

## Community-Based Diffuse Efficiency Algorithm for Mining Diffusion Nodes in Social Network<sup>1</sup>

Chen Yunfang, Lu Youwei, Zhang Wei and Qiu lei

Nanjing University of Posts and Telecommunications  
[chenyf@njupt.edu.cn](mailto:chenyf@njupt.edu.cn)

### Abstract

*The emerging of social networks opens opportunities for viral marketing. It is a fundamental issue to find a subset of diffusion nodes such that targeting them initially will maximize the range of the information spreading. The problem of finding the most influential spreaders is unfortunately NP-hard. The best known approximation algorithm has been proven to be with an approximation ratio of  $(1 - \frac{1}{e})$ , however, the performance and the time complexity of the approximation algorithm are not suitable for large-scale social networks. In this paper, we propose a community-based diffuse efficiency algorithm, which is differing from approximation algorithm for mining diffusion nodes through the whole networks, our algorithm identity nodes from the view of the community. The algorithm encompass two steps: Firstly, detect the number of communities in the networks by taking into account information diffusion; and then a dynamic programming algorithm for selecting communities to find diffusion nodes. The performance of the proposed algorithms is evaluated by experiments on a data set of 4000 people call logs, the results show that the community based diffuse efficiency algorithm performs better than the other two algorithms.*

**Keyword:** viral marketing, social networks, diffuse efficiency

### 1. Introduction

A social network is a social structure connecting individuals or organizations, which plays an important role for spreading information, idea and influence among its members. Nowadays, social networks have been evolving to online social networks that connect humans, computers and the Internet, and information spreading in social networks has been changed from the way of “word-of-mouth” to “word-of-text”, “word-of-voice”, “word-of-photo”, and “word-of-video” [1]. User mobility and social connectivity bring numerous communication opportunities.

The emerging of social networks also opens opportunities for viral marketing [2]. Different from traditional televised or roadside-billboard advertising campaign, viral marketing makes use of the power of “word-of-mouth” to increase brand popularity or product sale, and it has attracted considerable attentions from social computing researchers [3, 4]. However, before fully utilizing mobile social network as a platform for viral marketing, many challenges have to be addressed.

As the essence of viral marketing is information diffusion from a small number of individuals to the entire network by “word-of-mouth”, in this paper, we address the problem of identifying a small number of influential spreaders through whom the information can be diffused to the entire network as soon as possible. About this problem,

---

<sup>1</sup> The paper is supported by National Science Foundation of China (Grant 61272422)

researchers have put forward two kinds of discussions: **influence maximization problem** and **diffusion minimization problem**. Influence maximization problem focus on how to make people be influenced as much as possible in the network so as to accept their information. Different from influence maximization problem, diffusion minimization problem describes how to make the diffusion time minimum. The common of these two problems is the relationship between the number of friends participating in spreading information and the probability of adopting the information [5].

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 gives problem statement. Section 4 presents the community based diffuse efficiency algorithm. Section 5 evaluates the performance of the algorithm and Section 6 concludes the paper.

## 2. Related Work

The question of who are the key influential users, and why, has not attracted a lot of attention in the past, this is mainly because nodes sort by degree, betweenness and closeness can get a good result. Kitsak et al. [6] firstly systematically analyses this problem, they point out that these metrics are often not accurate description of the spread capacity of a node, however, using the k-shell decomposition analysis to assign an integer (coreness) to each node that describes the spread capacity is better than metrics. Soon afterward, researchers pay more attention to how individuals influence each other. Domingo's and Richardson are the first to study the influence maximization problem and give a probabilistic solution [7]. Kempe et al. designs a greedy algorithm (GA) with approximation ratio of  $(1 - \frac{1}{e})$  [8]. Due to the low efficiency of GA, Chen et al. propose two faster greedy algorithms called NewGreedy and MixedGreedy [9], respectively. Experiments show that MixedGreedy slightly outperforms NewGreedy. The main idea behind NewGreedy is to remove the edges that will not contribute to propagation from the original graph to get a sub graph and do the influence diffusion on the sub graph. Besides, Chen et al. also presents a degree discount heuristic algorithm called Degree Discount. Degree Discount assumes that the influence spread increases with the degree of nodes. Unlike Greedy algorithm, Degree Discount algorithm has no provable performance guarantee. Recently, Jiang, *et al.*, [10] propose a simulated annealing algorithm to address the influence maximization problem.

