Application of Community Detection Algorithm with Link Clustering in Inhibition of Social Network Worms

Yibing Wang[1], Jie Fang[1,2], and Fuhu Wu[3]
(Corresponding author: Yibing Wang)

Center of Computer Teaching, Anhui University[1]
No.111 Jiulong Road, Hefei, Anhui 236061, China
School of Electronics and Information Engineering, Chinese Academy of Sciences[2]
No.350 Shushanhu Road, Hefei, Anhui 230031, China
Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University[3]
No.3 Feixi Road, Hefei, Anhui 230039, China
(Email: wyb@ahu.edu.cn)
(Received Dec. 15, 2015; revised and accepted Mar. 6 & Mar. 28, 2016)

Abstract

The community detection was performed from the perspective of links, and we proposed an inhibition method against social network worms. Firstly, a community detection algorithm was proposed, which based on link clustering, and we got related link incremental information through the network structure information at various time points. In order to obtain the link communities, we adopted an improved link partition density function to dispose the link incremental information. Next, we gave three selection strategies of key nodes in community and proposed corresponding worm inhibition method. Finally, on the basis of real web data sets, we applied community detection and worm inhibition experiments to prove validity of algorithm in this paper.

Keywords: Community detection, link clustering, partition density, worm inhibition

1 Introduction

Social network connects users in the virtual network space, extends the human communication, information sharing and the social activity space, which is becoming the most influential internet application. The typical applications include Facebook, QQ, Renren, Sina Weibo, BBS and other shared spaces, etc. [25, 30].

As an extension of real world in the virtual network world, "birds of a feather flock together, and people of one mind fall into the same group", community structure is an important structure of social networks, which is also a kind of important structure for mesoscopic observation and the network topology analysis [29]. It makes the community internal nodes closer, and the connection between communities looser [19]. The process of finding community structure in the complex network is the community detection, which has important theoretical basis and practical significance for the network structure analysis in real world.

In recent years, based on the different understanding of community structure, scholars have put forward a lot of community detection algorithms [2, 8, 13, 18, 24, 28, 33]. At present, some algorithms can correctly extract community structure from small-scale social networks, which can be roughly divided into three categories: the method based on graph theory, such as GN [8] and FastGN [18]; Algorithms based on matrix decomposition, such as SymNMF [13]; The method based on the optimization, such as N-Cut and A-Cut [24], etc. Among them, in 2001, Girvan and Newman proposed the GN algorithm, setting off a new wave of research. In recent years, it has become a standard algorithm of community structure analysis. GN makes up the inadequacy of some traditional algorithms, which does not have to rely on redundant information, and can directly analyze from the network topological structure. But the biggest drawback is that it is unable to determine when to terminate the operation, eventually making the results too granular. The time complexity of this algorithm is $O(n^3)$, where n is the number of nodes in network. In order to improve the time-consuming shortcoming of GN, Newman proposed the FastGN where each node was seen as a community, and the two communities combined with the maximum Q value in each iteration until the entire network integrated into one community. The whole process can be represented as a tree diagram, and choose the hierarchical division with maximum Q value

to get the final community structure. The overall time complexity of the algorithm is $O(m(m+n))$, where m is the number of edges in network, and n is the number of nodes. This method makes to lower the time complexity of GN greatly. Clauset et al. [5] used stack to calculate and update the network modularity, and proposed a new greedy algorithm — CNM. The algorithm further accelerates the FastGN, getting close to linear complexity.

The community structure of large-scale complex networks, however, often has overlapping characteristics, that is, a node belongs to different communities. Both GN and its improved algorithms have a problem, that is, a node only belongs to one community. But it is not the case; each node can have different identities in different circumstances. In order to solve this problem, Palla et al. [22] proposed a clique filtering algorithm to analyze the overlapping community structure, introduced the concept of k-clique community. Ahn et al. [1] put forward the new idea to detect the community structure with overlaps and hierarchy — the edge detection algorithm. But these methods failed to solve the problem that a node belonged to multiple communities.

In addition, in order to avoid the limitation of priori information, Raghavan et al. [23] proposed a fast Label Propagation Algorithm (LPA) based on the idea. The algorithm firstly assigned the only label for each node. In every iteration, each node updated its own label to the label most frequently appears in neighboring nodes. If there were many same labels, randomly selected one as an updated value, after several iterations, the densely connected nodes would converge to the same label. In the end, the node with the same label would come into one community. LPA is simple, rapid and effective, but lacks high accuracy.

The above algorithms are only effective in the community detection in the small-scale networks; when the network scale is increasing, the efficiency will decrease obviously, and the algorithm complexity also increases exponentially along with the growth of the network dimension. Researchers adopt different standards and policies when partitioning nodes, deriving a lot of different styles of the new algorithms [17], including module optimization algorithm, spectrum analytical method, information theory method, and label transmission method. However, it is hard for these methods to find a good balance point between time complexity and accuracy.

Although many achievements have been made about community detection of complex network at home and abroad, some problems exist in these methods, that is, the algorithms are usually designed for a specific network or certain features of network, which are not suitable for most networks. At present, through the relevant research and analysis, it can be found that mining overlapping community structure is of great significance from the angle of link [1]:

1) Compared to the independent nodes in the network, the link between nodes can express more information;
2) Abstract network into a large number of links, mine these links sets to directly get the overlapping community structure, which is an intuitionistic expression without other auxiliary measures. These suggest that: finding overlapping community structure in complex networks from the perspective of link is more convenient. Therefore, this paper proposes the Link Clustering based Community detection algorithm (LCC); first, obtain the related link incremental information [6] through the network structure at any time, and then handle the link incremental information based on the improved link partition density function, with the improved link module as the objective function, then the link communities are obtained.

