Community Centrality-Based Greedy Approach for Identifying Top-K Influencers in Social Networks

Bundit Manaskasemsak(✉), Nattawut Dejkajonwuth, and Arnon Rungsawang

Massive Information and Knowledge Engineering Laboratory,
Department of Computer Engineering, Faculty of Engineering,
Kasetsart University, Bangkok 10900, Thailand
{un,arnon}@mikelab.net, nattawut.d@ku.th

Abstract. Online social network today is an effective media to share and disperse tons of information, especially for advertizing and marketing. However, with limited budgets, commercial companies make hard efforts to determine a set of source persons who can highly diffuse information of their products, implying that more benefits will be received. In this paper, we propose an algorithm, called community centrality-based greedy algorithm, for the problem of finding top-k influencers in social networks. The algorithm is composed of four main processes. First, a social network is partitioned into communities using the Markov clustering algorithm. Second, nodes with highest centrality values are extracted from each community. Third, some communities are combined; and last, top-k influencers are determined from a set of highest centrality nodes based on the independent cascade model. We conduct experiments on a publicly available Higgs Twitter dataset. Experimental results show that the proposed algorithm executes much faster than the state-of-the-art greedy one, while still maximized nearly the same influence spread.

Keywords: Social network · Community detection · Node centrality · Influence maximization · Influencer

1 Introduction

During the past decade, social network has played an important role as a virtual community where people with common interests can connect to share information, ideas, or even their thoughts. Though this network, commercial companies can gain a lot of benefit by a mechanism of spreading information about their products (or services) from one person to others, called *word-of-mouth* marketing. However, with the limited budget (say, k pieces of the sample product), the companies need to make an attempt to determine a set of source persons who can diffuse information to their friends in a social network as many as possible, so that the number of persons adopting the product is maximized. This effort has been introduced as the *influence maximization* problem [5], and those source persons are called the *influencers*.

P.C. Vinh and V. Alagar (Eds.): ICCASA 2015, LNICST 165, pp. 141–150, 2016.
DOI: 10.1007/978-3-319-29236-6_15

To simulate the mechanism of influence propagation, two models are formally introduced according to a stochastic cascade model [8], named the *independent cascade model* (ICM) and the *linear threshold model* (LTM). Since a social network, by nature, is very large, developing an efficient algorithm to find top-k influencers is not trivial. Kempe *et al.* [8] have proven that the optimization of influence maximization is NP-Hard. They then suggest applying the greedy approach and have shown that the optimal solution can be approximated. However, their algorithm still takes much time. Recent studies have been proposed algorithms for efficiently maximizing influence in several ways, for instance, by enhancing the naive greedy version [1,11], by employing the centrality heuristics [1,2,9], and by applying graph clustering or community-based detection [3,6,10,12].

In this paper, we propose an algorithm applying community detection technique (i.e., the Markov clustering [14]) before determining top-k influential nodes in social networks. The key contributions of our approach are as follows.

- We define a social network as a *weighted* directed graph constructed from a combination of topological graph (i.e., relationship such as friendship in case of Facebook or following in case of Twitter) and interaction one (e.g., wall posting, user tagging, commenting, liking, and sharing in case of Facebook; or tweeting, mentioning, replying, favoriting, and retweeting in case of Twitter). The difference from other community-based approaches is that those existing ones concentrate on the former type of graph only.
- We employ various *node centrality* heuristics: in-degree, out-degree, betweenness, and closeness, in the analysis.

The remainder of this paper is organized as follows. Section 2 briefly mentions to some studies related to ours. Section 3 details the proposed community centrality-based greedy algorithm. Section 4 reports performance evaluation. Finally, Sect. 5 concludes the paper.

