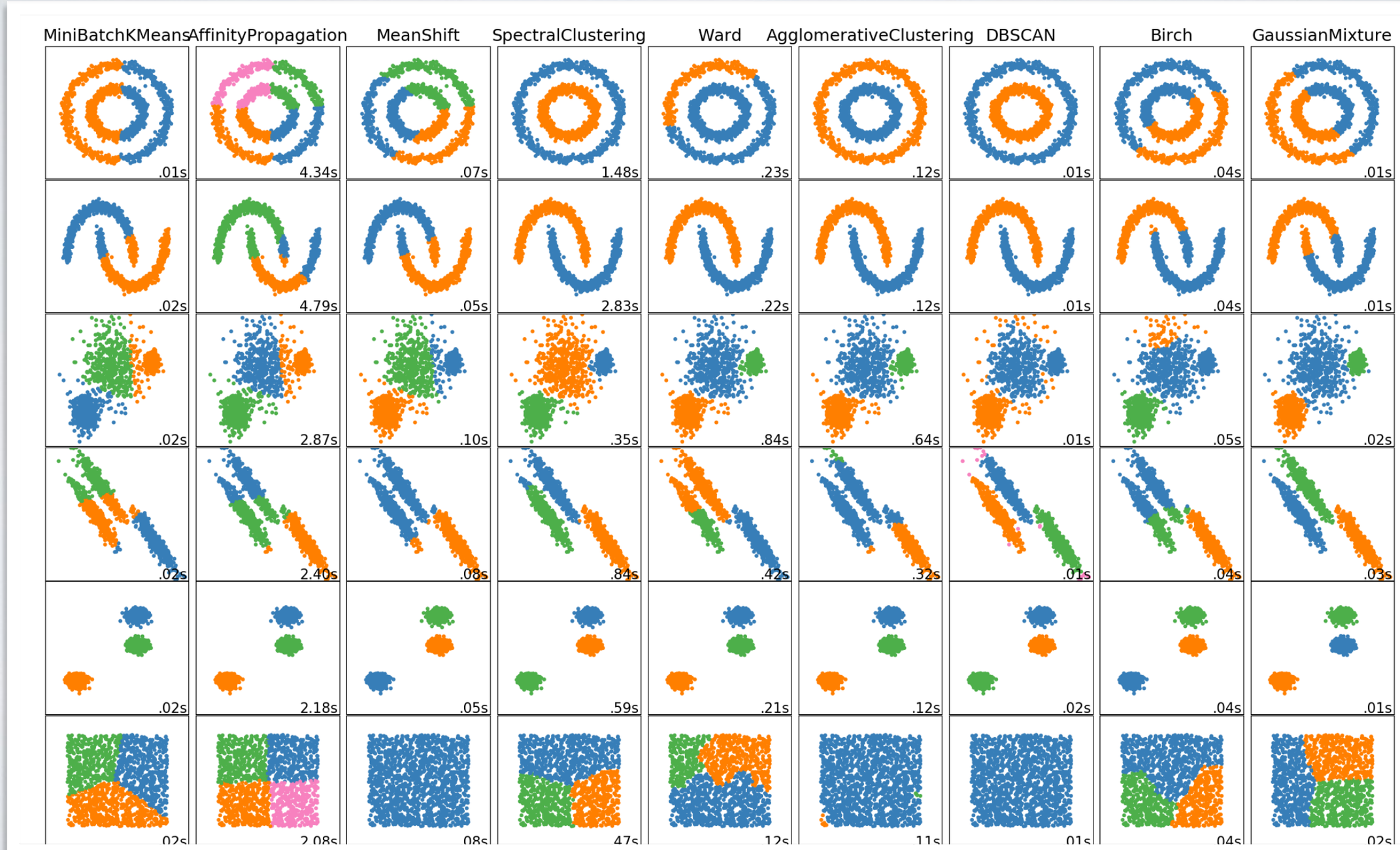
COMMUNITY DETECTION (GRAPH CLUSTERING)

COMMUNITY DETECTION

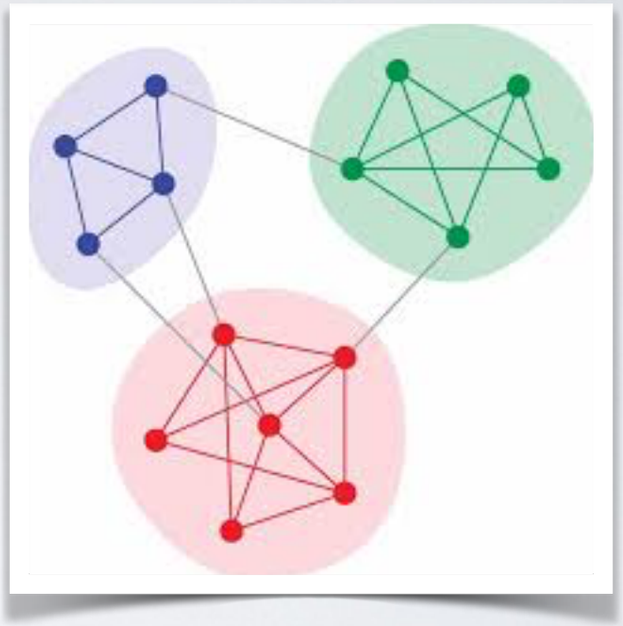
- Community detection is equivalent to "clustering" in unstructured data

- Clustering: unsupervised machine learning
  - Find groups of elements that are similar to each other
    - People based on DNA, apartments based on characteristics, etc.
  - Hundreds of methods published since 1950 (k-means)
  - Problem: what does "similar to each other" means ?

COMMUNITY DETECTION



COMMUNITY DETECTION



- Community detection:
  - Find groups of nodes that are:
    - Strongly connected to each other
    - Weakly connected to the rest of the network
    - Ideal form: each community is 1)A clique, 2) A separate connected component
  - No formal definition
  - Hundreds of methods published since 2003

WHY COMMUNITY DETECTION ?

- One of the key properties of complex networks was
  - High clustering coefficient
  - (friends of my friends are my friends)

- Different from random networks. How to explain it ?
  - Watts strogatz (spatial structure?)

- => In real networks, presence of dense groups: communities
  - Small, dense (random) networks have high density.
  - Large networks could be interpreted as aggregation of smaller, denser networks, with much fewer edges between them

SOME HISTORY

- The *graph partitioning problem* was a classic problem in graph theory

- It goes like this:
  - How to split a network in **k** equal parts such that there is a minimal number of edges between parts.
  - Variants were proposed:
    - What if partitions are not exactly same size ?
    - What if the number of parts is not exactly k ?
    - …

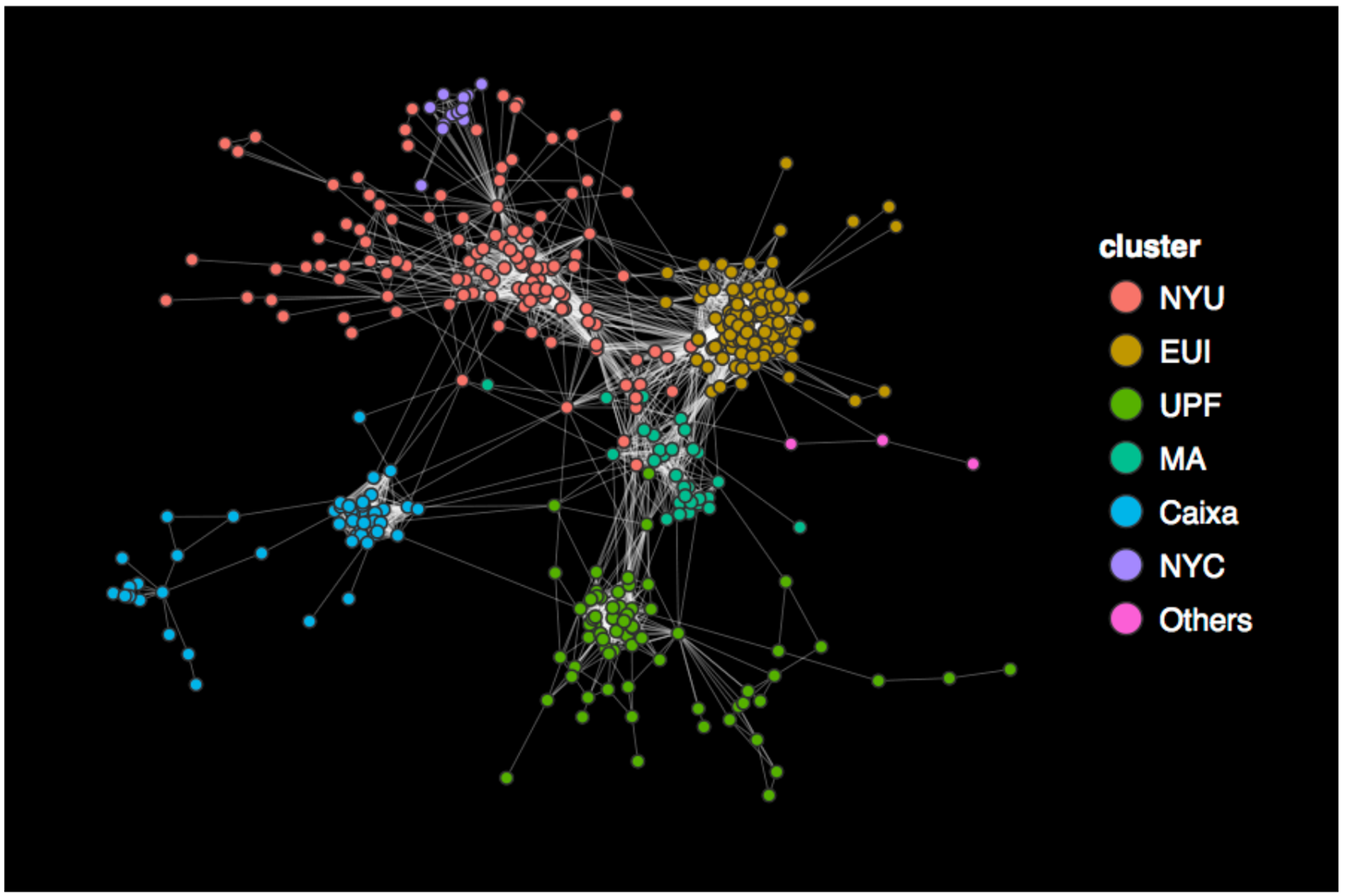
SOME HISTORY

- Then in 2002, [Girvan & Newman 2002], introduction of the problem of "community discovery":
  - Observation that social networks are very often composed of groups
  - The number and the size of these groups is not known in advance
  - Can we design an algorithm to discover automatically those groups ?

Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." *Proceedings of the national academy of sciences* 99.12 (2002): 7821-7826.

