COMMUNITIES IN COMPLEX NETWORKS: IDENTIFICATION AT DIFFERENT LEVELS

Alex Arenas, Jordi Duch and **Sergi Gómez**
Departament Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Spain

Leon Danon
Mathematics Institute, University of Warwick, Great Britain

Albert Díaz-Guilera
Departament Física Fonamental, Universitat de Barcelona, Spain

Keywords: Communities, hierarchies, overlap, dynamics

Contents

Summary

We present here and compare the most common approaches to community structure identification in terms of sensitivity and computational cost. The work is intended as an introduction as well as a proposal for a standard benchmark test of community detection methods.

1. Introduction

The analysis of complex networks has received a vast amount of attention from the scientific community during the last decade. Statistical physicists in

particular have become interested in the study of networks describing the topologies of a wide variety of systems, from biological technological or social networks. Although several questions have been addressed (see the review paper by Costa et al. for a complete set of measurements), many important ones still resist complete resolution. One such problem is the analysis of modular structure found in many networks. Distinct modules or communities within networks can loosely be defined as subsets of nodes which are more densely linked, when compared to the rest of the network. Such communities, as usually called in social sciences, have been observed, using some of the methods we shall go on to describe, in many different contexts, including biological networks, economic networks and most notably social networks. As a result, the problem of identification of communities has been the focus of many recent efforts. As a concrete example we show in Figure 1 the network representing the Spanish research community of Statistical and Nonlinear Physicists (FISES, http://www.fises.es).

Figure 1. FisEs network. Network of scientists that contributed to the Statistical Physics (Física Estadística) conferences in Spain. We consider two scientists linked if they have co-authored a panel contribution to any of the conferences. To be able to consider the historical structure of this network we "accumulate" the network over all the conferences, that is, once a link is created, it remains, even if the authors never collaborated again. The final network (accumulated over all the years) is comprised of 784 nodes with 655 (84%) of those belonging to the giant component. Green nodes denote the member of the scientific committees.

Nodes belonging to the same community are more than likely to have other properties in common and hence community detection in large networks is potentially very useful for instance when trying to understand dynamical properties. In the world wide web, community analysis has uncovered thematic clusters. In biochemical or neural networks, communities may be functional groups, and separating the network into such groups could simplify the functional analysis considerably.

The problem of community detection has been the subject of study in various disciplines. A simpler version of this problem, the graph bi-partitioning problem (GBP) has been the topic of study in the realm of computer science for decades. Here one looks to separate the graph into two equal-size communities, which are connected with the minimum number of links. This is indeed an NP complete problem; however several methods have been proposed to reduce the complexity of the task. In real networks one cannot

assume how many communities there are, but in general it is more than two. This makes the process much more costly.

Furthermore communities can be organized in hierarchies, meaning that different organizational levels can be simultaneously important and the question to the best partition has not a single answer. This hierarchical organization strongly affects the dynamical properties of networks. Another additional issue is that sometimes there is not a clear separation among communities and they present a certain degree of overlapping.

In this chapter we would like to present the recent advances made in the field of community identification in networks in a clear and simple fashion. To this end, the sections are organized as follows. In the next section we describe some ways to define communities in a network context. Following this, we present a method to evaluate a particular partition of a network. Then, we go on to describe the various recent methods starting with link removal methods, going on to agglomerative methods, followed by methods optimizing modularity and finally "other" methods. Some of the methods presented do not necessarily fit into just one of these classifications, and there may be some overlap. We finally introduce different structural organizations in networks and dynamical applications of modular networks.

2. Definitions of Communities 

There is not a unique definition of what a community is, instead the idea of communities is different and has been evolving depending on the field that defines it. The first definitions of community come from the field of social networks, where the communities are studied and understood according to the effect that an individual player has on the network and vice versa. Some of these ideas have been used and developed by some of the methods we present below, while new approaches have also been adopted from other fields such as physics or mathematics.

The different definitions of what is a community are all based in the concept of a subgraph, that is, groups of nodes and all the connections between them. The definitions can be classified into two main conceptual categories, those who use self-referral information and those based on comparative definitions.

Self referring definitions only use information of the structure of the network to decide what groups of nodes can be considered as a community. The most restricting and simple community structure is a clique, defined as a subgraph that is fully connected (i.e. it has all the possible edges between its nodes). Since this constraint is rarely fulfilled in real sparse networks, there are other approaches that relax it, such as n-cliques, n-clans and n-clubs. Self-referring definitions, while useful in characterizing communities, which are already known, are not the best choice while trying to find them since the methods to find the cliques in a network is very costly.

