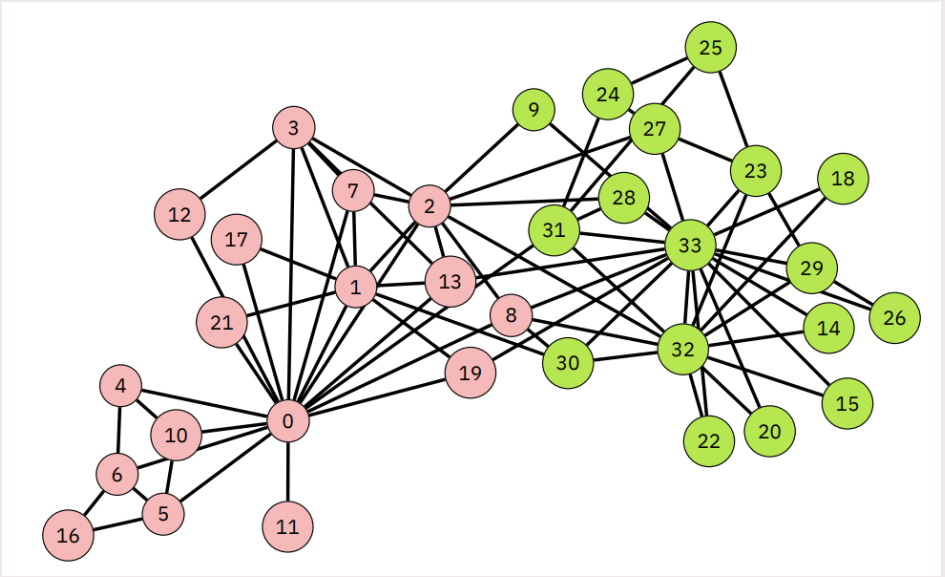
Community Detection

31-03-2023

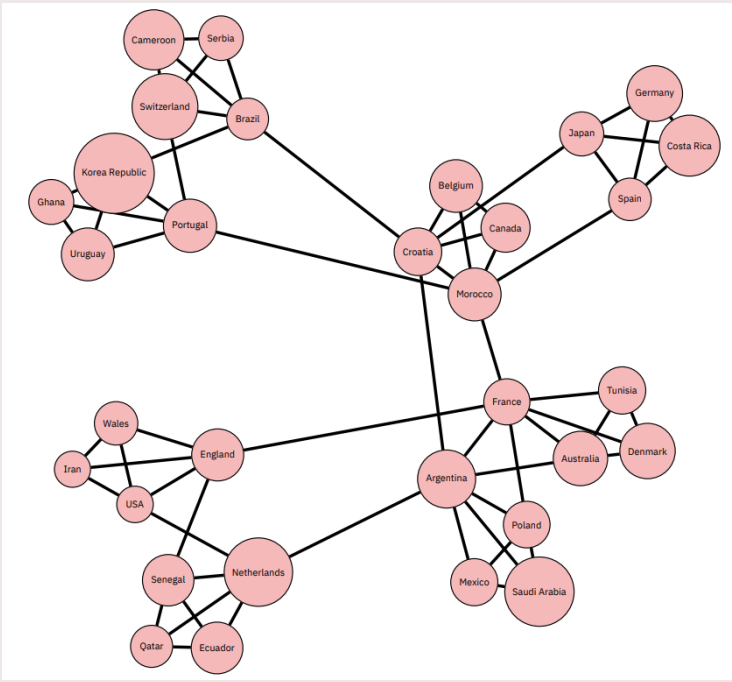
Martijn Gösgens

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Communities

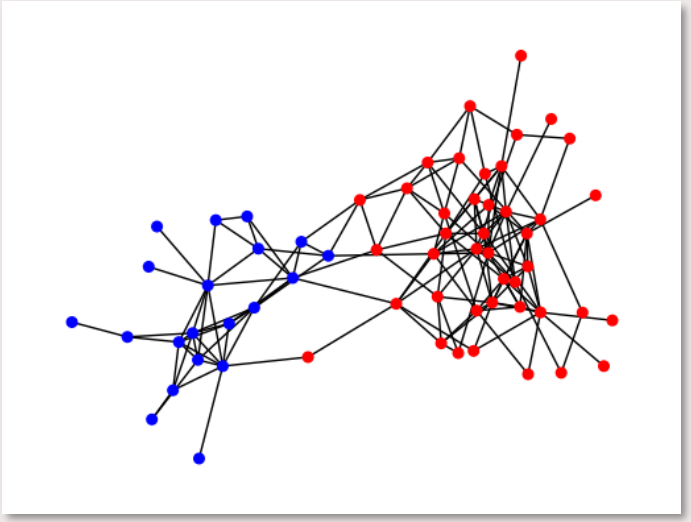


Network of a Karate club as collected by an anthropologist. Members are connected if they met outside the karate club.