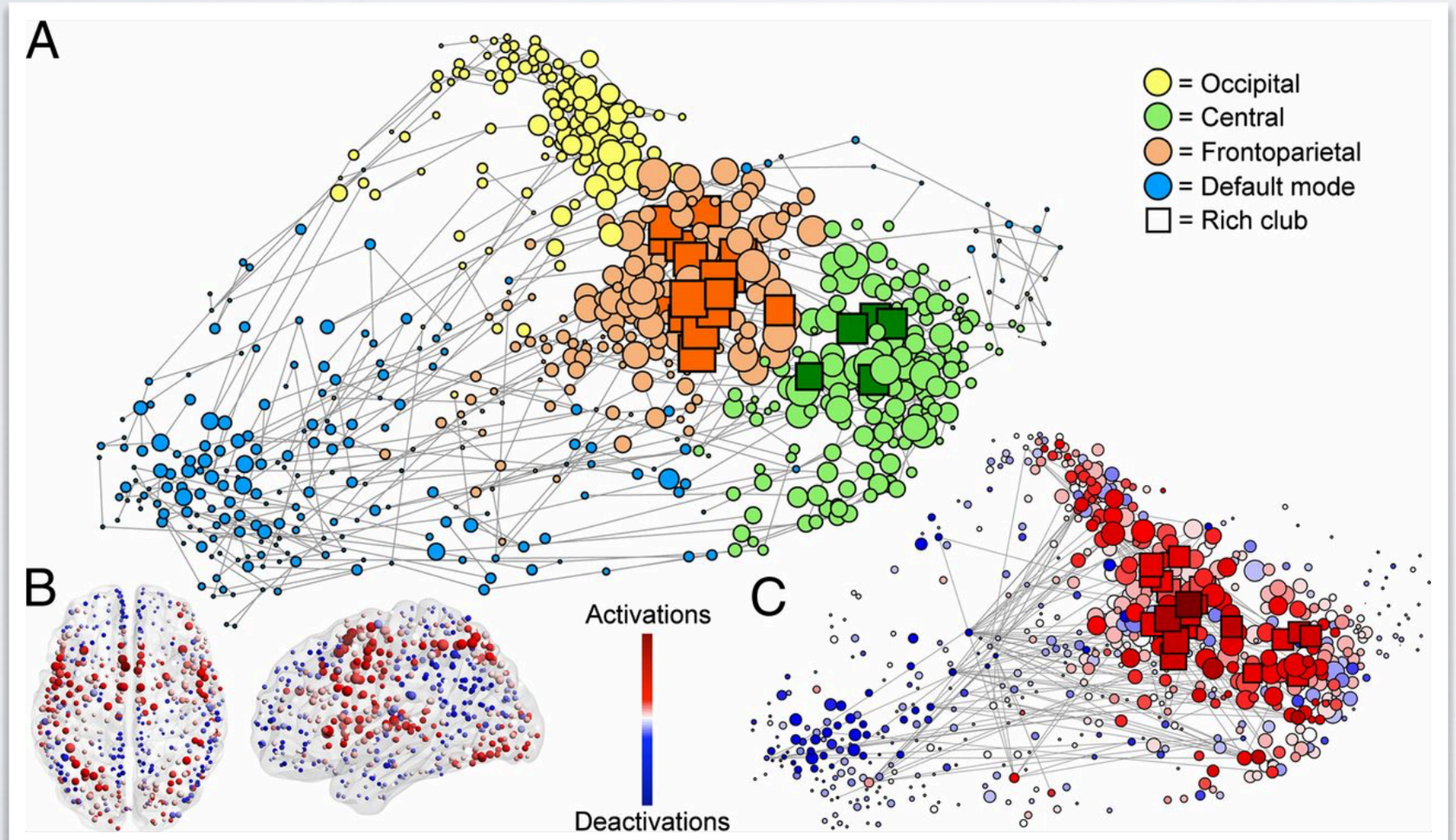
COMMUNITY STRUCTURE IN REAL GRAPHS

- If you plot the graph of your facebook friends, it looks like this



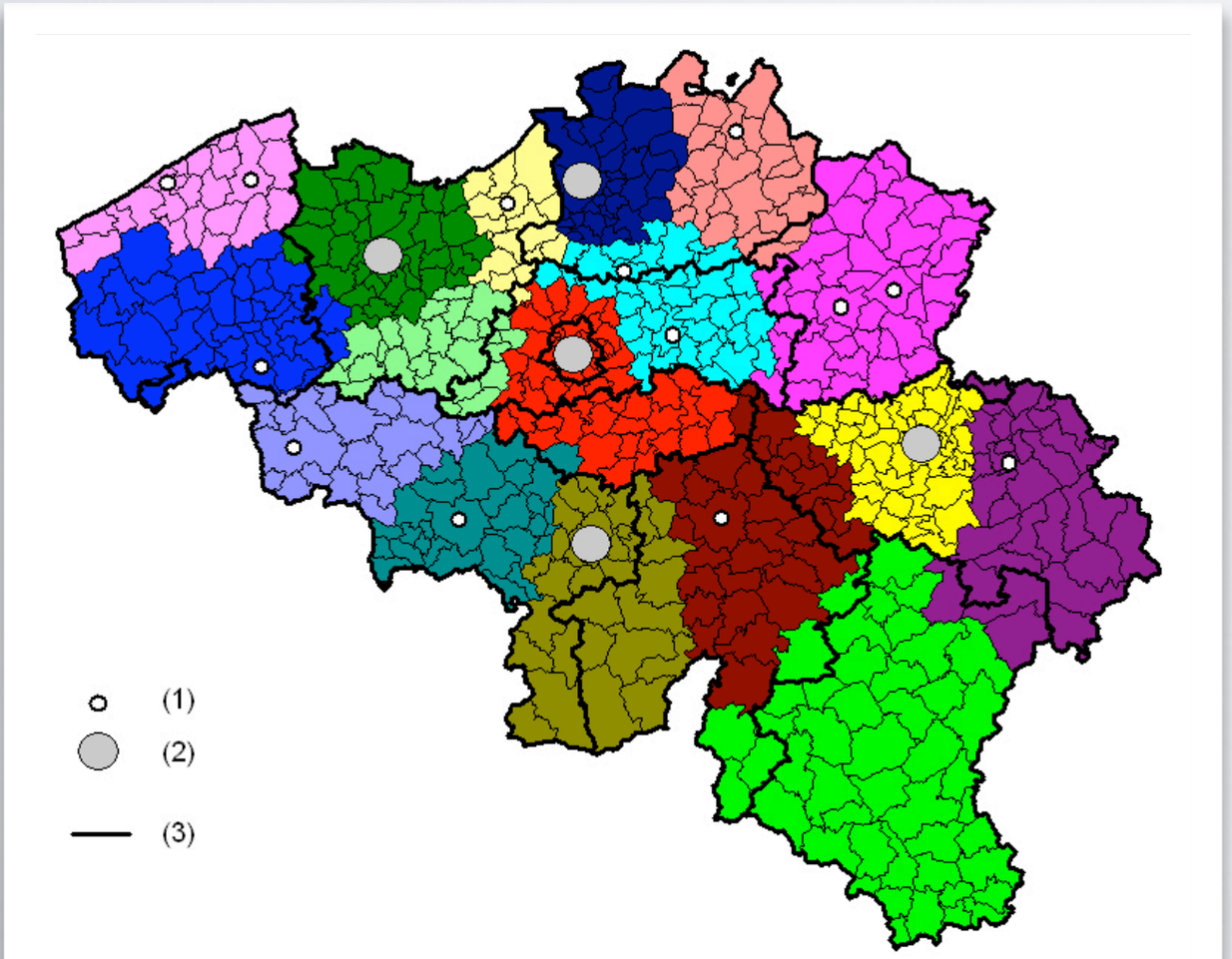
COMMUNITY STRUCTURE IN REAL GRAPHS

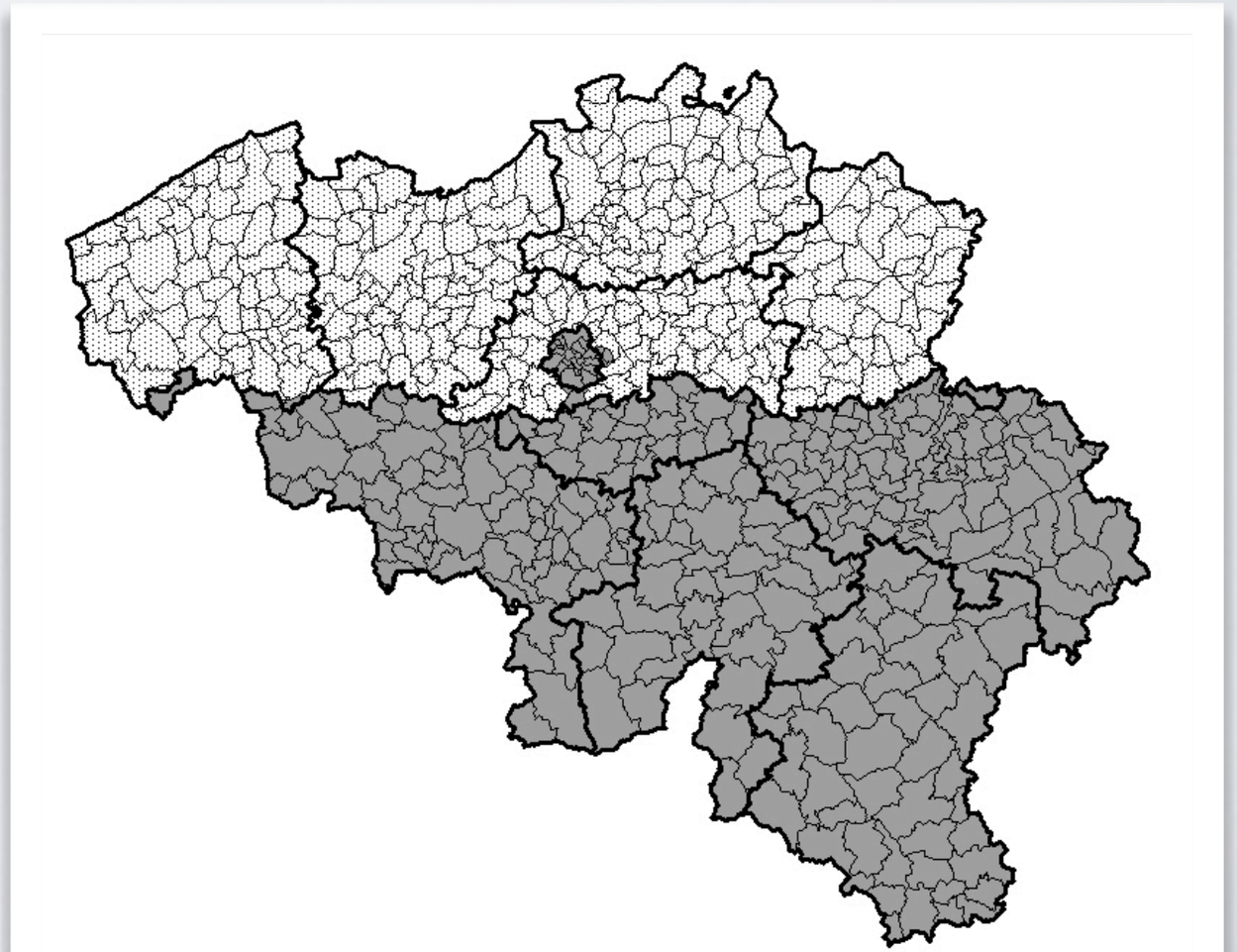
- Connections in the brain ?



COMMUNITY STRUCTURE IN REAL GRAPHS

- Phone call communications in Belgium ?

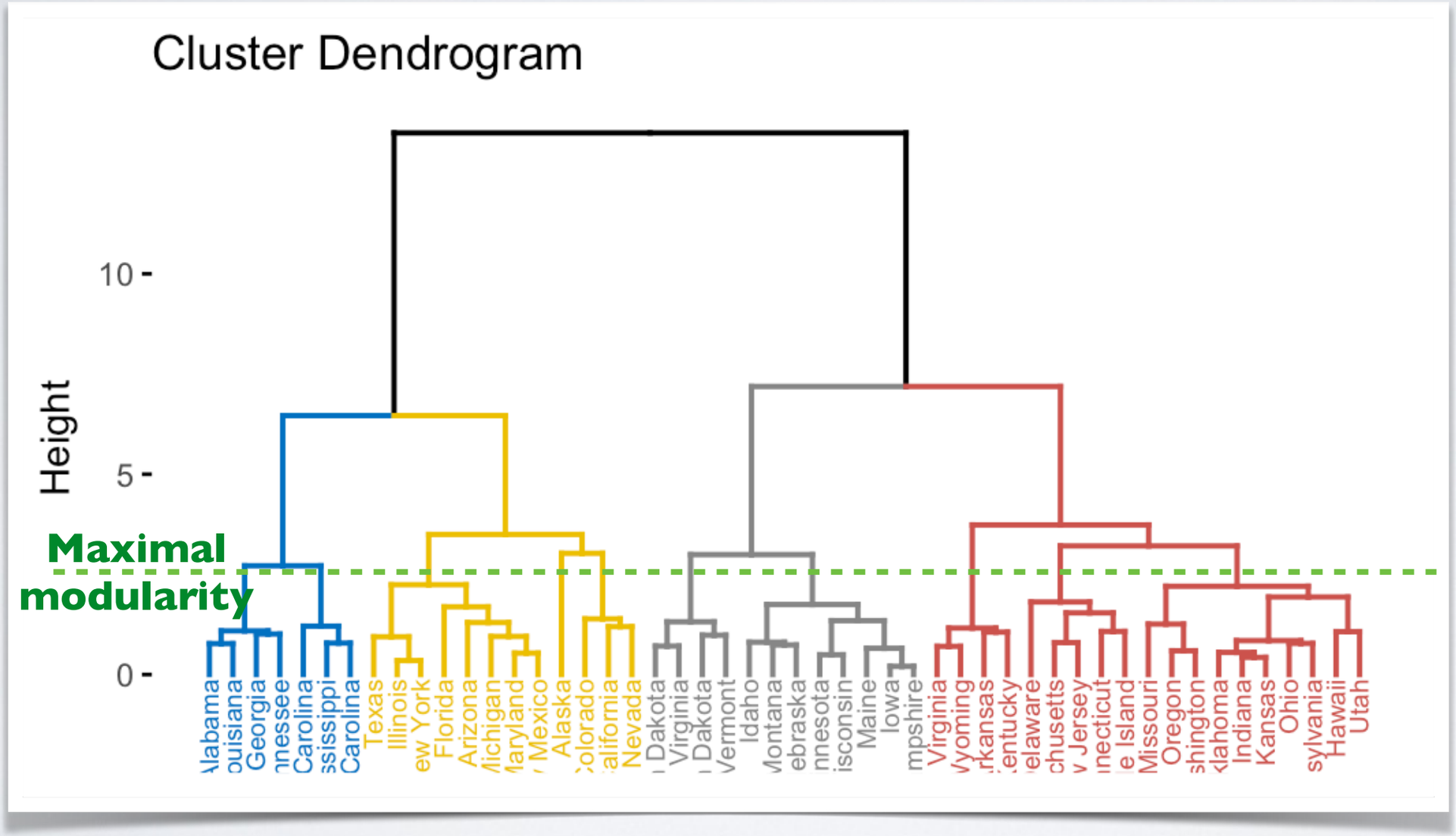




FIRST METHOD BY GIRVAN & NEWMAN

- 1)Compute the betweenness of all edges
- 2)Remove the edge of highest betweenness
- 3)Repeat until all edges have been removed
  - Connected components are communities
- => It is called a *divisive* method
- =>What you obtain is a dendrogram
- How to cut this dendrogram at the *best* level ?

FIRST METHOD BY GIRVAN & NEWMAN



FIRST METHOD BY GIRVAN & NEWMAN

- Introduction of the **Modularity**

- The modularity is computed for a partition of a graph
  - (each node belongs to one and only one community)

- It compares :
  - The **observed** *fraction of edges inside communities*
  - To the **expected** *fraction of edges inside communities* in a random network

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

Original formulation

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

Sum over all pairs of nodes

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

1 if in same community

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

1 if there is an edge between them

MODULARITY

$$Q = \frac{1}{(2m)} \sum_{vw} \left[A_{vw} - \frac{k_v k_w}{(2m)} \right] \delta(c_v, c_w)$$

Probability of an edge in
a configuration model

MODULARITY

Can also be defined as a sum by community

$$Q = \frac{1}{L} \sum_{i=1}^{|C|} (L_i - \frac{1}{2} K_i^2)$$

with $L_i = L(H(c_i))$ the number of edges inside community i and $K_i = \sum_{u \in c_i} k_u$ the sum of degrees of nodes in community i.

MODULARITY

- Modularity compares the observed network to a **null model**
  - Usually the configuration model
    - Multi-edges and loops are allowed
  - Other models could be used, such as ER random graphs.
- Natural extension to weighted/multi-edge networks

FIRST METHOD BY GIRVAN & NEWMAN

- Back to the method:
  - Create a dendrogram by removing edges
  - Cut the dendrogram at the best level using modularity

- =>In the end, your objective is… to optimize the Modularity, right ?

- Why not optimizing it directly !

MODULARITY OPTIMIZATION

- From 2004 to 2008: The golden age of Modularity
- Scores of methods proposed to optimize it
  - Graph spectral approaches
  - Meta-heuristics approches (simulated annealing, multi-agent…)
  - Local/Gloabal approaches…
- => 2008: the Louvain algorithm

LOUVAIN ALGORITHM

- Simple, greedy approach
  - Easy to implement
  - Fast
- Yields a hierarchical community structure
- Beat state of the art on all aspects (when introduced)
  - Speed
  - Max modularity obtained
  - Do not fall in some traps (see later)

LOUVAIN ALGORITHM

- Each node start in its own community

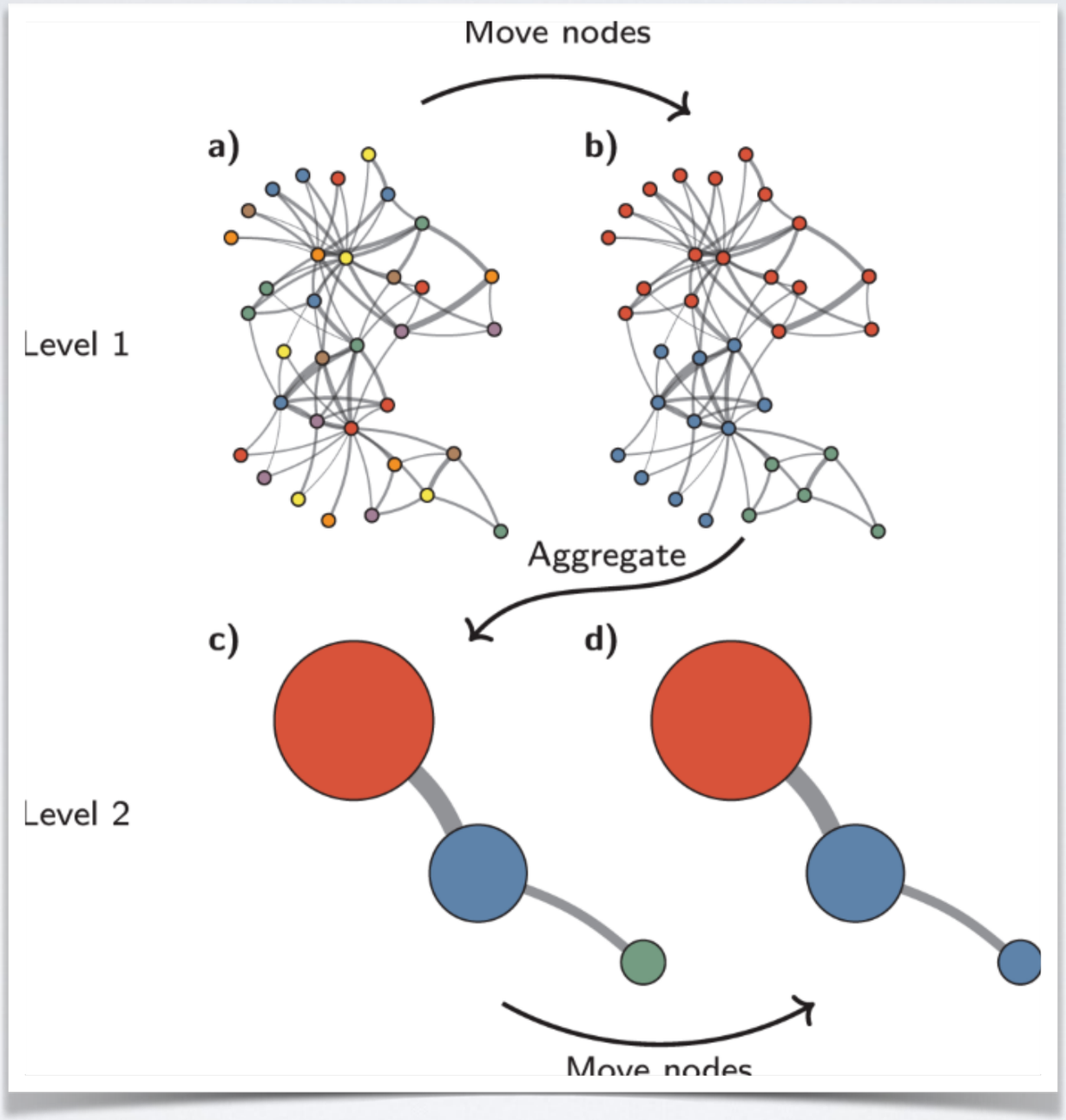
- Repeat until convergence
  - FOR each node:
    - FOR each neighbor:
      if adding node to its community increase modularity, do it

- When converged, create an *induced network*
  - Each community becomes a node
  - Edge weight is the sum of weights of edges between them

- Trick: Modularity is computed *by community*
  - Global Modularity = sum of modularities of each community

Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.

LOUVAIN ALGORITHM



Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.

RESOLUTION LIMIT

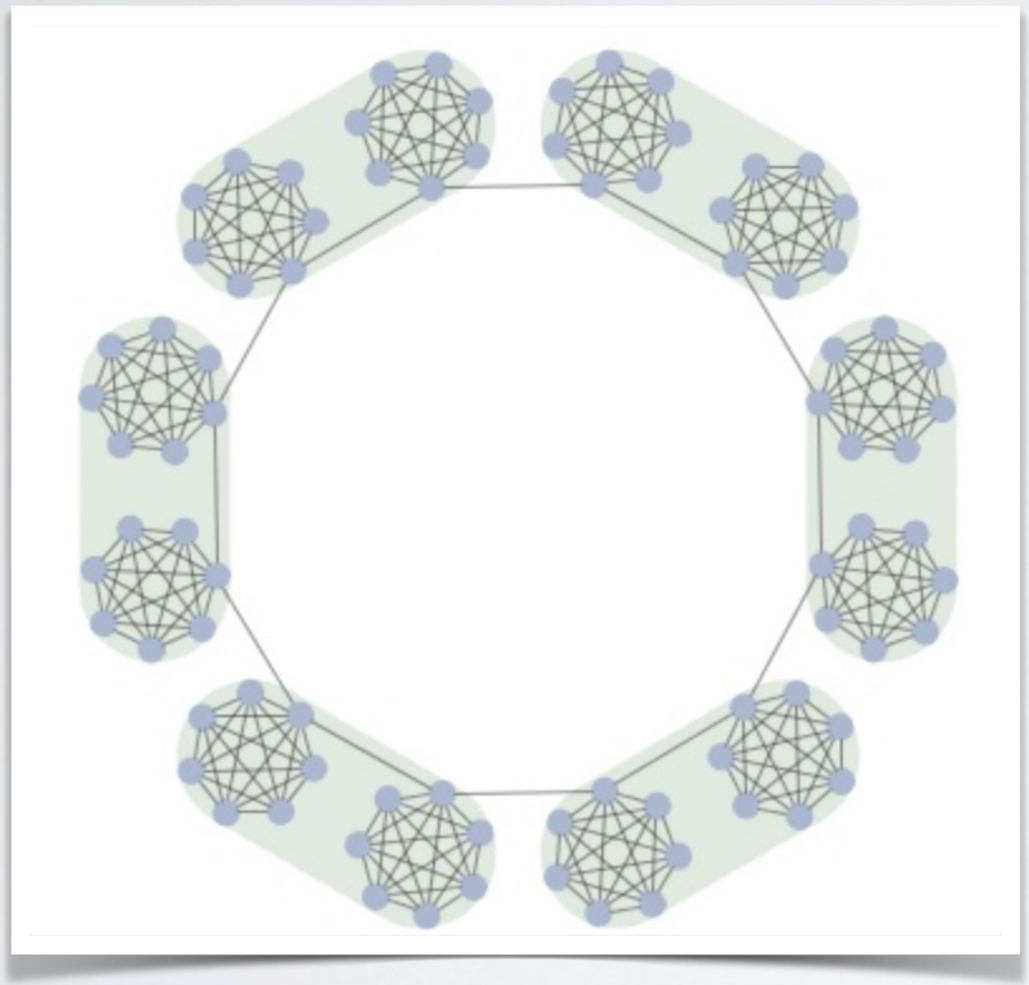
- Modularity == Definition of good communities ?