A second type of definitions use topological information to compare if a group of nodes is a community or not, for instance, counting how many links

have the nodes of the subgraph inside of it and how many links have them with nodes outside the subgraph. The strong definition of community requires that all the nodes of a community must have a larger number of links to members of the same community than to members of other communities. A lighter version of this definition is the weak definition of community proposed by Radicchi et al., where it is required that the sum of links inside the community is larger than the total number of links to the outside. This definition and some small variations of it is the most used in the majority of the methods that we will present later, since comparing the internal structure of a community to the external structure gives rise to a measure of how good a particular partition is.

3. Evaluating Community Identification 

Once a partition of the network into communities has been identified, the problem turns on to evaluate how good is the partition. Girvan and Newman proposed a simple approach, based on the intuitive idea of lack of community structure in random networks. Consider an arbitrary partition of a given network into N_c communities. We can define a $N_c \times N_c$ size matrix **e** where the elements e_{ij} represent the fraction of total links starting at a node in partition i and ending at a node in partition j. Then, the sum of any row of **e**, $a_i = \sum_j e_{ij}$ corresponds to the fraction of links connected to i.

If there is no community structure in the network the expected value of the fraction of links within partitions can be estimated. It is simply the probability that a link begins at a node in i, a_i, multiplied by the fraction of links that end at a node in i, a_i. Then the expected number of intra-community links is just $a_i a_i$. We also know that the *real* fraction of links exclusively within a partition is e_{ii}. Comparing the two and summing over all the partitions in the graph we get

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

This is a measure known as *modularity*. As an example, we can consider a network comprised of two disconnected components. If we then have two partitions, corresponding exactly to the two components, modularity will have a value of 1. For particularly "bad" partitions, for example, when all the nodes are in a community of their own, the value of modularity can take negative values.

It is tempting to think that random, Erdos-Renyi networks have little or no community structure. However, as Guimerà *et al.* showed, this in general is not the case. In fact, it is possible to find a partition which not only has a nonzero value of modularity for random networks of finite size, but that this value is quite high. For example a network of 128 nodes and 1024 links has a maximum modularity of 0.208. This suggests that community structure

appears in random networks due to fluctuations.

From here on we will look at different methods of community identification presented recently. First we consider methods based on link removal.

4. Link Removal Methods

Divisive methods extract the partition into communities of a network by removing some (or all) of its links until the network is no longer connected or we have a division into communities that meets certain requirements. However, to be able to obtain useful results we need to remove the appropriate links, otherwise the communities will be meaningful. Several methods have been proposed to identify the links that we should remove, which we will revise in this section.

4.1. Shortest Path Centrality

One of the first divisive methods presented in uses the idea of centrality, a measure of how central the node or link is in the network, to decide which links need to be removed. The algorithm uses a particular type of centrality, shortest path centrality, which measures the number of shortest paths between pairs of nodes that pass through a certain node or link. The links with the highest centrality usually act as a bridge between the communities, so if we remove them we can split the network into densely connected communities.

The method works recursively eliminating all the links of the network, and stops when there are no more links and all the nodes are isolated. Every time a link is removed, all the centralities are recalculated, otherwise we will obtain an erroneous community detection. This part of the algorithm is the one that requires most computer power and, for a network of size n with m links, using the fastest methods developed independently by Newman and Brandes the speed of calculating all link betweenness-es in one step still remains of $O(m^2 n)$ for unweighted networks. This limits the size of the graph that we can process in a reasonable time to a maximum of around 10000 nodes. Figure 2 shows the application of this algorithm to the network depicted in Figure 1.

Figure2. Binary tree showing the result of applying the Girvan-Newman algorithm and our visualization technique to the network of coauthors in FisEs. Each branch corresponds to a real community and the tips of the branches correspond to the people that have played a major role in the different research groups. One can identify here that the members of the scientific committees over the years have indeed played an important role in

the development of the community and that they are precisely quite central nodes in the respective local communities.

4.2. Extensions of the Shortest Path Centrality

The same authors of the previous method have also presented two alternative methods to detect community structure by betweenness centrality by calculating this value using two alternative approaches. However, although they are conceptually interesting, both approaches require higher computation than the previous method, and they do not improve the accuracy of it.