With the continuous development and large-scale popularity of social networks, some security issues start to emerge. For example, real-time resource sharing and interactive services provided in social network have attracted a large number of users, while, in the meantime, the frequent interaction between users also provides an effective way for the rapid spread of Internet worm. Different from traditional worm viruses, social network worm is a kind of malicious program that does not rely on particular system vulnerabilities. It uses its own camouflage to deceive users to click and execute the program to get infected, then it spreads through social networks to infect the user's friends, a large number of clicks and sharing among friends accelerate the proliferation of worms. Social network worm is characterized by high concealment, long life cycle, difficult to eradicate, etc. It is difficult to effectively control its dissemination through technology of patches release, which increases its potential damage. At the same time, with the increasing number of Internet users and the rapid development of various forms of virtual social networks, the social network worm has become one of the major hidden dangers for network security.

Traditional worm model is based on mathematical model, considering the similarity in propagation between computer worm and biological virus, it introduces SIS, SIR and other models which are widely used in biological virus propagation modal into computer worm model, so as to analyze and predict the features and trends of worm propagation [14]. Researchers begin to realize such external factors as the network topology, bandwidth and user countermeasures impact on the spread of worm propagation. For example: Yang and others [32] took Rose mail worm as an example, by establishing the mathematical model, they researched worm propagation in different social occasions such as Print Service Office Internet Cafe Friendship Network, and these occasions have added immune factors; Considering the following two factors could affect velocity of worm propagation: first one, the network users' countermeasures to worm, second one, fast-moving worm leads to retardation because of router block, Zou and others [35] proposed Two-Factor model, worm infection rate , host immunity and some parameters

were expressed as a Function of time T, and adjusted its value according to the change of infected host quantity. The models mentioned above only describe nodes infected number in unit time, but they can't reflect worm's infection route in network topology. In Reference 18, based on social topology, Faghani and others proposed XXS worm model which used undirected network topology, whereas, it was not conform to aeolotropism in real social network topology.

In Reference 19, Nguyen and others presented primarily worm inhibition method which based upon community structure, this method made use of popular BGLL, it was not necessary to provide division quantity to detect reasonable community structure for users. What's more, Nguyen and others gave selection strategies on key nodes in community, which means the nodes which possess the most connections between the community and other communities could be defined as key nodes, and gave these nodes immunization or issued patches primarily. But in their paper, the authors can't prove the selection of key nodes in theories or experiments.

In view of all kinds of hazards caused by social network worms, the defensive measures put forward by researchers mainly include two aspects:

1) Social network worm detection [3, 16, 26];

2) Social network worm inhibition [20, 36].

Among them, the social network worm detection can be divided into client and server detection according to the location of the detection. The client detection method mainly uses constantly updated feature library to match and detect the spread of worm. But when there is a new type of worm, limited by the bandwidth of existing network, it is hard for it to distribute the new features to the network test system of all users, so the method has certain delay. Server detection mainly captures the number of malicious messages in the network through the website server, but this method cannot detect worms until the malicious message spreads to a certain degree. So this method also has the unavoidable delay.

Although the worm inhibition method in social network is unable to timely detect the spread of the worm, it can reduce the number of infected users to maximum extent. At present, researchers generally start from the community structure of complex networks, in other words, find the community in the network first, and then adopt relevant measures to select key nodes in the community, finally, conduct immune operation for these key nodes, thus ensuring to immunize other nodes at full speed. This paper, based on such idea, conducts worms inhibition in social network. According to the choice strategy of key nodes in the community, we define the nodes connected with most other communities as the key nodes, then immune these nodes, finally immune the neighboring nodes with the help of these key nodes, thus effectively restraining the rapid spread of network worms.

The remained paper is divided into four sections. Section 2 introduces the concrete realization of Link Clustering based Community detection (LCC). Section 3 introduces the selection of key nodes and worms inhibition method, and Section 4 describes the experimental results on real data sets, and Section 5 summarizes the full paper.

2 Link Clustering Based Community Detection Algorithm

On the basis of existing methods, this work proposes a worm inhibition method, which based on dynamic community mining, as shown in Figure1. This method can be divided into four stages: original data pre-processing, dynamic community detection, key nodes abstraction, worm inhibition. First, through network structure information at various points, we get related link increment, and then we adopt improved Link Partition Density Function to process the increment information, counting the improved link modularity value as objective function, so as to detect community to get community structure. Next, we propose three different strategies to choose key nodes, analyze and compare worm inhibition effect under different strategies by comparison experiments in the fourth section. At last, we give these key nodes immunization, with the help of these nodes, give immunization to the neighbor nodes, in order to achieve the desired inhibitory effect of worm rapid spreading.

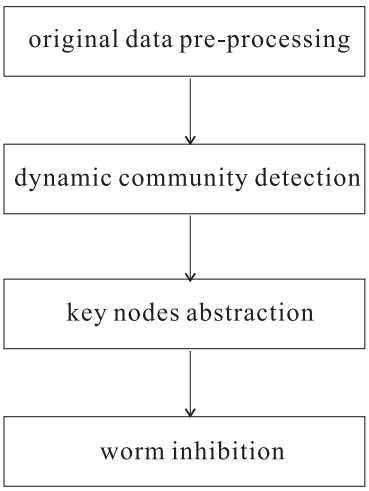

Figure 1: The diagram of worm inhibition

2.1 Link Partition Density

For a given network $G(V, E)$, where V is the node set in the network, and E is the edge set, $C = \{C_1, C_2, \cdots, C_k\}$ represent the community sets in the network.

