Network Science
SICSS Summer School

Program

1st hour:

Introduction to networks

Types of analysis

Networks in Python

2nd hour:

Network representation

Linear algebra

Networks at CBS

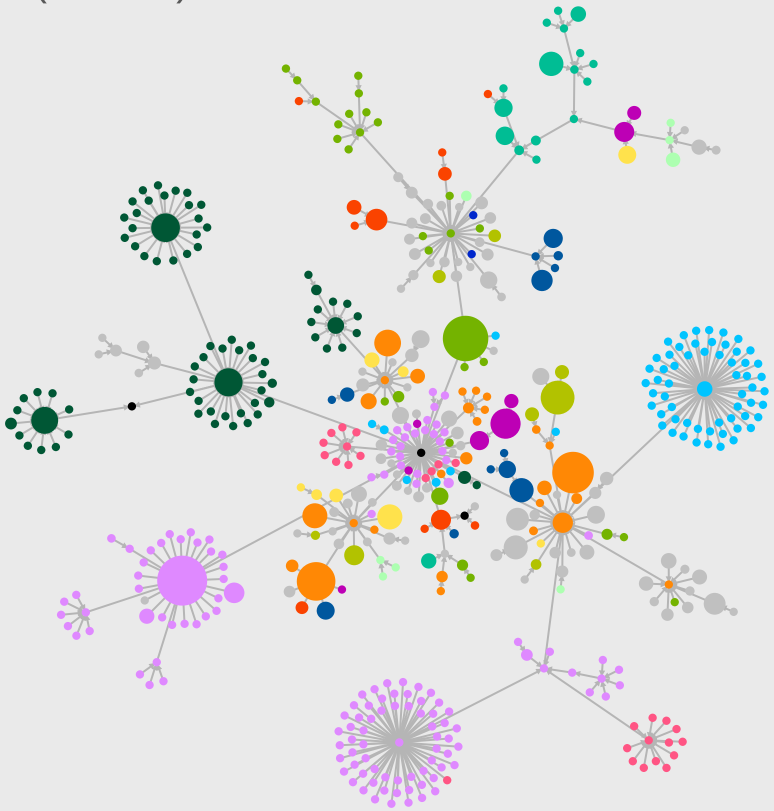
Introduction to networks

What is a network?

Mathematical representation of the relationships (edges) between entities (nodes)

The most important question to ask yourself:

What are the nodes and what are the edges?



Types of networks

	Network	Nodes	Edges
Social	Friendship	People	Friendships
	Follower	Online accounts	Followers/likes
	Psychological	Symptoms	Co-ocurrence
Biology	Gene regularory	Genes	Activations/inhibations
	Food web	Animals	Predating
Economic	Trade	Countries/companies	Money flows
	Ownership	Companies	Ownership stakes
Intrastructure	Internet	Computers (IPs)	Data transmission
	Power grid	Power stations	Power lines
	Airplane network	Airports	Flights

https://aaronclauset.github.io/courses/5352/csci5352_F21_L1.pdf

Type of networks and characteristics

Type 1: Interaction and flow → "Real networks".

- Offline interactions
- Online interactions

Type 2: Affiliation → Node 1 is part of/related to node 2

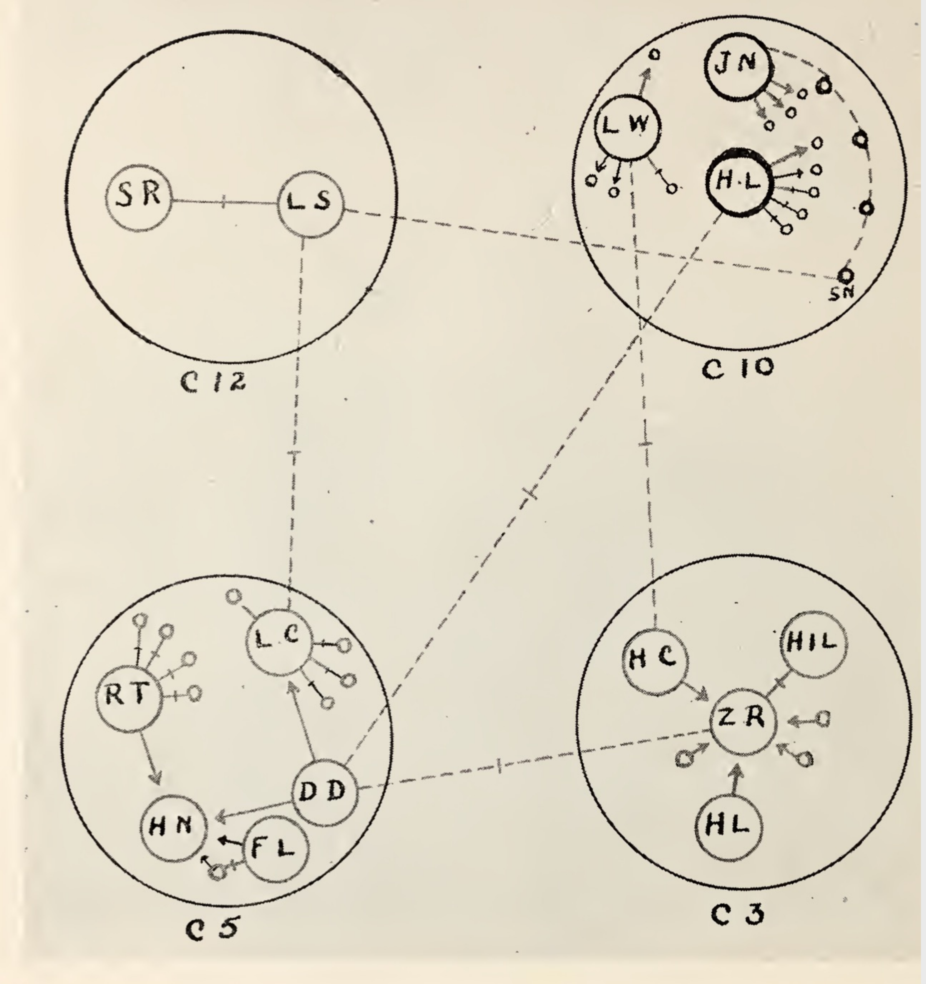
- Most administrative data: e.g. students in classrooms
- Bipartite networks

Type 3: Co-occurrence → Node 1 is correlated with node 2

- Stock market networks
- Brain networks

What about family networks?

Brief history of social network science:



Moreno. Who shall survive?

Mathematical **representation** of an underlying system (not the system itself)

Network science: Social and behavioral scientists in the XX century (e.g. Jacob Moreno & Hellen Hall Jennings, Harrison White, Mark Granovetter)

- Hellen Hall Jennings and Jacob Moreno (1930s): Hudson School for girls: Sociometry. Networks can represent the systems and how information spreads
- Jeffrey Travers and Stanley Milgram's (1969): Small-world studies
- Nancy Howell (1969): *The Search for an Abortionist*, women acquired scarce information through short chains of weak ties.
- Mark Granovetter (1973) *The Strength of Weak Ties.* Diffusion of information takes place primarily through bridges (weak ties). Strong links are redundant.
- Harrison White (1976): Blockmodels for networks
- Duncan Watts, Steven Strogatz (1998): Next wave of network science

Why do we care?

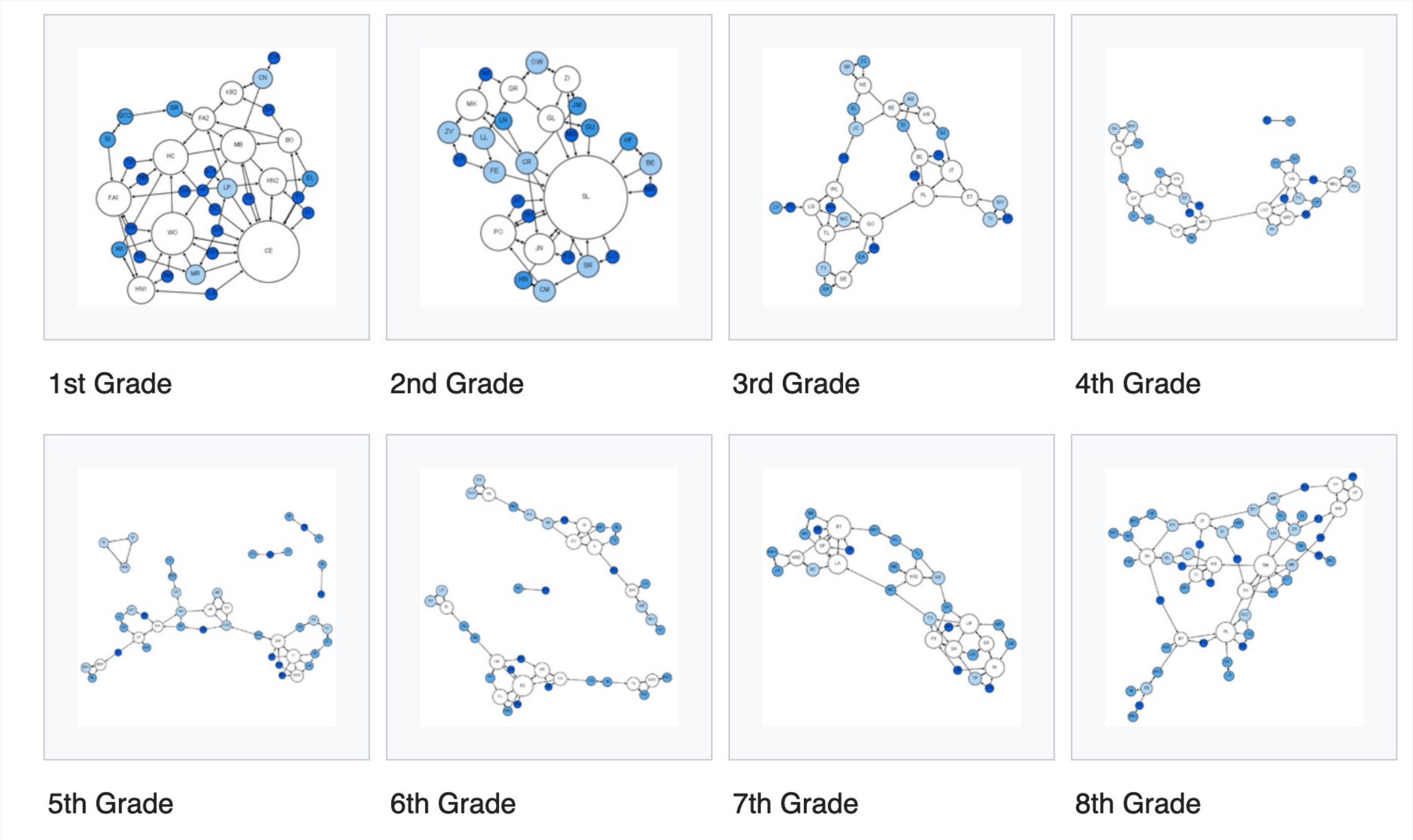
Theoretical links to social science (dangerous generalizations below!):

- Social capital: The position of an individual in their social network (embeddedness) determines opportunities and outcomes.
- Network measures map to social theories: e.g. structural holes and network closure (Burt, 2001)
  - **Structural holes**: social capital is created by a network in which people can broker connections between otherwise disconnected segments ~ similar to betweenness centrality
  - **Network closure**: social capital is created by a network of strongly interconnected element ~ clustering coefficient

Networks:

- Reflect preferences (**selection**)
- **Influence** us: spread of information, diseases, opportunities

Why do we care?



Moreno, source: wikipedia

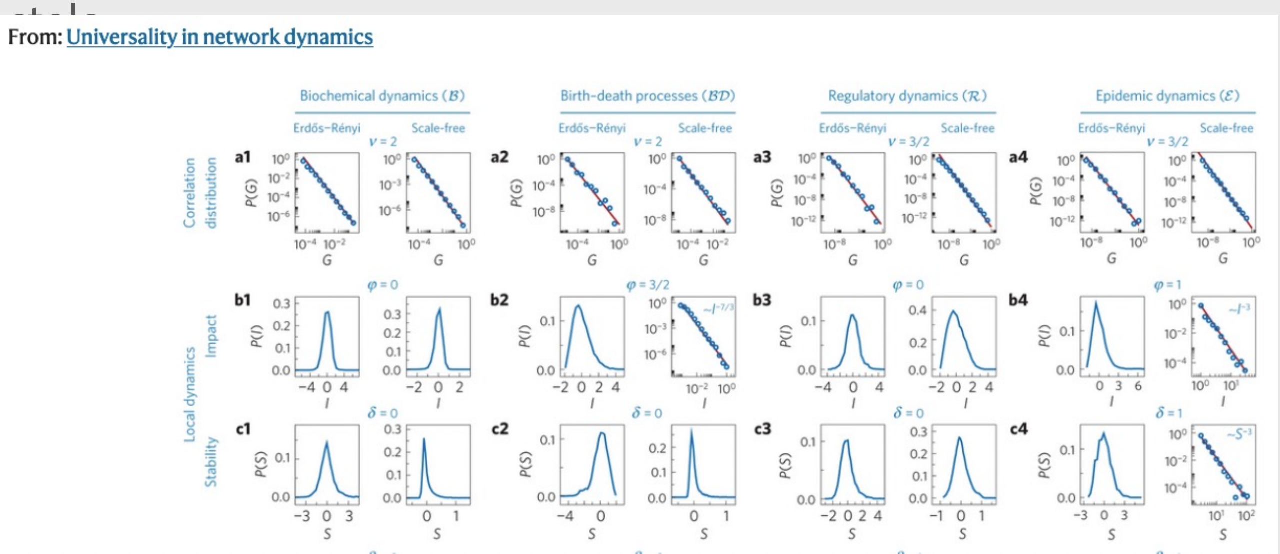
Why do we care?

Mathematical **representation** of an underlying system (not the system itself).

Allow us to find insights that we would miss if we would study the nodes independently (one person != society)

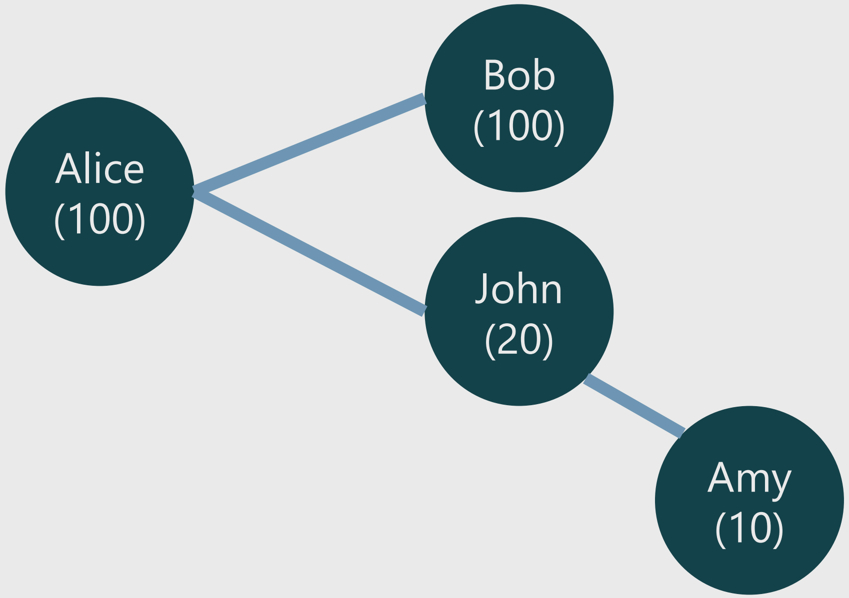
Complex systems view (dangerous generalizations below!):

- Network structure determines how information/epidemics spread
- Interested in emergent behaviours:
  - Universality / scale-invariance (heavy tails) / fractals
  - Phase transitions and percolation



Basic definitions

Networks (graphs)



Nodes (vertices) connected by **edges** (links)

