Journal of Statistical Mechanics: Theory and Experiment
An IOP and SISSA journal

Detecting network communities: a new systematic and efficient algorithm

Luca Donetti[1,2] **and Miguel A Muñoz**[1,2,3]

[1] Instituto de Física Teórica y Computacional Carlos I, Facultad de Ciencias, Universidad de Granada, 18071 Granada, Spain
[2] Departamento de Electromagnetismo y Física de la Materia, Facultad de Ciencias, Universidad de Granada, 18071 Granada, Spain
E-mail: donetti@onsager.ugr.es and mamunoz@onsager.ugr.es

Received 27 April 2004
Accepted 18 October 2004
Published 27 October 2004
Online at stacks.iop.org/JSTAT/2004/P10012
doi:10.1088/1742-5468/2004/10/P10012

Abstract. An efficient and relatively fast algorithm for the detection of communities in complex networks is introduced. The method exploits spectral properties of the graph Laplacian matrix combined with hierarchical clustering techniques, and includes a procedure for maximizing the 'modularity' of the output. Its performance is compared with that of other existing methods, as applied to different well-known instances of complex networks with a community structure, both computer generated and from the real world. Our results are, in all the cases tested, at least as good as the best ones obtained with any other methods, and faster in most of the cases than methods providing similar quality results. This converts the algorithm into a valuable computational tool for detecting and analysing communities and modular structures in complex networks.

Keywords: random graphs, networks

[3] Author to whom any correspondence should be addressed.

PII: S1742-5468(04)87880-4 1742-5468/04/P10012+15$30.00

Contents

1. Introduction

The outburst of activity in the field of complex networks in recent years has been rather spectacular and amazing. Networks of any thinkable (and sometimes ‘unthinkable’) type, including social, biological and technological ones, have been described, and their topological as well as dynamical features studied. A whole line of research has emerged and a new perspective for tackling complex problems created. See [1]–[5] for reviews from different perspectives and for exhaustive lists of references.

One particular aspect which has drawn much attention is the existence of subsets of nodes highly linked among themselves but loosely connected to the rest of the network, i.e. *communities*. These are believed to play a central role in the functional properties of complex structures [6, 7]. Identifying communities and analysing their nature is an important task in some fields—for instance, computer science [8, 9], sociology [6, 10], biochemistry [11], bibliometrics [12], taxonomy—and, as a more specific instance, in the development of efficient search engines for the WWW. According to Flake *et al* [13], ‘as the Web is self-organized into communities, search engines implementing such a concept would help surfers to find what they look for and avoid other contents’.

The concept of ‘community’ may be retained as rather vague and phenomenological. Indeed, depending on the network under scrutiny, it might be quite an artificial one, while, in other cases, it emerges as a very natural and useful structure analysis tool. A way to make the concept more clear-cut and practical is through the definition of the *modularity*, Q (see below and [14, 15]), a quantity which provides a way to quantify the community structure of a given network. Other quantities have been proposed with the same purpose [7, 16, 17].

The problem of finding communities is not new and is closely related to the problem of graph partitioning, profusely studied in the context of computer science [18, 19]. A review of some techniques used, including further references, can be found in [6, 20]. Related problems are image processing and pattern recognition, or more generically data clustering: in these cases there is no underlying network, but instead some relation or similarity between existing elements can be established [21]–[23].

In recent years many algorithms for detecting communities have been proposed, starting with the seminal work by Girvan and Newman [6, 14]. These authors proposed an iterative, *divisive* (as opposed to *agglomerative*) method based on the progressive removal of links with the largest *betweenness*, a quantity proportional to the number of shortest paths passing through a given edge [24]. The edges (or links) with the largest betweenness have the most prominent role in connecting different parts of the graph and, therefore, by removing them recursively a good separation of the network into its components or communities can be found. This method generates very good results and has been employed by different authors in studies of various kinds [25]. Unluckily, as already pointed out by the authors themselves, it has a main disadvantage: its computational demand is very high. For instance, for sparse networks with N nodes, the computation time grows like N^3. In order to deal with large networks, for which the previous algorithm turns out to be not viable, Newman himself developed a faster method (of order N^2). It is based on the iterative agglomeration of small communities, starting from isolated nodes, by locally optimizing the modularity. This method generates worse results[4] than the previous one.

Some alternative algorithms, both divisive and agglomerative (which we do not attempt to exhaustively overview here), have been proposed in recent months. Some of them are listed here in chronological order (see [20] for a more critical discussion of some of them):

- The Radicchi–Castellano–Cecconi–Loreto–Parisi [17] method is of order N^2. It is a divisive algorithm that works nicely whenever triangular (or higher order) loops are present in the network.
- The Wu–Huberman algorithm [26]. This is a fast method (linear in N), based on the idea of voltage drops, which visualizes the network as an electric circuit. It can be used to locate the community to which one specific node belongs, but it requires successive iterations of the method in order to provide a global network division in communities.
- The Reichardt–Bornholdt method [27]. In their recent paper these authors introduced an algorithm inspired by the celebrated *superparamagnetic clustering* algorithm devised by Blatt *et al* [28]. It is based on a q-state Potts Hamiltonian, and allows, for the first time, for the identification of fuzzy communities.
- The Capocci–Servedio–Colaiori–Caldarelli method [29]. This algorithm combines the use of spectral properties (which are nicely reviewed and generalized to study

[4] The goodness of a given division (or division method) can be decided in *absolute* terms (when the underlying community structure is known, as for example, in computer-generated networks) or in *relative* terms (when the community structure is not known, but it maybe quantified in terms of modularity or similar measurements [14, 15, 17]; large modularity values correspond to better divisions).

different kinds of networks—for instance, directed ones) with the use of correlation measurements to determine community closeness.

