Pattern Recognition Letters 36 (2014) 46–53

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Optimal local community detection in social networks based on density drop of subgraphs

Xingqin Qi [a], Wenliang Tang [b], Yezhou Wu [b], Guodong Guo [c], Eddie Fuller [b], Cun-Quan Zhang [b,*,1]

[a] School of Mathematics and Statistics, Shandong University (Weihai), Weihai 264209, China
[b] Department of Mathematics, West Virginia University, Morgantown, WV 26506, USA
[c] Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26506, USA

ARTICLE INFO

Article history:
Received 12 April 2013
Available online 20 September 2013

Communicated by M.A. Girolami

Keywords:
Hierarchical clustering
Community selection
Graph density
Density drop

ABSTRACT

The determination of community structures within social networks is a significant problem in the area of data mining. A proper community is usually defined as a subgraph with a higher internal density and a lower crossing density with others subgraphs. Hierarchical clustering algorithms produce a set of nested clusters, sometimes called dense subgraphs, organized as a hierarchical system and the output is always referred as a dendrogram. However, determining which of clusters in the dendrogram will be selected to form communities in the final output is a difficult problem. Most implementations of data mining algorithms require expert guidance in the implementation of the algorithm in order to establish the appropriate selection of such communities, and ultimately the output may not be optimized as with fixed height tree-cutting algorithms. In this paper, a novel algorithm for community selection is proposed. The intuition of our approach is based on drops of densities between each pair of parent and child nodes on the dendrogram – the higher the drop in density, the higher probability the child should form an independent community. Based on the Max-Flow Min-Cut theorem, we propose a novel algorithm which can output an optimal set of local communities automatically. In addition, a faster algorithm running in linear time is also presented for the case that the dendrogram is a tree. Finally, we validate this approach through a variety of data sets ranging from synthetic graphs to real world benchmark data sets.

1. Introduction

Networks can be used to describe the pairwise relationships between nodes. Thinking of these nodes as vertices, we can in turn view such networks as graphs where the edges are defined by these same pairwise relationships. Sociologists use networks to describe the relationship among n persons in terms of their connection strength, reflecting how of a connection exists between pair, common behaviors, or the level of collaboration. The subgraph with denser connections inside and sparser connections to other subgraphs can provide invaluable insight into the structure of the whole network or data visualization. Detecting such communities or clusters of closely related objects remains one of the most interesting problems in the field of bioinformatics, social networks, epidemiology and data mining. Many clustering algorithms have been proposed in the literature (Girvan and Newman, 2001; Newman, 2004, 2006; Hastie et al., 2001; Kaufman and Rousseeuw, 1990; Scott, 2000).

Hierarchical clustering, see Scott (2000), is one of the most popular approaches to clustering problems and the output is called dendrogram. A dendrogram is a diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering (see Fig. 1(a)).

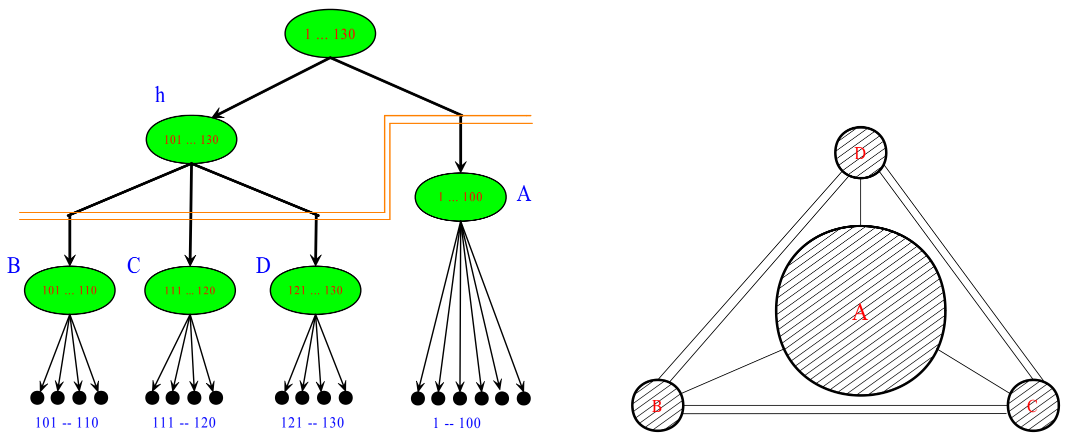
The dendrogram has a root, say v_0, on the top of diagram representing the whole network and leaves on the bottom representing each individual. Every node among internal levels represents a subgraph of the original network. The node on the lower level is called *child*, and one on higher level is called *parent*. Each edge connecting two nodes in the dendrogram forms a parent–child relationship. The subgraphs induced by these parent–child relationships, specifically the hierarchy of sets of nodes in the original graph, are called *clusters* and form the candidate pool of communities for output. However, which of those clusters in the dendrogram will be selected to form communities in final output?