- 2006-2008: series of articles [Fortunato,Lancicchinetti,Barthelemy]
  - Resolution limit of Modularity

- Let's see an example

Fortunato, Santo, and Marc Barthelemy. "Resolution limit in community detection." *Proceedings of the national academy of sciences* 104.1 (2007): 36-41.

RESOLUTION LIMIT



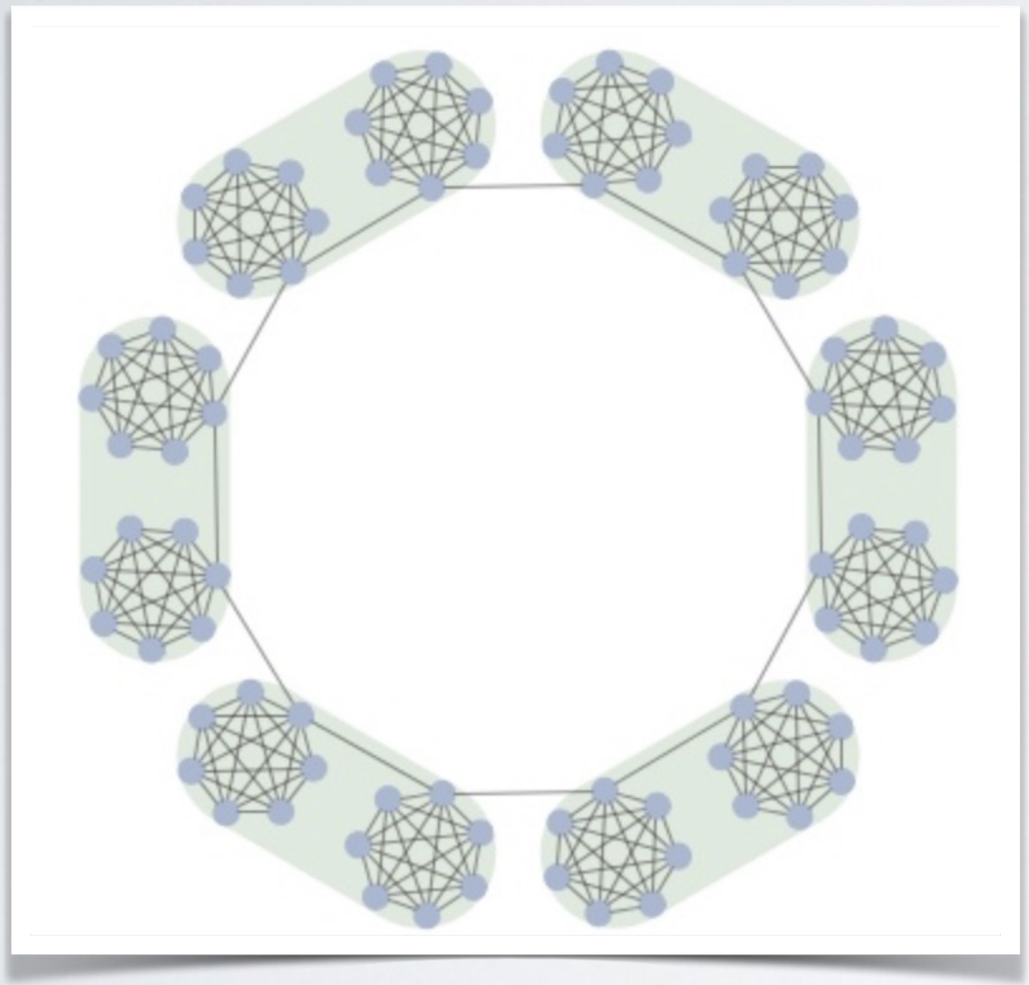
Let's consider a ring of cliques

Cliques are as dense as possible

Single edge between them:
=>As separated as possible

Any acceptable algorithm=>Each clique is a community

RESOLUTION LIMIT



But with modularity:

Small graphs=> OK

Large graphs=>
The max of modularity obtained by merging cliques

RESOLUTION LIMIT

- Discovery that Modularity has a "favorite scale":

- For a graph of given **density** and **size:**
  - Communities cannot be smaller than a fraction of nodes
  - Communities cannot be larger than a fraction of nodes

- Modularity optimisation will never discover
  - Small communities in large networks
  - Large communities in small networks

RESOLUTION LIMIT

- Multi-resolution modularity

$$\sum_{i}^{c} e_{ii} - a_i^2 \quad \Rightarrow \quad \sum_{i}^{c} e_{ii} - \lambda a_i^2$$

λ = Resolution parameter

More a patch than a solution…

OTHER WEAKNESSES

- Modularity has other controversial/not-intuitive properties:
  - Global measure => a difference in one side of the network can change communities at the other end (imagine a growing clique ring…)
  - Unable to find no community:
    - Network without community structure: Max modularity for partitions driven by random noise

- To this day, Louvain and modularity remain most used methods
  - Results are usually "good"/useful
  - It's a "standard" tool, like k-means or linear regression
  - Some newer methods gain popularity (SBM, Leiden,…)

ALTERNATIVES

- 1000+ Algorithms published, and counting
- What unfortunately many methods still do:
  - They define their own criteria of good communities without being grounded on existing literature
  - They show empirically on a few networks using a single validation method that their method is better than Louvain
- Common saying:
  - "No algorithm is better than other, it depends on the type of network"(no free-lunch theorem)
  - "The best method depends on the objective"

ALTERNATIVES

- Most serious alternatives (in my opinion)
  - Infomap (based on information theory —compression)
  - Stochastic block models (bayesian inference)

- These methods have a clear definition of what are good communities. Theoretically grounded

INFOMAP

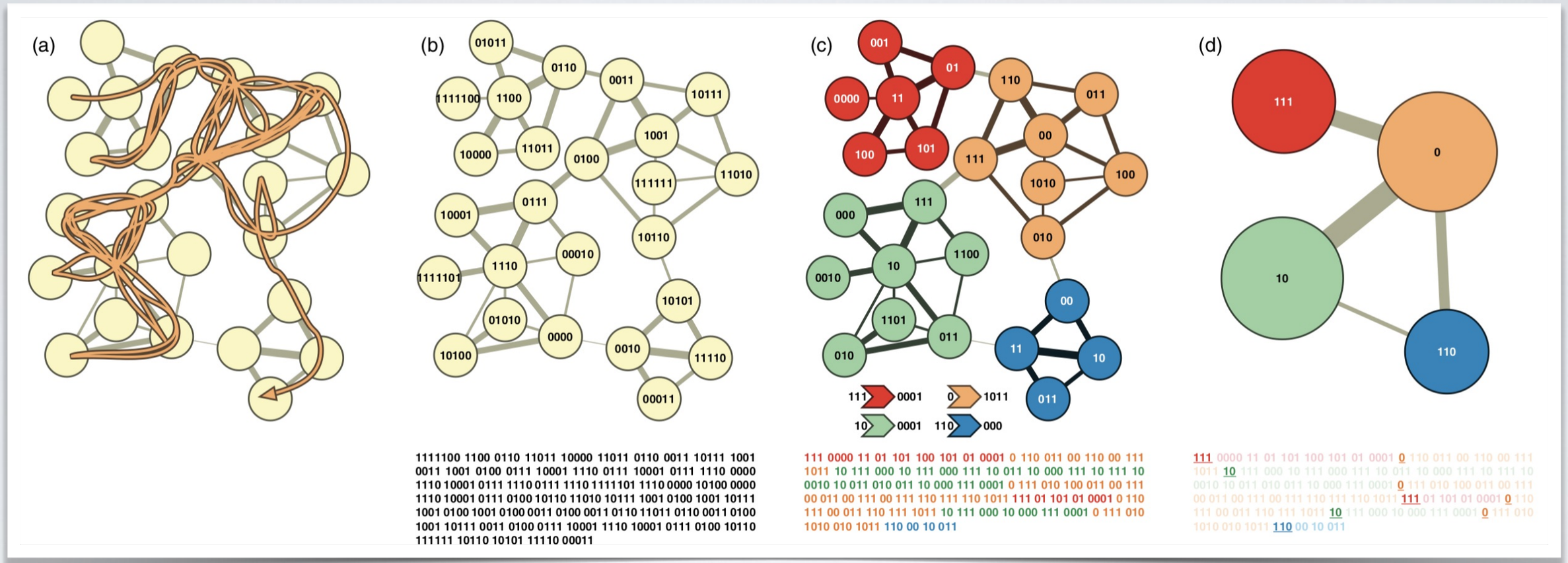
- [Rosvall & Bergstrom 2009]

- Find the partition minimizing the *description* of any *random walk* on the network

- We want to *compress* the description of random walks

Rosvall, Martin, and Carl T. Bergstrom. "Maps of random walks on complex networks reveal community structure." *Proceedings of the National Academy of Sciences* 105.4 (2008): 1118-1123.

INFOMAP



Random walk

Description Without Communities

With communities

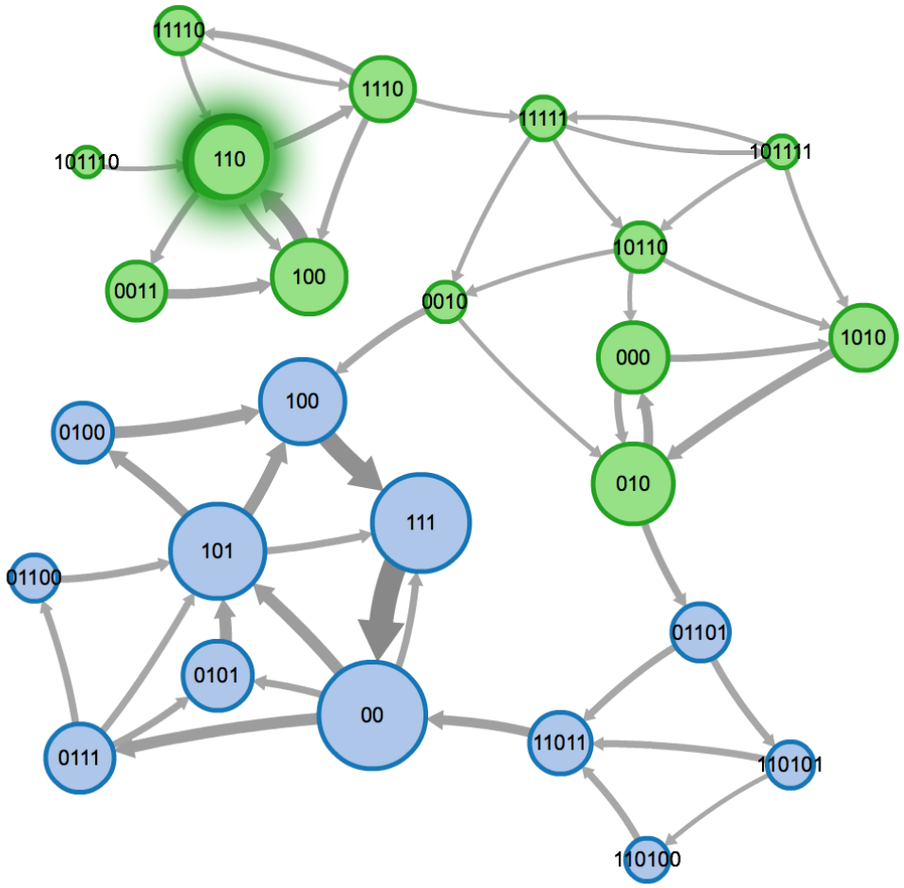
Huffman coding: short codes for frequent items

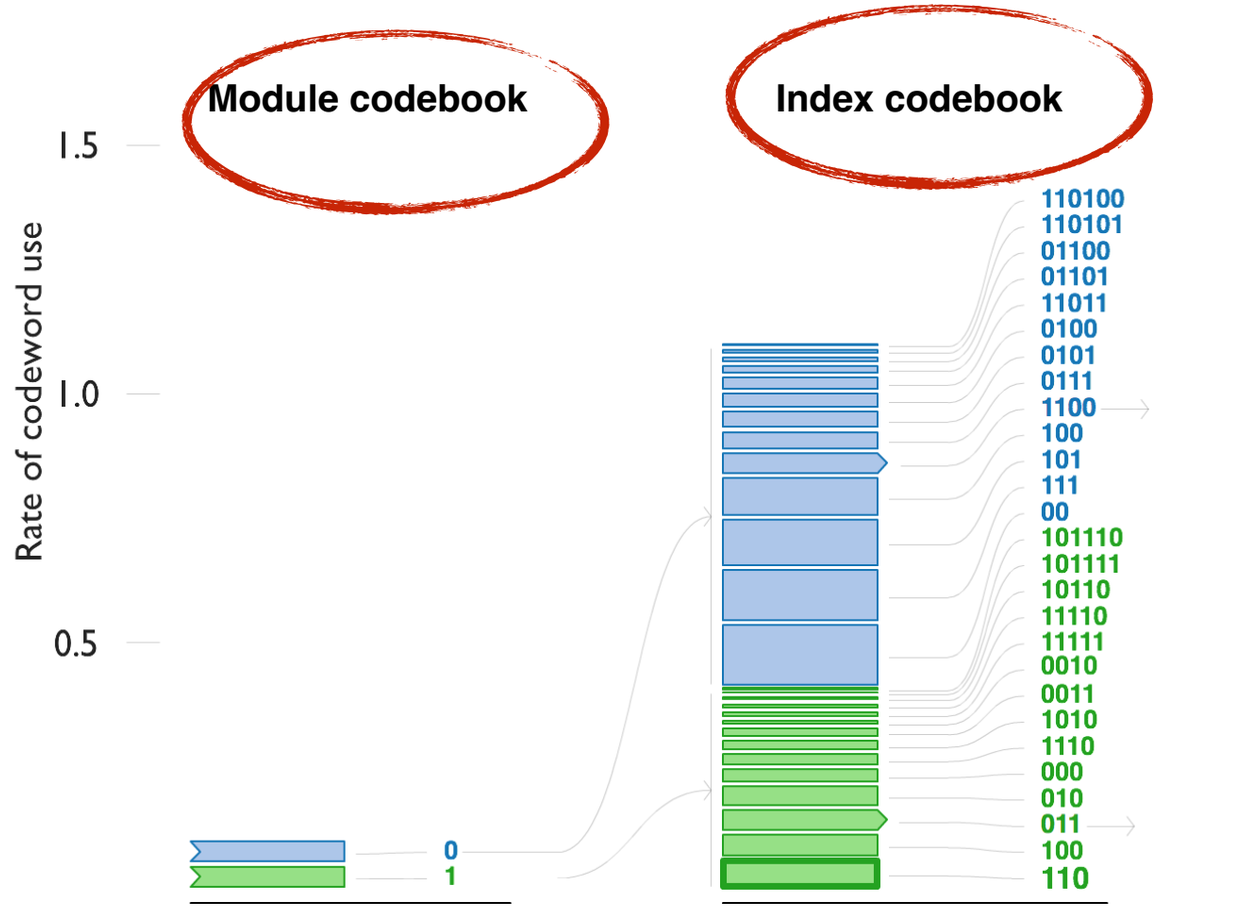
Prefix free: no code is a prefix of another one (avoid fix length/separators)

The Infomap method

Rosvall&Bergstrom (2008)

Finding a compressed description of a random walk taking place on a graph





Encoded random walker

Start/stop | Step | Reset

10010**0110**10011100011001010100**1100**1110111011110001101000010100100000100000101011111011010100100000100001010010**0110**10**1100**1001110011000111001101011**0110**1101001101100011101011011001110011100**1100**1001110011010011010111011011101111011101111011**0110**100111000111010**1100**10011100110

