Introduction to Network Theory

Network Science '21: Session 1.2

Claudio J. Tessone

Blockchain & Distributed Ledger Technologies
UZH Blockchain Center

Lecture Objectives

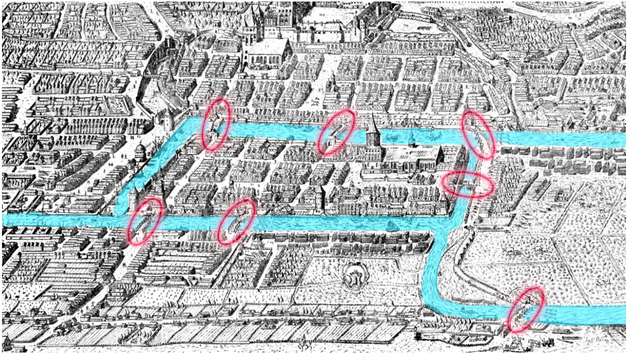
1. Define what is a network and identify basic types
2. Understand the concepts of degree and degree distribution
3. Understand the distributions we will see often in the course

Outlook

Contents

A Classic Example

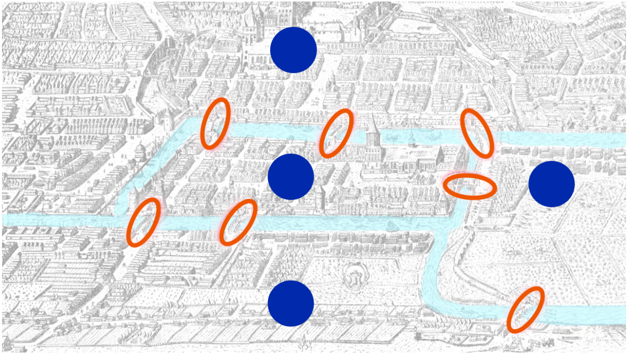
The bridges of Königsberg



Is there a trail that transverses each bridge exactly once?

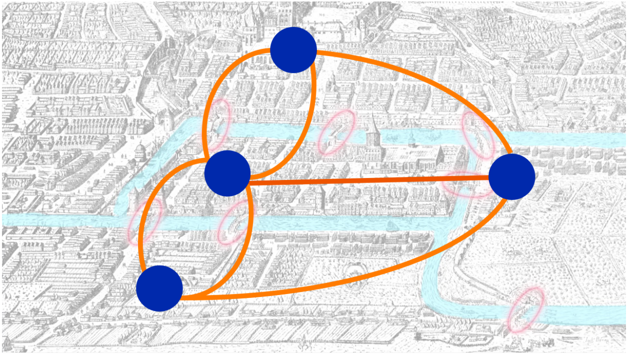
Euler's abstraction (1736): Regardless of where we are in a landmass, for the sake of the problem it is the same: Let us call them "stops"

The bridges of Königsberg



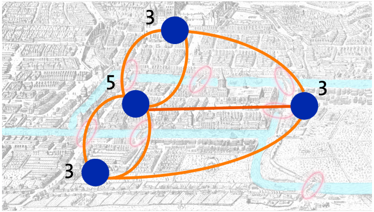
Geometry is unimportant. Only number of entries/exits to the stops matter

The bridges of Königsberg



Geometry is unimportant. Only number of entries/exits to the stops matter

Eulerian Path

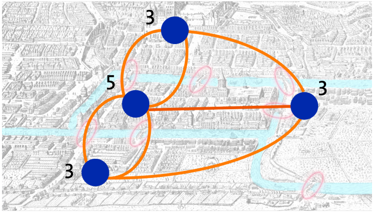


- *Intermediate stops* The number of bridges touching them must be **even**: there must be an entry and an exit point
- *Start and end points* might have an odd number of entries/exits

Result

It is not possible to make a trail as the problem asks

Eulerian Path

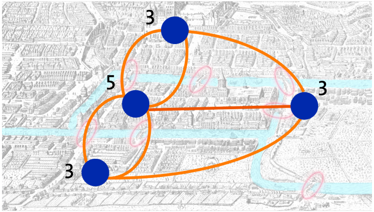


+ *Intermediate stops* The number of bridges touching them must be **even**: there must be an entry and an exit point
+ *Start and end points* might have an odd number of entries/exits