"There are no completely satisfactory algorithms that can be used for determining the number of population clusters for many type of cluster analysis" said in SAS/STAT 9.2 User's Guide. It is much harder to find out an optimal community partition than determining the number of communities, and therefore it remains one of the most challenging problems in current research of data mining.

[*] Corresponding author. Fax: +1 304 293 3982.
E-mail addresses: cqzhang@math.wvu.edu, cqzhang@mail.wvu.edu (C.-Q. Zhang).

[1] This research is partially supported by an NSA Grant H98230-12-1-0233 and an NSF Grant DMS-126480.

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.09.008



(a) A dendrogram obtained by Quasi Clique Merger on left example

(b) A network consisting of 4 strongly connected dense subgraphs.

Fig. 1. An example of dendrogram obtained by Quasi Clique Merger. Four clusters are formed by picking up all nodes immediately adjacent to edge cut, colored with orange. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A subgraph with denser connections indicates that members are more similar to each other than they are to portions of the graph outside the subgraph. If we use "density" of a subgraph, defined to be the average number of edges between nodes in the vertex set of the subgraph, to describe its global strength connections (see Section 2.2 for more detailed definition), a good community detection algorithm should detect a partition of the input networks into subgraphs satisfying:

1. Higher internal connection density.
2. Lower external connection density.

The traditional algorithm of identifying communities of a dendrogram is referred to as *tree cutting*, branch cutting or branch pruning. One kind of tree cutting algorithm needs the number of communities as an input aforehand, but the problem of determining the number of clusters itself is hard in most cases. Another most widely used tree cut algorithm is called *fixed height cutting*: the user chooses a fixed height on the dendrogram, and all nodes in the branches immediately below the height of the cut form the family of communities. The fixed height tree cutting is simple and rather naive, but the output sometimes does not make any sense especially for complicated cases. The following example will reveal the downside of fixed height cutting.

Let G be a network consists of 4 giant clusters A–D (see Fig. 1(b)). The subgraph induced by A is a complete subgraph of order 100 and each edge is weighted by 3, and ones induced by B–D are also complete and of order 10. Each edge in those three clusters is assigned 4 and all rest of crossing edges are weighted by 1. Fig. 1(a) is the dendrogram generated by a density driven clustering algorithm. By applying traditional fixed height cut algorithm, one may produce an output consisting of two communities of orders 100 and 30, respectively (see Fig. 1(a)), while the output of 4 communities consisting of A–D respectively should be the true clustering result.

Fixed height cutting is a simple and naive technique with many desirable properties, but unreliable when the dendrogram is large and complicated as we have seen from the above example. Another community selection technique, called "dynamic tree cut", was discussed in Carlson et al. (2006), Dong and Horvath (2007), Ghazalpour et al. (2006) and Gargalovic et al. (2006), which is mainly based on the shape of branches of dendrogram and the inner structure of each node was not reflected in the process of community detecting.

In this paper, we will propose a new community detection algorithm (Algorithm 1), different from other traditional algorithms (fixed height cutting algorithm where all edges to be cut are in the same height level or pre-defined community number algorithm where the number of communities should be inputted aforehand), where the community result will be output automatically and the edges to be cut could be located in any level of the dendrogram. To be more specific, our algorithm will find an edge cut of a given dendrogram, separating the root and all leaves, where the edges in the edge cut could be located in any level. The family of all nodes (children) immediately below the edge cut will be the output of our algorithm and form all desired communities automatically.

The basic idea of our algorithm is as follows. As we know, each node v in the dendrogram is a candidate of the optimal local community and the induced subgraph G_v by its members has relatively higher inter-connection than extra-connection. Those arcs in the dendrogram with larger density drop indicate improper agglomeration and hence form candidates for edge cutting. Based on those observations, we assign weights on arcs of T based on density drop between child and parent. Our community detection algorithm could catch all arcs with larger density drop and automatically generate a proper community partition. When we test our algorithm on the above example (see Fig. 1(b)), on which traditional community detection algorithm fails, 4 communities consisting of A–D respectively are obtained as we expected.

The outline of this paper is as follows. In Section 2 we shall describe classical algorithms and review the Quasi Clique Merge (QCM) algorithm, whose output dendrogram will be the start point of our new community detection algorithm. In Section 3, the new algorithm (Algorithm 1) will be described in detail. In addition, a faster algorithm (Algorithm 2) running in linear time is also presented in Section 3 for the special case when the dendrogram is a tree. In Section 4 we apply our algorithm to some classic social networks and compare its result with that of known clusters, which verify our new algorithm's utility.