N: **Nodes** = {Alice, Bob, John, Amy}

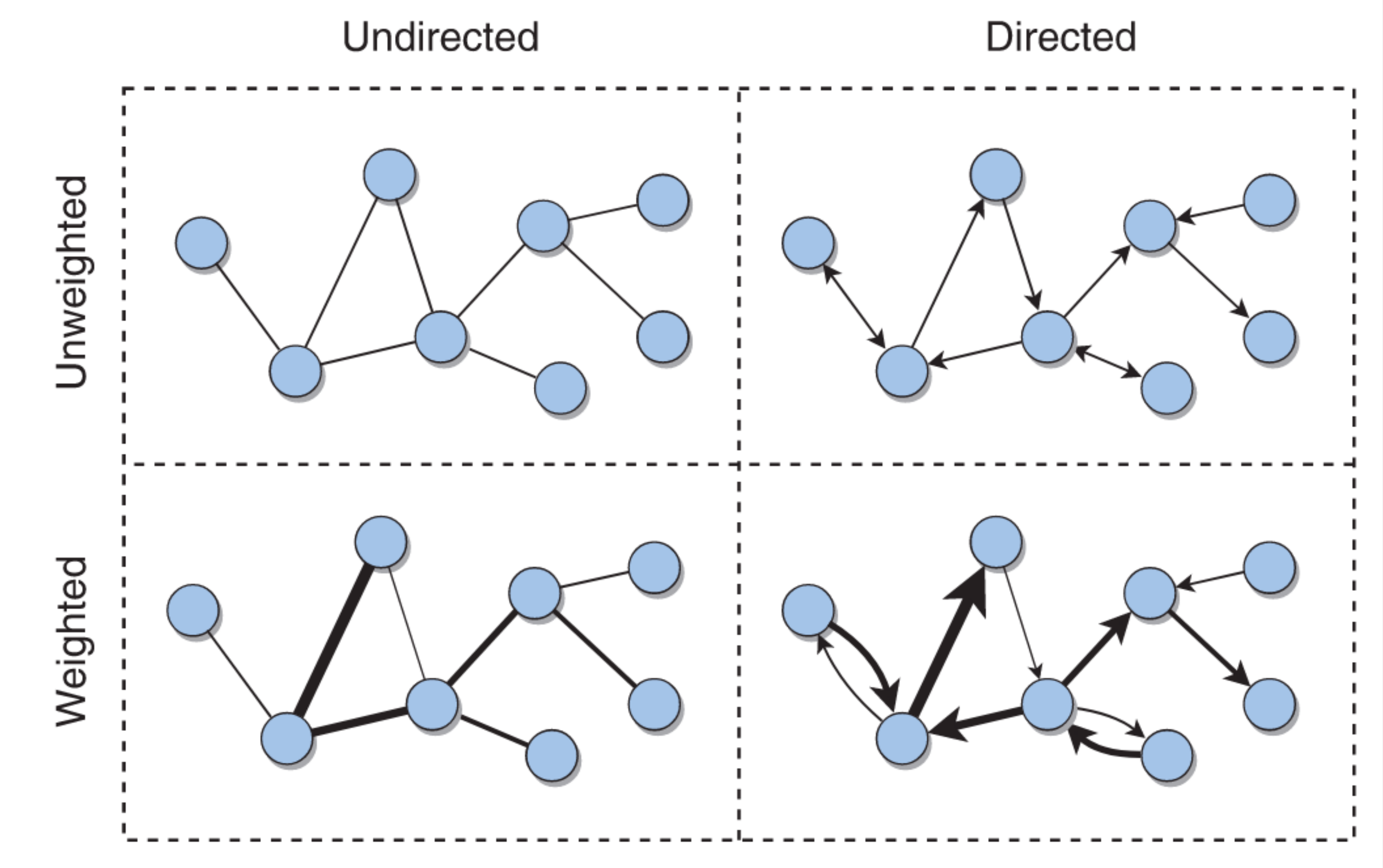
E: **Edges** = {(Alice, Bob), (Alice, John), (John, Amy)}

The edge (i,j) connects node i to node j

Nodes can have **attributes** (e.g. gender, income, etc)

Edges can have **attributes** (e.g. type, strength, etc)

Directed vs undirected; weighted vs unweighted



Source: A first course in network science (2020)

Undirected: The link (i,j) connects node i to node j in both directions

Directed: The link (i,j) connects node i (source) to node j (target)

Weighted: There is a weight associated to each edge

Degree in undirected networks

Definition: Number of neighbors in the network

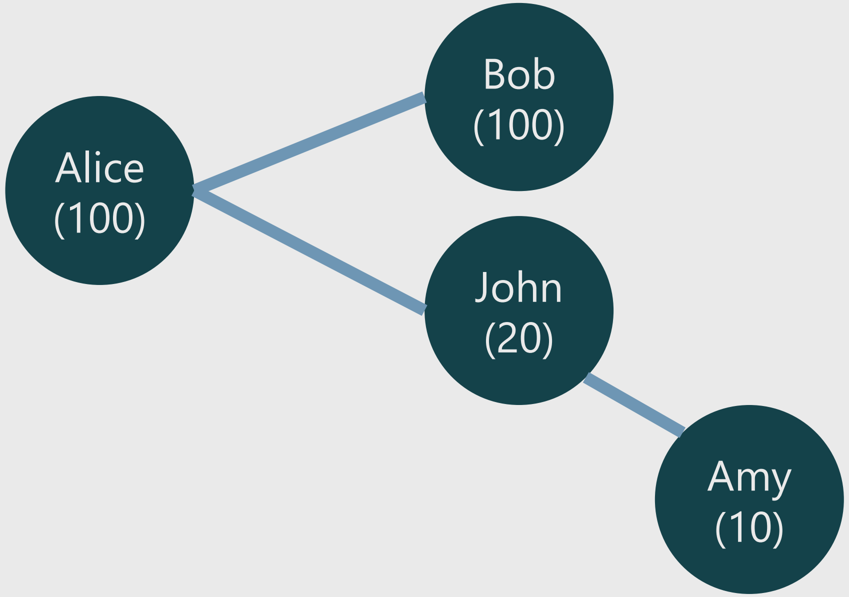
Node: degree

Alice: 2

Bob: 1

John: 2

Amy: 1



Degree in directed networks

Out-degree: Number of outgoing edges

In-degree: Number of incoming edges

Total degree: Sum of out and in degree

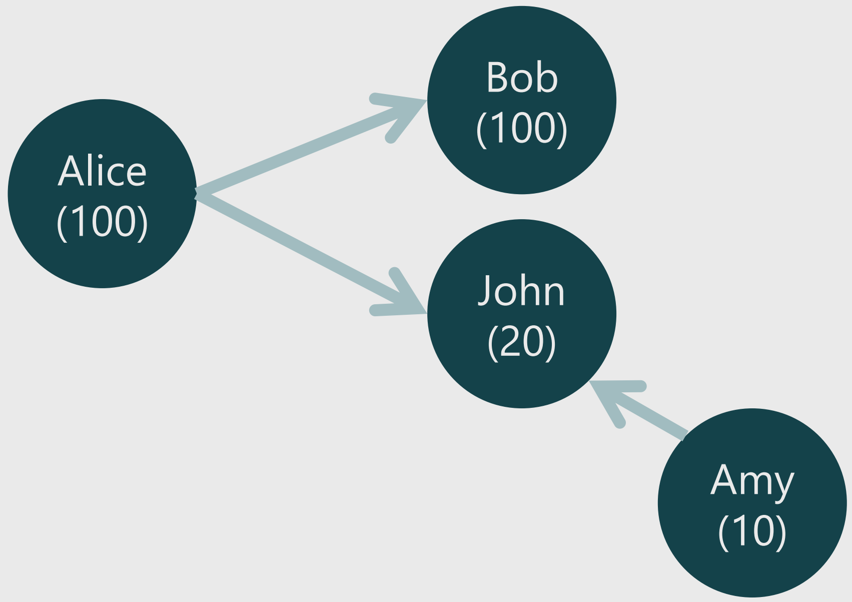
Node: (out, in, total)

Alice: (2, 0, 2)

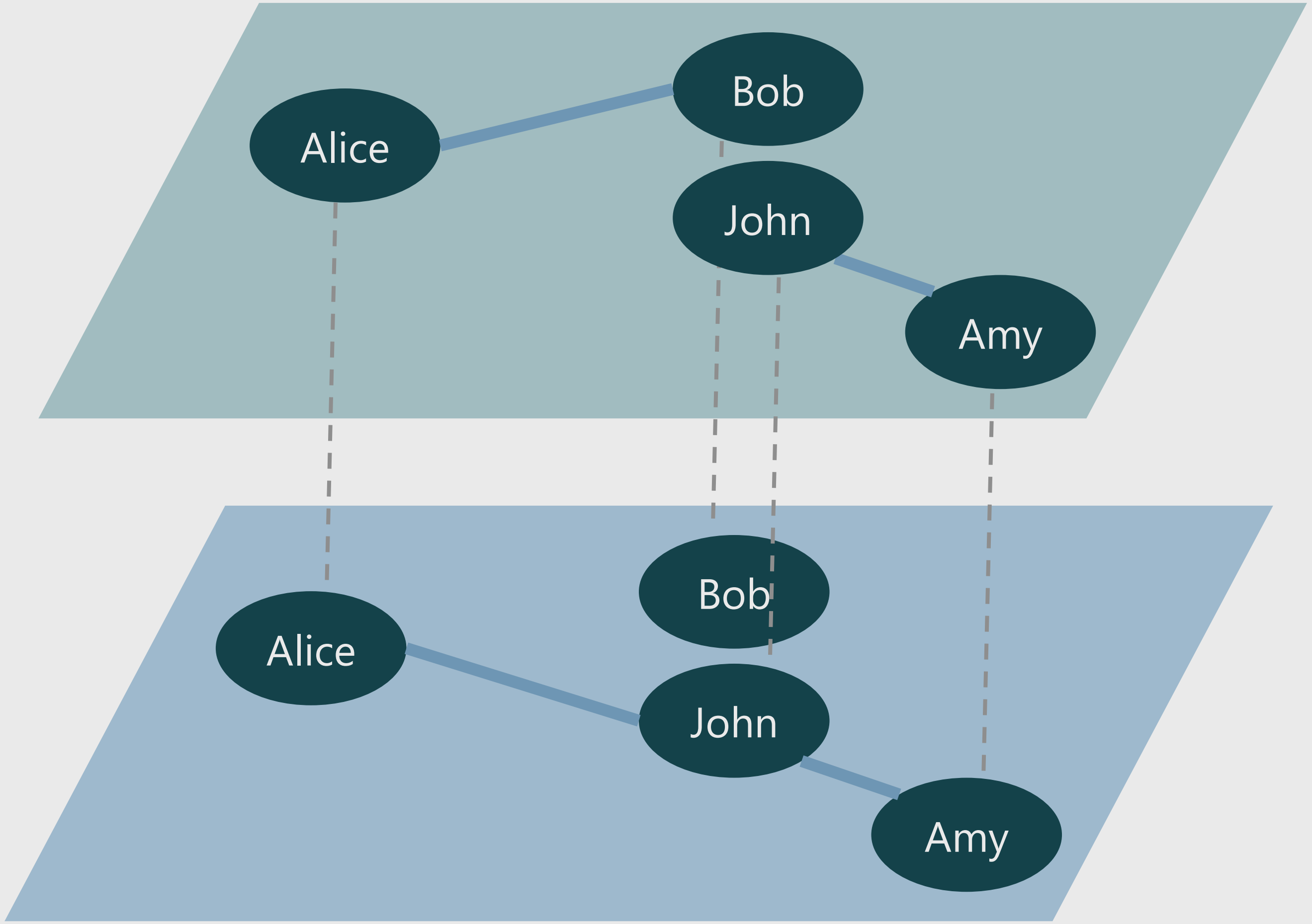
Bob: (0,1, 1)

John: (0,2, 2)

Amy: (1,0, 1)

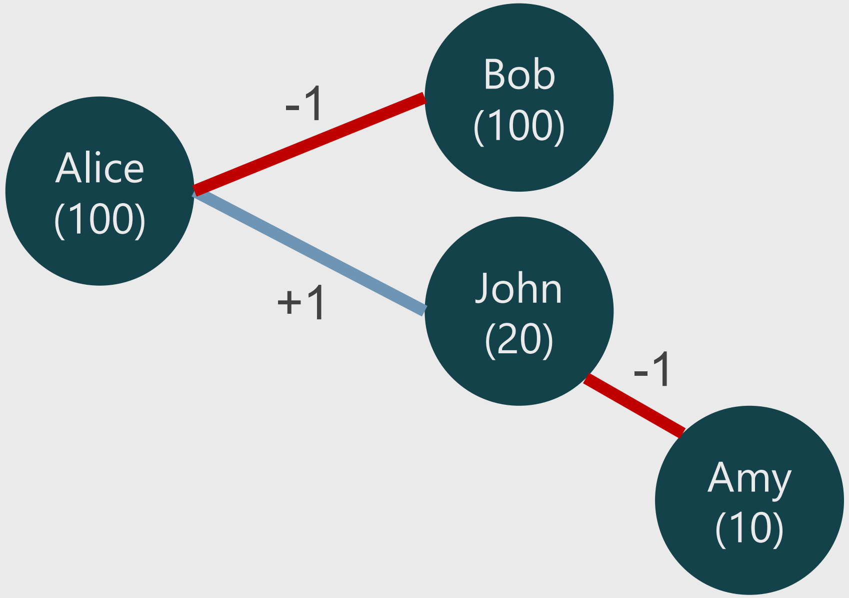


Other types of networks: Multiplex



Other types of networks: Signed

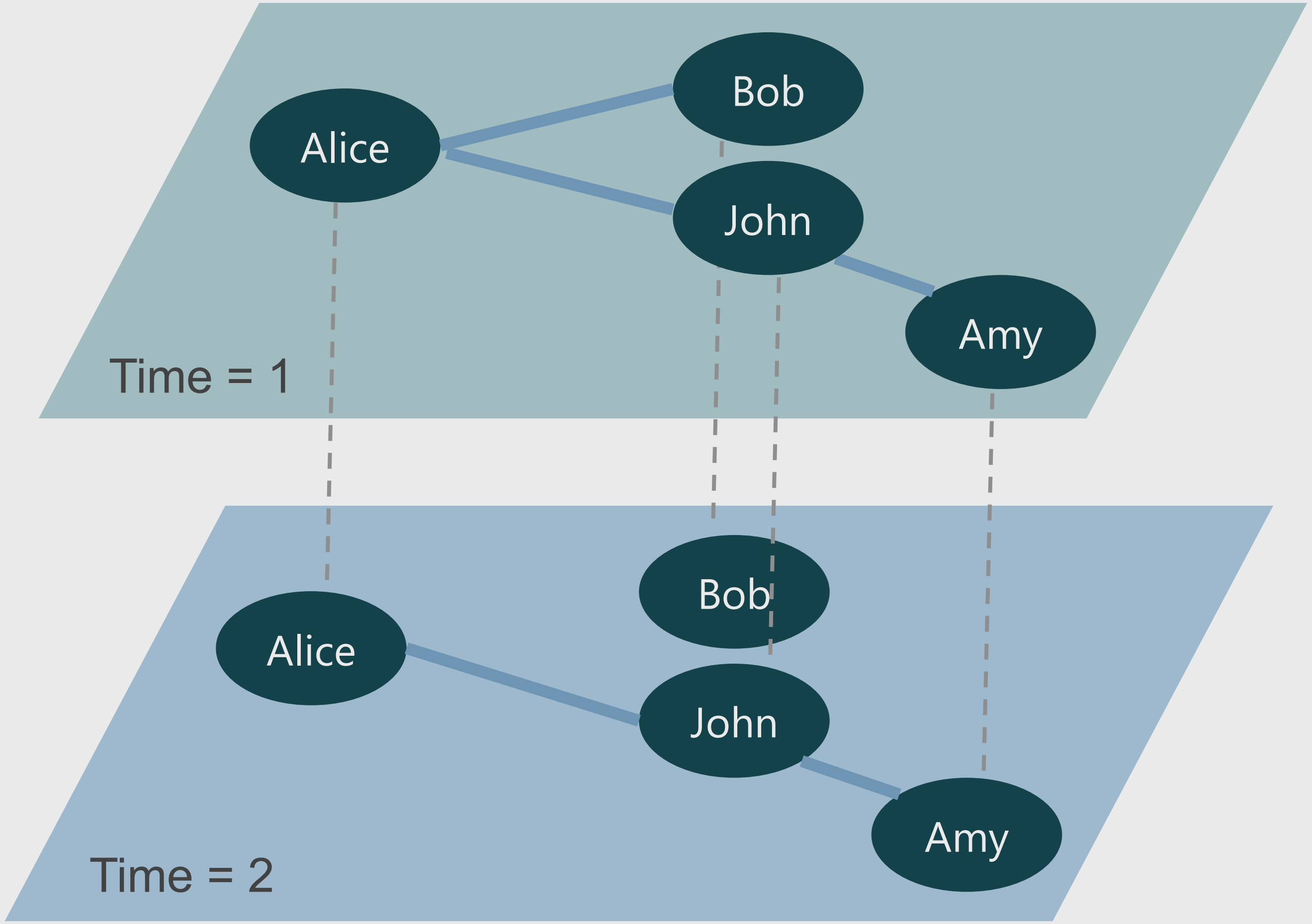
Structural balance



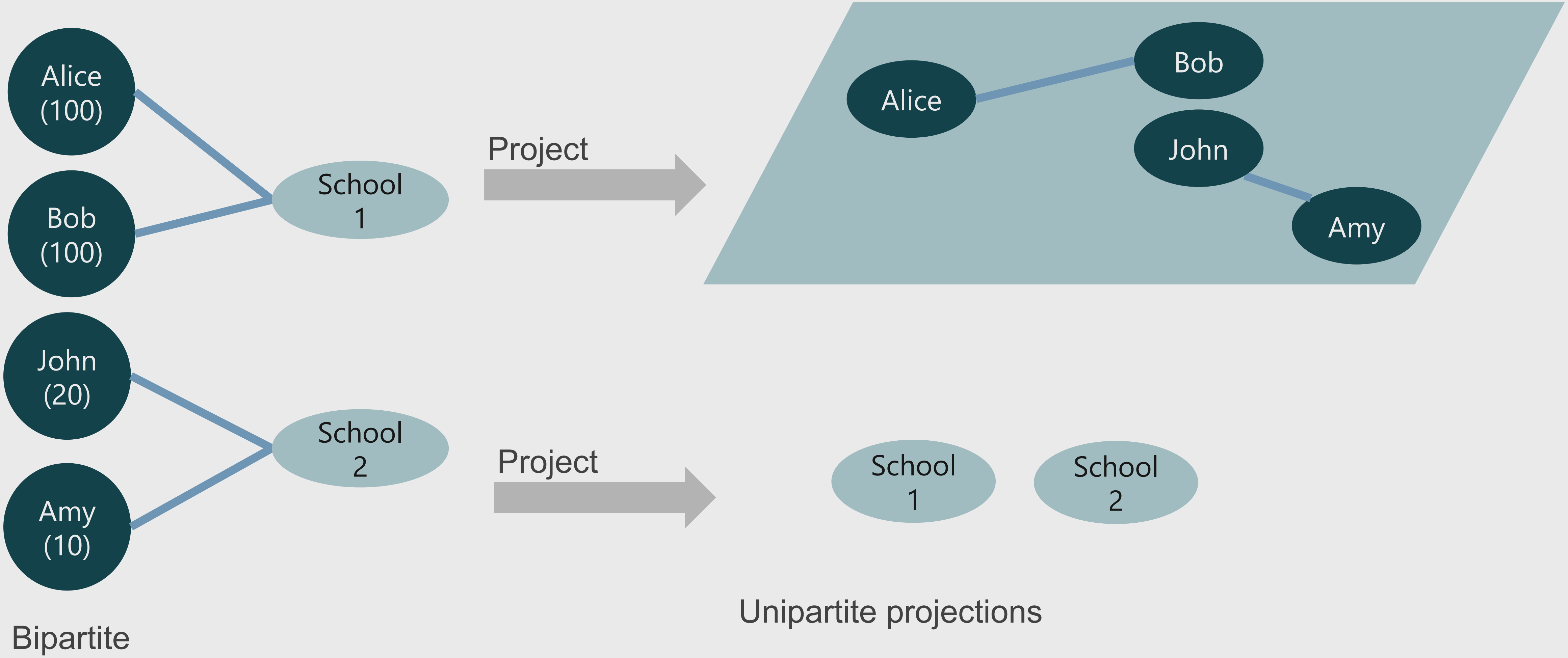
Other types of networks: Temporal

Either:

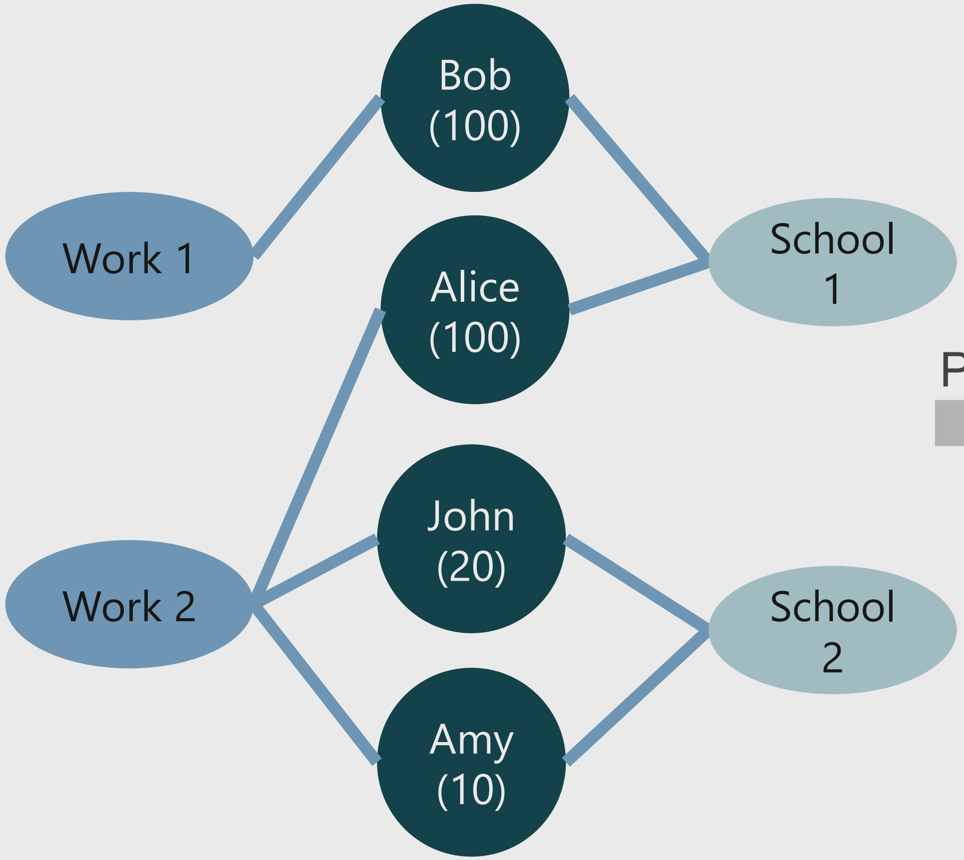
- Snapshots
- Time of events



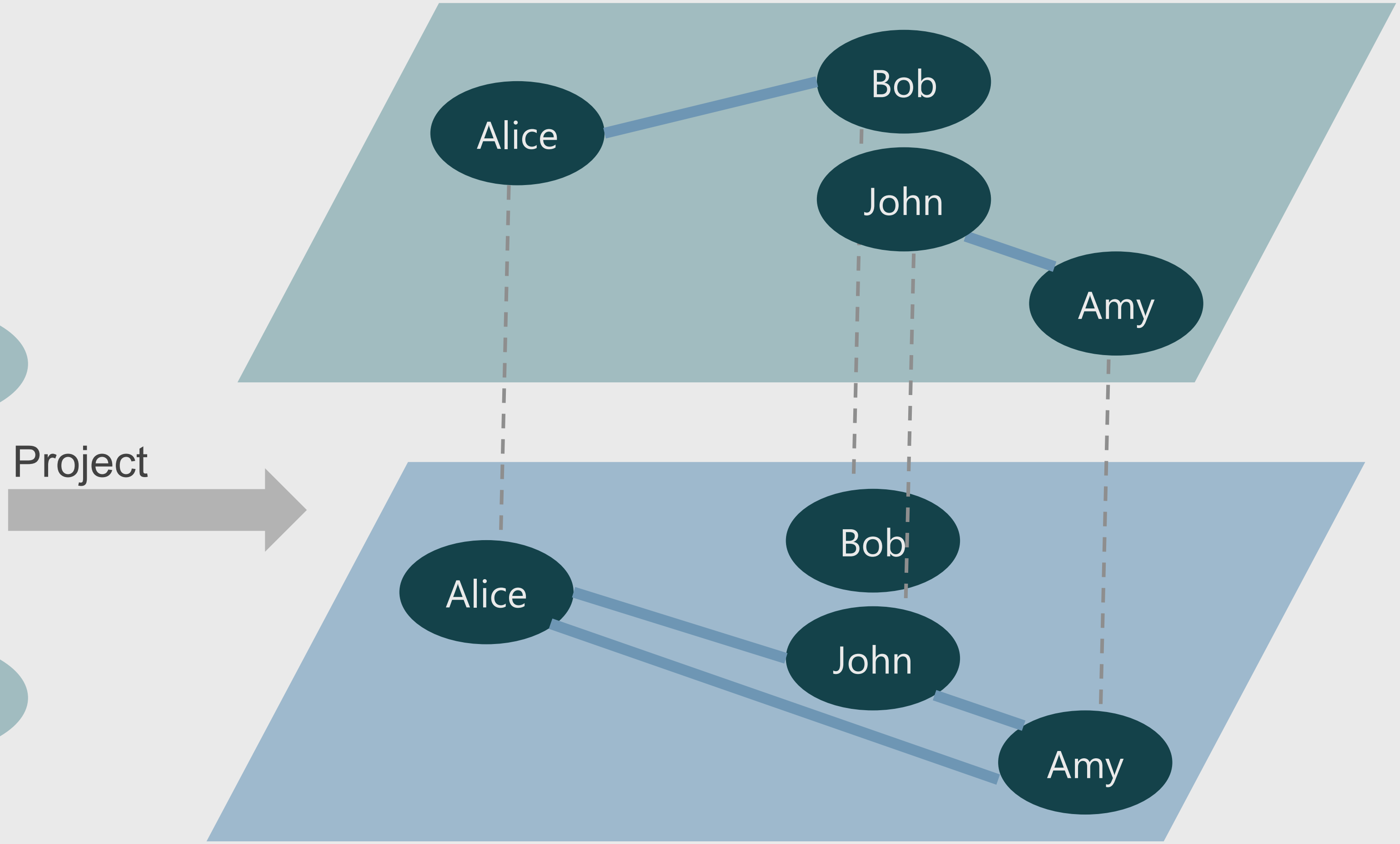
Other types of networks: Bipartite



Other types of networks: Multipartite

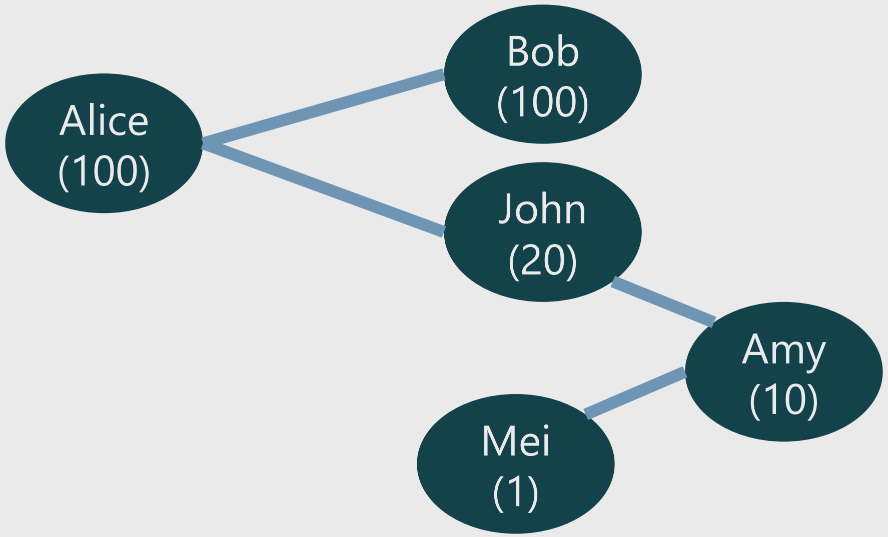


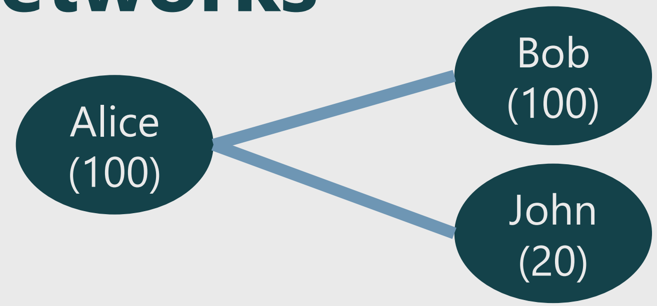
Multipartite network



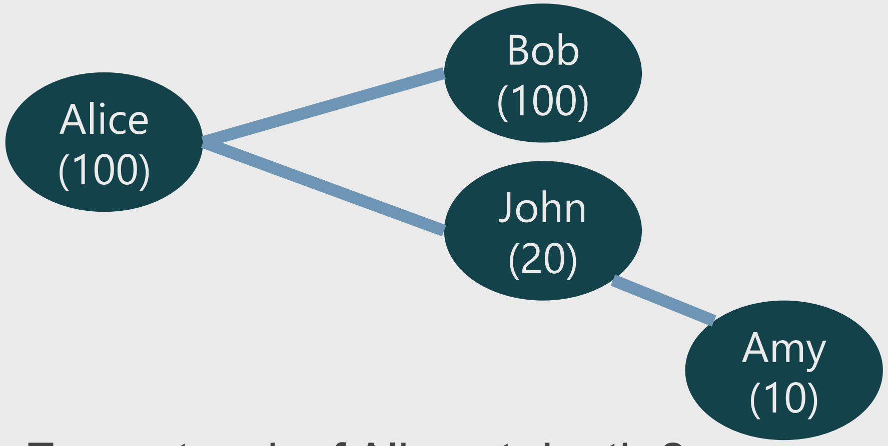
Multiplex projection

Other types of networks: Ego-networks



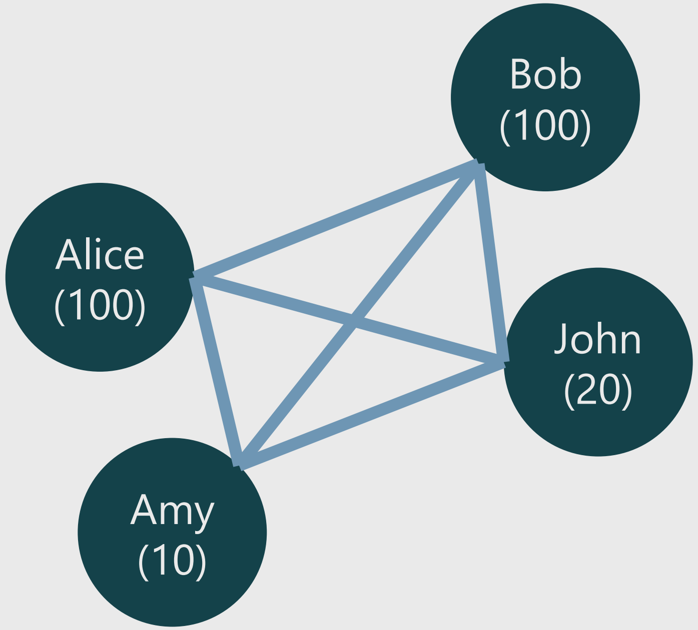


Ego network of Alice at depth 1



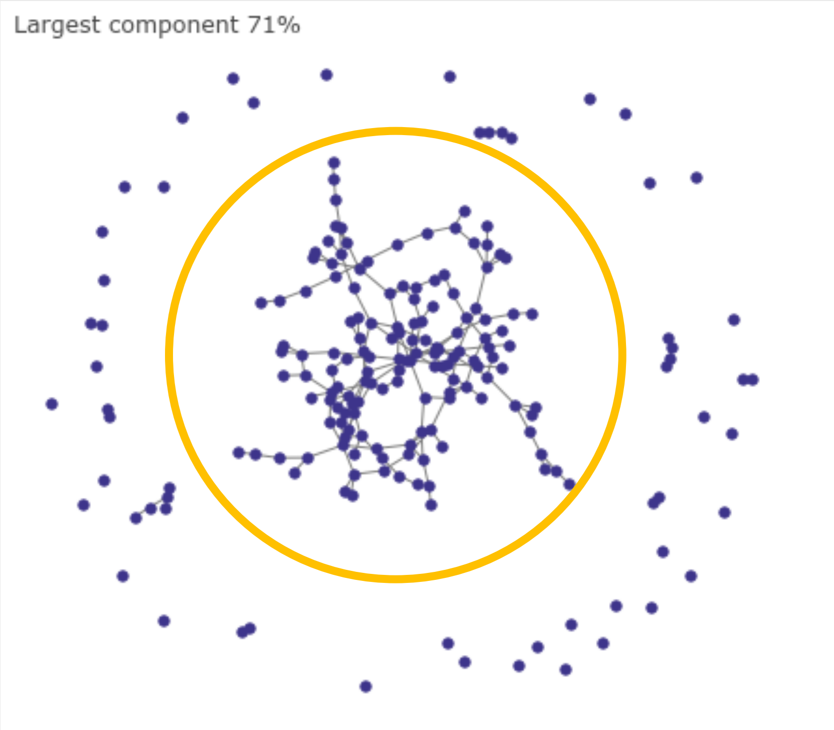
Ego network of Alice at depth 2

Other types of networks: Clique



Network characteristics

Connectedness



Real networks are typically connected, forming a **"giant component"**

If the average degree < 1 → many small components

If the average degree > 1 → suddenly the system becomes connected

Let's try this!