Result

It is not possible to make a trail as the problem asks

Eulerian Path

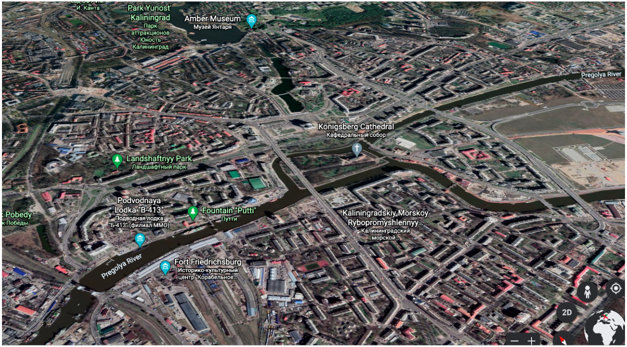


- *Intermediate stops* The number of bridges touching them must be **even**: there must be an entry and an exit point
- *Start and end points* might have an odd number of entries/exits

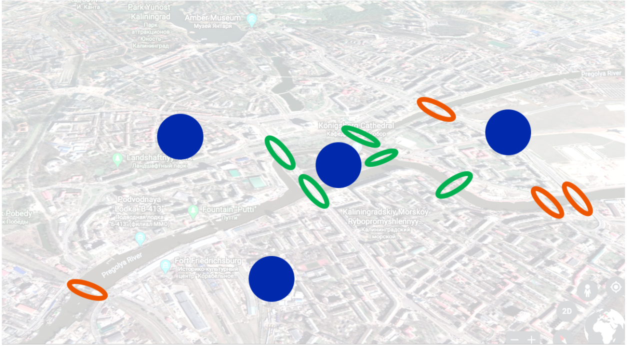
Result

It is not possible to make a trail as the problem asks

The bridges of Kaliningrad

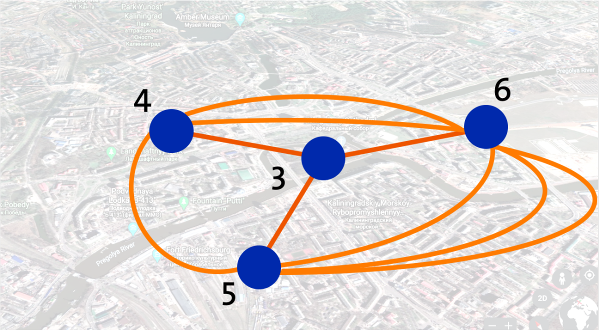


The bridges of Kaliningrad



- On the modern map of Kaliningrad: Green bridges survived until today; two bridges were destroyed in WWII; new bridges were built

The bridges of Kaliningrad



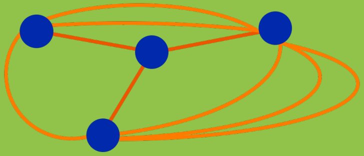
Now it is possible to create a trail. We know where (not) to start and finish

Graph and network theory

Applications of Graph theory

- **Computer Science** - graphs themselves are the *objects* of interest
- **Social Sciences** - connections between people in society
- **Electrical Engineering** - designing circuit connections
- **Linguistics** - graphs model grammar structures
- **Epidemiology** - contagion and diffusion processes in connected society
- **Chemistry** - graphs represent molecular structure
- ...

Constituents of a Network

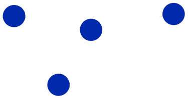


- There is a set of *nodes* or *vertices*
- There is a set of *edges* (or *links*) connecting them

Nodes

A set of nodes, denoted as V

Fundamental units composing the graph

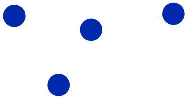


- *Names*: nodes, vertices, points, actors
- *They may abstract:* individuals, websites, geographical locations, banks
- *They are usually featureless*

Nodes

A set of nodes, denoted as V

Fundamental units composing the graph

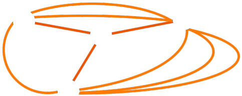


- *Names*: nodes, vertices, points, actors
- *They may abstract:* individuals, websites, geographical locations, banks
- *They are usually featureless*

