Lecture 8: Communities and Global structure

Eduardo López

1 An overview of the situation

One of the key strengths of networks is that they capture large systems in an integrated way. But large systems tend to be an amalgam of many parts, some of which might only be marginally connected to other parts. In other circumstances, there may be a hierarchy of parts such that larger groups contain smaller ones within them. One might also encounter hybrid situations in which there are weak relations, hierarchies, and even parts that bridge and overlap other parts of the network.

A key question in the study of networks is how to use the recorded network structure to help identify these parts. This is not a trivial or even well defined task, but it is certainly an important. Parts are conventionally called communities, and constitute the expression of some concrete mechanism that establishes relations between individuals. Some of these relations can be family ties, work settings, church or community, etc.

As we shall learn in this lecture, there are various challenges associated with finding communities. In a review article discussing the state of the art in community detection [1], the Fortunato and Hric provide this illuminating comment about the grounding of the problem:

"Community detection in networks, also called graph or network clustering, is an ill-defined problem though. There is no universal definition of the objects that one should be looking for. Consequently, there are no clearcut guidelines on how to assess the performance of different algorithms and how to compare them with each other. On the one hand, such ambiguity leaves a lot of freedom to propose diverse approaches to the problem, which often depend on the specific research question and (or) the particular system at study. On the other hand, it has introduced a lot of noise into the field,

slowing down progress. In particular, it has favoured the diffusion of questionable concepts and convictions, on which a large number of methods are based."

I will not enter into this discussion here, nor will I try to advocate for the multiple points of view that relate to this topic. Instead, I will outline some of the most used and robust methods available, and allow you explore the vastness of this literature further if it is part of your larger set of interests.

2 Communities and their detection

The title of this section says something about the situation, consistent with what Fortunato and Hric point out. The process of community detection has a lot to do with the kind of community we are interested in finding. This makes it less useful to try to construct universal notions of community because such notions would then be undermined by the design of certain algorithms to detect those communities.

Instead, we will focus on *one particular notion of community, without trying to claim complete generality,* that conforms with one intuitive version of community. This particular kind of community is *assortative*, which basically means that it is constituted by nodes in a network that are more tightly related to each other than to the rest of the network. This is the kind of community we envision when thinking about a family or a work team.

A contrasting kind of community, about which we say very little, are disassortative communities, those in which items tend to *relate by rejection.* Thus, in word networks nouns and verbs behave in this way, where the need to form sentences requires that one relates a noun and a verb with each other so that it is more frequent to find nouns together with verbs than to find nouns with nouns and verbs with verbs. I will stop here.

In assortative communities, there have been many approaches proposed for their analysis. Many of the approaches have relied on the count of links within and outside a community to draw conclusions about the existence of such communities. However, link counting leads to some confusing situations that make such methods harder to use, and in many cases ill-defined.

To focus on less controversial and more effective methods, I present discussions on *stochastic block models [2], hierarchical clustering [3], and (briefly touch on) a random walk method [4].* But before this, we mention one complication that almost all methods have to contend with: the problem of the number of communities to find.

3 Number of communities and network cover

Most algorithms for finding communities in networks require that they be provided a specific number of communities to look for even before they are executed. At first glance, this would seem like an important shortcoming. However, we must think about the context of the problem to understand why this may be the case.

At the start of these notes, I mentioned that many large networks are the composed of various parts and that these parts are in some sense independent of other parts. Intuitively, this says that the mechanisms that bring together the various groups in a network, because they are independent from one another, also are likely to lack a cohesive behavior that makes all such groups equivalent. For instance, imagine a co-authorship network. Some groups of individuals in that network may form large communities of co-authors and furthermore, have very strong links among the co-authors as a consequence of writing many articles together. But the large network can also have smaller groups that generate less papers and thus appear in the network as a *different kind of community* in that the numbers that may characterize it would look very different than those that characterize the large community. Therefore, when we ask a general algorithm to solve the community detection problem on a network, we are asking it to *simultaneously* find very different thing. This leads to challenges in how the algorithms need to be designed and what they will look for.