Above of these methods, researchers consider the problem from the point of either the spread ability of the node or the influence of the node, Kitsak et al. propose k-core decomposition is a good way to analysis the spread ability of a node, but when need to find a group of spread nodes, the performance of k-core decomposition is not as good as identify a node. As mentioned above, greedy algorithm has high time and compute complexity, researchers propose either heuristic algorithm based on greedy algorithm or new metrics, few of them consider the community structure. We also note that Scripps et al. present a metric to estimate the number of communities to which a node is attached and define community-based roles for a node (*e.g.*, a node that links too many nodes from different communities takes a so-called ambassador role). They also briefly discuss the application of community based roles, selecting nodes with an ambassador role to maximize the number of communities influenced by the selected nodes [11, 12]. Lu, *et al.*, [13] also design a community based algorithm; they first detect community, and then compare the number of the communities ( $m$ ) and the number of diffusion nodes ( $k$ ). If  $k \geq m$ , more than one diffusion nodes are identified in one community; if  $k < m$ , some communities need to be merged until  $k = m$ , then select one node in each community. Clearly, diffusion nodes in each community need to be computed which is a waste of computing resources. Unlike these community-based algorithms, in this paper we present a community-based diffuse efficiency algorithm to identify diffusion nodes.

### 3. Preliminaries and Problem Statement

We use  $G = (V, E, \mathcal{W})$  to present an undirected weighted mobile social network, where  $V$  and  $E$  represent the set of nodes and edges, respectively. For two neighboring nodes  $u, v \in V$ ,  $\omega_{uv}$  denotes the weight of the edge and  $\omega_{uv} = \omega_{vu}$ . The edge weight indicates the frequency of contacts between two nodes. For a node  $u \in V$ ,  $d_u$  is the degree of node  $u$  and  $N_u$  is the neighbor set of  $u$ , and we have  $d_u = \sum_{v \in N_u} \omega_{uv}$ .

In the operational model of information diffusion, each node can be either active or inactive. Active nodes are the adopters of the information and are ready to diffuse the information to their inactive neighbors. The state of a node can be switched from inactive to active, but not vice versa. In **Independent Cascade Model** [14], at time  $t$  active node  $u$  only has one chance to activate its inactive neighbor node  $v$ , and succeeds with probability  $p_{uv}$  (active probability). If multiple neighbors of  $v$  first become active at time  $t$ , then their activation attempts are sequenced in an arbitrary order, but performed at time  $t$ . Whether or not  $u$  succeeds, it cannot make any further attempts to activate  $v$  in subsequent rounds. The process terminates if no more activations are possible. In **Linear Threshold Model**, every node  $v$  is influenced by its neighbors and we use  $b_{uv}$  to represent the influence degree, for all neighbors of  $v$ , we have  $\sum b_{uv} \leq 1$ , threshold  $\theta_v$  represent the difficult of  $v$  be influenced. When  $\sum b_{uv} \geq \theta_v$ ,  $v$  becomes active. The process terminates if no more activations are possible. In **Probabilistic Diffusion Model** [15], when an active node  $u$  contacts an inactive node  $v$ ,  $v$  becomes active with some probability  $\lambda_{uv} = \frac{\omega_{uv}}{d_u}$ . This is because the probability of information spreading from node  $u$  to the neighboring node  $v$  should be proportional to the connection fraction of node  $v$  over the degree of  $u$ . From the social relation point of view, a person most likely shares the information with his best friends rather than others. The process terminates when the information spread to the whole networks.

**Definition 1(diffusion degree)** Let  $S$  be the initial set of active nodes. The diffusion degree of set  $S$  is computed as:

$$R(S) = V_S / N \quad (1)$$

Where  $V_S$  is the number of nodes activated by  $S$  during information diffusion process,  $N$  is number of nodes in the network.

The diffusion time of initially selected node set is defined as the time interval between the start and the end of the information diffusion process denoted by  $\tau(S, V)$ . As proved in [16], probabilistic model and Independent Cascade model can be unified with proper parameter initialization, thus we compute diffusion degree in Independent Cascade Model and compute diffusion time in Probabilistic Diffusion Model. The weight in Independent Cascade Model means the active probability between two nodes and in Probabilistic Diffusion Model represents the contact frequency of two nodes, respectively.