Edges

A set of edges, denoted as E

Represent relationships/connections between nodes

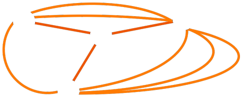


- *Names*: edges, arcs, lines, links
- They may abstract: friendships / followers / subscribers, web-links, reachability, loans
- *Might have features:* weight, distance, assets

Edges

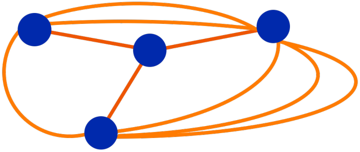
A set of edges, denoted as E

Represent relationships/connections between nodes



- *Names*: edges, arcs, lines, links
- They may abstract: friendships / followers / subscribers, web-links, reachability, loans
- *Might have features:* weight, distance, assets

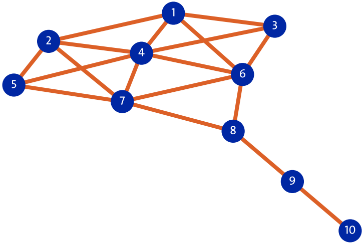
Graph



- Graph is an **ordered pair** $G = (V, E)$
- In networks, *network size*; In graph theory, **order** of the graph: $|V|$
- In graph theory, **size** of the graph: $|E|$

Graph

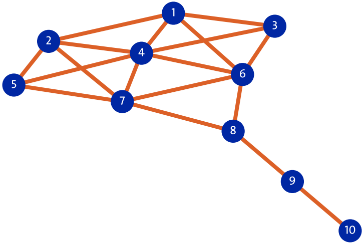
- E consists of 2-element subsets of V
- Some *vertices* may *not belong* to any edge, but *all edges belong* to a pair of vertices
- Vertices belonging to an edge are called *ends* of the edge



*Vertices connected by an edge are called **neighbouring** or **adjacent***

Graph

+ E consists of 2-element subsets of V
+ Some *vertices* may *not belong* to any edge, but *all edges belong* to a pair of vertices
+ Vertices belonging to an edge are called *ends* of the edge



Vertices connected by an edge are called ***neighbouring*** *or* ***adjacent***

A graph is the mathematical object formally defined above

A network is the representation of a real-world system. Nodes and links have a specific meaning within the context of the application. Also, they have attributes

Graph theory vs network science

- Graph theory mainly deals with formal / theoretical questions
- Network Science deals with applied questions
- Graph techniques can be used to analyse networks
- All networks are graphs
- Not all graphs are networks

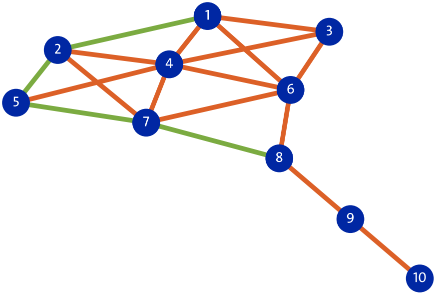
Simplest graphs

Trivial graph has only one vertex

Null graph has no edges

Path

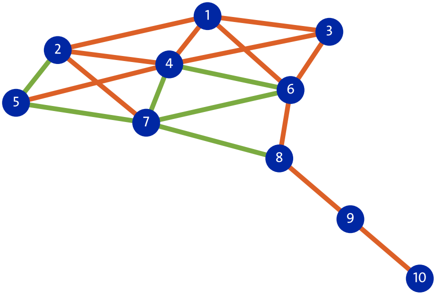
An alternating sequence of nodes and edges, beginning and ending at given node(s). ***Nodes or vertices are visited only once***



1 - 2 - 5 - 7 - 8
is a ***path***

Walk

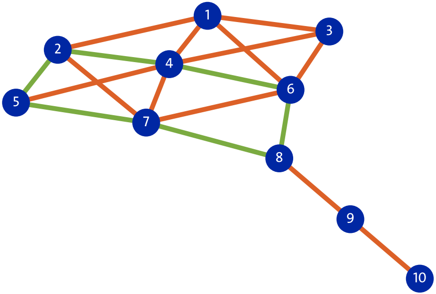
An alternating sequence of nodes and edges, beginning and ending at given node(s). ***Nodes or vertices can be visited more than once***