CSV

Speed

Node size shows rate

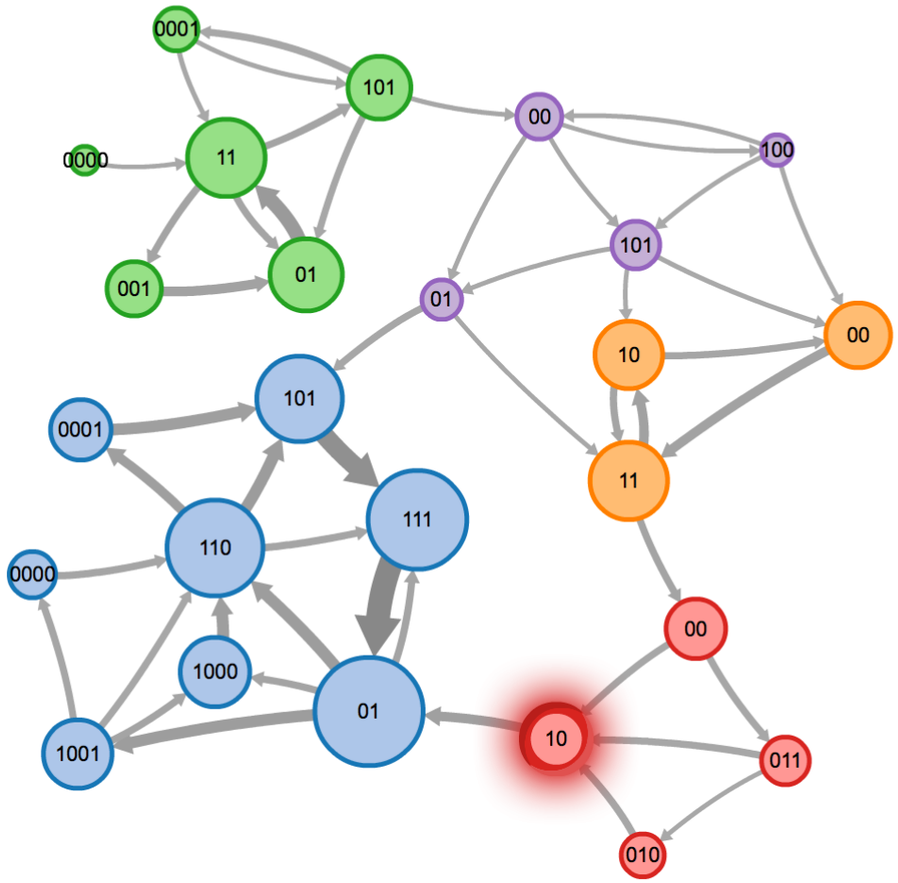
Link size shows flow

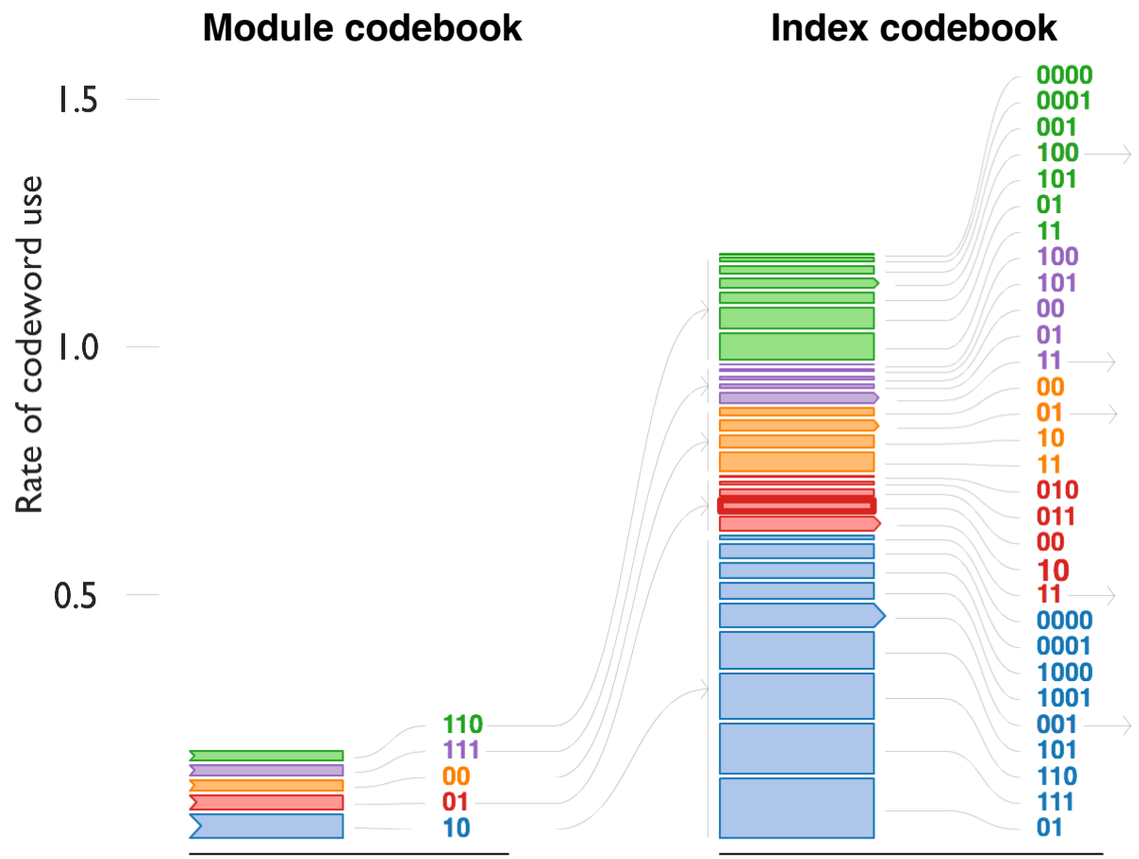
mapequation.org

The Infomap method

Rosvall&Bergstrom (2008)

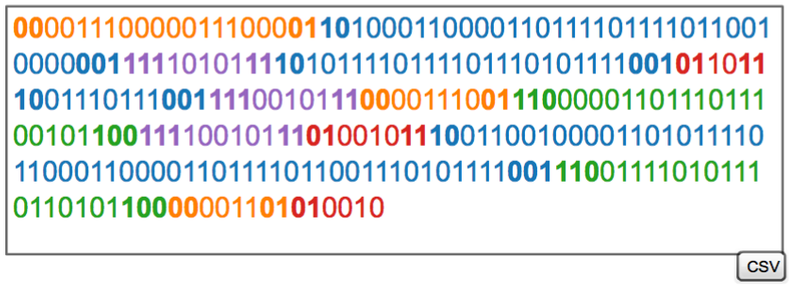
Finding a compressed description of a random walk taking place on a graph





Encoded random walker

Start/stop | Step | Reset



Speed

Node size shows rate

Link size shows flow

mapequation.org

The Infomap method

Finding the optimal partition M:

- Shannon's source coding theorem (Shannon's entropy)

for a probability distribution P = {p_i} such that $\Sigma_i p_i = 1$, the lower limit of the per-step code-length is

$$L(\mathcal{P}) = H(\mathcal{P}) \equiv -\sum_i p_i \log p_i$$

- Minimise the expected description length of the random walk

Sum of Shannon entropies of multiple codebooks weighted by the rate of usage

probability of between modules movements of a RW, i.e. the rate of usage of the index codebook

probability of within modules movements of a RW, i.e. the rate of usage of the module codebook

$$L(\mathbf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i)$$

Expected decryption length of partition M

Entropy of movement between modules, i.e. the frequency weighted average length of codewords

Entropy of movement inside modules, i.e. the frequency weighted average length of codewords in the module codebook

Rosvall, Martin, and Carl T. Bergstrom. "Maps of random walks on complex networks reveal community structure." *Proceedings of the National Academy of Sciences* 105.4 (2008): 1118-1123.

INFOMAP

- To sum up:
  - Infomap defines a *quality function* for a partition different than modularity
  - Any algorithm can be used to optimize it (like Modularity)

- Advantage:
  - Infomap can recognize random networks (no communities)

STOCHASTIC BLOCK MODELS

- Stochastic Block Models (SBM) are based on statistical models of networks

- They are in fact more general than usual communities.

- The model is:
  - Each node belongs to 1 and only 1 community
  - To each pair of communities, there is an associated density (probability of each edge to exist)

Stochastic block models

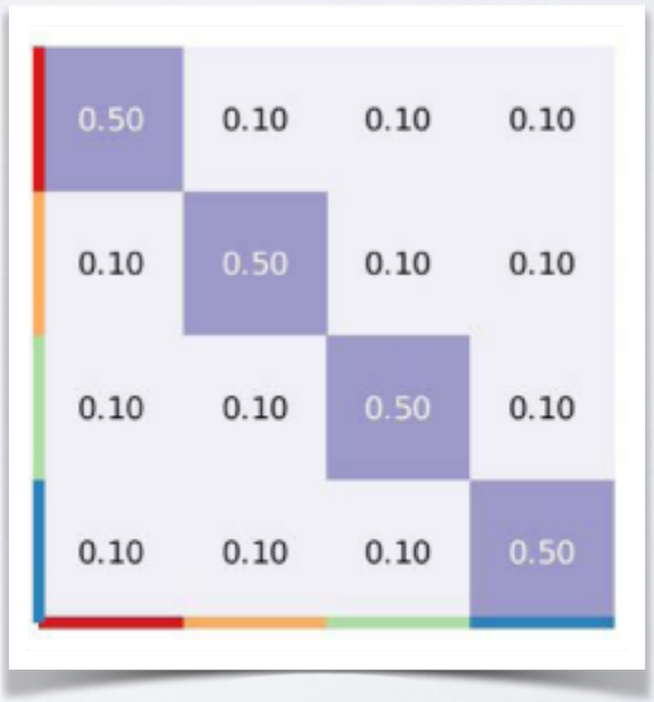
Stochastic Block Models (SBM)

A stochastic block model is a random graph model defined by:

b	$n \times 1$ vector such as b_i describes the index of the block of node i.
E	$k \times k$ **stochastic block matrix**, such as E_{ij} gives the number of edges between blocks i and j (or the probability to observe an edge between any pair of nodes chosen with one node in each of the two blocks).

b

E



Stochastic block models

Stochastic Block Models (SBM)

A stochastic block model is a random graph model defined by:

b	$n \times 1$ vector such as b_i describes the index of the block of node i.
E	$k \times k$ **stochastic block matrix**, such as E_{ij} gives the number of edges between blocks i and j (or the probability to observe an edge between any pair of nodes chosen with one node in each of the two blocks).

Generating networks

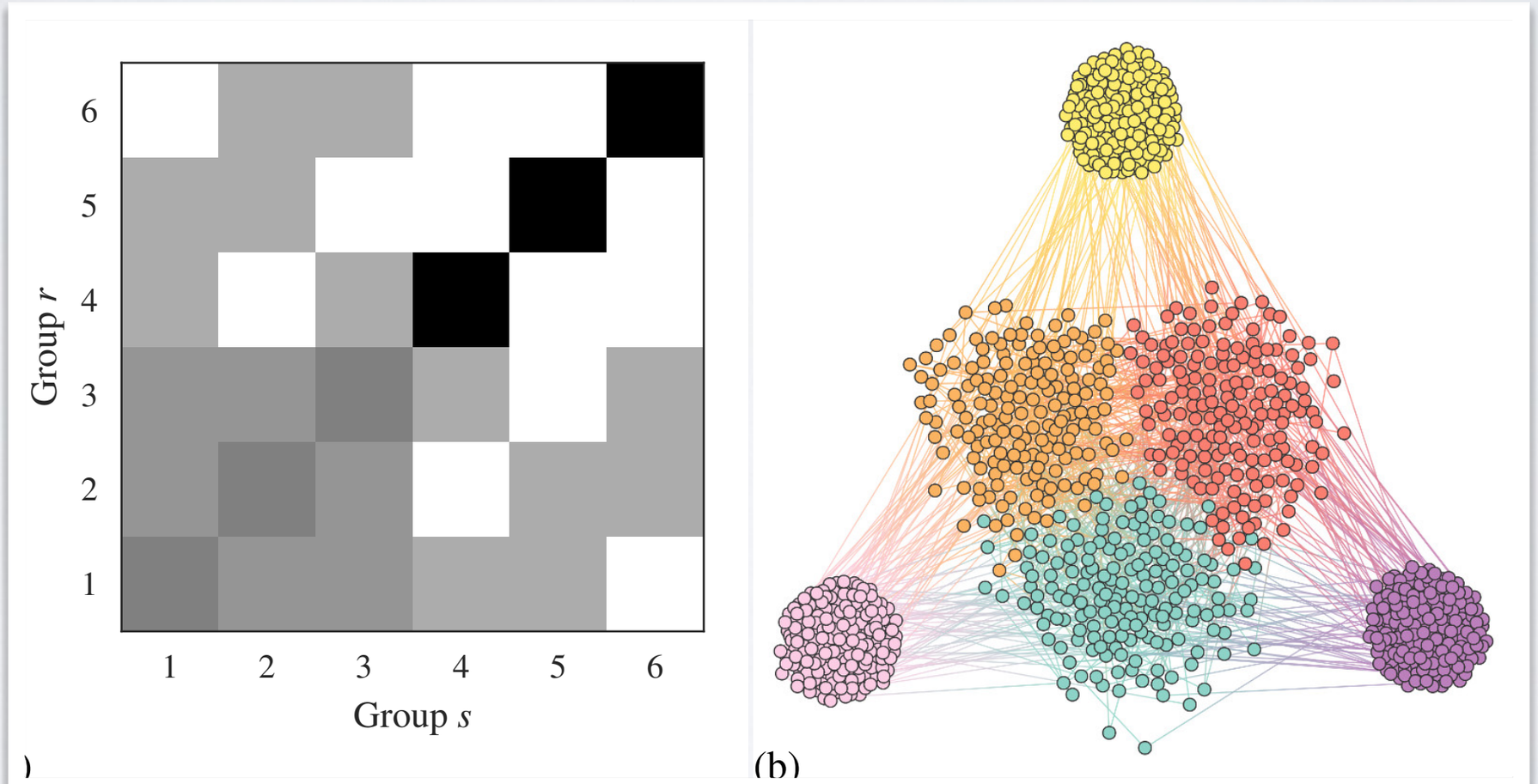
1. Take n disconnected nodes
2. Connect each $u,v \in V$ nodes with probability $E_{b(u),b(v)}$

Properties:

- Every vertices in a same module are statistically equivalent
- Vertices in a module are connected by a random graph
- Emergent degree distribution is a combination of Poisson distributions

STOCHASTIC BLOCK MODELS

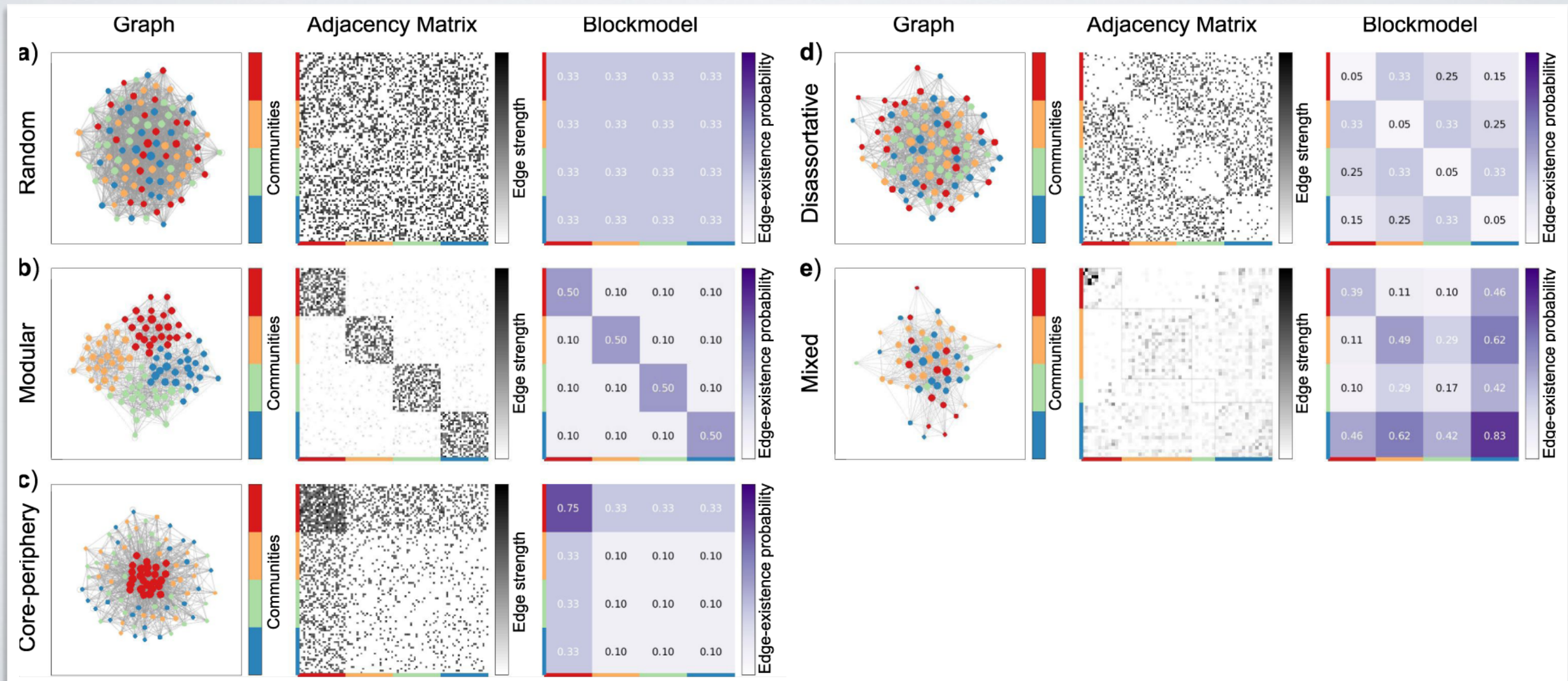
- SBM can represent different things:
  - Associative SBM: density inside nodes of a same communities >> density of pairs belonging to different communities.



https://arxiv.org/pdf/1705.10225.pdf

STOCHASTIC BLOCK MODELS

- SBM can represent different things:
  - Associative SBM: density inside nodes of a same communities >> density of pairs belonging to different communities.