As a related aside, one that I will not develop in greater detail, notice that in an implicit way, we have been thinking about communities as being non-overlapping. But in reality communities do overlap. A family unit may relate in its entirety to another family unit, for instance, because the children of the families go to the same school and/or play together. Coworkers can also be friends, play sports together, or even be neighbors. These and many other examples can be found to illustrate how communities can overlap. So how would one determine how many communities there are in such a setting. Does one count communities only up to a certain level? If one community is inside another, do they both count?

The selection of methods I present here tries in some way to address all of these possibilities, but there is no perfect method or perfect answer for the variety of situations one can encounter in real systems.

As a general comment about this problem, even if a method is able to decide by itself how many communities it needs to identify, any method that allows it should be executed with a number of communities already given as an input. This is because the methods invariably become better at

identifying the communities.

Now, having discussed this practicality about communities, I mention one more term that is useful to know. It is the concept of a *network cover*. This concept reflects the idea that once communities are identified, they *cover* the network. We can imagine that each community identified is characterized by a color. Since all nodes end up in a community, then the entire network becomes a patchwork of colors.

4 Stochastic Block Models

The basic principle of stochastic block models, easy to introduce in terms of assortative communities, is that nodes that belong to the same community are more likely to be connected to each other than to nodes outside of the community. Let us introduce the details.

Imagine that we consider a network to be covered by q communities. This means that any node i has an attribute or flag, g_i, that indicates to which of the q communities it belongs to. We label the communities $1, 2, \ldots, q$. One can then define probabilities $p_{r,s}$ that two nodes, one in community r and the other in community s are connected to one another. In terms of our notation, the probability can also be writte as

$$\Pr(\text{nodes } i \text{ and } j \text{ are connected}) = p_{g_i,g_j}. \tag{1}$$

The values of $p_{r,s}$, where $r, s = 1, \ldots, q$ represent the communities, specify the model. It is valid to have $r = s$, which just means that there is a probability for two nodes to belong to the same community. These probabilities form a matrix $\mathbf{P}$ of dimension $q \times q$, where the matrix element $\mathbf{P}_{r,s} = p_{r,s}$.

There are some important cases that one must consider in terms of possible models, encoded in the properties of $\mathbf{P}$.

4.1 Strong or weak communities

The choices of $p_{r,s}$ provide some general rules about what sorts of communities one can find. There are two situations:

1. When $p_{t,t} > p_{r,s}$ for all $r \neq s$ and r, s, t are community labels, any of the probabilities along the diagonal are larger than any of the off-diagonal elements of $\mathbf{P}$. In this case, these are strong communities

2. When $p_{t,t} > p_{t,s}$ with $s \neq t$ and s, t community labels, it is guaranteed only along each row of the matrix that the diagonal element is larger than the off-diagonal elements. In this case, the communities are weak.

The names associated with these communities are relevant because they reflect the consequences of the choices of parameter values. For strong communities, the expectations that arise from the probabilities are that each and every node inside a community has more links to other members of that community than to nodes outside of the community. In contrast, for weak communities the expectation is that *on average* the nodes in the community have more links to other nodes in the community than to nodes outside the community. This is a weaker situation because it allows some nodes to have less links to the community than to other nodes outside that community.

4.2 Model development

Regardless of the choice of model strength, what is the consequence of choosing to use $\mathbf{P}$? We can calculate some results now.

First, we must consider the consequences of the values of $p_{r,s}$. Note that the collection of values in $\mathbf{P}$ does not need to satisfy conditions such as normalization; the values of $\mathbf{P}_{r,s}$ only inform the result of independent attempts to create links between nodes of communities r and s. This does not even inform about *how many nodes belong to a community*. Therefore, to specify the model one also may need to make a choice about the sizes of communities or, equivalently, perform a set of *tentative* community assignments on the nodes so that for each i we provide a value to g_i.