**Definition 2(the expected diffusion time)** under the probabilistic diffusion model, the expected information diffusion time from node  $u$  (active) to neighboring node  $v$  (inactive) can be formulated as

$$t_{uv} = \frac{1}{\lambda_{uv}} \cdot \frac{1}{\omega_{uv}} = \frac{d_u}{\omega_{uv}^2} \quad (2)$$

Where  $\frac{1}{\omega_{uv}}$  denotes the average time interval between contacts. Similarly, we have  $t_{vu} = \frac{d_v}{\omega_{uv}^2}$  from node  $v$  to node  $u$  (the expected diffusion time from  $u$  to  $v$  and that from  $v$  to  $u$  are different, except  $d_u = d_v$ ). For any pair of nodes, for example node  $u$  and  $v$ , the shortest expected diffusion time from  $u$  to  $v$  is denoted as  $|u, v|$ .

Since the diffusion time between any pair of nodes can be estimated by the expected diffusion time, the diffusion minimization problem under the probabilistic diffusion

model can be mathematically formulated as finding a subset  $S \subseteq V (|S| \leq K)$  to minimize the expected diffusion time:  $\tau(S, V) = \min \max_{v \in V} |S, v|$ , where  $|S, v|$  the expected diffusion time from set  $S$  to node  $v$ .

We want to spread information as short time as possible to as many people as possible, which can be mathematically formulated as:

$$E(S) = R(S)/\tau(S, V) \quad (3)$$

The equation can be understood as compute the number of nodes in network per unit time that  $S$  spread information.

For a given network  $G = (V, E, \mathcal{W})$ , we aim to find a subset  $S$  consist of top-k diffusion nodes to make  $E(S)$  maximum. We propose a community-based diffuse efficiency algorithm to solve this problem.

## 4. Algorithm

For information diffusion in social network, community has the prominent property. Within a community, nodes frequently contact each other and hence information can be rapidly spread. Information diffusion from one community to another community is much slower compared to that within the community. This property indicates that it is fast and efficient to find diffusion nodes in the community rather than the entire network. In this section, firstly, we introduce the traditional metrics and GA, and then present our community-based diffuse efficiency algorithm.

### 4.1. Traditional Metrics and Algorithm

In traditional social network analysis, the influential spreaders are these have high degree, betweenness and closeness. Node degree is defined as the number of its neighbors which is a simple and effective method to measure the influence of the node. The higher degree of a node, the more neighboring nodes it connects and the more influence it has. The betweenness of a node  $i$  is defined as all possible shortest paths between two nodes in proportion to the path pass through the node  $i$  in the network. The closeness of node  $u$  is defined as the reciprocal of the sum of the shortest distances to all other nodes in the network. Under the Probabilistic Diffusion Model, it can be mathematically formulated as:

$$C(u) = \frac{1}{\sum_{v \in V} |u, v|} \quad (4)$$

Closeness is a measure of how fast it will take to spread information from a node to all other nodes [17]. With regard to identifying  $S$  from  $V$ , an easy solution is compute the degree, betweenness and closeness of all the nodes in the network, then rank them and select top-k nodes as the initial diffusion set  $S$ .

When consider the influence maximization problem, GA is proved to guarantee the approximation ratio of  $(1 - \frac{1}{e})$ , the algorithm needs to define a gain function, then iteratively select the node with the highest marginal gain from the set of unselected nodes until  $|S| = k$ , however, GA based on the marginal gain needs a larger amount of computing on large-scale network, next, we present our community-based diffuse efficiency algorithm.

### 4.2. Community-Based Diffuse Efficiency Algorithm

The basic idea of our algorithm is to detect communities and identify top-k nodes from the community. Suppose that we already divide a network into  $m$  communities and let  $C = \{C_1, C_2, \dots, C_m\}$  denote the  $m$  communities, for simplicity, there is no nodes belong to multiple communities. The remaining challenge is to choose which communities to

find the nodes. We propose a dynamic programming algorithm to choose which community to find top-k diffusion nodes.