2 Related Work

Motivated by marketing applications, the influence maximization problem in social networks is first investigated by Domingos and Richardson [5]. Later, Kempe *et al.* [8] formulate it as a discrete optimization problem. They show that the optimal solution is NP-hard, and present a greedy algorithm (GA) that guarantees the influence spread within $(1 - \frac{1}{e})$ of the optimal solution. However, their algorithm is very slow in practice and not scalable with the network size. Leskovec *et al.* [11] and Goyal *et al.* [7] propose *CELF* and *CELF++* algorithms, respectively. Both are relied on the lazy-forward optimization that uses the submodularity property to reduce the number of evaluations on the influence spread of nodes. Although the algorithms significantly speed up the greedy, they still cannot scale to very large networks. Chen *et al.* [1] propose two faster greedy-based algorithms: *NewGreedy* and *MixedGreedy*. The main idea behind the former is to reduce the original social graph into a smaller one by removing

edges that tend to have no contribution on information propagation, while the latter is a combination of *NewGreedy* and *CELF*. That is, its iterative computation employs *NewGreedy* at the first round and *CELF* for the rest rounds.

Centrality heuristics also have been proven to be an efficient alternative for maximizing influence spread in social networks. The most classic approach is the degree centrality heuristic [16]. The key concept is that a user having a lot of connections (i.e., friends) tends to highly influence others and thus should be selected as an influential candidate seed. Based on such intuition, the degree centrality heuristic selects k nodes that have the highest degree. Chen *et al.* [1] propose the degree discount heuristic based on general idea that if one node is considered as seed, then the links connecting with the node will not be counted as a degree of the other nodes. Thus, when considering the next influential node, a node with the highest degree after the discount is selected as a member of the seed set. This procedure will be repeated until the first k highest degree seeds are selected. Lastly, Chen *et al.* [2] use the eigenvector centrality heuristic to select influential nodes based on their PageRank value [13]. That is, when a social network is represented as a transitional matrix, a PageRank value for each node is first calculated. Then, the k nodes with the highest PageRank values are selected as seeds.

Recently, community-based greedy approaches are introduced in several studies; but, we will mention to some of them here. Most algorithms formally consist of two phases: a graph partitioning and an influence examining on each partition. Wang *et al.* [15] propose the community-based greedy algorithm (*CGA*) which first detects communities in a social network by taking into account information diffusion. Then, top-k influential nodes are selected and examined from those communities using a dynamic programming to speed up the computation. Kim *et al.* [10] propose the variations of a Markov clustering-based algorithm that first partition a network and consider most k influential candidates in those communities. Afterwards, an attractor identification procedure is performed again to find the influencers. Similar to their work, based on the community structure of the network, our community centrality-based greedy algorithm proposed in this paper also employs the Markov clustering. However, the main differences are that (1) top-k candidates are selected from each community using several node centrality heuristics, and (2) a community combination is performed by grouping some very small and dispersed communities to produce more proper ones.

3 Community Centrality-Based Greedy Algorithm

Given a social data—Twitter in our case study, the network is represented by a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V}$ is a set of nodes, denoted individuals. $\mathcal{E}$ is a set of edges, referred to reverse direction of followings, identical direction of interactions, or both; for example, if a person v has followed a person u, then an edge $e(u, v)$ is defined. $\mathcal{W}$ is a set of normalized weights assigned on each edge, determined by both topological structure and interactions. Let r_{uv} be a topological indicator, assigned to either 1 if v has followed u, or 0 otherwise;

and let i_{uv} be the number of interactions that u has acted to v. Then, a weight w_{uv} assigned on $e(u, v)$ is defined as:

$$w_{uv} = \omega \frac{r_{uv}}{\sum_{\forall x \in \mathcal{V}} r_{xv}} + (1 - \omega) \frac{i_{uv}}{\sum_{\forall x \in \mathcal{V}} i_{ux}},$$

where ω is a pre-defined coefficient determining the effect of topological graph and interaction one. For example, suppose that a person v has followed three persons, including u; whereas the person u has two followers in total, i.e., v and x. If u has publicly tweeted thrice and also directly mentioned to v twice, implying that the actions from u may influence v and x with five and three attempts, respectively. Then, a weight value of $\omega(\frac{1}{3}) + (1 - \omega)(\frac{5}{5+3})$ is assigned on the edge $e(u, v)$. Note that, in our experiments, we simply set ω by a uniform value (i.e., 0.5).

