Applied Mathematics and Computation 217 (2010) 2007–2014

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A fast and efficient algorithm to identify clusters in networks ☆

Francesc Comellas *,[1], Alicia Miralles

Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya, Spain

ARTICLE INFO

Keywords:
Graphs
Clusters
Networks
Complex systems

ABSTRACT

A characteristic feature of many relevant real life networks, like the WWW, Internet, transportation and communication networks, or even biological and social networks, is their clustering structure. We discuss in this paper a novel algorithm to identify cluster sets of densely interconnected nodes in a network. The algorithm is based on local information and therefore it is very fast with respect other proposed methods, while it keeps a similar performance in detecting the clusters.

1. Introduction

Many real life networks like the WWW, Internet, transportation and communication networks, or even biological and social networks have a strong clustering structure (they contain groups of vertices which are highly interconnected, having many mutual neighbors). Here we consider the notion of cluster in a general way. Therefore, depending on the context, it can be synonymous of community, class, module, etc. The problem of detecting clusters in a given network is an important issue in social studies, biological (epidemiology, ecological webs, metabolic), and computer science (WWW, Internet, distributed systems, cluster computing). Clusters are also interesting as they reflect hierarchical aspects and are related to classification issues for information retrieval. Clusters also play an important role when executing most communication algorithms and should be considered to improve their performance.

The construction of efficient and fast algorithms for the identification of the clustering structure in a generic network is a nontrivial task. The first problem is the nonexistence of a precise definition of cluster. Intuitively, a network can be said to have cluster structure if it consists of subsets of nodes, with many connections among the same subset, but few links between subsets, see, for example [1,2]. Algorithms to detect these subsets have appeared in the literature and they can be classified in two main groups (see the above two references for more details): hierarchical clustering methods (also known as agglomerative), which consist of generating a tree (dendrogram) from a complete graph with as many vertices as the original network and where each edge has a weight measuring how close the corresponding vertices are. Starting from the set of all vertices with no edges between them, edges are iteratively added between pairs of vertices in the order of

☆ Research supported by the Ministerio de Educación y Ciencia, Spain, and the European Regional Development Fund under project TEC2005-03575 and by the Catalan Research Council under project 2005SGR00256.

* Corresponding author.
E-mail addresses: comellas@ma4.upc.edu (F. Comellas), almirall@ma4.upc.edu (A. Miralles).
URL: http://www.ma4.upc.edu/comellas/ (F. Comellas).
[1] Avda. Canal Olímpic s/n, 08860, Castelldefels, Catalonia, Spain.

0096-3003/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2010.06.060

decreasing weight. From the tree one can then infer the different clusters. To obtain the weights, some algorithms consider the spectrum of the adjacency matrix of the graph representing the network. The other class of algorithms is called divisive. From the whole graph, by iteratively cutting the edges, one obtains a set of disconnected subgraphs identified as clusters. Of course, the selection of the edges to be cut, i.e. identifying those connecting clusters, is the crucial point of a divisive algorithm. Girvan and Newman (GN) [1] have recently provided a divisive algorithm based on the "edge betweenness": the betweenness of an edge is the number of shortest paths, between all pairs of vertices of a graph, that go through this edge. If a graph is made of dense loosely interconnected clusters clearly all shortest paths between vertices in different clusters will go through a few edges, joining the clusters, which will have a large betweenness value. The main step of the GN algorithm is the computation of the edge betweenness of all the edges and then the removal of those with the highest value. An iterative process allows the obtention of the clusters. This process, however, is computationally expensive as for a graph with n vertices and m edges the cost is $O(nm^2)$, making it impractical even for relatively small graphs.