2 - 5 - 7 - 4 - 6 - 7 - 8
is a ***walk***

Cycle

Cycle *or* ***circuit*** *is a path that starts and ends in the same node*



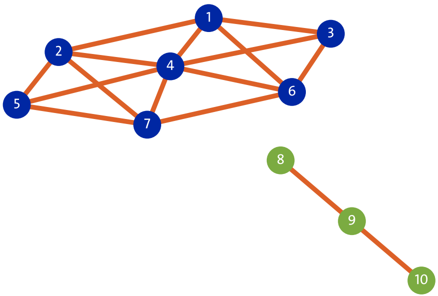
2 - 4 - 6 - 8 - 7 - 5 - 2
is a ***cycle***

Connectivity

- A node is reachable from another node if there exists a path of any length from one node to another
- A graph is connected if there exists a path of any length between any pair of nodes (i.e., all nodes are reachable from each other)

Connected components

connected component is a subgraph in which all nodes are reachable from each other



$\{1,2,3,4,5,6,7\}$ *is a* ***connected component***

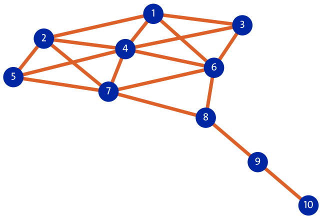
$\{8,9,10\}$ *is a* ***connected component***

Graph representations

Adjacency matrix

Definition

$\mathbb{A} = \{a_{ij}\}_{i,j=1}^{N}$ is a matrix whose indices are nodes and the entries of the matrix represent the edges



... to...

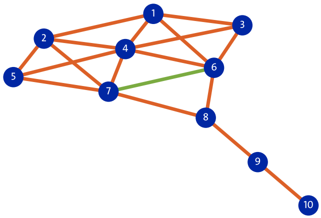
points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0

Adjacency matrix

Definition

$\mathbb{A} = \{a_{ij}\}_{i,j=1}^{N}$ is a matrix whose indices are nodes and the entries of the matrix represent the edges



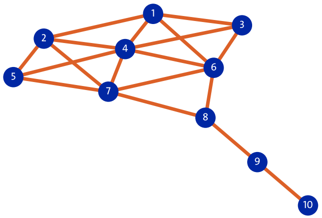
... to...

points from ...	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0

Adjacency matrix

Definition

$\mathbb{A} = \{a_{ij}\}_{i,j=1}^{N}$ is a matrix whose indices are nodes and the entries of the matrix represent the edges



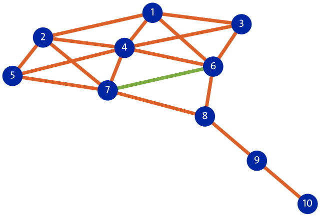
... to...

points from ...	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0

Edge list

Definition

Each entry contains an edge between vertices

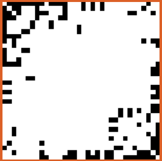


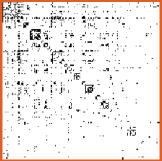
1	2
1	3
1	4
1	6
2	4
2	5
2	7
3	4
3	6
4	5
4	6
4	7
6	7
6	8
5	7
7	8
8	9
9	10

Adjacency matrix vs. Edge list

	Adjacency matrix	Edge list
Memory	$O(N^2)$	$O(E)$
Lookup specific edge	Fast, $O(1)$	Slow
Iterate over all edges	Slow, $O(N^2)$	Fast
Find neighbours of a node	Time $O(N)$	Time $O(E)$
Better for	Dense graphs	Sparse graphs
Adding new vertices	Hard	Easy
Adding new edges	$O(1)$	$O(1)$ or $O(E)$

Empirical fact: most real-world networks are *sparse*





Network types

Network types

1. By direction of edges:
   1.1 Directed
   1.2 Undirected
2. By weights of edges:
   2.1 Weighted
   2.2 Unweighted

Any combination is possible

One-mode networks: All nodes are of the same nature. e.g. in a social network, all nodes represent persons