First of all, an important concept employed in most greedy-based approaches for the independent cascade model is that a node u is said to influence a node v if the node u attempts to activate the node v so that v becomes active from inactive. This activation must be success at least $\mathcal{R}/2$ times out of $\mathcal{R}$ simulations of the diffusion process. In addition, the ability that the node u can influence the node v depends on the weight from u to v.

We now propose the community centrality-based greedy algorithm (*CCGA*). The algorithm workflow, depicted in Fig. 1, is composed of 4 main modules: community detection, centrality analysis, community combination, and influencer identification, respectively.

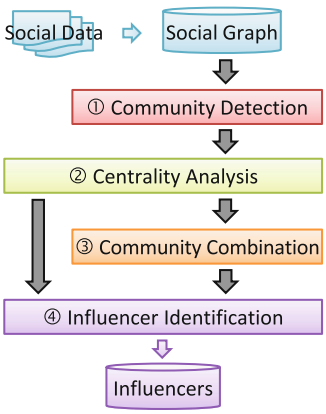

Fig. 1. The community centrality-based greedy workflow.

(1) *Community Detection:* The main idea of our method is to first partition a large network into communities, and then a number of influential candidates are examined from each community. Fortunately, most community detection algorithms rely on the intrinsic property of social networks, i.e., individuals grouped

together into a community will interact with each other more frequently than with those outside the community. So that, within a community, they are more likely to influence each other, in contrast to individuals across communities. This property suggests a good approximation task for choosing and examining influencers only within communities instead of the entire network, in order to reduce the computational time.

In *CCGA*, the original social network is partitioned based on the topological structure into several communities, referred to $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$, using the Markov clustering [14]. Markov clustering (MCL) is an attractive algorithm adopted in many domains since it divides the network without requiring the number of communities as an input parameter. The algorithm assumes that there exist communities in a network, and takes a random walk approach to clustering. That is, a random walk through the network will result in longer time spent walking within a community, and less time spent traveling along edges joining two different communities. Thus, MCL uses such intuition and groups nodes whose random walker stops at the same node.

(2) *Centrality Analysis:* In the graph theory, node centrality can heuristically be identified as the most important vertices in a graph. We then apply this centrality concept in a social network to determine a number of individuals with the highest centrality values (i.e., top-k) from each community, and mark them as the influential candidates. Furthermore, four criterions of centrality analysis are employed in this paper, including:

- *In-degree centrality* – A simplest analysis measures a node importance by counting the number of ties directed to that node. In other words, the in-degree centrality can be interpreted as a form of node popularity. Suppose a node u belongs to a community $\mathcal{C}_i$. Then, the in-degree centrality of u is defined as:
$$\varphi_I(u) = |\{(v, u) : \forall v \in \mathcal{C}_i\}|.$$
- *Out-degree Centrality* – In contrast to in-degree, the out-degree analysis measure a node importance by counting the number of ties that the node directs to others. Thus, the out-degree centrality can be interpreted as a form of node socialness. Similarly, if a node u belongs to a community $\mathcal{C}_i$ then the out-degree centrality of u is defined as:
$$\varphi_O(u) = |\{(u, v) : \forall v \in \mathcal{C}_i\}|.$$
- *Betweenness Centrality* – A betweenness of a node is defined as the number of pairs of individuals would have to go through that node in order to reach one another with the minimum number of hops. Consider a community $\mathcal{C}_i$, if we let σ_{st} be total number of shortest paths from a node s to a node t within $\mathcal{C}_i$, and $\sigma_{st}(u)$ be the number of those paths that pass through the node u. Then, the betweenness of u is defined as:
$$\varphi_B(u) = \sum_{\forall s,t \in \mathcal{C}_i : s \neq t \neq u} \frac{\sigma_{st}(u)}{\sigma_{st}}.$$