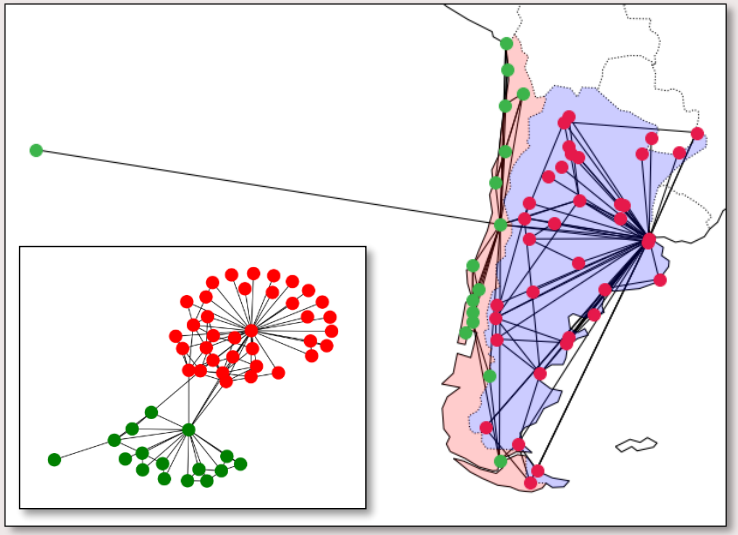


Network of FIFA World Cup 2022. Teams are connected if they played a football match.

Communities



Network of dolphins collected by a biologist. Dolphins are connected if they swam together more than average.

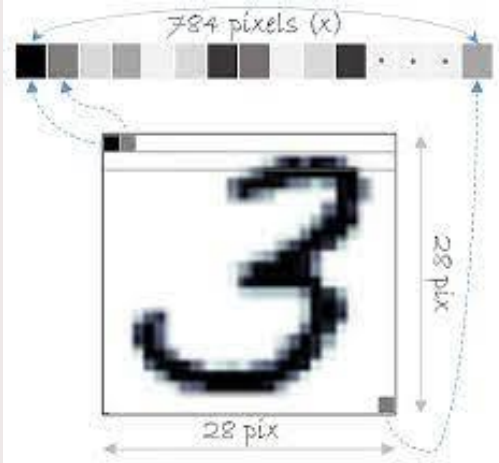


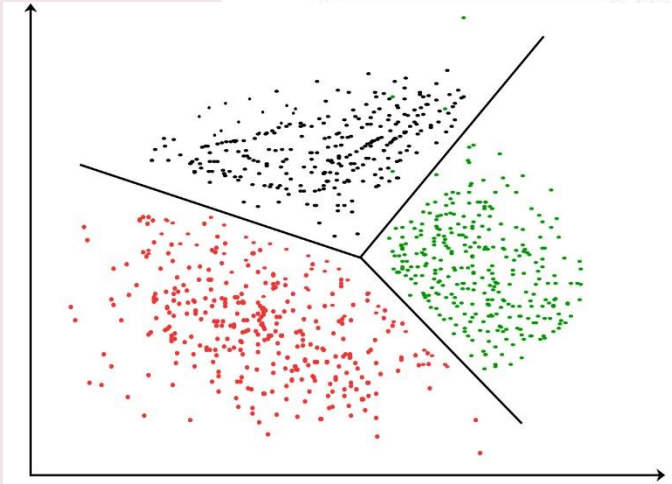
Network of airports in Argentina and Chile. Links indicate direct flights.

TU/e

Community detection vs Clustering

- Clustering: dividing a set of ‘things’ into ‘natural groups’.
- ‘Things’ can be anything, e.g.:
  - Grouping points in space
  - Grouping texts based on similarity
- Community detection: clustering network-nodes