Small world: six degrees of separation

Milgram's experiment: six degrees of separation

Strogatz, Watts: small number of random links are enough to create small world networks

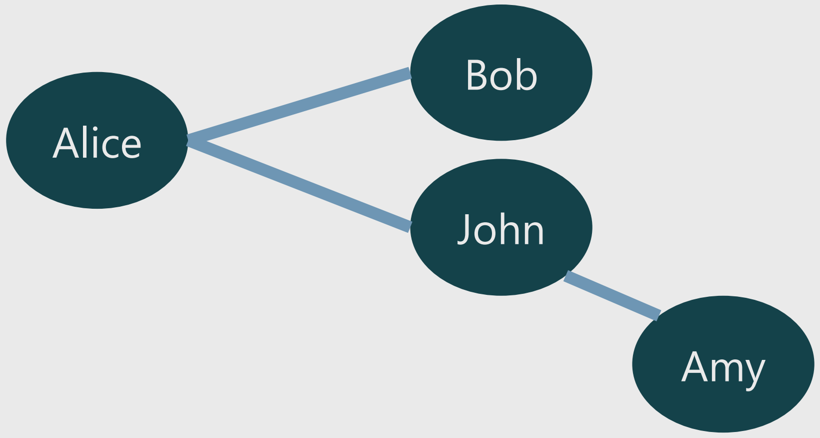
Shortest path between node 1 and node 2:

- Minimum number of steps requires to go from node 1 to node 2
- Between Alice, Amy → 2

Diameter:

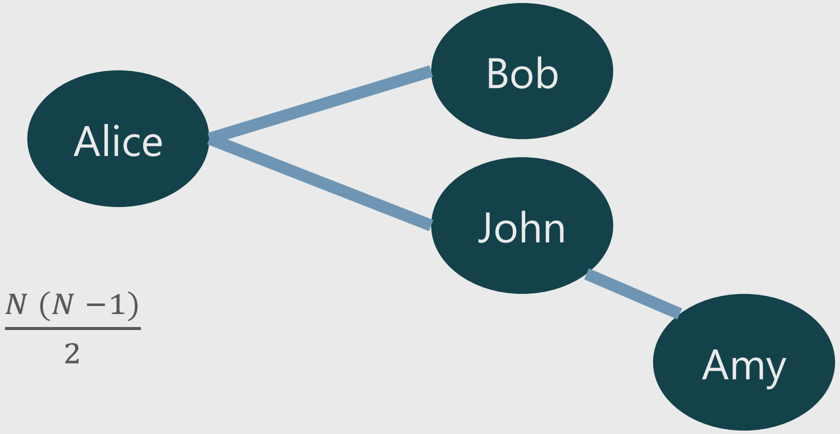
- Longest "shortest path" between two nodes
- In our network: 2 (Alice -> John -> Amy)

Real networks have **small diameters** (hubs facilitate this)



Density

Definition: Number of edges present / potential number of edges



- Number of edges = 3
- Potential number of edges in directed network = (4*3)
- Potential number of edges in undirected network = (4*3)/2 $= \binom{N}{2} = \frac{N\,(N-1)}{2}$

Density = 3/6 = 50%

Real networks are typically **sparse**

As size increases density decreases (average degree is usually fixed)

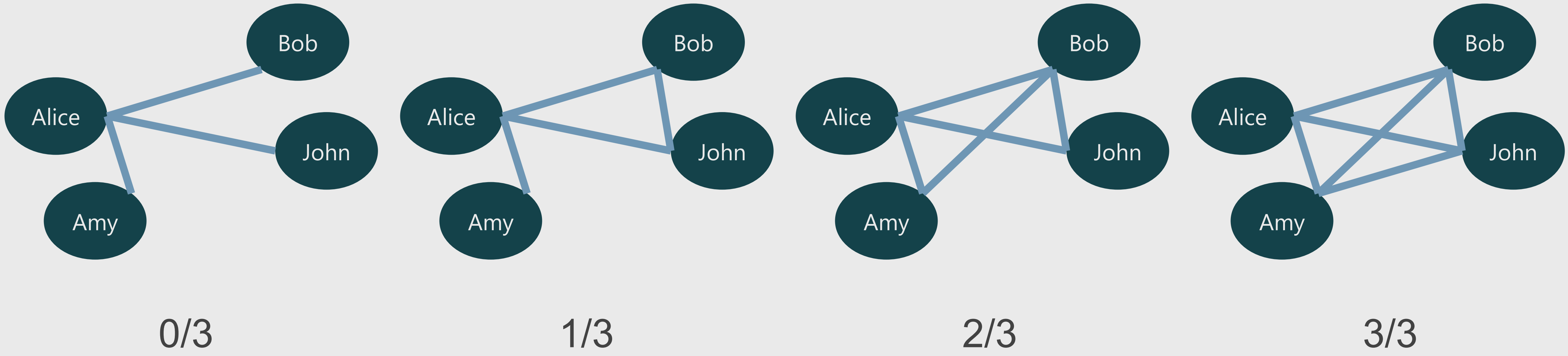
Local clustering (~transitivity)

Strogatz, Watts (1998): How many of your neighbors are connected to each other

Average clustering of a network: Average clustering of the nodes

Real networks have **high clustering**

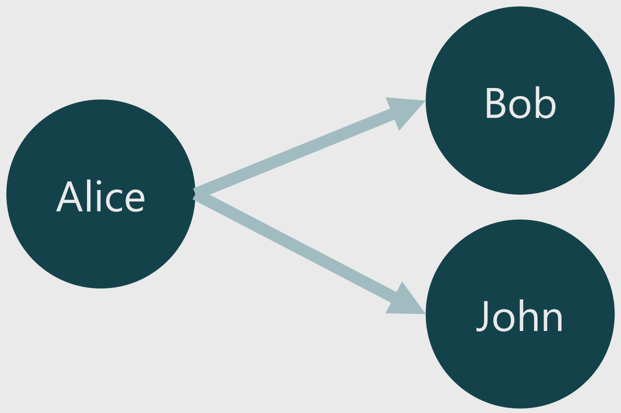
Clustering of Alice:

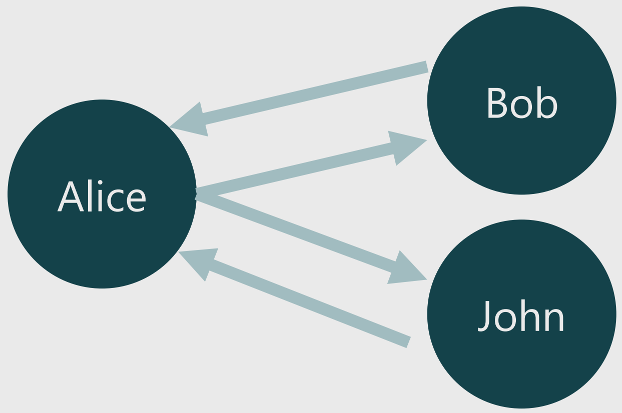


Reciprocity

Directed networks

Ratio of the number of edges pointing in both directions to the total number of edges in the graph.

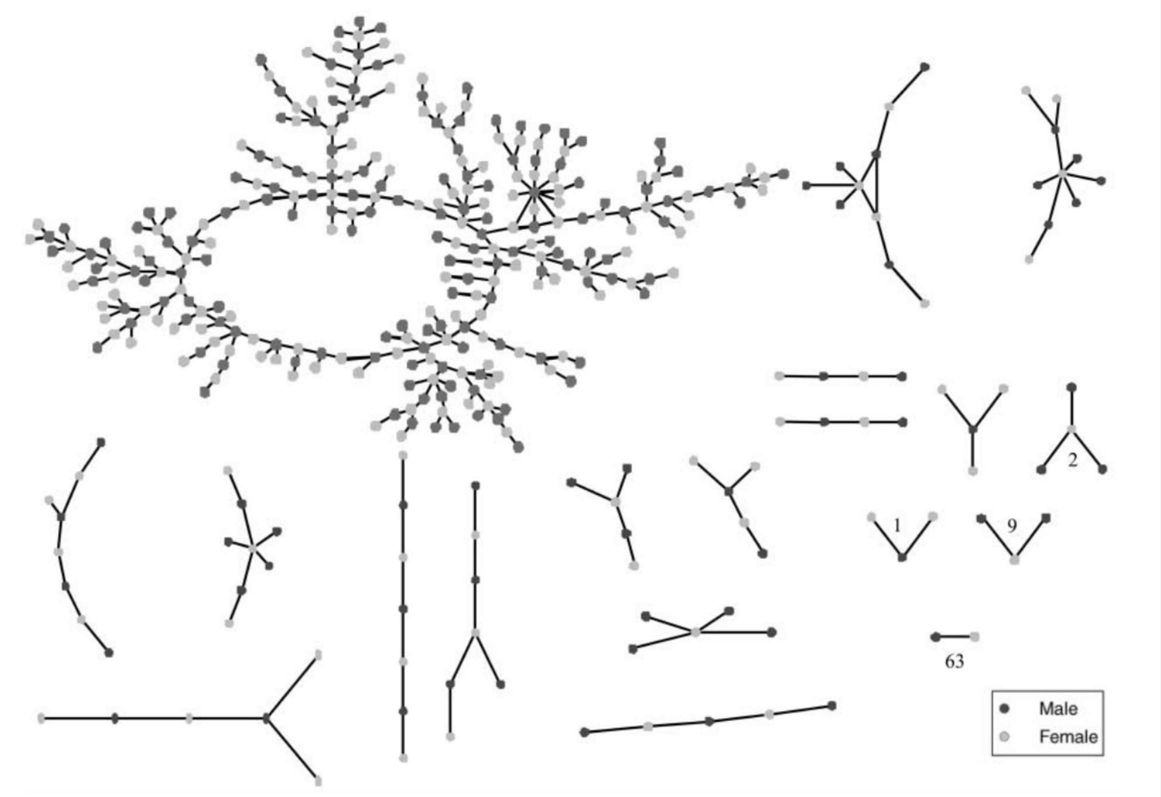




Assortativity (homophily)

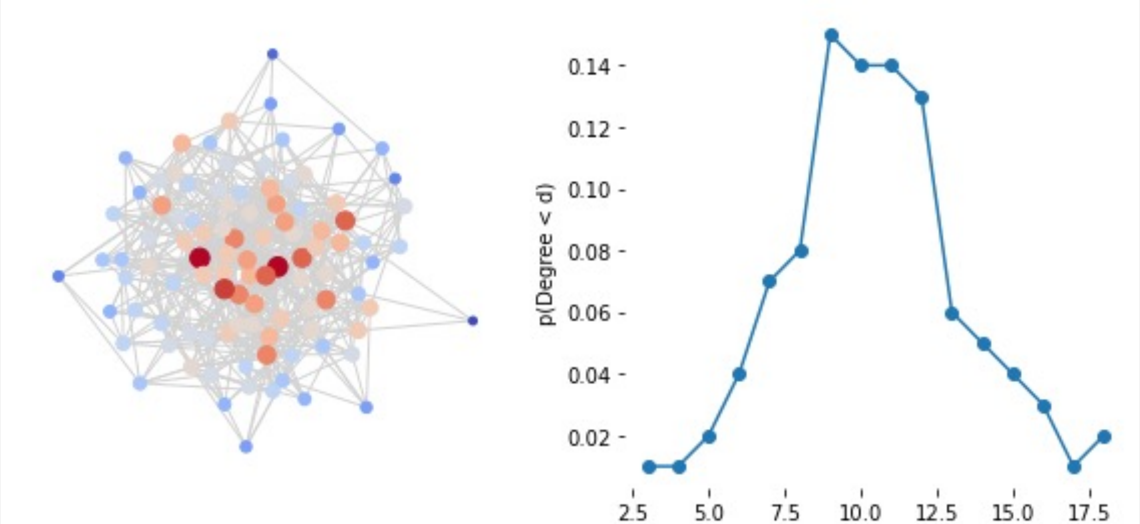
Preference for nodes to attach to others that are similar in some way

Paraisópolis favela and Morumbi, in São Paulo
Photography by Tuca Vieira (the guardian)

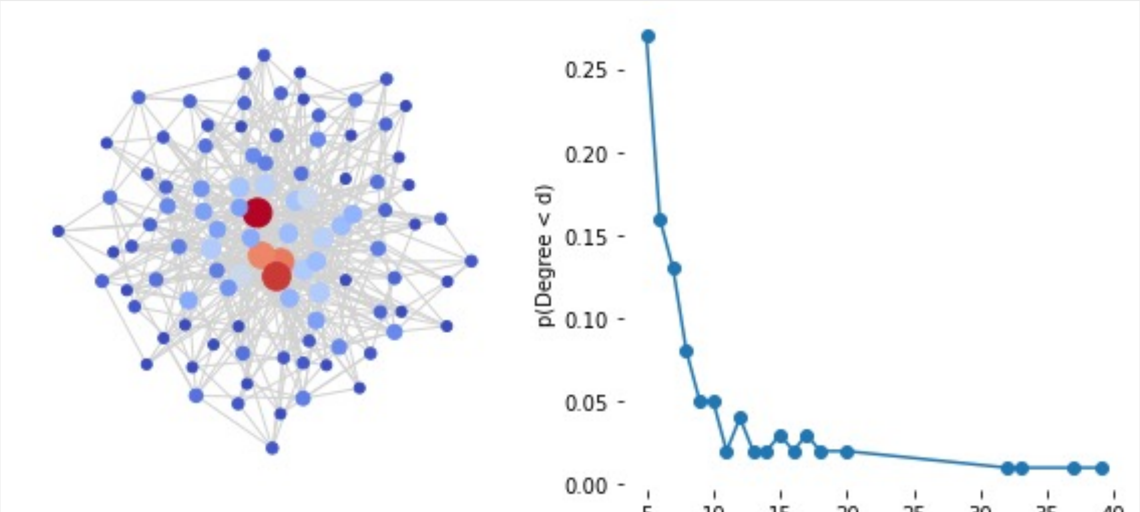


Romatic links between teenagers
Bearman, Moody, Stovel (1991)v

Heavy tails / scale-free



Random network



Scale-free network

Networks are not random, they have heavy degree distributions

PDF (probability density function)

→ Degree vs probability of degree

→ Represented by histogram

Many possible mechanisms:

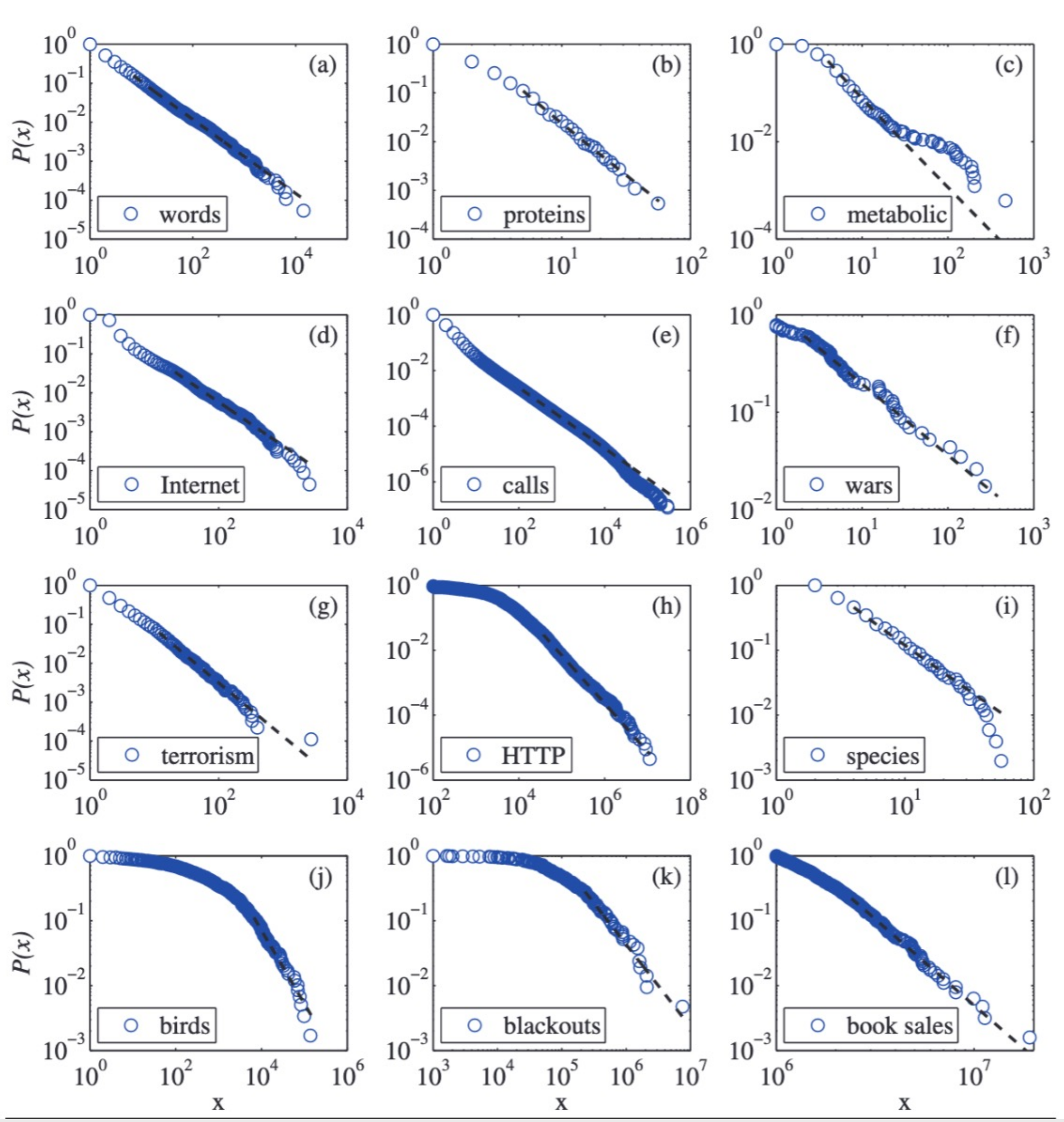
- Multiplicative growth
- Preferential attachment (Rich get richer, Mathew effect)
- Copying models

Growing networks: https://www.stat.cmu.edu/~cshalizi/networks/16-1/lectures/08/li.pdf

Heavy tails

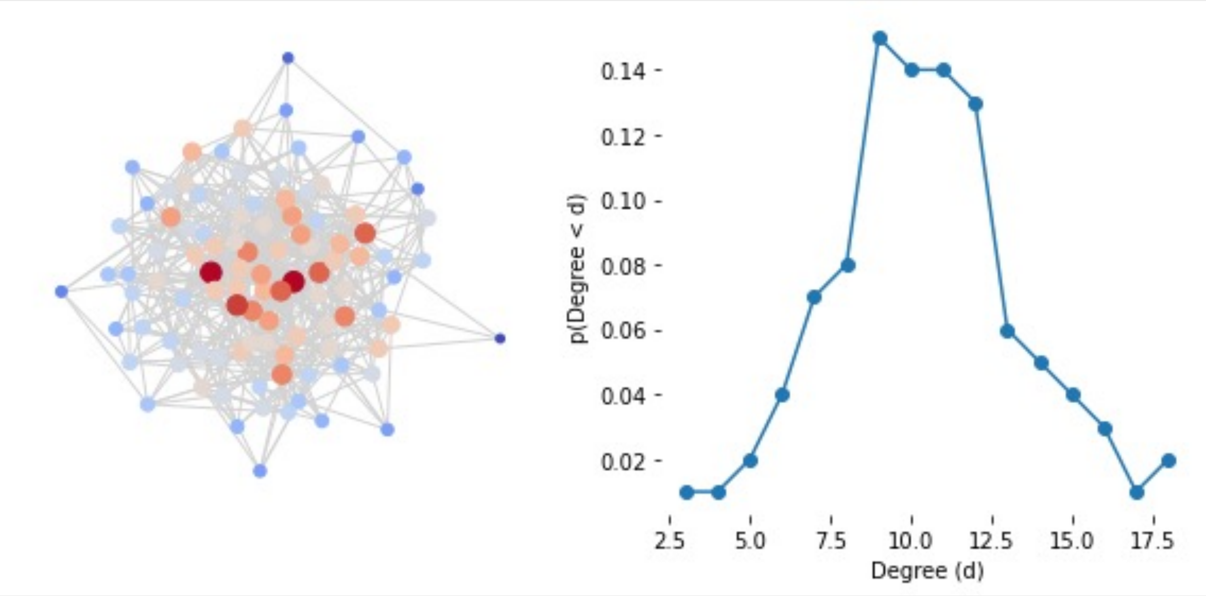
Most complex systems have **heavy tail distributions**

