Capitolo 1

Evaluation methods

In this section, I am going to explain how we evaluate the partitions created by the community detection algorithms, as I introduce in the summary, this method, that we decided to use, is called modularity. Its symbol is Q.
The modularity is a value that is typically contained in the range of $[-1, 1]$, but this is not all times true because it depends on the implementations. The first persons who have worked on the modularity were Newman and Girvan, that like us, they had needed a method for choosing between two partitions of communities on the nodes.
They start from an assumption a node can not belong to more than one community, this assumption is not always true for our situation but it simplifies a lot the complex problem of calculating modularity. So for the first steps, we use the same assumption.
During the time of the internship we use three different methods, not ideate from our, now I am going to explain them from the simplest to the most complex. The formulas reported are for undirected graphs at the end of each subsection there is the differences with directed graphs.

1.1 Modularity with Maximum

We named it in this method because it permits at maximum one community for each node, perfectly in line with the assumption of Newman and Girvan.
The idea is that this method calculates a modularity value for each community in the graph and the total modularity is simply the summation of all the parts. We must specify that, in this method, the overlapping is not considered then the summation does not consider the same elements more than one times.
This method is based on the observation that we can identify a community on the base of the frequency of its edges, in fact, if a set of points have a huge amount of edges inside it probably this is a community if it is not, this is a bad situation that must penalize all the partition.
We chose it because is easy to calculate and because it penalizes a lot the wrong partitions.
Here the formula explained:

$$Q = \sum_{c}^{C_r} \left(\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right) \tag{1.1}$$

Where the elements of the formula are:

- Q is the symbol for modularity
- C_r is the set of the communities
- c is the iterator on the communities
- L is the total number of edges in the graph
- l_c is the total number of edges inside, both nodes of each edge are inside the community c

- d_c is the degree of the community c, defined as the summation of the degree of the nodes belong to it

We can see that $\frac{l_c}{L}$ is the weight of the community compared to the others in the graph, and $\frac{d_c}{2L}$ is the density of edges, in other words, the percentage of edges that belong to this community compared to the total number.

To note that when the graph is directed change only a multiplicative factor indeed the "density of edges" change from $\frac{d_c}{2L}$ to $\frac{d_c}{L}$, and this is all.

1.2 Modularity with Overlap

This algorithm is taken from the paper named "Modularity measure of networks with overlapping communities".
As the name says, now the overlapping of the community it is allowed. Then now we do not have a summation but an average. The results are inside a range of $[-1, 1]$.
Here the formula explained:

$$Q = M^{ov} = \frac{1}{K}\sum_{r=1}^{K}\left[\frac{\sum_{i\in c_r}\left(\frac{\sum_{j\in c_r, i\neq j}(a_{ij}) - \sum_{j\notin c_r}(a_{ij})}{d_i \cdot s_i}\right)}{n_{c_r}} \cdot \frac{n^e_{c_r}}{\binom{n_{cr}}{2}}\right] \tag{1.2}$$

Where the elements of the formula are:

- K is the number of communities $|C_r|$
- c_r is the actual community
- n_{cr} number of nodes in the community
- $n^e_{c_r}$ number of edges in the community
- $\binom{n_{cr}}{2}$ maximum number of edges in the community
- d_i degree of the node i
- s_i number of community to which the node i belong
- a_{ij} 1 if the edge from the node i to the node j exist, 0 otherwise

Now, we try to figure out the particular things that appear in this formula.
As before we calculate the density of the community (with the equation number **??**), but not compared to the total number of edges of the graph, but compared to the hypothetical maximum number of edges inside the nodes considered:

$$\frac{n^e_{c_r}}{\binom{n_{cr}}{2}} \tag{1.3}$$

In addition, we consider the relationship between the number of inward and outward edges, compared to the total number of edges of the community (in the equation **??**). Then if the outward edges are more than the inward edges, the modularity is negative otherwise is positive.

$$\sum_{i\in c_r}\left(\frac{\sum_{j\in c_r, i\neq j}(a_{ij}) - \sum_{j\notin c_r}(a_{ij})}{d_i \cdot s_i}\right) \tag{1.4}$$

Lastly, when we look at the inward and outward edges we consider the two parameters d_i and s_i, this is important because if a node has only one edge and it belongs only to one community it has a high weight, differently if it has a hundred of nodes and belongs to dozens of communities, the weight of one of its edges is very low.

We chose this method because it allows the overlapping of communities and this is necessary for our situation. In addition is it possible to figure out which community have a high value, then increase the final results and which one decrease it, cause the low value.
From each modularity value, we can understand two things of the graph:

- If the value is negative mean that the group of nodes is not densely connected inside it, but rather is connected to some external point. It is possible that there are more inwards edges compared to the outwards, but if the value is negative this means that the outwards edges are more relevant
- If the modularity in absolute value is very near at 0 this means that the graph is extremely sparse in fact the multiplication factor (shown in equation **??**) collapse all the elements of the formula near zero

To note that when the graph are directed only the equation **??** change in fact the maximum number of edges is no more $\binom{n_{c_r}}{2}$ but became $2\binom{n_{c_r}}{2}$.

1.3 Modified modularity

This algorithm is taken from the paper named "Incorporating Implicit Link Preference Into Overlapping Community Detection".
Finally, I arrive to explain this method, this is the algorithm choose in the paper of CNRL, for this reason, we looked for it. It is very important because allow our to replicate the results shown in the paper of CNRL, then we can proceed to find a fixed point. Unfortunately, this is the slowest algorithm of the three that I show you in this section because it is the most complex.
Let's look at how it works:

$$Q = \frac{1}{M} \cdot \sum_{u,v \in V} \left(\left(A\left[u,v\right] - \frac{d_{in}\left(u\right) \cdot d_{out}\left(v\right)}{M} \right) \cdot \left| C_u \cap C_v \right| \right) \tag{1.5}$$

Where the elements of the formula are:

- M stands for m if the graph is undirected and for $2m$ if the graph is directed, where m is the number of edges of the graph
- u, v are the iterators on the graph's nodes named V
- $A\left[u,v\right]$ is the weight of the edge uv if the edge does not exist this value is 0, normally a node without weight is considered as 1. It is taken from the adjacency matrix
- $d_{in}\left(u\right)$ and $d_{out}\left(u\right)$ are the incoming/outgoing degree namely the number of incoming/outgoing edges of the node u. If the graph is undirected those values are equal
- C_u is the set of communities to which the node u belong
- $\left| C_u \cap C_v \right|$ is the number of communities that node u and v share

I try to explain this equation in words.[1]
Starting from the end we see the element $\left| C_u \cap C_v \right|$, this is important because allow ignoring the edges that link two nodes of different communities, so only the edges inside a community is considered. In addition, this part is a multiplicative factor, if an edge is inside lots of communities it has a high weight

[1] Non sono completamente sicuro delle meccaniche alla base della modularità modificata

if it belongs only to one community is weight is lower.
But what is the base of this multiplier factor? We can see that this is a positive value if the edge exists (thanks to $A[u, v]$), in fact, more edges a community has more it is defined. From the weight of the edge we subtract a factor (defined from $\frac{d_{in}(u) \cdot d_{out}(v)}{M}$), this represents the relevance/likelihood of the edge, if it connects two nodes with a high degree, this edge is not too important therefore we decrease its weight slightly, at the opposite is very important if connect two nodes that have low values of degree.