An Algorithm for Detecting Community Structure of Complex Networks based on Clustering

Sheng Bin, Gengxin Sun

Abstract

There are considerable interest in algorithms for detecting community structure, which is fundamental for analyzing the relationship between structure and function in complex networks. In this paper, after the introduction of some traditional approaches for detecting community structure and data mining clustering algorithms, we propose Mapping Vertex into Vector (MVV) algorithm, which can convert network nodes to vector, based on the algorithm, we propose K-means algorithm for detecting community structure based on MVV, and use fuzzy c-means to deal with overlapping community. Finally, experiments show that the algorithm presented in this paper is of high accuracy with good performance.

Keywords: Complex Network, Community Structure, Data Mining, Clustering, Similarity Measure

1. Introduction

There have been numerous and various complex networks with the development of science, technology and human society, such as biochemical network [1], scientific collaboration network [2], World Wide Web [3], etc. Most social, biological, and technological networks display substantial non-trivial topological features, with patterns of connection between their elements that are neither purely regular nor purely random. Such features include a heavy tail in the degree distribution, a high clustering coefficient, assortativity or disassortativity among vertices, community structure, and hierarchical structure. An important property that seems to be common to many complex networks is community structure [4]. Community detection, which can help us simplify functional analysis, is potentially very useful.

Complex networks consist of a set of nodes, joined in pairs by edges indicating interaction. Many actual networks are inhomogeneous, consisting not of an undifferentiated mass of vertices, but of distinct groups. Within these groups there are many edges between vertices, but between groups there are fewer edges, Fig. 1 illustrates a network that can be divided into groups.

Figure 1. A network with community structure

The ability to find communities within large networks in some automated fashion could be of considerable use. Communities in World Wide Web might correspond to sets of web sites dealing with related topics [3,5], while communities in a biochemical network or an electronic circuit might correspond to functional units of some kind [6-8]. In this paper we discuss computer algorithms for the extraction of communities from raw network data.

The outline of the paper is as follows, firstly, we describe some of the basic concepts and models of complex network. Secondly we describe some traditional approaches for detecting community structure. Thirdly we propose and describe our algorithm for detecting community structure. At last we give our conclusions through experiments result.
2. Traditional approaches for detecting community structure

Although community structure has attracted a great deal of interest, there is a common concept of community. Qualitatively, a community is defined as a subset of nodes within the graph such that connections between the nodes are denser than connections with the rest of the network. As mentioned above, a community is generally thought of as a part of a network where internal connections are denser than external ones. To explain the use of detection algorithms a more precise definition is needed. Here we give a definition of community structure which translates the sentence above into formulas.

The basic quantity to consider is $k_i$, the degree of a generic node $i$, which in terms of the adjacency matrix $A_{i,j}$ of the network $G$ is $k_i = \sum_j A_{i,j}$. (The adjacency matrix fully specifies the topology of the network. In the simplest case of an unweighted, undirected network, it is equal to 1 if $i$ and $j$ are directly connected; it is equal to zero otherwise.) If we consider a subgraph $G \subset G'$, to which node $i$ belongs, we can split the total degree in two contributions:

$$k_i^{\text{in}}(G') = \sum_{j \in G'} w_{ij}$$

Equation (1) is the number of edges connecting node $i$ to other nodes belonging to $V$.

$$k_i^{\text{out}}(G') = \sum_{j \notin G'} w_{ij}$$

Equation (2) is clearly the number of connections toward nodes in the rest of the network.

Based on this, Radicchi proposed two plausible definitions of community.

Definition of Community in a Strong Sense. The subgraph $G'$ is a community in a strong sense if

$$k_i^{\text{in}}(G') > k_i^{\text{out}}(G'), \forall i \in G$$

In a strong community each node has more connections within the community than with the rest of the graph.

Definition of Community in a Weak Sense. The subgraph $G'$ is a community in a weak sense if

$$\sum_{i \in G} k_i^{\text{in}}(G') > \sum_{i \in G} k_i^{\text{out}}(G')$$

In a weak community the sum of all degrees within $G'$ is larger than the sum of all degrees toward the rest of the network.

Clearly a community in a strong sense is also a community in a weak sense, whereas the converse is not true.

It is worth mentioning that our definitions of community, although very natural, do not represent the only possible choice. Several other possible definitions, possibly more appropriate in some cases.

The problem of detecting such communities within networks has been well studied recently. Newman divided approaches for detecting community structure into traditional and recent approaches.