STOCHASTIC BLOCK MODELS

- SBM can represent different things:
  - Associative SBM: density inside nodes of a same communities >> density of pairs belonging to different communities.
- This is very powerful and potentially relevant
- Problem: Often hard to interpret in real situations.
  - SBM can be "constrained": we impose that intra d.>inter d.

SBM SIMPLE GRAPHS

- Probability to generate a given graph with a given b, E (likelihood $(A \mid b, E)$)
  - Note that for a given A, b, maximizing E is trivially obtained as the fraction of edges existing between each block pair.

- Assuming a simple graph, Bernouilli distribution
  - Product of probability for each observed edge to be present and for each non existing edge to be not present
  - $$p(A \mid b, E) = \prod_{i<j} \begin{cases} E_{b_i b_j} & \text{if } A_{ij} = 1 \\ 1 - E_{b_i b_j} & \text{if } A_{ij} = 0 \end{cases}$$

SBM POISSON

- Different statistical models can be used
  - The poisson model is a popular choice (simple computation)

- Poisson distribution of edges:
  - Assume possibility of multiple edges between nodes
    - But little difference in practice for sparse graphs

$$P(A \mid b, E) = \prod_{i<j} \frac{(E_{b_i b_j})^{A_{ij}}}{A_{ij}!} e^{-E_{b_i b_j}}$$

    - Since Poisson PMF: $P(k) = \frac{\lambda^k}{k!} e^{-\lambda}$
    - With λ the probability to observe an edge
    - Here I ignore self-loops for simplicity

Karrer, B., & Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, *83*(1), 016107.

DC-SBM

- As with modularity, we would like to preserve nodes degrees
  - Else, high-degree nodes tend to end up in a same community, since they are "densely connected" (but expected according to degree)

- $$P(A \mid b, E, \theta) = \prod_{i<j} \frac{(\theta_i \theta_j E_{b_i b_j})^{A_{ij}}}{A_{ij}!} e^{-\theta_i \theta_j E_{b_i b_j}}$$
  - θ is a vector corresponding to nodes degree
  - As E, optimal θ can be deducted from b, A:
    - $\hat{\theta}_i = \frac{k_i}{\kappa_{b_i}}$ with κ_{b_i} the sum of degrees in i's cluster

Karrer, B., & Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, *83*(1), 016107.

SBM INFERENCE

- Likelihood maximization:
  - $\hat{b} = \text{argmax}_b \, p(A \,|\, b, E, \theta)$

- Leads to quality functions simple to optimise.
  - Poisson, with self-loops, log and without unnecessary constant:
    - $\mathscr{L}(A \,|\, b) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{n_r n_s}$
  - Same, degree corrected
    - $\mathscr{L}(A \,|\, b) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{\kappa_r \kappa_s}$
  - With
    - m_{rs} : number of edges between blocks r, s
    - n_r : number of nodes in block r
    - κ_r: sum of degrees on block r

Karrer, B., & Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, *83*(1), 016107.

SBM INFERENCE

- Problem: trivial solution
  - Each node in its own cluster
    - $E = A$
    - Probability of 1 to generate the observed network

- Classic solution: fix a number of clusters k

MDL SBM

- [2016 Peixoto]
  - Non-parametric SBM
  - Bayesian inference
  - Minimum Description Length (MDL) (Occam's razor)

Bayesian Formulation

$$P(A \mid b) = P(A \mid \theta, E, b)P(\theta \mid E, b)P(E \mid b)P(b)$$

STOCHASTIC BLOCK MODELS

Information Theoretic Formulation

$$P(A, \theta, E, b) = 2^{-\Sigma} \qquad \Sigma = S + L$$

$$S = -\log_2 P(A \mid \theta, E, b)$$ # bits necessary to encode the graph knowing the model

$$L = -\log_2 P(\theta, E, b)$$ # bits necessary to encode the model

Objective = maximize the graph compression.
-Too many communities: over-complexifying the model
-Too few communities: More costly to encode the graph, since the model provides few useful information
Occam's razor

Peixoto, Tiago P. "Bayesian stochastic blockmodeling." *arXiv preprint arXiv:1705.10225* (2017).

STOCHASTIC BLOCK MODELS

- To sum up:
  - SBM have a convincing definition of communities
  - Have a richer expression power
  - Can also say if there is no community
  - And also suffer from a form of resolution limit

- Less often used, but regain popularity since works by Peixoto.
  - Variants:
    - Overlapping (Mixed membership, Fuzzy)
    - Hierarchical
    - …

EVALUATION OF COMMUNITY STRUCTURE

EVALUATION

- Two main approaches:
  - Intrinsic/Internal evaluation
    - Partition quality function
    - Individual Community quality function
  - Comparison of observed communities and expected communities
    - Synthetic networks with community structure
    - Real networks with Ground Truth

INTRINSIC EVALUATION

INTRINSIC EVALUATION

- Partition quality function
  - Already defined: **Modularity**, **graph compression**, etc.
- Quality function for individual community
  - Internal Clustering Coefficient
  - Conductance: $\frac{|E_{out}|}{|E_{out}| + |E_{in}|}$
    - Fraction of external edges

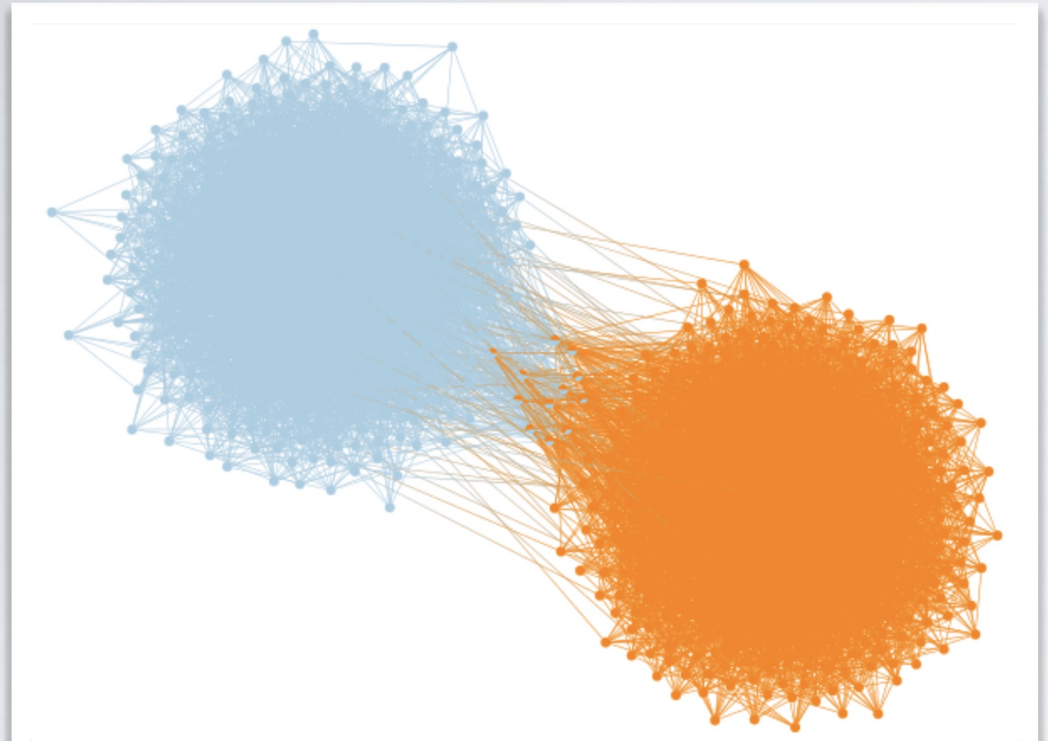
$|E_{in}|, |E_{out}|$:
# of links to nodes inside (respectively, outside) the community

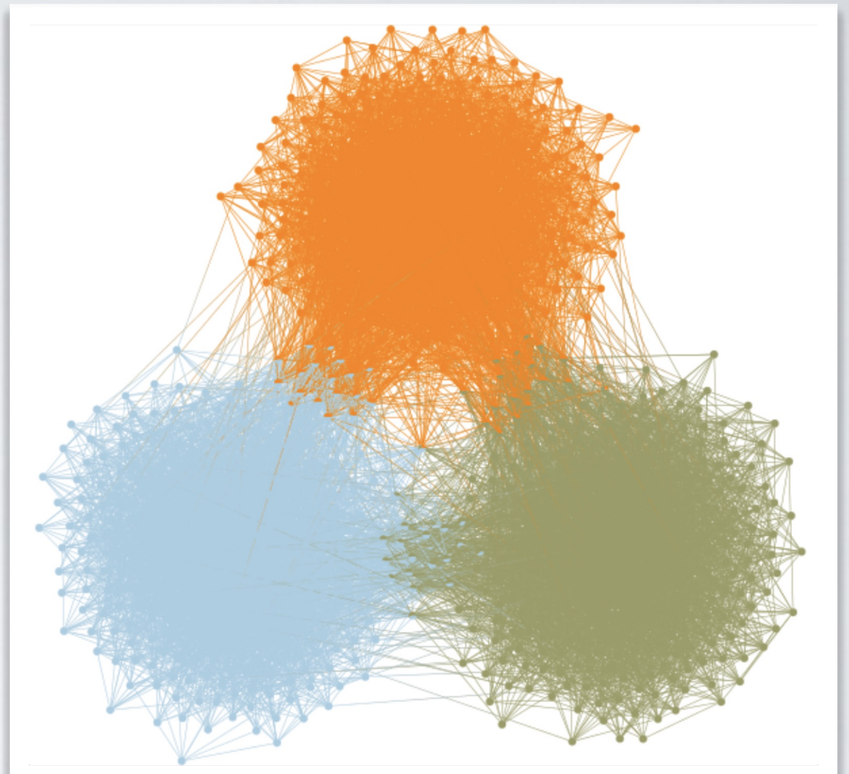
COMPARISON WITH GROUND TRUTH

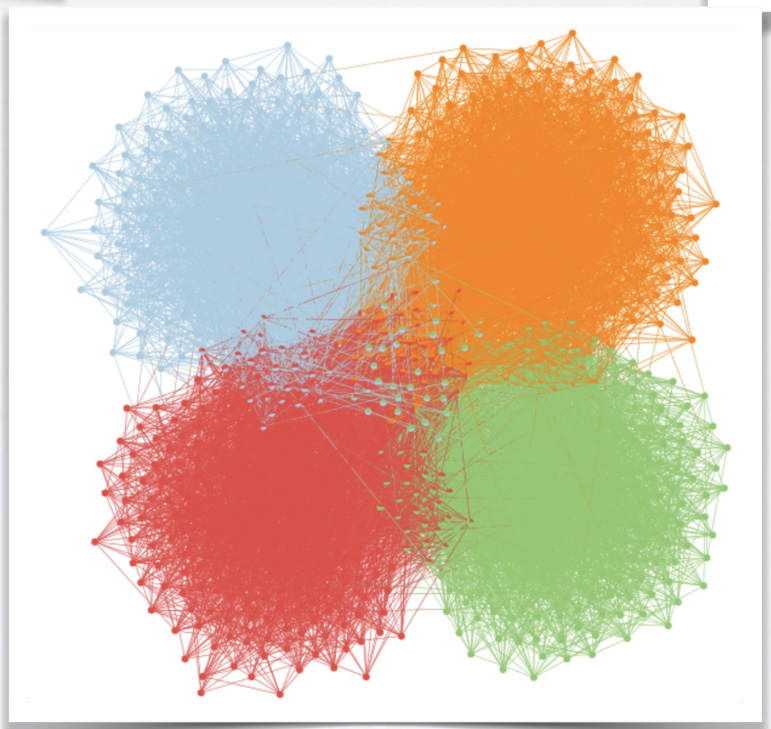
SYNTHETIC NETWORKS

- Planted Partition models:
  - Another name for SBM with manually chosen parameters
    - Assign degrees to nodes
    - Assign nodes to communities
    - Assign density to pairs of communities
    - Attribute randomly edges
  - Problem: how to choose parameters?
    - Either oversimplifying (all nodes same degrees, all communities same #nodes, all intern densities equals…)
    - Or ad-hoc process (sample values from distributions)

SYNTHETIC NETWORKS





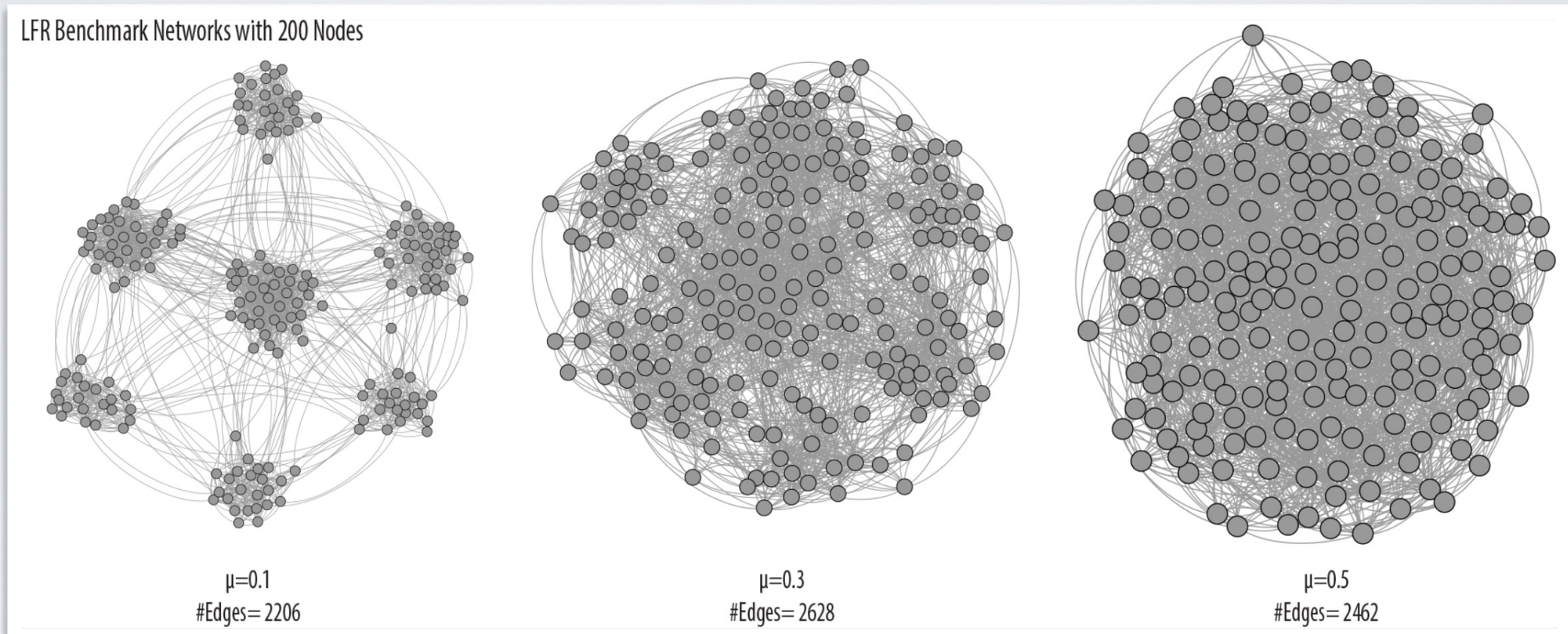


SYNTHETIC NETWORKS

- LFR Benchmark [Lancichinetti 2008]
  - High level parameters:
    - Slope of the power law distribution of degrees/community sizes
    - Avg Degree, Avg community size
    - Mixing parameter: fraction of external edges of each node
  - Varying the mixing parameter makes community more or less well defined

- Currently the most popular

SYNTHETIC NETWORKS



SYNTHETIC NETWORKS

- Pros of synthetic generators:
  - We know for sure the communities we should find
  - We can control finely the parameters to check robustness of methods
    - For instance, resolution limit…

- Cons:
  - Generated networks are not realistic: simpler than real networks
    - LFR: High CC, scale free, but all nodes have the same mixing coefficient, no overlap, …
    - SBM: depend a lot on parameters, random generation might lead to unexpected ground truth (it is *possible* to have a node with no connections to other nodes of its own community…)

REAL NETWORKS WITH GT

- In some networks, **ground truth** communities are known:
  - Social networks, people belong to groups (Facebook, Friendsters, Orkut, students in classes…)
  - Products, belonging to categories (Amazon, music…)
  - Other resources with defined groups (Wikipedia articles, Political groups for vote data…)

- Some websites have collected such datasets, e.g.
  - http://snap.stanford.edu/data/index.html

