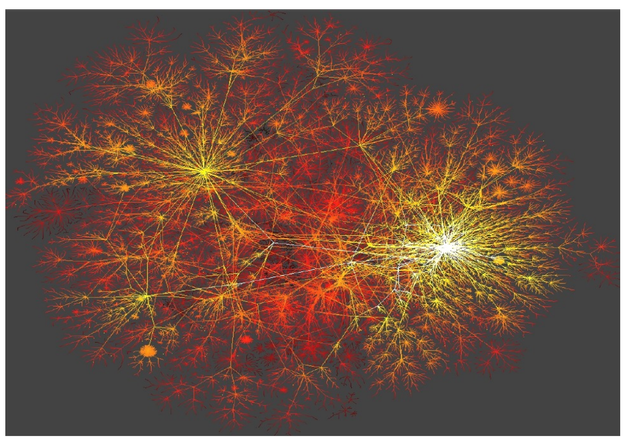
Community detection and proximity measures

Maximilien Danisch[1,2], Jean-Loup Guillaume[1] and Bénédicte Le Grand[2]

[1] LIP6 (CNRS – UPMC) and [2] CRI (Université Paris 1)

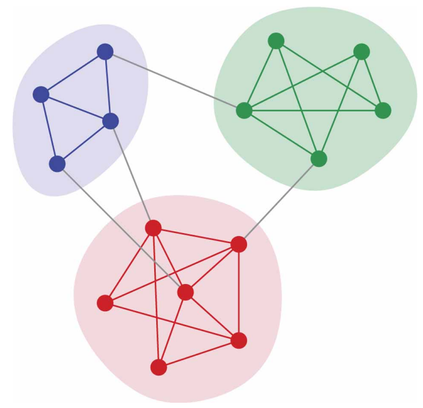
Context: in complex networks, there are communities



networks	nodes	edges	communities
Wikipedia	Wikipedia pages	hypertext links	Wikipedia categories
Facebook	profiles	friendships	colleagues, families, sports club
P2P	peers	file exchanges	communities of interests

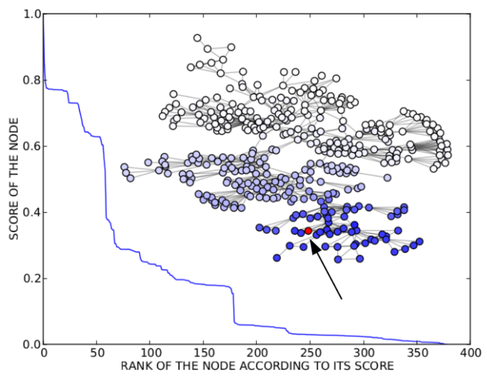
Problematics: How to find these communities?

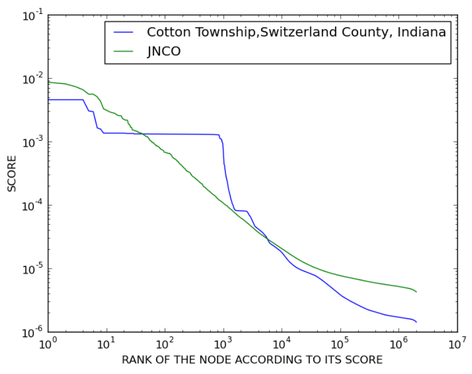
- **The greedy optimization of a quality function suffers from: (i) local minima and (ii) hidden scale parameters.**
- **Let's look for an alternative approach!**



Idea 1: use the related notion of proximity measure

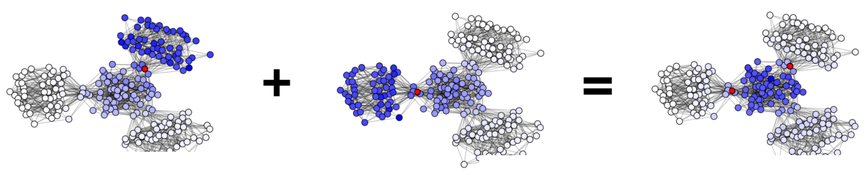
Rank nodes according to their proximity to a given node.
Irregularities in the decrease can indicate the presence of communities:

