An Improved Community Detection Method in Massive Social Networks

Yong Yao, Bian Li, Lei Peng and Zhijing Liu

School of Computer Science, Xidian University, Xian, China, 710071

yaoyong@xidian.edu.cn

Keywords: community detection, the recursive shingling algorithm, ELDN.

Abstract. Community detection is one of the most important tools to analyze social network. The paper studies a community detection algorithm to find a dense community in massive social networks, the recursive shingling algorithm. Then an improved method based on the recursive shingling algorithm is proposed as ELDN, and we have proved that the improved algorithm is better than the recursive shingling algorithm through experiment.

Introduction

Community detection is a core problem in social network analysis [1] and it can be applied in many real world. Community detection algorithms are fundamental tools that allow us to uncover organizational principles in networks [2].

Many community detection approaches have been proposed in the past. These methods mainly include classical k-means clustering or hierarchical clustering algorithm based on similarity [1], the clustering algorithm based on the idea that cluster centers are characterized by a higher density than their neighbors proposed by Alex Rodriguez [3]. However, some algorithms typically look for very small communities containing on the order of ten nodes in essence [4]. There are few efficient methods in common usage for dense community detection [4]. Thus, D.Gibson et al. [4] present an efficient recursive shingling algorithm for fast community detection of social networks. This recursive shingling algorithm can be efficiently computed even for large-scale social networks. Therefore, the main goal of this paper is to analyze and improve the recursive shingling algorithm.

In view of this, the paper describes some questions needed to address:

(1) Analyze the phenomena and reasons of not too dense community detected by the recursive shingling algorithm.

(2) Propose ELDN method based on the recursive shingling algorithm to mine much denser community.

Background Works

The main goal of this paper is to analyze and improve the recursive shingling algorithm, this section mainly introduces the principle of the recursive shingle algorithm.

Recursive Shingling Algorithm. The recursive shingling algorithm, as the name implies, means that shingle algorithm is recursive. In the process of community detection, shingle algorithm is used to solve the following problem: given a domain U and a subset S of it, generate a constant-size fingerprints such that two subsets A and B may be compared by simply comparing their fingerprints [4]. The basic idea of shingle algorithm [5] is to use a constant number of shingles that fingerprints are generated above, to represent the surrounding environment of a node, that is to say, the adjacent nodes. The recursive shingling algorithm based on the shingle algorithm proceeds as the following steps:

(1) Find out all shingles sets of all nodes.

(2) For each shingle s_i, calculate the node set V_i containing the shingle s_i.

(3) Take V_i as the adjacency nodes set of s_i, and put shingle algorithm into use again.

After applying the above recursive shingling algorithm, we obtain second-level shingles, and then we cluster first-level shingles effectively using second-level shingles. It can be seen from the description of the recursive shingling algorithm, only if the first-level shingles have the same

second-level shingles, are they related. Regard first-level shingles as nodes in graph, only if the first-level shingles have the same second-level shingles, is there an edge between them. At the same time, the clustering of the first-level shingles is equivalent to finding connected components in graph efficiently. By using the Union-Find algorithm to find the connected components, namely clustering results, the clustering results, in return, correspond to the original graph as required.

Set adjacent nodes of node a_j as{ $a_1,\ldots,a_n$ },the algorithm for the shingles of a_j proceeds as follows:

Algorithm Shingle($a_1,\ldots,a_n,s,c$)
Let H be a hash function from strings to integers
Let p be a large random prime(say,32 bits)
Let $a_1,b_1,\ldots,a_c,b_c$ be random integers in [1…p]
For i=1 to n do x_i=H("a_i")
For j=1 to c do
For i=1 to n do $y_i=(a_j*x_i+b_j) \bmod p$
Let $y_1',\ldots,y_s'$ be s minimal elements of y
Let z_j =H ("$y_1' \circ \ldots \circ y_s'$")
Output $z_1,\ldots,z_c$

Set adjacent nodes of node v_i as $\Gamma(v_i)$,then the second-level shingles algorithm is shown as follows:

Algorithm Shingle2($v_1,\ldots,v_n,s_1,c_1,s_2,c_2$)
For i=1 to n do
$S_1(v_i)$=Shingle($\Gamma(v_i)$, c_1,s_1)
Let $S=U^n_{i=1} S_1(v_i)$
For sϵS do
Let $\Gamma(s)=\{v|S_1(v) \ni s\}$
$S_2(s)$=Shingle($\Gamma(s),c_2,s_2$)
Let $T= U_{s\epsilon S} S_2(s)$
For tϵT do
Let $\Gamma(t)=\{ s\epsilon S|S_2(s) \ni t \}$
Output <t,Γ(t)>

Improvement of the Recursive Shingling Algorithm

Analysis of the Result of the Recursive Shingling Algorithm.

Result of the Recursive Shingling Algorithm. With parameters set to s_1=2, c_1=10, s_2=2, c_2=10, we apply the recursive shingling algorithm on the Douban experimental data and the final result shows that the denser community has 314 nodes. In order to visualize the final result, we identify the edges between these nodes according to the raw Douban experimental data and import the edges into Gephi, finally, the experimental result is described as Fig. 1. As can be seen from Fig. 1, community structure detected by the recursive shingling algorithm is not much denser.