- The Fortunato–Latora–Marchiori method [30]. This is a variation of the method by Girvan and Newman, in which the betweenness is replaced by the alternative concept of *information centrality*, as a way to measure edge centrality. The method generates good results but its performance (N^4 for a sparse graph) is rather poor.

Apart from these techniques recently introduced in the field of complex networks, many other algorithms have been developed mainly in the context of computer science. Most of them employ spectral analysis, which provides, in a very natural way (using the first non-trivial eigenmode) a tool for bi-partitioning [31] as will be illustrated within this paper. By iterative applications of bi-partitioning more elaborate divisions into communities or components can be achieved [9, 32, 33]. Alternatively, some other spectral methods employ more than one eigenmode leading directly to a splitting [16, 34, 35].

Without neglecting any of these algorithms, which can be applicable depending on the situation under consideration, this paper introduces yet another new method, allowing for a systematic analysis and detection of communities. It combines the following features: (i) it generates good results in all the cases tested; (ii) it is relatively fast, as compared with methods providing comparable results; (iii) it includes a way to optimize the output, as will be explained in what follows.

The method proposed in this paper combines spectral methods with clustering techniques, and uses the concept of modularity in order to develop a working algorithm. More precisely, the main lines of the algorithm are as follows: spectral analysis of the Laplacian matrix allows us to project the network nodes into an *eigenvector space* of variable (tunable) dimensionality. Afterwards, a *metric* is introduced in various possible fashions and then, finally, by applying standard clustering techniques a *dendrogram* [6] is built up. The modularity of possible groupings (sections of the dendrogram) is maximized for every dimension considered for the eigenvector space and, finally, the global maximum over all possible number of eigenvectors (i.e. dimensions of the space) is found.

In the forthcoming sections we review some basic ideas and definitions of spectral analysis and we introduce our algorithm step by step. Then we apply it to different workbench networks, comparing its performance with that of other existing methods, and, finally, the conclusions are presented.

2. Using the Laplacian eigenvectors to detect communities

2.1. Spectral analysis: Laplacian eigenvectors

The topology of a network with N vertices can be expressed through a symmetric $N \times N$ matrix $\boldsymbol{L}$, the *Laplacian* matrix [36]. The diagonal elements L_{ii} are given by the degree k_i of the corresponding vertex i, while off-diagonal elements L_{ij} are equal to -1 if an edge between the corresponding vertices i and j exists and 0 otherwise. The sum of elements over every fixed row or column is, trivially, equal to zero. Therefore, a 'constant vector' (with all its components taking the same value) is an eigenvector with eigenvalue 0.

Furthermore, since the quadratic form

$$\sum_{i,j=1}^{n} L_{ij} x_i x_j$$

can be written as

$$\sum_{\text{links}} (x_i - x_j)^2,$$

which is positive semidefinite, the eigenvalues of $\boldsymbol{L}$ are either zero or positive [37]. The use of other matrices, employed to study network spectral properties, has been recently considered in [16, 29, 33].

If the graph under analysis is connected, there is only one zero eigenvalue corresponding to a constant eigenvector. In contrast, for non-connected graphs (composed of m connected components) the Laplacian matrix is block diagonal. Each block is the Laplacian of a subgraph and it admits a constant eigenvector with eigenvalue 0. Therefore, the Laplacian of the whole graph has m degenerate eigenvectors (corresponding to eigenvalue zero), each of them having non-zero constant components for nodes in the associated subgraph and 0 in the rest.

If the subgraphs are not fully disconnected but, instead, a few links exist between them, the degeneracy disappears. This leaves only one trivial eigenvector with eigenvalue 0 and $m-1$ approximate linear combinations of the old ones with slightly non-vanishing eigenvalues [20, 29]. As the Laplacian matrix is real symmetric, with orthogonal eigenvectors, and since the first of them has equal components, all the others must have components whose total sum vanishes. In order to illustrate how these ideas can be applied to identify communities, let us take, as a particular example, the number of subgraphs to be 2. In this case, the components of the second (first non-trivial) eigenvector are positive for one subgraph and have to be negative for the other, providing a clear-cut criterion for bisecting the graph [31]. If the two subgraphs are not very well separated, then this distinction between positive and negative values becomes fuzzier. In such cases, more elaborate criteria for deciding how to effect the separation into two subgroups have been profusely studied in the specialized literature. Some of them optimize purposely defined quantities such as the *normalized cut* [33] and the *conductance* [32], which are defined as functions of the number of links that exist between the two components and their sizes[5]. By iterating successive bisections, techniques for obtaining more elaborate splittings can be constructed [9, 32, 33].

An alternative strategy is to assume that if there are more than two weakly connected blocks it should be somehow possible to find them all by inspecting the eigenvalue spectrum more accurately, instead of considering just the first non-trivial eigenmode [34, 35]. Let us explore this idea, which is the one that we will exploit, in more detail. Figure 1(a) shows the components of the first non-trivial eigenvector of a computer-generated graph including four communities, each composed by 32 nodes (see forthcoming sections for details). The group structure is clear, even if the two communities at the bottom are very near to each other and some nodes could be misclassified. In other

[5] In principle, the minimization of the conductance or the normalized cut among all possible splits is an NP-hard problem. However, it can be shown that the cuts based on the components of the second eigenvector of the Laplacian or some related matrix give a guaranteed approximation to the optimal cut [35, 38].

