Research Article

Xue-Guang Wang*

Research on Critical Nodes Algorithm in Social Complex Networks

DOI 10.1515/phys-2017-0008
Received Aug 10, 2016; accepted Sep 14, 2016

Abstract: Discovering critical nodes in social networks has many important applications and has attracted more and more institutions and scholars. How to determine the K critical nodes with the most influence in a social network is a NP (define) problem. Considering the widespread community structure, this paper presents an algorithm for discovering critical nodes based on two information diffusion models and obtains each node’s marginal contribution by using a Monte-Carlo method in social networks. The solution of the critical nodes problem is the K nodes with the highest marginal contributions. The feasibility and effectiveness of our method have been verified on two synthetic datasets and four real datasets.

Keywords: Critical Node Problem; Complex Networks; Community Structure; Influence Maximization

PACS: 01.20.+x; 07.05.Mh

1 Introduction

The analysis of social networks has many important applications and has attracted more and more institutions and scholars. How to identify influential individuals from a network is called a critical node problem (CNP), and this is one of the basic problems in the field of social network analysis. An effective solution for this problem will have an important practical value [1]. For example, we can better target and isolate the source of the disease and block its spread and diffusion in the disease network; or companies may identify individuals with influence and let them recommend products to their friends so that a cascade spreads by the greatest extent.

Domingos and Richardson first studied the CNP as an algorithmic problem and presented a greedy approximation algorithm. Currently, the research on the CNP of social networks mainly concentrates on improving the greedy algorithm or developing new methods by graph theory and complex network theory. Considering the widespread community structure in social networks, this paper gives a solution for the CNP, which assigns a marginal contribution for every node in a community of a social network using the solution concept and union concept of cooperative games. Then we sort all nodes by their contribution and obtain critical nodes according to some rules.

2 Background

The models for information propagation in networks have been widely studied [2]. We consider two models in this paper: independent cascade model (ICM) and linear threshold model (LTM) [2].

1) ICM Model. In this model, a propagation probability \( p_{u,v} \) is given for each edge \((u, v) \in E\), that is, vertex \( v \) is activated with probability \( p_{u,v} \) by \( u \). When an initial set \( A_0 \) of active vertices is given, the diffusion process spreads up according to the following randomized rule. When a vertex \( u \) is activated at time-step \( i \), it has a single chance of activating its neighbor \( v \) with \( p_{u,v} \). If \( u \) succeeds, \( v \) will become active at time-step \( i + 1 \). Here, if \( v \) has multiple parent vertices that become active at time-step \( i \) for the first time, then their activation attempts are sequenced in an arbitrary order. Whether or not \( u \) succeeds, it cannot make any further attempts to activate \( v \) in subsequent steps. The process runs until no more activation is possible.

2) LTM Model. In this model, vertex \( v \) is influenced by each neighbor \( u \) according to a weight \( w_{v,u} \) such that \( \sum_u w_{v,u} \leq 1 \). Each vertex has a predefined threshold \( \theta_v \in [0, 1] \), which is chosen uniformly at random. When an initial set \( A_0 \) of active vertices is given, the diffusion process unfolds according to the following randomized rule. All activated vertices at time-step \( i \) still keep active at time-
step $i + 1$. Whether or not any inactivated vertex is activated is determined by its neighbors’ weights such that $\sum_v w_{v,u} \geq \theta_v$. The process runs until no more activation is possible.

The difference between ICM and LTM is that (for ICM,?) each attempt of activation is independent of the attempts by all the other active individuals, while in the latter model each inactive individual is influenced by the aggregated weight of all its active neighbors.

A network is modeled as a graph $G = (V, E)$ with vertices in $V$ modeling the individuals in the network and edges in $E$ modeling the relationship between individuals. A vertex has two states: active and inactive, which in this context means whether a product or idea is accepted by individuals or not. We assume that a vertex can only change from inactive to active and not vice versa; an inactive vertex can be activated by its active neighbor vertices and an active vertex can activate its inactive neighbors; the increment of activated vertices represents the dissemination of information.

