Community structure of complex networks

M. Mitrović Dankulov and A. Alorić

January 13, 2023

Outline

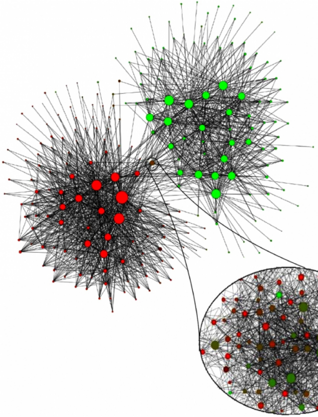
1. Motivation
2. Basic communities
3. Community detection
4. Modularity
5. Testing the partition

What we learnd so far

- Real complex networks are:
  - heterogeneous: they have hubs, and not all node and links are of the same importance
  - small-world - average shortest path grows slow with N
  - correlated - assortative and disassortative
  - clustered - some of them
  - they can be multiplex or temporal

Real networks: mesoscopic heterogeneities

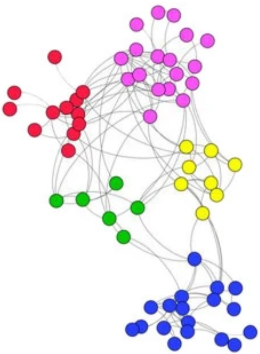
Belgian network of phone connection between citizens



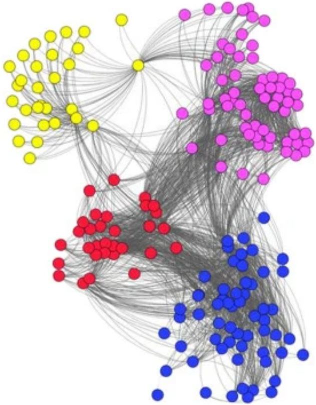
Source: Blondel et al., J. Stat. Mech., 2008.

Real networks: mesoscopic heterogeneities

Dolphin social network



Food web network



Sah et al., BMC bioinformatics, 15(1), 1-14 (2014)

Community structure

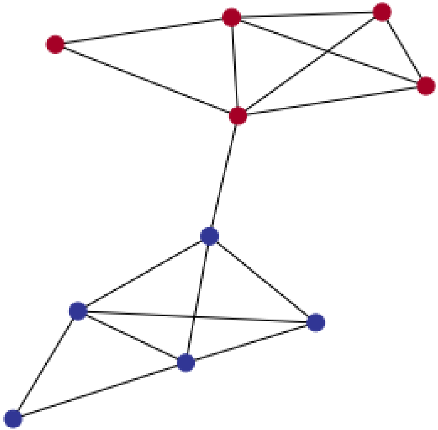
- A community is a group of nodes that have a higher likelihood of connecting to each other than to nodes from other communities
- A community is a group of nodes that connect to other groups in similar ways
- Communities can be simple, overlaping, hierarchical

We will cover

- Basics
- Types of communities
- Modularity
- Some community detection algorithms

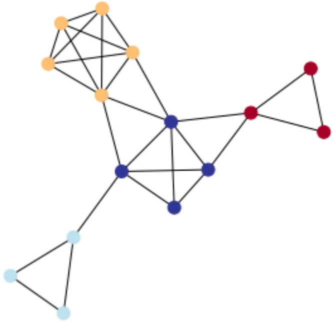
Connectedness and density hypothesis

A community is a locally dense connected subgraph in a network



Cliques

A clique is a subgraph in which each node is connected to all other nodes



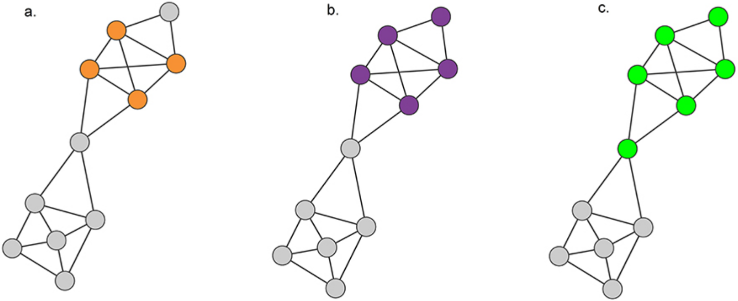
But networks do not consist of cliques only!