For a given assignment g_i over all i, one can then calculate a set of quantities. First, the number of links that node i should have to nodes in its own community g_i is given by

$$k_i^{(int)} = p_{g_i,g_i} n_{g_i} \tag{2}$$

where n_{g_i} is the number of nodes that belong to community g_i; technically this formula should be $p_{g_i,g_i}(n_{g_i} - 1)$ so that one does not count a node trying to connect to itself, but for large networks this is not an issue (but remember this for small networks). This result can be understood very intuitively in the following way: a node i tries to connect with n_{g_i} other nodes, and the success rate of each attempt is p_{g_i,g_i}, leading to $p_{g_i,g_i} n_{g_i}$ successes. This is a binomial random variable.

The expected number of links inside the community that node i belongs

to is given by

$$m_{g_i,g_i} = p_{g_i,g_i}\binom{n_{g_i}}{2} \approx \frac{p_{g_i,g_i} n_{g_i}^2}{2} \tag{3}$$

where the second approximate equality applies in the large system limit. If we write this in terms of a given community identifier r, we have

$$k^{(int)}_{\text{community } r} = p_{r,r} n_r, \quad m_{r,r} = p_{r,r}\binom{n_r}{2} \approx \frac{p_{r,r} n_r^2}{2}. \tag{4}$$

The expected links from a node in community r to another node in community s are given by

$$k^{(ext)}_{\text{communities } r,s} = p_{r,s} n_s \tag{5}$$

and the overall number of links between communities r and s is

$$m_{r,s} = p_{r,s} n_r n_s. \tag{6}$$

In the case of strong communities, because $p_{r,r} > p_{r,s}$ (which is also the condition for weak communities), we have that

$$\frac{k_r^{(int)}}{k_{r,s}^{(ext)}} = \frac{p_{r,r} n_r}{p_{r,s} n_s} \Rightarrow \frac{k_r^{(int)} n_s}{k_{r,s}^{(ext)} n_r} = \frac{p_{r,r}}{p_{r,s}} > 1 \tag{7}$$

and therefore

$$\frac{k_r^{(int)}}{n_r} > \frac{k_{r,s}^{(ext)}}{n_s}. \tag{8}$$

In other words, the expected per node number of links that any node has to other members of its community is larger than the same expected number of links per node to other communities. This condition applies to the expectations of both strong and weak communities. A similar result applies to the overall number of links for communities, namely

$$\frac{m_{r,r}}{n_r} > \frac{1}{2}\frac{m_{r,s}}{n_s}. \tag{9}$$

When the communities are strong, variants of these results can be constructed as well, with only a minor increase in the expressions. But their meanings are what is more relevant. Thus, with regards to degree

$$\frac{k_r^{(int)}}{n_r} > \frac{k_{s,t}^{(ext)}}{n_t} \tag{10}$$

it means that the per node number of links inside of *any* community is larger than the per node number of links between *any two distinct* communities in the network. A similar interpretation can be drawn for the total number of links in a community, namely

$$\frac{m_{r,r}}{n_r^2} > \frac{m_{s,t}}{n_s n_t}. \tag{11}$$

Up to this point, we have not said anything about the role of n in this problem. For any assignment g_i of nodes to communities, it is clear that

$$\sum_{r=1}^{q} n_r = n, \tag{12}$$

which means that whatever the community assignment is, the sum of nodes over all the different communities has to be equal to n. This has implications with regards to the total number of links in the network. Thus, to determine the expected number of links m for the entire network, we need to count all expected links between all pairs of communities and add to that the internal numbers of links within a community. Therefore

$$m = \frac{1}{2}\sum_{r\neq s} p_{r,s} n_r n_s + \sum_r \frac{p_{r,r} n_r^2}{2} = \frac{1}{2}\sum_{r,s} p_{r,s} n_r n_s, \tag{13}$$

where in the last expression, s and r can be equal.