REAL NETWORKS WITH GT

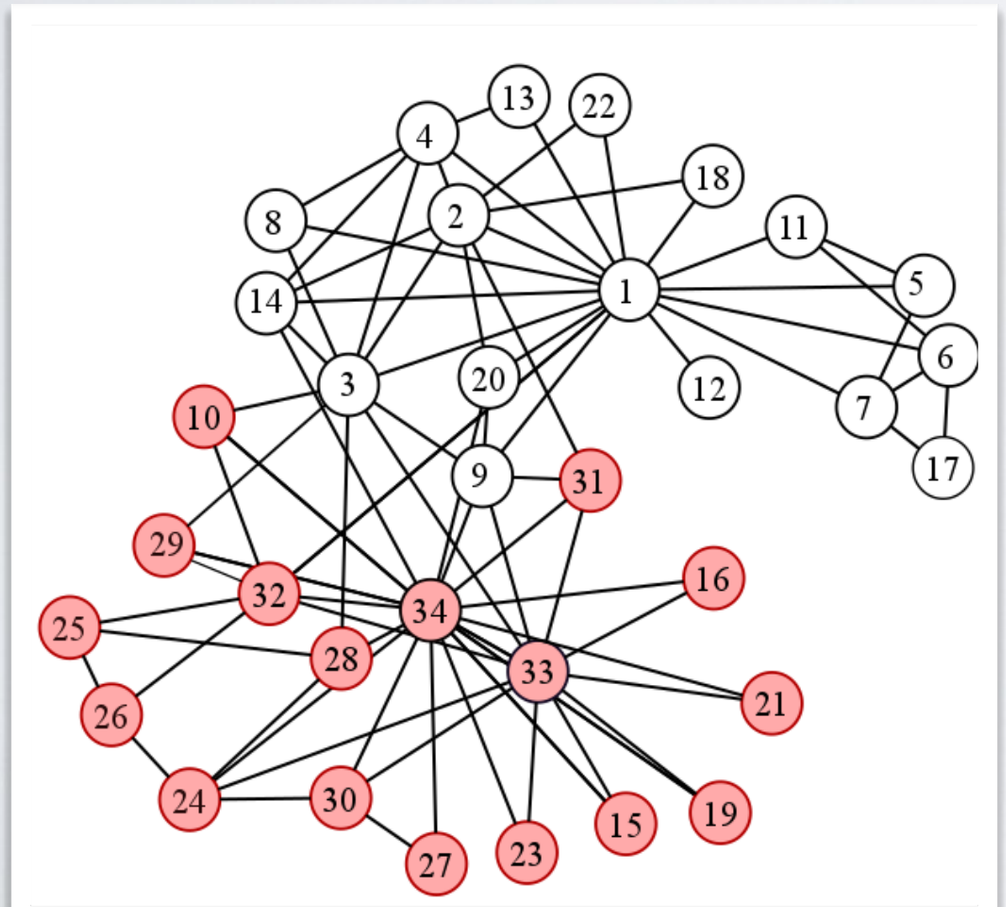
- Pros of GT communities:
  - Retain the full complexity of networks and communities

- Cons:
  - No guarantee that communities are **topological** communities.
  - In fact, they are not: some GT communities are not even a single connected component…

- Currently, controversial topic
  - Some authors say it is non-sense to use them for validation
  - Some others consider it necessary

REAL NETWORKS WITH GT

- Example: the most famous of all networks: Zackary Karate Club

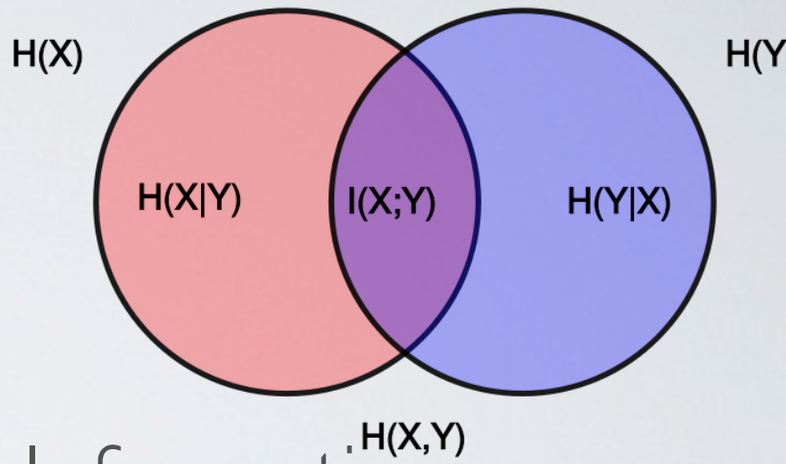


If your algorithm find the right communities,
Then it is wrong…

MEASURING PARTITION SIMILARITIES

- Synthetic or GT, we get:
  - Reference communities
  - Communities found by algorithms
- How to measure their similarity ?
  - NMI => AMI
  - ARI
  - …

MEASURING PARTITION SIMILARITIES



- NMI: Normalized Mutual Information
- Classic notion of Information Theory: Mutual Information
  - How much knowing one variable reduces uncertainty about the other
  - Or how much in common between the two variables

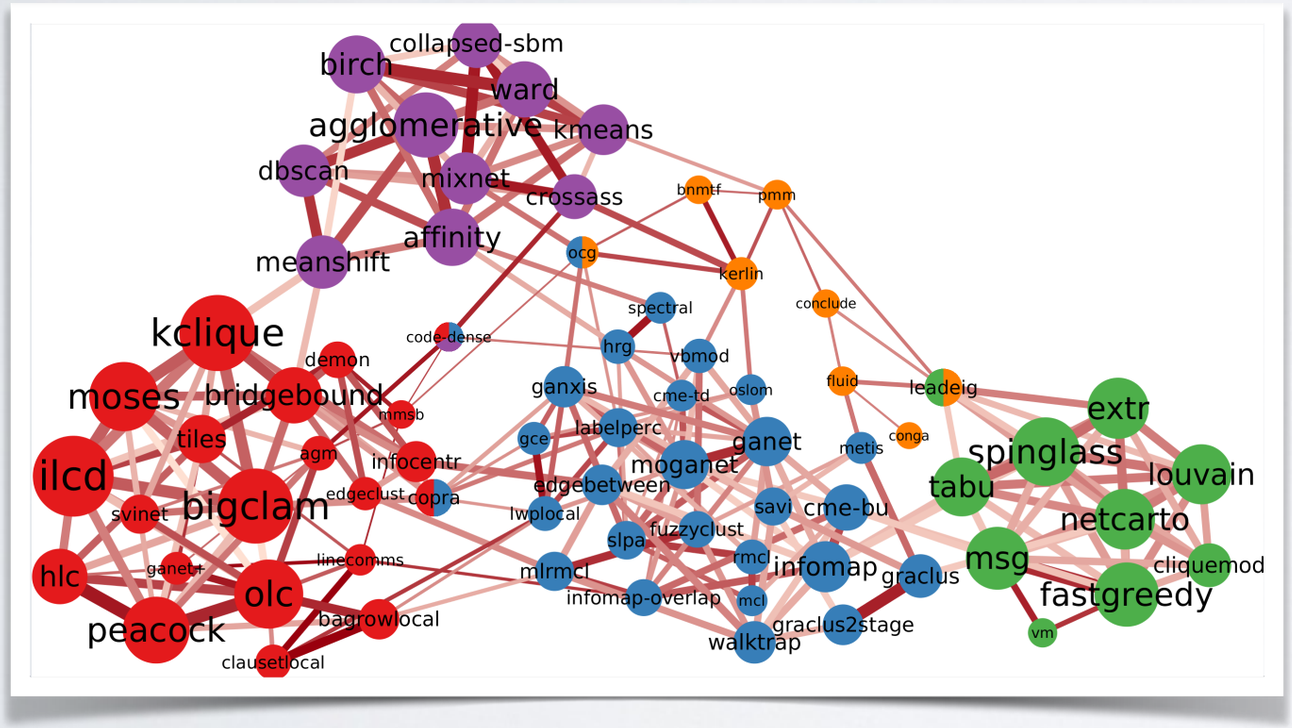
$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

- Normalized version: NMI
  - 0: independent, 1: identical
- Adjusted for chance: aNMI

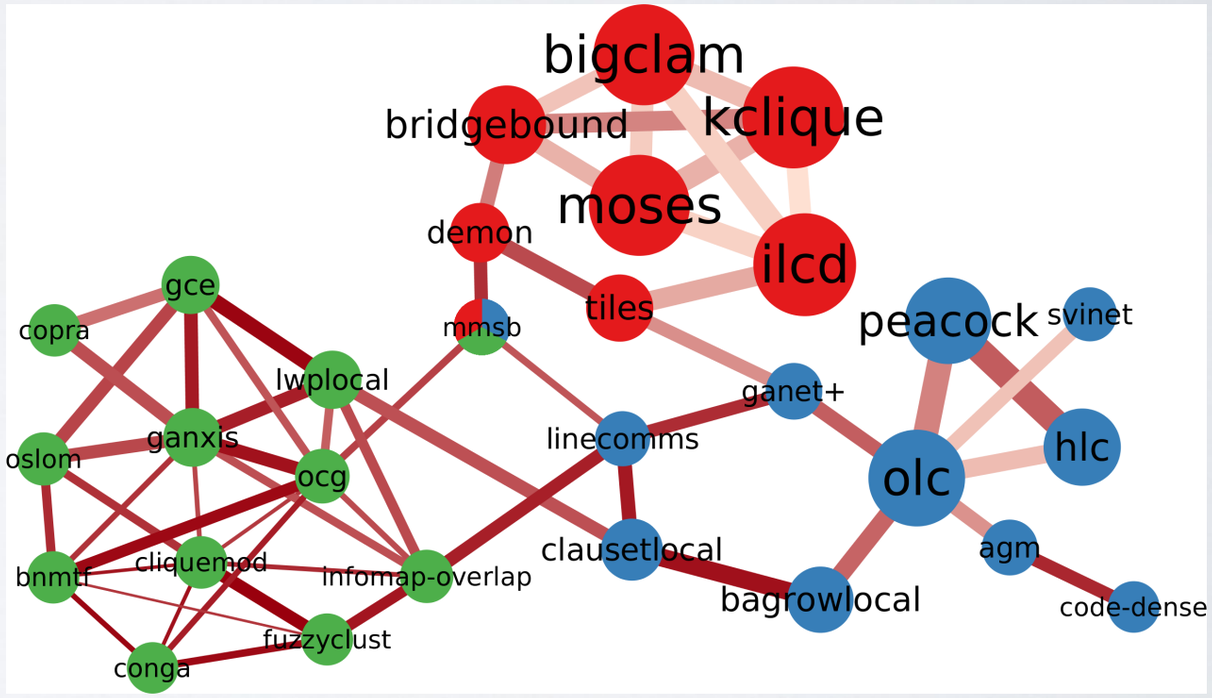
$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U), H(V)\} - E\{MI(U,V)\}}$$

ALGORITHMS COMPARATIVE ANALYSIS

Rank	Algorithm	oNMI MAX
1	linecomms	165
2	oslom	73
3	infomap-overlap	64
4	savi	62
5	labelperc	57
6	rmcl	54
7	edgebetween	41
7	leadeig	41
7	vbmod	41
10	gce	32



All methods



Overlapping only

Coscia, Michele. "Discovering Communities of Community Discovery." *arXiv preprint arXiv:1907.02277* (2019).

OTHER MESO-SCALE ORGANIZATIONS

MESO-SCALE

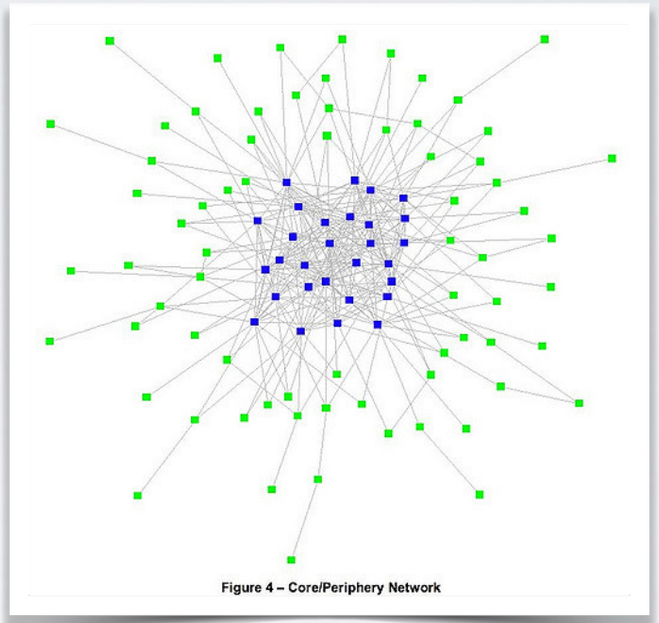
- MACRO properties of networks:
  - degree distribution, density, average shortest path…

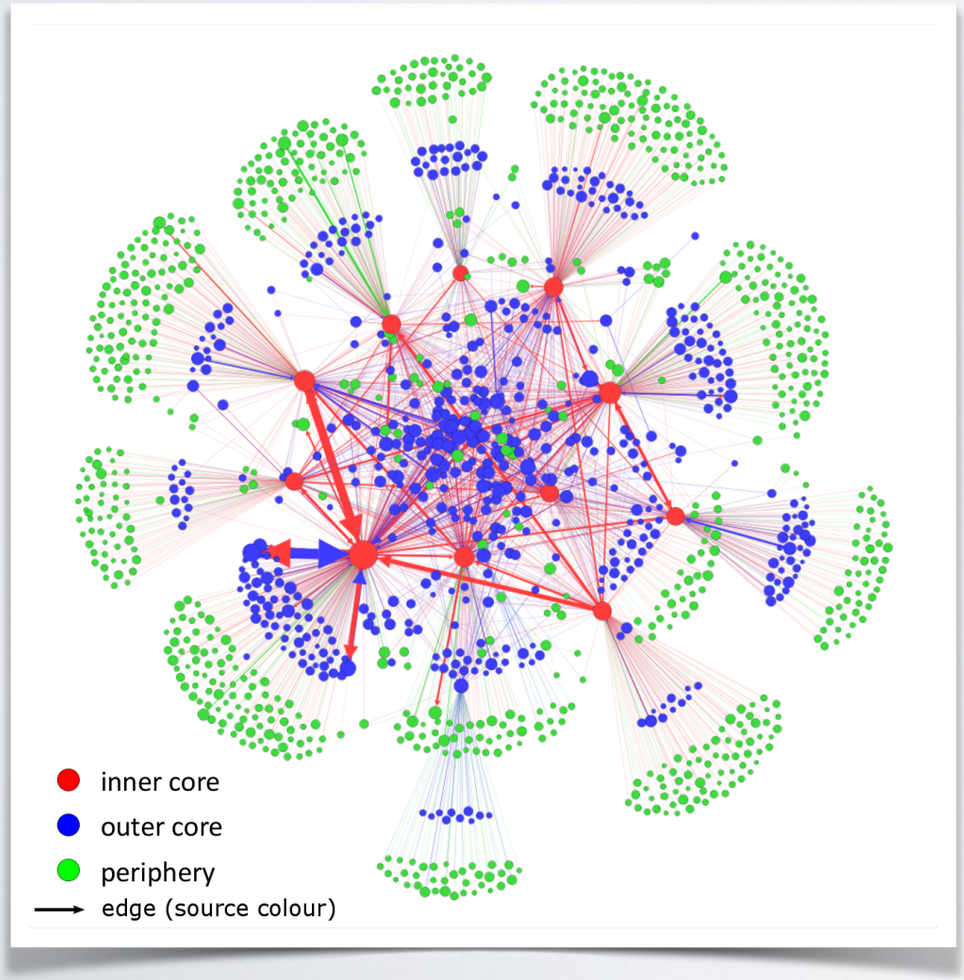
- MICRO properties of networks:
  - Centralities

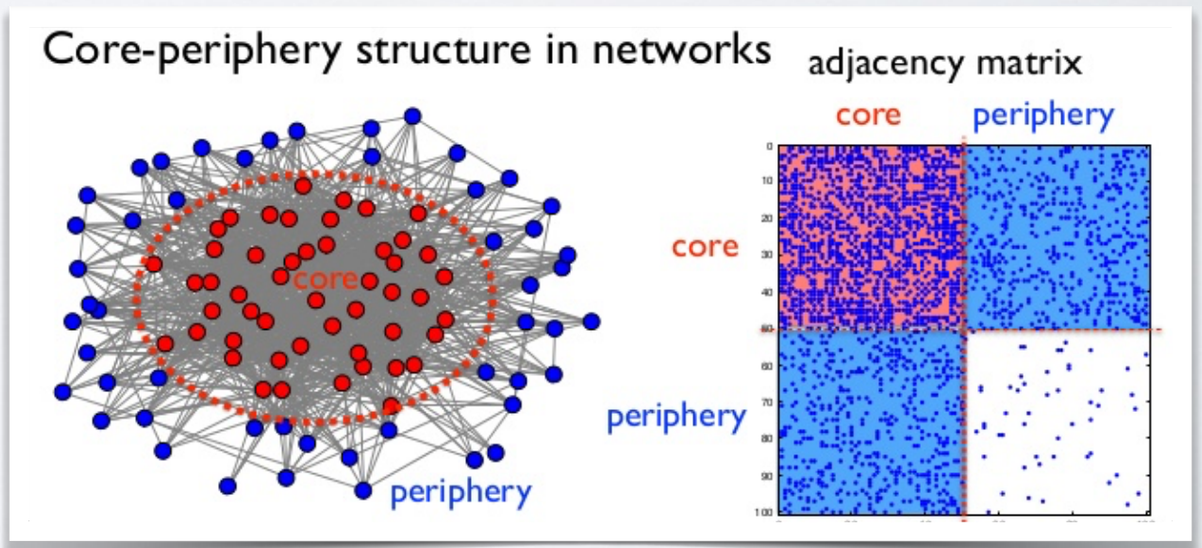
- MESO-scale: what is in-between
  - Community structure
  - Overlapping Community Structure
  - Core-Periphery
  - Spatial Organization (another class)

CORE-PERIPHERY

- Already introduced in the first class, k-cores, etc.







