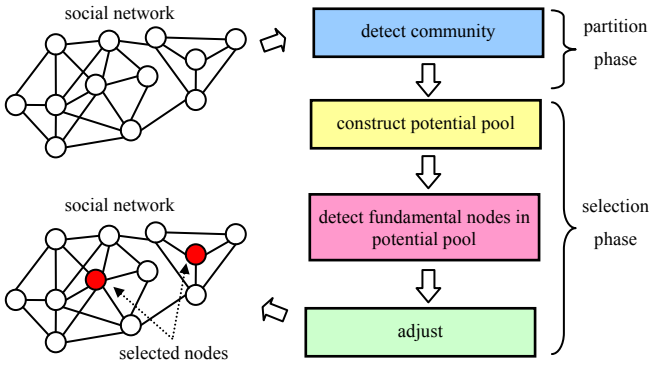
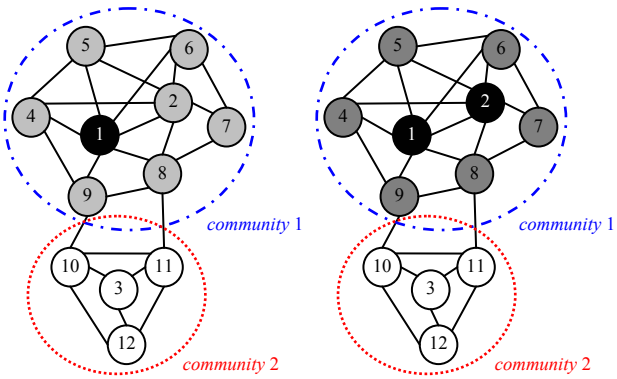


Fig. 1: Overview of CDH algorithm

For example, in Fig. 2, the color of each node means the amount of heat; more dark blue means a larger amount of heat. Nodes in the same community are captured by dotted circle. As in Fig. 2(a), if choosing node 1 as the seed, we can see that most gains of heat are the nodes in node 1's community. However, if choosing node 1 and node 2 as the seeds, most heat is spread around node 1's community, as in Fig. 2(b). Nodes in the other community gains very little amount of heat. We can conclude that if we choose many nodes in the same community as seeds, most gains of heat are in their own community; the nodes in other communities gain little amount of heat. Therefore, information of community is very useful to avoid the influence of overlapping in heat diffusion model.

The reason for using community detection algorithms rather than conventional graph partitioning algorithm is that we intend to detect "the best", or in other words, the "most natural" partitioning of a network without providing any heuristic information, such as the number of partitions. For example, if it is natural to partition the network into 3 communities, we should not force the network to be partitioned into 4 communities.

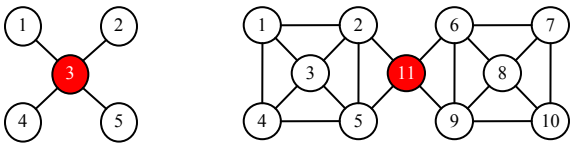


(a) The distribution of heat as node 1 is seed

(b) The distribution of heat as node 1 and 2 are seeds

Fig. 2: Examples of seed node selection.

Based on the result of *partition phase* and some parameters of heat diffusion model, i.e., flow duration, thermal conductivity and activation threshold, the second phase, ***selection phase***, finds the most influential nodes. For example, in Fig. 2(a), community 1 is larger than community 2. We can observe that selecting nodes in community 1 as seeds instead of nodes in community 2 could trigger more individuals to be activated. The degree of each node in a social network also fit the power-law distribution [1, 2], i.e., a very large number of nodes have very small numbers of neighbors. Hence, most large-degree nodes are in large communities. Due to this reason, we only consider nodes in the large communities as seed candidates and put them in a potential pool.



(a) Node 3 is a fundamental node

(b) Node 11 is a fundamental node

Fig. 3: Examples of fundamental node selection.

Then, we detect the "fundamental nodes" from the potential pool. A fundamental node indicates that this node has more potential to be a seed due to 1) larger degree than other nodes in the same community, or 2) location on the important position in the network. The important position means connecting many communities. Fig. 3(a) and 3(b) show two kinds of fundamental nodes. As in Fig. 3(a), node 3 is a fundamental node since it has the largest degree among all nodes. In Fig. 3(b), node 11 is a fundamental node since it has better position which can easily influence two sets {1, 2, 3, 4, 5} and {6, 7, 8, 9, 10}. How to detect the fundamental nodes is one of differences between proposed CDH-Kcut and CDH-SHRINK algorithms.