Strong and weak communities

- Strong communitiy is a community in which every nodes have more edges with nodes in a community than with nodes outside of the community: $k_i^{int}(C) > k_i^{out}$ for every i in C

- Weak community is a community whose total internal degree of a subgraph exceeds its total external degree: $\sum_{i\in C} k^{int}(C) > \sum_{i\in C} k^{ext}(C)$

Example



Source: Network science book

Number of communities

- In how many ways we can split N nodes into k non-overlaping communities (groups)?

- $k = 2$: $\frac{N!}{N1!N2!}$, for $N1 = N2$ we get $\frac{2^{N+1}}{\sqrt{N}} = e^{(N+1)log(2)-\frac{1}{2}log(N)}$

- Brute force strategy is unefficient and computationally expensive

Community detection

- We need algorithm to detect communities
- We do not know the size of communities and their number
- Good algorithm should be able to find both paramteres
- Algorithm will depend on the definition of the community
- Algorthm needs to be able to performe in polinomial time

Hierarchical clustering

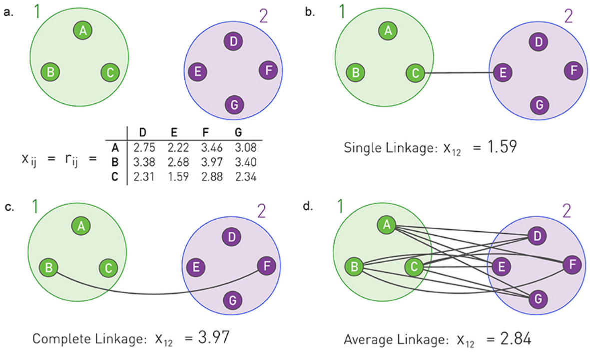
- Running time of hierarchical clustering grows polinomially with network size
- Similarity matrix: x_{ij} distance of node i from node j
- Community detection - similarity matrix describes relative position of nodes
- Hierarchical clustering iteratively identifies groups of nodes with high similarity:
  - agglomerative algorithms - merge nodes with high similarity into the same community
  - divisive algorithms - isolate communities by removing low similarity links that tend to connect communities

Ravasz Algorithm - similarity matrix

- Agglomerative algorithm - high similarity between nodes that belong to same community, low similarity between nodes from different communities
- Similarity matrix - $x_{ij}^0 = \frac{J(i,j)}{min(k_i,k_j)+1-\Theta(A_{ij})}$; $J(i,j)$ number of common neighbors of nodes i and j and $+1$ if i and j are connected; $\Theta(A_{ij})$ is Heaviside step function which is 1 if nodes i and j are connected
    - $x_{ij}^0 = 1$ - if i and j are connected and have the same set of neighbours;
    - $x_{ij}^0 = 0$ - if i and j are not connected and do not have common neighbors

Ravasz Algorithm - group similarity

Group similarity: single, complete and average



	D	E	F	G
A	2.75	2.22	3.46	3.08
B	3.38	2.68	3.97	3.40
C	2.31	1.59	2.88	2.34

Ravasz Algorithm uses average group similarity

Apply hierarchical clustering

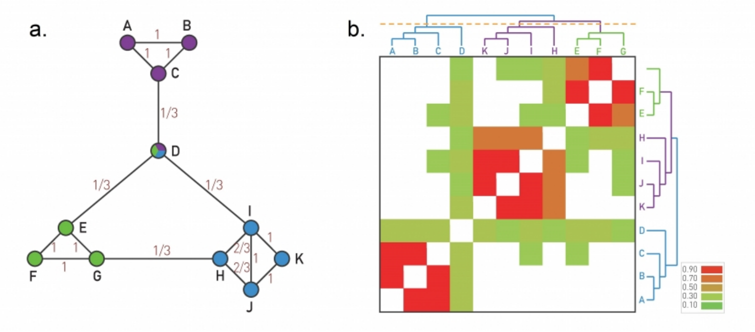
- 1. Assign each node to separate communities, $C = N$
- 2. Find two communities that have the highest similarity and merge them into one community, $C - 1$
- 3. Calculate similarity between new communities and all other communities