OVERLAPPING COMMUNITIES

- In real networks, communities are often overlapping
  - Some of your High-School friends might be also University Friends
  - A colleague might be a member of your family
  - …
- Overlapping community detection is considered much harder
  - And is not well defined

OVERLAPPING COMMUNITIES

- Many algorithms
  - Adaptations of modularity, random walks, label propagations, SBM…
  - Original methods
  - Many local methods (local criterium), unlike global optimization for non-overlapping methods.

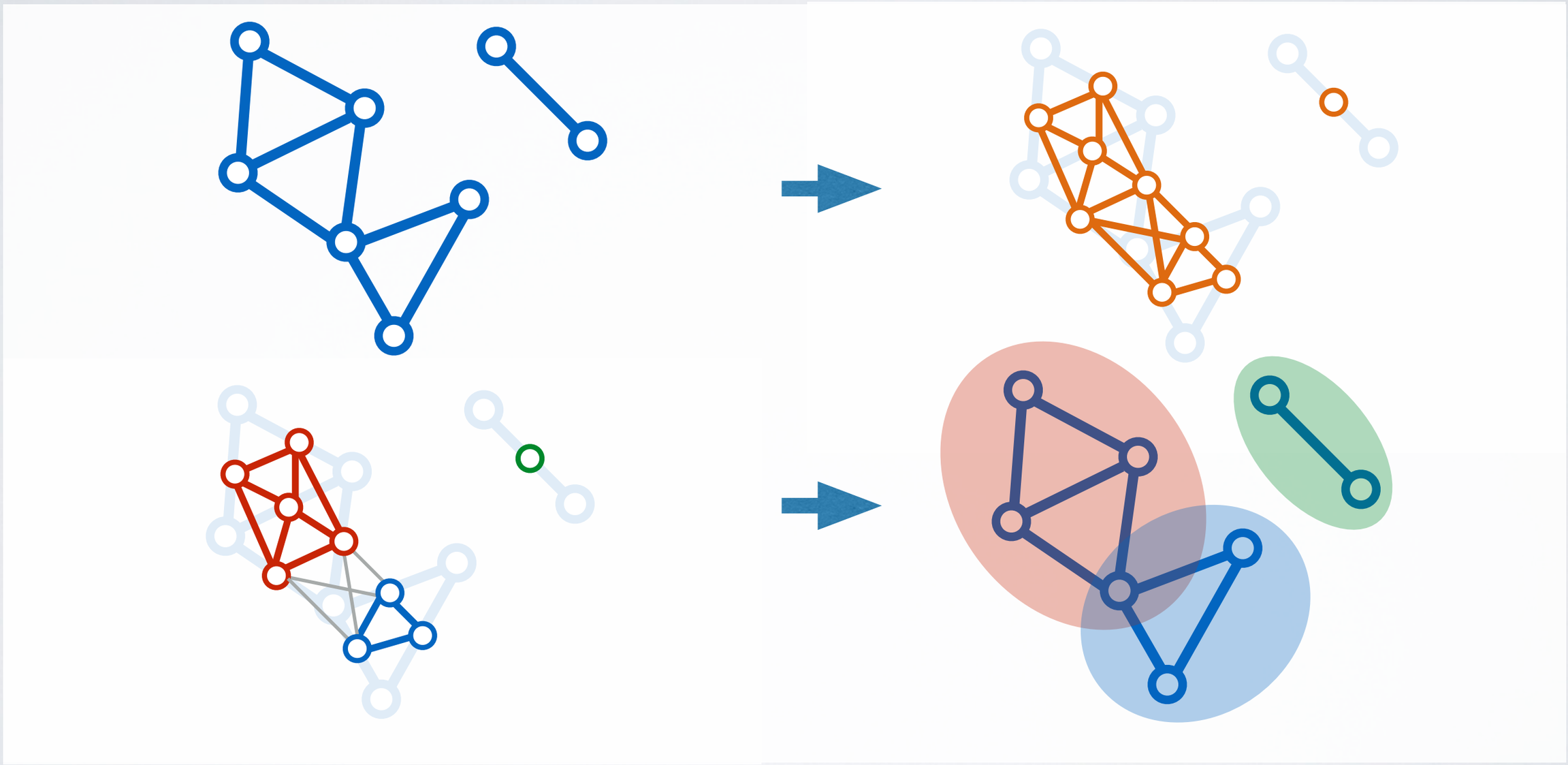
OVERLAPPING COMMUNITIES

- Motif-based definitions:
  - Cliques
    - Of a given size
    - Maximal cliques
  - N-cliques
    - Set of nodes such as there is at least a path of length <=N between them
    - Generalization of cliques for N>1
    - Computationally expensive

Link clustering - overlapping communities

Link graphs

- Links are replaced by nodes which are connected if the original links share a node

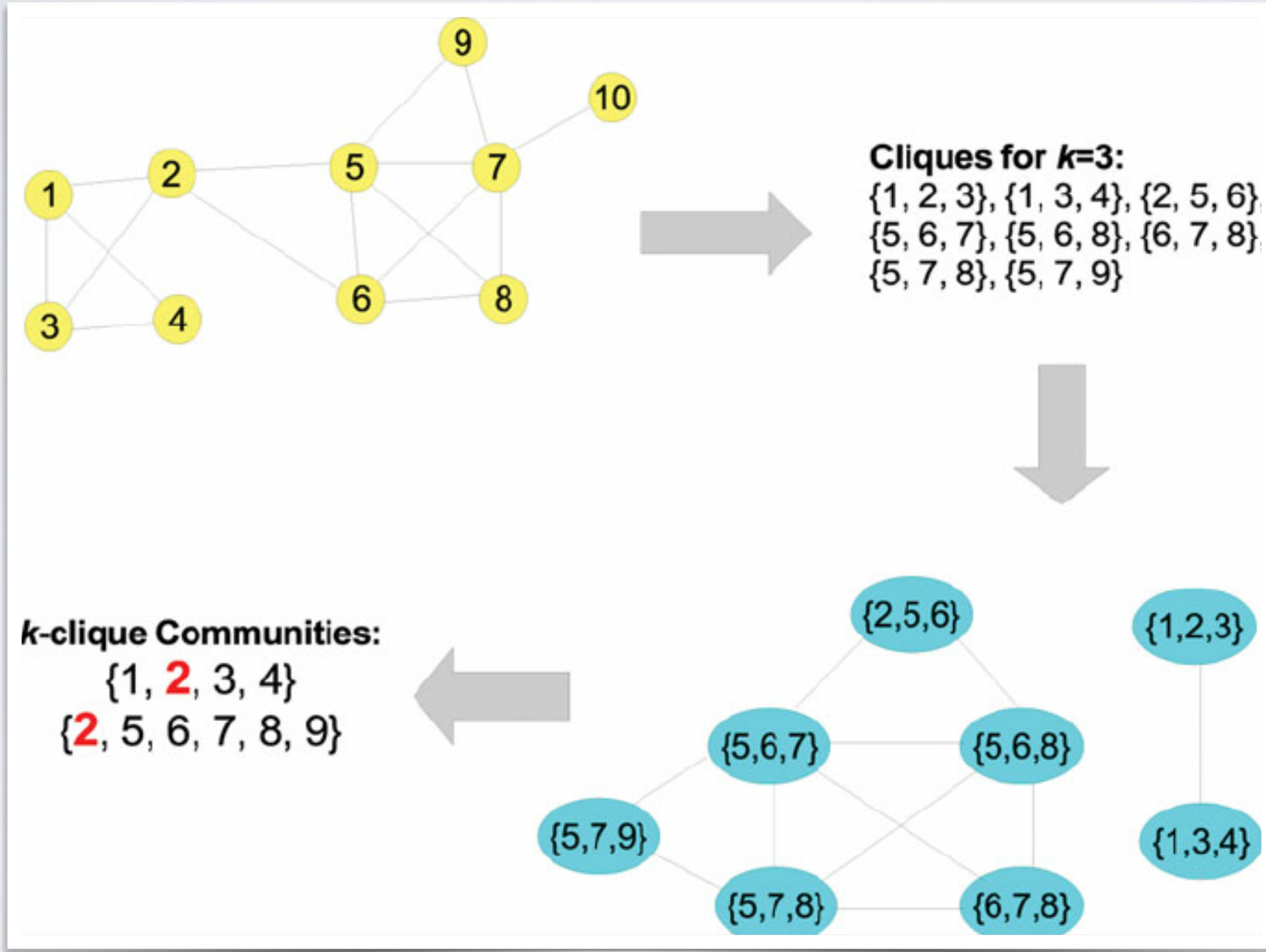


- Community detection on link graphs allows for overlapping communities

K-CLIQUE PERCOLATION

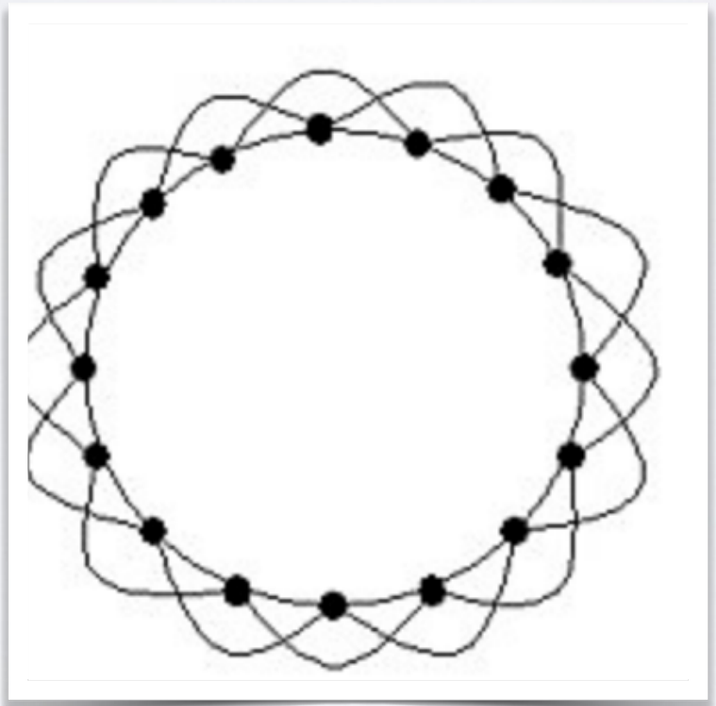
- (Other name: CPM, C-finder)
- Parameter: size k of atomic cliques
- 1)Find all cliques of size k
- 2)merge iteratively all cliques having k-1 nodes in common

K-CLIQUE PERCOLATION



K-CLIQUE PERCOLATION

- Weakness: communities can be very far from random networks



HIERARCHICAL COMMUNITIES

Lancichinetti, Andrea, et al. "Finding statistically significant communities in networks." *PloS one* 6.4 (2011): e18961.

SUPERVISED MACHINE LEARNING I: LINK PREDICTION

LINK PREDICTION

- Do you know why Facebook "People you may know" is so accurate?
- How youtube/Spotify/amazon recommend you the right item?
- =>Link prediction
  - More generally, recommendation, but link prediction is a popular way to do it

LINK PREDICTION

- Observed network: current state

- Link prediction: What edge
  - Might appear in the future (*future link* prediction)
  - Might have been missed (*missing link* prediction)

LINK PREDICTION

- Overview:
- Link prediction based on network structure:
  - Local: High clustering (friends of my friends will become my friends)
  - Global: Two unrelated hubs more likely to have links that unrelated small nodes
  - Meso-scale organisation: different edge probability for nodes in different communities/blocks
- Link prediction can also be based on node properties
  - e.g., age, revenue, gender, etc.
  - Combining with usual machine learning, outside of the scope of this class

FIRST APPROACH TO LINK PREDICTION:

HEURISTIC BASED

(HEURISTICS, NOT MACHINE LEARNING)

HEURISTICS

- Network science experts can design **heuristics** to predict where new edge might appear/be missing

- Principle: design a score based on network topology f(v1,v2) which, given two nodes, express their likeliness of being connected (if they aren't already)
  - Common neighbors
  - Jaccard coefficient
  - Hub promoted
  - Adamic Adar
  - Ressource allocation
  - Community based

Zhou, T., Lü, L., & Zhang, Y. C. (2009). Predicting missing links via local information. *The European Physical Journal B*, *71*(4), 623-630.

COMMON NEIGHBORS

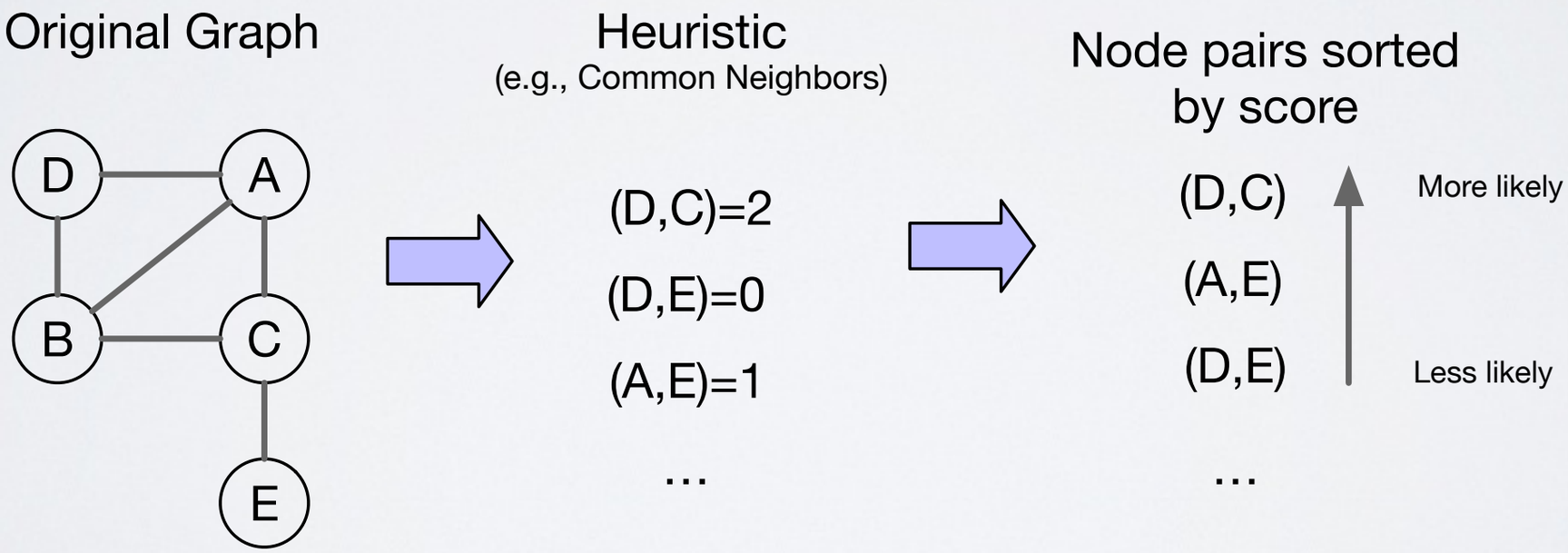
- "Friends of my friends are my friends"
- High clustering in most networks
- =>The more friends in common, the highest probability to become friends

$$\mathrm{CN}(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

$\Gamma(x) =$ Neighbors of x

PREDICTION

- How to predict links based on Common Neighbors (CN)?



JACCARD COEFFICIENT

- Used in many applications:
  - Measure of similarity of sets of different sizes
- Intuition:
  - Two people who know only the same 3 people
    - =>high probability
  - Two people who know 1000 people, only 3 in commons
    - =>Lower probability

$$\mathrm{JC}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

HUB PROMOTED

- Intuition:
  - Normalized by min-number of neighbours
  - Variant: hub depressed (max instead of min)
  - Two stars have 10 common followers or I have ten friends following a star

$$\mathrm{HP}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min(|\Gamma(x)|, |\Gamma(y)|)}$$

ADAMIC ADAR

- Intuition:
  - For previous scores: all common nodes are worth the same
  - For AA:
    - A common node with ONLY them in common is worth the most
    - A common node connected to everyone is worth the less
    - The higher the size of its neighborhood, the lesser its value

$$\mathrm{AA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|}$$

RESSOURCE ALLOCATION

- Similar to Adamic Adam, penalize more higher degrees

$$\mathrm{RA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|}$$

$$\mathrm{AA}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|}$$

PREFERENTIAL ATTACHMENT

- Preferential attachment:
  - Every time a node join the network, it creates a link with nodes with probability proportional to their degrees
  - In fact, closer to the definition of the configuration model

- Score not based on common neighbors
  - =>Assign different scores to nodes at network distance >2

- Intuition: Two nodes with many neighbors more likely to have new ones than nodes with few neighbors

$$\mathrm{PA}(x, y) = |\Gamma(x)| \cdot |\Gamma(y)|$$

OTHER SCORES

Examples of other scores proposed

Sorenson Index

$$\mathrm{SI}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|}$$

Salton Cosine Similarity

$$\mathrm{SC}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| \cdot |\Gamma(y)|}}$$

Hub Depressed

$$\mathrm{HD}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{max(|\Gamma(x)|, |\Gamma(y)|)}$$

Leicht-Holme-Nerman

$$\mathrm{LHN}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| \cdot |\Gamma(y)|}$$

COMMUNITY STRUCTURE

- General idea:
  - 1)Compute community structure on the whole graph
  - 2)Assign high score for 2 nodes in a same community, a low score otherwise

- How to choose the score?