Traditional approaches include computer science methods and sociological methods. The computer science methods derive from graph partitioning, such as the spectral bisection method, which is based on the eigenvectors of the graph Laplacian [9], or the Kernighan-Lin algorithm, which is based on greedy optimization for the cost function [10]. Sociological methods are based on the idea of hierarchical clustering and make use of dendrograms to detect the community structure. Generally, the computing time to detect these structures is very high.
Early approaches work well for specific types of problems, but perform poorly in more general cases. To combat this problem a number of new algorithms have been proposed in recent years. Recent approaches to detecting community structure include the Newman-Girvan algorithm, which uses edge betweenness as a metric to identify the boundaries of communities [4]; the Radicchi algorithm, which is based on counting short loops of edges [11]; and the W-H algorithm, which is inspired by the properties of resistor networks [12]. Because it is widely used and has been incorporated into some network analysis programs, the Newman-Girvan algorithm is very popular [13, 14].

3.1. Spectral bisection

The Laplacian of an n-vertex undirected graph $G$ is the $n \times n$ symmetric matrix $L$ whose diagonal element $L_{ii}$ is the degree of vertex $i$, and whose off-diagonal element $L_{ij}$ is -1 if vertices $i$ and $j$ are connected by an edge and zero otherwise. Those can be formulized as,

$$L = D - W$$

Where $D$ is the diagonal matrix of vertex degrees and $W$ is the adjacency matrix. Since the degree $D_{ii} = \sum_{j} W_{ij}$, it follows that all rows and columns of the Laplacian sum to zero, and hence that the vector $l = (1; 1; 1 \ldots)$ is always an eigenvector with eigenvalue zero.

If the network separates perfectly into communities, i.e., divides into $g$ non-overlapping groups of vertices $G_k (k = 1 \ldots g)$ such that there are only within-community edges and no between-community ones—the groups are components of the graph—then the Laplacian will be block diagonal. Each diagonal block will form the Laplacian of its own component, and will therefore also have an eigenvector $v^{(k)}$ with eigenvalue zero and elements $v^{(k)}_i = 1$ if $i \in G_k$ and 0 otherwise. Thus there will be $g$ degenerate eigenvectors with eigenvalue 0.

If the network separates well but not perfectly into communities, that is to say, if there are just a few edges that do not fit the block-diagonal pattern, then this will no longer be perfectly true. Instead there will in general be the one eigenvector 1 with eigenvalue zero, and $g$-1 eigenvalues slightly different from zero, indeed slightly greater than zero, since all eigenvalues of the graph Laplacian are non-negative. The corresponding eigenvectors will approximately be linear combinations of the eigenvectors $v^{(k)}$ defined above. Hence, by looking for eigenvalues of the graph Laplacian only slightly greater than zero and taking linear combinations of the corresponding eigenvectors, one should in theory be able to find the blocks themselves, at least approximately.

A particular special case of this argument is when there are only two blocks. Noting that all eigenvectors corresponding to non-degenerate eigenvalues of a real symmetric matrix are orthogonal, it is clear that all eigenvectors other than that corresponding to the lowest eigenvalue must have both positive and negative elements. And for the case of two weakly coupled communities there will be one eigenvector with eigenvalue slightly greater than zero and elements all positive for one community and all negative for the other, since all elements are nearly equal within a community. Thus, the network can be divided into two communities by looking at the eigenvector corresponding to the second lowest eigenvalue and separating the vertices by whether the corresponding element in this eigenvector is greater than or less than zero. This is the spectral bisection method.

3.2. Kernighan-Lin algorithm

The Kernighan-Lin algorithm is a greedy optimization method that assigns a benefit function $Q$ to divisions of the network and then attempts to optimize that benefit over possible divisions. The benefit function is the number of edges that lie within the two groups minus the number that lie between them. The algorithm requires the user to specify the size of the two groups into which the network should be split and to choose a starting configuration for the groups.

The algorithm then has two stages. First, we consider all possible pairs of vertices in which one vertex is chosen from each of the groups, and calculate the change $\Delta Q$ in the benefit function that would result from swapping them. Then we choose the pair that maximizes this change and perform
the swap. This process is repeated, with the restriction that any vertex that has previously been swapped is never swapped again. When all vertices in one of the groups have been swapped once, this stage of the algorithm ends.

In the second stage, we go back over the sequence of swaps that were made and find the point during this sequence at which $Q$ was highest. This is taken to be the bisection of the graph.

This two-stage process allows for the possibility that the value of $Q$ does not increase monotonically. Even if $Q$ decreases, a higher value that occurs later in the sequence of swaps will still be found by the algorithm.