  Hierarchical clustering does
- 4. Repeat steps 2. and 3. untill $C == 1$

Ravasz Algorithm - dendrogram

- Dendrogram shows the underlying community organization
- By cutting the dendrogram at specific place we reveal community structure
- Hierarchical clustering does not show the optimal C

Ravasz Algorithm



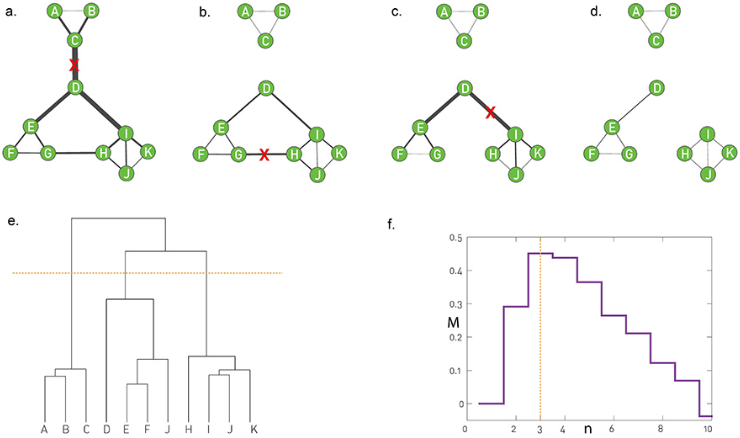
Source: Network science book

Girvan-Newman Algorithm: centrality

- Centrality measure - high if i and j are in different communities, small if they are in the same community
- Edge betweenness centrality - links that connect different communities have high betwwnness centrality
- Random-walk betweenness - how many time walker passes through link (i, j) if it starts from node n and ends in node m
- Edge betweenness centrality faster to calculate than Random-walk betweenness

- 1. Compute the centrality of each edge
- 2. Remove the edge with the largest centrality; in case of a tie, choose one link randomly
- 3. Recalculate the centrality of each link for the altered network
- 4. Repeat steps 2. and 3. until all links are removed
- Procedure is known as "maximal flow-minimal cut"

Girvan-Newman Algorithm



Source: Network science book

Modularity: definition

- Random networks lack an inherent community structure

- We need measure that measures the quality of each network partition

- One measure is modularity - measures the quality of partition compared to randomly wired network

- Modularity $M = \sum_{c=1}^{n_c} [\frac{L_c}{L} - (\frac{k_c}{2L})^2]$; L_c - number of links within the community c, k_c - total degree of nodes in community c

- Higher the M the better the partition of the network

- $n_c = 0 \rightarrow M = 0$ and $n_c = N \rightarrow M < 0$

Modularity: optimization

- Maximal value of modulariti means the best partition
- Optimization of modularity is NP-hard
- We need heuristic algorithms that are able to find the maximum of M and the most optimal partition for a given network
- There are many algorithms that find communities by finding maximum for modularity