Fig.1: Visualization of community structure, after the recursive shingling algorithm applies on Douban experimental data

We get the number of nodes and the number of edges of the community in Gephi discovered by the recursive shingling algorithm, as shown in Fig. 2. However, it can only auto-complete the

information of nodes which are linked to other nodes in the community so that the isolated nodes are not showed in Gephi. Obviously, the number of remaining nodes, namely isolated nodes, is 314-239=75, so community structure detected by the recursive shingling algorithm is not much denser. Next, we will analyze the reasons in detail for this phenomenon.

Fig. 2: Nodes and edges of the community discovered by the recursive shingling algorithm

Cause Analysis of Above-mentioned Phenomena. According to principle of the recursive shingling algorithm, the algorithm is based on nodes similarity for community detection. The metric of node similarity is based on environment nodes (adjacent nodes). If the similarity of environment nodes set of two nodes reaches the threshold of the parameter s_2, which means the two nodes have high similarity, the algorithm just puts the two nodes into the same community. While environment nodes determining the similarity of two nodes would not be put into the same community, the algorithm completely ignores the contribution of environment nodes to this community(Generally speaking, in a community, the more edges the node is adjacent to, the greater contribution the node makes to the community). As environment nodes determining the similarity of two nodes would not be put into the same community, there will be such a situation where the contribution of two nodes to the community may be less than the contribution of the other nodes to the community. In this way, it is possible that community detected by the recursive shingling algorithm has some isolated nodes or low-degree nodes, which lead to not much denser community.

Proposed ELDN Method. Based on the reasons above mentioned, this paper presents ELDN (eliminate low-degree nodes) method selectively makes the density of the community increase as much as possible.

For a directed graph G, its density formula is defined as:

$$d = l \Big/ n(n-1) \tag{1}$$

Here, d is the density of the directed graph G, l represents the number of the edges and n denotes the number of the vertexes in the directed graph G.

C(i) is the nodes set of the i_{th} community, and c_i is one element of the C(i). For the node c_i, if removing it from the community can increase the density of the community, then eliminate it from the community.

Let k denotes that the community C(i) includes k elements which are directly linked to c_i. That is to say, if the node c_i is removed from the community C(i), and that will make the number of the edges in the community C(i) decrease k. Suppose the community C(i) has c_i nodes and l_i edges, the density formula of the community c_i is denoted as:

$$d_i = l_i \Big/ n_i(n_i-1) \tag{2}$$

After we remove node c_i from the community C(i), the density formula of the community C(i) can be formulated as:

$$d'_i = (l_i - k) \Big/ (n_i-1)(n_i-2) \tag{3}$$

After we remove node c_i to the community C(i), if the density of the community C(i) has increased, The d_i and d_i' should satisfy the following condition:

$$d'_i > d \tag{4}$$

Substituting the Eq. (2) and the Eq. (3) into Eq. (4), then we have the following inequation:

$$(l_i - k) \Big/ (n_i - 1)(n_i - 2) > l_i \Big/ n_i(n_i - 1) \tag{5}$$

After simplification:

$$k < 2l_i \Big/ n_i \tag{6}$$

The results can be obtained by the above calculation, for any environment node c_i in the community C(i), with k nodes adjacent to, if k satisfies the condition Eq.6, we can say that remove node c_i can increases the density of the community C(i), so it can be removed from the community C(i).

Still using the above notation for describing, the ELDN method is shown as follows:

Algorithm ELDN_Shingle(C(i),k,Adj)

Let Dic be an dictionary, key is node and value is number of edges between the key and C(i).

```
For jєC(i) do
    For iєadj(j) do
        if iєC(i)
            Let Dic[j]=Dic[j]+1
    For jєC(i) do
            if Dic[j]<k
            eliminate j from C(i)
    Output C(i)
```

Experimental Evaluation of ELDN Method

Evaluation of Graph Density. First, upon substituting the number of nodes and the number of edges into Eq.6, we obtain the minimum value of K:

$$k < 2l \Big/ n = 2 \times 468 \Big/ 314 \approx 2.98 \tag{7}$$

With parameters set to k = 3, s_1 = 2, c_1 = 10, s_2 = 2, c_2 = 10, we apply the ELDN method on the experimental data set and the final result shows that the denser community has 96 nodes. Through visualizing community structure in Gephi, we get the Fig. 3 and Fig. 4.