Most real networks have heavy tail degree distributions

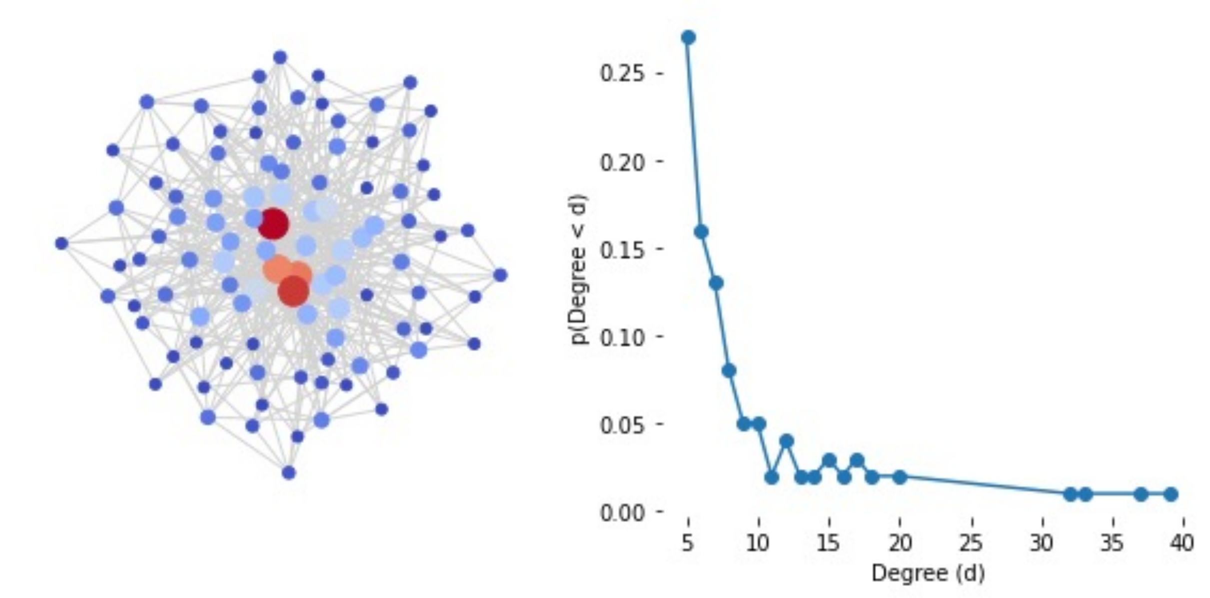


Clauset, Shalizi & Newman (2009)

Robustness to failures
Fragility to targeted attacks

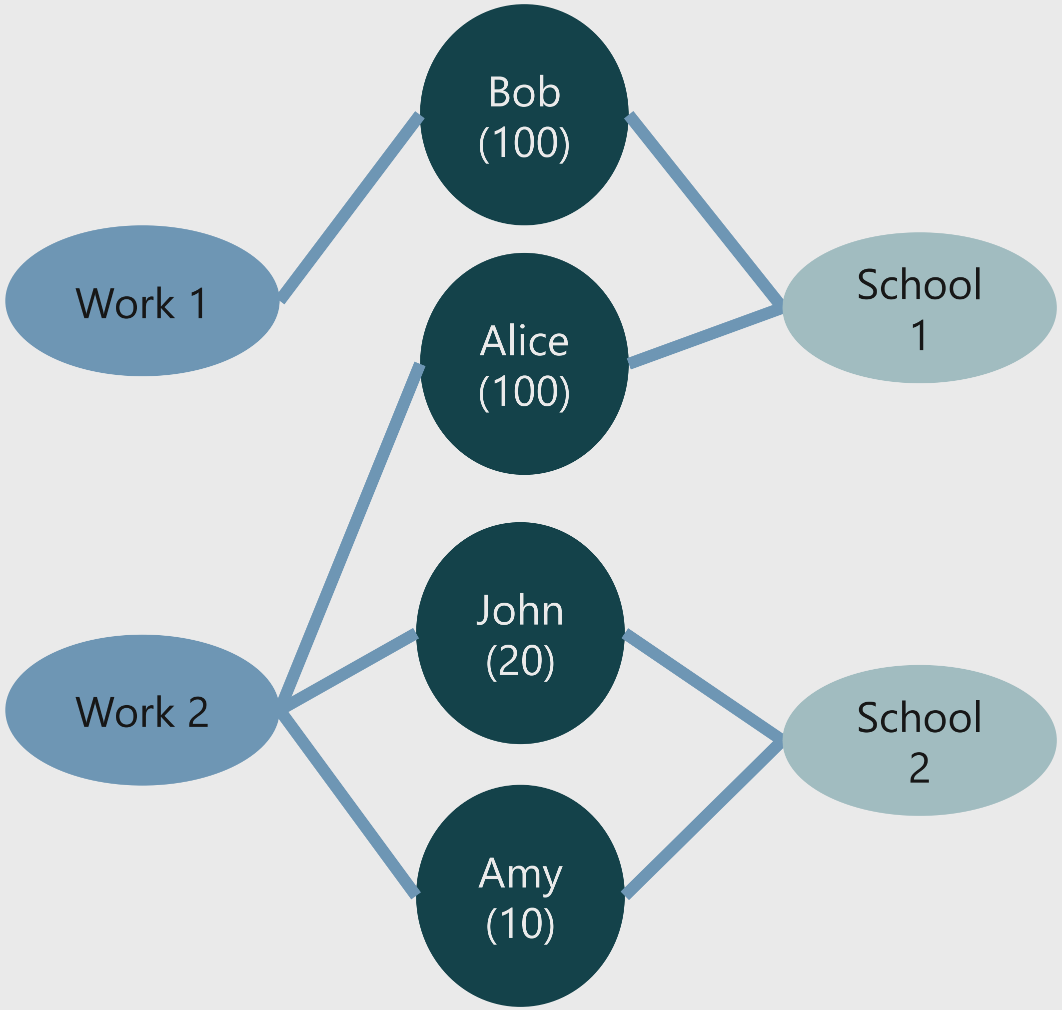


Random network



Heavy-tailed network

Affiliation networks at CBS



Multipartite network

Available networks:

- Family (up to couisins)
- School (program-level)
- Household
- Work (up to 100)
- Closest neighbors (up to 50)

How does the CBS newtorks look like?

→ Giant component → Most nodes are connected

→ Small world → Small diameter

→ Low density → Low average degree

→ Form cliques! (high clustering/transitivity)

→ Assortative (homophilic)

→ Heavy tail distributions

Types of analysis

They should fit your research question

Types of analysis: Descriptive statistics

Describe the network characteristics (density, diameter, average degree, clustering, etc)

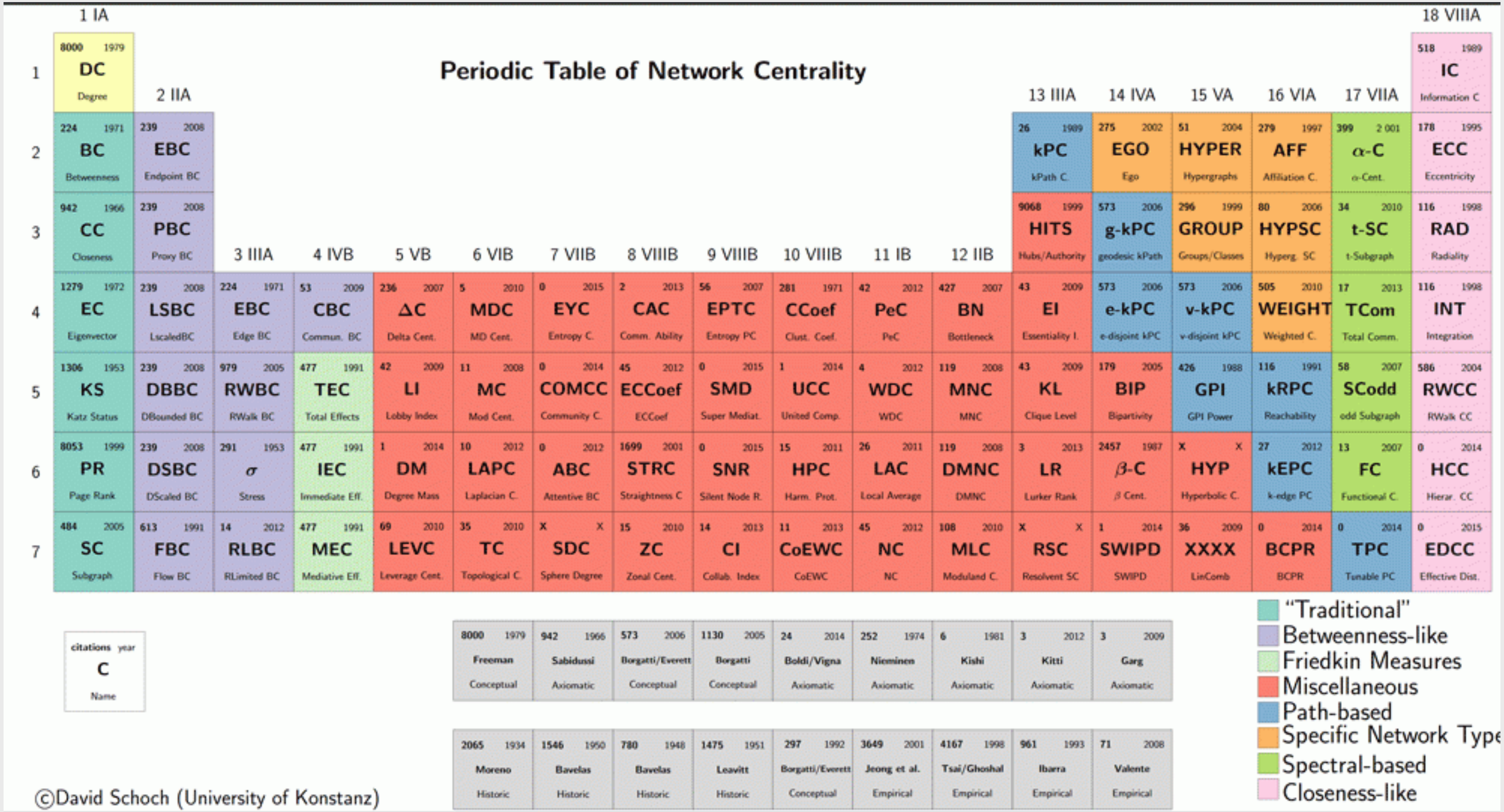
Types of analysis: Centralities

What are the **most important nodes** in the network?

- The one with **more connections** → Degree centrality
- The one **linked to more important neighbors** → Pagerank / Eigenvector / Katz centrality
- The one **closest** to all other nodes → Closeness centrality
- The ones that act as **brokerage**? → Betweenness centrality

Use a centrality measure that fits your theory, not the one that gives you the best results

Consider what is the objective (e.g. is it to enable low-income individuals to increase their social capital?) *(https://petterhol.me/2019/01/11/the-importance-of-being-earnest-about-node-importance/)*

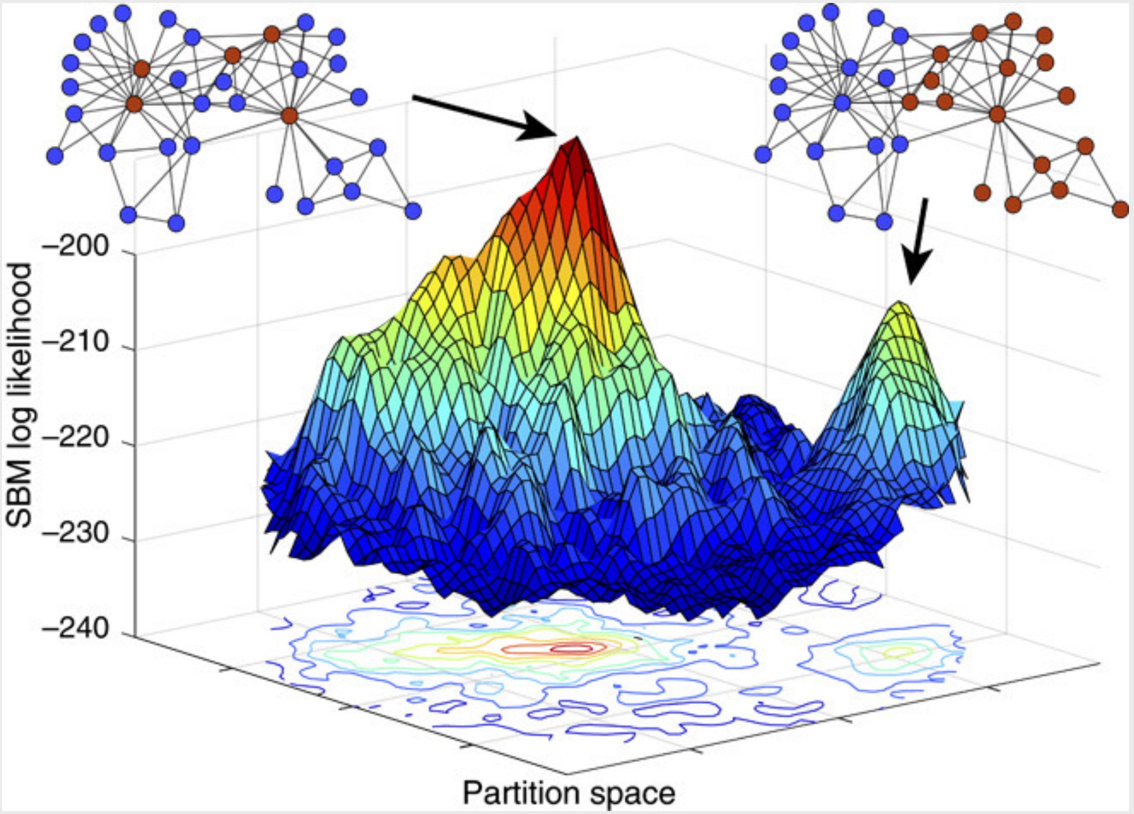


Types of analysis: Node-level regression

Calculate node-level features:

- Centrality
- Local clustering (transitivity / embeddedness)
- Local reciprocity
- Local assortativity (homophily)
- Ad-hoc variables: e.g. number of children within your immediate network
- …
- Include in your model (e.g. a regression)

Types of analysis: Community detection



What clusters of nodes can we find in the network?