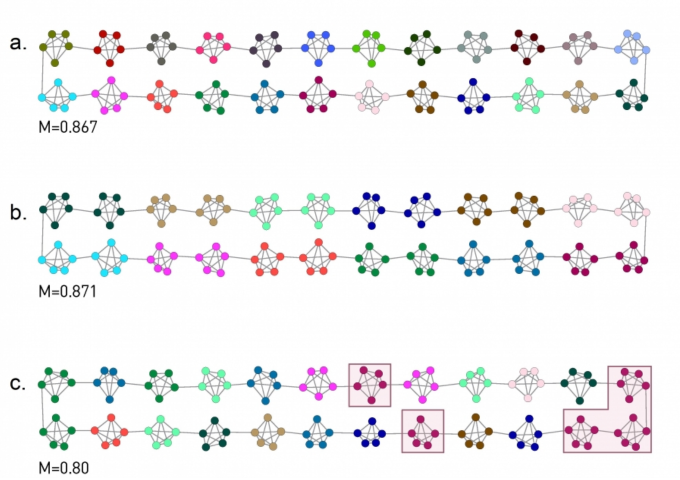
Greedy algorithm

- 1. We start with each node in its own community, $n_c = N$
- 2. For each pair of connected communities we calculate the change of modularity ΔM if we merge them. For the pair of communities for which we obtain the largest ΔM we merge them and thus $n_c \rightarrow n_c - 1$
- 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

Modularity limits: resolution

- Modularity maximization has preference toward big communities
- Modularity maximization cannot detect communities that are smaller than the resolution limit $k \leq \sqrt{2L}$, where k is total degree of new community
- Modified version of modularity $M = \sum_{c=1}^{n_c} [\frac{L_c}{L} - \gamma(\frac{k_c}{2L})^2]$ where γ - resolution

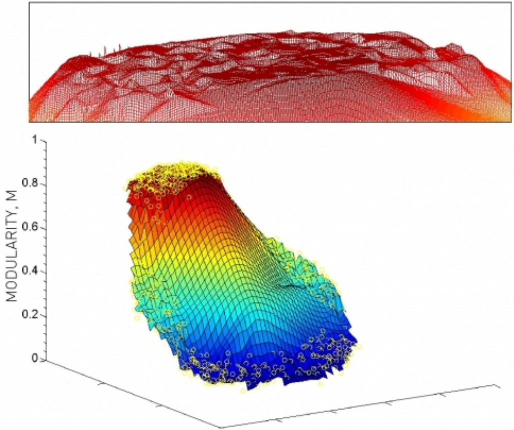
Modularity limits: maxima



Source: Network science book

Modularity limits: maxima

Modularity has large number of local maxima which are close



Source: Network science book

Modularity algorithms

- Greedy algorithm is computationally intensive
- Louvain algorithm is fast and can be used on large networks: more details in Advanced Topic 9.C (Section 9 Network science book)
- Louvain algorithm - fast, choice of resoultion, number of communities
- Other community detection algorithms that are not based on modularity: INFOMAP, OSLOM
- IMPORTANT: heuristic algorithms (Louvain, Infomap, Oslom) may find different communities for different runs

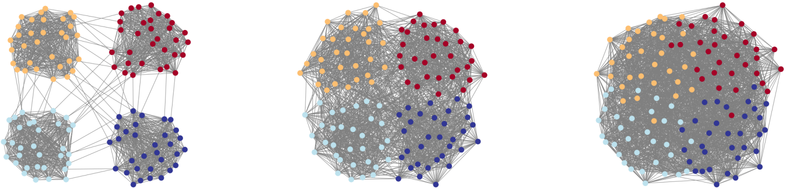
Approach

- We measure the accuracy of the algorithm by comparing it to:
  - Ground truth network partition
  - Network benchmark models
- Density of links $\mu = \frac{k^{ext}}{k^{ext} + k^{int}}$
- Accuracy measure *mutual information* I_n

Girvan-Newman (GN) Benchmark

- It is based on Erdos-Renyi graphs
- The number of communities n_c and number of nodes in each community $[N_1, N_2, \dots, N_{n_c}]$ is given; each nodes is assigned to one community
- There are two linking probabilities: p_{in} - probability that two nodes belonging to same community are connected, p_{out} - probability that two nodes from different communities are connected