Definition 1. *For the given network* G, *link graph* L_G *is the link aggregation formed by connection be-*

tween nodes in G, *as shown in Figure 2, where* $L_G = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16\}$.

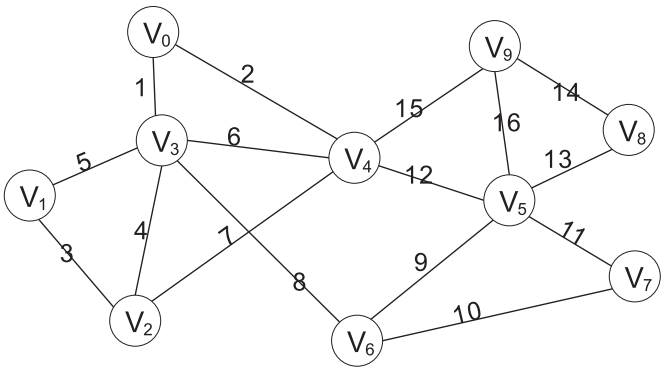

Figure 2: The link graph

Definition 2. *In the given link* $L_1(L_1 =< v_1, v_2 >)$ *and* $L_2(L_2 =< v_3, v_4 >)$, v_1, v_2, v_3, v_4 *are the nodes. If* $L_1 \cap L_2 =< v_1, v_2 > \cap < v_3, v_4 >\neq \emptyset$, *the link* L_1 *and* L_2 *are the neighboring links.*

In Figure 2, Link 1 and Link 2 share the node V_0, so Link 1 and Link 2 are the neighboring links.

Definition 3. *Given link communities* C_i, C_j *and link* $L =< v_1, v_2 >$ *in moment* t, *if* $\{L|L \notin C_i, L \notin C_j, v_1 \in C_i, v_2 \in C_j, i \neq j\}$, L *is the bridge link.*

Inspired by the literature [1], this paper proposes an improved link partition density function for dealing with incremental information. Assuming that a network has M links, and the network is divided into C link subsets by $\{P_1, P_2, \cdots, P_c\}$, the link partition density of community is:

$$D_c = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{m_b - (n_b - 1)}{\frac{n_b(n_b-1)}{2} - (n_b - 1)}. \tag{1}$$

In the above formula, m_c and n_c respectively represent the link numbers and node numbers in the subset P_c, m_b represents the bridge link numbers between communities, n_b represents the node numbers between communities. And meet $m_c = |P_c|$, $n_c = |\cup_{e_{ij} \in P_c} \{i, j\}|$, $n_b = |\cup_{e_{ij}, i \in P_c, j \notin P_c} \{i, j\}|$. Then, the improved link partition density D_L is defined as:

$$\begin{aligned} D_L &= \sum_c \frac{m_c}{M} D_c \\ &= \frac{2}{M} \sum_c [m_c \frac{m_c - (n_c - 1)}{(n_c - 1)(n_c - 1)} - m_b \frac{m_b - (n_b - 1)}{(n_b - 2)(n_b - 1)}]. \end{aligned} \tag{2}$$

In the given network $G(V, E)$, $C = \{C_1, C_2, \cdots, C_k\}$, set the incremental information in the network as $\varepsilon = \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n\}$. In the network, the links can be divided into two categories: Intra-community Links (IL), that is, the two endpoints are within the community; Bridge-community Links (BL), that is, the two endpoints are located in different communities. For each community C in G, when adding IL or removing BL, the community will be closer and the network structure will be clearer. On the contrary, removing IL link or adding BL will make the network structure even vaguer. When there is no interference between the two communities, or the disturbance is small, adding or removing the link may form a new community structure. Therefore, in the update of community structure, the subtle changes of network structure will lead to the huge change of community. From the perspective of the link, with the passage of time, the change of network is in fact the link adding or removing. Thus, the change information in the network can be simply described as the adding of new link or the removing of existing link.

2.2 Adding Link Algorithm

Theorem 1. *If* C_i *is a community in network* G, *add any IL to* C_i, C_i *won't break down into smaller modules.*

Proof. Formula (1) shows the link partition density of C_i. Assuming that the incremental information ε_i represents adding an internal link e to the community C_i. Set $D_c^{'}$ as the link partition density while adding e into the community C_i, then

$$D_c^{'} = \frac{(m_c + 1) - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{m_b - (n_b - 1)}{\frac{n_b(n_b-1)}{2} - (n_b - 1)}. \tag{3}$$

Obviously, $D_c^{'} > D_c$, therefore, when adding internal link e to community C_i, the community structure will be stronger. □

Theorem 2. *If the added link is between* C_i *and* C_j, *when the bridge link needs to be re-assigned, the community* C_i *and* C_j *are the first choice.*

Proof. Assuming that the added link e between C_i and C_j, because e's nodes are in the communities C_i and C_j, so when adding the link e to other communities, the value of D_L is not changed. For the community C_i, before adding link e, the link partition density is:

$$D_{L,i} = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{m_b - (n_b - 1)}{\frac{n_b(n_b-1)}{2} - (n_b - 1)}. \tag{4}$$

□

When adding link e, the link partition density is:

$$D_{L,i}^{'} = \frac{m_c - (n_c - 1)}{\frac{n_c(n_c-1)}{2} - (n_c - 1)} - \frac{(m_b + 1) - n_b}{\frac{(n_b+1)n_b}{2} - n_b}. \tag{5}$$

Set $\Delta_1 = D'_{L,i} - D_{L,i}$, obviously $\Delta_1 > 0$, so after adding the link e into C_i, the link partition density of C_i will increase. Similarly, set $\Delta_2 = D'_{L,j} - D_{L,j}$, obviously $\Delta_2 > 0$, the link partition density of C_j also increases.

