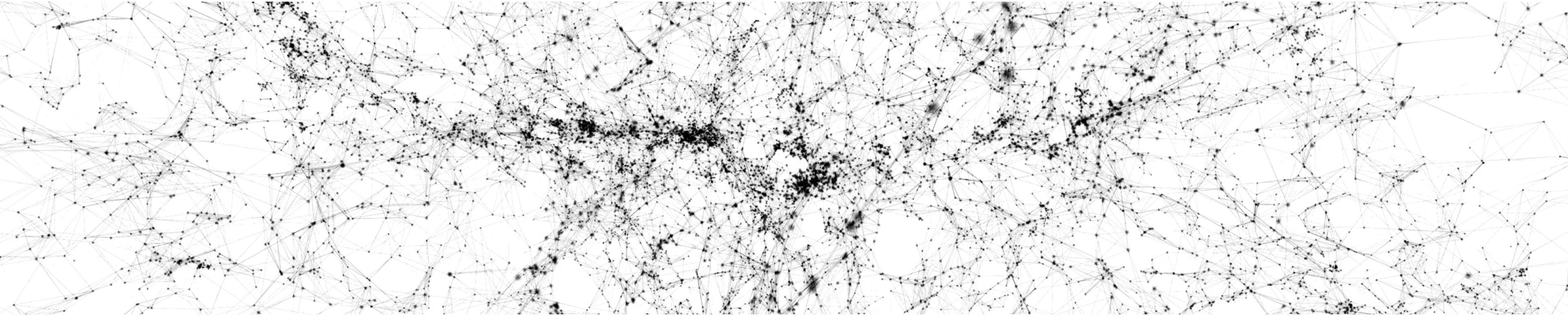
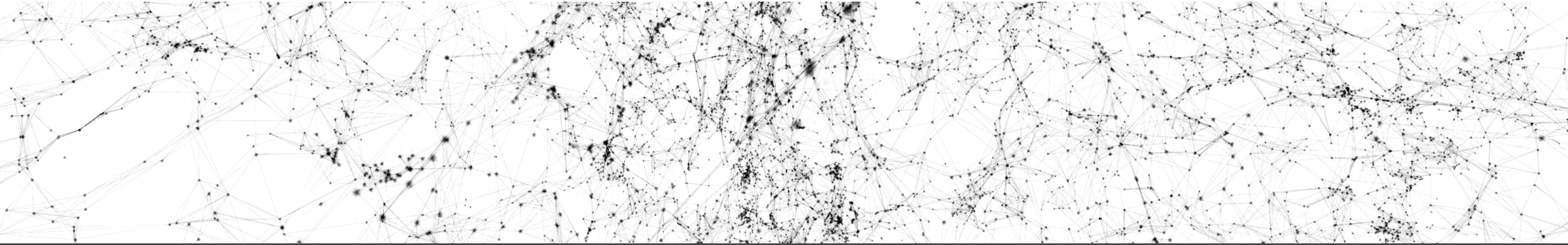


Background

Analysis of complex interconnected data



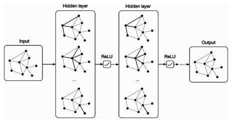
Timeline of notable works in network science

- **Temporal Graph Learning**
- **Graphs + LLMs**

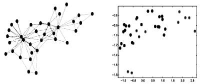
2023~

2015~2023

- **Graph Neural Networks**
- Deep Learning for Networks
- High-Order Networks [Benson et al.]



Recent Trend: Deep Learning for Graphs

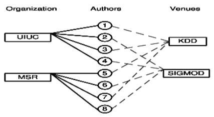


- Info. vs. Social Networks (Twitter) [Kwak et al.]
- **Signed** Networks [Leskovec et al.]
- Semantic Social Networks [Tang et al.]
- Four Deg. Of Separation [Backstrom et al.]
- Structural Diversity [Ugander et al.]
- Computational Social Science [Watts]
- **Network Embedding** [Perozzi et al.]

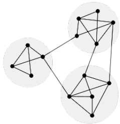
2010~2014

2005~2009

- Graph Evolution [Leskovec et al.]
- 3 Deg. Of Influence [Christakis & Fowler]
- Social **Influence** Analysis [Tang et al.]
- Six Deg. Of Separation [Leskovec & Horvitz]
- Network **Heterogeneity** [Sun & Han]
- Network **Embedding** [Tang & Liu]
- Computer Social Science [Lazer et al.]