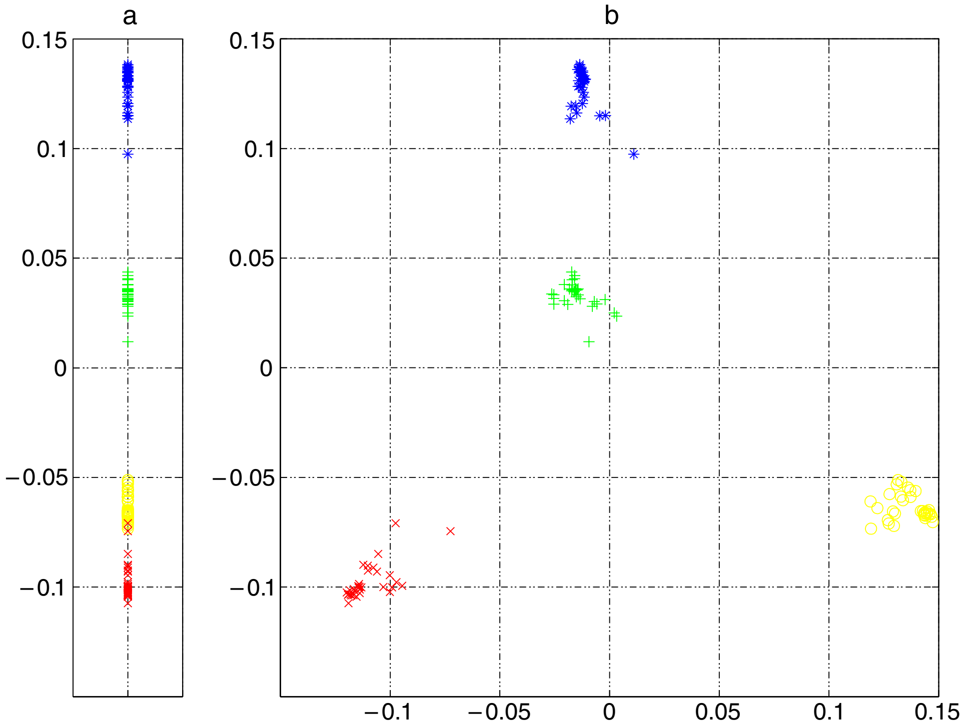

Figure 1. (a) Components of the first non-trivial eigenvector for a computer-generated network with four communities (see the main text). Two communities are clearly identified while the other two overlap. (b) All communities can be clearly identified when the components of the second eigenvector are plotted versus those of the first one—i.e. when the dimensionality of the eigenvector space is enlarged.

examples, with a number of inter-group connections larger than here, the communities become more entangled, and the prospects for extracting clear-cut subdivisions using this type of one-dimensional plot worsens. This difficulty can be circumvented by taking into account some more eigenvectors, i.e. by enlarging the projection space. This is illustrated in figure 1(b), where the nodes of the same graph are plotted using the components of the first two non-trivial eigenvectors as coordinates. Simple inspection by eye shows that all communities are distinctly separated now. Actually, using three eigenvectors the nodes of the different groups fall around the vertices of a (slightly distorted) tetrahedron, with some further improvement in inter-community separation. Generalizing this idea, *each vertex in the graph is represented by a point in a D-dimensional space in which the coordinates are given by its projections on the first D non-trivial eigenvectors.*

2.2. Introducing a metric

Aimed at turning 'eye inspection' of communities into a more quantitative measure, the explicit introduction of a metric (or similarity measure) is required. The most straightforward choice would be the Euclidean distance. However, this is *not* the only possibility; another one is to consider the *angular distance*, defined as the angle between the vectors joining the origin of the D-dimensional space with the two points under consideration. This possibility is inspired by empirical observations: loosely connected nodes could be quite ‘Euclideanly’ far from each other within a community, but still lying in the same ‘direction’ in the eigenvector space[6]. Moreover, when networks are large, nodes in the same community form a roughly one-dimensional ‘bundle’ (see for example figure 3 in [8]). Note also that using angular distances is tantamount to normalizing the position vectors in the corresponding space and then measuring the Euclidean distance, similarly to what was proposed in [35]. As will be shown, the angular metric generates, as a matter of fact, better results than the Euclidean one.

2.3. Cluster analysis

Having introduced a way to measure distances in the eigenvector space, a method for grouping nodes into communities is required. Such a method is provided by standard clustering techniques [18]—for example, *hierarchical clustering*. Starting from N clusters, composed by individual nodes, the two closest ones are iteratively joined together. In order to define cluster-to-cluster distance or ‘closeness’ (for a given metric), different criteria can be employed, generating among others the following clustering algorithms [18]:

- All possible pairs of nodes, taking one from each of the two clusters under examination, are considered. The minimum possible node-to-node distance is declared to be the cluster-to-cluster closeness. This leads to *single-linkage clustering*.
- Proceeding as before, but replacing the ‘minimum possible node-to-node distance’ between pairs by the ‘maximum’ one, *complete-linkage clustering* is defined.
- Another possibility consists in taking the average distance between all possible pairs. This leads to *group-average clustering*.
- A cluster is represented by a single point located at its ‘centre of mass’; the cluster-to-cluster distance is defined as the node-to-node distance between these two points. This leads to *centroid clustering*.

