Applied Soft Computing 13 (2013) 4303–4319

Contents lists available at ScienceDirect

Applied Soft Computing

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

A traffic-based evolutionary algorithm for network clustering

Maurizio Naldi[a,*,1], Sancho Salcedo-Sanz[b,2], Leopoldo Carro-Calvo[b,2], Luigi Laura[a,1], Antonio Portilla-Figueras[b,2], Giuseppe F. Italiano[a,1]

[a] Università di Roma "Tor Vergata", Dipartimento di Ingegneria Civile e Ingegneria Informatica, Via del Politecnico 1, 00133 Rome, Italy
[b] Department of Signal Theory and Communications, Universidad de Alcalá, Madrid, Spain

ARTICLE INFO

Article history:
Received 27 September 2012
Received in revised form 23 April 2013
Accepted 19 June 2013
Available online 5 July 2013

Keywords:
Clustering
Traffic matrices
Genetic algorithms

ABSTRACT

Network clustering algorithms are typically based only on the topology information of the network. In this paper, we introduce traffic as a quantity representing the intensity of the relationship among nodes in the network, regardless of their connectivity, and propose an evolutionary clustering algorithm, based on the application of genetic operators and capable of exploiting the traffic information. In a comparative evaluation based on synthetic instances and two real world datasets, we show that our approach outperforms a selection of well established evolutionary and non-evolutionary clustering algorithms.

1. Introduction

Clustering, organizing a collection of items into groups on the basis of their similarity, is a well known problem in many different areas. Its applications span fields as different as image segmentation, object and character recognition, customer classification for marketing, and genomic, just to name a few. In the two survey papers by Jain, written more than ten years apart, we can recognize the expansion of clustering, both in methods and in applications [1,2].

Graphs also can be clustered into groups of nodes, each cluster including vertices that are strongly interconnected among them: there should be many edges within each cluster and relatively few between clusters [3]. Graphs can represent a number of interrelated entities. For example, when a graph represents a social network, the vertices are individuals, and the edges represent relationships among them. The problem of detecting communities of individuals within the whole society is itself a clustering problem. In the case of a communications network, the vertices represent network nodes (routers or other switching devices), and the edges represent transmission links. In the following, we refer to nodes and links rather than vertices and edges.

In spite of its importance, the problem of network clustering has been approached so far mainly by considering topology information only. The criterion employed is then based on the number of links lying respectively inside a cluster and among different clusters: the relation between two nodes is entirely embodied by their sharing a link. Adding a weight to each link (so that the relation between two nodes may be stronger or weaker) recognizes the relevance of the intensity of the relationship, but, even in weighted networks, two nodes are related only if they are connected by a direct link. Variations of this approach consider the energy consumption (related to the physical distance) to minimize the total consumption in a wireless network [4,5].

In [6], we put forward the use of traffic information to cluster the nodes of a network. Such information is contained in the traffic matrix of a network, and represents the actual intensity of the communication between two nodes, regardless of the network topology and the route employed to get the messages from the sender to the receiver: the more two nodes communicate between them, the larger their traffic is. The range of applications in which adding traffic information should lead to improvements is very wide: basically all networks in which traffic does not flow exclusively between neighboring nodes. For example, that's been shown in [7] for telephony traffic. Another example is given by social networks in which many relationships are indirect, and an individual is used as a transfer means to convey information (or any other mode of

* Corresponding author. Tel.: +39 0672597269.
E-mail addresses: naldi@disp.uniroma2.it (M. Naldi), sancho.salcedo@uah.es (S. Salcedo-Sanz), Leopoldo.Carro@uah.es (L. Carro-Calvo), laura@dis.uniroma1.it (L. Laura), antonio.portilla@uah.es (A. Portilla-Figueras), italiano@disp.uniroma2.it (G.F. Italiano).

[1] The work of Maurizio Naldi, Luigi Laura, and Giuseppe F. Italiano has been partially supported by the Italian Ministry of Education, University, and Research through the ALGODEEP Project, and by the European Union under the EuroNF Network of Excellence.
[2] The work of Sancho Salcedo-Sanz, Leopoldo Carro-Calvo, and José Antonio Portilla has been partially supported by the Spanish Ministry of Science and Innovation, under a project number ECO2010-22065-C03-02.

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.06.022

relationship, so that traffic is meant here in a broad sense as anything that is exchanged by two nodes) between two other parties (see, e.g., [8] for the discovery of hidden relationships and [9] for gossip networks). In [6], we adapted two quality metrics from the context of topology-based clustering algorithms, to make them applicable in a traffic-based approach, namely the *Traffic-aware Scaled Coverage Measure* and the *Modularity* measure. In [10], we proposed a preliminary version of an evolutionary clustering algorithm, and performed a first comparison against the Spectral Filtering algorithm, a major non-evolutionary clustering algorithm.