To sum up, if the added link is between C_i and C_j, the communities C_i and C_j are the first choice.

Deduction 1. *If the added bridge link* e *is between* C_i *and* C_j*, when meeting* $\Delta_d = D_{L,i}(E+e) - D_{L,j}(E+e) + D_{L,j}(E) - D_{L,i}(E) > 0$*, the bridge link* e *will be assigned to the community* C_i*; otherwise, the bridge link* e *will be assigned to* C_j*.*

Proof. Theorem 2 shows that if the added bridge link e is between C_i and C_j, the communities C_i and C_j are the first choice, then

$$\begin{aligned}\Delta_d &= \Delta_1 - \Delta_2 \\ &= (D'_{L,i} - D_{L,i}) - (D'_{L,j} - D_{L,j}) \\ &= D_{L,i}(E+e) - D_{L,j}(E+e) + D_{L,j}(E) - D_{L,i}(E).\end{aligned} \tag{6}$$

□

When $\Delta_d > 0$, the bridge link e should be assigned to C_i, while $\Delta_d < 0$, the bridge link e should be assigned to C_j.

When adding a new link e, there are two kinds of situations:

1) Link e is completely in community C_i;

2) Link e is between C_i and C_j, where $i \neq j$. For Case (1), according to Theorem 1, the community structure remains the same. For Case (2), based on Theorem 2, if the bridge link e is assigned to the new community, the community must be one of C_i and C_j. Deduction 1 shows the assigning criteria of bridge link e.

Therefore, the algorithm of adding link is described as Algorithm 1.

Algorithm 1 *adding_link*

1: Enter new link e and link community structure C_t in moment t.
2: Output the link community structure C_{t+1} in moment $t+1$.
3: If e is the internal link, then $C_{t+1} \equiv C_t$, otherwise $k = argmax(\Delta d_i, \Delta d_j)$, add e to C_k, and update C_{t+1}.

2.3 Removing Link Algorithm

Deduction 2. *If the link* e *is the bridge link between* C_i *and* C_j*, when removing the link, the structures of* C_i *and* C_j *will be more apparent, and the whole community structure remains the same.*

Proof. When the removed link e is the bridge link between C_i and C_j, the link relation among nodes within the community does not change, but when the link between communities is removed, the connection of community will become looser, and the community structure in the network will be stronger and more obvious. As a result, the overall community structure will not change. □

When link e is removed, it can be divided into two cases:

1) The bridge link e is between C_i and C_j, $(i \neq j)$;

2) The link e is fully inside the community C_i. According to the deduction 2, for case (1), when removing the bridge link, the community will not change. For case (2), when the removed link e is an IL, set $S(e)$ as the neighboring link set of e, $\forall l \in S(e)$. If $C_k = argmax(D_{L,k}(l))$, assign the link l to the community C_k, where N is total number of link communities at present, and $1 \leq k \leq N$.

Removing link algorithm is as Algorithm 2.

Algorithm 2 *removing_link*

1: Input the removed link e and link community structure C_t in moment t.
2: Output the link community structure C_{t+1} in moment $t+1$.
3: If e is the community external link, then $C_{t+1} \equiv C_t$, otherwise $C_k = argmax(D_{L,k}(l))$, $l \in S(e)$, $k \in (1, N)$, add the link l into C_k, and update C_{t+1}.

To sum up, the Link Clustering based Community detection algorithm is described as Algorithm 3.

Algorithm 3 The LCC algorithm

1: Input $G_0 = (V_0, E_0)$, incremental information $\varepsilon = \{\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n\}$.
2: Output the community structure C_t of network G_t in moment t.
3: Find the link community structure G_0 at the initial moment.
4: Start from the initial moment, if $e \in adding\_link(L(u,v))$, then $adding\_link(C_t, L(u,v))$, otherwise $removing\_link(C_t, L(u,v))$.
5: Map the link community structure C_t into the node community, and obtain the nodes community structure at each moment.

3 Worm Inhibition Method in Social Network

3.1 Selection of Key Nodes

When inhibiting the social network worms, in addition to the community detection algorithm, the selection strategy

of key nodes can also affect the inhibition effect of social network worms. The formalized definitions of the key nodes are given here first.

In the given network $G(V, E)$, where $C = \{C_1, C_2, \cdots, C_k\}$ represent the community sets in the network. V_j^i represent the nodes in C_i, we use $|V_j^i|_{in}$ to indicate the connection numbers of V_j^i with other nodes in the community, also known as the internal node degree, and $|V_j^i|_{out}$ to signify the connection numbers of V_j^i with nodes in other communities, also known as the external node degree, use $|V_j^i|$ represent the connection numbers of V_j^i with other nodes. Obviously, $|V_j^i| = |V_j^i|_{in} + |V_j^i|_{out}$.