– *Closeness Centrality* – A closeness of a node is defined as the length of the average shortest path between that node and all others in a connected network. Thus, the more central a node is, the lower its total distance from all other nodes. Let $d(u, v)$ be the distance from a node u to a node v within a community $\mathcal{C}_i$. Then, the closeness of u is defined as:

$$\varphi_C(u) = \left(\frac{\sum_{\forall v \in \mathcal{C}_i : v \neq u} d(u, v)}{|\mathcal{C}_i| - 1} \right)^{-1},$$

where $|\mathcal{C}_i|$ denotes the number of nodes existing in $\mathcal{C}_i$.

(3) *Community Combination:* Since the MCL algorithm sometimes generates too many small and dispersed communities, finding influential nodes within those small communities may lead to get useless results. To avoid this problem, the community combination module is introduced to merge some communities in order to produce more proper ones.

Suppose that we have already partitioned a network into n communities, and top-k influential candidates are chosen from each community. Here, we hypothesize that if any two candidates belonging to two different communities are connected (i.e., via either topological structure, interaction, or both), then those communities should be merged together. Mathematically, we let $\mathcal{I}_i$ and $\mathcal{I}_j$ be a set of k candidates with the highest centrality values extracted from individual communities $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively. Both $\mathcal{C}_i$ and $\mathcal{C}_j$ will be further combined if an edge $e(u, v)$ or $e(v, u)$ exists in the social network $\mathcal{G}$, where $u \in \mathcal{I}_i$ and $v \in \mathcal{I}_j$. Recall that the node centrality value can be defined as one of the above four criterions.

(4) *Influencer Identification:* After we obtain the final communities, the influencer identification module aims to find top-k influential nodes over the entire network. More precisely, the top-k candidates are chosen again from each community using the same centrality criterion, and later collected them together as a candidate set $\mathcal{D}$. Then, we employ the independent cascade model (ICM) [8] to simulate the influence propagation. Based on the iterative greedy-based computation, the number of activated nodes (i.e., influence spread) is obtained by examining each combination of candidates in $\mathcal{D}$. Finally, a combination of k candidates with most corresponding influence spread is returned from the algorithm as the first k influencers.

The proposed *CCGA* is outlined in Fig. 2. The algorithm first detects communities using MCL (line 1). Then, some communities are combined with respect to connections between centrality nodes (line 2). At lines 3–7, all top-k influential candidates are collected from each community obtained after the combination process. Statements at lines 8–18 perform the greedy-based ICM for finding the most k influential nodes from all candidates. Given a random process $RanCas()$, in each round i, the algorithm selects a node v (line 10) such that this node together with the previously selected ones in the set $\mathcal{S}$ maximizes the influence spread (line 17). In other words, the node v is selected and further included in $\mathcal{S}$ as it can maximize the incremental influence spread in this round.

Algorithm: *CCGA*

Input: network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, size of results k
Output: set $\mathcal{S}$ denoted the top-k influencers