The first approach considers the network as a circuit, where links are assigned a unit resistance and we select two nodes that we define as unit voltage source and sink. Using Kirchoff's laws we can calculate the current flow between these two nodes. Adding the flows we will obtain a mesure similar to the centrality, where those links with the lowest resistance (shortest path) carry the most current and, therefore, are the most central. The second approach uses random walks to determine the betweenness centrality of the links. The network is used as a substrate for signals that perform a random walk between pairs of nodes. The link betweenness in this case is simply the rate of flow of random walkers through a particular link summed over all pairs of vertices.

4.3. Information Centrality

Another divisive algorithm available uses the network efficiency measure proposed by Latora and Marchiori. This measure quantifies how efficient is a network in the context of information exchange. If we remove links of the network, its efficiency decreases a certain amount of information centrality.

This method, presented by Fortunato et al., is based on the idea that if we remove the links that act as bridges between communities we should observe the largest drops in network efficiency. from this premise, the method operates similarly to the shortest path centrality method, removing recursively all the links and recalculating the efficiency of all the links at every step. The process is slower than the GN running at $O(n^4)$, but instead the accuracy obtained in the detection is better when the communities to be found are more diffuse.

4.4. Link Clustering

Another approach uses the idea that linked nodes belonging to the same community should have a high clustering coefficient, that is, they share larger number of common neighbors. Based on this idea, the algorithm of Radicchi et al. postulates that the proportion of possible number of loops that go through internal links should be much larger than the proportion of loops for links pointing to outside of the community. The algorithm also works recursively as the previous ones, but in this case by recalculating the *link-clustering coefficient*, which measures the number of loops of a certain

length that pass through each link. Longer loops require more computer resources but provide more accurate results.

This algorithm provides a way to stop the detection process when a certain condition is fulfilled, instead of decomposing the whole network until all the nodes are separated. It is also faster than the previous ones, since to compute the *link-clustering coefficient* we only need local information. However, it is not very useful with networks with a very low clustering coefficient, such as trees, sparse graphs or disassortative networks, where we do not have the necessary loops to compute the *link-clustering coefficient*.

5. Agglomerative Methods 

Another approach to identify the communities of a network is to start from all the nodes being in separate communities, and some strategy to join or agglomerate them in larger groups. Here we present some of these methods and their grouping algorithms.

5.1. Hierarchical Clustering

Hierarchical clustering has been used traditionally in social networks analysis to extract the communities of the network. The idea of this method is based on the measurement of the similarity between the elements of the nodes according to some property. Starting from an empty network, the method selects those node (or groups of nodes) that have the highest similarity and joins them. This process is again repeated recursively until all the links are added or when we meet a certain condition. The method is very fast and it can work almost in linear time, being able to analyze networks that cannot be processed otherwise. However, the results are highly dependent on the similarity metric that is used to detect the communities.

5.2. L-Shell Method

A second approach focuses on identifying the community around one node of the network by agglomerating its neighbors until a condition is fulfilled. In particular, the algorithm consists on constructing a L-shell around one node, where a L-shell is a subset of the nodes with a maximum distance of the shortest path to the node origin is less or equal to L . The algorithm starts from the origin and adds more nodes by increasing the distance L until the emerging degree (number of links to nodes outside the L-shell) is lower than a cut-off value, and then it is stopped. Those nodes that fall inside the L-shell are grouped within one community.

This algorithm is particularly interesting when one is more interested in finding a single community and not in detecting the entire community structure. If we want to make the algorithm global, the authors suggest that we should repeat the process for each node, and then perform a statistical analysis of the results to detect the communities. Since the method uses local information, it is one of the fastest available.

5.3. K-Clique Method

Another approach introduced the idea that communities can overlap. In their definition of community, one node can belong to various "thematic" communities (i.e. one can belong to a scientific group, a family, a sports team, ...), which usually share a certain amount of nodes. The methodology to detect the overlapped communities is based on the concept of '*k*-clique communities'. A *k*-clique is a group of k nodes that is a complete subgraph, and a '*k*-clique community' is the union of all *k*-clique that are adjacent (two *k*-cliques are adjacent if they share k − 1 nodes).

In terms of accuracy, this method is not comparable with the others presented, since it uses a different definition of community structure. However, it has interesting applications, i.e. it can be used to observe the level of relationship between communities or to determine the communities where a certain node belongs.

6. Maximizing Modularity Methods 

Since the modularity measure introduced previously provides a good way to evaluate quantitatively a network partition into communities, many authors have presented methods that focus on optimizing this value to obtain the best partition. The benefit of all these methods is that they do not require extra information about the optimal number of communities, since there is a point where the modularity value cannot be improved further. On the contrary, the optimization process is not straightforward because the partition space of any graph (even relatively small ones) is extremely large. The following approaches present different methods to navigate the space of possible partitions to find the highest possible value of modularity, while balancing between accuracy and speed.

