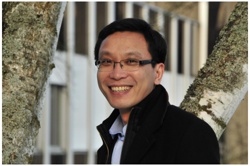
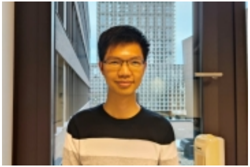
Modelling and Analysis of Complex Networks

— Semester 3, Master of Data Science —

Jun Pang
University of Luxembourg

Who

- Jun Pang (lectures), MNO E03-35-010,
- George Panagopoulos (lectures), MNO E03-25-070,
- Tsz Pan Tong (exercises), MNO E03-25-100
- Email: `firstname.lastname@uni.lu`

Course Topics

- Representation and metrics of (complex) networks

  (degree distribution, community detection, etc.)

- Network models

  (random graphs, small-world, preferential attachment, etc.)

- Processes on networks

  (epidemic models of diffusion, etc.)

- Machine learning with graph data
  (feature-based methods, node embeddings, GNNs)

As planned today – may go through unpredictable changes.

Course Prerequisites

- Introduction to Graph Theory (MADS semester 1)
- Programming with R and Python (MADS semester 1)
- Data Visualization (MADS semester 1)
- Introduction to Machine Learning Methods and Data Mining (MADS semester 2)
- Prototyping with Deep Learning (MADS semester 2)

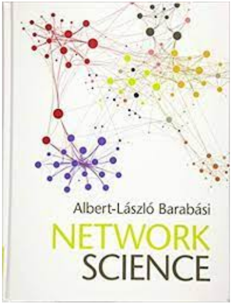
Organisation

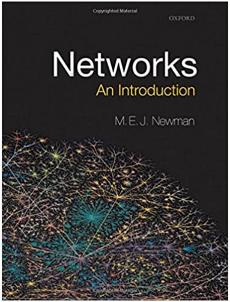
- Lectures and in-class exercises (Fridays 13:00-16:00, active & ask questions!)
- Evaluation: class participation (4 points) and course projects (16 points)
- No written exam!
- Moodle system: `https://moodle.uni.lu`
  (course materials, schedule, exercises, etc.)

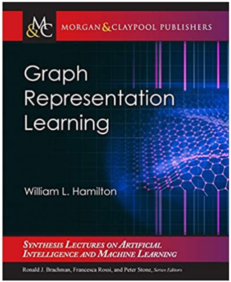
Tentative Schedule (Fridays)

- Introduction & motivations (20/09, MNO 1.050)
- Network measures (27/09, MNO 1.050)
- Network models (04/10, MNO 1.050)
- Centrality measures (11/10, MNO 1.050)
- Motifs, graphlets, and communities (18/10, MNO 1.050)
- Graph partitioning and spectral clustering (25/10, MNO 1.050)
- Course project reviews (08/11, MNO 1.050)
- Network cascade models (15/11, MNO 1.050)
- Epidemic models and influence maximisation (22/11, MNO 1.050)
- Machine learning on graphs (29/11, MNO 1.050)
- Node embedding (06/12, MNO 1.050)
- Deep learning on graphs (13/12, MNO 1.050)
- Course project reviews (20/12, MNO 1.050)

Materials







Youtube Videos: CS224W: Machine learning with graphs

Slides are built on course materials by ©Jure Leskovec, ©Argimiro Arratia, ©Ramon Ferrer-i-Cancho, ©Cazabet Rémy. (Mistakes are all mine!)

Software and Tools

- Network analysis tools:

  SNAP.PY, SNAP C++ (https://snap.stanford.edu/snappy/),

  NetworkX, iGraph

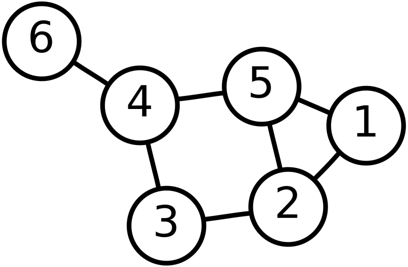
- Graph machine learning tools:

  PyTorch Geometric (PyG) (https://pytorch-geometric.readthedocs.io/),

  DeepSNAP (https://snap.stanford.edu/deepsnap/)

Graphs (Networks): A Universal Language

Graph (discrete mathematics)

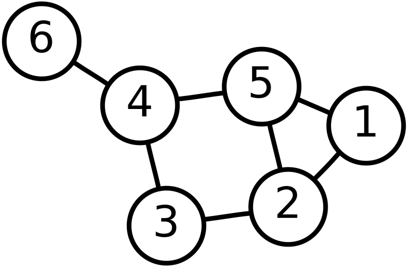


©Wikipedia

Undirected, directed, mixed, weighted, ...