Although fundamental nodes are likely to become the final seeds, they are not the best seeds in different situations (parameters) of heat diffusion model. For example, seeds which perform well in a short flow duration may not be good in a long flow duration. Therefore, adjusting the fundamental nodes to become more significant seeds is very important.

3.2 CDH-Kcut

CDH-Kcut utilizes Kcut algorithm to first discover communities in the partition phase. Note that Kcut algorithm [17] combines the recursive partitioning and direct k-way method based on the eigenvectors of the Laplacian matrix of a graph. It achieves the efficiency of a recursive approach, while also having the same accuracy as a direct k-way method. If there are multiple communities, using multiple eigenvectors to directly compute a k-way partition is better than recursive bi-partitioning method. Every node in social network graph G will belong to only one community, and overlapping community is not allowed in Kcut. We assume G is partitioned to l communities. In most cases, l is larger than k, so in this paper we do not discuss the case $l < k$. For more details of Kcut algorithm, interesting readers could refer to [17].

Fig. 4 illustrates the pseudo code of CDH-Kcut algorithm. First, we select top-k biggest communities from the discovered communities by Kcut, denoted as SC_i, $1 \le i \le k$ (lines 1-2, algorithm 1). Then we construct a potential pool, $PP(G)$ (lines 3-5, algorithm 1). Potential pool is defined as, $PP(G) = \{SC_1^p, SC_2^p, \ldots, SC_k^p\}$, where SC_i^p is the set of top-p degree nodes in i-th largest community SC_i. $PP(G)$ keeps the top-p degree nodes in each community of top-k largest communities. In most cases, p = 10% of community size is enough for selecting good seeds.

Then the fundamental nodes are selected from the potential pool. Since Kcut cannot identify the important location of nodes in each SC_i^p, degree has been considered as the only attribute that distinguishes good fundamental nodes. We select the largest degree node in each SC_i^p as the set of fundamental nodes $S = \{s_1, s_2, \ldots, s_k\}$ (lines 6-8, algorithm 1). By potential pool and fundamental nodes constructing, we can significantly narrow down the range of possible seeds.

Moreover, we adjust the fundamental nodes to be the final seeds. Our basic idea of adjustment is a heuristics that tries to use an add-node, *a_node*, to replace a delete-node, *d_node*. By our observation, seeds selected from a large community could trigger more adoptions of a product or an innovation than seeds selected from a small community. For example, in Fig 3(a), the community 1 is a large one and community 2 is a small one. If the size of a community is larger than $AvgSC = \text{avg}(\Sigma_{i=1}^{k} \text{size}(SC_i))$, this community is deemed as a large community (lines 10-13, algorithm 1). As a result, we incline to select add-node from large communities and delete-node from small communities. We would like to investigate whether selecting nodes from large communities can gain more influence spread or not (lines 17-20, algorithm). If the number of influenced nodes after replacing is larger than before replacing, we replace the delete-node, *d_node* with add-node, *a_node* in S (lines 21-24, algorithm 1). Notice that delete-nodes must be fundamental nodes, i.e., *d_node* is selected

from S. Finally, we output the nodes in set S as the selected seeds (line 27, algorithm 1).

Algorithm 1: *CDH-Kcut* (G, k, p)

Input: Graph of social network G ; number of total seeds k ; parameters p

Output: k seeds

```
01:  call Kcut (G );
02:  S ← ∅;  //  seed set
03:  select top-k biggest communities from the communities in Kcut (G );
04:  for each selected community SC_i do
05:     add top-p degree nodes into set S C_i^p ;
06:  for each S C_i^p  do
07:     S ← S ∪ the most degree node in S C_i^p ;
08:  I_s(t) ← execute HDM on G with S;  //I_s(t): the set of influenced nodes
     of current S
09:  IM ← |I_s(t)|;  // IM : save the max number of influenced nodes
10:  for each community SC_i do
11:     if size(SC_i) > avg( Σ_{i=1}^{k} size( SC_i ) ) then
12:        add SC_i in LC;  // LC : the set of large communities
13:  sort LC based on community size;
14:  di ← 0;  // di: index of d_node
15:  for each SC_i in LC
16:     ai ← 2;  // ai: index of a_node
17:     while true do
18:        select the ai-th large degree node from S C_i^p  as a_node;
19:        select the seed candidates s_{k-di} from S as d_node;
20:        replace the d_node with a_node in S;
21:        I_s(t) ← execute HDM on G with S;
22:        if |I_s(t)| < IM then
23:           restore the replacement in line 20;
24:           break;
25:        IM ← |I_s(t)|;
26:        ai ← ai +1; di ← di +1;
27:  output S;
```

Fig. 4: Pseudo codes of CDH-Kcut algorithm

The adjustment is to avoid influence spread being spoiled from the effect of different value of parameters, such as flow duration, activation threshold and thermal conductivity. We discuss the effect of flow duration, activation threshold and thermal conductivity individually. Comparing the different effect by time duration, information will diffuse farther in long flow duration. That is, in long flow duration the seeds would influence more individuals than that in short flow duration. Therefore, we should not select too many seeds from a single community in long flow duration. In contrast, it is appropriate selecting more seeds from a single community if flow duration is very short.