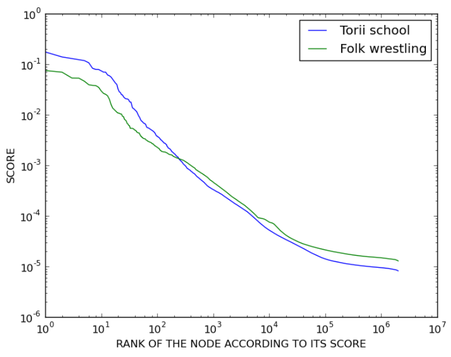


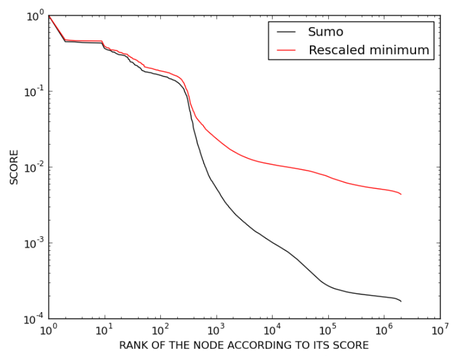


- **Often a powerlaw is obtained = no scale can be extracted = the node belongs to several communities of various sizes.**

Idea 2: use the notion of multi-ego-centered community

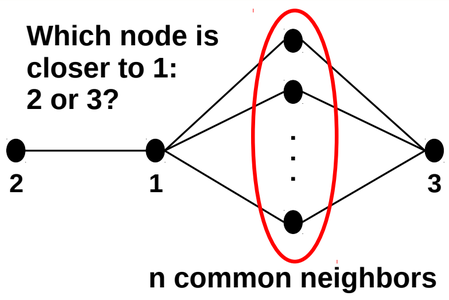






- **While one node generaly belongs to many communities, a small set of nodes, e.g. 2, is enough to define a single community.**

Idea 3: a proximity measure has to be parametrized



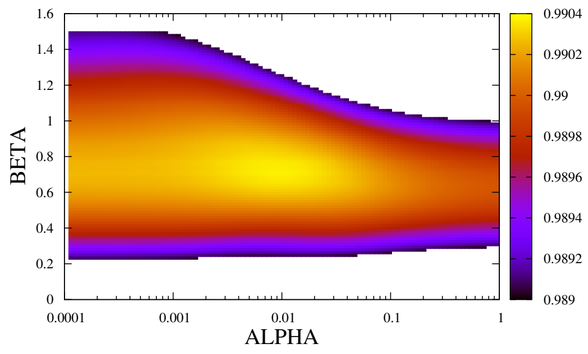
What to do with HUBs?

$$\mathrm{Prox}_{\alpha\beta\lambda}(i,j) = \frac{1}{d_j^{\beta}} \sum_{l=0}^{\lambda} \alpha^l N_l(i,j)$$

$N_l(i,j)$: number of paths of length l between i and j. d_j: degree of node j.

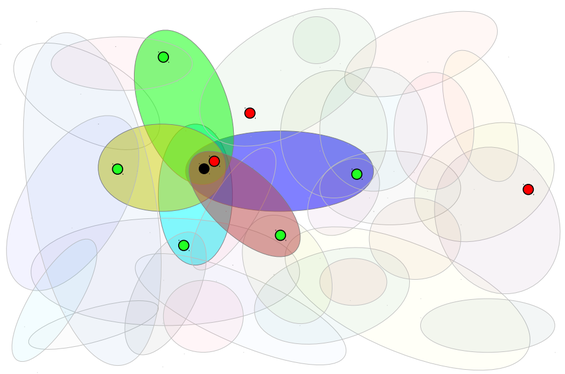
- **If we know some nodes that are near to one another, e.g. nodes belonging to a same community, we can learn the parameters that rank these nodes as close as possible to one another.**

AUC optimization: the AUC is a measure of the accuracy of a classifier, it is equal to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one.