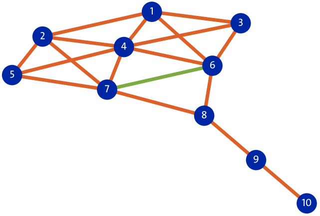
Undirected, unweighted network

The simplest way of visualising is to think that pairs describing edges are unordered $(a, b) \equiv (b, a)$

... to...

points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0



1	2
1	3
1	4
1	6
2	4
2	5
2	7
3	4
3	6
4	5
4	6
4	7
6	7
6	8
5	7
7	8
8	9
9	10

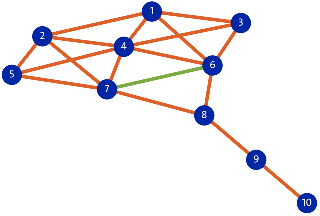
Undirected, unweighted network

- The adjacency matrix $\mathbb{A}$ is symmetric
- The elements are binary: $a_{ij} \in \{0, 1\}$ .. *Nodes are connected or not*

... to...

points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0

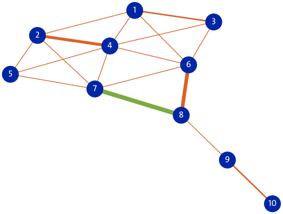


1	2
1	3
1	4
1	6
2	4
2	5
2	7
3	4
3	6
4	5
4	6
4	7
6	7
6	8
5	7
7	8
8	9
9	10

Friendship network of Facebook users

Undirected, weighted network

Edges are unordered, but carry a weight



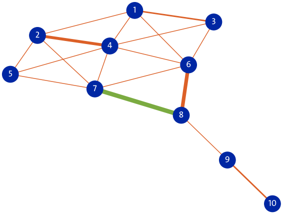
... to...

points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	1	2	1	0	1	0	0	0	0
2	1	0	0	4	1	0	1	0	0	0
3	2	0	0	1	0	1	0	0	0	0
4	1	4	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	5	0	0
7	0	1	0	1	1	1	0	6	0	0
8	0	0	0	0	0	5	6	0	1	0
9	0	0	0	0	0	0	0	1	0	2
10	0	0	0	0	0	0	0	0	2	0

Undirected, weighted network

- The adjacency matrix $\mathbb{A}$ is symmetric
- The elements are binary: $a_{ij} \in \mathbb{R}$ *Connection between nodes have a* **weight**



... to...

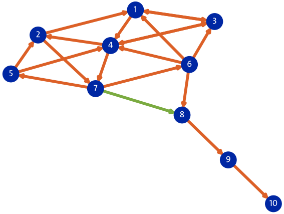
points from ...	1	2	3	4	5	6	7	8	9	10
1	0	1	2	1	0	1	0	0	0	0
2	1	0	0	4	1	0	1	0	0	0
3	2	0	0	1	0	1	0	0	0	0
4	1	4	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	5	0	0
7	0	1	0	1	1	1	0	6	0	0
8	0	0	0	0	0	5	6	0	1	0
9	0	0	0	0	0	0	0	1	0	2
10	0	0	0	0	0	0	0	0	2	0

Undirected, weighted network

Cooperation network between individuals in ICIC (1919-1927). Weight is the number of cooperation works

Directed, unweighted network

Edges are ordered. An edge $(a, b) \neq (b, a)$



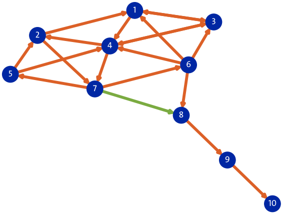
... to...

points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	0	0
3	1	0	0	1	0	0	0	0	0	0
4	0	1	1	0	0	0	1	0	0	0
5	0	1	0	1	0	0	0	0	0	0
6	1	0	1	1	0	0	0	1	0	0
7	0	0	0	0	1	1	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

Directed, unweighted network

+ The adjacency matrix $\mathbb{A}$ is **asymmetric**
+ The elements are binary: $a_{ij} \in [0, 1]$ if i is pointing to j