4.3 Model application

How would one apply this method in practice? This is where things can get complicated. Note that all the results above have been written under the assumption that the communities (and therefore their sizes) are known. But in fact learning the communities *is* the objective of the model. Furthermore, we have also made use of the probabilities $p_{r,s}$ as if they are known, but this is unlikely.

To a great extent, the application of this method requires further assumptions that allow us to formulate a model that we test against. If one knew that the network did fully satisfy the stochastic block model structure with at least some of the assumed parameters above known, all that would be needed would be to find the values of its remaining missing parameters. However, what one actually does in assortative community detection is *assume* that the stochastic block model applies but none of the parameters are known.

Without attempting to enter into the various practical approaches that can be taken to fully utilize stochastic block models for community detection, let us demonstrate in a rather crude way how the concepts can be deployed into a potential algorithm.

Consider that one may start by comparing an observed network to one in which *no community presence is assume*. In this case, all $p_{r,s} = p$ regardless of r, s. In fact, such a model is called an Erdős-Rényi random network model (a large topic in the lecture on random networks). Therefore, for any choice of node groupings g_i, the number of expected links between the groupings would be given by Eq. 6 with $p_{r,s} = p$, or

$$m_{r,s}^{(h)} = p n_r n_s, \tag{14}$$

where the superindex h stands for homogeneous network. This means that if community structure *were present* in the network and, just for argument's sake, the same communities s and r as chosen here were picked, the quotient of the number of actual links $m_{r,s}$ between them and $m_{r,s}^{(h)}$ above would be

$$\frac{m_{r,s}}{m_{r,s}^{(h)}} = \frac{p_{r,s}}{p}. \tag{15}$$

The quantity p is in fact straightforward to introduce into this picture, because the only choice applicable to a network without community structure is the one that leads to m, the observed number of links in the network, which makes it

$$p = \frac{m}{\binom{n}{2}}. \tag{16}$$

With this choice of p, the fraction in Eq. 15 should be < 1 if the network has community structure and $r \neq s$, and > 1 if $r = s$. This is because if there is assortative community structure, the same m links would need to be distributed so that nodes within the same community would have a higher per node link count than nodes between communities.

Since the actual communities are assumed unknown, one would not hope to find the communities r and s above to test directly the ratio in Eq. 15. However, this does not prevent us from detecting that there are local variations of the link density in the network. A somewhat simple such test involves the separation of the network into two potential global communities, say 1 and 2. These may even be of basically the same size although this is not totally necessary. Assignment of a node to each community can done arbitrarily initially, but then one can move nodes between communities if doing so reduces the number of links that need to go between communities

1 and 2. If one continues this process and the end result is that one community has *sufficiently* more links than the other, it is likely that the network does possess community structure. The reason why we have highlighted the word sufficient is because it is possible to have a discrepancy in the number of links between communities, but for that discrepancy to be entirely explainable from random variance, which would then make it probabilistically impossible to argue that indeed there is community structure; it could be that all one is seeing is a statistical fluke.

The general problem of making all the assignments of the n nodes to q communities from scratch so that one can test the ratio in Eq. 15 systematically is NP-complete (more than exponentially costly). However, there are heuristic approaches to this problem that help find good enough communities, if there are any. Some of these heuristics are greedy, assuming that a local choice that improves the situation also leads to a global improvement.

For instance, if a network is suspected of having q communities, one such greedy approach would assign to all nodes an initial community. Then, one node at a time, one can test whether reassigning that node to another one of the q communities simultaneously reduces the links out of the current state of the community while increasing the links within the community. Once this node has been reassigned to a community, it is not visited again. Also, a new node is picked for reassignment. When all nodes have been explored in this way, one obtains a community assignment that can be checked against Eq. 15 to determine if indeed communities are present.

Note that these approaches do not require knowing the values of the elements of matrix $\mathbf{P}$. Rather, they only test whether there are locally high concentrations of links in a network among groups of nodes and also sparse relations between other groups of nodes. Once this is established, it becomes clear there is community structure and the next step is to determine its details.