COMMUNITY STRUCTURE

- For methods based on a quality function optimization (Modularity, Infomap's information compression, etc.)
  - Assign a score to each pair proportional to the change in quality function associated with adding an edge between them

- For instance, Louvain optimize Modularity.
  - Each edge added between communities:
    - Decrease in the Modularity
  - Edge added inside community:
    - Increase in Modularity, depends on properties of the community and nodes

Ghasemian, A., Hosseinmardi, H., & Clauset, A. (2019). Evaluating overfit and underfit in models of network community structure. *IEEE Transactions on Knowledge and Data Engineering*.

OTHER SCORES

- Distance based:
  - Length of the shortest path
  - Probability to reach a node from another on a random-walk of distance *k*
    - See next classes on embeddings
  - Number of paths of length d between the nodes

- Problem: computational complexity

ML APPROACH TO LINK PREDICTION:

SIMILARITY SCORE, SUPERVISED

SUPERVISED MACHINE LEARNING

- Use Machine Learning algorithms to **learn** how to combine heuristics for optimizing predictions

- Two steps:
  - Training: show features + value to predict
  - Using/Validating: try to predict value from features

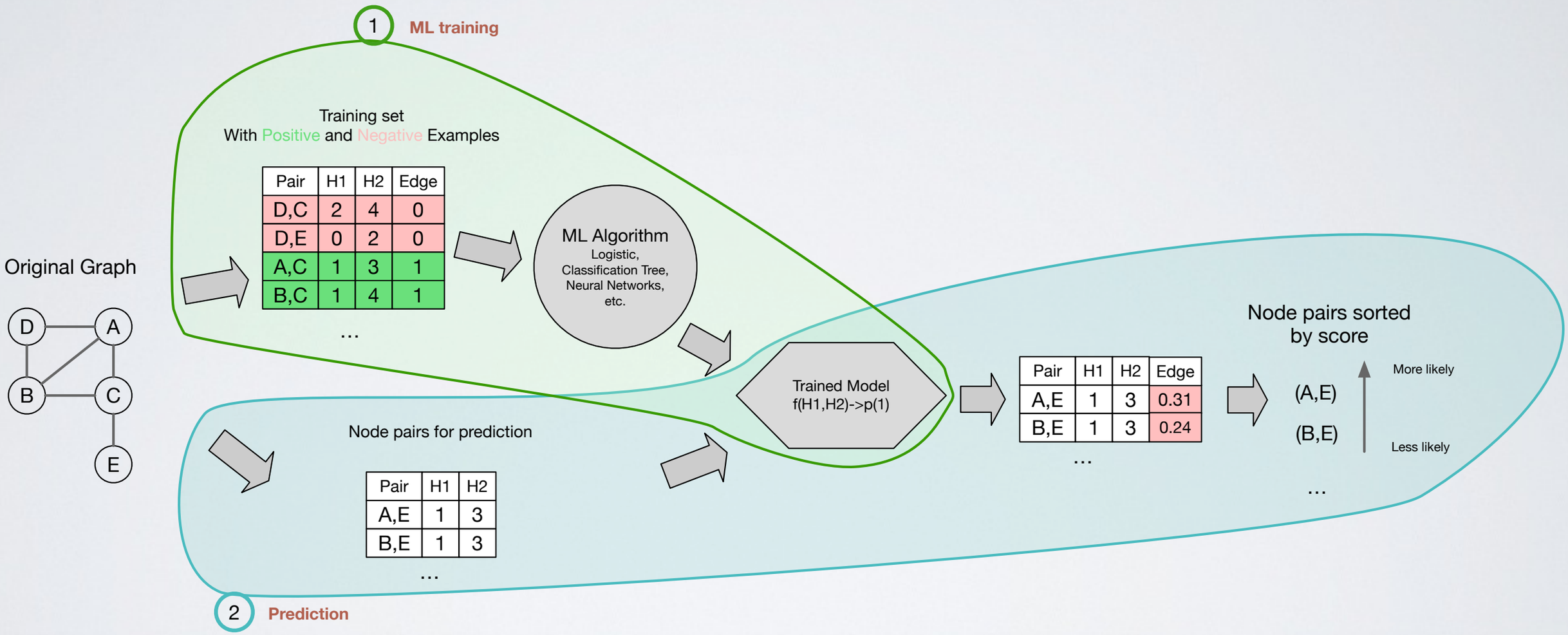
SUPERVISED MACHINE LEARNING

- Our features: similarity indices (CN, AA, PA, …)
  - **One** (limited interest) or, obviously, **several**
  - Nodes attributes can be added of available (age, salary, etc.)

- Our label/value to predict: *Link(1)* or *No link(0)* (2 **classes**)

- These types of ML algorithms are called **classifiers**
  - Logistic Classifier
  - Decision Tree Classifier
  - Neural networks Classifier
  - …

SUPERVISED MACHINE LEARNING



SUPERVISED MACHINE LEARNING

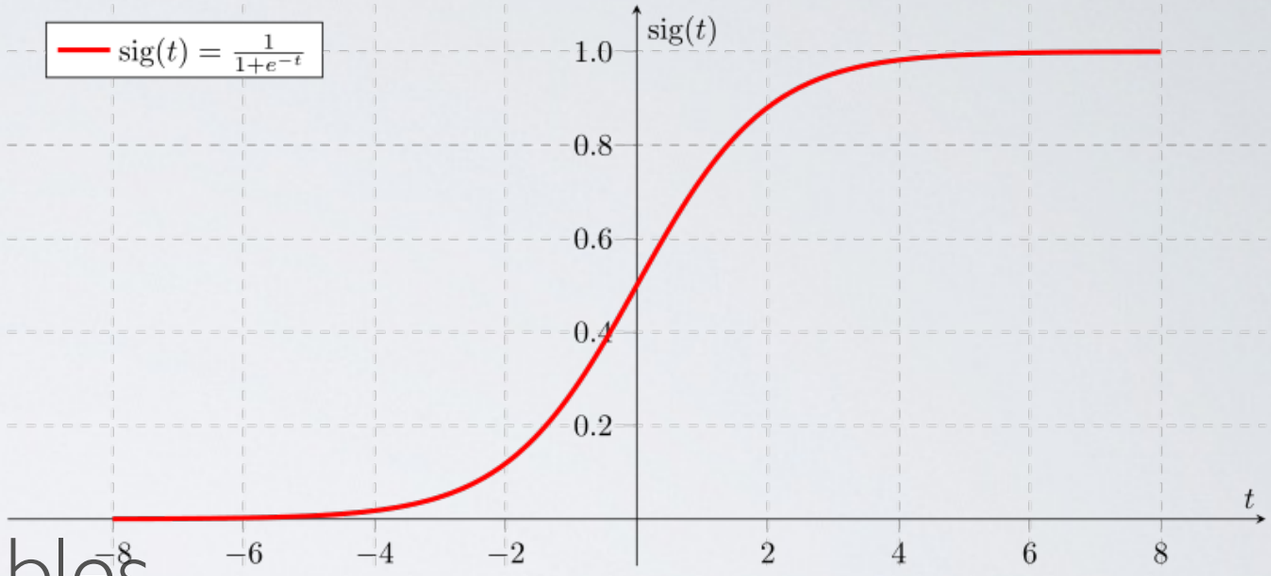
- Scores of methods, very different in their mechanisms, but same **input** and **output**

```
#lm = linear_model.LinearRegression()
#lm = linear_model.ElasticNet()
#lm = ensemble.GradientBoostingRegressor()
#lm = ensemble.RandomForestRegressor()
lm = MLPRegressor(hidden_layer_sizes=(3,3,3)



lm.fit(X_train,y_train)
```

Let's see 2 simple examples: Logistic classification, Decision Trees

https://en.wikipedia.org/wiki/Logistic_regression

LOGISTIC CLASSIFICATION



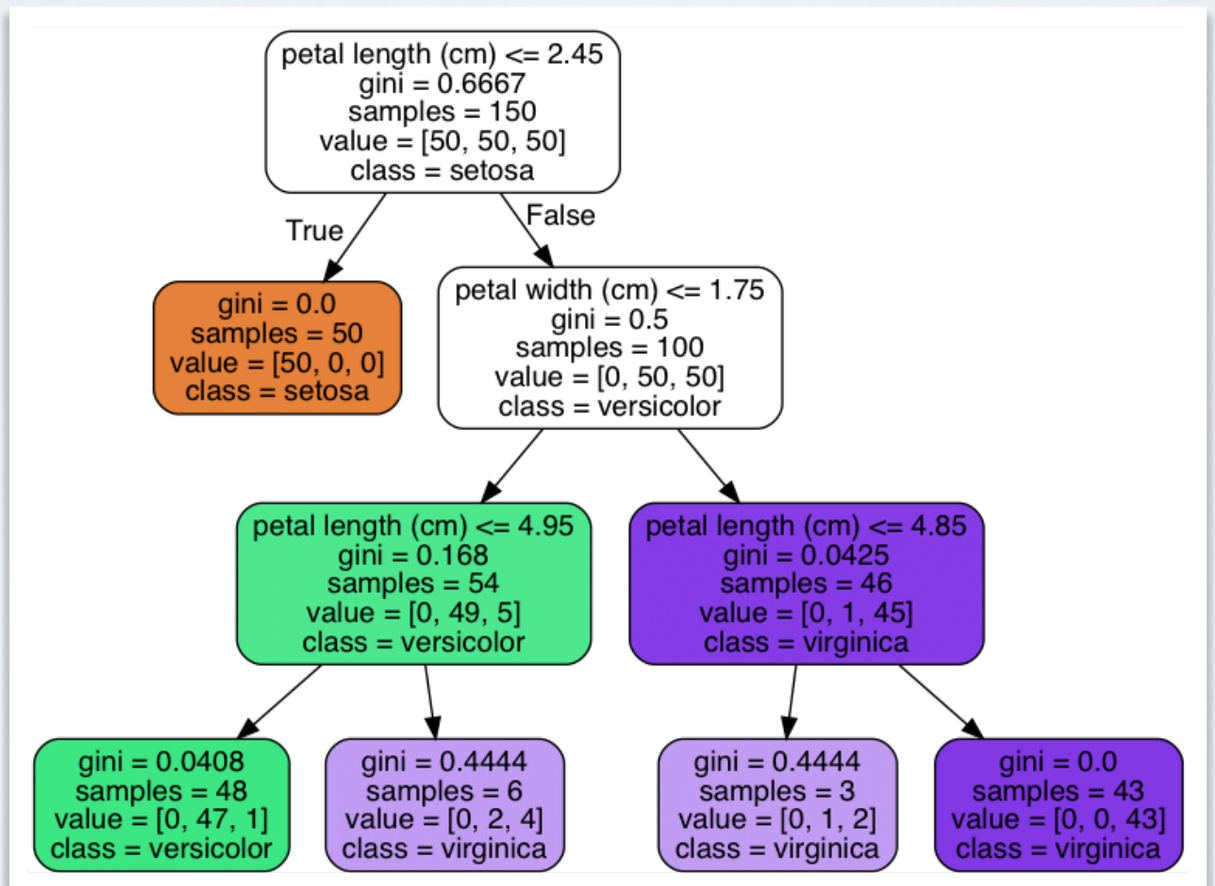
- Value to predict y_t:
  - 0 (no edge)
  - 1 (edge)

- Linear relations between variables
  - $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i$

- Find $\beta_0, \beta_1, \ldots$ that minimizes $y_t - y_i$

DECISION TREES

- Measure of heterogeneity (Gini, entropy…)
- Split recursively data in 2 to maximize homogeneity in child nodes



https://en.wikipedia.org/wiki/Decision_tree

DECISION TREES

- Example of possible outcomes with a decision tree:
- If CN <1
  - IF PA>1000 => Predict 1
  - ELSE => Predict 0
- ELSE
  - IF PA > 10000 => Predict 1
  - ELSE
    - IF AA > 10 => Predict 1
    - ELSE
    - IF JC < 0.2 => Predict 0
    - …

NODE CLASSIFICATION

Bhagat, S., Cormode, G., & Muthukrishnan, S. (2011). Node classification in social networks. In *Social network data analytics* (pp. 115-148). Springer, Boston, MA.

NODE CLASSIFICATION

- For the node classification task, we want to predict the class/ category (or numerical value) of some nodes
  - Missing values in a dataset
  - Learn to predict, in a social network/platform(Netflix…) individuals':
    - Political position, opinion on a given topic, possible security threat, …
    - Interests, tastes, etc.
    - Age, gender, sexual orientation, language spoken, salary, etc.
    - Fake accounts, spammers, bots, malicious accounts, etc.
    - …
  - Wikipedia article category, types of road in an urban network, etc.

NODE CLASSIFICATION

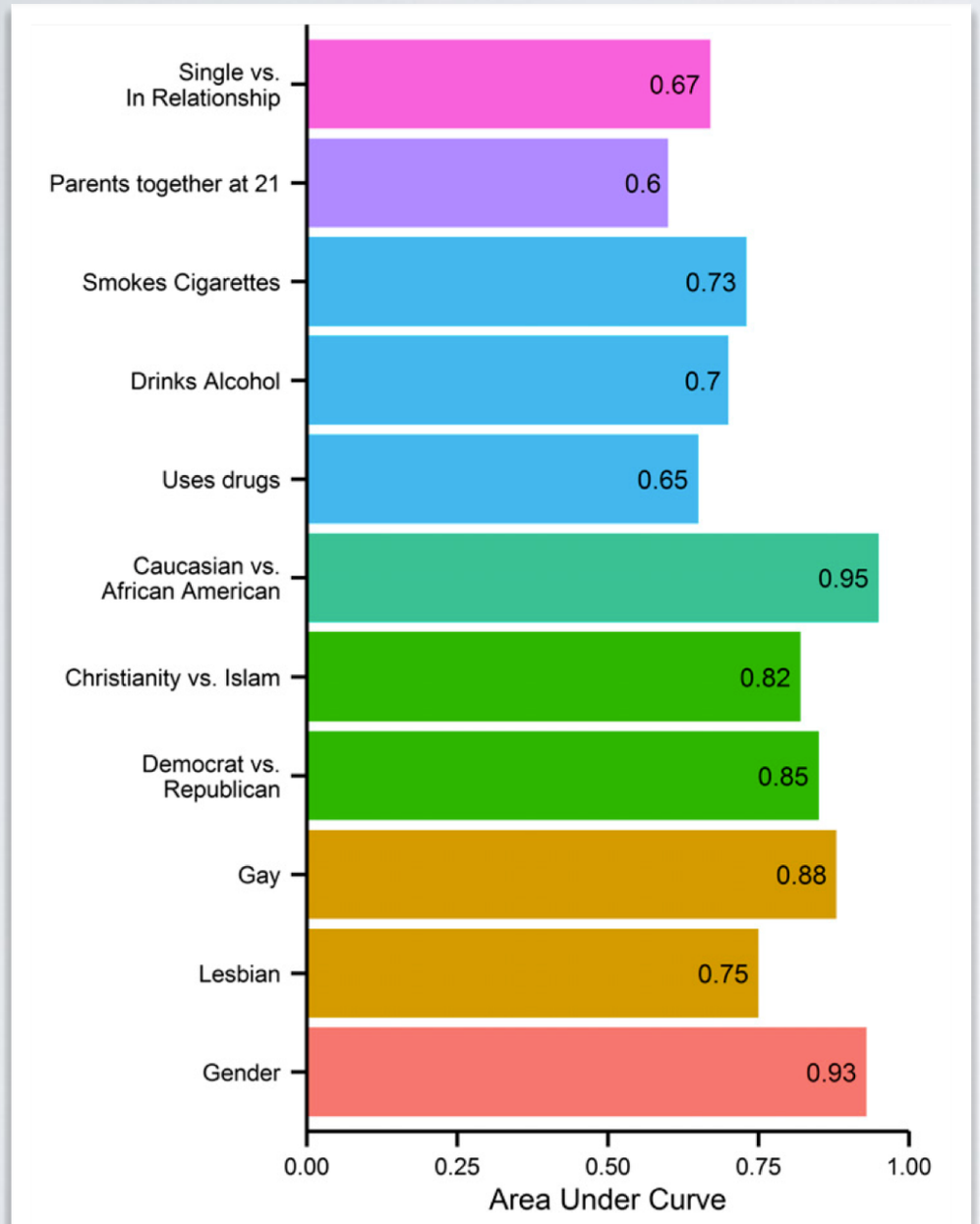


Fig. 2. Prediction accuracy of classification for dichotomous/dichotomized attributes expressed by the AUC.

Example of risks

Jernigan, C., & Mistree, B. F. (2009). Gaydar: Facebook friendships expose sexual orientation. *First Monday*, *14*(10).

Kosinski, M., Stillwell, D., & Graepel, T. (2013). Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, *110*(15), 5802-5805.

NODE FEATURES

- Non-network approach: Use a classification algorithm based on features of the node itself (age, salary, etc.)
- The network structure can be integrated using node centralities: Degree, clustering coefficient, betweenness, etc.
- But we can do much better:
  - "Tell me who your friends are, and I will tell you who you are"

NEIGHBORHOOD BASED CLASSIFICATION

- Classification based on the distribution of features in the neighborhood

- For each node, compute the distribution of labels in its neighborhood (vectors of length *m*, with *m* the set of all possible labels)
  - Pick the most frequent
    - e.g., political opinions
  - Train a classifier on this distribution
    - e.g., distribution of age, language in the neighborhoods to recognize bots (unexpectedly random)