Journal of Network and Computer Applications 40 (2014) 73–84

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Community detection in social networks using hybrid merging of sub-communities

Mohsen Arab [a], Mohsen Afsharchi [b,*]

[a] Department of Computer Science, IASBS, Zanjan, Iran
[b] Department of Computer Engineering, University of Zanjan, Zanjan, Iran

ARTICLE INFO

Article history:
Received 17 October 2012
Received in revised form
10 June 2013
Accepted 21 August 2013
Available online 29 September 2013

Keywords:
Network communities
Bottom up merging
Social networks

ABSTRACT

Network vertices are often divided into groups or communities with dense connections within communities and sparse connections between communities. Community detection has recently attracted considerable attention in the field of data mining and social network analysis. Existing community detection methods require too much space and are very time consuming for moderate-to-large networks. We propose a bottom up community detection method in which starting with fine-grained communities we find real communities of a network. Merging preliminary small communities is done in a hybrid way to maximize two quality functions: modularity and NMI. We show that our way of community detection is better or as effective as the other community detection algorithms while it has better time and space complexity.

1. Introduction

In recent years, community detection has been in the center of attention due to its wide use in data mining, information retrieval and social network analysis. Most of the complex networks usually have modular or community structure and appear as a combination of groups that are fairly independent of each other. Vertices of the same community usually share some common behaviors. For instance people of the same community usually have a set of common properties such as having similar hobbies, working on a research with the same topic and so on. Thus, finding communities enables us not only to extract useful information of complex networks but also to understand how different groups or communities in a network evolve.

The issue of community detection closely corresponds to the idea of graph partitioning in computer science and graph theory, and hierarchical clustering in sociology. Recently, the computer revolution has provided scholars with a huge amount of data and computational resources to process and analyze these data. The size of real networks one can potentially handle has also grown considerably, reaching millions or even billions of vertices. The need to deal with such a large number of units has produced a deep change in the way that graphs are approached (Fortunato et al., 2010).

Since moderate-to-large networks are becoming ubiquitous in our real world, current methods are not satisfactory from the time complexity point of view. In this paper, we present an effective algorithm for finding communities of the graph with a good time and space complexity and also with an acceptable quality of output which is comparable with the existing outputs of recent community detection algorithms. We follow a bottom up approach in which we start community detection by considering every vertex or two vertices as preliminary communities. Then based on a well known criterion which is called "modularity" (Newman and Girvan, 2004), we merge these preliminary communities.

Merging subcommunities must be repeated several times. Although merging all pairs of neighbor communities with highest increase in modularity (i.e. pairwise merging) is a good idea but it is too slow. Merging multiple communities together is more quick but it is less accurate. Therefore, we use both of them and call it "Hybrid" merging. We also use a vertex similarity measure to find small communities which we denote them as preliminary communities and then apply the modularity maximization strategy on these preliminary communities that will result in community detection with better modularity value. Merging is stopped when the maximum modularity achieved.

The structure of the paper is as follows: In the next section we present a review of the literature. In Section 4 we provide a detail discussion of our work which is followed by complexity analysis of the algorithm. Finally in Section 6 we present the result of our experiments.

2. Related works

The most well-known algorithm for community detection was proposed by Girvan and Newman (2002). This method is

* Corresponding author. Tel.: +98 912 772 1914.
E-mail addresses: m_arab@iasbs.ac.ir (M. Arab), afsharchim@znu.ac.ir (M. Afsharchi).

1084-8045/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2013.08.008

historically important due to the opening a new era in the field of community detection. This method uses a new similarity measure called *edge betweenness*. Edge betweenness is referred to the number of shortest paths between all vertex pairs that run along that edge. The algorithm has a complexity $O(n^3)$ on a sparse graph. In the following we will refer to it as GN. In another work (Newman, 2006) Newman reformulated modularity in terms of eigenvectors of a new characteristic matrix for the network and called it modularity matrix. He obtained a time complexity $O(n^2 \log n)$ for sparse graphs(denoted as N_{eig}).

Clauset et al. (2004) have proposed a fast greedy modularity optimization method. Starting from a set of isolated nodes, the links of the original graph are iteratively added such to produce the maximum possible increase in the modularity of Newman and Girvan (2004) at each step. The algorithm has a complexity of $O(n \log^2 n)$ on sparse graphs. In the following we will refer to it as CNM.

A novel divisive algorithm for modularity maximization is presented by Duch and Arenas (2005). The total cost of their algorithm is $O(n^2 \log^2 n)$. In the following we will refer to it as EO.

Another modularity optimization has been presented by Blondel et al. (2008). This is a multi-step technique based on the local optimization of Newman–Girvan modularity in the neighborhood of each node. The computational complexity is essentially linear in the number of links of the graph.