All these criteria have been broadly studied and applied. None of them can be proved to be generically more efficient than the others. In particular, the single-linkage method, being very simple, can be useful for analysing large data sets, and possesses some further mathematical advantages [18]. On the other hand, it has a tendency to cluster together, at a relatively low level, distant nodes linked successively by a series of intermediates. This is usually called the *chaining property*, which constitutes in some cases a serious drawback.

On the other hand, a convenient advantage of both single-linkage and complete-linkage clusterings is that only the ordering of the similarity measure is important: every other metric which produces the same ordering of distances leads to the same results.

The output of these algorithms can be represented by a hierarchical tree, usually called a *dendrogram*. The starting single-node communities are the branch tips of such a tree, which are repeatedly joined until the whole network has been reconstructed as a single component (see, for instance, figure 2 in [6]). Each level of the tree represents

[6] The attentive reader could argue that figure 1(b) provides a counterexample to this general assertion; i.e. the two uppermost groups are nearby angularly but far apart Euclideanly. Indeed, taking the three-dimensional version of the net analysed in such a figure, the four communities lie within the main directions on a tetrahedron, circumventing this apparent contradiction.

a possible splitting of the network into a set of communities, obtained by halting the clustering process at the corresponding level. However, the clustering algorithm gives no hint about the 'goodness' of such a partition.

2.4. Modularity

In order to quantify the validity of possible subdivisions (obtained as explained above) and to optimize the chosen splitting, we use, following [14, 15], the concept of *modularity*. It is defined as follows: given a network division, let e_{ii} be the fraction of edges in the network between any two vertices in the subgroup i, and a_i the total fraction of edges with one vertex in group i (where edges 'internal' to each group have weight 1 while inter-group links have weight 1/2). The modularity, Q, is then defined as

$$Q = \sum_i (e_{ii} - a_i)^2. \qquad (1)$$

It measures the fraction of edges that fall between communities minus the expected value of same quantity in a random graph with the same community division.

The maximization of modularity has been proposed as a possible way to detect communities; since a full maximization is not possible in practice (the algorithm would take an amount of time exponential in the number of nodes to explore all possible splittings) an approximate algorithm has been suggested [15]. In our case, modularity measurements are simply used to find the best splitting among all the possible partitions of the dendrogram obtained following the previous steps [14].

Other indices quantifying the quality of splittings have also been proposed in the literature. Some of them are the 'conductance', the 'performance' and the 'coverage', to name but a few (see [16] and references therein for more details). None of these, taken by itself, provides a fully useful criterion; they have to be combined somehow. It seems that the modularity is a better, more efficient, choice.

2.5. Implementing a functioning algorithm

Summarizing the ideas introduced in the previous subsections, our algorithm can be synthesized and implemented to build up a functioning algorithm as follows. First *a few* eigenvalues and eigenvectors of the network Laplacian matrix are computed. The question of what 'a few' means will be tackled afterwards. Since the Laplacian is usually a sparse matrix and not all eigenpairs are required (that will require a time N^3), the relatively fast Lanczos method [39] can be employed. Nonetheless, the eigenvector computation is still the most computationally expensive step of the algorithm.

For any given number D of eigenvectors (i.e. for a fixed dimension of the space) a similarity measure (or metric) is chosen, providing a basis for applying one of the previously introduced clustering techniques. Typically, Euclidean or, better, angular distances are employed.

Among the various hierarchical clustering methods available, we test single-linkage and complete-linkage clustering algorithms. These two have the advantage that no new distances have to be calculated during cluster formation: when two subgroups merge to form a larger one, its distance to any other cluster is given by the shortest (single-linkage) or by the largest (complete-linkage) of the distances from the two original components.

As said before, the single-linkage approach performs poorly in many cases owing to the previously discussed 'chaining' property, converting complete linkage into the preferential choice. Other linkage methods will be explored in the future; in particular, group-average linkage could be suitable when studying tree-like graphs [40].

An important difference between the way we apply clustering techniques and other standard applications is that we know in advance the underlying network structure. Using this knowledge, we implement the constraint that *two clusters are susceptible to merging only if there exists a link between them in the original network*.

At every step of the clustering process the modularity is computed. Once the whole dendrogram is completed, the splitting with the maximum modularity is chosen as the output for the corresponding D.

The optimal value of D to be taken is not known *a priori*, but since the eigenvalue calculation is the slowest part of the algorithm, we can repeat the hierarchical clustering using all possible values of D, and look for the largest value of the modularity. Typically the curve for the largest modularity versus D exhibits a maximum whose corresponding splitting provides the algorithm final output. If, instead, the curve keeps on growing up to the largest D, the number of computed eigenpairs has to be enlarged in order to extend the range of the curve, until a clear-cut maximum is pinpointed.

3. Tests of the method

3.1. Artificial community networks

To prove the algorithm we first test it on computer-generated random graphs with a well-known pre-determined community structure [6]. Each graph has $N = 128$ nodes divided into four communities of 32 nodes each. Edges between two nodes are introduced with different probabilities depending on whether the two nodes belong to the same group or not: every node has k_{in} links on average to its fellows in the same community, and k_{out} links to the outer-world, keeping $k_{\mathrm{in}} + k_{\mathrm{out}} = 16$.

In figures 2 and 3 we plot the modularity corresponding to the best splitting identified by the algorithm normalized by that of the known answer, and the average number of correctly classified vertices, respectively. Data for both Euclidean and angular measures, and both single-linkage and complete-linkage algorithms, are shown. The number of eigenvalues leading to the largest modularity is between 3 and 5 for the angular distance, and between 2 and 4 for the Euclidean one. Let us remark that these are roughly equal to the number of communities and that the performance is much better using the angular distance.