The definition of the CNP is as follow:

Input: $G = (V, E)$ and an integer $k$;
Output: $A = \arg \max_{A \subseteq V, |S| = k} \sigma(S)$

For $k - CNP$, we hope to find out a set $A$ with $k$ elements and maximize $\sigma(A)$ which is a NP-hard (define). But, Nemhauser et al. have approved that there is a greedy algorithm (see Algorithm 1) which approximates the optimum within a factor of $(1 - 1/e)^4$.

**Algorithm 1 Greedy Algorithm**

1. $A = \emptyset$
2. for $i = 1$ to $k$
3. $v_i = \arg \max_{u \in V \setminus A} (\sigma(A \cup \{u\}) - \sigma(A))$
4. $A = A \cup \{v_i\}$
5. end for

There is a key problem in how to compute the value of $\sigma(A)$ in Algorithm 1. Currently, we do not have any efficient method to obtain its exact solution. However, we can use a Monte-Carlo method to simulate the process of influence spread for obtaining approximate results by high probability.

### 3 The Algorithm

#### 3.1 Introduction to the algorithm

Given a finite set of players $N$, a cooperative game with transferable utility as a pair $(N, \nu)$, characteristic function $\nu : 2^N \to \mathbb{R}$ and $\nu(\emptyset) = 0$. For all $i \in N$, if payoff vector satisfies $x_i \geq \nu(\{i\})$ and $\sum_{i=1}^N x_i = \nu(N)$, then it is so-called an allocation of $(N, \nu)$. The solution of a cooperative game is a kind of allocation rule and the allocated payoff for every player denotes a method to measure the negotiation strength of the players in the game. Shapely (cite) presented a solution concept which determines the only allocation distribution scheme from the solutions with different property, i.e., it assigns the player’s payoff according to the importance of every player for the game [5]. The Shapely value of the player $i$ in the game $(N, \nu)$ is

$$Sh_i(\nu) = \sum_{\{S \subseteq N \mid i \in S\}} \frac{(n-s)(s-1)!}{n!} (\nu(S \cup \{i\}) - \nu(S)), \quad \forall i \in N$$

where $n = |N|$ and $s = |S|$.

However, the Shapely value does not consider the impact of coalition structure and Owen extends it [6]. Each union obtains its payoff from the game between the unions, and then the payoff is allocated by the internal game among the members of the union. All the payoffs are computed by the Shapely value.

Assuming that $N = \{1, 2, \ldots, n\}$ and $M = \{1, \ldots, m\}$, a partition $P = \{N_1, N_2, \ldots, N_m\}$ is a coalition structure on $N$. Let $N_k$ a union and $\bigcup_{1 \leq k \leq m} N_k = N$. When $i \neq k$, $N_i \cap N_k = \emptyset$. For $i \in N$, $k(i)$ denotes the index of the union containing player $i$, so $k(i)$ is defined by the relation $i \in N_{k(i)}$. For $k \in M$ and $S \subseteq N_k$, the game $\bar{\nu}_S$ is defined by

$$\bar{\nu}_S(Q) = \begin{cases} \nu(\cup_{h \in Q} N_h), & k \not\in Q \\ \nu(\cup_{h \in Q \setminus \{k\}} N_h \cup S), & k \in Q \end{cases}$$

where $Q \subseteq M$.

The game $\bar{\nu}_k$ is defined by $\bar{\nu}_k(S) = Sh_k(\bar{\nu}_S)$, then the Owen value of the player $i \in N$ in the game $\bar{\nu}_{k(i)}$ is $Ow_i(\nu, P) = Sh_i(\bar{\nu}_{k(i)})$.

#### 3.2 A new algorithm

Because the community structure is prevalent in social networks [7], we respectively consider the community’s influence on the information diffusion and every node’s influence in the community. We take the nodes in the social network as the players in the cooperative game and information diffusion as coalition formulation. We can map the information diffusion into a cooperative game with transferable utility which is formalized in the literature [8]. Since we can use the Owen value to obtain the marginal contribution of every node. Because the Owen value can be seen as a two-step procedure in which the Shapely value applies twice, we firstly compute a node’s Shapely value.
Given a node $i \in N$ and a subset $S \subseteq N$ such that $i \not\in S$, the marginal contribution of node $i$ is $\nu(S \cup \{i\}) - \nu(S)$, $\forall S \subseteq N \setminus \{i\}$. Consider the set of all possible permutations $\Psi$ on $N$, let $\psi \in \Psi$ and define $S_i(\psi)$ to be the set of all nodes appearing before node $i$ in the permutation $\psi$. So, the average marginal contribution of node $i$ to the given coalition game is

$$\frac{1}{n!} \sum_{\psi \in \Psi} [\nu(S_i(\psi) \cup \{i\}) - \nu(S_i(\psi))].$$