It is more difficult to make individuals adopt products if activation threshold is high. Individuals need more heat to be activated in higher activation threshold, so we tend to select more seeds in one community with high activation threshold. High thermal conductivity makes information diffuse more quickly. Compared with low thermal conductivity, information of high thermal conductivity makes information diffuse longer distance. Hence, we do not select many seeds from one community in high thermal conductivity. We conclude that different parameters may cuase different level of seed clustering. It is better to select more seeds in one community, i.e., higher level of seed clustering, with short flow duration, high threshold and low thermal conductivity. On the contract, in long flow duration, low threshold and high thermal conductivity social networks, not many seeds in a single community are needed, i.e., lower level of seed clustering. Therefore, the adjustment in CDH-Kcut is to test and verify whether large communities should need more seeds

3.3 CDH-SHRINK

Same as CDH-Kcut, CDH-SHRINK also consists of two phases, partition phase and selection phase. However, SHRINK [12] is utilized to discover communities in partition phase. SHRINK is a parameter-free hierarchical network clustering algorithm combining the advantages of density-based clustering and modularity optimization methods. It uses density-based method to quickly discover which set of nodes may be the in the same cluster, and then uses modularity optimization to decide whether results of clustering are good or not. SHRINK not only detects hierarchical communities, but also identifies hubs and outliers. For community partition, a *hub*, i.e., a node connecting different communities, can provide more information about the community structure property. As the example in Fig. 5, node 5 is a hub connecting communities 1 and 2. A brief glance of SHRINK algorithm is introduced here and please refer to [12] for more details. Since the communities detected by SHRINK are more precise than by Kcut, we can select more productive fundamental nodes.

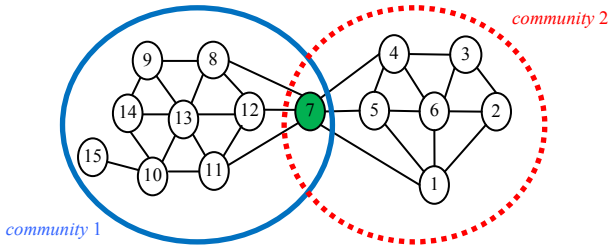


Fig. 5: Illustration of community size reduction.

In the selection phase, we construct the potential pool and select fundamental nodes, and then adjust fundamental nodes to find final seeds. Since hub can provide more information about community structure, constructing potential pool and selecting fundamental nodes is quite different from CDH-Kcut. If we have to find k seeds, we need k iterations of selecting fundamental nodes. In the i-th iteration, $1 \le i \le k$, we only choose largest community SC_i among all remaining communities and select top-p degree nodes in SC_i, denoted as $S\,C_i^p$, for potential pool construction, and then select a fundamental node from $S\,C_i^p$. By the observation of experiments, p = 10% of SC_i's size is sufficient to select good seed nodes. After selecting a fundamental node in $S\,C_i^p$ (how to select will be discussed later), the size of each un-chosen community covered by this fundamental node has to reduce the degree of this fundamental node. For example, as shown in Fig. 5, assume that node 5 is the fundamental node in community 1 (size = 10) in 1st iteration. In 2nd iteration, when we choose the biggest community, the size of community 2 (size = 7) will reduce the degree of node 7 in community 2, i.e., 7 − 3 = 4. Community size reduction can effectively avoid the influence overlapping.

Now we discuss how to select a fundamental node in potential pool. A good fundamental node should "cover" communities as much as possible while having as much influence on its

communities as possible. In Fig. 6, node z belongs to community C_1, C_2 and C_3. That is, z covers C_1, C_2 and C_3. We define two evaluation metrics, *position score* and *hub purity*, to measure how good a fundamental node is.