Let  $S_{k-1}$  be the set of diffusion nodes obtained in the previous k-1 steps. We are then ready to compute the maximal increase of the diffusion degree in the shortest time with regard to community  $C_m$ . The maximal increase is denoted as  $\Delta E_m$  and we have:

$$\Delta E_m = \max\{E_m(S_{k-1} \cup v_j) - E_m(S_{k-1}) | v_j \in C_m\} \quad (5)$$

Note that in equation (5) the diffusion efficiency  $\Delta E_m(\cdot)$  is computed with regard to the community  $C_m$  rather than the whole network.

To find  $k^{\text{th}}$  diffusion node, we need to choose the community that yield the largest increase of diffusion efficiency. Let  $E[m, k]$  ( $m \in [1, M], k \in [1, K]$ ) be the diffusion degree of mining the  $k^{\text{th}}$  diffusion node in the first m communities. We have:

$$E[m, k] = \max\{E[m-1, k], E[m, k-1] + \Delta E\} \quad (6)$$

$$E[m, 0] = E[0, k] = 0$$

The equation (6) can be explained as follows: if the diffusion degree of mining the  $k^{\text{th}}$  diffusion node in the first m-1 communities is smaller than that of mining the  $k^{\text{th}}$  diffusion node in  $C_m$ ; otherwise, we mine it in the former m-1 communities.

We select a community from the first m communities to mine the  $k^{\text{th}}$  diffusion node, and the selected community is represented by a sign functions  $s[m, k]$ . It is defined as follows:

$$s[m, k] = \begin{cases} s[m-1, k], & E[m-1, k] \geq E[M, k-1] + \Delta E_m \\ m, & E[m-1, k] < E[M, k-1] + \Delta E_m \end{cases} \quad (7)$$

$$s[0, k] = 0$$

To identify the  $k^{\text{th}}$  diffusion node, we choose the community  $s[M, k]$ . In principle, we can use any existing algorithm to identify the  $k^{\text{th}}$  diffusion node in community  $s[M, k]$ . We adopt the MixedGreedy algorithm since it has the same approximation precision as previously proposed greedy algorithms and is shown to be more efficient.

---

**Algorithm 1:** Community based Algorithm

---

**Input:** G, K

**Output:** S

1:  $C \leftarrow$  detect communities in G

2:  $M = |C|$

3:  $S = S_1 = S_2 = \dots = S_M = \emptyset$

4: **for** k=1 **to** K **do**

5:      $E[0, k] = 0; s[0, k] = 0;$

6: **end for**

7: **for** m=1 **to** M **do**

8:      $E[m, 0] = 0;$

9: **end for**

10:   **for** k=1 **to** K **do**

11:       **for** m=1 **to** M **do**

12:              $\Delta E_m = \max\{E_m(S_{k-1} \cup v_j) - E_m(S_{k-1}) | v_j \in C_m\}$

13:              $E[m, k] = \max\{E[m-1, k], E[M, k-1] + \Delta E\}$

14:             **if**  $E[m-1, k] \geq E[M, k-1] + \Delta E_m$  **then**

15:                  $s[m, k] = s[m-1, k];$

16:             **else**

17:                  $s[m, k] = m$

```

18:         end if
19:     end for
20:     j= s[M, k]
21:     vmax = arg maxvi ∈ Cj (Ej(Sj ∪ {vi}) - E(Sj))
22:     Sj = Sj ∪ {vmax}, S = S ∪ {vmax};
23: end for
    
```

The algorithm is displayed in Algorithm 2. It first detects communities (line 1), and initializes the set of diffusion nodes detected in the whole network (S) and in each community (S<sub>1</sub>, ..., S<sub>M</sub>) as null (line 3). The algorithm initializes E[m, k] and s[m, k] (lines 4–9). In lines 10-19, the algorithm chooses which community to mine k<sup>th</sup> diffusion node using dynamic programming algorithm. If the sum of the diffusion efficiency ΔE<sub>m</sub> and E[M, k - 1] is smaller than E[m - 1, k], the diffusion efficiency of mining k<sup>th</sup> node in the first m-1 communities (line 14), and thus label s[m, k] as s[m - 1, k] (line 15); otherwise we mine the k<sup>th</sup> node in community C<sub>m</sub> and set s[m, k] as m (line 17). The algorithm will find the k<sup>th</sup> diffusion node in the community C<sub>j</sub>, j = s[M, k] (line 20). We employ MixedGreedy algorithm [10] to find the node v<sub>max</sub> that maximizes E<sub>j</sub>(S<sub>j</sub> ∪ {v<sub>i</sub>}) - E(S<sub>j</sub>) in community C<sub>j</sub>(line 21).