Note that, this method must work with $n!$ permutations and its computational complexity is $O((n/e)^n)$. Therefore, we give the approximate method for computing the Shapely value. Randomly generated $t$-sets $\Psi_t$ with $t$ permutations, see Algorithm 2, let $\psi \in \Psi_t$ and $\psi(i)$ denotes the $i$th node in the permutation. The number of activated nodes after running the diffusion model when the node $i$ is activated is the contribution of $\psi(1)$. Next, we consider the node $\psi(2)$. If $\psi(2)$ becomes active after $\psi(1)$ is activated, then the contribution of $\psi(2)$ is 0. Otherwise, the contribution of $\psi(2)$ is the number of activated nodes by $\psi(2)$. Therefore, we can obtain the contributions of $\psi(3), \ldots, \psi(n)$. For $\psi \in \Psi_t$, we repeat the above process $R$ times. Then the average contribution of each node in the diffusion process can be calculated. We can obtain the top-k nodes by sorting by the greatest influence and ensure that they are not adjacent to each other. See Algorithm 3 and Algorithm 4.

**Algorithm 2. CREATE $\Psi_t(n)$**

1: $t$=random(n)
2: m=random(n)
3: Select_t (n)
4: for $i=0$ to $m$
5: \hspace{1em} arr[$i$] = random() % (n-i)
6: \hspace{1em} for $j=0$ to $i$
7: \hspace{2em} if arr[$j$] <= arr[$i$]
8: \hspace{3em} arr[$j$]=arr[$j$]+1
9: \hspace{1em} end if
10: \hspace{1em} end for
11: \hspace{1em} v = arr[$i$]
12: \hspace{1em} for $k=i+1$ to $j$
13: \hspace{2em} arr[$k$]=arr[$i$]
14: \hspace{1em} $k=k-1$
15: \hspace{1em} end for
16: \hspace{1em} arr[$j$] = v
17: end for
18: for $h=2$ to $t$
19: \hspace{1em} $|\Psi_{h-1}|$=arr
20: \hspace{1em} n=n-m
21: \hspace{1em} m=random(n)
22: Select_t(n)
23: end for
24: $|\Psi_n|$=arr

**Algorithm 3. ShapelyValue(v, R)**

1: $n=|\psi|$
2: $t=|CREATE_\Psi_t|$
3: tmp[1...n]=0
4: ShV[1...n]=0
5: for $i=1$ to $t$
6: \hspace{1em} for $r=1$ to $R$
7: \hspace{2em} if $j=1$ to $n$
8: \hspace{3em} tmp[$j$]=v($S_i(\psi_i) \cup \{j\}$) - v($S_i(\psi_i)$)
9: \hspace{2em} end for
10: \hspace{1em} end for
11: end for
12: end for
13: for $i=1$ to $n$
14: \hspace{1em} ShV[i]=tmp[i]/$t$
15: end for

**Algorithm 4. TOP_K(V)**

1: TopK[1...k]=0
2: AsceSort(V);
4: i=1
5: j=2;
6: while $i < k$
7: \hspace{1em} if V[j] is not adjacent to TopK[1...i-1] then
8: \hspace{2em} TopK[i]= V[j]
9: \hspace{2em} i=i+1
10: \hspace{1em} end if
11: \hspace{1em} j=j+1
12: end while

According to the description in Section 3.1, we provide the calculation method of the Owen value. Roger Guimerà, et al studied node roles in a community according to within-module degree $z$ and participation coefficient $P$, which are divided into seven categories $Role = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7\}^{10}$. We consider two types of the roles: Non-hub connector node ($z < 2.5$ and $0.62 < P \leq 0.80$, R3) and Connector hub ($z \geq 2.5$ and $0.30 < P \leq 0.75$, R6). Most of the nodes which belong to these two roles connect to other communities.