21st Century: More CS

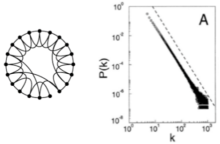


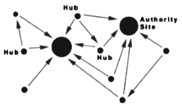
- Influence Max'n [Domingos & Kempe et al.]
- **Community Detection** [Girvan & Newman]
- Network Motifs [Milo et al.]
- Link Prediction [Liben-Nowell & Kleinberg]

2000~2004

1998~1999

- **Small Worlds** [Watts & Strogatz]
- **Scale Free** [Barabasi & Albert]
- **Power Law** [Faloutsos x3]



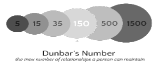


- **HITS** [Kleinberg]
- **PageRank** [Page & Brin]
- Hyperlink Vector Voting [Li]

1997

1992

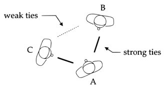
- Structural Hole [Burt]
- **Dunbar's Number** [Dunbar]

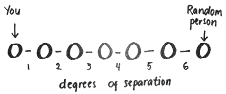


Late 20th Century: CS & Physics

1970s

- The Strength Of **Weak Tie** [Granovetter]





- **Small Worlds** [Migram]

1960s

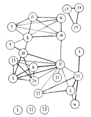
1950s

- **Random Graph** [Erdos, Renyi, Gilbert]
- Degree Sequence [Tuttle, Havel, Hakami]

- **Homophily** [Lazarsfeld & Merton]
- Balance Theory [Heider et al.]

1930s

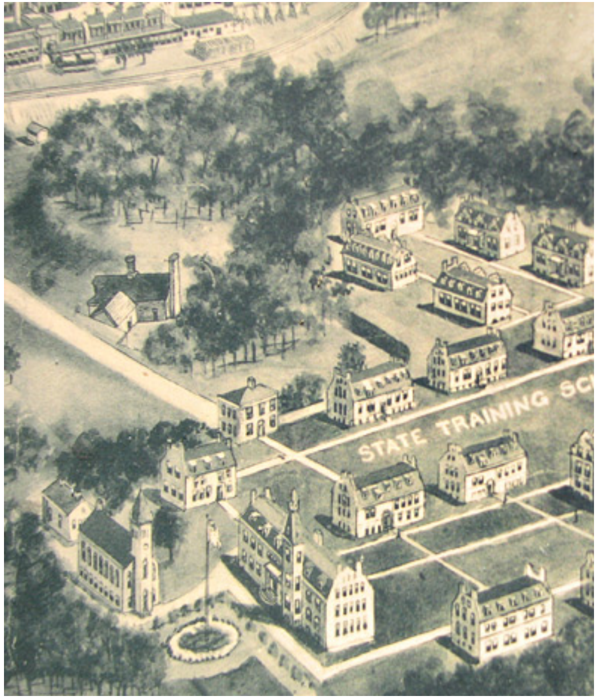
- **Sociogram** [Moreno]



20th Century: Sociology

Based on Slides from Jie Tang

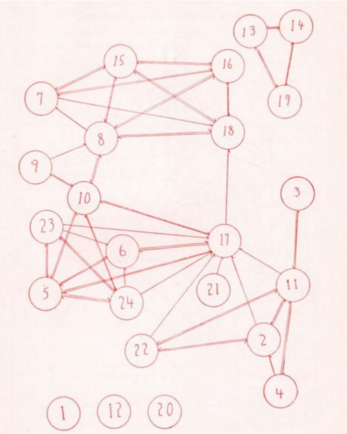
How to explain the pandemic of runaways?