**Illustrate the algorithm:**

We want to find top-2 nodes in the network. Suppose that the network partitioned into three communities C<sub>1</sub>, C<sub>2</sub> and C<sub>3</sub>, and based on Independent Cascade model and Probabilistic Diffusion model we have: the maximal diffusion efficiency increment, denoted by ΔE<sub>1</sub>, ΔE<sub>2</sub> and ΔE<sub>3</sub>, is 0.2, 0.3, 0.1, respectively, for each community; and the second diffusion efficiency increment is 0.08, 0.06, 0.04, respectively, for each community. The community based diffuse efficiency algorithm works as follows:

1. We proceed to select which community to identify top-1 node.

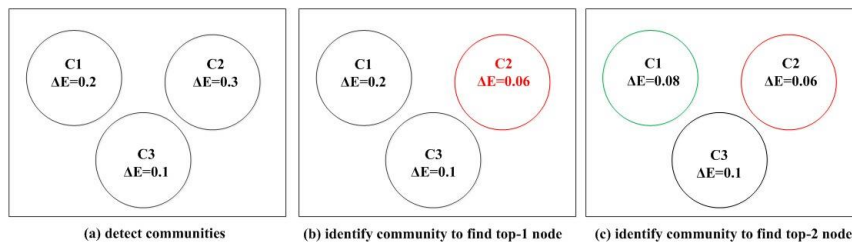
$$\begin{aligned}
 E[1,1] &= \max\{E[0,1], E[3,0] + \Delta E_1\} = \max\{0, 0 + 0.2\} = 0.2, s[1,1] = C_1; \\
 E[2,1] &= \max\{E[1,1], E[3,0] + \Delta E_2\} = \max\{0.2, 0 + 0.3\} = 0.3, s[2,1] = C_2; \\
 E[3,1] &= \max\{E[2,1], E[3,0] + \Delta E_3\} = \max\{0.3, 0 + 0.1\} = 0.3, s[3,1] = C_2;
 \end{aligned}$$

Hence, we identify the top-1 node in community C<sub>2</sub> because s[3,1] = C<sub>2</sub>.

2. Next, find top-2 node.

$$\begin{aligned}
 E[1,2] &= \max\{E[0,2], E[3,1] + \Delta E_1\} = \max\{0, 0.3 + 0.2\} = 0.5, s[1,2] = C_1; \\
 E[2,2] &= \max\{E[1,2], E[3,1] + \Delta E_2\} = \max\{0.5, 0.3 + 0.06\} = 0.5, s[2,2] = C_1; \\
 E[3,2] &= \max\{E[2,2], E[3,1] + \Delta E_3\} = \max\{0.5, 0.3 + 0.1\} = 0.5, s[3,2] = C_1;
 \end{aligned}$$

Note that ΔE<sub>2</sub> in this step is 0.06, but not 0.3. We find the second node in community C<sub>1</sub> because s[3,2] = C<sub>1</sub>.



**Figure 1. Illustration of the Community Based Diffuse Efficiency Algorithm**

Figure 1 show the above steps, where (a) displays the network partitioned into 3 communities  $C_1$ ,  $C_2$  and  $C_3$ , the label in the circle indicates the diffusion efficiency increment of each community. (b) And (c) displays the result of select which community to identify top-1 node, top-2 node, respectively.

### Complexity Analysis:

We consider the complexity of lines 2–23 of Algorithm 2. The algorithm needs  $O(K)$  time in lines 4–6, and  $O(M)$  time in lines 7–9. Suppose the largest community after community detection is  $C_p$ , thus lines 10–19 take  $O(MKT_p)$  time, and lines 20–23 of mining nodes using MixedGreedy algorithm [4, 14] take  $O(K|C_p|T_p)$  time, where  $T_p$  is the time to compute the influence degree of a node in community  $C_p$ . Hence the worst-case complexity of lines 2–23 of Algorithm 1 is  $O(MKT_p + K|C_p|T_p)$ , which is smaller than the complexity of GA with  $O(n^2)$ , where  $n$  is the number of nodes in the network.