Graphs are a general language for describing and modelling complex dynamical systems.

Graph (discrete mathematics)



Undirected, directed, mixed, weighted, ...

Graphs are a general language for describing and modelling complex dynamical systems.

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

What do these systems have in common? How to represent them?

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

What do these systems have in common? How to represent them?

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

What do these systems have in common? How to represent them?

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

What do these systems have in common? How to represent them?

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

What do these systems have in common? How to represent them?

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

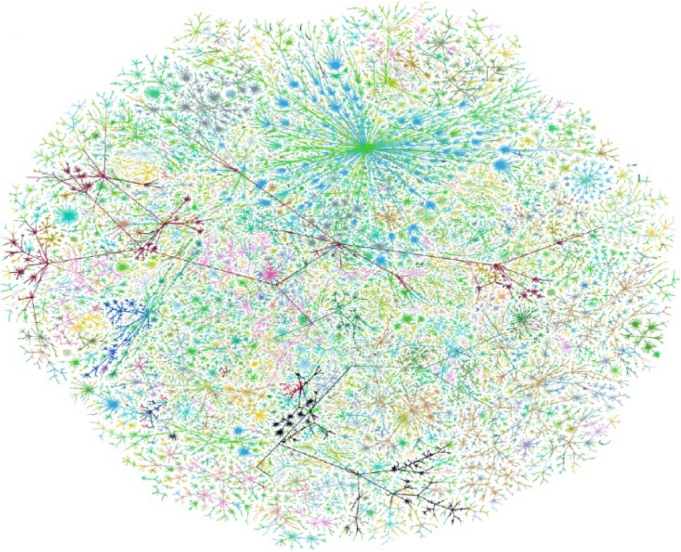
What do these systems have in common? How to represent them?

Complex Systems

There are complex systems all around us:

- Society is a collection of 7+ billion individuals.
- Communication systems link electronic devices.
- Information and knowledge are organised and linked.
- Interactions between thousands of genes/proteins regulate life.
- Our thoughts are hidden in the connections between billions of neurons in our brain.

What do these systems have in common? How to represent them?

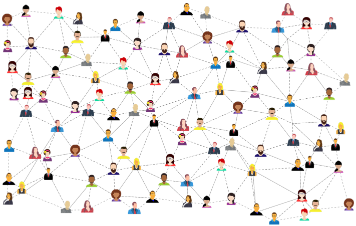


The Internet

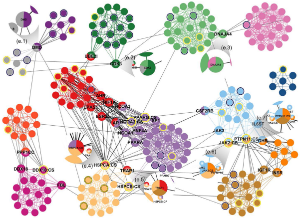
(Complex) Networks

- Behind many systems there is an intricate wiring diagram, a network, defining the interactions between the components.
- We will never be able to model, predict, and control these systems unless we understand the networks behind them.

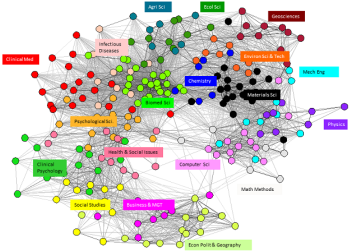
Data as Graphs (Explicit)



Social networks



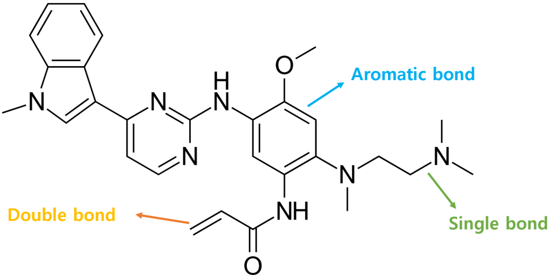
PPI networks



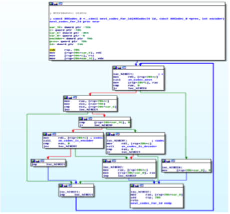
Citation networks

Transportation networks

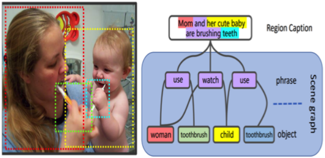
Data as Graphs (Implicit)



Molecular graphs



Program graphs



Scene graphs

(Many) Data as Graphs

How do we take advantage of relational structure in the data for better modelling, analysis, and control?