The principal disadvantage of the Kernighan-Lin algorithm is that we have to specify the sizes of the two communities before starting. Even if this shortcoming could be overcome, the Kernighan-Lin algorithm still suffers from the drawback of all bisection algorithms, as mentioned above for the spectral method: it only divides the network into two groups and not an arbitrary number.

4. Algorithms for detecting community structure based on clustering

The clustering analysis has been developed quickly and is an extraordinary important technology in the data mining and exploration analysis, especially is widely used in the fields of data mining, statistics, machine study, space database technology, biology and marketing analysis etc. Community structure is also called clustering in complex network, so we can use approaches and theories of data mining for detecting community structure of complex network, but there is extreme difference between data structure of complex network and data structure which is processed by clustering of data mining, at the same time, similarity measure between vertices, to a great extent, impacts the results of clustering algorithm. If network nodes can be converted to adapted data structure of data mining clustering algorithm and select adapted similarity measure, a mass of mature data mining clustering algorithm can be used to detecting community structure. Therefore, before using data mining clustering algorithm to detecting community structure, we should choose adapted similarity measure to convert network nodes to data structure of data mining.

4.1. Mapping vertex into vector algorithm

There are two kinds of common similarity measure: Euclidean distance and cosine similarity.

Euclidean distance is a measure of structural equivalence developed by Burt [15]. For an undirected network, the Euclidean distance between two vertices is written as,

$$
dist(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} 
$$

(6)

Euclidean distance is actually a measure of dissimilarity, as $x_{ij} = 0$ for vertices that have a perfectly similar local structure and increases as the local structure becomes less similar.

The measure cosine similarity was proposed by Salton in 1983 [16]. Originally a method for measuring the similarity between any two vector quantities, it was reshaped as a network measure and for undirected networks is written as,

$$
r(X_i, X_j) = \frac{\sum_{k=1}^{n} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{n} x_{ik}^2} \sqrt{\sum_{k=1}^{n} x_{jk}^2}} 
$$

(7)

In this paper, a algorithm for converting network nodes to vector is proposed, we can use Euclidean distance, cosine similarity to measure similarity degree between nodes after converting.

Probability spreading matrix which is used to define the way of transfer information between nodes is defined, a $n \times n$ probability spreading matrix $\hat{T}$ is defined, element $\hat{T}_{ij}$ of $\hat{T}$ denote the proportion of
information, which node j spreads to node i every time, in node i received information, probability spreading matrix is defined as,

$$\hat{T}_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{N} w_{ik}}$$

(8)

where $N$ is quantity of nodes in network, $w_{ij}$ is the element of adjacency matrix.

Matrix symbol of probability spreading matrix is denoted as,

$$\hat{T} = D^{-1}W$$

(9)

where $D$ is diagonal matrix which consist of degree of node, $W$ is adjacency matrix of network.

Mapping Vertex into Vector (MMV) algorithm is composed of the following steps:

- **Step 1:** Calculate adjacency matrix $W$;
- **Step 2:** Calculate diagonal matrix $D$;
- **Step 3:** Calculate $\hat{T} = D^{-1}W$;
- **Step 4:** $X_{s+1} = \hat{T}X_{s}$, $X_{1} = 1$
- **Step 5:** If times of spread $t$ is less than $T$, go to Step 4, else output result $X_{s}$.

Based on mapping vertex into matrix algorithm, we can use clustering approaches of data mining to detect community structure of complex network.

### 4.2. K-means algorithm for detecting community structure based on MMV algorithm

The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume $k$ clusters) fixed a priori. The main idea is to define $k$ centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to recalculate $k$ new centroids as barycenters of the clusters resulting from the previous step. After we have these $k$ new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the $k$ centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x^{(i)} - c_j||^2$$

(10)

Where $x_{i}$ is a data point, $c_{j}$ is the cluster centre, (18) is an indicator of the distance of the $n$ data points from their respective cluster centres.

The algorithm is composed of the following steps:

- **Step 1:** Place $K$ points into the space represented by the objects that are being clustered. These points represent initial group centroids;
- **Step 2:** Assign each object to the group that has the closest centroid;
- **Step 3:** When all objects have been assigned, recalculate the positions of the $K$ centroids;
- **Step 4:** Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

The algorithm is significantly sensitive to the initial randomly selected cluster centres. Based on Mapping Vertex into Vector Algorithm, when using K-means algorithm to detect community structure, we use following algorithm to select initial cluster centres.
Supposing that the network does split into two communities.
Step 1: Find node s whose degree is max;
Step 2: Use MVV algorithm for mapping node s to a vector;
Step 3: Find node t which is affected most slightly by node s from vector;
Step 4: Node s and node t act as initial cluster centres.