## 5. Experiments and Discussions

### 5.1. Experiment Data Sets

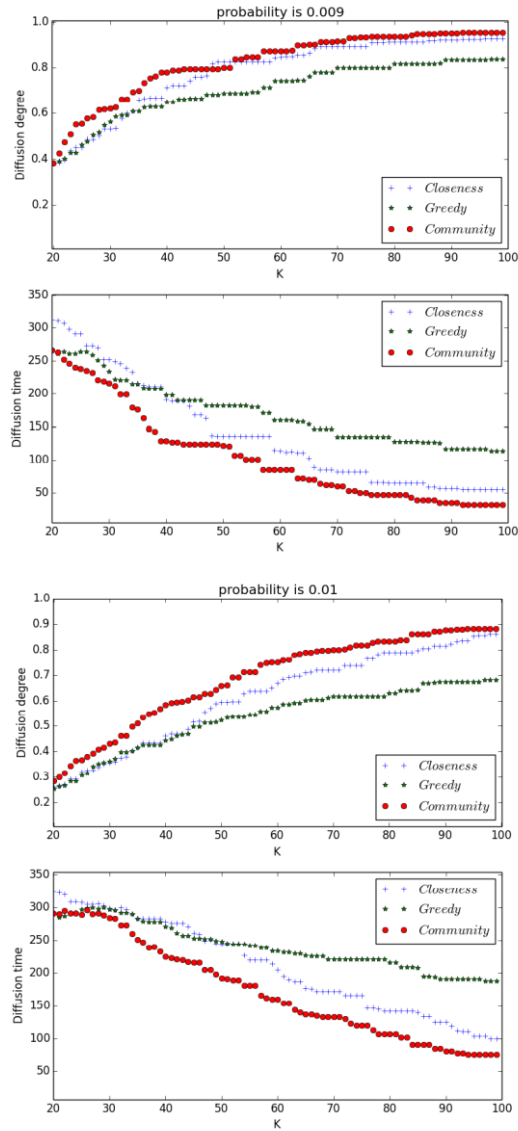
We extract a mobile social network from the call log of 4000 individuals and obtain an undirected weighted graph. In this paper, we compute the average communication time between two nodes as the edge weight, which can be formulated as  $\omega_{uv} = \frac{T_{uv}}{N_{uv}}$  or  $\omega_{vu} = \frac{T_{vu}}{N_{vu}}$ , where  $T$  denotes the sum of communication time between two nodes and  $N$  denotes the number of calls between two nodes.

We compare the community based diffuse efficiency algorithm, greedy algorithm (GA) and closeness algorithm by setting different active probability, in terms of diffusion degree and expected diffusion time. We use k-clique algorithm to detect communities and simulation 100 times.

### 5.2. Performance Evaluation

#### (1). Varying K

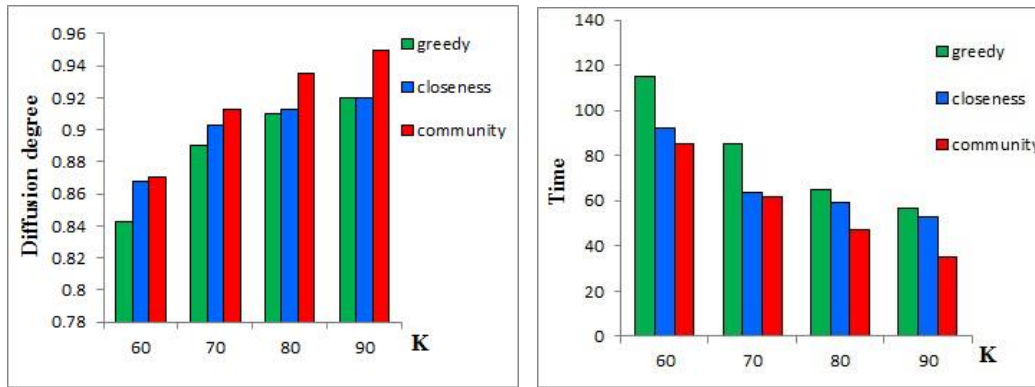
This experiment is to evaluate the effect of the parameter  $K$  on the diffusion degree and time of different algorithms. We fix the active probability at 0.009 and 0.01, respectively. We vary  $k$  from 20 to 100. The results are shown in Figure 2. It can be seen in Figure 2 that the smaller active probability is, the more easy information can be diffused into the network. The top of each picture is the relationship between  $k$  and diffusion degree, and the bottom of each picture is the relationship between  $k$  and diffusion time. With  $k$  growing, diffusion degree is increasing and time is decreasing, respectively. Specially, the community-based diffuse efficiency algorithm performs better than other two algorithms in both two different settings.