Why Networks? Why Now?

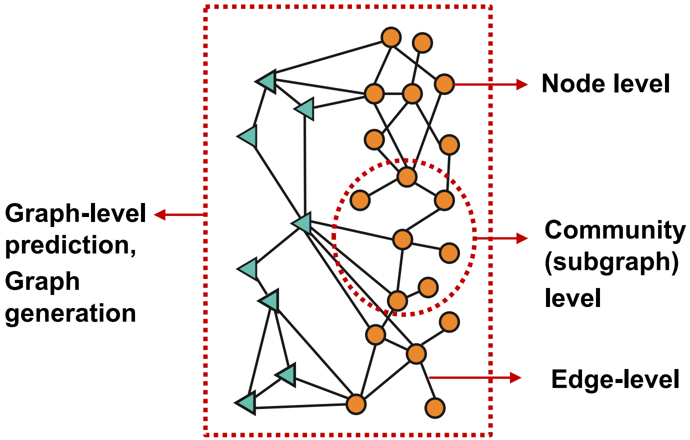
- Universal language for describing complex data
  Networks/graphs from science, nature, and technology are more similar than one would expect
- Shared vocabulary between many fields
  Computer Science, Social Sciences, Physics, Biology, Economics
- Data availability
  Social/internet, text, logic, program, bio, health, and medical
- Impact in many areas
  Social media, drug design, event detection, natural language processing, computer vision, and logic/symbolic reasoning

Why Networks? Why Now?

- Universal language for describing complex data
  Networks/graphs from science, nature, and technology are more similar than one would expect
- Shared vocabulary between many fields
  Computer Science, Social Sciences, Physics, Biology, Economics
- Data availability
  Social/internet, text, logic, program, bio, health, and medical
- Impact in many areas
  Social media, drug design, event detection, natural language processing, computer vision, and logic/symbolic reasoning

Analysis and Applications

Network Analysis



Network Analysis

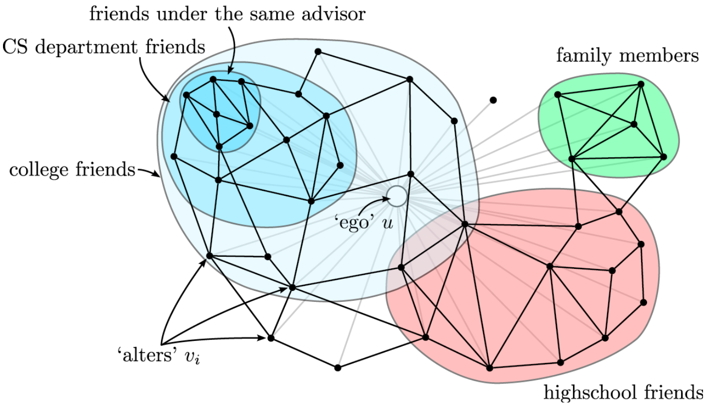
Classical tasks

- Node classification
- Link prediction
- Community detection
- Graph similarity

Recent tasks

- Graph classification
- Graph generation
- Graph structure learning
- ...