Definition 1 (Position Score and Hub Purity) To evaluate the importance of a node's position in a network, ***position score*** of a node u is defined as the number of communities which u belongs to, i.e., *position_score* (u) = |{C_i | $u \in C_i$, $u \in V$ and $C_i \in$ set of discovered communities}|. If the node u is a non-hub node, the *position_score*(u) is 1. Otherwise, the *position_score*(u) is larger than 1. The ***hub purity*** of a node is defined as follows,

$$hub\_purity\ (u) = \frac{|\{C_i \mid u \in C_i, u \text{ is a } hub \text{ and } C_i \notin FC\}|}{position\_score(u)}$$, where FC is the set of communities which contain fundamental node u and u is a hub. For example, in Fig. 8, C_1 and C_2 are communities containing fundamental node x. C_1, C_2 and C_3 are communities containing node z. C_2 and C_4 are communities containing node y. Therefore, *position_score*(z) = 3, *hub_purity* (z) = 1/3 and *position_score*(y) = 2, *hub_purity* (y) = 1/2.

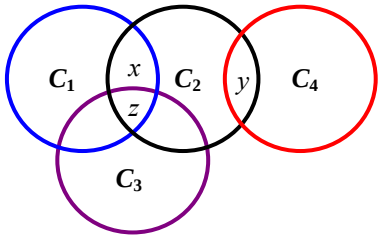


Fig. 6: An example of how to compute purity. C_1, C_2, C_3 and C_4 are communities.

We choose the "*MAX priority*" nodes from $S\,C_i^p$ as fundamental nodes, $1 \le i \le k$. Selecting fundamental nodes in CDH-SHRINK is different from that in CDH-Kcut. To compare the *priority* of nodes, we use a function, ***compare_priority***, which works as follows, 1) If both nodes are hubs, we compare their position_score other than their degrees. We compare hubs with how many communities they belong to, since we want to cover more communities. That is, we want to choose a hub which has important positions in the network. Besides, if the node is a hub, its purity must exceed the purity threshold. a low-purity hub may cause information overlapping since it may cover too many communities. 2) When comparing non-hub nodes with hub or non-hub nodes, we compare their influence on their neighbors, i.e., degree, due to non availability of information about the importance of location of non-hub nodes.

After comparing top-p degree nodes in SC_i, we can find the fundamental nodes successfully. The fundamental node may have a very good position which connecting SC_i with many other communities and have much influence on their neighbors. The set of selected fundamental nodes are denoted as $S = \{s_1, s_2, \ldots, s_k\}$, $1 \le i \le k$. Finally, we adjust fundamental nodes to be final seed nodes.

Adjustment in CDH-SHRINK is also a heuristic, which try to choose an add-node to replace a delete-node. Then test whether the influence spread (influenced nodes) after replacing is larger than that before replacing. Two evaluation metrics, *left* and *seed load* are defined as the auxiliaries to determine add-nodes and delete-nodes.

Definition 2 (Left and Seed Load) Given a community $C_i \in$ a set of discovered communities in G, we define *left* (C_i) = |{u | $u \in C_i$ and u is a non-activated node}|. *left* (C_i) might be thought of as "the need of adding more seeds in C_i". As *left* (C_i) is increasing, the need for selecting more seeds in C_i is increasing. We define $seed\_load\ (C_i) = \frac{\text{size}(C_i)}{|\{u \mid u \in C_i, u \in S\}|}$. Implication of the *seed_load* (C_i) is whether too many seeds in C_i. When *seed_load* (C_i) is small, too many seeds exist in C_i.

Algorithm 2: *CDH-SHRINK* (G, k, p)

Input: Graph of social network G ; number of total seeds k ; parameters p

Output: k seeds

```
01:  call SHRINK (G);
02:  S ← ∅; SC ← ∅;  // seed set
03:  while |SC| < k do  // select the fundamental node
04:     SC ← SC ∪ biggest community SC_i among all remaining
        communities;
05:     add top-p degree nodes of SC_i into set S C_i^p ;
06:     for each node in S C_i^p  do
07:        max_node = compare_priority (n_i, max_node);  // n_i : the i-th
           largest degree node in S C_i^p
08:     S ← S ∪ max_node;
09:     for each community C_i which has max_node do
10:        size(C_i) = size(C_i) – degree(max_node);
11:  I_s(t) ← execute HDM on G with S;  //I_s(t): the set of influenced nodes
     of current S
12:  IM ← |I_s(t)|;  // IM : save the max number of influenced nodes
13:  for 1 to r do  //adjustment
14:     select a_node = u | max{ Σ_{i=1}^{t} left (C_i | u∈C_i), t = position_ score
        (u)}
15:     select d_comm = argmin_{Ci∈SC} seed_load (C_i);
16:     select d_node = argmin_{u∈S, u∈d_comm} |{v | v∈A(u), A(u)⊆ G(V)}|;
17:     replace the d_node with a_node in S;
18:     I_s(t) ← execute HDM on G with S;
19:     if I_s(t) < IM then
20:        restore the replacement in line 17;
21:     IM = I_s(t) ;
22:  Output individuals in S;

Procedure: Compare_priority (a, b)
23:  if (a is hub) ∧ (hub_purity (a) < purity_threshold) then
24:     return b;
25:  if (a is hub) ∧ (b is hub) then
26:     return max (position_score (a), position_score (b));
27:  if a is non-hub then
28:     return max (degree(a), degree(b));
```