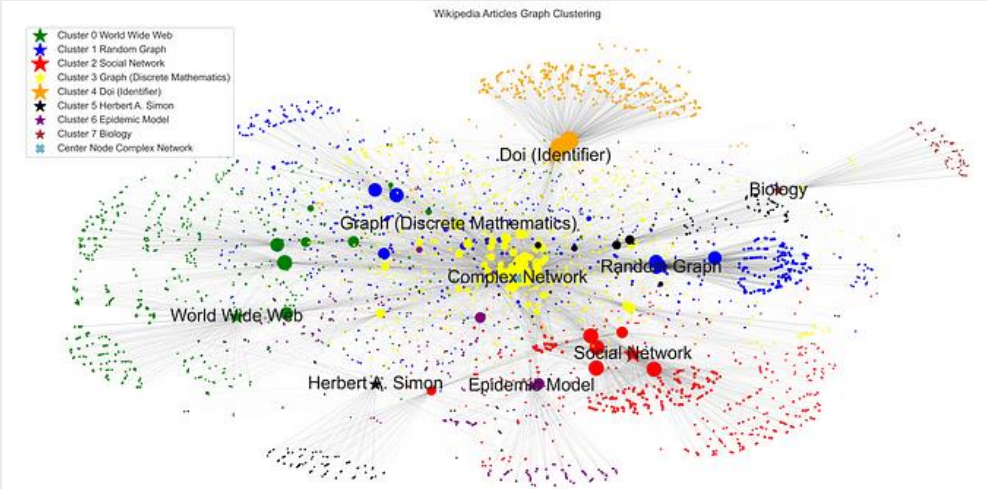




TU/e

What are communities?

- Groups of network-nodes that are better connected internally than externally
- No clear definition what communities are
  - Many different ideas of what communities should look like
- Why detect them?
  - To better understand the network
  - To help with classifying the nodes
  - Friend suggestions for social media



Source: Danie Mendez, medium.com

Notation

A network $\mathcal{N}$ with

- Nodes N
- Edges E

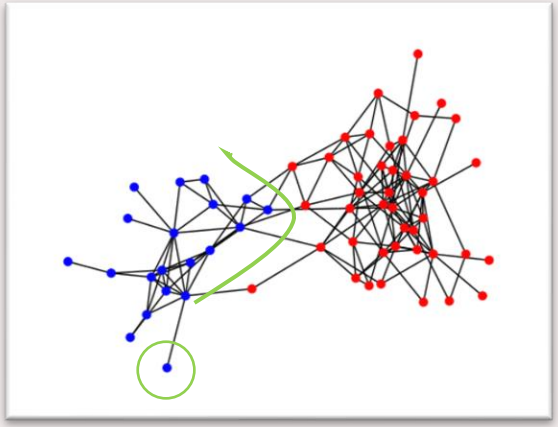
For two sets of nodes $A, B \subset N$, $\ell(A, B)$ denotes the number of links starting from A and ending in B. (Also: $\ell(A) = \ell(A, A)$).

A community structure is a *partition* of N, i.e., $C = \{C_1, C_2, \dots, C_k\}$.
The community of a node i is denoted by $C(i)$.

Size of the j-th community: $|C_j|$
Number of communities: $|C|$.

What do communities look like?

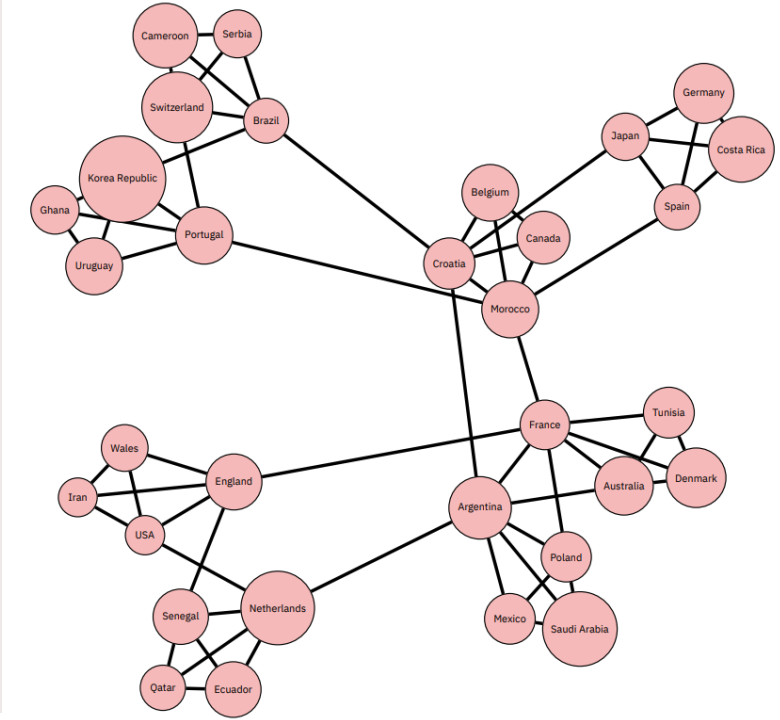
- Many links inside communities
  - Communities resemble *cliques*
- Few links between communities
  - Low *cut size:* $\mathrm{Cut}(C) = \ell(C, N \setminus C)$