Definition 4. *(The maximum internal degree nodes.) In C_i, V_{maxin}^i is called as the maximum internal degree node, if and only if meeting the following formula:*

$$\forall V_j^i \in C_i, |V_j^i|_{in} \leq |V_{max}^i|_{in}. \quad (7)$$

Definition 5. *(The maximum external degree node.) In C_i, V_{maxout}^i is called as the maximum external degree node, if and only if meeting the following formula:*

$$\forall V_j^i \in C_i, |V_j^i|_{out} \leq |V_{max}^i|_{out}. \quad (8)$$

Definition 6. *(The maximum degree node.) In C_i, V_{max}^i is called as the maximum degree node, if and only if meeting the following formula:*

$$\forall V_j^i \in C_i, |V_j^i| \leq |V_{max}^i|. \quad (9)$$

In the real community, although the maximum internal degree node, the maximum external degree node and the maximum degree node could be the same node in most cases, there are also different situations, so three different node selection strategies are needed here.

Maxin strategy. Once the social network worms outbreak, select the maximum internal degree node V_{maxin} in the community. The nodes have the most links with other nodes in the community; this selection strategy is mainly based on local thoughts, because the nodes can immune other nodes in the community at full speed, thus inhibiting the spread of the worm in the community.

Maxout strategy. Once the social network worms outbreak, select the maximum external degree node V_{maxout} in the community. The nodes have the most links with other outside communities, which can not only prevent the worm spreading from other communities to this community, but also inhibit the spreading of the worm from this community.

Max strategy. Once the social network worms outbreak, select the maximum degree node V_{max} in the community. The nodes have the most neighbors, and the neighboring nodes can be either in the same community, or in other communities. The selection strategy is mainly based on the greedy thought, that is, immune the node with the strongest local transmission capacity first.

3.2 Inhibition Algorithm for Social Network Worms

According to three selection strategies of key nodes, we give the worm inhibition algorithm in social network after using LCC to obtain community structure of the social network.

Algorithm 4 The worm inhibition

```
Input: edge sets E, community structures C = {C_i}, and selection strategy P
Output: the key nodes set R
for C_i in C
    if (P is the maxin strategy)
        v ⟵ get_maxin(C_i, E)
    elseif (P is the maxout strategy)
        v ⟵ get_maxout(C_i, E)
    else
        v ⟵ get_max(C_i, E)
    if (v= NULL )
        R.add(v)
for v in R
    Issue immune notice to v
    Being immune, v spreads immune notice to its neighbors
return R
```

4 Experiments

4.1 Data Sets and Evaluation Indexes

In order to prove the validity of algorithm proposed in this paper, our method was tested in real-world web data sets and compared with other classical community detection algorithms, followed by verifying the validity of social network worms inhibition. 5 typical real-world web data sets were adopted for experimental analysis as illustrated in Table 1.

Table 1: Data sets

Data Sets	Nodes Amount	Edges Amount
Zachary Karate	34	78
Dolphin	62	159
Book US politics	105	441
Amercian college football	115	613
LiveJournal Social Networking Dataset	4847571	68993773

To evaluate the quality of community partition, the first evaluation criterion adopted was Q Modularity proposed by Newman and Girvan and the second was Normalized Mutual Information (NMI) proposed by Danon. Their definitions respectively as follows.

Standard definition of Q Modularity:

$$Q = \sum_i (e_{ii} - a_i^2) = Tre - \|e^2\| \tag{10}$$

$\|x\|$ means the sum of all elements in the x-matrix. First of all, a symmetric matrix of $k \times k$ was defined as $e = (e_{ij})$, in which e_{ij} refers the proportion of the lines connecting two nodes of different communities on the network in all lines. The two nodes are in the *ith* community and the *jth* community respectively. Suppose the sum of all the elements in the diagonals of matrix is $Tre = \sum_i (e_{ii})$ which refers the proportion of the lines connecting every node in some community on network in total of all lines. Then define the sum of every element in each line or each column as $a_i = \sum_j e_{ij}$, which refers the proportion of the lines connecting nodes in the *ith* community in all the lines. The upper limit of Q is $Q = 1$, thus the more closer to the value Q is, the more obvious community structure will be.

Standard definition of NMI:

$$NMI = \frac{-2\Sigma_{i,j} N_{ij} log(\frac{N_{ij} N}{N_{i.} N_{.j}})}{\Sigma_i N_{i.} log(\frac{N_{i.}}{N} + \Sigma_j N_{.j} log \frac{N_{.j}}{N})} \tag{11}$$

N_{ij} is the number of public nodes in clustering X_i and Y_j, N_i is the sum of the line *ith*, N_j is the sum of the column *jth*. NMI's value is between 0 and 1. When NMI=0, it indicates the two consequences completely inconsistent; When NMI=1, it indicates the two consequences completely consistent.

4.2 Community Detection Results

In order to verify the validity of LCC, in this section it was compared with algorithms like GN, LPA and BGLL [20, 21]. The comparative results between the average Q modularity acquired from 10 runs of LCC and from other 3 algorithms were given in Table 2.

Table 2: Comparison of Q modularity between our method and other algorithms

Data Sets	GN	LPA	BGLL	LCC
Zachary Karate	0.401	0.407	0.419	0.435
Dolphin	0.519	0.511	0.516	0.517
Book US politics	0.517	0.516	0.498	0.523
Amercian college football	0.599	0.598	0.602	0.611

Known from Table 2, as for Dolphin, the Q value acquired from LCC was slightly lower than that of GN while for other 3 data sets, the Q modularity value of LCC was the highest. Thus it was clear that LCC was able to perform community detection against large-scale complex networks.