6.1. Greedy Algorithm

The first approach introduced by Newman optimizes the value of Q using a greedy algorithm. Starting from a configuration where each node corresponds to one community, the authors compute all the changes in modularity obtained by joining any possible pair of nodes. The highest increment is selected and the two communities are joined, and the process is repeated until a maximum value of Q is obtained.

This method is really fast, since the recalculation of the increments only uses local information, and can analyze a network in almost linear time. However, the accuracy achieved is the lowest of all the modularity optimizing methods.

The benchmark most commonly used to measure the sensitivity of a particular community identification algorithm does not take into account the fact that networks exhibit highly skewed community size distributions, as this shown in Fgiure 3, and is potentially flawed. By comparing the results of the greedy algorithm to results obtained from a modified version which takes community size into account, the present authors showed that size

heterogeneity can alter the comparative accuracy of community detection.

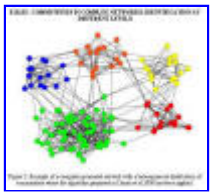

Figure 3. Example of a computer generated network with a heterogeneous distribution of communities where the algorithm proposed in Danon et al 2006 has been applied.

6.2. Extremal Optimization

A third approach, presented by Duch and Arenas, uses a different heuristic search procedure based on extremal optimization to find the best modularity value. The heuristic works at a local scale, by improving the contribution of each node to the global modularity. The nodes are assigned initially to two random partitions, and the local modularity optimization is performed by moving the nodes with the lowest local modularity from one group to another. When the optimization reaches an stationary state where the modularity cannot be improved anymore, the links between the two partitions are removed, and the process is repeated recursively while the total modularity keeps increasing. The algorithm is relatively fast, , scaling as $O\left(n^2 \log(n)\right)$, and it achieves the highest known modularity values for all networks studied.

A modified version of the algorithm has later been introduced by the same authors that adds two new improvements. First, it allows the analysis of weighted and directed networks, using new definitions of modularity. Second, the final results can be fine-tuned using a final bootstrap, which helps correcting small problems that appear due to the recursive process of the divisive part of the algorithm.

6.3. Simulated Annealing Methods

Another approach to optimize the modularity measure is to employ simulated annealing methods. This idea was introduced by Guimerà et al. when they studied modularity in random networks. The method starts with an initial random partition of the nodes into communities, and evolves randomly changing nodes from one community to another. The change is always accepted if the modularity increases, and with a certain probability otherwise. This is also repeated until the modularity cannot be improved anymore for a certain number of steps. The algorithm is slower than some of the other methods, but is one of the most accurate options available.

Later, Massen and Doye proposed two modifications of the simulated annealing approach. First, their algorithm stops periodically, analyzes all the possible node movements, and accepts the move that increases the modularity the most. Second, they use a Basin-Hopping approach, where in

each step a group of nodes are moved from one community to another, and this movement is accepted depending on the change of the modularity. The modifications make the process of maximization slower than the original, but are able to find even higher modularity values.

6.4. Information Theoretic Approach

One of the most recent approaches to the community detection problem is based on an information-theoretic framework, where the community detection problem is now treated as an information compression problem. The idea is to reduce the link connectivity of the network (the adjacency matrix) into a more simple description (a module assignment vector and a module matrix). To discover the configuration that provides the best "compression" of the network structure, they maximize the mutual information between the encoded and the global descriptions.

The results presented in their paper show that this method performs better than the others when detecting asymmetric communities. Another advantage is that changing the encoding function we can detect other types of clustering beyond the classical community structure. Similar to the mixture models, the method is also able to identify partitions where the nodes have similar patterns of connection to other nodes.

7. Spectral Analysis Methods 

An alternative to the adjacency matrix to represent the information of the connectivity of a graph is the Laplacian matrix. The position n_{ij} of the matrix informs about the existence of a link between i and j, and the diagonal contains the degree of node i, so that the sum of each row and column is equal to zero. The following methods use the algebraic properties of these matrices to identify the nodes that belong to each partition.