With the spirit of Girvan and Newman, Radicchi et al. have presented another algorithm (Radicchi et al., 2004). In fact, it is a divisive hierarchical method where links are iteratively removed based on the value of their edge clustering coefficient. The algorithm is $O(n^2)$ on a sparse graph.

Cfinder is a local algorithm proposed by Palla et al. (2005) that looks for communities that may overlap. The complexity of this procedure can be high as the computational time needed to find all k-cliques of a graph is an exponentially growing function of the graph size.

Markov Cluster Algorithm (i.e. MCL) is an algorithm developed by van Dongen (2000), which simulates a peculiar diffusion process on the graph. The algorithm is $O(nk^2)$ where $k < n$. The structural algorithm is presented by Rosvall and Bergstrom (2007). Here the problem of finding the best cluster structure of a graph is turned into the problem of optimally compressing the information on the structure of the graph, so that one can recover as closely as possible the original structure when the compressed information is decoded.

Donetti and Munoz presented spectral algorithm (Donetti and Munoz, 2004). The idea is that eigenvector components corresponding to nodes in the same community should have similar values, if communities are well identified. The algorithm is $O(n^3)$. In the following we will refer to it as DM.

Expectation-maximization is another algorithm by Newman and Leicht (2007). Here Bayesian inference is used to deduce the best fit of a given model to the data represented by the actual graph structure. The complexity is parameter dependent.

Liu et al. (2008) utilized several similarity metrics of vertex to transform a community detection problem into a clustering problem, and adopted affinity propagation to extract communities from graphs.

Gregory (2010) used label propagation to find communities. This algorithm has been asserted that can find different kind of communities such as: overlapping communities, weighted and bipartite networks. Vertices have labels that can propagate between neighboring vertices. Once labels have been propagated, it is expected that the vertices within a community have the same labels (is denoted as COPRA).

A fast fine-tuning algorithm presented by Granell et al. (2011) for finding clusters at different topological levels (will be referred as RFT)

Rosvall and Bergstorm (2008) introduced a random walk based algorithm for detecting modules of networks. The modules can be detected by compressing information on the network (is referred as Infomap).

3. Evaluation criteria

Finding ideal algorithms of community detection aims at two main goals, i.e. improving the accuracy in the determination of meaningful modules and reducing the computational complexity of the algorithm. Reducing the computational complexity is a well defined objective: in many cases (i.e. this work) it is possible to compute analytically the complexity of an algorithm, in others one can derive it from simulations of the algorithm on systems of different sizes. The main problem is then to estimate the accuracy of a method and to compare it with other methods. To evaluate the accuracy of a community detection algorithm, it should be tested on artificial and real world networks. For artificial networks, we use the normalized mutual information (NMI) measure (Danon et al., 2005) to compare the known partition with the partition found by each algorithm. For real-world networks, since we do not know the real community structure, we use the modularity measure (Newman and Girvan, 2004) to assess the quality of a partition.

3.1. Modularity

Basically we need a function to evaluate the goodness of partitioning of a graph into clusters. The first criterion is *modularity* which has the unique privilege of being at the same time a global criterion to define a community, a quality function and the key ingredient of the most popular method of graph clustering. This criterion which is introduced by Newman and Girvan (2004) formally defined as follows:

$$Q = \sum_i e_{ii} - a_i^2 \tag{1}$$

where e_{ii} is the fraction of edges that connects two nodes inside the community i and a_i represents the fraction of edges that connect two vertices in community i (i.e. having one or both vertices inside the community i). The sum extends to all communities i in a given network. The larger the Q is, the corresponding partition would be more accurate.

In the other words, e_{ii} is the real fraction of edges within a community i. With disregarding the underlying structure, the expected value of the fraction of links within a community can be estimated. ${a_i}^2$ is simply the probability that an edge begins at a vertex in community i, multiplied by the fraction of edges that end at a vertex in community i. So, the expected number of intra-community edges is just $a_i a_i$. We can compute these two values directly and sum over all the communities in the graph (Danon et al., 2005). In the next section we will elaborate modularity criterion in the context of our own work.

3.2. Normalized mutual information

In recent years, modularity maximization technique has been revealed to have some limitations in finding communities (Fortunato and Barthelemy, 2007; Good et al., 2010; Lancichinetti and Fortunato, 2011). In fact, it has an intrinsic tendency to merge small communities and also split big ones at the same time. These limitations made it questionable to consider it as a quality function. Instead, another quality function called normalized mutual information (NMI) have been introduced (Lancichinetti et al., 2008). NMI has become a standard as a measure of