The LCC and other algorithms such as GN, LPA, NFA, BGLL were acted on four known community structures (Zachary Karate, Dolphins, Book US politics, American college Football) and then the comparative results in NMI accuracy among such algorithms were given in Table 3.

Table 3: Comparison among different algorithms in term of NMI on real-world networks

Data Sets	GN	LPA	NFA	BGLL	LCC
Zachary Karate	0.58	0.84	0.69	0.59	1.0
Dolphin	0.55	0.59	0.57	0.52	0.63
Book US politics	0.56	0.51	0.52	0.57	0.66
Amercian college football	0.88	0.90	0.79	0.90	0.91

From Table 3 it was indicated that (1) The NMI value of optimum community partition by LCC for Zachary Karate was 1 and the community structure partitioned was shown in Figure 3. It could be indicated from Figure 3 that the community structure partitioned by LCC had a completely consistent structure with the real-world community structure. (2) For Dolphin, the NMI value of community partition by LCC was 0.63 and the community structure partitioned was shown in Figure 4. Seen from Figure 4, it was partitioned into 4 communities by LCC, in which the part represented by purple circle was corresponding with real-world community structure in Dolphin dataset, and by LCC the other part of real-world community structure was further partitioned into 3 tighter communities, which were represented as red square, green diamond, and blue triangle, respectively. (3) As for Book US polities and American college football sets, the NMI value acquired by LCC was higher than that by any other algorithm. Thus it was indicated that the community structure detected by LCC had a high accuracy.

4.3 Worm Inhibition Results

Related experiments were performed in order to verify the validity of algorithm proposed in this paper using for the inhibition of social network worms. Since the iterative method was used to analyze the propagation process of worms, here the end condition of iterative process was required for discussion. "Newly infected nodes" and "most nodes infected" were taken as two judgment conditions, by satisfying either of which the iteration might be terminated. In this experiment, the LiveJournal data set in Table 1 was selected to conduct four experiments. Then no inhibition means were adopted and the worm inhibition effects under 3 key node selection strategies stated in

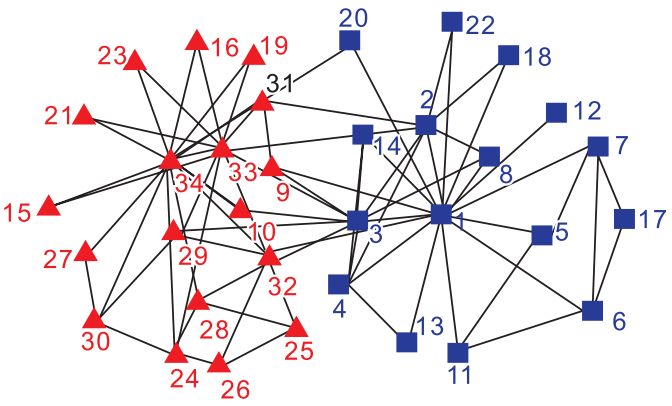

Figure 3: Detection results of LCC against Zachary Karate

Section 3.1 were used for comparison. The experimental results were shown in Figure 5.

During the experiments, the worms inhibition course started when the worm infection rate was over 2%. Seen from Figure 5, when adopting key node inhibition strategy, it was indicated that during the worms' propagation process, after 600 time units, the infected user amount basically remained at about 30%, showing that the inhibition scheme of key nodes in social network had better defense against worms. Meanwhile, seen from the inhibition effect under three key node selection strategies listed in Figure 5, the max strategy had the best worm inhibition effect.

The worm inhibition effect of maxout strategy and other worm inhibition algorithms were compared. In consideration of larger community amount and key nodes amount in large-scale network, in this experiment different proportions of immunization nodes were selected for such comparison. The experiment results were shown in Figure 6.

Seen from Figure 5, only a certain proportion of key nodes were required for immunization to acquire a better worm inhibition effect. For instance, the adoption of max strategy only required about 40% of nodes to guarantee the final proportion of nodes infected were not more than 20%. The final proportion of infected nodes would not be more than 25% even when only 20% of key nodes were selected. Therefore, the validity of algorithm in this paper was proved.

According to the experiments above, we can find that the strategy used in the article has less suppression effective than Livshit's method, the main cause is that Q Modularity value of community structure found by LCC is slightly lower, which indicates it is very important to improve accuracy for Community Detection, in order to get better worm inhibition results, and this part will be our key research in the next step.

5 Conclusions

At first, adopting improved Link Partition Density Function, this paper makes Community Detection. Then we propose three different strategies to choose key nodes, and give these key nodes immunization to get better worm inhibition results. Finally, to verify the validity of mentioned algorithm, we perform it on a lot of real network data sets.

Acknowledgment

This study was supported by the National Science Foundation of China under Contracts 51477001. The authors would like to thank the Associate Editor and the Reviewers for their valuable comments and suggestions.

References

[1] Y. Y. Ahn, J. P. Bagrow, S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[2] P. Brodka, T. Filipowski, P. Kazienko, "An introduction to community detection in multi-layered social network," *Informatio Systems, E-learning, and Knowledge Management Research*, pp. 185–190, 2013.

[3] Y. Cao, V. Yegneswaran, P. Porras, Y. Chen, "PathCutter: Severing the self-propagation path of XSS JavaScript worms in web social networks," in *Proceedings of the 19th Network and Distributed System Security Symposium (NDSS'12)*, San Diego, USA, 2012.