Summing up: on these computer-generated networks, our algorithm (equipped with the angular distance and complete linkage) generates excellent results as compared with other methods (see, for instance, figure 1 in [15] and figure 3 in [30]).

3.2. Zachary karate club

Now we consider the well-known karate club friendship network studied by Zachary [41], which has become a commonly used workbench for community-finding algorithm testing [6, 14, 15, 17, 26, 27, 30].

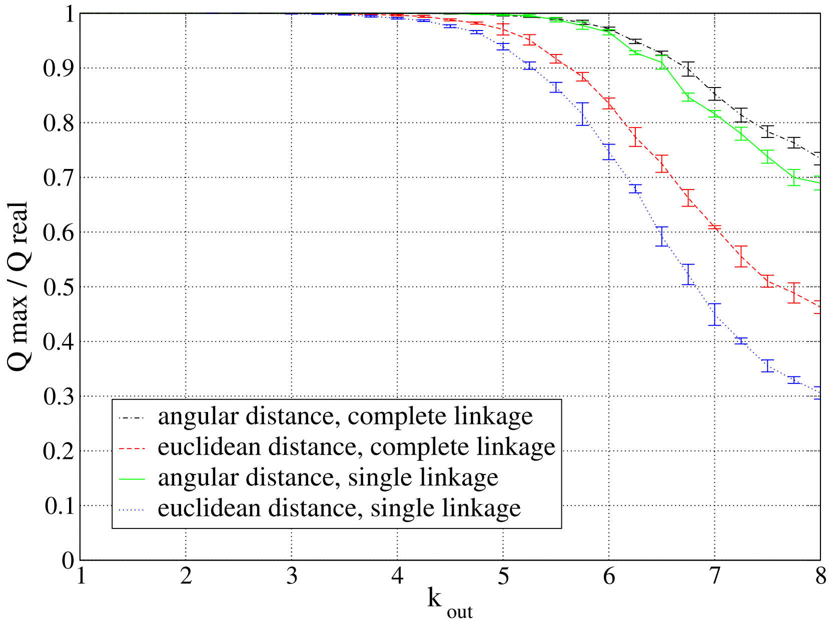

Figure 2. The maximum modularity found by the algorithm, divided by that of the known splitting of a computer-generated random graph (see the main text); the average over 200 graphs is plotted as a function of k_{out}.

Table 1. Modularity of the best splitting of the Zachary club network obtained for different metrics and clustering algorithms.

	Angular	Euclidean
Single linkage	0.412	0.319
Complete linkage	0.412	0.368

Table 1 shows the maximum modularity found by the algorithm: the best value is again obtained using angular distances combined with either single-linkage or complete-linkage clustering.

The best splitting is shown in figure 4; it is different from the ‘actual’ breakdown of the club—i.e. the two groups reported by Zachary are further subdivided. Let us stress the presence of a single-node community (node 12), and the fact that the modularity value of this splitting is larger than Zachary’s one (0.371), and larger than the ones found using other methods [15, 27, 30].

In this case single-linkage and complete-linkage approaches give the same best splitting. Nevertheless, the hierarchical structures given by the dendrograms in the two cases are quite different. Figure 5 shows how clusters merge after the best splitting is identified, as well as the modularity value corresponding to each division. For complete linkage the modularity value remains close to the best one until the whole network is merged in one community. On the other hand, for single linkage it falls rather abruptly right after the first merging, owing to the chaining problem. Moreover, in the former case, the two Zachary communities are first reconstructed and then joined together, while in

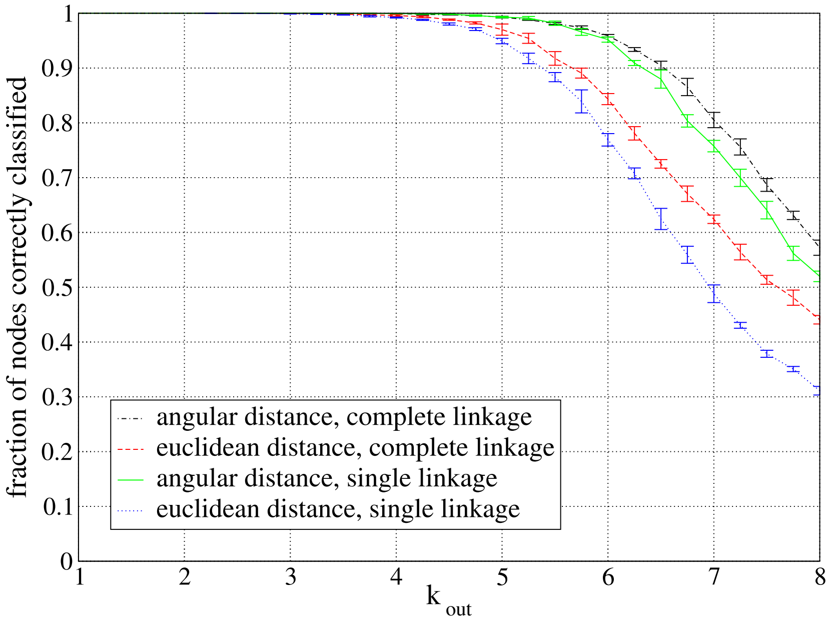

Figure 3. The fraction of nodes of computer-generated random graphs correctly identified by the algorithm, averaged over 200 graphs, as a function of k_{out}.

