Connectedness and Density Hypothesis

Communities are locally dense connected subgraphs in a network. This expectation relies on two distinct hypotheses:

（1）Connectedness Hypothesis

Each community corresponds to a connected subgraph. The hypothesis also implies that on the same component a community cannot consist of two subgraphs that do not have a link to each other.

（2）Density Hypothesis

Nodes in a community are more likely to connect to other members of the same community than to nodes in other communities.

In other word, it is a connected subgraph with maximal link density.

Modularity

<http://networksciencebook.com/chapter/9#modularity>

Consider a network with N nodes and L links and a partition into communities, each community having nodes connected to each other by *links,* where c=1,...,n c. If is larger than the expected number of links between the *Nc* nodes given the network’s degree sequence, then the nodes of the subgraph could indeed be part of a true community, as expected based on the Density Hypothesis H2 (Image 9.2). We therefore measure the difference between the *network’s real wiring diagram* (*)* and the expected number of links between i and j if the network is randomly wired (),

Here can be determined by randomizing the original network, while keeping the expected degree of each node unchanged. Using the degree preserving null model (7.1) we have

If Mc is positive, then the subgraph Cc has more links than expected by chance, hence it represents a potential community. If Mc is zero then the connectivity between the Nc nodes is random, fully explained by the degree distribution. Finally, if Mc is negative, then the nodes of Cc do not form a community.

Using (9.10) we can derive a simpler form for the modularity (9.9) (ADVANCED TOPICS 9.B)

where is the total number of links within the community and is the total degree of the nodes in this community.

To generalize these ideas to a full network consider the complete partition that breaks the network into communities. To see if the local link density of the subgraphs defined by this partition differs from the expected density in a randomly wired network, we define the partition’s modularity by summing (9.11) over all communities [23]

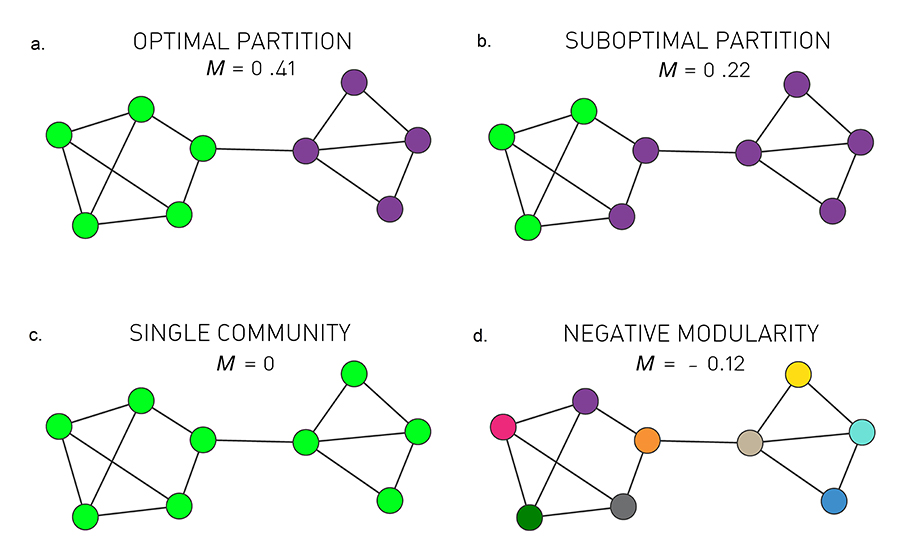
Modularity has several key properties:

(1)Higher Modularity Implies Better Partition

The higher is M for a partition, the better is the corresponding community structure. Indeed, in Image 9.16a the partition with the maximum modularity (M=0.41) accurately captures the two obvious communities. A partition with a lower modularity clearly deviates from these communities (Image 9.16b). Note that the modularity of a partition cannot exceed one [31,32].

(2)Zero and Negative Modularity

By taking the whole network as a single community we obtain M=0, as in this case the two terms in the parenthesis of (9.12) are equal (Image 9.16c). If each node belongs to a separate community, we have Lc=0 and the sum (9.12) has nc negative terms, hence M is negative (Image 9.16d).



We can use modularity to decide which of the many partitions predicted by a hierarchical method offers the best community structure, selecting the one for which M is maximal.

Greedy Algorithm

The first modularity maximization algorithm, proposed by Newman, iteratively joins pairs of communities if the move increases the partition's modularity. The algorithm follows these steps:

(1) Assign each node to a community of its own, starting with N communities of single nodes.

(2) Inspect each community pair connected by at least one link and compute the modularity difference ΔM obtained if we merge them. Identify the community pair for which ΔM is the largest and merge them. Note that modularity is always calculated for the full network.

(3) Repeat Step 2 until all nodes merge into a single community, recording M for each step.

(4) Select the partition for which M is maximal.

Resolution Limit:

---Modularity maximization forces small communities into larger ones. Even a single link between them will force the two communities together when we maximize M. This resolution limit has several consequences:

(1) Modularity maximization cannot detect communities that are smaller than the resolution limit

(2) Real networks contain numerous small communities [36-38]. Given the resolution limit (9.14), these small communities are systematically forced into larger communities, offering a misleading characterization of the underlying community structure.

---Computational Complexity:

Since the calculation of each ΔM can be done in constant time, Step 2 of the greedy algorithm requires O(L) computations. The O(N2) computational complexity of the greedy algorithm can be prohibitive for very large networks.

Merging Two Communities

Consider communities A and B and denote with kA and kB the total degree in these communities (equivalent with kc above). We wish to calculate the change in modularity after we merge these two communities. Using (9.12), this change can be written as

where

lAB is the number of direct links between the nodes of communities A and B, and

*kAB=kA+kB*

After inserting (9.55) and (9.56) into (9.54), we obtain

*Δ=lABL−kAkB/2L^2*

The Louvain Algorithm

The O(N2) computational complexity of the greedy algorithm can be prohibitive for very large networks. A modularity optimization algorithm with better scalability was proposed by Blondel and collaborators. The Louvain algorithm consists of two steps that are repeated iteratively:

Step I

Start with a weighted network of N nodes, initially assigning each node to a different community. For each node i we evaluate the gain in modularity if we place node i in the community of one of its neighbours j. We then move node i in the community for which the modularity gain is the largest, but only if this gain is positive. If no positive gain is found, i stays in its original community. This process is applied to all nodes until no further improvement can be achieved, completing Step I.

The modularity change ΔM obtained by moving an isolated node i into a community C can be calculated using

where Σin is the sum of the weights of the links inside C (which is LC for an unweighted network);

Σtot is the sum of the link weights of all nodes in C;

ki is the sum of the weights of the links incident to node i;

ki,in is the sum of the weights of the links from i to nodes in C

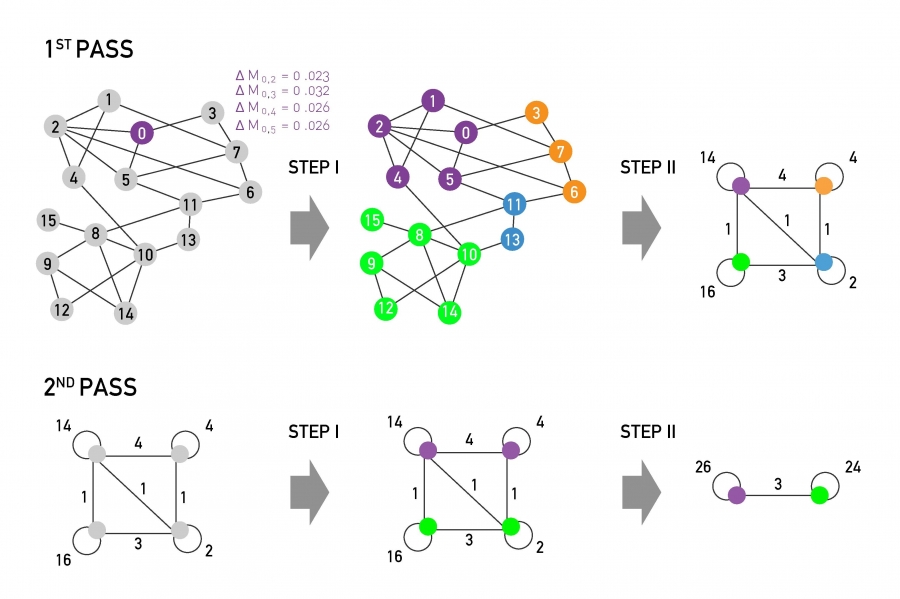
W is the sum of the weights of all links in the network.

