Lezione persa contagion networks

Today we talk about community detection algorithms

Topology is a key word here.

Networks are not homogeneous structures from the point of view of degree distribution, some nodes can have higher degree than others, way higher!

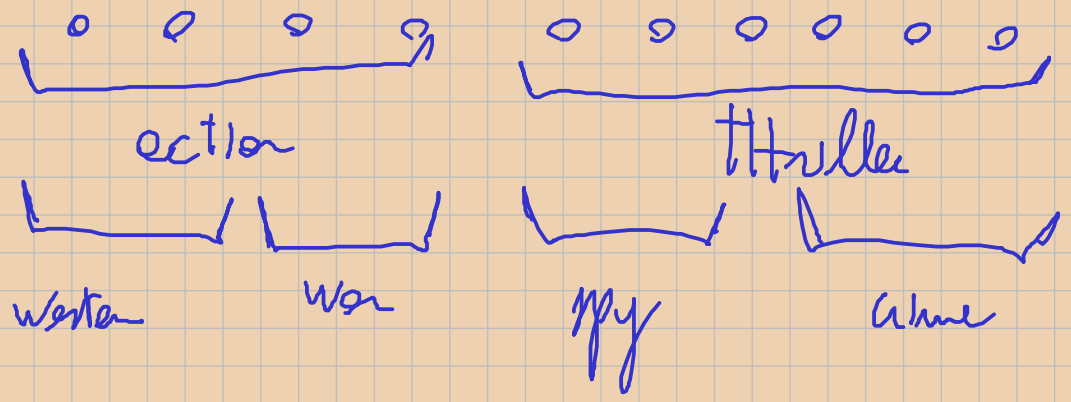
> Community: a set of nodes that are more connected to each other than to any other node in the network

To isolate communities we can break the intercommunity links in order to leave only the intracommunity ones.

We want to have an algorithm that runs in a PC and gives us all the different communities. This is a very difficult task.

After some community research based on topology, we proceed with community characterization to see if the results make sense or if there's some manual "adjustments" to do

Communities can have a hierarchical structure also!



To not have communities overlapping, for us a node will be assigned to one and one only community.

Clusters: group of nodes that emerge from the other nodes in the network because they are similar. "similar" here means that you can introduce a distance between nodes, and the lower is the distance the higher is the similarity. This "distance" is different than the fact that 2 nodes are linked.

Clusters are different than community!!!

- $\delta_{in}(C) = \dfrac{\text{n of edges inside } C}{\frac{N_C(N_C-1)}{2}}$

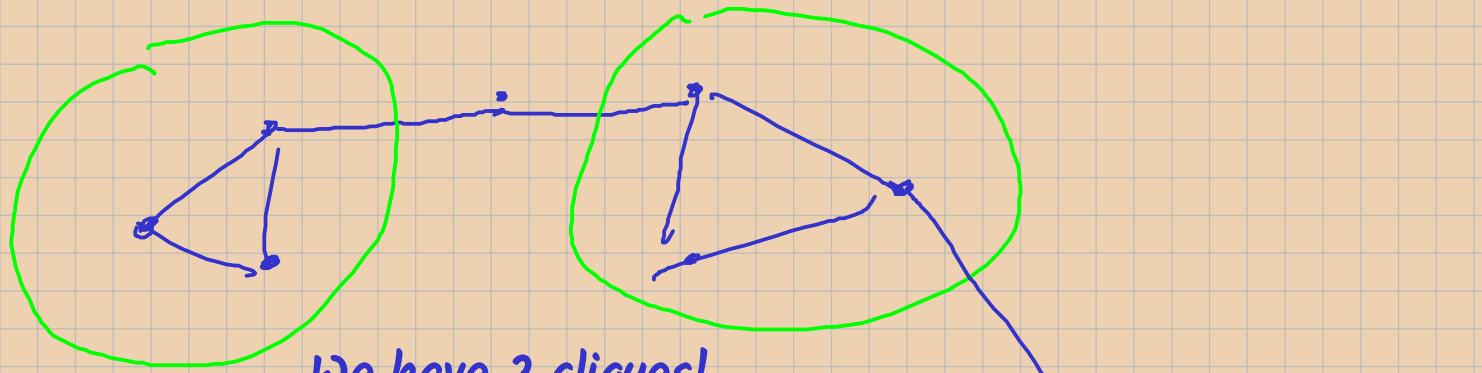
intra-cluster density of the community C

N_C: number of nodes inside the community

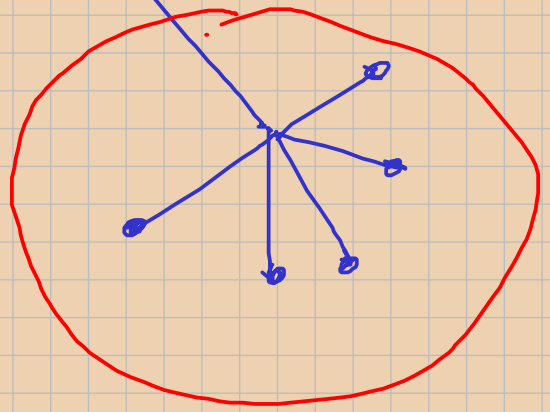
- $\delta_{ext}(C)$ $\dfrac{\text{number of edges from nodes within C with outside nodes}}{N_C(N-N_C)}$

We have a community if the ratio between those two is very high! $\delta_{int} >> \delta_{ext}$

A "clique" is a subset of nodes all connected to each other!



We have 2 cliques!



This is not a clique! It's called "motif"

But it's a community nonetheless!

GLOBAL definitions:

A way to check if our network has communities (so it have inhomogeneities), we can create a comparison network, we create it as an E-R model with the same number of nodes and links, but the links are random of course.
By comparison between those 2, we can see if the networks differ in significant metrics.

The problem with this, is that in ER the degree distribution is a gaussian with "low" std. dev. so basically every node has about the same degree.
But this is not the case in real life.

To avoid this, we can use the configuration model instead of ER! So this problem is solved.

We skipped some slides

We go directly to Modularity!

By confronting the adjacency matrix of the original model and that of the comparison model, we can define the Modularity function, for the configuration model as comparison:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \cdot \delta(c_i, c_j)$$

Q → 1

We want to maximize the value of the modularity function!

cronecker delta, we sum only if two nodes are inside the same community

We change the original network by breaking links between different communities, and compute Q each time, we keep the network "partition" with higher Q

This is very computationally difficult to do.

The links that separate different communities have high betweenness! So they're kinda easy to find this way!