Fig. 7: Pseudo codes of CDH-Kcut algorithm

In each iteration, we firstly select the add-node, a_node = u | max{ $\Sigma_{i=1}^{t}$ *left* (C_i | $u \in C_i$), t = *position_ score*(u)} and choose a delete-community, *d_comm*, which has the smallest *seed_load* among SC_i, $1 \le i \le k$. Then we select delete-node, *d_node*, which has minimum | A(*d_node*) | among all seeds in *d_comm*. Function $A(u)$ outputs a set of active nodes adjacent to node u. Finally, we test whether we should substitute add-node for delete-node. If the influence spread after substitution is more than that before, we make a substitution. To quickly find a productive *a_node*, we do

not consider very low degree nodes. We could assume selecting low degree nodes as seeds is not productive.

The pseudo code of CDH-SHRINK algorithm is given in Fig. 7. The information of community structure and hubs can be discovered by SHRINK algorithm (line 1, algorithm 2). If k seeds are desired, k iterations of selecting fundamental nodes (lines 3-9, algorithm 2) are executed. Then we adjust the fundamental nodes to select final seeds (lines 13-21, algorithm 2). The adjustment uses r iterations to test the substitution. In most cases, $r = 2k \sim 3k$ is sufficient to obtain satisfactory influence spread. Finally, we output the nodes in set S as the selected seeds (line 22, algorithm 2).

4. EXPERIMENTAL RESULTS

To evaluate the performance of CDH, three influence maximization algorithms, *CDH-Kcut*, *CDH-SHRINK* and *Enhanced Greedy Algorithm* (*EGA*) [14], are implemented for comparison. We also implement a naïve algorithm, *Degree Heuristic* (*DH*) as a baseline, which only selects the top-k largest degree nodes as the seed nodes. All algorithms are designed based on heat diffusion model and implemented in C++ language and tested on a Pentium D 3.0 GHz with 2 GB of main memory running Windows XP system. The comprehensive performance study conducts on two real world datasets, NETHep network and Facebook network. In each experiment, we vary parameters, of heat diffusion model, to compare the influence spread (number of activated nodes) and efficiency (execution time) of four algorithms.

4.1 NETHep network

In this section, we extract a large real-life academic collaboration network, NETHep, from the e-print. Each node in the network represents an author. If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by k authors, this generates a completely connected graph on k nodes. With the papers in the period from January 1993 to April 2003 (124 months), NETHep contains 12008 nodes and 237010 edges.

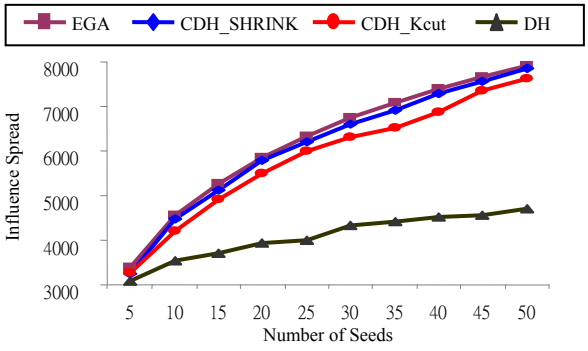


Fig. 8: Influence spread of different algorithms on NETHep with $t = 0.1$, $\theta = 0.1$, $\alpha = 0.1$.

On the large real collaboration network, NETHep, we report the efficiency and influence spread of EGA, CDH-SHRINK, CDH-Kcut and DH with different numbers of seeds and values of parameters. In CDH-SHRINK, purity threshold is set to 0.35, which can get satisfactory influence spread. Fig. 8 shows the influence spread of different algorithms with different number of seeds on NETHep. The x-axis indicates the number of seeds and y-axis indicates influence spread. In most cases, EGA's influence spread ≈ CDH-SHRINK's influence spread > CDH-Kcut's influence spread > DH's influence spread. With the increasing number of seeds, CDH-SHRINK and CDH-Kcut is getting better and is better than DH since most seeds selected by DH are only in few communities.