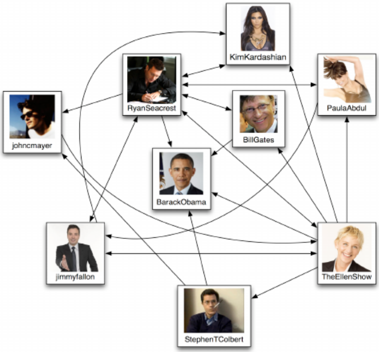


... to...

points from ...	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	0	0	0	0	0
2	1	0	0	0	0	0	1	0	0	0
3	1	0	0	1	0	0	0	0	0	0
4	0	1	1	0	0	0	1	0	0	0
5	0	1	0	1	0	0	0	0	0	0
6	1	0	1	1	0	0	0	1	0	0
7	0	0	0	0	1	1	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0

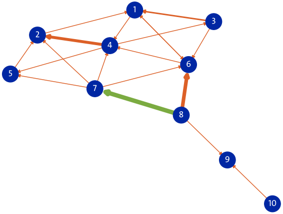
Directed, unweighted network

Twitter follower network



Directed, weighted network

- The adjacency matrix $\mathbb{A}$ is **asymmetric**
- The elements are binary: $a_{ij} \in \mathbb{R}$ Order of the nodes is important and carry information about the weights

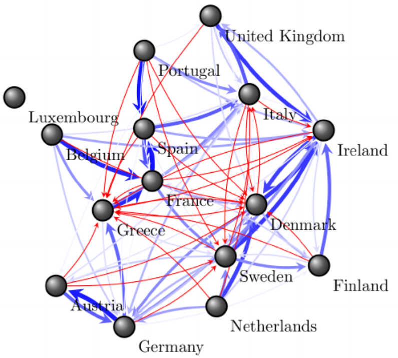


... to...

points from ...	1	2	3	4	5	6	7	8	9	10
1	0	0	0	1	0	1	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0
3	2	0	0	0	0	1	0	0	0	0
4	0	4	1	0	1	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	0
6	1	0	0	1	0	0	0	0	0	0
7	0	1	0	1	1	1	0	0	0	0
8	0	0	0	0	0	5	6	0	1	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0

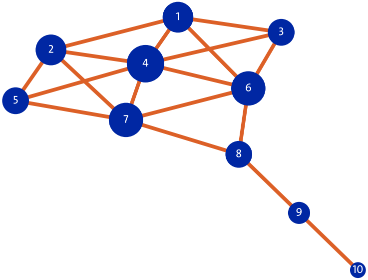
One-mode directed weighted

Affinity network of EU countries at Eurovision.



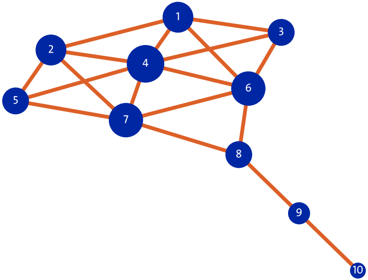
Node Degree and Degree Distribution

Node degree



Two nodes are neighbours if there is a link between them

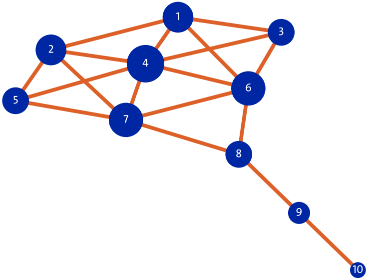
Node degree



Definition

For an undirected, unweighted network, the degree of a node i is equal to the number of edges that have i as an end. It is denoted by k_i