Applications



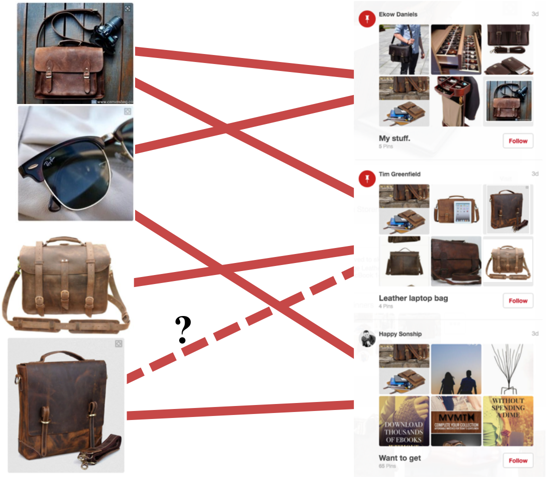
Discover circles and why they exist

Applications

Misinformation detection

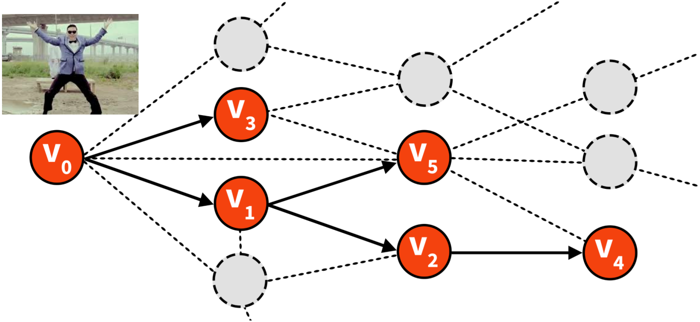
(random 50%, human 66%, network 86%)

Applications



Link prediction: content recommendation

Applications



Information cascade in social networks

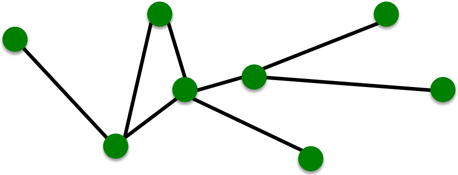
Applications

- Drug & antibiotic discovery
- Molecule generation and optimisation
- Physics simulation
- ... more and more exciting applications!

Graph Representations

Networks = Graphs

Networks are just collections of "points" joined by "lines".



- Points (N): vertices (math), nodes (CS), sites (physics), actors (sociology)
- Lines (E): edges, arcs (math), links (CS), bonds (physics), ties, relations (sociology)
- $G(N, E)$: graphs (math), networks (CS)

We will use the terms (vertices vs. nodes, edges vs. links) interchangeably.

How to Build a Graph?

- If you connect individuals that work with each other, you will explore a professional network.
- If you connect scientific papers that cite each other, you will be studying the citation network.

Choice of the proper network representation of a given problem determines our ability to use networks successfully.

- In some cases there is a unique, unambiguous representation.
- In other cases, the representation is by no means unique.
- The way you assign links will determine the nature of the question you can study.

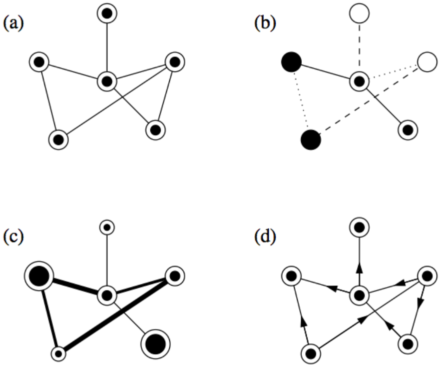
How to Build a Graph?

- If you connect individuals that work with each other, you will explore a professional network.
- If you connect scientific papers that cite each other, you will be studying the citation network.

Choice of the proper network representation of a given problem determines our ability to use networks successfully.

- In some cases there is a unique, unambiguous representation.
- In other cases, the representation is by no means unique.
- The way you assign links will determine the nature of the question you can study.

Types of Networks



(a) Unweighted, undirected

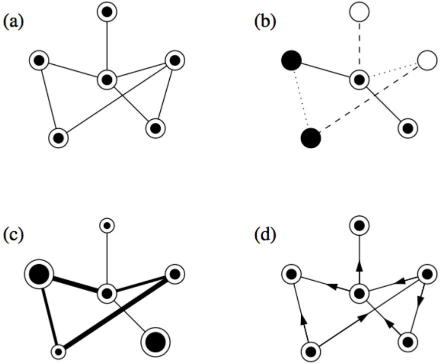
(b) Discrete vertex/edge types, undirected

(c) Varying vertex/edge weights, undirected

(d) Directed

Can you give real-world networks of different types?