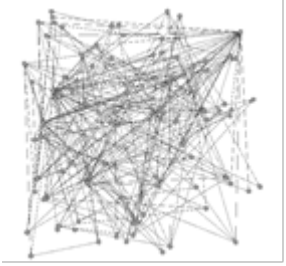

Fig. 3: Visualization of community structure, after the ELDN method applies on Douban experimental data

Fig. 4: Nodes and edges of the community discovered by the ELDN method

According to Fig. 2 and Fig. 4, we can respectively calculate the density of the community:

$$d_o = l_o \Big/ n_o(n_o - 1) = 468 \Big/ (314 \times 313) \approx 0.00476 \tag{8}$$

$$d_e = l_e \Big/ n_e(n_e - 1) = 283 \Big/ (96 \times 95) \approx 0.03103 \tag{9}$$

As can be seen above, the density of the community in Fig. 1 is significantly less than that in Fig. 3 which is shown in the Table 1 below:

Table 1. The community density of the recursive shingling algorithm and the ELDN method

type	The recursive shingling algorithm	The ELDN method
nodes	314	96
edges	468	283
density	0.00476	0.03103

Evaluation of Community Strength. Although there is no strict definition of the community, it is generally considered that the closer the interaction in community and the sparser the interaction between the community and the others, the better the community [6]. That is to say, the larger the ratio of the number of edges in the community to the number of edges between the community and the others, the better the community. Therefore, in this paper, the strength of community F is defined as the ratio of the number of edges in the community to the number of edges between the community and the others. Then the community is evaluated by the strength of community F, the larger the ratio F, the better the community and vice versa.

A social network graph G = (V, E) consists of a set V of nodes and a set E of edges. Set $v_j \epsilon V$ and put v_j into the community C_i after community detection. The outdegree of v_j is the number of nodes v_i such that $(v_j,v_i) \epsilon E$ and $v_i \epsilon C_i$, v_i is an outlink of v_j and the set of outlinks of v_j is denoted outdegreeC(v_j). In the same way, the set of outlinks of v_j is denoted outdegreeG(v_j) such that $(v_j,v_i) \epsilon E$ and $v_i \epsilon G$; and the set of outlinks of v_j is denoted outdegreeO(v_j) such that $(v_j,v_i) \epsilon E$, $v_i \epsilon G$ and $v_i \notin C_i$. Therefore, the relationship among them is:

$$outdegreeO(v_i) = outdegreeG(v_i) - outdegreeC(v_i) \tag{10}$$

So, for all nodes in the community C(i), the strength of community F can be shown:

$$F = \sum_{j \in Ci} \text{outdegreeC}(v_j) \Big/ \sum_{j \in Ci} outdegreeO(v_j) \tag{11}$$

Next, the paper analyzes the recursive shingling algorithm and the ELDN method, as shown in Fig. 2 and Fig. 4. Then calculate the strength of community F of the recursive shingling algorithm and the ELDN method respectively, the results are shown in Fig. 5 and Fig. 6.

```
penglei@penglei:~/experiment$ python orignal_strong_weak.py
22369
F=outdegreeC/outdegreeO=0.0209218114355
```

Fig. 5: The strength of community F of the recursive shingling algorithm

```
penglei@penglei:~/experiment$ python eldn_strong_weak.py
96
F=outdegreeC/outdegreeO=0.0274410937652
```
Figure 6: The strength of community F of the ELDN method

As can be seen above, the community strength in Fig. 5 is significantly less than that in Fig. 6, which is shown in the Table 2 below:

Table 2. The community strength of the recursive shingling algorithm and the ELDN method

type	The recursive shingling algorithm	The ELDN method
outdegreeC	468	283
outdegreeO	22369	10313
F	0.02092	0.02744

In conclusion, through comparing Table 1 and Table 2, it is clearly observed that the community by the ELDN method is much denser than that detected by the recursive shingling algorithm.

Conclusions and Future Work

In this paper, we have mainly analyzed and improved the recursive shingling algorithm carried out by David Gibson. Through theoretical analysis and experimental verification, it demonstrates the "low intensity" phenomenon in the community detected by the recursive shingling algorithm. Then ELDN method based on the recursive shingling algorithm is presented, and we identify that the improvement algorithm is better than the recursive shingling algorithm, through the community density and the strength of community experiments.

For the problems, we have proposed them in the section 1, have basically solved, however, the ELDN method still has some shortages, it should be improved in many aspects in future.

ACKNOWLEDGEMENT

The research was supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2012JM8040), the Fundamental Research Funds for the Central Universities and National Natural Science Foundation of China (NO. 61202177).

References

[1] Lei Tang and Huan Liu, in: Community Detection and Mining in Social Media, edtied by Lise Getoor, Morgan and Claypool Publishers(2010).

[2] Jaewon Yang, McAuley, J. and Leskovec, J. in: Community Detection in Networks with Node Attributes, ICDM , 2013:p. 1151-1156.

[3] Alex Rodriguez and Alessandro Laio, Clustering by fast search and find of density peaks. Science, 344:1492-1496, 2014.

[4] D.Gibson, R.Kumar and A.Tomkins: Discovering large dense subgraphs in massive Graphs. Proceedings of the 31st international conference on Very large data bases, 2005, 721 -732.

[5] A. Z. Broder, S. Glassman, M. Manasse and G.Zweig: Syntactic clustering of the web. Palo Alto: Digital Equipment Corpo ration, 1997,7.

[6] Xiangjie Yin, in: Research on Community Detection and Node Evaluation Algorithm in Social Networks. Jilin University, 2014, 10-11.