Although K-means algorithm can be used to detect community structure of complex network if the community structure is such that it can be interpreted in terms of separated sets of communities, most real networks are characterized by well defined statistics of overlapping and nested communities. For such networks, we can use fuzzy c-means algorithm to detect community structure.

Fuzzy c-means is a method of clustering which allows one piece of data to belong to two or more clusters. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - c_j\|^2$$

(11)

where $m$ is any real number greater than 1, $u_{ij}$ is the degree of membership of $x_i$ in the cluster $j$, $x_i$ is the $i$th of d-dimensional measured data, $c_j$ is the d-dimension center of the cluster, and $\|*\|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership $u_{ij}$ and the cluster centers $c_j$ by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\|x_i - c_k\|^2}{\|x_i - c_k\|^2} \right)^m}$$

(12)

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}$$

(13)

This iteration will stop when $\max_{i,j} \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| < \epsilon$, where $\epsilon$ is a termination criterion between 0 and 1, whereas $k$ are the iteration steps. This procedure converges to a local minimum or a saddle point of $J_m$.

The algorithm is composed of the following steps:
Step 1: Initialize $U=\{u_{ij}\}$ matrix, $U^{(0)}$;
Step 2: At k-step: calculate the centers vectors $C^{(k)}=\{c_j\}$ with $U^{(k)}$ through (21);
Step 3: Update $U^{(k)}$, $U^{(k+1)}$ through (20);
Step 4: If $\|U^{(k+1)} - U^{(k)}\| < \epsilon$ then STOP; otherwise return to step 2.

5. Experiment and results

In order to prove validity of our algorithm, and compare the ability of different algorithms to find communities in network, Zachary Karate Club network is used to testing those algorithms.

Zachary Karate Club is a social network of friendships between 34 members of a karate club at a US university in the 1970s [17]. The Karate Club network is a useful test case for community-detecting algorithms because we expect any calculated communities to be very similar to the actual group memberships.
Zachary Karate Club in which an internal dispute led to the schism of a karate club into the formation of two smaller clubs [17]. A plot showing the connections between members of both clubs in real world is shown in Fig. 4.

Figure 4. Plot of the members of Zachary's Karate Club network

A plot showing the communities structure of Zachary Karate Club network detected by hierarchical clustering algorithm is shown in Fig. 5.

Figure 5. Plot of communities structure detected by hierarchical clustering algorithm

From Fig. 5, we can see, a majority of nodes have been measured into corresponding communities, but several nodes, especially, centre node of every community such 1, 33,34 have not been measured into any community.

A plot showing the communities structure of Zachary Karate Club network detected by K-means algorithm based on MVV algorithm is shown in Fig. 6.

Figure 6. Plot of communities structure detected by K-means algorithm based on MVV algorithm

From Fig. 6, only node 3 is measured incorrectly, the algorithm is better than hierarchical clustering algorithm in detecting community structure.

A plot showing the communities structure of Zachary Karate Club network detected by fuzzy c-means algorithm based on MVV algorithm is shown in Fig. 7.
An Algorithm for Detecting Community Structure of Complex Networks based on Clustering
Sheng Bin, Gengxin Sun

From Fig. 7, overlapping community is detected, we can aim to the overlapping community for using other algorithm to determine nodes in overlapping community to belong to which communities. After dealing with overlapping community, plot as Fig. 8 shown.

From Fig. 8, community structure detected by our algorithm is accordant completely to real-world network data.

6. Conclusion

In this paper we have reviewed algorithmic methods for detecting communities of densely connected vertices in complex network data. We have discussed some of approaches, such as spectral bisection, Kernighan-Lin algorithm, algorithm of Girvan and Newman and hierarchical clustering, but,
as we have pointed out, these have a number of shortcomings as far as the analysis of large real-world networks is concerned.

We proposed mapping vertex into vector algorithm, which can be used to convert all vertices in network into vectors, based on this algorithm, we discussed briefly two data mining clustering algorithms for detecting community structure of complex network. We also compared results produced by these algorithms and outlined their strengths and weaknesses. As a result of experiment results, it appears that our data mining clustering algorithms which based on MVV algorithm is highly effective at detecting community structure in real-world complex network.

7. References