1: $\mathcal{C} \leftarrow CommunityDetection(\mathcal{G})$
2: $\mathcal{C}' \leftarrow CommunityCombination(\mathcal{C}, k)$
3: $\mathcal{D} = \emptyset$
4: **for** $i = 1$ to $|\mathcal{C}'|$ **do**
5: $\quad \mathcal{I}_i \leftarrow CentralityDetection(\mathcal{C}'_i, k)$
6: $\quad \mathcal{D} = \mathcal{D} \cup \mathcal{I}_i$
7: **end for**
8: $\mathcal{S} = \emptyset, \mathcal{R} = 20000$
9: **for** $i = 1$ to k **do**
10: $\quad$ **for each** node $v \in \mathcal{D} \setminus \mathcal{S}$ **do**
11: $\quad\quad s_v = 0$
12: $\quad\quad$ **for** $j = 1$ to $\mathcal{R}$ **do**
13: $\quad\quad\quad s_v += |RanCas(\mathcal{S} \cup \{v\})|$
14: $\quad\quad$ **end for**
15: $\quad\quad s_v = s_v / \mathcal{R}$
16: $\quad$ **end for**
17: $\quad \mathcal{S} = \mathcal{S} \cup \{\text{argmax}_{v \in \mathcal{D} \setminus \mathcal{S}}(s_v)\}$
18: **end for**
19: **return** $\mathcal{S}$

Fig. 2. The community centrality-based greedy algorithm.

However, to ensure the influence spread of $\mathcal{S} \cup \{v\}$, the $RanCas()$ process is repeated $\mathcal{R}$ times, and the values of those spreads are then averaged (lines 11–15). Finally, the algorithm is terminated by returning k selected influencers in $\mathcal{S}$ at line 19.

4 Experiments

4.1 Experimental Setup

We conducted experiments to evaluate the effectiveness and efficiency of the proposed *CCGA*, compared with the state-of-the-art greedy algorithm (*GA*) [8] and *NewGreedy* [1]. We used a dataset excerpted from the publicly available Higgs Twitter dataset [4], which contains 19,483 individuals and 393,136 connections including topological relationships and interactions.

All the experiments were conducted on a server with 2.4 GHz Intel Xeon 8-Core CPU and 32 GB main memory, running Centos/7.0 operating system. The programs were coded using JAVA language.

4.2 Evaluation Metrics

We evaluate the effectiveness of an algorithm in term of the *influence degree*, i.e., the proportion of active nodes to the entire ones in a network. Let $\mathcal{S}$ be the

initial set of influencers, and $\mathcal{V}_{\mathcal{S}}$ be the set of nodes influenced by $\mathcal{S}$ during the information diffusion process. Then, the influence degree of set $\mathcal{S}$ is calculated as:

$$\mathcal{A}(\mathcal{S}) = \frac{|\mathcal{V}_{\mathcal{S}}|}{|\mathcal{V}|}.$$

To evaluate an efficiency of the algorithms, we measure it in term of the running time spent during the information diffusion process. However, for fairness comparisons, we therefore report the time consumed by *CCGA* in total, including all four processes as described in Sect. 3.

4.3 Results

Figures 3 and 4 report experimental performances in term of the influence degree and the running time for each individual parameter k, respectively. Notice that,

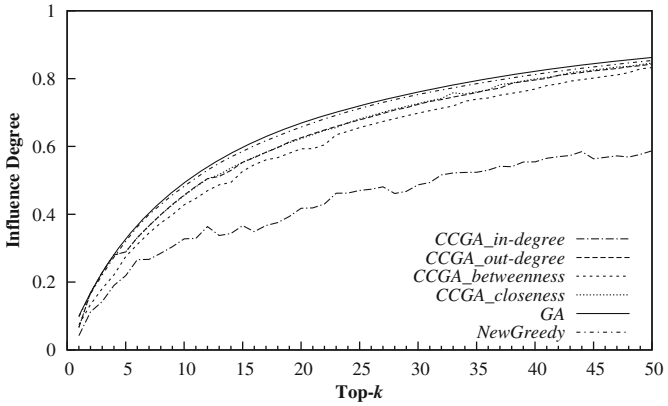

Fig. 3. Influence degree of different algorithms.