$N \setminus C$: all nodes *not* in C

The *Dolphins* network has a cutsize of 6

However…



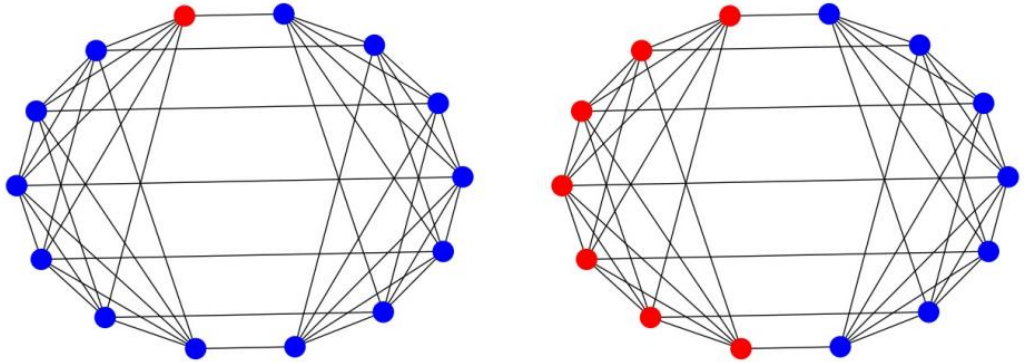
TU/e

Ratio Cut

- Minimizing the cut size often leads to *imbalanced* communities
- Solution: normalize by the sizes

$$\text{RatioCut}(C) = \frac{\text{Cut}(C)}{|C| \cdot (|N| - |C|)}$$

Left minimizes the *Cut,*
right minimizes the *RatioCut*



Work on exercises 4 and 6

TU/e

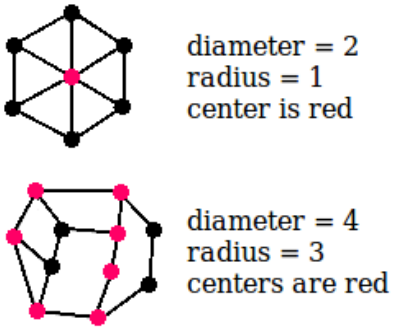
Short distances inside communities

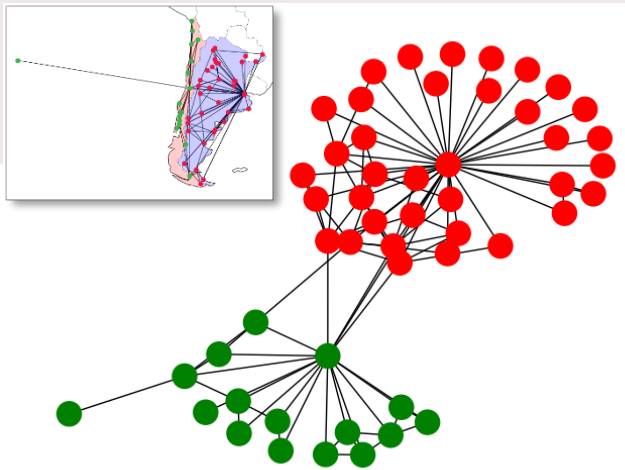
- Distances inside communities are short
- Eccentricity: $\mathrm{Ecc}(i,S) = \max_{j \in S} \mathrm{Dist}(i,j)$