**Figure 2. Diffusion Degree and Time of 3 Algorithms with Different Active Probability**

Figure 3 shows more details about the condition of the active probability is fixed at 0.01. We compare the 3 different algorithms in terms of k is 60, 70, 80 and 90. We can easily observe the gap of diffusion degree and time between community-based diffuse efficiency algorithm and two other algorithms is more and more big with k increasing.

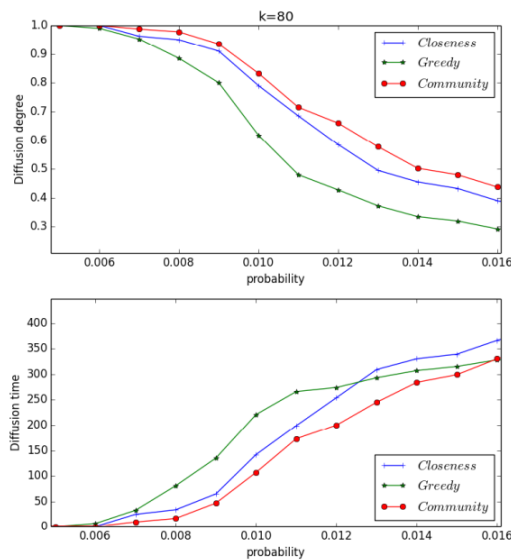


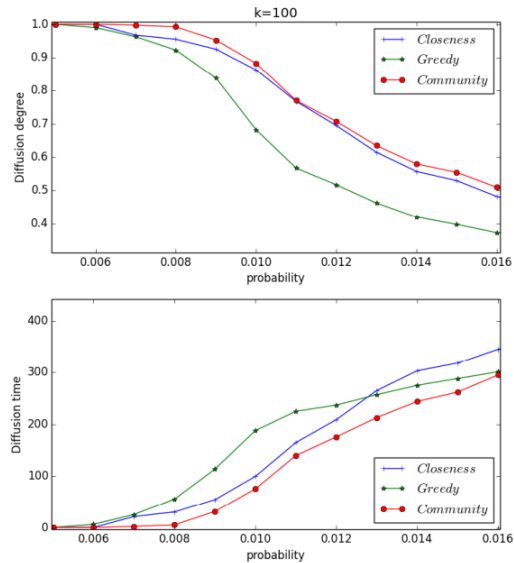


**Figure 3. Diffusion Degree and Time of 3 Different Algorithms with Active Probability 0.01**

**(2). Varying Active Probability**

Next, we fix k at 80 and 100, respectively, and vary active probability from 0.006 to 0.016. We evaluate the effect of the parameter active probability on the diffusion degree and time of different algorithms. As is shown in Figure 4, with the probability increasing, the diffusion degree is decreasing and the time is growing, it can be easily understood as: the more difficult people accept the information, it spends more time diffusing into the network, but the community based algorithm is always performs the best. When k=80, probability=0.015, the diffusion degree of community algorithm is above 50%, closeness is about 50% and GA is less than 50%, respectively, in additions, the community based algorithm spends the least time. In other words, on the question of who can use as less time as possible to spread as more information as possible to the network, the community based diffuse efficiency algorithm performs better than others.





**Figure 4. Diffusion Degree and Time of Top-80 VS Top-100 with Different Algorithm**

In summary, Community performs better than GA and closeness in terms of diffusion degree and time. Because Community relies on the community structure and identifies diffusion nodes from individual communities rather than the entire network as in GA and closeness.

## 6. Conclusions

In this paper, we address the fundamental problem for the viral marketing: identify a small number of influential spreaders through whom the information can be diffused to the entire network as soon as possible. We propose a community based diffuse efficiency algorithm and evaluate its performance in a social network with 4000 nodes, compared with the greedy algorithm and closeness, the results shows the performance of community based algorithm is better than the other two algorithms. This is because community based algorithm leverages the community structure and identify diffusion nodes from the community rather than the whole network.