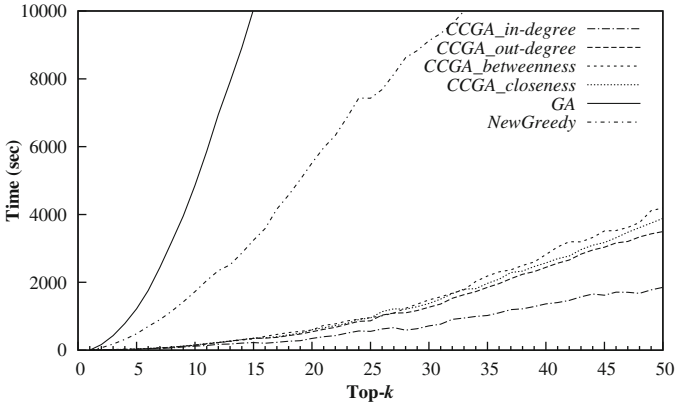

Fig. 4. Running times of different algorithms.

in Fig. 4, we excerpt the curves to show only at most 10,000 s for clear comparison reason.

As it can be seen from the results, all variations of *CCGA* (except *CCGA_in-degree*) can produce the influence degree closed to *GA* and *NewGreedy* while consume significantly lower time, indicating that they are quite effective and indeed efficient. Although *CCGA_in-degree* tents to spend the lowest time as k increases, it results the lowest influence degree. Consequently, *CCGA_out-degree* seems to be the best one since it can produce quite high influence degree and takes the second lowest time.

5 Conclusion

In this paper, we investigate the problem of influence maximization. We propose four variations of community centrality-based greedy algorithm. The experimental results show that our algorithms not only can execute much faster than the state-of-the-art greedy and *NewGreedy* algorithms but also still provide nearly the same effectiveness in influence spread.

For the future work, we anticipate to explore other graph-based clustering algorithms to detect the communities. We also interest to experiment with other social network datasets derived from Facebook, Google+, etc. To accelerate the running time of influence maximization, we plan to extend our algorithm in a parallel computing environment.

References

1. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208 (2009)
2. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the IEEE International Conference on Data Mining, pp. 88–97 (2010)
3. Chen, Y.C., Zhu, W.Y., Peng, W.C., Lee, W.C., Lee, S.Y.: CIM: community-based influence maximization in social networks. ACM Trans. Intell. Syst. Technol. **5**(2), 25:1–25:31 (2014)
4. De Domenico, M., Lima, A., Mougel, P., Musolesi, M.: The anatomy of a scientific rumor. Scientific Reports 3(2980) (2013)
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 57–66 (2001)
6. Galstyan, A., Musoyan, V.L., Cohen, P.R.: Maximizing influence propagation in networks with community structure. Phys. Rev. E. **79**(5), 056102 (2009)
7. Goyal, A., Lu, W., Lakshmanan, L.V.S.: CELF++: Optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th International Conference on World Wide Web (Companion Volume), pp. 47–48 (2011)

8. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)
9. Kempe, D., Kleinberg, J.M., Tardos, É.: Influential nodes in a diffusion model for social networks. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1127–1138. Springer, Heidelberg (2005)
10. Kim, C., Lee, S., Park, S., Lee, S.: Influence maximization algorithm using markov clustering. In: Hong, B., Meng, X., Chen, L., Winiwarter, W., Song, W. (eds.) DASFAA Workshops 2013. LNCS, vol. 7827, pp. 112–126. Springer, Heidelberg (2013)
11. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429 (2007)
12. Mehmood, Y., Barbieri, N., Bonchi, F., Ukkonen, A.: CSI: community-level social influence analysis. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013, Part II. LNCS, vol. 8189, pp. 48–63. Springer, Heidelberg (2013)
13. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford Digital Libraries (1999)
14. Van Dongen, S.: Graph clustering via a discrete uncoupling process. SIAM J. Matrix Anal. Appl. **30**(1), 121–141 (2008)
15. Wang, Y., Cong, G., Song, G., Xie, K.: Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1039–1048 (2008)
16. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Structural Analysis in the Social Sciences. Cambridge University Press, New York (1994)