7.1. Spectral Bisection

The Laplacian matrix always has an eigenvector with eigenvalue 0, since the sum of elements over each row or column of the Laplacian matrix is equal to $\mathbf{0}$. Also, for each disconnected component of the graph, the Laplacian matrix has a degenerate eigenvector with its corresponding eigenvalue 0. If the components are not completely disconnected (i.e. there are some links between them), the degeneration is no longer present, and we obtain one eigenvalue with value zero and a few eigenvectors with an eigenvalue slightly greater from zero. Therefore, one method to find communities is to find the blocks that give the eigenvalues slightly greater than zero and looking at the components of their eigenvectors.

7.2. Multi Dimensional Spectral Analysis

Another different approach that also uses the properties of the Laplacian matrix was introduced by Donetti and Muñoz. The method consists in extracting the first few non-trivial eigenvectors using the Lanczos method,

which is very fast when applied to sparse matrices. They consider the values of the eigenvectors as coordinates in M-dimensional space, where M is corresponds to the number of non-trivial eigenvectors extracted. Finally, they measure the distances between the nodes in this space, and cluster them using hierarchical agglomerative methods, obtaining the desired partition into communities. The method is reasonably fast, but the results depend on how many vectors are extracted to separate the communities properly.

7.3. Constrained Optimization

Another method uses the information contained in the spectral properties of the simple adjacency matrix (instead of using the Laplacian as the previous ones). The authors use constrained optimization to extract the eigenvectors much faster, obtaining again a multidimensional space where the eigenvectors contain the coordinates of the nodes. To detect the groups that appear, they use a correlation of the average values of the eigenvectors to measure how close two nodes are in this space. Instead of providing a clear cut community structure, this method gives us an idea of how close any pair of nodes is in the context of communities. The method is able to obtain good results in mid size networks (thousands of nodes.

7.4. Approximate Resistance Networks

Wu et al. presented and extension of the resistance approach method presented before to reduce the time complexity of it. The idea is the same, they select a pair of nodes that act as voltage source and sink, and approximate the voltage of the rest of the nodes. However, instead of using the costly matrix inversion used by Newman, they use an iterative process to approximate the voltage of the other nodes. The accuracy of this approximation is dependent on how many times the iterative step is repeated. It is also dependent on having a good idea of the sizes of communities, which make it difficult to use it in large networks. However, this is one of the few methods that is able to identify the community around one node in linear time.

8. Other Methods 

In this last section we include all those methods that do not fit in any of the previous categories.

8.1. Clustering and Curvature

Eckmann and Moses, propose an alternative method based on the concept of curvature of a node. The curvature reflects the average distance between nodes, using the information of the average distance between neighbors of any node (Which is between 1 if they are directly connected and 2 if they do not have any other common neighbor). Since this value is directly related to the clustering, the authors show that finding the connected components that have high curvature gives good insights about the community structure of the graph. The authors have applied successfully this method to study

communities in e-mail networks.

8.2. Random Walk Based Methods

Zhou and collaborators have developed different methodologies for community detection based on random walks. Also worthy of note is that the method is applicable to both directed and undirected networks. They also define the concept of 'local' and 'global' community, using the information of attractor nodes (i.e. nodes that are the closest to its neighbors) and a set of rules.

The first approach uses the information contained in the adjacency matrix to determine algebraically the distance between two nodes, so they do not need to actually perform the random walk on the network. This way they obtain faster and more accurate results about which are the attractor nodes, and therefore, about what communities do we obtain.

Later, Zhou and Lipowsky present a different method based on biased random walks. In this modification, each walker has a higher probability to perform jumps from the source node to the node which shares the highest number of neighbors with the source (i.e. biasing the random walker to go down the link with the highest link clustering).

A third different method proposed by Latapy and Pons is based on the idea that a random walker will get trapped for a longer time in a densely connected community. They calculate a distance measure between two nodes, and apply an agglomerative method starting with all nodes in their own community, and joining them two by two.

8.3. Q-Potts Model

Another different approach detects communities by mapping the problem of community detection to the study of a spin system. The authors propose that if the system is in the ground state, communities are identified as groups with equal spin values. To identify the groups, they initialize each node with a random spin state between 1 and q, and determine the energy of the system using a q-Potts Hamiltonian. Then, the system is allowed to evolve using a simple Monte-Carlo method with simulated annealing until it reaches a stationary state.

An interesting feature of this approach is that allows the detection of 'fuzzy' communities, so we can know the level of overlapping between them. The method is reasonably fast, since the calculation of the Hamiltonian only uses local information, and its sensitivity is also good. However, the method needs the input of how many communities we want to find (appropriate values for q are also discussed in the paper).

9. Further Structural Complexity