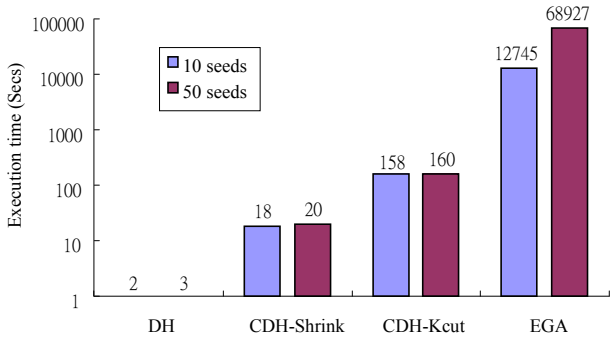
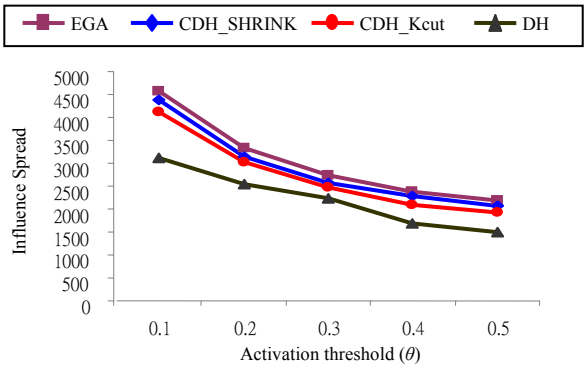
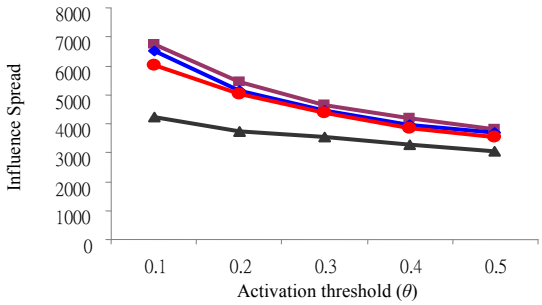


Fig. 9: Running time of different algorithms on NETHep network with selecting 10 seeds and 50 seeds.



(a) Influence spread of 10 seeds with different activation threshold.



(b) Influence spread of 30 seeds with different activation threshold.

Fig. 10: Influence spread of different algorithms on NETHep with different activation threshold.

We present the execution time of four algorithms in Fig. 9 with 10 seeds and 50 seeds. The x-axis indicates different algorithms and y-axis (logarithmic scale) indicates the execution time. Since DH only needs to select top-k degree nodes as seeds, DH is extremely efficient in execution time. CDH-SHRINK has least execution

time among other three algorithms and is about 3446 times faster than EGA (68927/20). We also can see that the running time of EGA is proportional to the number of seeds. Execution time of CDH-SHRINK and CDH-Kcut are only slightly different between 10 seeds and 50 seeds. This is because CDH-SHRINK and CDH-Kcut only have to spend a little more time on adjustment when the number of nodes increases.

Fig. 10(a) and 10(b) perform the influence spread of 10 seeds and 30 seeds with different θ from 0.1 to 0.5 with a span of 0.1, respectively. The x-axis indicates activation threshold and y-axis indicates influence spread. The results reflected in the figures show that although total influence spread of the four algorithms will decrease as θ increases, CDH-SHRINK and CDH-Kcut still maintain good influence spread. Notice that DH improves its influence spread with the increasing of θ, which results from the effect of seeds in high θ. However, it is still worse than CDH-SHRINK and CDH-Kcut when selecting more seeds.

As shown is Fig. 10, although, EGA has about 1.3% better than CDH-SHGINK in influence spread in average, the execution is tremendously slower than proposed CDH algorithms. Nowadays, the social network becomes bigger and bigger. If a time-consuming algorithm takes a long time to decide which set of individuals are seed nodes, its selection may be ineffective and lose the superiority due to the dynamic variation of social networks. The efficiency of an algorithm is also a critical issue.

4.2 Facebook Network

In this section, we discuss the influence maximization on a large real-life dataset, Facebook network. Each node in the network represents a user. If user i is a friend of user j, the graph contains an undirected edge from i to j. The network is in the period from April 2004 to 2009 January, which contains 63731 nodes and 817090 edges. We analyze the efficiency and influence spread of our algorithms with respect to different numbers of seeds and values of parameters. In CDH-SHRINK, purity threshold is set to 0.2, which can demonstrate satisfactory influence spread.