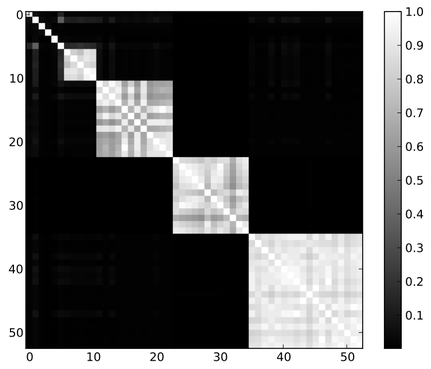


Application 1: find all communities of a node

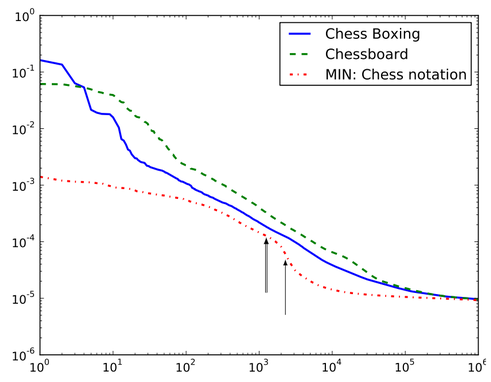
A- Choose a set of candidates:

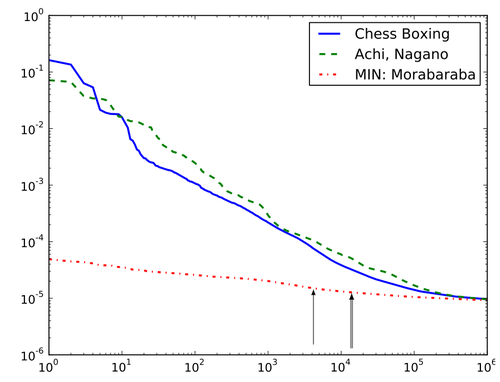


C- Clean and label the communities:



B- Find bi-ego-centered communities:

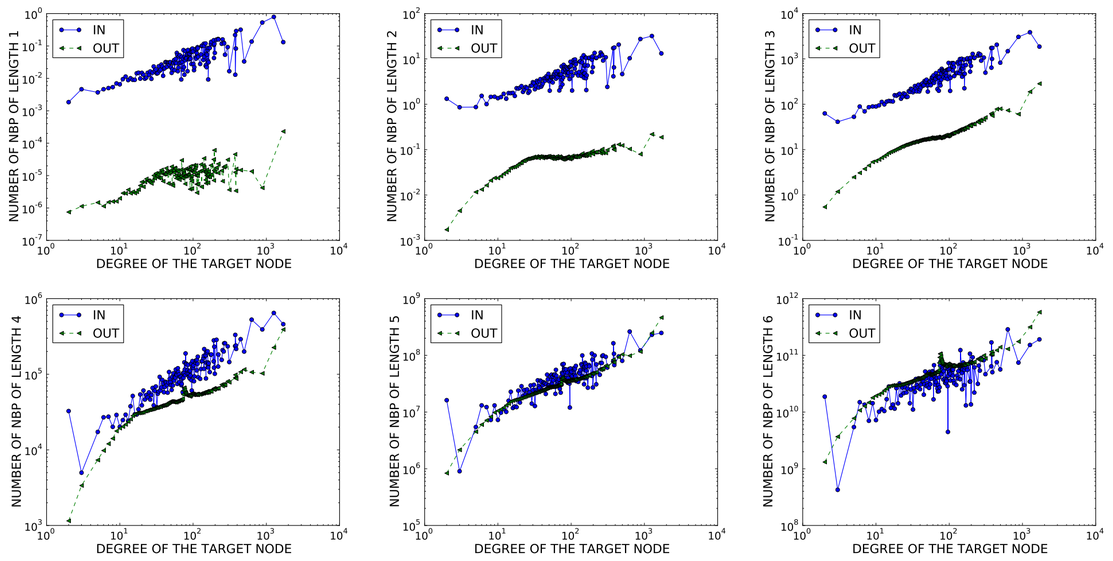




- **We can use a similar framework to find all overlapping communities in a networks.**

Application 2: community completion

Average number of paths of length 1 to 6 between two nodes in the same community and two nodes in different communities:



Learning a proximity measure, combining scorings and unfolding the community: