CGnal

business innovation through algorithms

Community Detection

CGnal S.p.A – Corso Venezia 43 - Milano

21 Novembre 2022| Milano

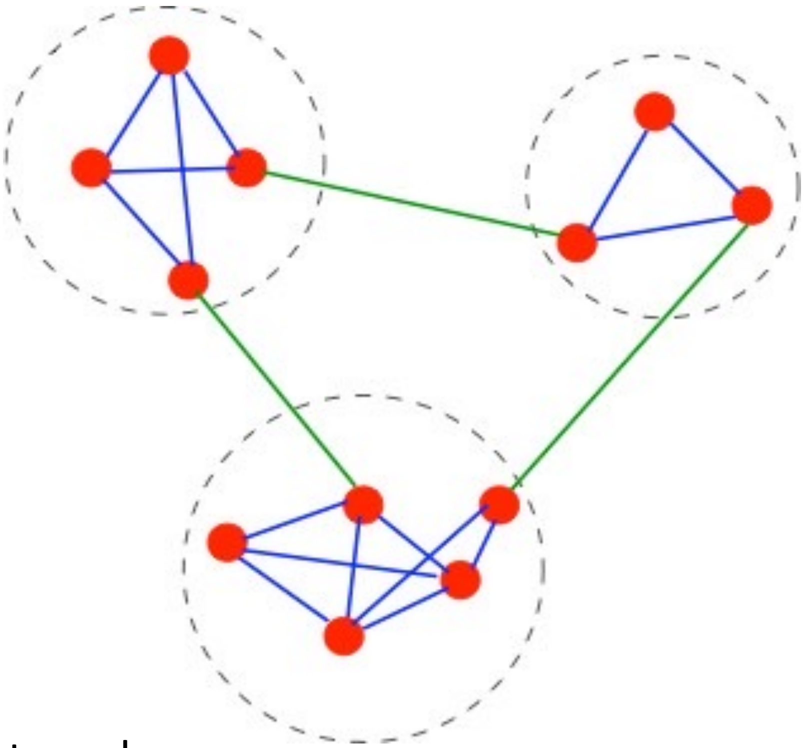
Why communities detection?

A subgraph of G=(V,E) is a graph G’=(V’,E’) such that V’ ⊆ V and E’ ⊆ E i.e.,V’ and E’ are subsets of nodes and edges of G

Community Detection vs Clustering

One can argue that **community detection** is similar to **clustering**. Clustering is a machine learning technique in which similar data points are grouped into the same cluster based on their attributes. Even though clustering can be applied to networks, it is a broader field in unsupervised machine learning which deals with multiple attribute types.

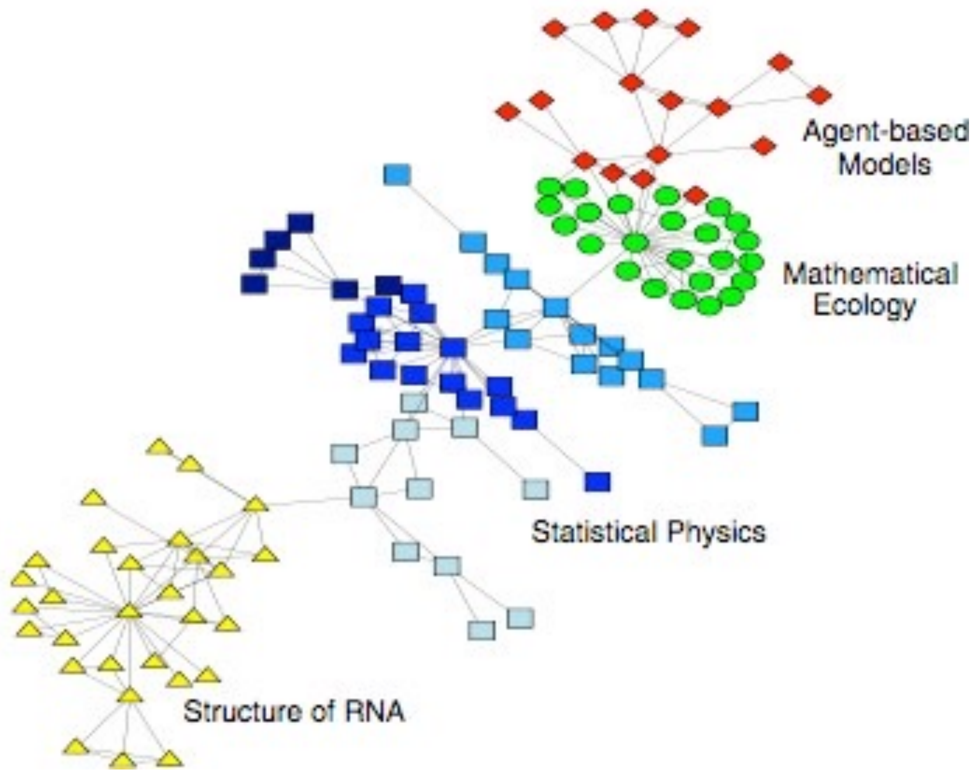
On the other hand, community detection is **specially tailored for network analysis** which depends on a **single attribute type called edges.**

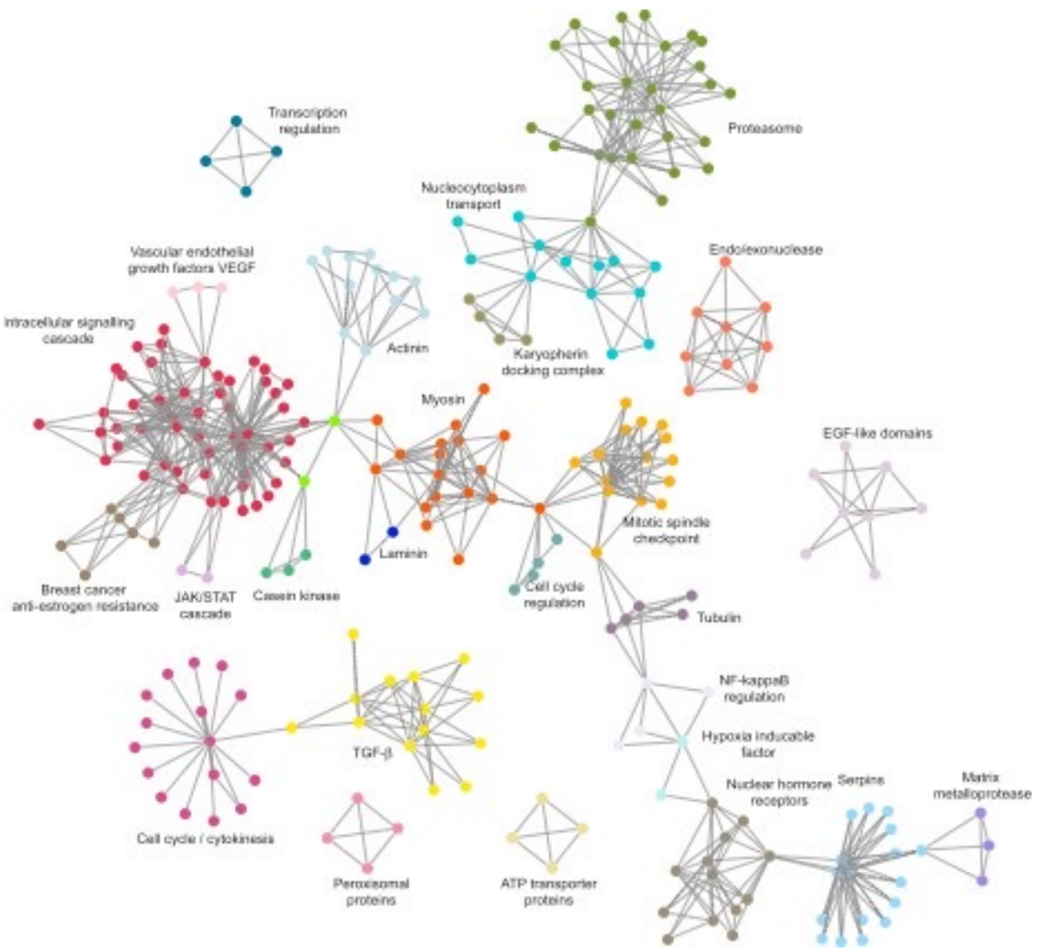


<u>Example:</u>

If we would like to group our clients we can apply a clustering over the client data and characteristics (i.e. using a sort of similarity/distance between the clients) or we can apply a community detection over the graph of the bank transfers.

Communities: examples





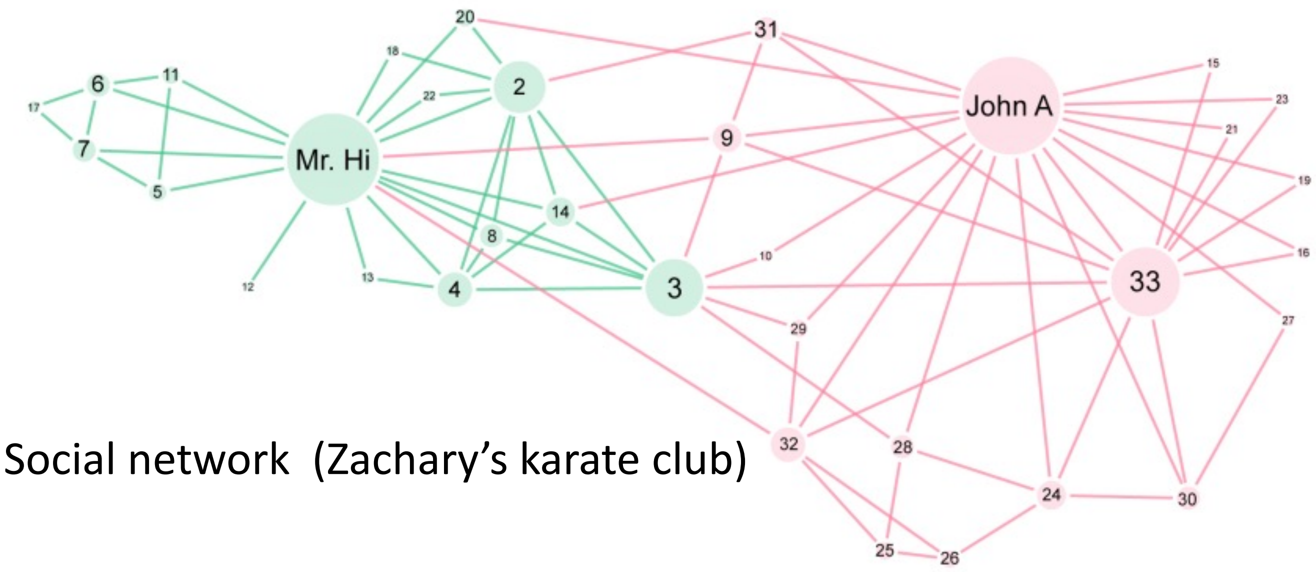
Scientist collaboration network (Santa Fe Institute)

each community represents a group of scientists working with each other in the same domain.

Protein-protein interaction network

Understanding this circuitry could improve the prediction of gene function and cellular behavior in response to diverse signals.

Communities: examples



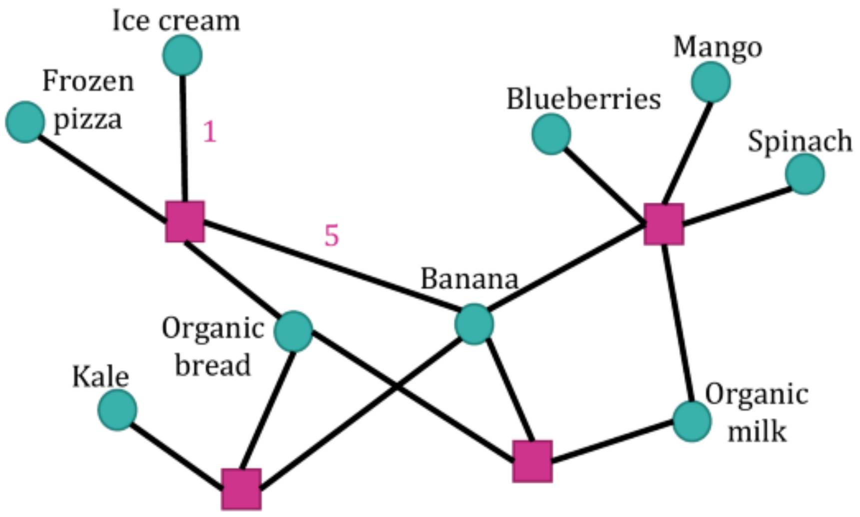
Social network (Zachary's karate club)

More generally in a company the community detection apply to the employees network can be usefull undersand the existing group of peaple and some possible centers

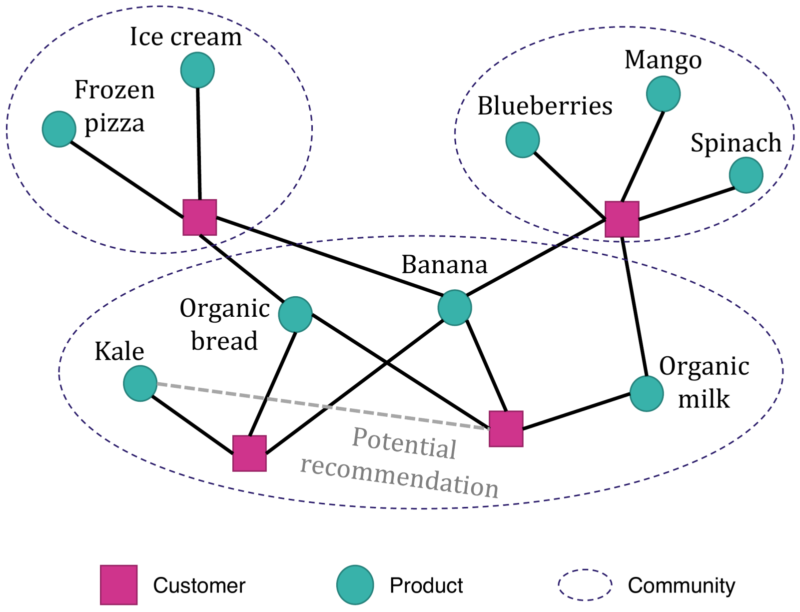
HR application?

Communities: examples

Customer ID	Basket ID	Date	Product	Quantity
12345	99999	01-05-2018	Banana	2
12345	99999	01-05-2018	Ice cream	1
12345	78987	07-05-2018	Banana	3
…	…	…	…	…

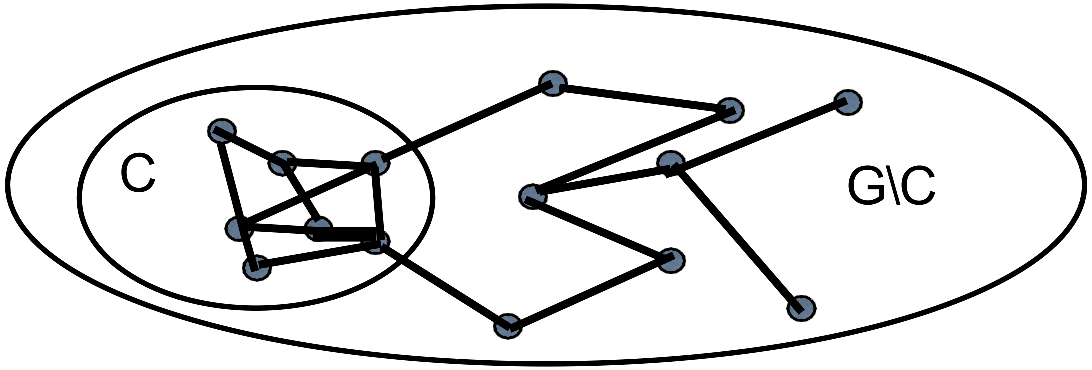


1. [Recommendations] What new items should we recommend to a given customer?
2. [Targeting] Which users should we contact in a promotional campaign for a specific product?

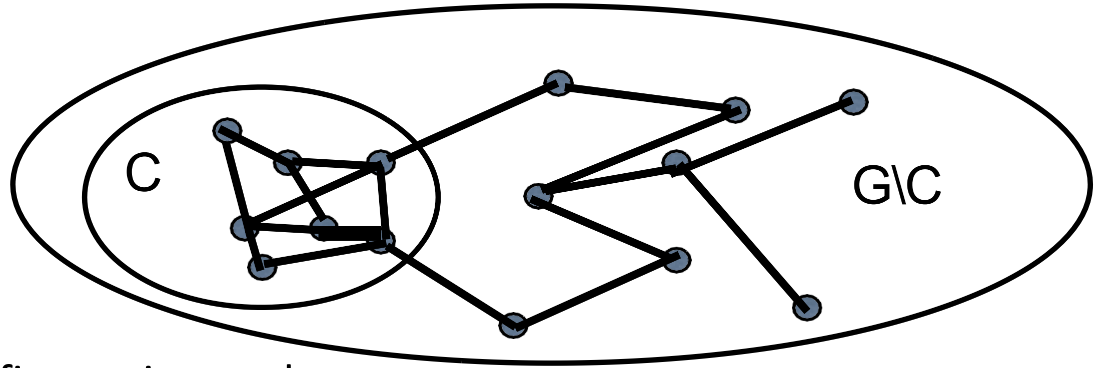


Communities: logical definition

Definition



Communities: logical definition



Definition
Group of nodes that are more tightly linked together than with the rest of the graph.

Subgraph C of G induced by n' nodes V' with *e'* edges, we can define an internal density

$$d = \frac{2\, e'}{n'(n'-1)}$$

For C to be a community, we expect that:

- d (much) larger than density of G
- d (much) larger than the density of links towards G\C, given by

$$d'' = \frac{2\, e''}{n\prime(n - n\prime)}$$

where *e''*=number of links between nodes of C and nodes of G\C, and n the total number of nodes of the graph G

n=7, e=10

N-n=8, e'=2

d=0.24, d'=0.035

Using the Adjacency Matrix

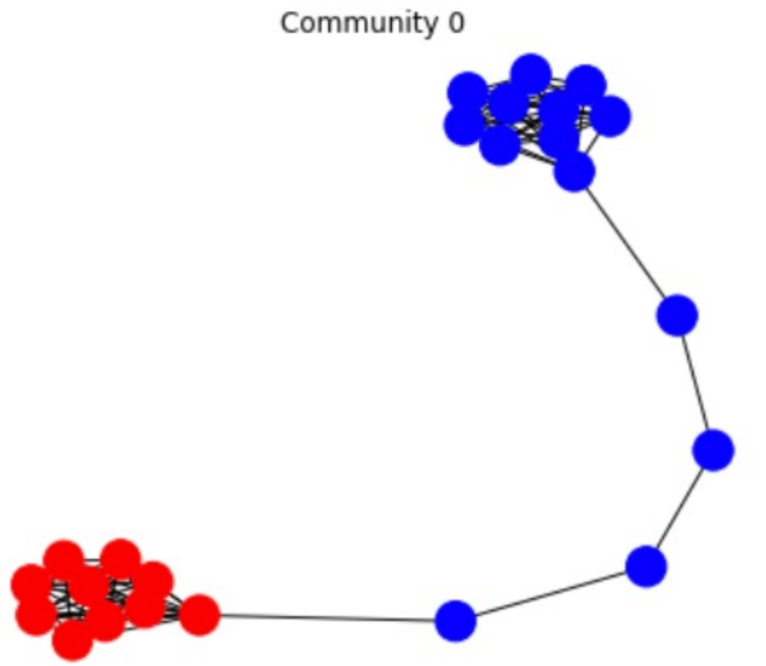
Spectral clustering

Laplacian Matrix $L = D - A$

The Laplacian carries information about the structure of the graph

Basic Algorithm

1. Calculate the Laplacian L (or the normalized Laplacian)
2. Calculate the first k eigenvectors (the eigenvectors corresponding to the smallest k eigenvalues of L)
3. Consider the matrix formed by the first k eigenvectors; the l-th row defines the features of graph node l
4. Cluster the graph nodes based on these features (e.g., using k-means)



Using the Adjacency Matrix and Embeddings

Spectral clustering

Laplacian Matrix $L = D - A$

The Laplacian carries information about the structure of the graph

Basic Algorithm

1. Calculate the Laplacian L (or the normalized Laplacian)
2. Calculate the first k eigenvectors (the eigenvectors corresponding to the smallest k eigenvalues of L)
3. Consider the matrix formed by the first k eigenvectors; the l-th row defines the features of graph node l
4. Cluster the graph nodes based on these features (e.g., using k-means)

Other techniques of matrix factorization can be used in place of these steps (e.g. NMF or embeddings)

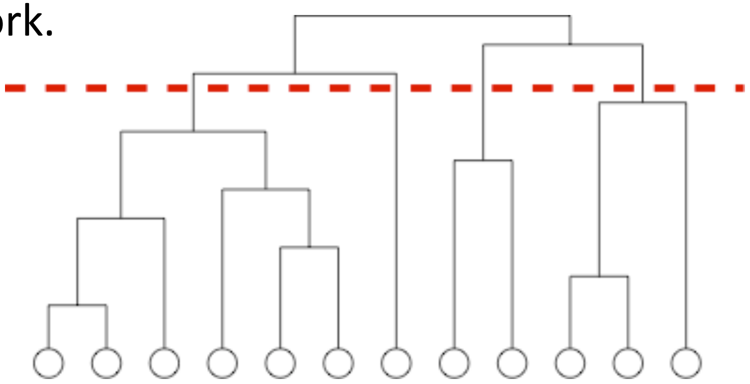
Other clustering can be used here

Communities: detection problem

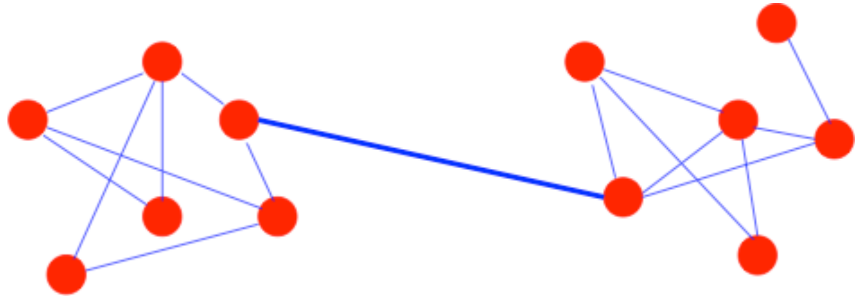
C. Centrality. Assess the importance of individual nodes inside a network.

Two main classes of methods (as in clustering problems):

- **agglomerative**: merging clusters iteratively
- **divisive**: splitting clusters by removing edges



Dendogram



Girvan-Newman algorithm (*Divisive method*)

Splitting clusters by removing edges and use edge betweenness centrality to cut into separated connected components

1. Computation of the centrality for all edges;
2. Removal of edge with largest centrality: in case of ties with other edges, one of them is picked at random;
3. Recalculation of centralities on the new graph;
4. Iteration of the cycle from step 2.

Betweenness computation: O(E2N)
Rather slow algorithm!

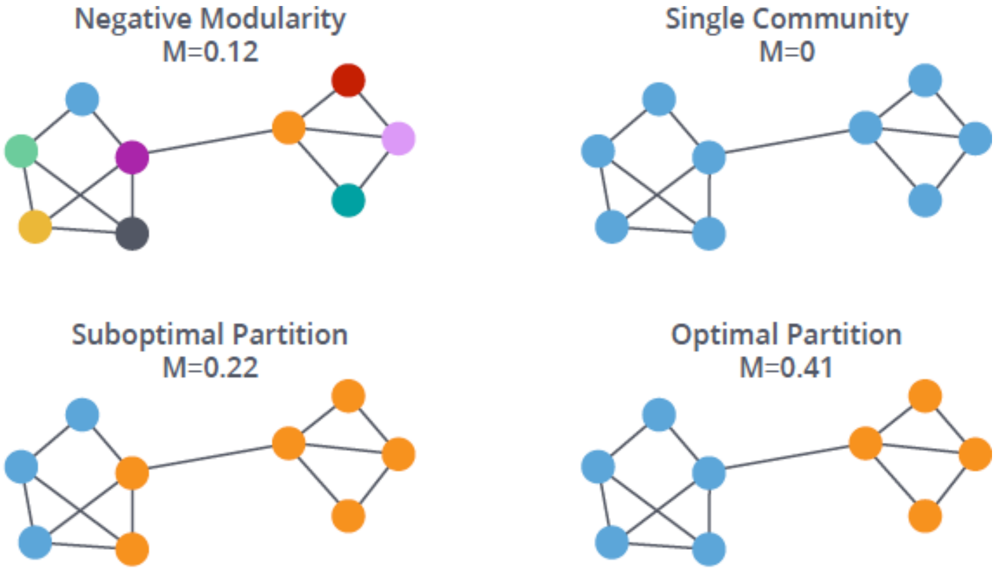
Communities: detection problem

<u>Modularity</u> was designed to quantify the division of a network in aggregated sets of highly interconnected nodes

Other approach

Optimize the node assegnation to a cluster C_i (cluster that the node V_i belongs to) in order to maximize a metric related to the partitioning of the graph into clusters

$$C_i, G(\boldsymbol{V}, \boldsymbol{E}) \longrightarrow Quality\ of\ clustering$$



Modularity

Given a partition of a graph, how to quantify if it is a "good" division into communities?

THE MODULARITY

Fundamental idea: compare the density of edges in each subgraph to a null model, i.e., a case in which no community structure is expected

Communities: detection problem

$$Q = \frac{1}{2\,E}\sum_{i,j}\left(a_{ij} - \frac{k_i k_j}{2\,E}\right)\delta(C_i, C_j)$$

Problem: find the best partitioning such that the modularity Q is maximum

We want to have an algorithm that optimize community association C_i in order to maximize the modularity

Newman

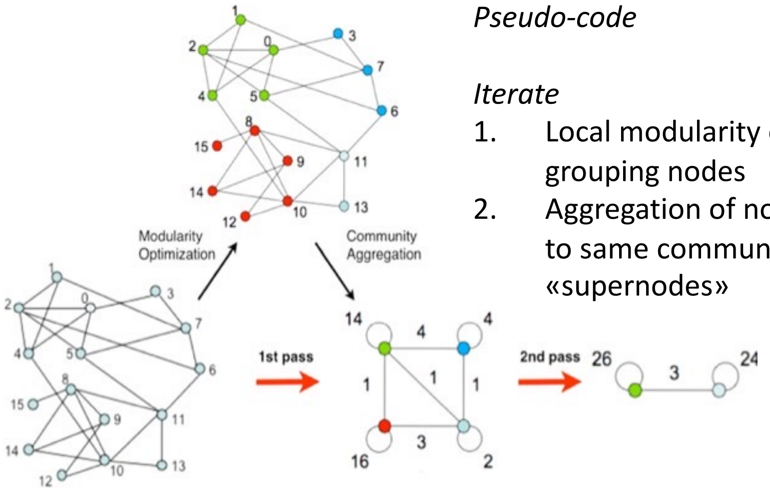
Pseudo-code

1. Start from N clusters: 1node => 1cluster
2. Iterate:
   Join two clusters in the way that increases most Q

Drawbacks

- Greedy method
- Tends to create large communities
- Not very effiicient
- Many proposed variation to get better results

Louvain



Pseudo-code

Iterate

1. Local modularity optimization by grouping nodes
2. Aggregation of node beloginig to same community into «supernodes»