$$\mathrm{Radius}(S) = \min_{i \in S} \mathrm{Ecc}(i,S)$$

$$\mathrm{Diameter}(S) = \max_{i \in S} \mathrm{Ecc}(i,S)$$

- $\mathrm{Diameter}(S) \leq 2 \cdot \mathrm{Radius}(S)$





The k-center algorithm

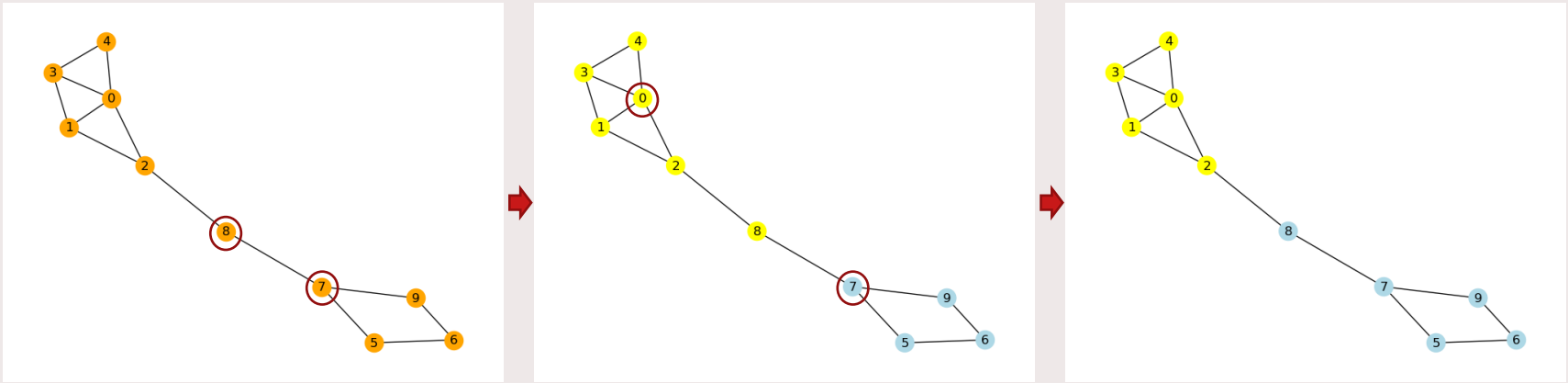
- The k-center algorithm divides the graph into k communities to (approximately) minimize

$$\max_{j=1,\dots,k} \text{Radius}\,(C_j)$$

- Algorithm:
  1. Assigning k vertices as *centers* $c_1, \dots, c_k$
  2. Assign each node to the nearest center: $C(i) \coloneqq \arg\min_{j=1,\dots,k} \text{Dist}(i, c_j)$.
  3. Update centers to the centers of these communities
  4. Repeat from step 2. (until convergence)

The k-center algorithm: example

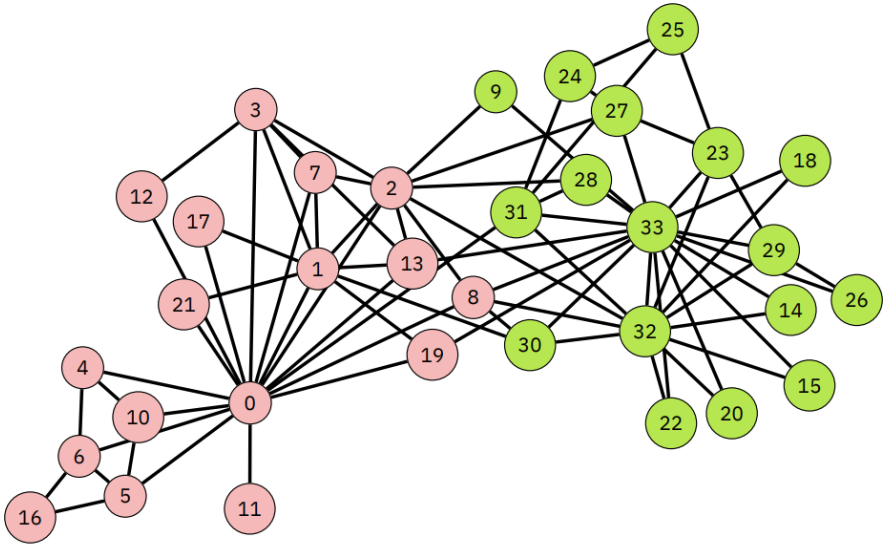
We take nodes 7 and 8 as initial centers:

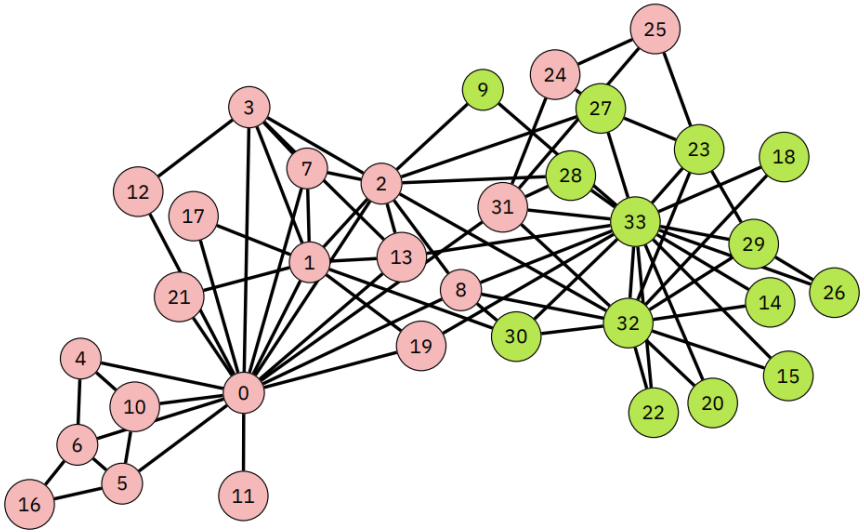


Work on exercise 8

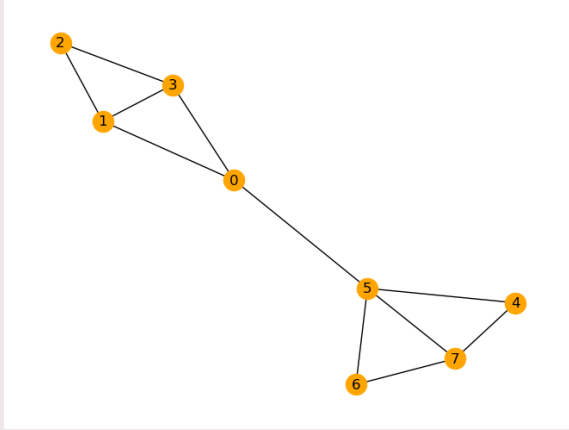
TU/e

Solution to 8





Bottleneck-ness



- Links between communities often form *bottlenecks*
- These links are used by many shortest paths

$$\text{Bottleneckness}(ij) = |\{(s,t) : \text{the link } ij \text{ is on a shortest path from } s \text{ to } t\}|$$

- Divisive community detection algorithm:
  - Repeatedly delete the link with the highest Bottleneck-ness until we are left with k disconnected subnetworks.

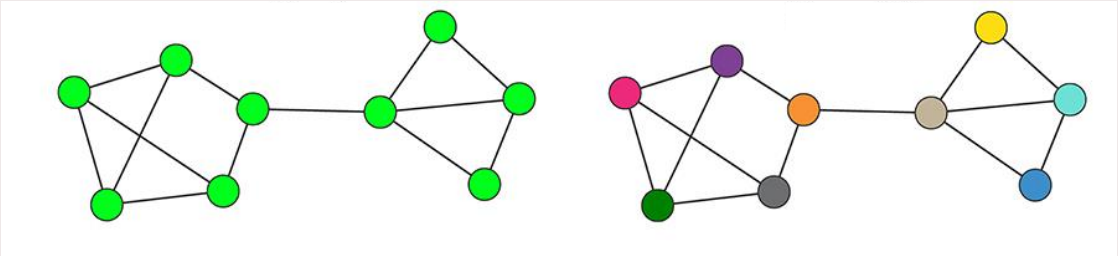
Work on exercise 9

TU/e

The number of communities

Many algorithms require the number of communities to be specified:

- (Ratio)Cut would otherwise be maximized by a single community
- k-center would be optimized by $|N|$ communities
- Bottleneck-ness needs a stopping criteria



Random networks

Networks with communities tend to have more edges inside the communities than one would expect in a *random network* without communities.

Two main models for randomness:

- 'Fully random model': randomly place $|E|$ edges among the $\binom{|N|}{2}$ node pairs.
  - For each pair, the probability of having an edge is $\frac{|E|}{\binom{|N|}{2}}$.
- '[Configuration model]': randomly place $|E|$ such that the degrees are equal to the degrees of the given network

