Physica A 513 (2019) 315–324

Contents lists available at ScienceDirect

Physica A

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

Community detection using boundary nodes in complex networks

Mursel Tasgin *, Haluk O. Bingol

Department of Computer Engineering, Bogazici University, Istanbul, Turkey

HIGHLIGHTS

- A new local community detection algorithm based on boundary nodes is proposed.
- Communities are identified by finding the borderlines of them based on boundary nodes.
- A new decision mechanism for label updates increases the quality of identified communities.
- Unnecessary label propagations are avoided by focusing only on boundary nodes.
- Proposed algorithm is scalable and suitable for very large networks.

ARTICLE INFO

Article history:
Received 3 April 2018
Received in revised form 3 August 2018
Available online xxxx

Keywords:
Complex networks
Community detection
Local algorithms
Label propagation
Boundary nodes
Common neighbors

ABSTRACT

We propose a new local community detection algorithm that finds communities by identifying borderlines between them using boundary nodes. Our method performs label propagation for community detection, where nodes decide their labels based on the largest "benefit score" exhibited by their immediate neighbors as an attractor to their communities. We try different metrics and find that using the number of common neighbors as benefit scores leads to better decisions for community structure. The proposed algorithm has a local approach and focuses only on boundary nodes during iterations of label propagation, which eliminates unnecessary steps and shortens the overall execution time. It preserves small communities as well as big ones and can outperform other algorithms in terms of the quality of the identified communities, especially when the community structure is subtle. The algorithm has a distributed nature and can be used on large networks in a parallel fashion.

1. Introduction

A system consisting of elements can be expressed by using network representation, i.e., nodes denote the elements and edges represent their relations. Many real-life systems, e.g., mobile communication networks, collaboration networks, protein–protein interaction networks are analyzed using network representation [1–3]. A *community* is defined as a group of nodes in a network where nodes within the same group have more connections with each other than the nodes from other groups [4]. *Community detection* is the task of identifying such groups in a network. Although there is not a universally accepted definition of a community, the above definition is used by many community detection algorithms [4–15]. There is a comprehensive survey on community detection methods and algorithms in complex networks by Fortunato [16]. Different aspects and purposes of community detection are investigated in a recent work by Schaub et al. [17]. Authors discuss that

* Corresponding author.
E-mail address: mursel.tasgin@boun.edu.tr (M. Tasgin).

https://doi.org/10.1016/j.physa.2018.09.044
0378-4371/

understanding the motivation of community detection for a specific problem is important for selecting the most suitable algorithm or approach, since there are many facets of community detection.

Many of the proposed community detection algorithms, some of which are nearly a decade old or more, are successful on small networks of hundreds or thousands of nodes. With the availability of very large network datasets having millions or billions of nodes and edges in recent years, there are challenges for community detection algorithms. Many of the existing community detection algorithms are not able to run on such large networks because of their high time-complexity. If a community detection algorithm needs to optimize a global value or a metric regarding the whole network, then it may need to perform an operation or calculation related with all elements of the network (i.e. nodes and edges) many times. Such an approach is computationally expensive and is not feasible on very large networks. Additionally, processing the whole network data may require storing and accessing it many times, which is expensive in terms of data storage, too. A *local community detection* approach, which uses local information around a node while identifying its community, can be a practical solution on very large networks. When the community of each node is decided using such a limited data and calculation, then overall time-complexity of the algorithm will be reasonably low on very large networks. Besides their practicality, local algorithms may be the only viable options on these networks.

In this paper, we propose a new community detection algorithm that has a local approach and tries to find communities by identifying borderlines between them using boundary nodes. Initially, every node is considered to be a boundary node. Our community detection process naturally decreases their numbers by identifying communities of them. In the final situation, only the actual boundary nodes remain and they constitute the borderlines between communities.

Outline of the paper is as follows. We first give background information about our notation, local algorithms and our method of testing. Then we briefly explain our community detection approach. We go into the details of experiments and present the results of our algorithm on both generated and real-life networks and compare it with other algorithms.

2. Background

2.1. Notation

Let $G = (V, E)$ be an unweighted and undirected graph where V is the set of nodes and E is the set of edges. A *community structure* is a partition of V. We label each block in the partition using a symbol in the set of *community labels* $\mathbb{L} = \{1, \ldots, |V|\}$. We define function $L: V \rightarrow \mathbb{L}$, which maps each node in V to a community label in $\mathbb{L}$. That is, the community of node $i \in V$ is given as $L(i)$. If two nodes i and j are in the same community, then we have $L(i) = L(j)$.

In community detection, *triangles*, i.e., three nodes connected by three edges, play an important role [18]. We use two metrics related to triangles. First one, the *clustering coefficient* CC_i of node i, is the probability that two of its neighbors are friends of each other, given as

$$CC_i = \frac{\triangle_i}{\wedge_i}$$

where $\triangle_i$ is the number of triangles around node i and $\wedge_i$ is the number of *triplets*, i.e., i is connected to two nodes, centered at i [19]. The second metric is the number of common neighbors of two nodes, which is generally used for node similarity. The *number of common neighbors* of nodes i and j is given as

$$\cap_{ij} = |\Gamma(i) \cap \Gamma(j)|$$

where $\Gamma(i)$ is the *1-neighborhood* of i, i.e., the set of nodes whose distances to i are 1.

We use the concepts of Xie and Szymanski [12] to mark the nodes. A node i is called an *interior node* if it is in the same community with all of its 1-neighbors. If it is not an interior node, it is called a *boundary node*. Note that boundary nodes are positioned among nodes from different communities.

2.2. Local community detection algorithms

In recent years, several local community detection algorithms have been proposed [11–15]. These algorithms generally discover communities using local interactions of nodes or local metrics calculated in the 1-neighborhood of nodes in the network. Instead of performing a search or a calculation on the whole network (i.e. global), local approach splits the community detection task into separate subtasks on individual nodes and their neighborhoods. Results of these subtasks are then merged together to get the community structure of the whole network.

Raghavan et al. [11] proposed label propagation algorithm, denoted by *LPA*, which updates the community label of each node with the most popular label in its 1-neighborhood, i.e., majority rule of labels. Labels of all nodes in the network are updated asynchronously and algorithm terminates when there is no possible label update in the network. It is a linear-time algorithm, which can identify communities in a fast way. However, it tends to find a single large community, especially when community structure is subtle.

Xie and Szymanski [12] proposed an extension on LPA, which we denote by *LPAc*, using neighborhood-strength driven approach. LPAc improves the quality of identified communities by incorporating the number of common neighbors to the majority rule of labels in LPA. It calculates the scores of labels by first counting the number of members having these labels,