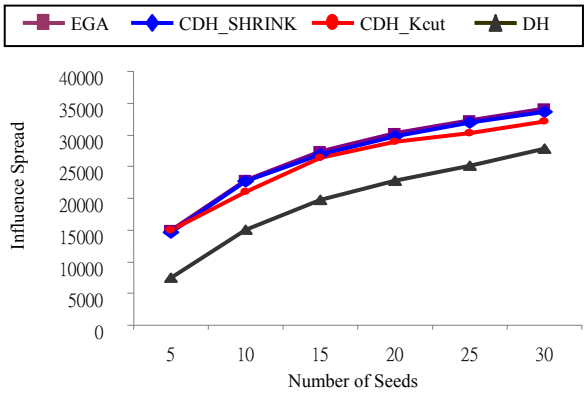


Fig. 11: Influence spread of different algorithms on FB with $t = 0.1$, $\theta = 0.1$, $\alpha = 0.1$.

Fig. 11 shows the influence spread of different algorithms with different numbers of seeds on FB. The x-axis indicates the number of seeds and y-axis indicates influence spread. As shown in Fig. 11, in most cases EGA's influence spread ≈ CDH-SHRINK's influence spread > CDH-Kcut's influence spread > DH's influence spread. Since EGA is too time-consuming, we only report the influence spread from 5 seeds to 30 seeds.

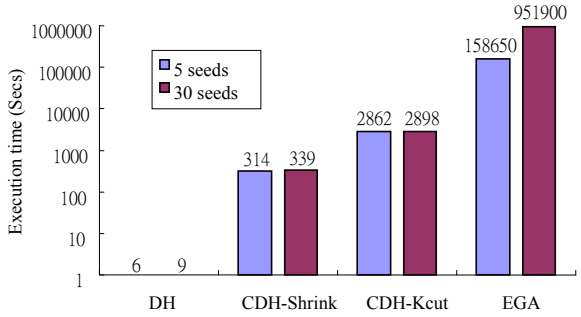
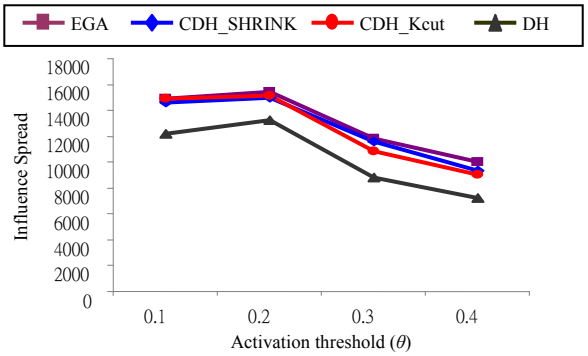
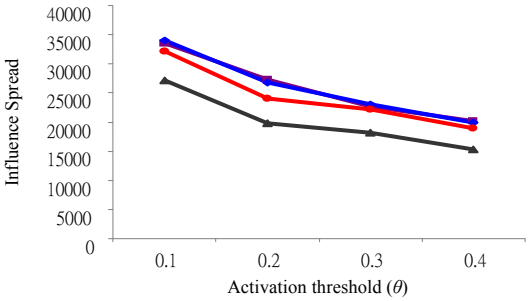


Fig. 12: Running time of different algorithms on FB network with selecting 5seeds and 30 seeds.

We present the execution time of four algorithms in Fig. 12 with 5 seeds and 30 seeds, respectively. The x-axis indicates different algorithms and y-axis (logarithmic scale) indicates execution time. CDH-SHRINK has the least running time. Just like on NETHep, we can see that the running time of EGA is proportional to the number of seeds.



(a) Influence spread of 10 seeds with different activation threshold.



(b) Influence spread of 30 seeds with different activation threshold.

Fig. 13: Influence spread of different algorithms on NETHep with different activation threshold.

Fig. 13(a) and 13(b) depict the influence spread of 10 seeds and 30 seeds with different θ from 0.1 to 0.4 with a span of 0.1, respectively. The x-axis indicates activation threshold and y-axis indicates influence spread. We can observe that, when only 10 seeds are selected, EGA's influence spread ≈ CDH-SHRINK's influence spread > CDH-Kcut's influence spread > DH's

influence spread, with θ = 0.2, 0.3 and 0.4. In Fig 13(b), we could see that the DH's influence spread is better than that in Fig 13(a). Moreover, CDH-SHRINK's influence spread is better than EGA with θ = 0.1 and 0.3.

Overall, in terms of influence spread, CDH-SHRINK ≈ EGA > CDH-Kcut > DH; in terms of efficiency, DH > CDH-SHRINK > CDH-Kcut >> EGA. One thing deserves mentioning, due to the phenomenon of seeds clustering, DH will get better performance of influence spread with high activation than that with low activation threshold.