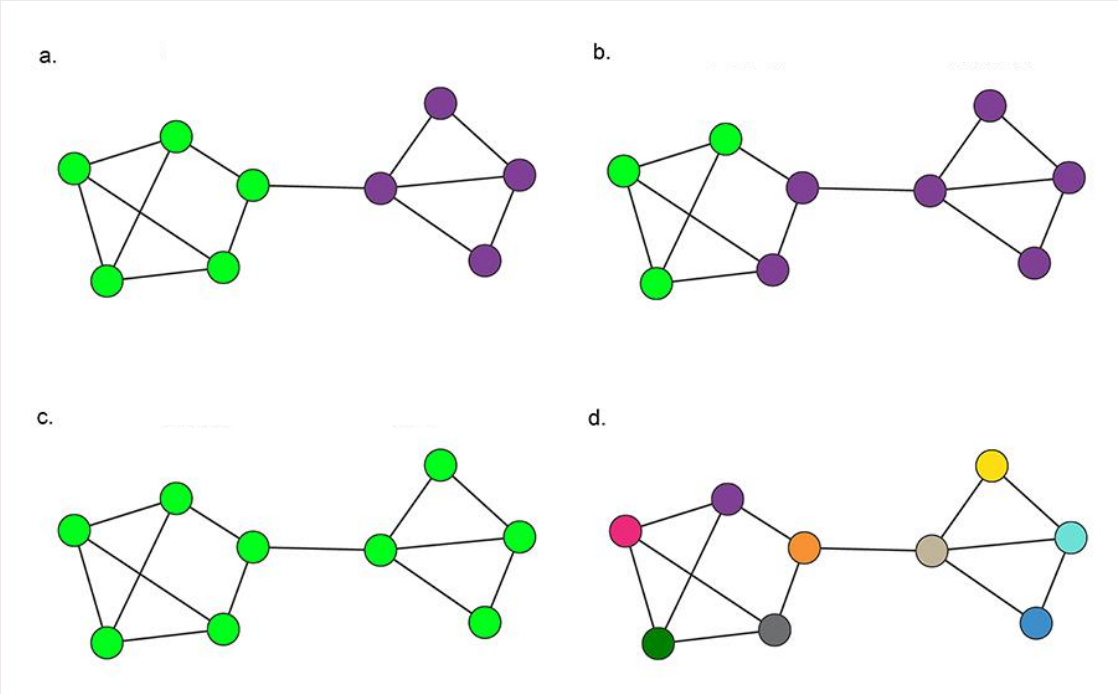
Fully random model: Simple modularity

Modularity: Fraction of edges inside communities minus expected fraction inside communities.

Simple modularity: Fraction of edges inside communities minus fraction of node-pairs inside communities.

$$\text{SimpleModularity}(C) = \frac{\sum_{i=1}^{|C|} \ell(C_i)}{|E|} - \frac{\sum_{i=1}^{|C|} \binom{|C_i|}{2}}{\binom{|N|}{2}}$$

Example

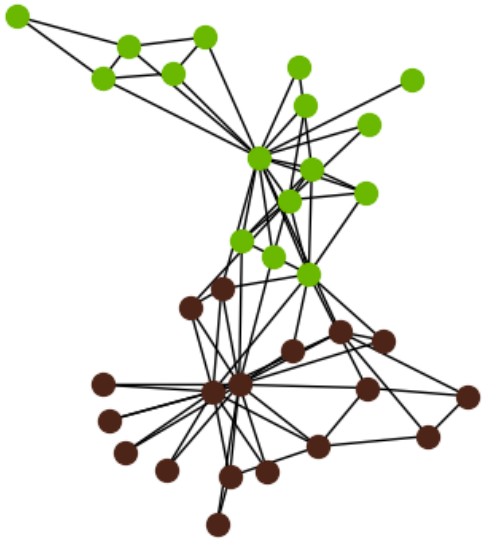


Configuration model: Standard modularity

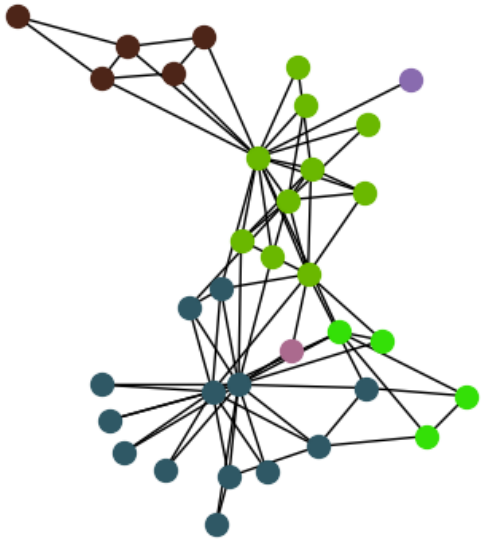
$$\text{StandardModularity}(C) = \frac{1}{|E|}\left(\sum_{i=1}^{|C|} \ell(C_i) - \frac{d(C_i)^2}{4|E|}\right)$$

