Physica A 460 (2016) 230–234

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Community detection in networks based on minimum spanning tree and modularity

Bilal Saoud [a,*], Abdelouahab Moussaoui [b]

[a] Department of Informatics, Faculty of Exact Sciences, University of Bejaia, Abderrahmane Mira, 06000, Bejaia, Algeria
[b] Intelligent Systems Laboratory, Faculty of Sciences, University of Ferhat Abbas, 19000, Setif, Algeria

HIGHLIGHTS

- In this paper, we introduce our method for community detection.
- Our method is designed to detect community structure for unweighted and undirected networks.
- We used the minimum spanning tree and nodes dissimilarity to construct communities (create disconnected groups of nodes).
- We used the modularity in the merging process to find the final community structure.
- Our method was tested on both artificial and real networks.

ARTICLE INFO

Article history:
Received 9 September 2015
Received in revised form 20 February 2016
Available online 18 May 2016

Keywords:
Community detection
Networks
Minimum spanning tree
Modularity
Normalized mutual information

ABSTRACT

In this paper we propose a novel splitting and merging method for community detection in which a minimum spanning tree (MST) of dissimilarity between nodes in graph is employed. In the splitting process, edges with high dissimilarity in the MST are removed to construct small disconnected subgroups of nodes from the same community. In the merging process, subgroup pairs are iteratively merged to identify the final community structure maximizing the modularity. The proposed method requires no parameter. We provide a general framework for implementing such a method. Experimental results obtained by applying the method on computer-generated networks and different real-world networks show the effectiveness of the proposed method.

1. Introduction

Almost real complex networks are often structured into groups, in other words have a community structure. However, the nodes in the network formed subgraphs connected between them, where each subgraph is more linked inside than the rest of network; these subgraphs represent communities [1]. Community structure is important because nodes in the same community share common properties or insure similar roles in the network. Community structure detection provides more information towards understanding the network from only its topology. In the following paper we consider a complex network as a graph $G(V, E)$, where V is the set of nodes ($|V| = n$) and E is the set of edges ($|E| = m$).

Many community structure detection methods have been proposed in different topics (social networks, metabolic networks, communication networks, etc.). These methods can be classified according to the type of networks (uniparitie or bipartite, weighted or unweighted) and the community structure (disjoint or overlapping) [1,2]. In this paper we focus on methods for uniparitie, unweighted networks to detect disjoint community structure. Among the most important methods

* Corresponding author.
E-mail address: bilal340@gmail.com (B. Saoud).

http://dx.doi.org/10.1016/j.physa.2016.05.014
0378-4371/

we find, Kernighan–Lin method [3], which uses a bisection algorithm to find the graph cut which minimizes the number of edges between two groups. Its time complexity is $O(n^3)$ in the worst case. The Girvan and Newman method [4] is based on the betweenness centrality (number of shortest paths passing through as edge), which requires a time $O(m^2n)$. Radicchi et al. method [5] is based on clustering coefficient of edges. The method removes the edge of lower coefficient at each step. The total complexity is $O(m^2)$. Fortunato et al. method [6] is a variant of Girvan and Newman method, based on the information centrality with a complexity $O(m^3n)$. Clauset et al. method [7], an improved Newman method [8], is based on the greedy optimization of modularity. This method is quick, with time complexity $O(n^3 \log n)$ in the worst case, but in most real-world cases at $O(n \log(n))$. The method proposed by Donetti and Muñoz in Ref. [9] uses a hierarchical clustering based approach with eigenvectors of the Laplacian matrix of the graph to find the similarity between nodes.