Jacob L. Moreno,

Mapped out the **channels for the flow of social influence and ideas**, and concluded that they **behaved based on how they are positioned in their social network**

Read more here



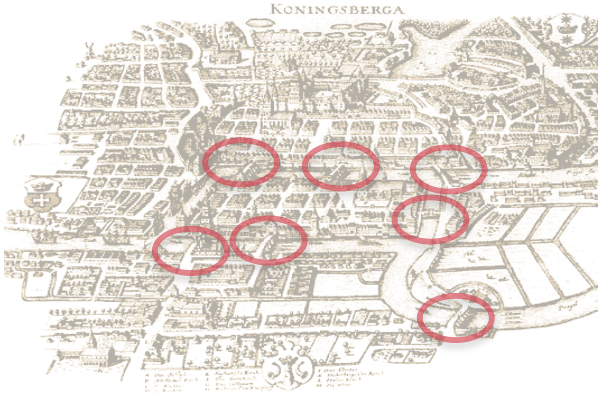
earliest graphical depictions of social networks (sociograms)

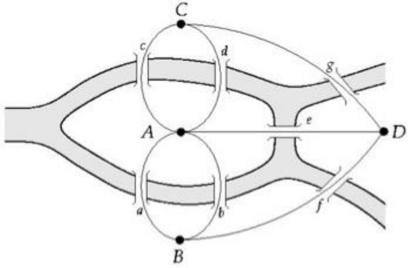
Who Shall Survive? (1934)

Graph Theory & Network Science

Graph theory is older than network science

Can one walk across the seven bridges and never cross the same bridge twice? [see the video]





1735: Euler's theorem:

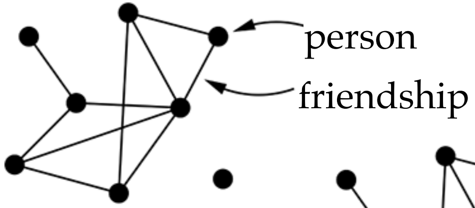
If a graph has more than two nodes of odd degree, there is no [Eulerian] path. If a graph is connected and has no odd degree nodes, it has at least one path.

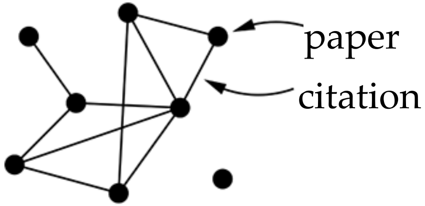
Network science borrows many concepts/theories from graph theory. The focus, however, is on **real world** graphs which have specific characteristics, and are different from random graph families commonly studied in math.

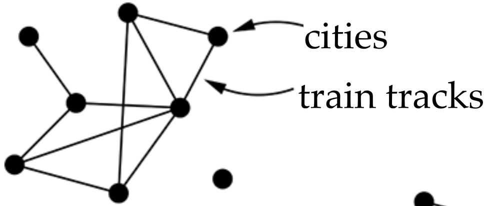
For example, regular graphs (same degree for all nodes), are irrelevant here.

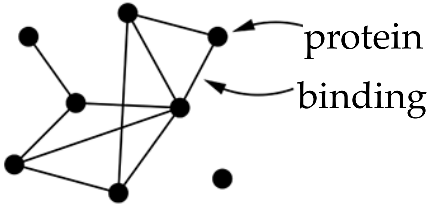
Interconnected Data as Graphs

- Nodes (or Vertices)
    - Proteins, Neurons, People
- Edges (or Links)
    - interactions, friendships









- Two vertices are **adjacent** if they share a common edge
- Two adjacent vertices are **neighbours**
- An edge is **incident** with another edge if they share a vertex
- An edge is incident with two vertices

Adjacency Matrix: the default data structure

Adjacency Matrix

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	0	0	0	0	0	0	0	0	1
1	1	0	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	1	1	0	0	0	0	0	0
4	0	0	0	1	0	1	1	0	0	0	0	0
5	0	0	0	1	1	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	1	1	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	1	1
10	0	0	0	0	0	0	0	0	1	1	0	1
11	1	0	0	0	0	0	0	0	0	1	1	0

$A \in \{0,1\}^{N \times N}$

A square matrix of size N (number of nodes)

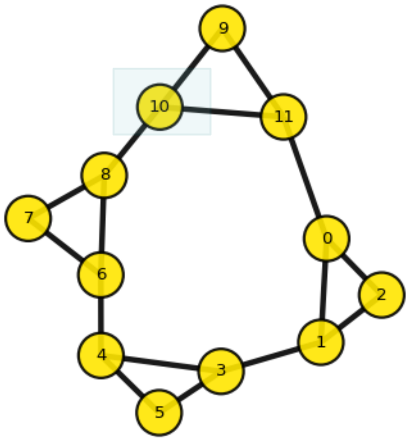
Adjacency List

0 : { 1 , 2 , 11 }
1 : { 0 , 2 , 3 }
2 : { 0 , 1 }
3 : { 1 , 4 , 5 }
4 : { 3 , 5 , 6 }
5 : { 3 , 4 }
6 : { 4 , 7 , 8 }
7 : { 6 , 8 }
8 : { 6 , 7, 10 }
9 : { 10 , 11 }
10 : { 8 , 9 , 11 }
11 : { 0 , 9 , 10 }