We can use ΔM to determine the modularity change when i is removed from the community it belonged earlier. For this we calculate ΔM for merging i with the community C after we excluded i from it. The change after removing i is –ΔM.

Step II

We construct a new network whose nodes are the communities identified during Step I. The weight of the link between two nodes is the sum of the weight of the links between the nodes in the corresponding communities. Links between nodes of the same community lead to weighted self-loops.

Once Step II is completed, we repeat Steps I - II, calling their combination a pass. The number of communities decreases with each pass. The passes are repeated until there are no more changes and maximum modularity is attained.



Computational Complexity

The Louvain algorithm is more limited by storage demands than by computational time. The number of computations scale linearly with L for the most time consuming first pass. With subsequent passes over a decreasing number of nodes and links, the complexity of the algorithm is at most O(L). It therefore allows us to identify communities in networks with millions of nodes.

Walktrap

<https://www.cnblogs.com/tychyg/p/5277137.html> (chinese)

<https://www-complexnetworks.lip6.fr/~latapy/Publis/communities.pdf>

reference not available: *An algorithm Walktrap-SPM for detecting overlapping community structure.* March 2017International Journal of Modern Physics B 31(15):1750121

Random walk: When moving from one vertex to the next, select a neighbour of the current vertex with equal probability as the next vertex.

Basic idea: The community is a relatively dense subgraph, so it is easy to "get" into a community when random walking in the graph.

The random walk process forms a Markov chain. Each vertex in the graph corresponds to a state; The transition probability between the different states is

The probability of going from i to j with the t walk is Pij to the power of T

Algorithm steps:

Step1 each point acts as a community and calculates the distance between adjacent points (communities)

Step2 merge the two communities that minimize the mean σk of the squared distances between each vertex and its community.

Repeat this step until all points are merged into a community.

Advantage: walktrap surpasses previously proposed ones concerning the quality of the obtained community structures and that it stands among the best ones concerning the running time.

Infomap

Consider a network partitioned into communities. We wish to encode in the most efficient fashion the trajectory of a random walker on this network. In other words, we want to *describe the trajectory with the smallest number of symbols*. The ideal code should take advantage of the fact that the random walker tends to get trapped into communities, staying there for a long time

Label Propagation

Variation of information

the variation of information or shared information distance is a measure of the distance between two clusterings (partitions of elements). It is closely related to mutual information; indeed, it is a simple linear expression involving the mutual information. Unlike the mutual information, however, the variation of information is a true metric, in that it obeys the triangle inequality

we have deﬁned a discrete random variable taking K values, that is uniquely associated to the clustering C. The uncertainty in our game is equal to the entropy of this random variable

We now deﬁne the mutual information between two clusterings, i.e. the information that one clustering has about the other. Denote by P(k), k= 1,...,K and P(k’), k = 1,...,K, the random variables associated with the clusterings C, C’. Let P(k, k’) represent the probability that a point belongs to Ck in clustering C and to C k in C, namely the joint distribution of the random variables associated with the two clustering