The complexity of the method is $O(n^3)$. Pons and Latapy [10] in their work proposed a hierarchical clustering method measuring the similarity between nodes in graph based on random walks. The method of Pons and Latapy has a time complexity $O(mn^2)$ in the worst case and $O(n^2 \log n)$ in most real-world cases. The method proposed by Rosvall and Bergstrom [11], also uses the concept of random walks and entropy communities to find the community structure. Blondel et al. method [12] is a heuristic method that is based on modularity optimization. It starts from each node as a community, and merged communities based on the modularity criterion. This operation is repeated several times on the set of nodes until no further optimization is possible. Raghavan et al. method [13] is based on label propagation. Initially, every node in the graph is initialized with a unique label and at every step of the method each node takes the label that most of its neighbors currently have. The iterative process converges when labels cannot be changed. The method proposed by Raghavan et al., is non- deterministic method. The time required to run the method is very close to linear time. Xiang et al. method [14] proposed a new metric to quantify the structural similarity between sub-graphs, based on this subgraph similarity an algorithm for community detection is designed.

In this paper, we propose a new split and merge method for community detection based on the minimum spanning tree of graph and modularity. The minimum spanning tree (MST) of the graph is constructed after the graph has been weighted by the dissimilarities of endpoints nodes for each edge. If $(n-1)/2$ highest edges dissimilarities in MST are removed then we get $(n+1)/2$ groups of nodes (communities). Next, these groups of nodes are merged iteratively when the modularity optimization is not possible. Finally, we can construct or produce the community structure.

The rest of the paper is organized in four sections. In Section 2, the proposed method and the corresponding algorithm are introduced. In Section 3, a brief description of the empirical data used in this paper and the performance of our proposed method are discussed. Finally, the paper provides some concluding remarks in Section 4.

2. Method

Given an undirected and unweighted network $G(V, E)$, where $V = (v_1, v_2, \ldots, v_n)$ is the set of nodes, $E = (e_1, e_2, \ldots, e_m)$ is the set of edges, each edge e_l has two endpoints (v_i, v_j) in V. The goal of our community detection method is to partition the network G into k communities (groups): $\pi = \{c_1, c_2, \ldots, c_k\}$, where $c_i \neq \emptyset$, $c_i \cap c_j = \emptyset$, $(i = 1 : k, j = 1 : k)$ and $V = \bigcup_{i=1}^{k} c_i$. The weights of the edges can be calculated with the function $\omega : E \to R$, where each weight value of edge $e_l(v_i, v_j)$ represents the dissimilarity between the nodes v_i and v_j. The equation of dissimilarity is as follows:

$$\omega(e_l) = \omega(v_i, v_j) = \frac{|\Gamma(v_i)| + |\Gamma(v_j)|}{|\Gamma(v_i) \cap \Gamma(v_j)|} \tag{1}$$

where $|\Gamma(v_i)|$ represents the length of neighbors set of node v_i.

The minimum spanning tree (MST) of the given network $G(V, E)$ with the weights $\omega(e_l)$ of each edge in G is constructed to guide the split process. The MST of $G(V, E)$ is a spanning tree $T(V', E')$ such that $W(T) = \sum_{(v_i, v_j) \in T} \omega(v_i, v_j)$ is the minimum [15], $V' = V$, $|V'| = |V| = n$ and $E' \subset E$, $|E'| = n - 1$. In the MST T we select a set R_e, where R_e is the set of $(n-1)/2$ edges in T with the highest weight value. Then, we remove the edges in R_e from T and we get $(n+1)/2$ disconnected components in T. Each component represents a community, then, we have $c_1, c_2, \ldots, c_k$ and $k = (n+1)/2$.

After the MST T has been split into k groups, the merge stage is performed to obtain the final community structure. In the merging process the crucial problem is to determine which pairs of communities should be merged. To solve this problem we base on the number of intra-community links between the communities in the graph G. Two communities should merge if they had more connection (intra-community) to other communities. Eq. (2) shows the function S_{ij} which permitted the selection of two communities to be merged.

$$S_{ij} = \frac{\text{number of intra-community links between } c_i, c_j}{\sqrt{d_{c_i} d_{c_j}}} \tag{2}$$

and

$$d_{c_i} = \sum_{j=1}^{|c_i|} \text{degree}(v_j) \tag{3}$$

where $\text{degree}(v_i)$ denotes the degree of node v_i in the graph G (the number of edges adjacent to v_i).