Types of Networks



(a) Unweighted, undirected

(b) Discrete vertex/edge types, undirected

(c) Varying vertex/edge weights, undirected

(d) Directed

Can you give real-world networks of different types?

Counting Nodes and Edges

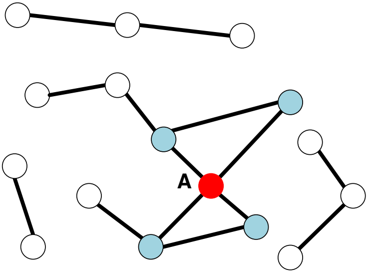
- Number of nodes: $|N|$ (N, n)
- Number of edges: $|E|$ (E, m)
- Maximum number of edges: $\frac{N(N-1)}{2}$ (undirected networks), $N(N-1)$ (directed networks)

Note the abuse of notations N and E.

Counting Nodes and Edges

	#nodes (n)	#edges (m)
Wikipedia HL	2M	30M
Twitter 2015	288M	60B
Facebook 2015	1.4B	400B
Brain c. Elegans	280	6393
Roads US	2M	2.7M
Airport traffic	3k	31k

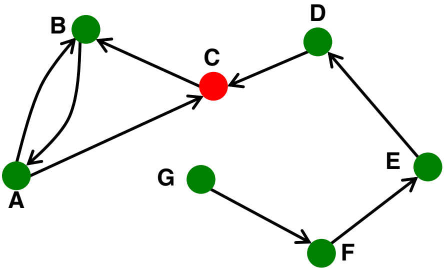
Node Degrees



- Node degree (k_i): the number of edges adjacent to node i ($k_A = 4$)
- Average degree:

$$\overline{k} = \langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i = \frac{2E}{N}$$

Node Degrees



For directed networks, we define an in-degree and out-degree. The total degree of a node is the sum of in- and out-degrees ($k_C^{in} = 2, k_C^{out} = 1, k_C = 3$).

$$\overline{k} = \frac{E}{N}, \ \overline{k^{in}} = \overline{k^{out}}$$

sink nodes ($k^{out} = 0$), source nodes ($k^{in} = 0$)

Density

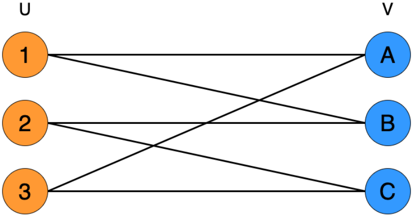
Density is defined as a fraction of pairs of nodes connected by an edge in a graph.

	#nodes	#edges	Density	avg. deg
Wikipedia	2M	30M	1.5×10^{-5}	30
Twitter 2015	288M	60B	1.4×10^{-6}	416
Facebook	1.4B	400B	4×10^{-9}	570
Brain c.	280	6393	0,16	46
Roads Calif.	2M	2.7M	6×10^{-7}	2,7
Airport	3k	31k	0,007	21

Density is hard to compare between graphs of different sizes. When graphs increase, the average degree increase (slowly), and the density decreases.

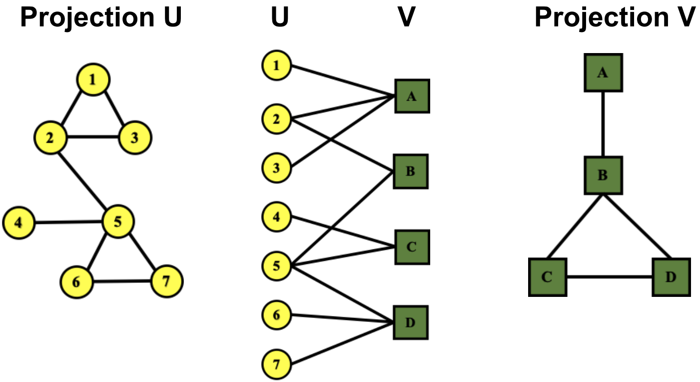
Bipartite Graph

Bipartite graph is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are independent sets.