We use the CNM algorithm [11] to divide the network $G = (V, E)$ into $l$ communities $C = \{C_1, C_2, \ldots, C_l\} = (V_l, E_l)$, $i = 1, \ldots, l$ and assign the role $r_{ij} \in Role$ for every node $v^l_i \in V_l$ in the community. Let $Role' = \{R_3, R_6\}$, $i = 1, \ldots, l$, $j = 1, \ldots, l$,

$$V' = \{v^l_i, v^l_j | \exists (v^l_i, v^l_j), v^l_i \in V_l, r_{ij} \in Role', i \neq j\},$$
\[ E' = \{ (v_{ih}^h, v_{ig}^j) \mid (v_{ih}^h, v_{ij}^h) \in E_i, v_{ih}^h \in V_i, v_{ig}^j \in V_i, rv_{ih}^h \in \text{Role'}_i, i \neq j \}, \]

define \( G' = (V', E') \) as the community game network. So, we can obtain the Shapely value of every node in the network \( G' \) and take the sum of the Shapely values of all nodes in the same community as the community’s payoff. Then we treat a community as a separate network and calculate the Shapely value of every node in the community. The node’s Owen value is assigned according to the normalized Shapely value of the node in the community and the community’s payoff. So, we can obtain the \( k \) critical nodes by Algorithm 4.

4 Experiments

We validate our method using two synthetic datasets and four real network datasets. All experiments are executed using a PC with 3.2GHz CPU, 4G Memory and Windows 7. The development tools are Matlab 2009 and Microsoft Visual Studio 2010.

4.1 Datasets

We use the BA model [12] and the Forest Fire model [13] to generate two synthetic datasets with 5000 nodes. The BA model takes the power-law distribution and has two important features: growth and preferential attachment. The model can generate a scale-free network whose power exponent is 3 and community structure is not obvious. The Forest Fire model can generate a network with degree power law, densification law, shrinking diameter, and obvious community structure.


4.2 The results

We compare our method CNP-OV with the greedy algorithm GREEDY (Algorithm 1), the degree-heuristic algorithm DEGREE and the random algorithm RANDOM. DEGREE selects \( k \) nodes with the greatest degree from the network as the initial set; RANDOM randomly selects \( k \) nodes as the initial set. In order to obtain the accurate influence of every algorithm, we use the average number of the activated nodes after running ICM and LTM 10000 times for every initial set. In ICM, the propagation probability is set to 0.05; in LTM, the edge-weight of a node is the reciprocal of the node’s degree. The size of the initial set is respectively set from 1 to 20. In the experiments, we only discuss the case of using the ICM because we obtain the same conclusions for ICM and LTM.

First, we consider the influence of community structure on the CNP-OV. We respectively compute the Shapely value and Owen value of every node in the BA and FF datasets and obtain the initial set and the number of activated nodes after running the ICM. The process is repeated 100 times and the average number of the nodes activated...
by the initial set with different size is drawn in Figure 1. The dashed Sv denotes the method based on the Shapely value and the solid line Ov is the method based on the Owen value. The Sv is almost the same as the Ov for the BA dataset with an unobvious community structure (Figure 1a). In contrast, the Ov is significantly better than the Sv for the FF dataset with an obvious community structure (Figure 1b).

Secondly, we analyze our method’s accuracy. We respectively use GREEDY, DEGREE, RANDOM and CNP-OV to determine the initial sets from above for four real datasets. Figure 2 describes the number of nodes activated by the initial set with different size based on ICM. The accuracy of the CNP-OV is similar to GREEDY, sometimes even better than it (Figure 2b). Compared with DEGREE and RANDOM, the CNP-OV provides a large advantage.

5 Conclusions

Considering the widespread community structure in social networks, for solving the CNP, in this paper we present a new algorithm for discovering critical nodes based on cooperative games. It assigns a marginal contribution for every node in a community of social networks using the solution concept and union concept of cooperative games. We sort all nodes by their contribution and obtain critical nodes according to selected rules. We have validated its feasibility and effectiveness. The next step will be to study how to improve the time efficiency of the CNP-OV.

Acknowledgement: This work is supported by the National Social Science Foundation of China (No.11BFX125), Pujiang Talent Project, Peak of law subject construction Project and Public Security Discipline Construction Foundation.
References