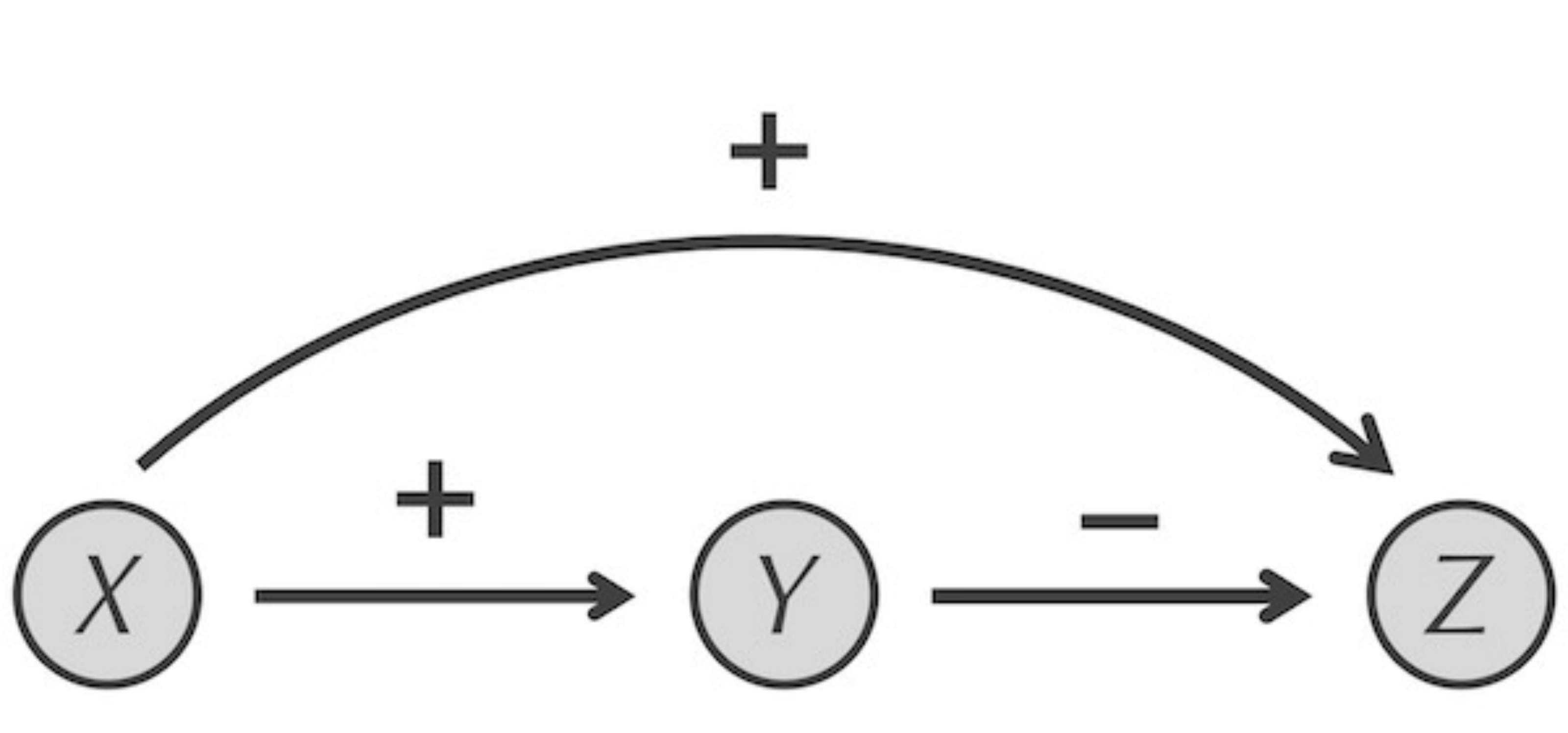
"It is standard practice to treat some observed discrete-valued node attributes, or metadata, as ground truth. **We show that metadata are not the same as ground truth and that treating them as such induces severe theoretical and practical problems**. We prove that no algorithm can uniquely solve community detection, and we prove a general No Free Lunch theorem for community detection, which implies that there can be no algorithm that is optimal for all possible community detection tasks" *(Peel, Larremore, Clauset, 2017)*

Stochastic Blockmodels (Harrison White, structural equivalence, core-periphery)

Modularity minimization

Types of analysis: Motif detection

Find overrepresented patterns

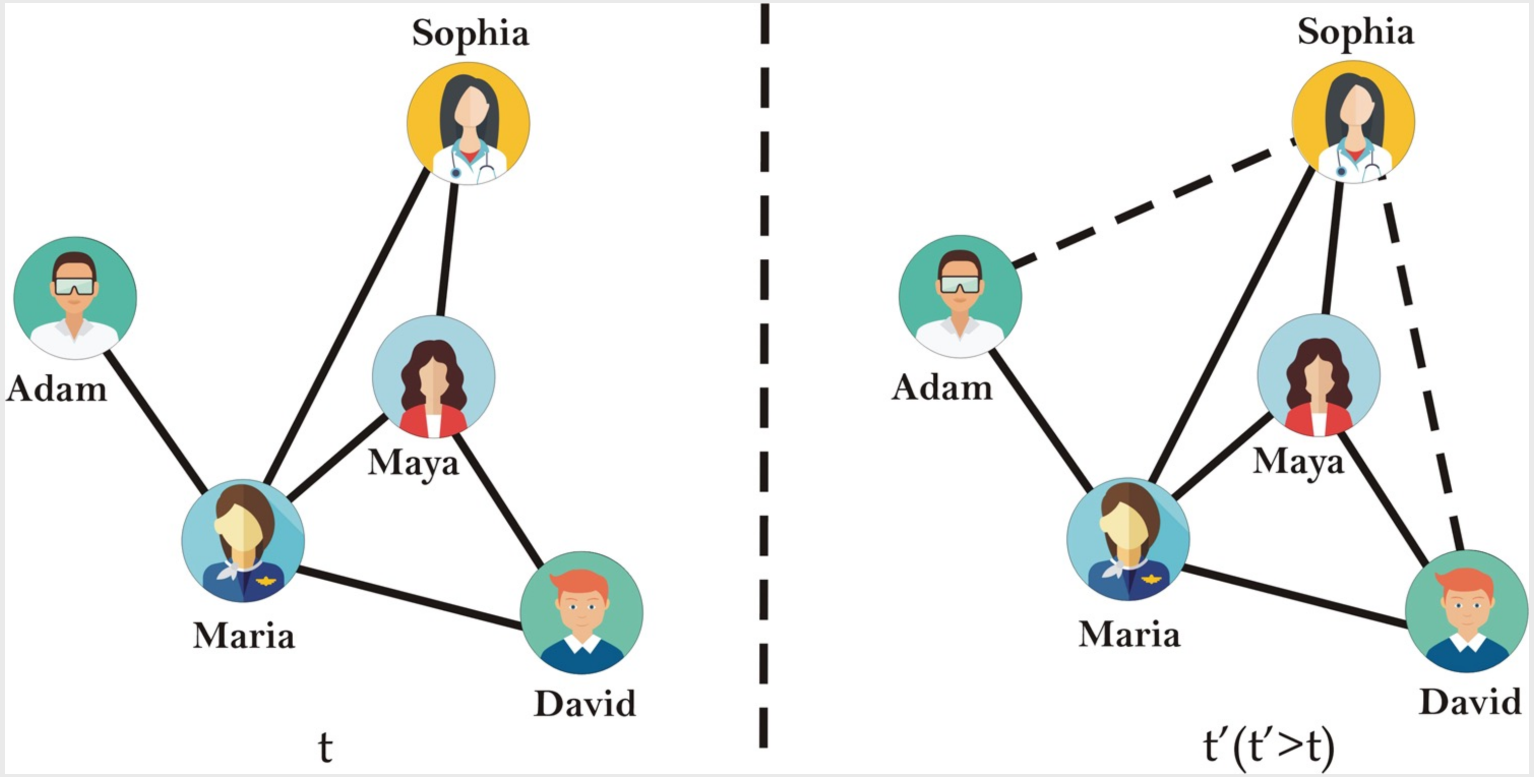


Feed-forward loop (https://biologicalmodeling.org/motifs/feedforward)

Types of analysis: Link/metadata prediction

Networks are rarely complete

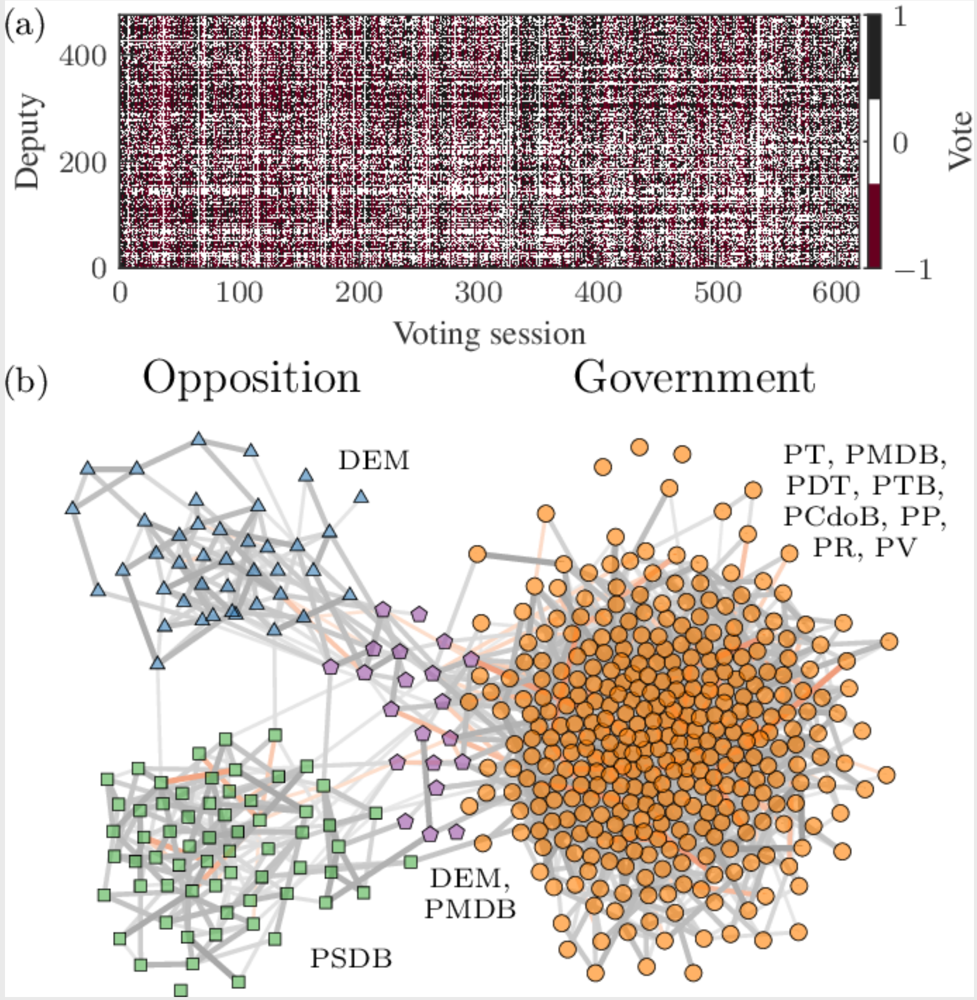
Link prediction approach: Approaches such as triangle closure



Ahmad et al 2020

Types of analysis: Network reconstruction

Network from co-occurrences



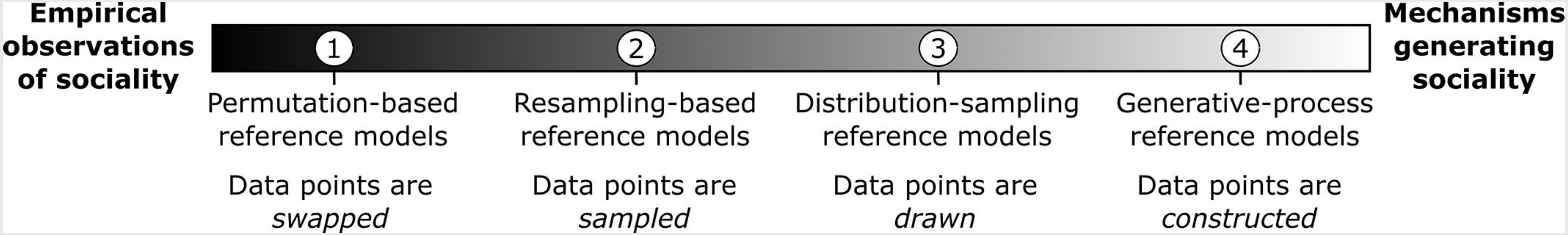
Network Reconstruction and Community Detection from Dynamics, Peixoto 2019

Types of analysis: Testing hypothesis

We observe some behavior in the network (e.g. the clustering is 0.5). Is this relevant?

Approach: Create a reference model (see *Hobson 2021* for a great guide) to compare with it

- Configuration model (permuting edges)
- Generative models (e.g. rich get richer model)
- ERGM (which features of dyads affect the presence or strength of edges.)
- ABM

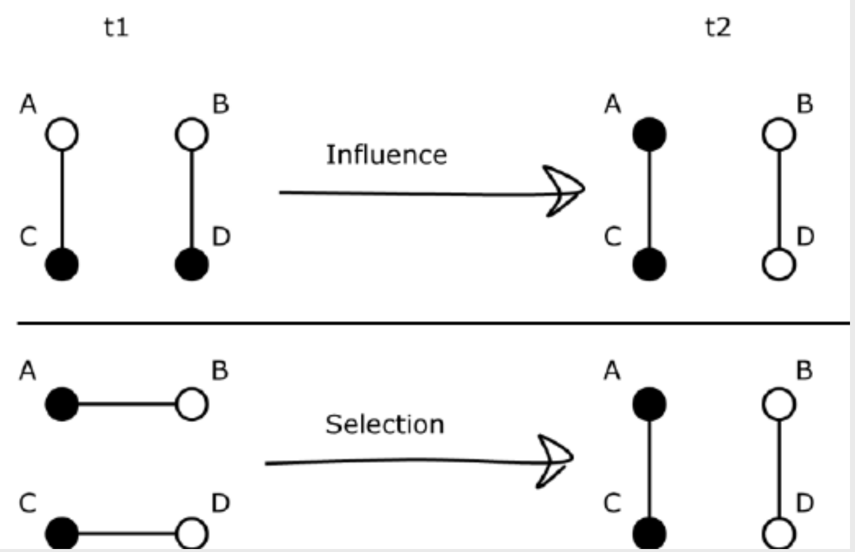


Hobson 2021

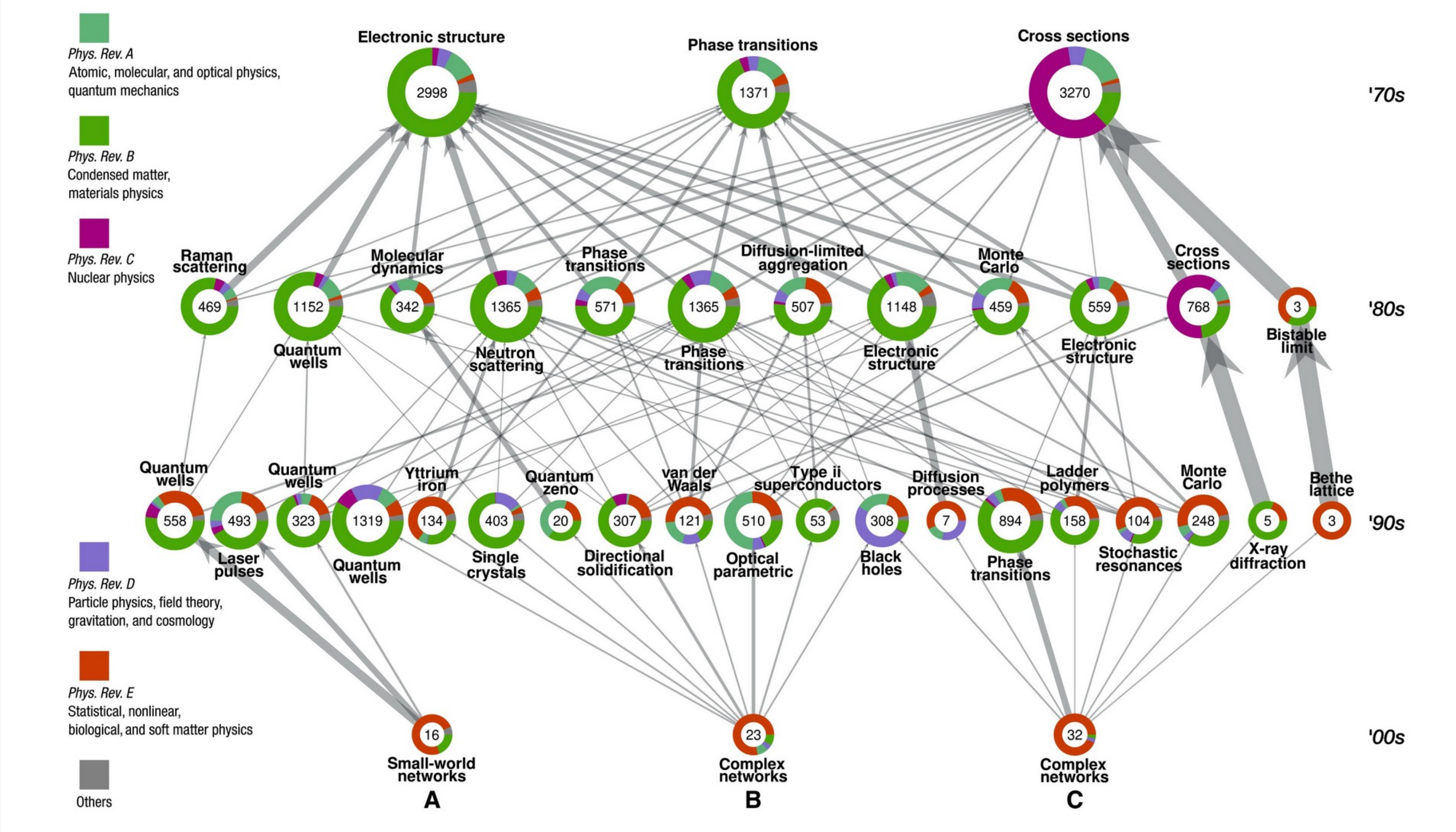
Types of analysis: Dynamics

How does behavior/diseases/information spread?

- Allow to test selection vs influence
- Run simulations on networks
  - Game theory
  - Epidemic spreading
  - Gene expression



Friemel, 2015



Bovet et al, 2022

Interested ?