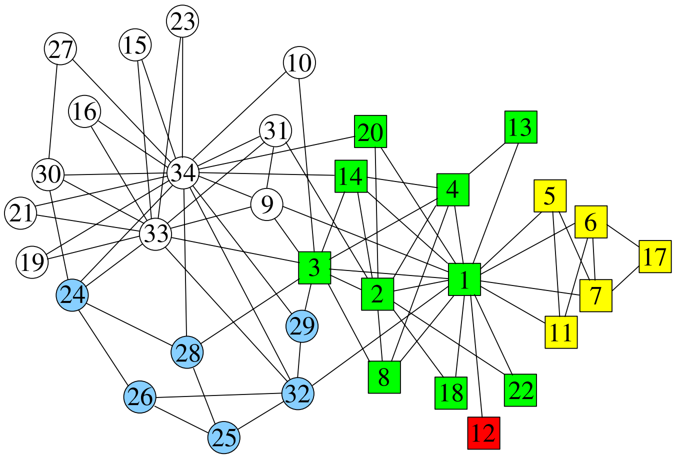

Figure 4. Splitting of the Zachary club network. Squares and circles indicate the two communities observed by Zachary; colours denote the further subdivision found by our algorithm.

the latter the merging proceeds differently. Therefore, even if the best splittings are the same in the two cases, complete linkage produces a more reliable dendrogram, describing more accurately the hierarchical structure.

3.3. Scientific collaboration networks

In order to test the method performance on larger networks we consider two scientific collaboration networks first analysed by Newman [42]. The vertices are the authors of the

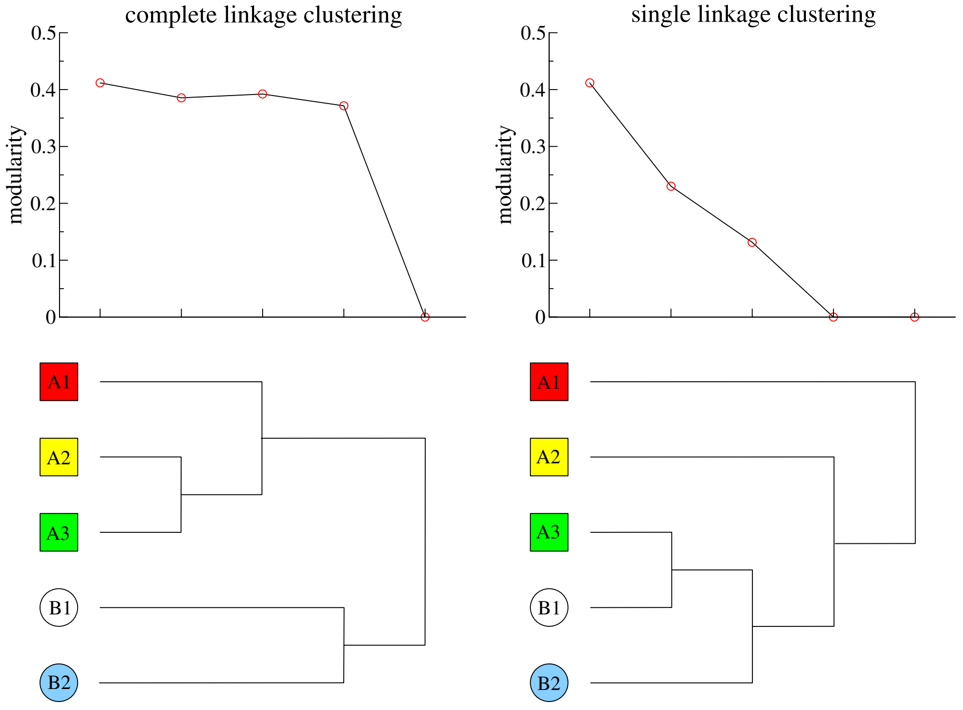

Figure 5. Comparison between the final part of the dendrogram for the Zachary club, obtained by using complete-linkage and single-linkage clustering (bottom), together with the corresponding modularity values (top).

papers that appeared in the `cond-mat` and `hep-th` archives at `ArXiv.org` between 1995 and 1999. Two authors are linked if they have co-authored a paper.

The `cond-mat` network contains 16 726 nodes, but we focus on its largest connected component, which contains only 13 861 authors. The computation of the first 1000 eigenvectors takes about two hours on a personal computer. The modularity curve computation, calculated using up to 999 eigenvectors, lasts around 15 min. Results for angular distance and complete-linkage clustering are plotted in figure 6. The largest value of the modularity, $Q = 0.736$, achieved for a splitting in 229 communities, corresponds to a 602-dimensional space. Obviously, we cannot compare the final splitting with a ‘true’ one, which is not defined. As the curve in figure 6 is rather flat in its tail, one can legitimately wonder how the best splitting compares to other ones obtained for similar dimensionalities. This question is difficult to answer in a rigorous way, and deserves further analysis, which will eventually lead to a functional definition of the *community structure robustness*.

Analogously, the `hep-th` network has 8361 authors with a connected component of 5835. The largest modularity value, $Q = 0.707$, is produced by a division into 114 communities, obtained using 416 eigenvectors. The computation of the first 1000 eigenvectors takes around 30 min and the search for the largest modularity value about 8 min. In this case, the initial number of eigenvectors could have been taken much smaller than 1000, without affecting the final output, with the consequent time saving.