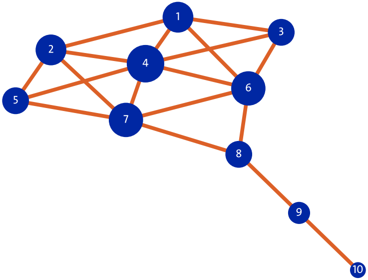
Node degree



By recourse of the adjacency matrix, the degree of a node i can be computed as

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

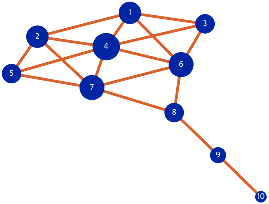
Node degree



The list of all nodes' degrees

Node i	1	2	3	4	5	6	7	8	9	10
Degree k_i	4	4	3	6	3	5	5	3	2	1

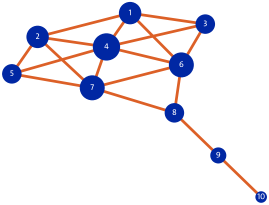
Degree distribution



Definition

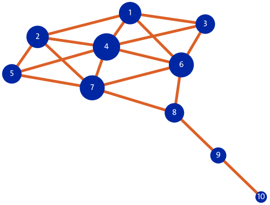
Denoted by $p(k)$. It is a function of the degree. Its value is the fraction of nodes with degree k

Degree distribution



- $p(k=1) = 1/10$ ... there is a single node with degree 1
- $p(k=5) = 5/10$ ... there are two nodes with degree 3

Degree distribution



Degree d	0	1	2	3	4	5	6
Degree distribution $P(d)$	0	0.1	0.1	0.3	0.2	0.2	0.1

Why are we interested in degree distributions?

Because in real-world networks...

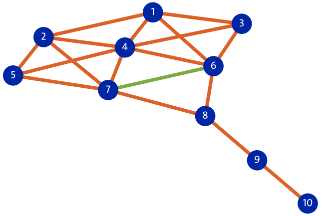
- it cannot be explained by simple network models
- it exhibits statistical regularities that allow us to understand *how networks are formed*
- it allow us to uncover properties of processes that take place on networks

Global measures related to degree

... to...

points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0



The number of links $L = |V|$ can be computed as a sum over the adjacency matrix

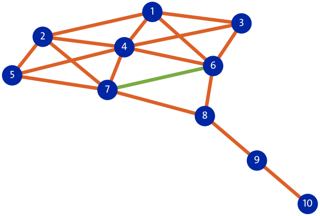
$$L = \sum_{i,j=1}^{N} a_{i,j}$$

Global measures related to degree

... to...

points from ...

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	0	1	0	0	0	0
2	1	0	0	1	1	0	1	0	0	0
3	1	0	0	1	0	1	0	0	0	0
4	1	1	1	0	1	1	1	0	0	0
5	0	1	0	1	0	0	1	0	0	0
6	1	0	1	1	0	0	1	1	0	0
7	0	1	0	1	1	1	0	1	0	0
8	0	0	0	0	0	1	1	0	1	0
9	0	0	0	0	0	0	0	1	0	1
10	0	0	0	0	0	0	0	0	1	0



The network density δ is the fraction of links compared to the theoretical maximum (in this case, for an undirected network)

$$\delta = \frac{2L}{N(N-1)}$$

For other networks

- For directed networks, there are two degrees
  - *in-degree*: Number of incident edges on i

$$k_i^{in} = \sum_{j=1}^{N} a_{ji}$$

  - *out-degree*: Number of edges starting from i

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

- For weighted networks, there is the related notion of *strength*

$$s_i = \sum_{j=1}^{N} a_{ij}$$

For other networks

- For **directed networks**, there are two degrees
  - *in-degree*: Number of incident edges on i

$$k_i^{in} = \sum_{j=1}^{N} a_{ji}$$

  - *out-degree*: Number of edges starting from i

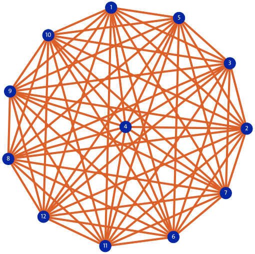
$$k_i^{out} = \sum_{j=1}^{N} a_{ij}$$

- For **weighted networks**, there is the related notion of *strength*

$$s_i = \sum_{j=1}^{N} a_{ij}$$

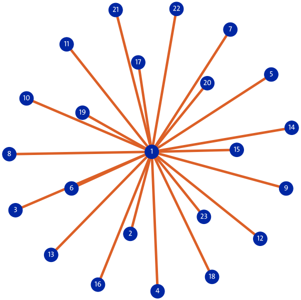
Simple networks

Basic networks: Fully connected



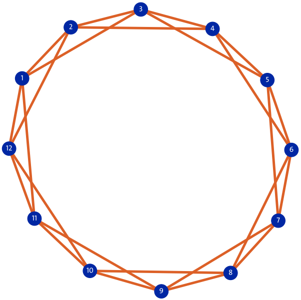
- Underlying network when all nodes are interconnected
- *In most situations: artificial, however, it is the base topology used in most analytical derivations*
- For N vertices: $k_i = N - 1$