Data Science: Network Science

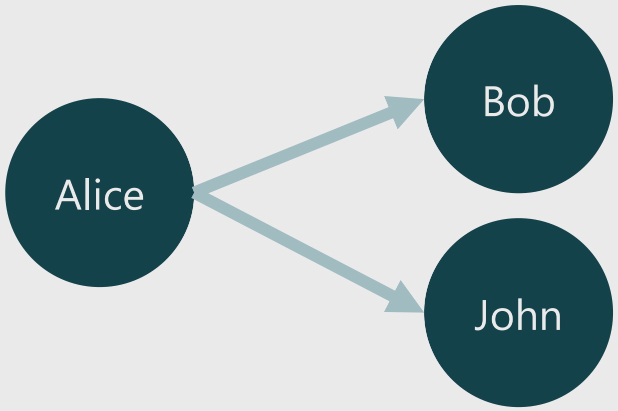
Organising institution Utrecht University - Faculty of Social and Behavioural Sciences	**Course code** S37
Period 10 July 2023-14 July 2023	**Course fee (excl. housing)** €720
Course location(s) Utrecht, The Netherlands	**Course Level** Master
ECTS credits 1.5	

Deadline: **26 June**

https://utrechtsummerschool.nl/courses/social-sciences/network-science

Matrix representation

Network representation



Adjacency list:

A. It is dense: Only keeping edges

Origin	Target	Weigth
Alice	Bob	1
Alice	John	1

Adjacency matrix:

- A. Linear algebra is easy
- Sparse: Many zeros → 1E6 nodes/10 million edges → 1 trillion options

Target → ↓ Source	Alice	Bob	John
Alice	0	1	1
Bob	0	0	0
John	0	0	0

In computer → Sparse matrices: Best of both worlds

Practical 1: Python

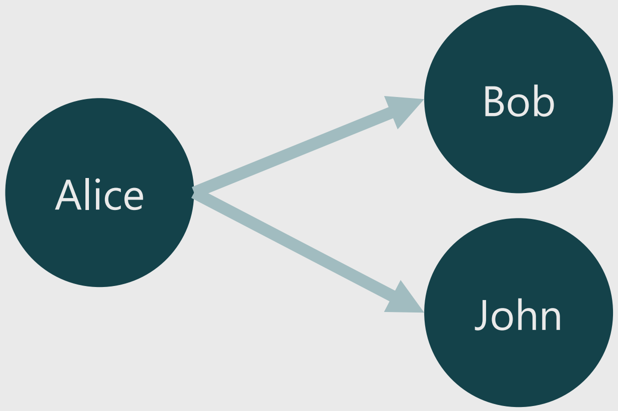
tinyurl.com/sicss-netsci

Underlying code/slides: https://github.com/jgarciab/3hours_network_science

Intro to linear algebra

Why? Multiplying matrices is fast (relatively)

Network representation



Adjacency list:

A. It is dense: Only keeping edges

Origin	Target	Weigth
Alice	Bob	1
Alice	John	1

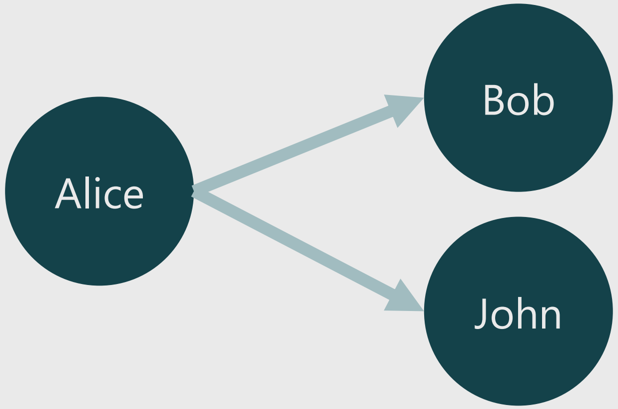
Adjacency matrix:

- A. Linear algebra is easy
- Sparse: Many zeros → 1E6 nodes/10 million edges → 1 trillion options

Target → ↓ Source	Alice	Bob	John
Alice	0	1	1
Bob	0	0	0
John	0	0	0

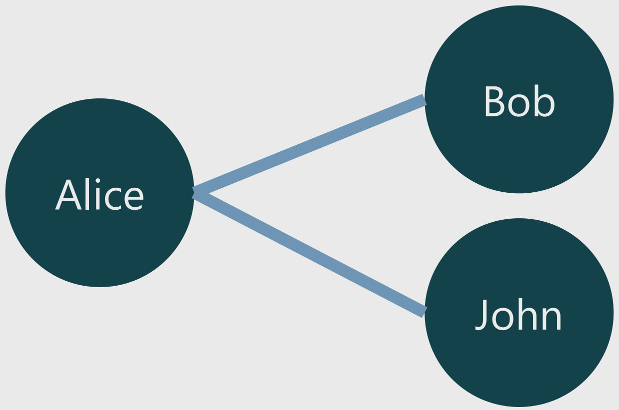
In computer → Sparse matrices: Best of both worlds

Directed networks



Target → ↓ Source	Alice	Bob	John
Alice	0	1	1
Bob	0	0	0
John	0	0	0

Undirected networks



Target → ↓ Source	Alice	Bob	John
Alice	0	1	1
Bob	1	0	0
John	1	0	0

Some terms

A =

Target → ↓ Source	Alice	Bob	John
Alice	0	1	1
Bob	0	0	0
John	0	0	0

Diagonal

Trace = Sum of elements in the diagonal

Identity matrix (I) =

	1	0	0
	0	1	0
	0	0	1

I @ A = A

Transpose (A^T) =

Target → ↓ Source	Alice	Bob	John
Alice	0	0	0
Bob	1	0	0
John	1	0	0

(python) A.T

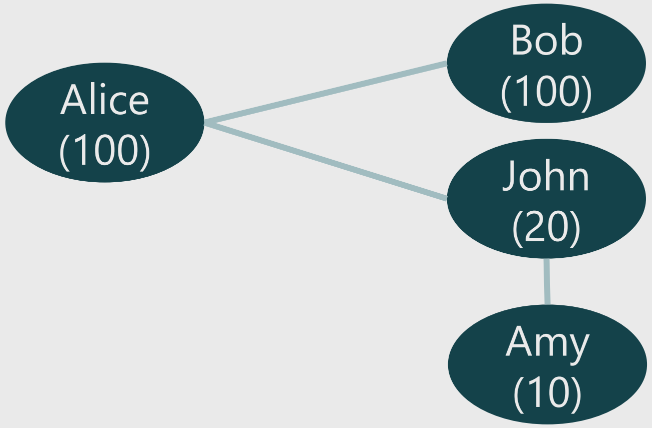
Symmetric matrix: A = A.T (e.g. undirected network)

Python exercise notebook 2, ex.1

Python:

- Convert between formats
- Plot matrix

Matrix multiplication: sum



Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	1	1	0
Bob	1	0	0	0
John	1	0	0	1
Amy	0	0	1	0

@

Node	kids
Alice	3
Bob	2
John	1
Amy	0

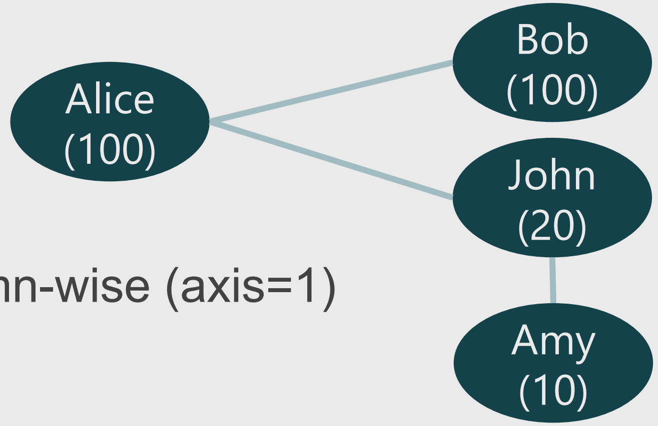
=

Node	kids
Alice	0*3 + 1*2 + 1*1 + 0*0 = 3
Bob	1*3 + 0*2 + 0*1 + 0*0 = 3
John	1*3 + 0*2 + 0*1 + 1*0 = 3
Amy	0*3 + 0*2 + 1*1 + 0*0 = 1

A @ M = SM

(N x N) @ (N x 1) = (N x 1)

Matrix multiplication: average



Divide by the degree. We get it by summing the adjacency elements column-wise (axis=1)

A @ M / A.sum(1) = AvM

(N x N) @ (N x 1) / (N x 1) = (N x 1) / (N x 1) = (N x 1)

Target → ↓ Origin	Alice	Bob	John	Amy
Alice	0	1	1	0
Bob	1	0	0	0
John	1	0	0	1
Amy	0	0	1	0

@

Node	Kids
Alice	3
Bob	2
John	1
Amy	0

/

Target → ↓ Source	Sum
Alice	2
Bob	1
John	2
Amy	1

=

Node	Kids
Alice	3
Bob	3
John	3
Amy	1

/

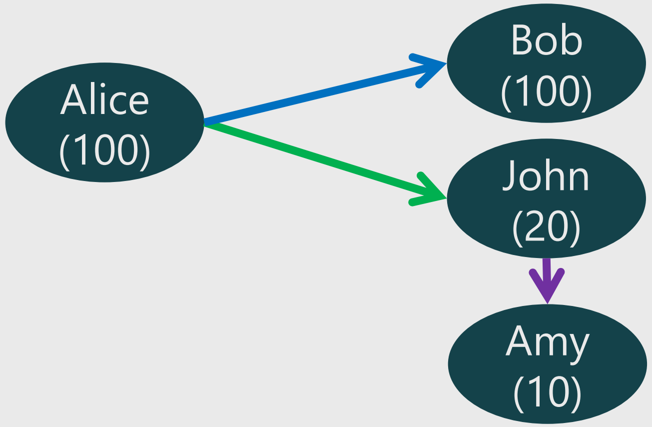
Target → ↓ Source	Sum
Alice	2
Bob	1
John	2
Amy	1

=

Node	Kids
Alice	1.5
Bob	3
John	1.5
Amy	1

Python exercise notebook 2, ex.2

Matrix multiplication: paths



Interpretation A: Presence of path between node i and j
Interpretation A^2: Number of path between node i and j in two steps
Interpretation A^3: Number of path between node i and j in three steps
…

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	1	1	0
Bob	0	0	0	0
John	0	0	0	1
Amy	0	0	0	0

@

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	1	1	0
Bob	0	0	0	0
John	0	0	0	1
Amy	0	0	0	0

=

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	0	0	1
Bob	0	0	0	0
John	0	0	0	0
Amy	0	0	0	0

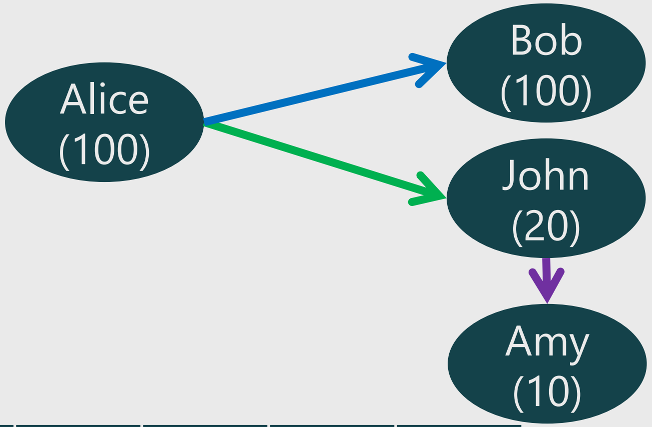
Alice → Alice (0) * Alice → Amy (0)
+ Alice → Bob (1) * Bob → Amy (0)
+ Alice → John (1) * John → Amy (1)
+ Alice → Alice (0) * Alice → Amy (1)

Python exercise notebook 2, ex.3a

Matrix multiplication: number of people reached in <3 steps

Number of paths in two or three steps from node i to node j: ***N = A + A^2 + A^3***
We need to remove duplicate paths: ***N = N>0***
We need to remove paths from us to ourselves ***N.setdiag(0)***

Matrix multiplication: number of triangles



A^2

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	0	0	1
Bob	0	0	0	0
John	0	0	0	0
Amy	0	0	0	0

@

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	1	1	0
Bob	0	0	0	0
John	0	0	0	1
Amy	0	0	0	0

=

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	0	0	0
Bob	0	0	0	0
John	0	0	0	0
Amy	0	0	0	0

Alice → Alice in two steps * Alice → Alice (0)
Alice → Bob in two steps * Bob -> Alice (0)
Alice → John in two steps * John -> Alice (0)
Alice → Amy in two steps * Amy -> Alice (0)

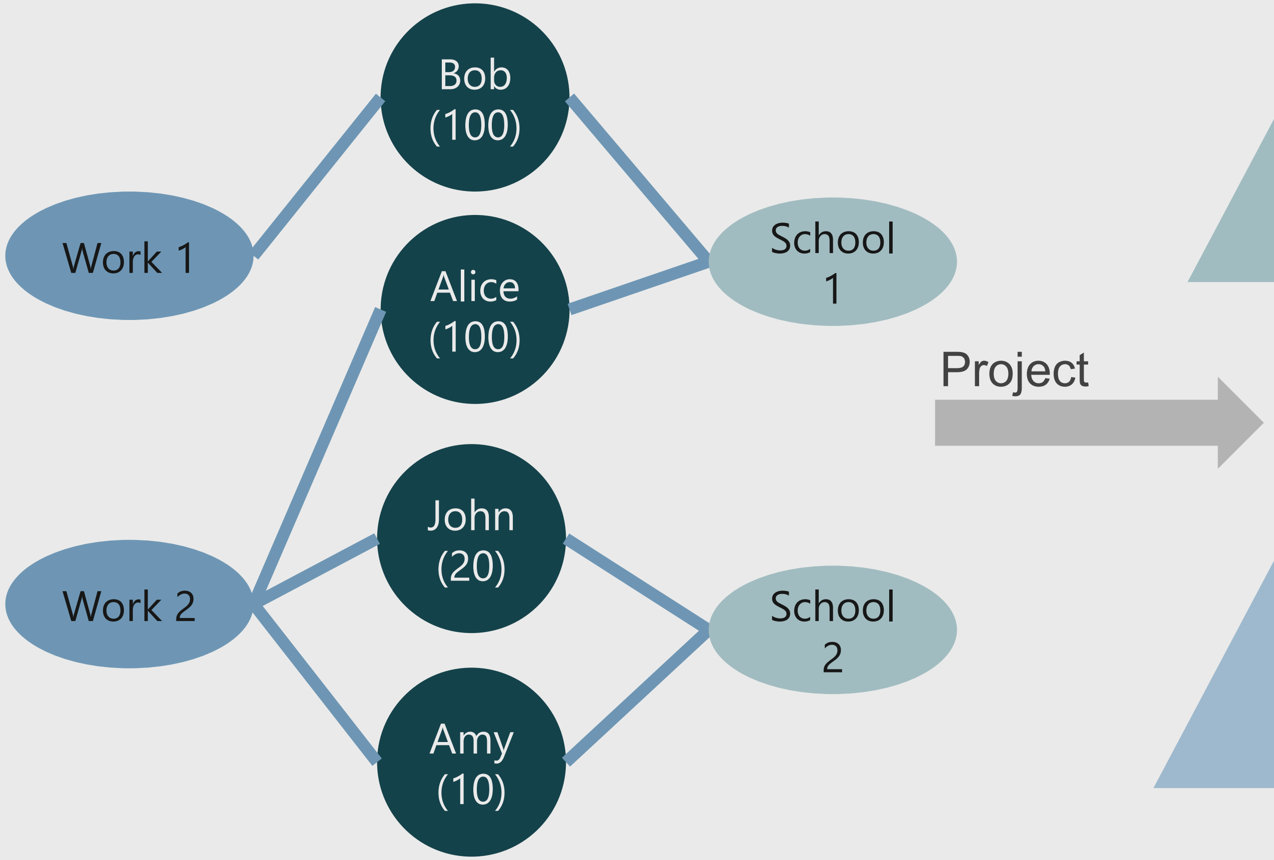
Diagonal of A^3

$\text{Alice} \rightarrow X_1 \; * \; X_1 \rightarrow X_1 \; * \; X_1 \rightarrow \text{Alice} \; +$
$\text{Alice} \rightarrow X_1 \; * \; X_1 \rightarrow X_2 \; * \; X_2 \rightarrow \text{Alice} \; +$
...