As in previous cases, the number of eigenvectors used to produce the best splitting is of the order of magnitude of the number of communities found. In these cases, comparison

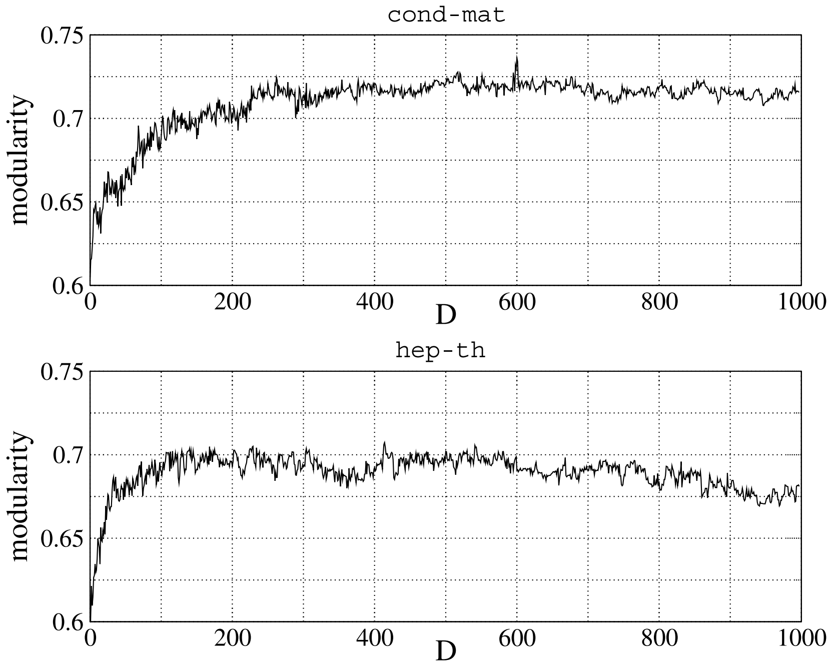

Figure 6. The maximum modularity as a function of the number of eigenvectors for the `cond-mat` (top) and `hep-th` (bottom) networks.

with previous community studies is not feasible, as modularity measurements have not been (to the best of our knowledge) reported in the literature.

4. Conclusions

We have introduced a new algorithm aimed at detecting community structure in complex networks in an efficient and systematic way. The method combines spectral techniques, cluster analysis and the recently introduced concept of modularity.

The nodes of the network are projected into a D-dimensional space, where D is the number of first non-trivial eigenvectors of the Laplacian matrix; their coordinates are the node projections on each eigenvector. Then a metric (either Euclidean or angular) is introduced in such an eigenvector space. Once distances are computed, standard hierarchical clustering techniques (for instance, complete-linkage clustering) are employed to generate a dendrogram. The subdivision of this dendrogram giving the maximum modularity is taken as the output of the algorithm for a fixed D. Then, D is also allowed to vary (from 1 to some arbitrary, maximum value) providing a way to maximize the modularity and enhance the performance of the method.

The best results are obtained using the angular distance and complete-linkage clustering; however, other kinds of distances, other clustering algorithms or even other means to quantify the goodness of a division could be used to improve the results. In this sense our algorithm is a 'block-modular' one: modifications of any of its ingredients could possibly lead to an overall improvement.

While spectral methods have been profusely used before to analyse similar problems, we believe that our algorithm represents a step forward in studying complex-network communities, as it combines spectral techniques with (i) the novel concept of modularity, which provides a very adequate estimate of the quality of a given splitting, and (ii) a way to optimize the number of eigenmodes taken into consideration.

The weakest part of the method is that the maximum number of eigenvectors to be computed in order to find the one generating the maximum modularity is not known *a priori*. The calculation of eigenvectors being the slowest part of the algorithm, what we do is take a reasonable number of them and, afterwards, verify that the maximum-modularity curve as a function of D decreases at its tail; i.e. we make sure that a maximum of the modularity function is located. If this is not the case, the number of eigenvectors needs to be enlarged, at the cost of higher computational effort. In the absence of a general criterion for establishing the monotonicity of the modularity curve, the only possible way to decide whether the identified local maximum is the global one would be to compute all possible eigenvalues. In practice, in all the cases studied, the best splitting is found with a relatively small number of eigenvectors, converting the algorithm into a reliable, relatively fast and very efficient one.

An open challenge is identifying a systematic criterion for estimating, *a priori*, what the order of magnitude of the number of eigenvalues to be computed is, to further optimize the output and efficiency.

We hope that this new algorithm will be employed with success in the search and study of communities in complex networks, and will help to uncover new interesting properties.

Acknowledgments

We acknowledge useful comments and discussions with F Colaiori, A Capocci, V Servedio, A Arenas, G Caldarelli and J Torres. We are especially grateful to M Newman for providing us with the data on scientific collaborations as well as for a reading of the manuscript, and to C Castellano for very helpful comments and suggestions. Financial support from the Spanish MCyT (FEDER) under project BFM2001-2841 and the EU COSIN project IST2001-33555 is acknowledged.

