Modularity clustering on trees

Sonia Cafieri, Pierre Hansen

Modularity Clustering on Trees

Sonia Cafieri[1], Pierre Hansen[2]

[1] École Nationale de l'Aviation Civile - ENAC (Lab. MAIAA)
7 av. Edouard Belin, 31055 Tououse (France)
sonia.cafieri@enac.fr
[2] GERAD, HEC,
3000 chemin de la Côte-S.te-Catherine, H3T 2A7 Montréal (Canada)
pierre.hansen@gerad.ca

Mots-clés : *clustering, tree, network, modularity maximization, dynamic programming*

1 Introduction

Network (or graphs) are often used to represent complex systems in a variety of fields. Tipically, the set of vertices of the graph corresponds to the set of entities under study and the edges represent relations holding for pairs of vertices. The analysis of networks is so gaining more and more attention, specially the analysis devoted to the detection of clusters, or communities. Clusters correspond in fact to subgroups of entities among which strong relations exist. In this paper we are particularly interested in clustering problems in which the underlying graph is a tree $T = (V, E)$. Trees are undirected graphs in which any two vertices are connected by exactly one simple path. Clustering on trees [3] arises in a variety of applications, such as phylogenetic analysis in biological classification and the analysis of communication or distribution networks which exibit a tree-like structure.

Graph clustering is usually formulated as an optimization problem, where the objective function expresses a given clustering criterion. Newman and Girvan [4] introduced the concept of *modularity*, which is currently a widely used criterion. It has been introduced for general graphs and is defined as the sum for all clusters of the difference between the fraction of edges they contain and the expected fraction of edges they would contain if all edges were drawn at random, keeping the same degree distribution. Maximizing modularity gives an optimal partition with its optimal number of clusters. Many heuristics have been developed to maximize modularity. They rely upon agglomerative and divisive hierarchical clustering, and partitioning schemes based on simulated annealing, genetic search, and a variety of other approaches (see [2] for an in-depth survey). Exact algorithms have been proposed only in a few papers (see [1] for recently proposed algorithms based on column generation). In this paper we present a new algorithm for modularity maximization, specialized for trees. The next section outlines the main features of this algorithm.

2 An algorithm for modularity maximization on trees

The proposed algorithm is based on dynamic programming and proceeds by bottom-up computations. First, in a preprocessing step, the algorithm proceeds to labeling of its vertices and edges. To that effect, a center of the tree is found. Then, vertices are given a level equal to the distance to the center and labeled accordingly beginning at the lowest level ; edges are labeled with the same label as their lower vertex. Then, to proceed with bottom-up computations, lists of triplets associated with edges are generated. They characterize the situation relative to the edges they are associated with and to the subtree rooted at their lower vertices. A triplet (m_s, d_s, q_s) has the following meaning : m_s is equal to the number of edges in the connected

component containing the upper vertex v of the edge with which the triplet is associated ; d_s is the sum of degrees of this subtree and q_s is the sum of modularities of the clusters within the subtree and not containing v. Two operations are considered to update the set of triplets : extension and merging. Extension considers the effect of adding or not an edge (u, v) to the subtree rooted at v. It applies to the current list of triplets associated with the edge (u, v). Merging considers the effect of combining two at a time, in increasing order of labels, two subtrees rooted at the same vertex v.

Several dominance rules allow elimination of a large number of triplets. First, one can note that all pendent edges must belong to all optimal solutions. Then, triplets within the same list may be dominated by other triplets. Furthermore, any cut edge induces a subtree not containing the root with maximum (local) modularity, i.e. this modularity can be optimized regardless of the other subtrees. This implies that in the list of triplets corresponding to cut edges only the triplet with maximum modularity needs to be kept.

To obtain the final solution, one has just to observe that the set of cut edges is complementary to the set of connected subtrees, i.e., the optimal partition is given by the connected subtrees induced by all cut edges.

We present numerical results on a set of randomly generated instances, which show the efficiency of the proposed algorithm.

Références

[1] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, L. Liberti, S. Perron. Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82(4) :046112, 2010.

[2] S. Fortunato. Community detection in graphs. *Physics Reports*, 486 :75-174, 2010.

[3] Maurizio Maravalle, Bruno Simeone, Rosella Naldini. Clustering on Trees. *Computational Statistics and Data Analysis*, 24 :217-234, 1997.

[4] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69 -026133, 2004.