Example: Karate network

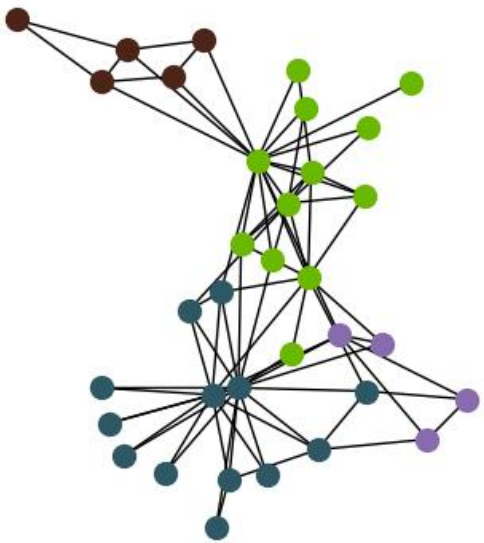
True communities



Simple modularity



Standard modularity

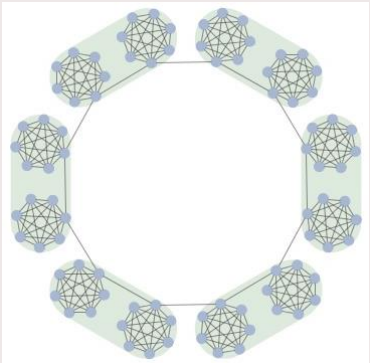


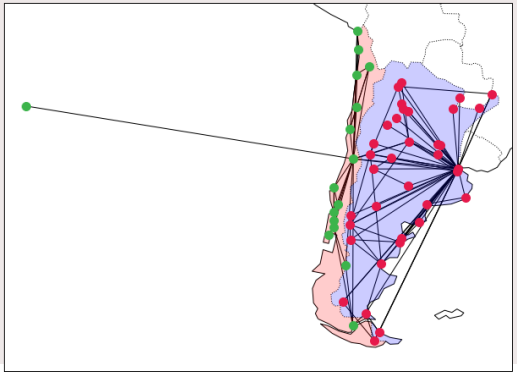
Work on exercises 10 and 11

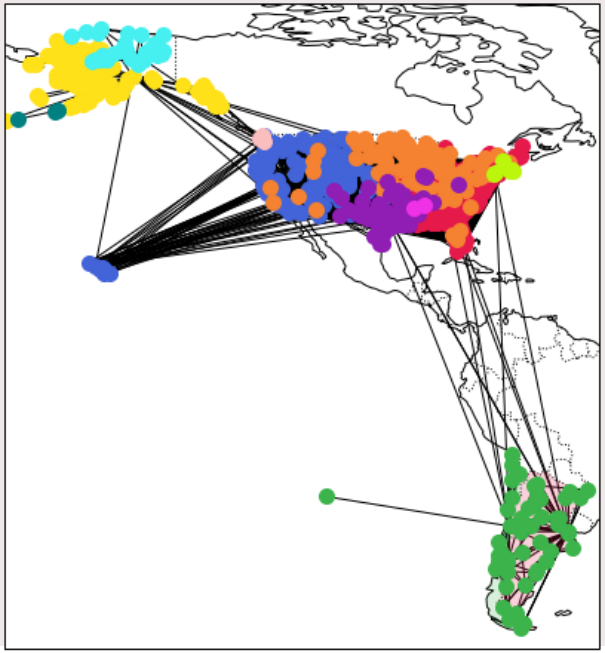
TU/e

Resolution limit of modularity

- The expected number of edges between two communities of fixed sizes vanishes for large n
- Thus, a single edge would be enough to merge



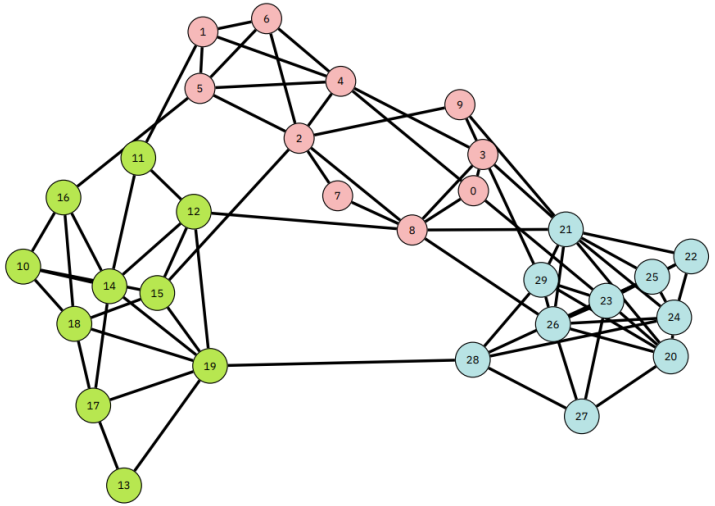




Models of communities

- Which community-detection method works well in what setting?
- Planted Partition Model
  - Edge-probability p_{in} inside communities,
  - Edge-probability p_{out} between communities.

$$\mathbb{P}(ij \in E) = \begin{cases} p_{in} \text{ if } C(i) = C(j), \\ p_{out} \text{ if } C(i) \neq C(j). \end{cases}$$



Work on exercises 12-18

TU/e

Summary

Different characteristics of communities in networks

- Few links between communities → Minimize (Ratio)Cut
- Short distances inside communities → k-center
- Links between communities are ‘bottlenecks’ → Eliminate bottlenecks
- Many links inside communities → Maximize modularity

Thank you for your attention!