References

[1] Strogatz S H, 2001 *Nature* **410** 268
[2] Albert R and Barabási A L, 2002 *Rev. Mod. Phys.* **74** 47
[3] Dorogovtsev S N and Mendes J F F, 2003 *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford: Oxford University Press)
[4] Pastor Satorras R and Vespignani A, 2004 *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge: Cambridge University Press)
[5] Newman M E J, 2003 *SIAM Rev.* **45** 167
[6] Girvan M and Newman M E J, 2002 *Proc. Nat. Acad. Sci.* **99** 7821
[7] Guimerá R, Sales-Pardo M and Amaral L A N, 2004 *Preprint* cond-mat/0403660
[8] Eriksen K A, Simonsen I, Maslov S and Sneppen K, 2003 *Phys. Rev. Lett.* **90** 148701
Eriksen K A, Simonsen I, Maslov S and Sneppen K, 2003 *Preprint* cond-mat/0312476
[9] Borgs C, Chayes J T, Mahdian M and Saberi A, 2004 *Proc. 10th ACM SIGKDD Int. Conf. on Knowledge, Discovery and Data Mining*
[10] Guimerá R, Danon L, Diaz-Guilera A, Giralt F and Arenas A, 2003 *Phys. Rev.* E **68** 065103(R)
Arenas A, Danon L, Diaz-Guilera A, Gleiser P M and Guimerá R, 2003 *Preprint* cond-mat/0312040
[11] Hartwell L H, Hopfield J J, Leibler S and Murray A W, 1999 *Nature* **402** C47
Ravasz E, Somera A L, Mongru D A, Olvai Z N and Barabási A L, 2002 *Science* **297** 1551
[12] Egghe L and Rousseau R, 1990 *Introduction to Informetrics* (Amsterdam: Elsevier)
[13] Flake G W, Lawrence S, Giles C L and Coetzee F, 2002 *IEEE Comput.* **35** 66

[14] Newman M E J and Girvan M, 2004 *Phys. Rev.* E **69** 026113
[15] Newman M E J, 2004 *Phys. Rev.* E **69** 066133
[16] Brandes U, Gaertler M and Wagner D, *ESA'03: Proc. 11th European Symp. Algorithms* (*LNCS* vol 2832) pp 568–79
[17] Radicchi F, Castellano C, Cecconi F, Loreto V and Parisi D, 2004 *Proc. Nat. Acad. Sci.* **101** 2658
[18] Jain A K and Dubes R C, 1988 *Algorithms for Clustering Data* (Englewood Cliffs, NJ: Prentice-Hall)
Everitt B S, 1993 *Cluster Analysis* (London: Edward Arnold)
[19] Domany E, 1999 *Physica* A **263** 158
[20] Newman M E J, 2004 *Eur. Phys. J.* B **38** 321
[21] Weiss Y, 1999 *Int. Conf. on Computer Vision, Proc. IEEE* p 975
[22] Duda R O and Hart P E, 1973 *Pattern Classification and Scene Analysis* (New York: Wiley)
[23] Fukunaga K, 1990 *Introduction to Statistical Pattern Recognition* (San Diego, CA: Academic)
[24] Freeman L, 1977 *Sociometry* **40** 35
[25] Tyler J R, Wilkinson D M and Huberman B A, 2003 *Proc. 1st Int. Conf. on Communities and Technologies* ed M Huysman, E Wenger and V Wulf (Dordrecht: Kluwer)
Wilkinson D and Huberman B A, 2004 *Proc. Nat. Acad. Sci.* **101** 5241
[26] Wu F and Huberman B A, 2004 *Eur. Phys. J.* B **38** 331
[27] Reichardt J and Bornholdt S, 2004 *Preprint* cond-mat/0402349
[28] Blatt M, Wiseman S and Domany E, 1996 *Phys. Rev. Lett.* **76** 3251
Blatt M, Wiseman S and Domany E, 1997 *Neural Comput.* **9** 1805
[29] Capocci A, Servedio V, Colaiori F and Caldarelli G, 2004 *Preprint* cond-mat/0402499
[30] Fortunato S, Latora V and Marchiori M, 2004 *Preprint* cond-mat/0402522
[31] Fiedler M, 1973 *Czech. Math. J.* **23** 298
Pothen A, Simon H and Liou K-P, 1990 *SIAM J. Matrix Anal. Appl.* **11** 430
[32] Kannan R, Vempala S and Vetta A, 2004 *J. ACM* **51** 497
[33] He X, Ding C H Q, Zha H and Simon H D, 2001 *Proc. IEEE Int. Conf. on Data Mining* p 195
Ding C H Q, He X and Zha H, 2001 *Proc. 7th Int. Conf. on Knowledge Discovery and Data Mining (KDD 2001)* p 275
[34] Kleinberg J M, 1999 *J. ACM* **46** 604
Gibson D, Kleinberg J M and Raghavan P, 1998 *Proc. 9th ACM Conf. on Hypertext and Hypermedia* p 225
[35] Ng A Y, Jordan M I and Weiss Y, 2002 *Adv. Neural Inf. Process. Syst.* **14** 849
[36] Biggs N L, 1974 *Algebraic Graph Theory* (Cambridge: Cambridge University Press)
[37] Mohar B, *The Laplacian spectrum of graphs*, 1991 *Graph Theory, Combinatorics, and Applications* ed Y Alavi, G Chartrand, O R Ollermann and A J Schwenk (New York: Wiley) pp 871–98
[38] Chung F, 1997 *Spectral Graph Theory* (*CBMS Region Conference Series in Mathematics* vol 92) (Providence, RI: American Mathematical Society)
[39] Golub G H and Van Loan C F, 1996 *Matrix Computations* (Baltimore, MD: Johns Hopkins University Press)
[40] Newman M, 2004 private communication
[41] Zachary W W, 1977 *J. Anthropol. Res.* **33** 452
[42] Newman M E J, *The structure of scientific collaboration networks*, 2001 *Proc. Nat. Acad. Sci.* **98** 404
See also Newman M E J, 2001 *Phys. Rev.* E **64** 016132