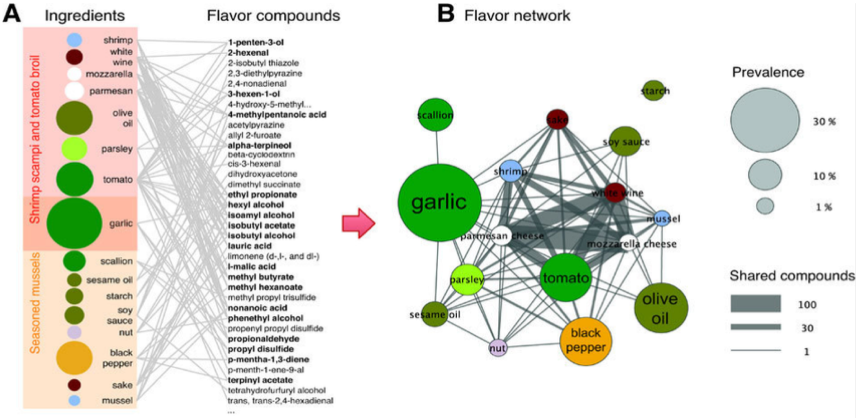


- Examples: authors-to-papers, actors-to-movies, users-to-movies, recipes-to-ingredients.
- Folded networks: author collaboration networks, movie co-rating networks.

Bipartite Graph

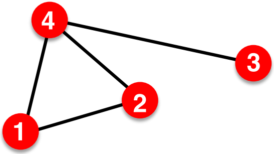


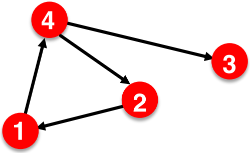
Bipartite Graph



Ahn et al. Flavor network and the principle of food pairing. Scientific Report, 2011.

Representing Graphs: Adjacency Matrix



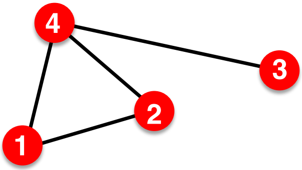


$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$A_{ij} = 1$ if there is a link/edge from node i to node j; $A_{ij} = 0$, otherwise. (For a directed graph, its matrix is not symmetric.)

Representing Graphs: Adjacency Matrix



$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$$k_i = \sum_{j=1}^{N} A_{ij}$$

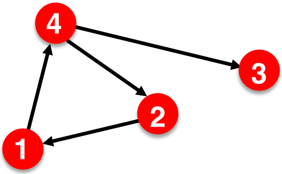
$$k_j = \sum_{i=1}^{N} A_{ij}$$

$$A_{ij} = A_{ji}$$

$$A_{ii} = 0$$

$$L = \frac{1}{2}\sum_{i=1}^{N} k_i = \frac{1}{2}\sum_{ij}^{N} A_{ij}$$

Representing Graphs: Adjacency Matrix



$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

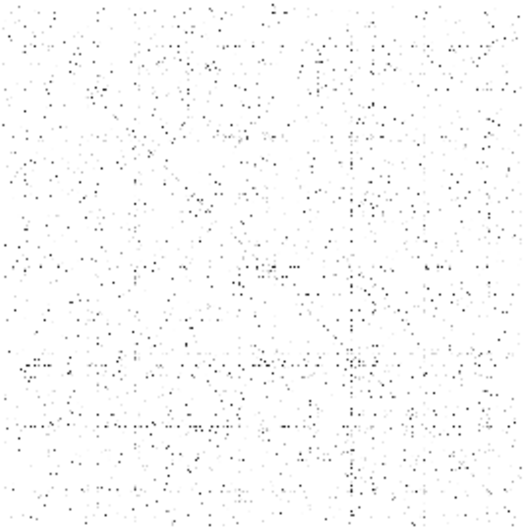
$$k_i^{out} = \sum_{j=1}^{N} A_{ij}$$

$$k_j^{in} = \sum_{i=1}^{N} A_{ij}$$

$$A_{ij} \neq A_{ji}$$
$$A_{ii} = 0$$

$$L = \sum_{i=1}^{N} k_i^{in} = \sum_{j=1}^{N} k_j^{out} = \sum_{i,j}^{N} A_{ij}$$

Real-life Networks are Sparse

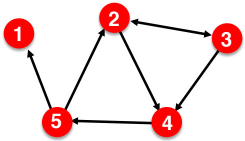


Adjacency matrix is filled with zeros.