Edge List

{ (0, 1), (0, 2), (0, 11),
(1, 0), (1, 2), (1, 3),
(2, 0), (2, 1),
(3, 1), (3, 4), (3, 5),
(4, 3), (4, 5), (4, 6),
(5, 3), (5, 4),
(6, 4), (6, 7), (6, 8),
(7, 8), (7, 6),
(8, 6), (8, 7), (8, 10)
(9, 10), (9,11),
(10, 8), (10, 9), (10, 11),
(11, 0), (11, 9), (11, 10) }

Simple Graph



$G(V, E), E \subseteq \{(i, j) \mid (i, j) \in V^2\}$

V is set of nodes, here: {0, 1, 2 … 11}

E *is* set of edges, here the edge list

Adjacency Matrix: sparse representation

Real world graphs are sparse (lots of zeros) => use sparse matrix representations to only store non-zero elements, in a specific format, often:

- LIL (List of lists): similar to adjacency list
- COO (Coordinate list): similar to edge list
- CSR (Compressed Sparse Row)
  - store only start index of each row
  - fast row access and matrix-vector multiplications

0 : { 1 , 2 , 11 }
1 : { 0 , 2 , 3 }
2 : { 0 , 1 }
3 : { 1 , 4 , 5 }
4 : { 3 , 5 , 6 }
5 : { 3 , 4 }
6 : { 4 , 7 , 8 }
7 : { 6 , 8 }
8 : { 6 , 7, 10 }
9 : { 10 , 11 }
10 : { 8 , 9 , 11 }
11 : { 0 , 9 , 10 }

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	0	0	0	0	0	0	0	0	1
1	1	0	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	1	1	0	0	0	0	0	0
4	0	0	0	1	0	1	1	0	0	0	0	0
5	0	0	0	1	1	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	1	1	0	0	0
7	0	0	0	0	0	0	1	0	1	0	0	0
8	0	0	0	0	0	0	1	1	0	0	1	0
9	0	0	0	0	0	0	0	0	0	0	1	1
10	0	0	0	0	0	0	0	0	1	1	0	1
11	1	0	0	0	0	0	0	0	0	1	1	0

COL: [1 , 2 , 11 , 0 , 2 , 3 , 0 , 1 , 1 , 4 , 5 , 3 , 5 , 6 , 3 , 4 , 4 , 7 , 8 , 6 , 8 , 6 , 7, 10 , 10 , 11 , 8 , 9 , 11 , 0 , 9 , 10]
ROW: [0 , 3 , 6 , 8 , 11 , 14 , 16 , 19 , 21 , 24 , 26 , 29 , 32]

- CSC (Compressed Sparse Column)

LIL and COO are good for constructing matrices. Once a matrix has been constructed, convert to CSR or CSC format for fast arithmetic and matrix vector operations