In this paper we fully embrace the traffic-based approach for network clustering, and propose a novel evolutionary algorithm based on the use of genetic operators, which we name EC (Evolutionary Clustering). We embed the quality metrics recalled above in the fitness function of the evolutionary procedure, so that our algorithm aims at maximizing the quality of the clustering solution as evaluated through those metrics. We have tested our approach against four competing topology-based clustering algorithms and against an existing evolutionary approach (EvoCluster [11]) on synthetic and real world datasets. We fully describe our evolutionary algorithm and report the results of that comparative evaluation. We compare the first two statistical moments of the two metrics (which represent a measure of central tendency and dispersion), while in previous works the comparison was limited to the scatterplot of the metrics. In this paper, we also compute the percentage of success of the Evolutionary Algorithm against its non-genetic competitors. The comparison performed in this paper is completed by a thorough analysis of the computational cost. The key results obtained in this work are: (1) We show that our evolutionary algorithm achieves better values of both quality metrics than the topology-based alternative clustering algorithms; (2) We provide an analysis of the computational cost of the EC algorithm; (3) we show that its computational cost is lower than that of the topology-based competitors (excepting *K*-means). We have also performed a comparison with the above selection of non-evolutionary algorithms and the evolutionary algorithm EvoCluster, using a synthetic dataset with larger traffic matrices. Also in the case of synthetic matrices, our EC algorithm outperforms all the other algorithms, with the only exception of Newman's and *K*-means for the larger traffic matrices when the modularity metric is used.

The rest of the paper is organized as follows. In Section 2 we recall the notion of traffic matrix and its use in the context of network clustering. In Sections 3 and 4 we describe respectively our traffic-based Evolutionary algorithm and the topology-based competitors employed in our comparative evaluation. Sections 5.1 and 5.2 are devoted to set the performance evaluation context, respectively through the definition of the quality metrics and the description of the real world datasets. Finally, in Sections 5.3–5.5, we describe the results of the comparison, under the two viewpoints of the quality of the clustering solutions (for the real world datasets and the synthetic one) and the computational cost.

2. Traffic information and clustering

Network clustering is traditionally performed on the basis of topology information. Roughly speaking, two nodes belong to the same cluster if they are strongly interconnected. In this paper, we advocate instead the use of traffic information to partition the network into clusters. In this section, we review the tools that gather respectively the topology information used for clustering in the traditional approach, and the traffic information employed in ours; we compare them, and provide motivations for the use of traffic information.

When we cluster a network on the basis of the connectivity information only, we employ the adjacency matrix A. The generic element A_{ij} of that matrix equals 1 if the nodes i and j are connected by a link, and 0 otherwise. When links are bidirectional (which is usually the case in communications networks), the adjacency matrix is symmetric. Though two nodes may be considered strongly related if they are directly connected, clustering based on the adjacency matrix fails to consider the case where two nodes have a strong relationship even if they are topologically distant.

For this reason, we introduce traffic matrices as the basis for network clustering. Traffic represents the intensity of the relation between two nodes, regardless of the way those nodes are connected. Nodes that communicate heavily between them, as indicated by the traffic matrix, should be put into the same cluster, though they are not directly connected. In a traffic matrix X, the element X_{ij} provides the traffic originated by node i and destined for node j. Despite the use of the term *traffic*, there are several possibilities as to the actual quantity used to represent traffic. In [12], a two-level taxonomy of traffic matrices is proposed, based on the spatial representation of network traffic used and the aggregation level for the sources and destinations engaging in traffic exchanges. In addition, we may consider either intensity values (averages over a measurement time window, typically an hour long) or volume values (accumulated over a typically much longer observation window, e.g., over a month), depending on the purpose of the traffic matrix [13]. The resulting matrix is generally asymmetric, even for a network with all bidirectional links. Their asymmetry makes methods employed for weighted networks unsuitable, since they typically assume a symmetric weight matrix [14]. Traffic matrices are dense, usually complete, as opposed to the usually sparse structure of adjacency matrices. The elements of traffic matrices are real numbers, rather than Boolean values. In addition, they often vary considerably even over small time frameworks, while the topology is much stabler, with changes due typically to failures or planned interventions. Contrary to adjacency matrices, traffic matrices are independent of the internal topology of the network. Moreover, when the nodes i and j are supposed to be respectively the ultimate source and destination of that traffic, the traffic matrix is also insensitive to routing changes, which represents a further advantage in their use and a spur to estimate them as accurately as possible [15,16].

3. The evolutionary clustering algorithm

The main objective of this paper is to introduce a new evolutionary algorithm to perform a partitional clustering of a network on the basis of traffic matrices and a fitness function that describes the quality of the clustering solution. By partitional clustering we mean an approach where each node is assigned to a single cluster: clusters do not overlap and represent a partition of the network. In this section, we describe that algorithm.

A primary issue in any clustering algorithm is the choice of the number of clusters. Some algorithms need it to be defined a priori, while others include the number of clusters as a variable to be optimized during the clustering process, jointly with the composition of each cluster. The review in [17] adopts that feature to classify clustering algorithms into two classes: algorithms with either a fixed or a variable number of clusters.

Our algorithm does not require the number of clusters to be decided a priori. Rather, the candidate solutions can be composed of different numbers of clusters, so that it can be considered to belong in the latter class. Nevertheless, our algorithm requires the maximum number k^* of clusters to be set, so that the candidate solutions can be made of any number of clusters in the range between one and that maximum. The number of clusters $k \leq k^*$ is therefore an outcome of the algorithm.