Girvan-Newman (GN) Benchmark



$p_{in} = 0.9$ $p_{out} = 0.01$
$\mu = 0.024$

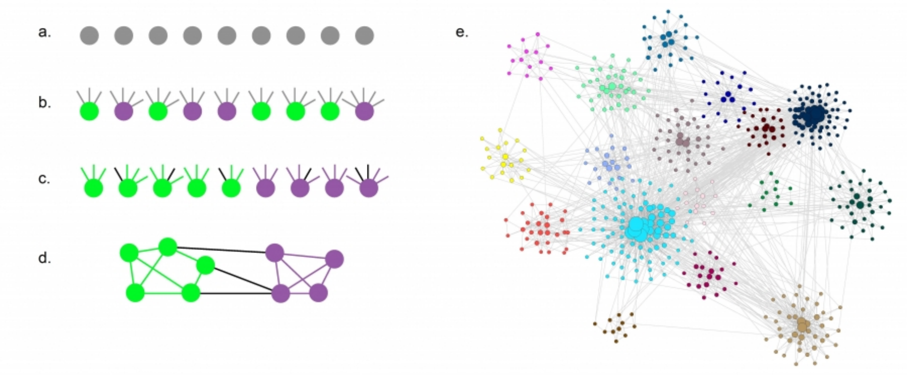
$p_{in} = 0.8$ $p_{out} = 0.1$
$\mu = 0.28$

$p_{in} = 0.7$ $p_{out} = 0.2$
$\mu = 0.47$

Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

- We start with N isolated node
- Each nodes is assigned to a community of size N_c, where $P(n_c \sim N_c^{-\xi})$
- To each node we assign degree k_i, where $P(k_i) \sim k_i^{-\gamma}$
- Each node receives internal degree $(1-\mu)k_i$ and external degree μk_i
- All stubs of nodes of the same community are randomly attached to each other, until no more stubs are *free*. In this way we maintain the sequence of internal degrees of each node in its community. The remaining μk_i stubs are randomly attached to nodes from other communities

Lancichinetti-Fortunato-Radicchi (LFR) Benchmark

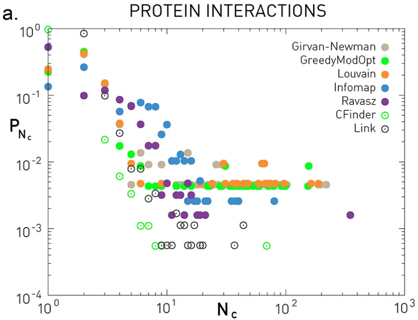


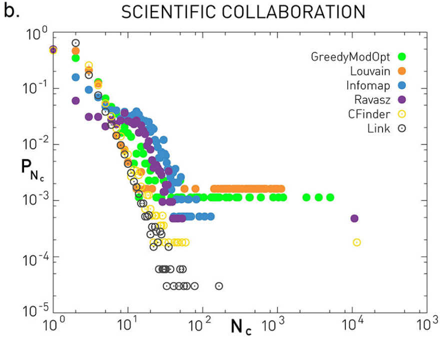
Source: Network science book

Mutual information

- $p(c)$ - a probability that randomly selected node in a network belongs to community c
- each partition c_i has it own probability $p(c_i)$
- $p(c_1, c_2)$ a probability that randomly chosen node belongs to community c_1 in a partition 1 and community c_2 in a partition 2
- $I_n = \frac{\sum_{c1,c2} p(c1,c2) \log_2 \frac{p(c_1,c_2)}{p(c_1)p(c_2)}}{\frac{1}{2}(H(\{p_{c_1}\})+H(\{p_{c_2}\})}$ where $H(\{p_c\}) = -\sum_c p(c) \log_2 p(c)$ is Shannon entropy
- $I_n = 1$ partitions are identical, $I_n = 0$ - partitions are independent

Community size distribution





Source: Network science book

Other types of communities

- Overlaping communities - nodes can belong to more than one community
- Weighted communities - communities in weighted networks
- Commmunities in directed networks also exist
- Some algorithms are able to find communities in binary, weighted and directed networks; some are specialized for specific type; overlaping communities usually demand special algorithm

Further reading

- Network science book Caphter 9:
  http://networksciencebook.com/chapter/9

- Fortunato et al., Community detection in networks: A user guide. Physics reports, 659, 1-44 (2016),
  https://arxiv.org/pdf/1608.00163.pdf