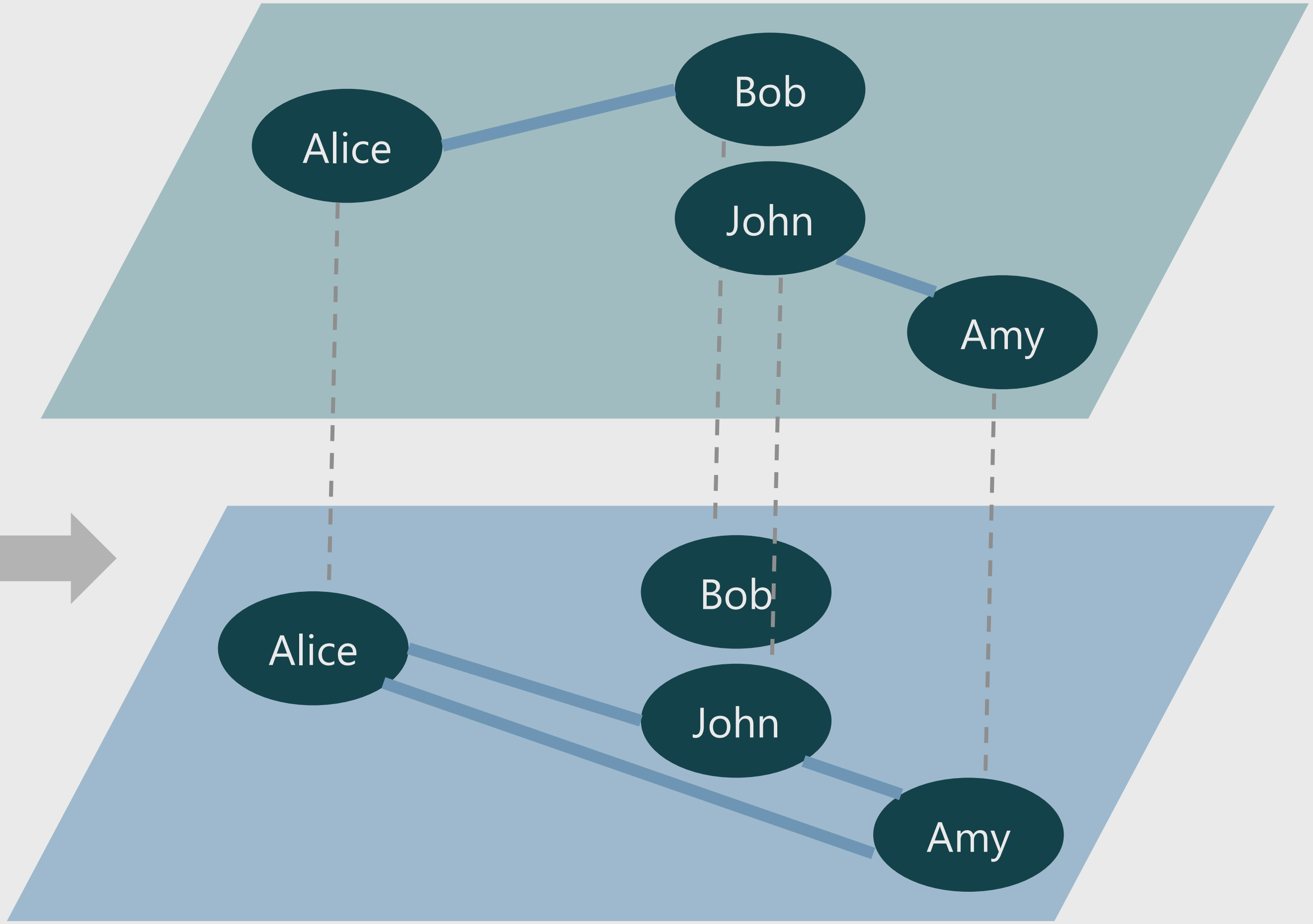
Python exercise notebook 2, ex.3b

(already done, just check solutions)

CBS data

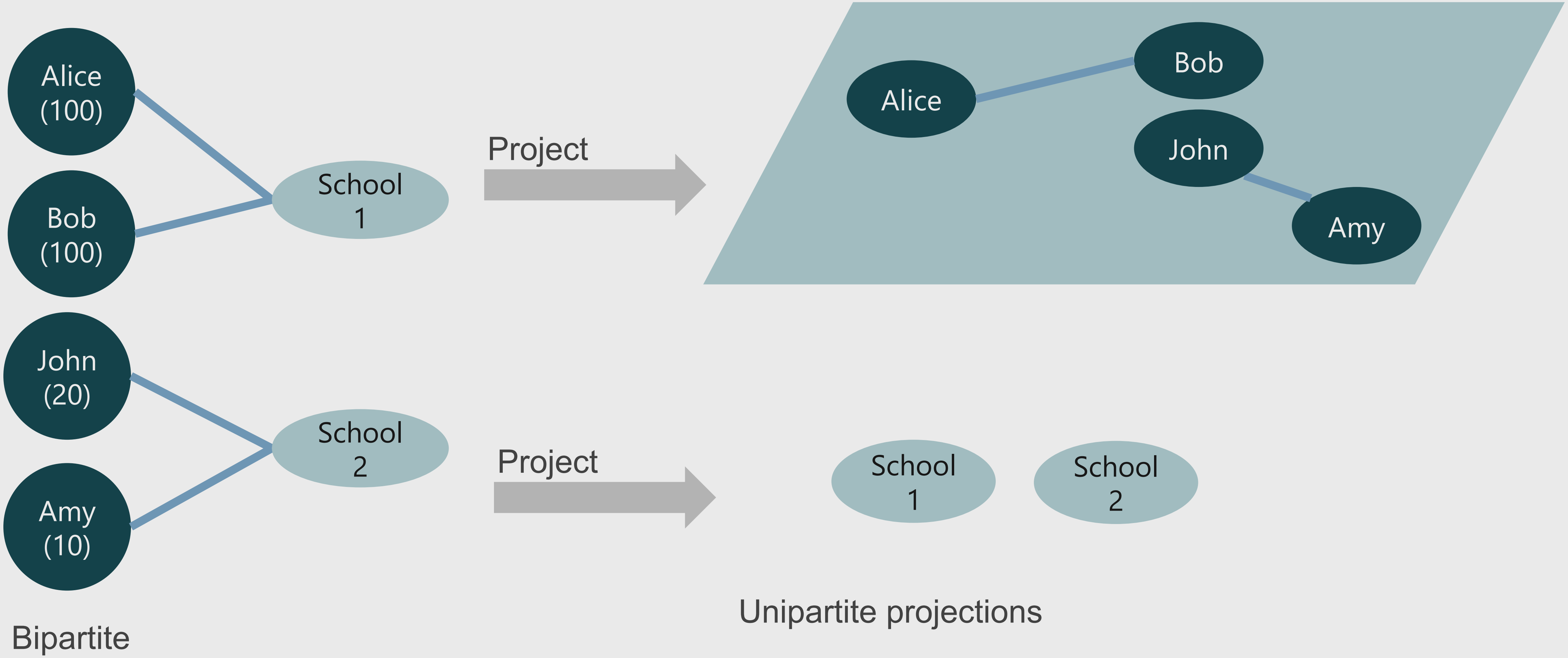


Multipartite network (CBS raw data)

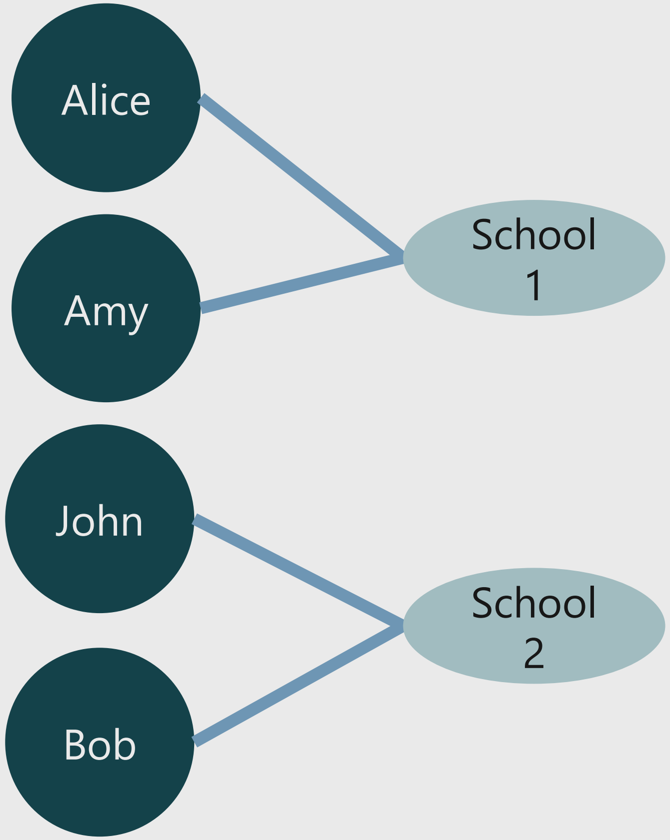


Multiplex projection (CBS network data)

Other types of networks: Bipartite



Matrix multiplication: projection



Target → ↓ Source	S1	S2
Alice	0	1
Bob	1	0
John	1	0
Amy	0	1

@

Target → ↓ Source	Alice	Bob	John	Amy
S1	0	1	1	0
S2	1	0	0	1

=

Target → ↓ Source	Alice	Bob	John	Amy
Alice	1	0	0	1
Bob	0	1	1	0
John	0	1	1	0
Amy	1	0	0	1

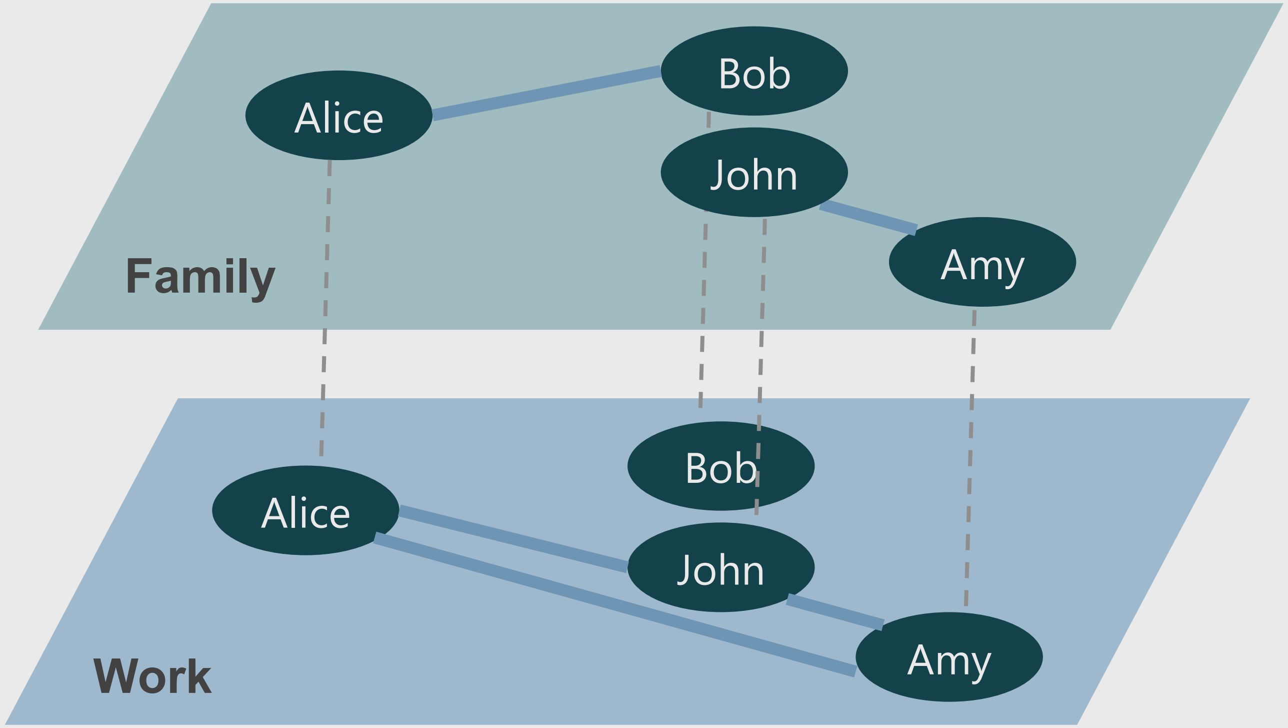
Alice → S1 (0) * S1 → Amy (0)

\+ Alice → S2 (1) * S2 → Amy (1)

Python exercise notebook 2, ex.4

(already done, just check solutions)

How many children does my family have?



Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	1	0	0
Bob	1	0	0	0
John	0	0	0	1
Amy	0	0	1	0

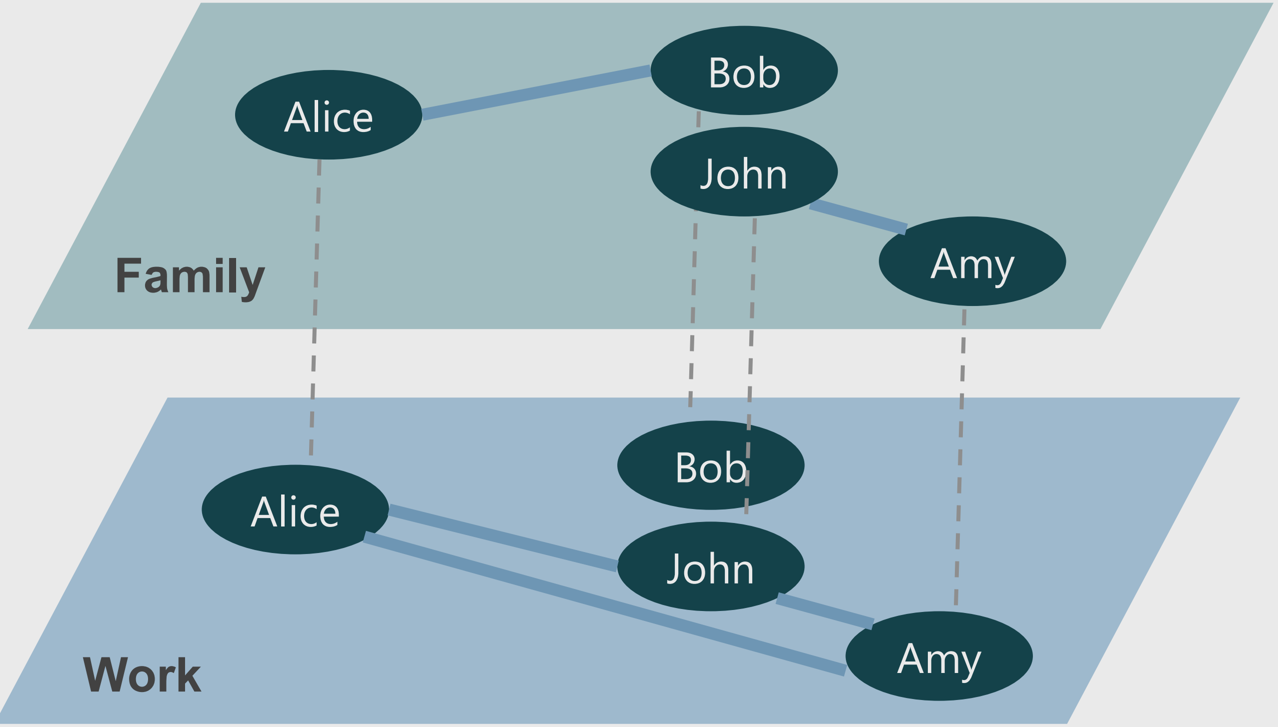
Family

@

Node	kids
Alice	3
Bob	3
John	3
Amy	1

=

Node	kids
Alice	3
Bob	3
John	1
Amy	3



How many children do the families of my colleagues have?

Node	kids
Alice	3
Bob	3
John	3
Amy	1

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	0	0	1
Bob	0	0	0	0
John	0	0	0	0
Amy	1	0	0	0

Work

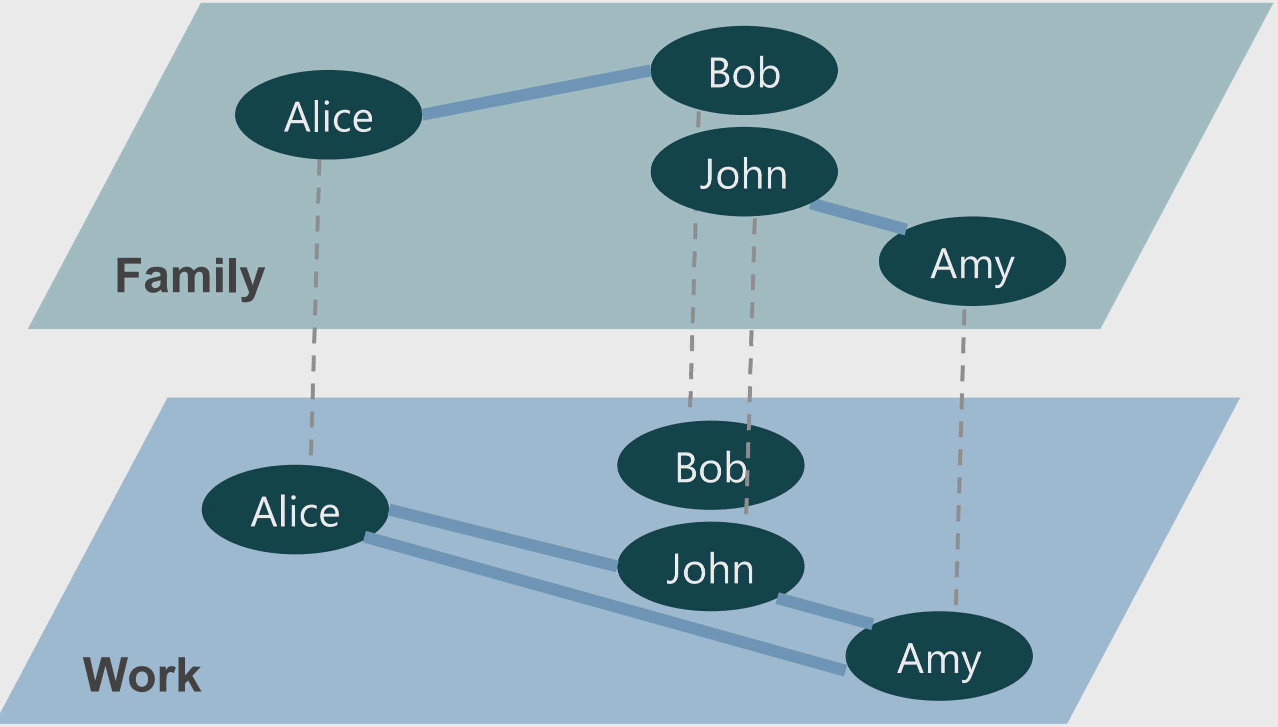
@

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	1	0	0
Bob	1	0	0	0
John	0	0	0	1
Amy	0	0	1	0

Family

=

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	0	1	0
Bob	0	0	0	0
John	0	0	0	0
Amy	0	1	0	0



How many children do the families of my colleagues have?

Target → ↓ Source	Alice	Bob	John	Amy
Alice	0	0	1	0
Bob	0	0	0	0
John	0	0	0	0
Amy	0	1	0	0

@

Node	kids
Alice	3
Bob	3
John	3
Amy	1

=

Node	kids
Alice	3
Bob	0
John	0
Amy	3

Python exercise notebook 2, ex.5a and 5b

Practical 3:
Working with networks using Gephi

Exercise 1: Gephi

Follow this tutorial (slides 1–23 only!): https://gephi.org/users/quick-start/

- In community detection use the “stochastic blockmodel” instead of modularity maximization

You can choose to use our own data: https://tinyurl.com/network-game