The simplest version of stochastic block model one can use is called a planted partition model, were $p_{t,t} = p_1$ and $p_{r,s} = p_2 < p_1$, for all r, s, t. Then, the choices of the model become simpler. If in addition the communities were of equal sizes, then much less freedom would allow us to calculate much more of the model, an approach that was typical in the early days of this model.

5 Hierarchical Clustering

This method is very effective when the community structure of the network under analysis is hierarchical. The idea is that the method progressively cuts out links that have a separating effect on the network. Every time that the elimination of a link leads to the formation of a separate cluster, we have found a community.

The method was proposed by Girvan and Newman in a now highly cited paper [3]. It relies on the use of *edge betweenness*, and equivalent concept to node betweenness in that it counts shortest passing through a spot in the network, but instead of counting paths on the nodes, the count is performed on the links. Although not a requirement, it is common to use the method together with Newman's algorithm to determine link betweenness [5].

If a link has a high betweenness, then it means that it acts as a bridge between parts of the network. The algorithm is then performed through the following steps:

1. starting with the original network, calculate the betweeneesses of all the links in the network,

2. eliminate the link with the largest between,

3. if the elimination of the link leads to two separate clusters, then these clusters are labeled as being two separate communities, but if the link elimination does not separate the network, then the nodes continue to be assumed to belong to the same community,

4. recalculate link betweenness and repeat steps 1 through 3.

Note that the elimination of a link is capable of either leaving the global number of clusters unchanged or of creating two clusters out of one cluster. This repeats hierarchically until all the nodes are isolated nodes, indicating that there is no more link elimination possible. This method is efficient, and has been applied to many settings. The drawback is that it does not give a proper opportunity for the exploration of community structure that is *not* hierarchical. It is very common to see the result of the analysis displayed in the form of a dendrogram. The main split between two communities occurring close to the beginning of the algorithm run is usually the most clear-cut community detection event.

6 Infomap: a dynamic approach to determine communities

There is another intuition that can be applied to assortative mixing. If a random walker travels inside a network that has community structure, then it is likely that the walker will more easily stay inside a community than leave it. This is an explanation that presupposes that there are more links within a community than there are leaving the community. But to be precise, this method does not really require that.

The actual principle behind the method is that if one needed to compress the record of a random walk travelling inside the network into a minimum amount of data, then the fact that there are communities effectively reduces the need for information. This is because communities mean that one can encode parts of the walk with less information because the walker remains in a certain part of the network rather than being able to go "from anywhere to anywhere". In other words, because there are "places" in the network more connected internally than to the outside, even if this is only a small deviation, walks on average will remain in such places a little longer than just travelling the breadth of the network without restriction. This means that on average, walks to do travers the network with equal speed everywhere, but instead they slow down around communities and then eventually do move on.

The algorithm associated with the method is called **Infomap**, and it captures the idea that in the network, the communities are a bit like cities. Then the savings in information that are gained by these "cities" is that one can write the information about movement in two levels: one representing the city, and another representing the internal movements within a city. The way that this economizes the need for information is that, in comparison to a network with no community structure, every node in the network has equivalent importance and therefore likelihood to be reached, so all nodes are at an even level. But with communities, the names of the nodes could be encoded with some savings: name all nodes within a community with a short common code plus an additional, within-community extra code; then nodes in another community are named with another community code and *repeat* the use of within-community codes.

The method is applied as described in Ref. [4], and it is one of the stronger methods for community detection.

References

[1] Fortunato S, Hric D (2016) Community Detection in Networks: A user guide. Phys Rep 659, 1-44.

[2] Fienberg SE, Wasserman SS (1981) Categorical Data Analysis of Single Sociometric Relations. Socio. Method. 12, 156-192.

[3] Girvan M, Newman MEJ (2002) Community structure in social and biological networks PNAS 99, 7821-7826.

[4] Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure PNAS 105, 1118–1123.

[5] Newman M E J(2001) Phys Rev E 64:016131.