At the same time, we note that it is hard to get the complete network data in reality, but we can easy get the geographic information of users, so the future work is research the information diffusion from the view of space and time.

## References

- [1] H. Ma, H. Yang, M. R. Lyu and I. King, "Mining social networks using heat diffusion processes for marketing candidates selection," in Proc. of ACM CIKM, (2008).
- [2] W. Chen, C. Wang and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks", in Proc. of ACM SIGKDD, (2010).
- [3] T. Ning, Z. Yang, H. Wu and Z. Han, "Self-interest-drive incentives for ad dissemination in autonomous mobile social networks", in Proc. of IEEE INFOCOM, (2013).
- [4] W. Peng, F. Li, X. Zou and J. Wu, "A privacy-preserving social aware incentive system for word-of-mouth advertisement dissemination on smart mobile devices", in Proc. of IEEE SECON, (2012).
- [5] E. Bakshy, I. Rosenn, C. Marlow and L. Adamic, "The role of social networks in information diffusion", in Proc. of ACM WWW, (2012).
- [6] M. Kitsak, L. K. Gallos, S. Havlin, F.Liljeros, L. Muchnik, H. E. Stanley and H. A. Makes,

- “Identification of influential spreaders in complex networks,” *Nature Physics*, vol. 6, (2010), pp. 888-893.
- [7] P. Domingos and M. Richardson, “Mining the network value of customers,” in *Proc. of ACM SIGKDD*, (2001).
- [8] D. Kempe, J. Kleinberg and E. Tardos, “Maximizing the spread of influence through a social network”, in *Proc. of ACM SIGKDD*, (2003).
- [9] W. Chen, Y. Wang and S. Yang. “Efficient influence maximization in social networks,” In *KDD*, (2009), pp. 199-208.
- [10] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si and K. Xie, “Simulated annealing based influence maximization in social networks”, in *Proc. of AAAI*, (2011).
- [11] J. Scripps, P. N. Tan and A. H. Esfahanian, “Exploration of link structure and community-based node roles in network analysis”, In *Data Mining, IEEE International Conference*, (2007), pp. 649-654.
- [12] J. Scripps, P. N. Tan and A. H. Esfahanian, “Node roles and community structure in networks”, In *Joint 9th WEBKDD*, (2007), pp. 26-35.
- [13] Z. Lu, Y. Wen and G. Cao, “Information Diffusion in Mobile Social Networks: The Speed Perspective”, in *Proc. of IEEE INFOCOM*, (2014).
- [14] D. Kempe, J. Kleinberg and E. Tardos, “Maximizing the spread of influence through a social network”, in *Proc. of ACM SIGKDD*, (2003).
- [15] D. Lopez-Pintado, “Diffusion in complex social networks,” *Games and Economic Behavior*, vol. 62, no. 2, (2008), pp. 573–590.
- [16] D. Kempe, J. Kleinberg and E. Tardos. “Influential nodes in a diffusion model for social networks”, In *International colloquium on automata, languages and programming*, no. 32, (2005), pp. 1127-1138.
- [17] M. E. Newman, “A measure of betweenness centrality based on random walks,” *Social networks*, vol. 27, no. 1, (2005), pp. 39–54.

## Authors



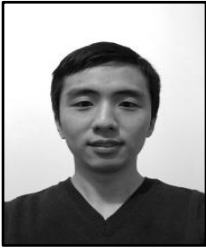
**Chen Yunfang**, Male, he was born in Zhenjiang, China in 1976. Now, he is an associate professor in Nanjing University of Posts and Telecommunications, and his research directions are social computing, artificial immune and so on.



**Lu Youwei**, Male, he was born in Zhenjiang, China in 1991. Now, he is a graduate student in Nanjing University of Posts and Telecommunications and his area of interest is social computing.



**Zhang Wei**, Male, he was born in Taizhou, China in 1973. Now, he is a professor in Nanjing University of Posts and Telecommunications and his research directions are social computing, information security and so on.



**Qiu lei**, Male, he was born in Yangzhou, China in 1990. Now, he is an undergraduate's student in Nanjing University of Posts and Telecommunications, and his area of interest is social computing.