5. CONCLUSION AND FUTURE WORK

In this paper, we present two algorithms CDH-Kcut and CDH-SHRINK under the undirected heat diffusion model by integrating the information of community structure and modified degree-centrality method. The purpose of our work is to solve the influence maximization problem efficiently and still have good influence spread based on heat diffusion model. Experimental results on the real-world datasets validate that our proposed algorithms achieve great performance in running time and influence spread. When comparing CDH-SHRINK with CDH-Kcut, CDH-SHRINK, in general, utilizes hubs and better community structure to achieve better influence spread. Although, both algorithms are efficient with time complexity O(Mlog$N$), CDH-Kcut, in practice, would cost more execution time.

In the future, to interpret the real world more realistically, influence maximization problem can be extended to weighted graphs. Furthermore, dynamic evolution is also an important property of social network. Static community detection algorithms could only detect community structure without considering the evolution of social networks. It is a worth investigation to utilize dynamic community structures to solve the influence maximization problem.

6. ACKNOWLEDGE

Suh-Yin Lee was supported by the National Science Council, Project No. NSC99-2221-E-009-128-MY2. Wen-Chih Peng was supported in part by the National Science Council, Project No. 100-2218-E-009-016-MY3 and 100-2218-E-009-013-MY3, by Taiwan MoE ATU Program, by ITRI JRC, Project No. B352BW3300, by D-Link and by Microsoft.

7. REFERENCES

[1] R. Albert, H. Jeong, and A. Barabasi, Diameter of the World Wide Web, *Nature*, vol. 401, pp. 130-131, 1999.

[2] A. Barabasi and R. Albert, Emergence of scaling in random networks, *Science*, vol. 286, pp. 509-512, 1999.

[3] D. Bortner and J. Han, Progressive Clustering of Networks Using Structure-Connected Order of Traversal, *IEEE ICDE'10*, pp. 653-656, 2010.

[4] W. Chen, Y. Wang, and S. Yang, Efficient Influence Maximization in Social Networks, *ACM SIGKDD'09*, pp. 199-208, 2009.

[5] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, Comparing Community Structure Identification, *Journal of Statistical Mechanics: Theory and Experiment*, pp. P09008, 2005.

[6] P. Domingos, Mining Social Networks for Viral Marketing, *IEEE Intelligent Systems*, vol.20, no. 1, pp. 80-93, 2005.

[7] P. Domingos and M. Richardson, Mining the Network Value of Customers, *ACM SIGKDD'01*, pp. 57-66, 2001.

[8] P. Domingos and M. Richardson, Mining Knowledge-Sharing Sites for Viral Marketing, *ACM SIGKDD'02*, pp. 61-70, 2002.

[9] M. Ester, H. Kriegel, J. Sander, and X. Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *ACM SIGKDD'96*, pp. 226-231, 1996.

[10] P. Estevez, P. Vera, and K. Saito, Selecting the Most Influential Nodes in Social Network, *IJCNN'07*, pp. 2397-2402, 2007.

[11] J. Goldenberg, B. Libai and E. Muller, Talk of Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth, Marketing Letters, vol.12, no.3, pp. 211-223, 2001.

[12] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, and Y. Liu, SHRINK: A Structural Clustering Algorithm for Detecting Hierarchical Communities in Networks, *ACM CIKM'10*, pp. 219-228, 2010.

[13] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the Spread of Influence through a Social Network, *ACM SIGKDD'03*, pp. 137-146, 2003.

[14] H. Ma, H. Yang, M. R. Lyu, and I. King, Mining Social Networks Using Heat Diffusion Processes for Marketing Candidates Selection, *ACM CIKM'08*, pp. 233-242, 2008.

[15] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society, *Nature*, vol. 435, pp. 814–818, 2005.

[16] E. Rogers, Diffusion of Innovations (5th edition), *Free Press*, New York, 2003.

[17] J. Ruan, and W. Zhang, An Efficient Spectral Algorithm for Network Community Discovery and Its Applications to Biological and Social Networks, *IEEE ICDM'07*, pp. 643-648, 2007.

[18] K. Saito, M. Kimura, K. Ohara and H. Motoda, Efficient Discovery of Influential Nodes for SIS Models in Social Networks, *Knowledge and Information Systems*, vol. 30, no. 3, pp. 613-635, 2011.

[19] T. Valente, Network Models of the Diffusion of Innovations, Hampton Press, 1995.

[20] Y. Wang, and X. Feng, A Potential-Based Node Selection Strategy for Influence Maximization in a Social Network, *ADMA'09*, pp. 350-361, 2009.

[21] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications, *Cambridge University Press*, 1994.

[22] X. Xu, N. Yuruk, Z. Feng, and T. Schweiger, SCAN: A Structural Clustering Algorithm for Networks, *ACM SIGKDD'07*, pp. 824-833, 2007.