[4] A. Chaudhary, V. N. Tiwari, A. Kumar, "A new intrusion detection system based on soft computing techniques using neuro-fuzzy classifier for packet dropping attack in MANETs," *International Journal of Network Security*, vol. 18, no. 3, pp. 514–522, 2016.

[5] A. Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, 026132, 2005.

[6] Z. Dong, P. Yi, "A community detection algorithm for dynamic networks using link clustering," *Journal of Xian Jiaotong University*, vol. 48, no. 8, pp. 73–79, 2014.

[7] M. R. Faghani, H. Sandi, "Social networks' XSS worms," in *Proceedings of the International Conference on Computational Science and Engineering*, pp. 1137–1141, 2009.

[8] M. Girvan, M. E. J. Newman, "Community structure in social and biological networks," *National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[9] L. He, D. G. Feng, P. R. Su, et al., "Parallel community detection based worm containment in online social network," *Chinese Journal of Computers*, vol. 38, no. 4, pp. 846–857, 2015.

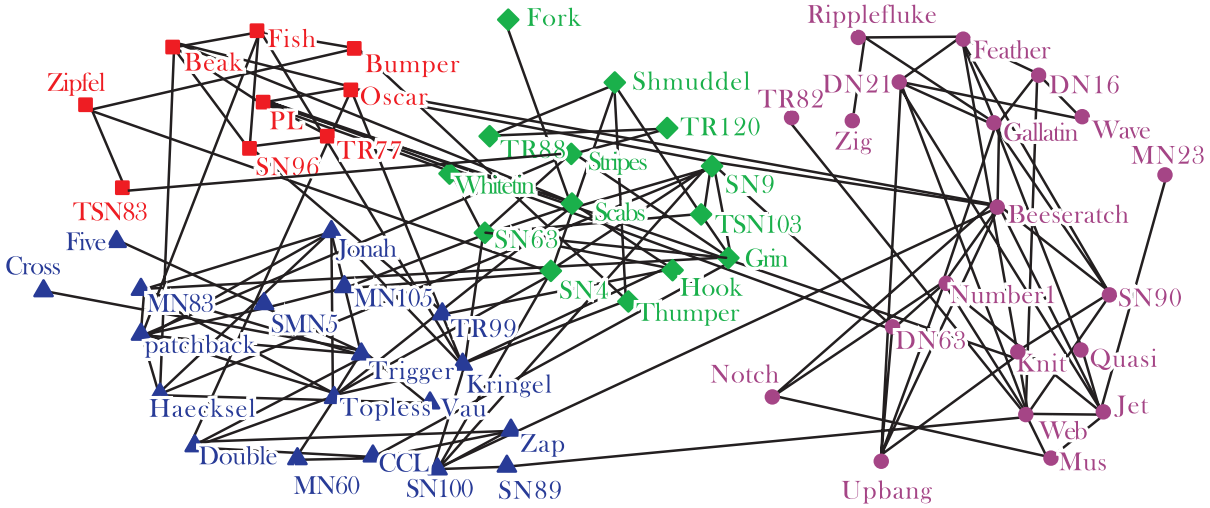

Figure 4: Detection results of LCC against Dolphin

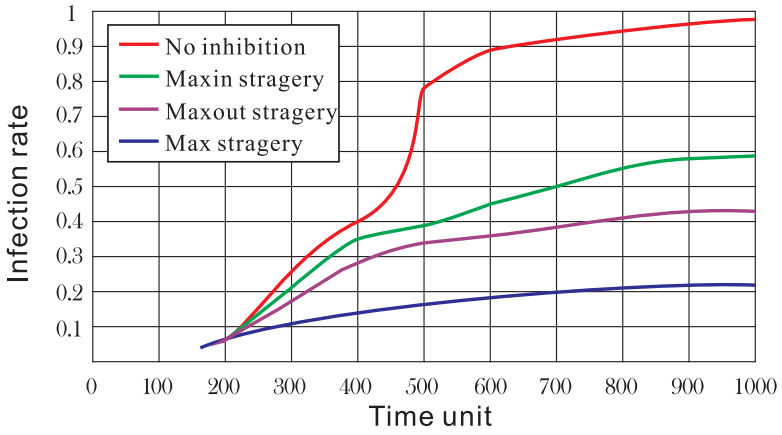

Figure 5: The results of worms inhibition by different strategies

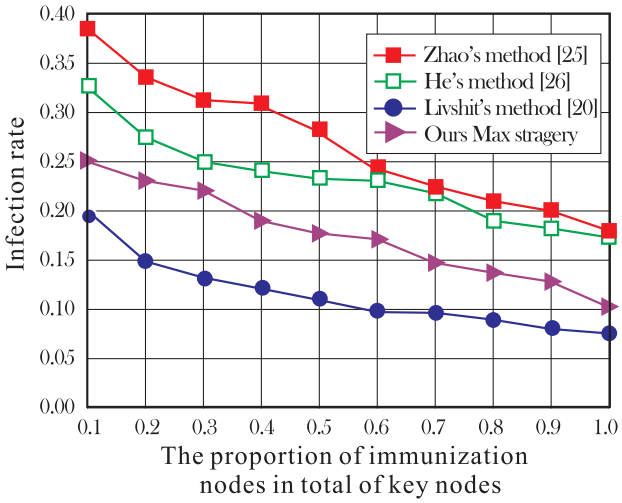

Figure 6: Comparison results of worms inhibition between our method and other algorithms

[10] M. S. Hwang, C. C. Lee, S. K. Chong, J. W. Lo, "A key management for wireless communications," *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 8, pp. 2045–2056, 2008.