More recently Newman [3] and Radicchi et al. [2] have provided faster, but less precise, methods. Newman's new method is based on what he calls "modularity" which measures the fraction of edges in a graph connecting different possible clusters and selecting these clusters by using a standard greedy optimization algorithm. The algorithm is $O((n+m)n)$ for the worst case and $O(n^2)$ on a sparse graph. On the other hand Radicchi et al. use the "edge clustering coefficient" defined as the number of triangles to which a given edge belongs, divided by the number of triangles that might potentially include it, given the degrees of the adjacent vertices. This algorithm is $O(m^2)$ in the worst case. A recent survey [4] compares these and other clustering algorithms.

In this paper we introduce a fast and efficient deterministic algorithm which uses local information based on each vertex. The algorithm is also agglomerative and has order $O(mn)$, providing similar results to the GN method and other faster methods.

2. Notation and definitions

We model networks as graphs. Given the graph $G=(V,E)$ with vertex set V and edge set E, we will denote by $\Gamma_k(v)$ the set of vertices at a distance k from a vertex v. Sometimes we will write, for short, $\Gamma(v)$ instead of $\Gamma_1(v)$. Thus, the *degree* of v is $\deg_G(v)=\delta(v):=|\Gamma(v)|$. In terms of the adjacency matrix of G, A_{ij}, we have $\delta(v_i)=\sum_{v_j\in V}A_{ij}=\sum_j A_{ij}$. We denote as Δ the maximum degree of a graph.

If we consider a subset of vertices $C\subset V$ and a vertex $v_i\in C$, we can split the degree of v_i into two contributions (with respect to the subset where it belongs): $\delta_C(v_i)=\delta_C^{in}(v_i)+\delta_C^{out}(v_i)$, where $\delta_C^{in}(v_i)=\sum_{v_j\in C}A_{ij}$ is the number of edges connecting this vertex v_i to other vertices of C and $\delta_C^{out}(v_i)=\sum_{v_j\notin C}A_{ij}$ is the number of edges towards vertices in the rest of the graph.

A cluster is roughly defined as a part of a network where internal connections are denser than external ones. In the implementation of our algorithm we consider a definition of cluster which is a normalized version of the definition of "cluster in the weak sense" as it appears in Radicchi et al. [2]. These authors define *cluster in a weak sense* as a subgraph C such that $\sum_{v_i\in C}\delta_C^{in}(v_i)>\sum_{v_i\in C}\delta_C^{out}(v_i)$. This is in contrast with the definition of *cluster in a strong sense* which for a subgraph C is $\delta_C^{in}(v_i)>\delta_C^{out}(v_i),\quad \forall v_i\in C$. This last definition has also been proposed in [5] for the identification of web clusters.

The problem of finding a partition of the vertex set which corresponds to an optimal clustering structure of the graph and in particular to the identification of this optimality has led to the introduction of modularity by Newman [3]. It has been shown that the problem is NP-hard [6]. Therefore it is of interest to provide algorithms that can produce near-optimal solutions quickly. On the other hand even the modularity method does not work for extremal artificial cases [7], and very often the experimentation with data coming from real networks provides best method to test the usefulness of a given method.

In our algorithm we will use the other definitions: If $C\subset V$, $\deg_C(v_i)$ is the number of edges between v_i and all other vertices in C. Then we have: $\deg_C(v_i)\leqslant\deg_G(v_i)$.The *average degree* of a graph $G(V,E)$ is $\mathrm{avgdeg}(G)=\sum_{v_i\in V}\deg_G(v_i)/|V|$. The *normalized average degree* of a set of vertices C is defined as

$$\mathrm{n\_avgdeg}(C)=\frac{\sum_{v_i\in C}\frac{\deg_C(v_i)}{\deg_G(v_i)}}{|C|}$$

and we have $0\leqslant \mathrm{n\_avgdeg}(C)\leqslant 1$. This parameter is a good measure to characterize if a set of vertices C is clustered. It takes into account, for each vertex in C, the distribution in and out of the set of its edges and also the total number of edges among all vertices in C. Thus, it is a clear improvement over the other cluster characterizations mentioned above. The parameter, in