Representing Graphs: Edge List

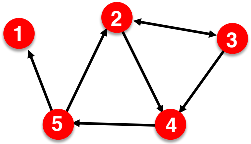
Represent graph as a list of edges.

- (2, 3)
- (2, 4)
- (3, 2)
- (3, 4)
- (4, 5)
- (5, 2)
- (5, 1)



Representing Graphs: Adjacency List

Represent graph as an adjacency list, easier to work with if a network is large and sparse.

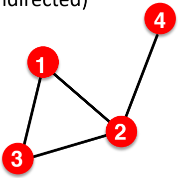


- 1:
- 2: 3, 4
- 3: 2, 4
- 4: 5
- 5: 1, 2

More Types of Networks

- **Unweighted**

  (undirected)



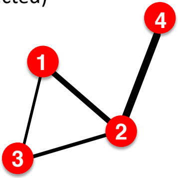
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad \bar{k} = \frac{2E}{N}$$

- **Weighted**

  (undirected)



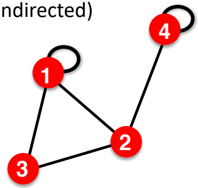
$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

More Types of Networks

- **Self-edges (self-loops)**
  (undirected)

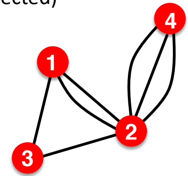


$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A_{ii} \neq 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1, i\neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii}$$

- **Multigraph**
  (undirected)



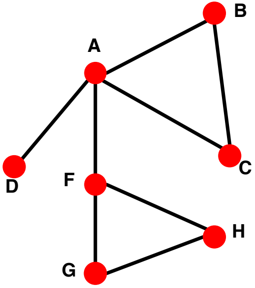
$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

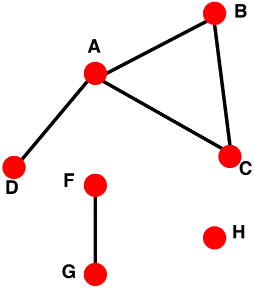
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Connectivity: Undirected Graphs

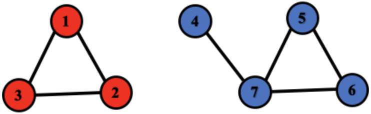
- Connected graph: any two nodes can be jointed by a path
- A disconnected graph is made up by two or more connected components.



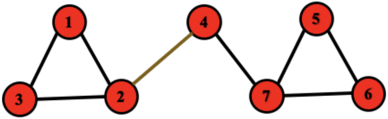


Connectivity: Undirected Graphs

The adjacency matrix of a network with several components can be written in a block-diagonal form: nonzero elements are confined to squares, with all other elements being zero



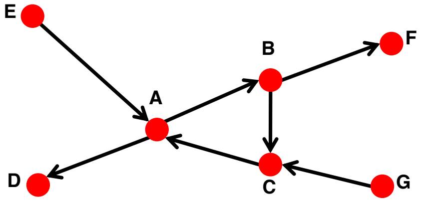
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

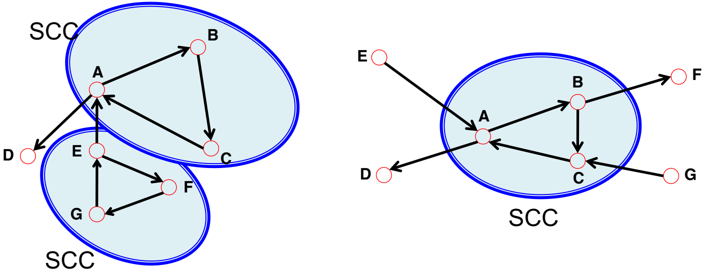
Connectivity: Directed Graphs

- Strongly connected directed graph has a path from each node to every other node and vice versa.
- Weakly connected directed graph is connected if we disregard the edge directions.



Connected but not strongly connected.

Connectivity: Directed Graphs



Strongly connected components (SCCs) can be identified, but not every node is part of a nontrivial strongly connected component.

Tarjan's SCC detection algorithm (worst-case $\mathcal{O}(N+E)$)