[11] M. S. Hwang, C. C. Lee, Y. C. Lai, "Traceability on RSA-based partially signature with low computation," *Applied Mathematics and Computation*, vol. 145, no. 2-3, pp. 465-468, Dec. 2003.

[12] M. S. Hwang, C. C. Yang, S. F. Tzeng, "Improved digital signature scheme based on factoring and discrete logarithms," *Journal of Discrete Mathematical Sciences & Cryptography*, vol. 5, no. 2, pp. 151–155, Aug. 2002.

[13] D. Kuang, C. Ding, H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of 2012 SIAM International Conference on Data Mining*, pp. 106–117, 2012.

[14] M. Kumar, K. Dutta, I. Chopra, "Impact of wormhole attack on data aggregation in hieracrchical WSN," *International Journal of Electronics and Information Engineering*, vol. 1, no. 2, pp. 70–77, 2014.

[15] W. T. Li, T. H. Feng, M. S. Hwang, "An intrusion detection technique based on continuous binary communication channels," *International Journal of Network Security*, vol. 16, no. 5, pp. 323–330, 2014.

[16] B. Livshit, W. Cui, "Spectator: Detection and containment of JavaScript worms," in *Proceedings of the USENIX Annual Technical Conference on Annual Technica1 Conference*, pp. 335–348, 2008.

[17] Z. G. Luo, X. Z. Jiang, "New progress on community detection in complex networks," *Journal of National University of Defense Technology)*, vol. 33, no. 1, pp. 47–52, 2011.

[18] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *American Physical Society*, vol. 69, no. 6, pp. 187–206, 2004.

[19] M. E. J. Newman, M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, 2004.

[20] N. P. Nguyen, T. N. Dinh, S. Tokala, "Overlapping communities in dynamic networks: Their detection and mobile applications," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, pp. 85–96, 2011.

[21] N. P. Nguyen, Y. Xuan, M. T. Thai, "A novel method for worm containment on dynamic social networks," in *Proceedings of the 2010 Military Communications Conference (MILCOM'10)*, pp. 2810–2815, 2010.

[22] G. Palla, I. Derenyi, I. Farkas, et a1., "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[23] U. Raghavan, R. Albert, S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, 036106-1, 2007.

[24] M. Shiga, I. Takigawa, H. Mamitsuka, "A spectral clustering approach to optimally combining numerical vectors with a modular network," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 647–656, 2007.

[25] J. Singh, "Cloud based technique for Blog search optimization," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 32–39, 2016.

[26] F. Sun, L. Xu, Z. Su, "Client-side detection of XSS worms by monitoring payload propagation," in *Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS'09)*, pp. 539–554, 2009.

[27] X. Sun, Y. Y. Liu, J. Q. Zhu, et al., "Research on simulation and modeling of social network worm propagation," *Chinese Journal of Computers*, vol. 34, no. 7, pp. 1252–1260, 2011.

[28] A. L. Traud, P. J. Mucha, M. A. Porter, "Social structure of Facebook networks," *Physical A: Statistical Mechanics and Its Applications*, vol. 391, no. 16, pp. 4165–4180, 2012.

[29] L. Wang, X. Q. Cheng, "Dynamic community in online social networks," *Chinese Journal of Computers*, vol. 38, no. 2, pp. 219–237, 2015.

[30] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[31] W. Xu, F. Zhang, S. Zhu, "Toward worm detection in online social networks," in *Proceedings of 26th Annual Computer Security Applications Conference*, PP.11–20, 2010.

[32] S. Yang, H. Jin, X. Liao, et al., "Modeling modern social-network-based epidemics: A case study of rose," in *International Conference on Autonomic and Trusted Computing*, PP.302–315, 2008.

[33] X. Z. Zhang, Y. Y. Pu, L. Yang, B. Wang, "Community discovery of large-scale web service network," *Journal of Chinese Computer Systems*, vol. 36, no. 5, pp. 1017–1020, 2015.

[34] Y. Zhao, P. K. Yi, "A dynamic worm propagation model based on social network," *Computer Engineering and Science*, vol. 35, no. 12, pp. 34–38, 2013.

[35] C. C. Zhou, W. Gong, D. Towsley, "Code red worm propagation modeling and analysis," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 138–147, 2002.

[36] Z. Zhu, G. Cao, S. Zhu, et a1., "A social network based patching scheme for worm containment in cellular networks," in *Proceedings of the IEEE INFOCOM*, pp. 1476–1484, 2009.

Biography

Yibing Wang is currently a lecturer at Anhui University. She chaired or participated the national, provincial and

municipal scientific research projects over 20 itemsand published more than 10 journal papers. what is moreas the first inventor, she obtained 2 items national patent. Her research interests include machine learning, pattern recognition and network security.

Jie Fang received her Doctor Degree in Computer Science from Chinese Academy of Sciences. She has published more than 30 papers in international conferences and journals (SCI or EI journals). She is currently a faculty member in the college of Computer and Information Engineering at Anhui University. Her research interests include network security and Artificial intelligence.

Fuhu Wu is currently a Doctoral student at Anhui University. Her research interests include network protocols and security, enterprise systems, etc.