Basic networks: Star network



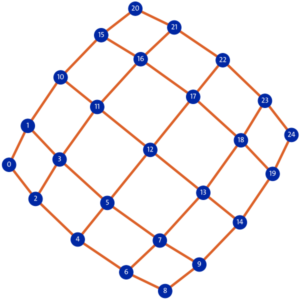
- Extreme centralisation / hierarchical organisation
- For N vertices: $k_1 = N - 1; k_i = 1$, for $i > 1$

One-dimensional lattice



- nodes are placed in a linear space. They are connected to κ nearest neighbours to the right and left
- κ is called the coordination number
- $k_i = 2\kappa$
- ”periodic boundary conditions” mean that the left-most node is a neighbour of the right-most node

Two-dimensional lattice



- nodes are placed in a bi-dimensional space. They are connected to κ nearest neighbours to east, west, north, south
- $k_i = 4\kappa$

A note on binning

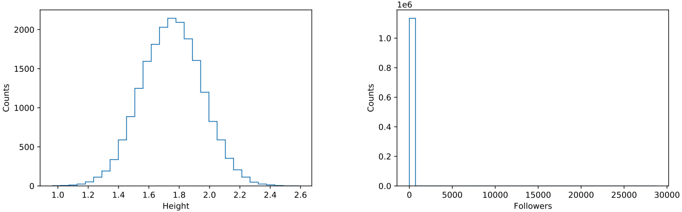
Distributions

For simplicity, this note is made for continuous distributions

Given a random variable x, *probability density function* $p(x)$ is one such that

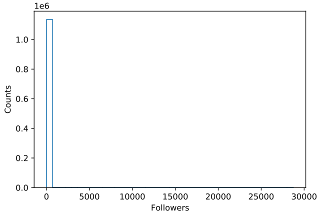
$$\text{Prob}[a \leq x \leq b] = \int_a^b p(x)\,dx$$

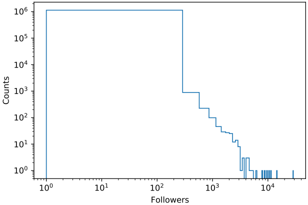
Histograms



- A raw histogram is just counting how many occurrences exist in each *bin*
- You first define a set of values $(x_1, x_2, \cdots, x_b)$,
- Then compute the number of observations in each bin: $\text{count}(x_1 \leq x \leq x_2)$, $\text{count}(x_2 \leq x \leq x_3)$, ...
- if all bins are of the same size, the plot looks similar (except for the scale) to the plot of the PDF

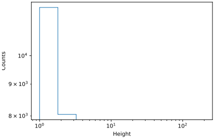
Heavy-tailed distributions

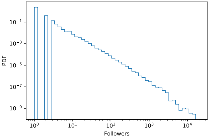




- In many situations (**and in networks it happens very often**), the distributions involved are *heavy-tailed*
- These distributions when plotted in linear scales do not convey information

Distributions in log-scale





- When plotting distributions in logarithmic scale, you bin logarithmically the data. I.e. $x_2 = a\,x_1, x_3 = a^2\,x_1$, ...$x_N = a^N\,x_1$
- Then, you must plot the distributions, i.e.

$$p(a \leq x \leq b) \approx \text{count}(a \leq x \leq b)/(b-a)$$

- *Bins should be of similar size in the scale in which they are plotted*

Claudio J. Tessone

Blockchain & Distributed Ledger Technologies
UZH Blockchain Center

claudio.tessone@uzh.ch
http://www.ifi.uzh.ch/bdlt