Adjacency Matrix: marginals

marginals of A => degree sequence

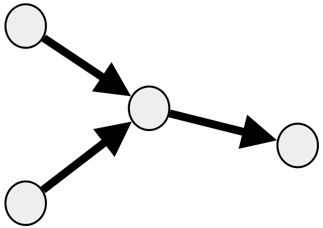
$$d_i = \sum_j A_{ij}$$

Simple graphs are symmetric, i.e., $A_{ij} = A_{ji}$

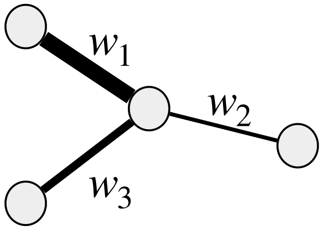
A	0	1	2	3	4	5	6	7	8	9	10	11	
0	0	1	1	0	0	0	0	0	0	0	0	1	3
1	1	0	1	1	0	0	0	0	0	0	0	0	3
2	1	1	0	0	0	0	0	0	0	0	0	0	2
3	0	1	0	0	1	1	0	0	0	0	0	0	3
4	0	0	0	1	0	1	1	0	0	0	0	0	3
5	0	0	0	1	1	0	0	0	0	0	0	0	2
6	0	0	0	0	1	0	0	1	1	0	0	0	3
7	0	0	0	0	0	0	1	0	1	0	0	0	2
8	0	0	0	0	0	0	1	1	0	0	1	0	3
9	0	0	0	0	0	0	0	0	0	0	1	1	2
10	0	0	0	0	0	0	0	0	1	1	0	1	3
11	1	0	0	0	0	0	0	0	0	1	1	0	3
	3	3	2	3	3	2	3	2	3	2	3	3	

Beyond Simple Graphs

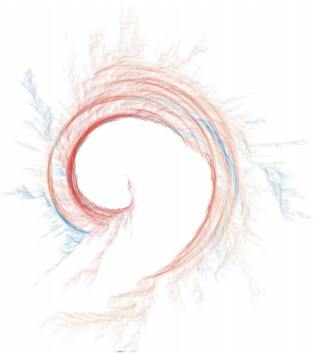
- Directions
  - E.g. who follows who at Twitter

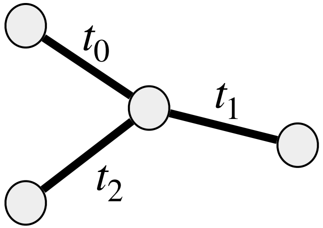


- Weights
  - E.g. friendship strength, or travel cost



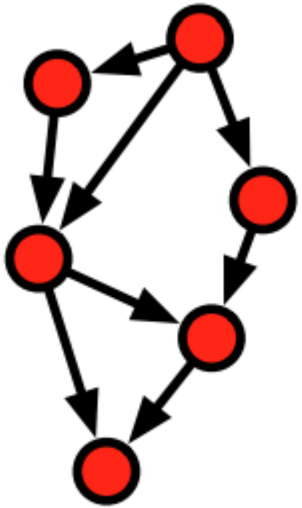
- Time
  - E.g. your friendships changes



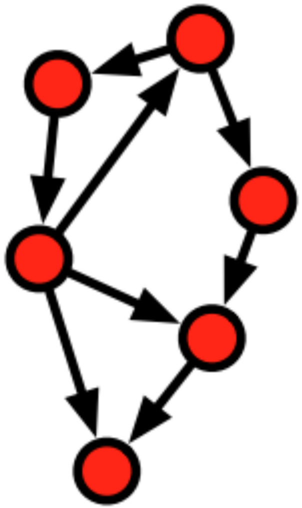


Directed Networks Examples

citation networks
foodwebs*
epidemiological



directed acyclic graph



directed graph

WWW
friendship?
flows of goods, information
economic exchange
dominance
neuronal
transcription
time travelers

From Clauset's slides

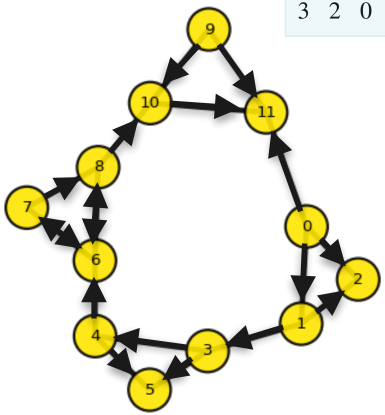
Adjacency Matrix: marginals of directed graph

marginals of A => in/out degrees

$$d_i^{in} = \sum_j A_{ij} \qquad d_i^{out} = \sum_j A_{ji}$$

$$A_{ij} = 1 \iff \exists \text{ an edge from } j \text{ to } i$$

A	0	1	2	3	4	5	6	7	8	9	10	11	in-degrees
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	0	0	0	0	0	2
3	0	1	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	1	0	0	0	0	0	0	0	0	1
5	0	0	0	1	1	0	0	0	0	0	0	0	2
6	0	0	0	0	1	0	0	1	1	0	0	0	3
7	0	0	0	0	0	0	1	0	0	0	0	0	1
8	0	0	0	0	0	0	1	1	0	0	0	0	2
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	1	0	0	2
11	1	0	0	0	0	0	0	0	0	1	1	0	3
out-degrees	3	2	0	2	2	0	2	2	2	2	1	0	



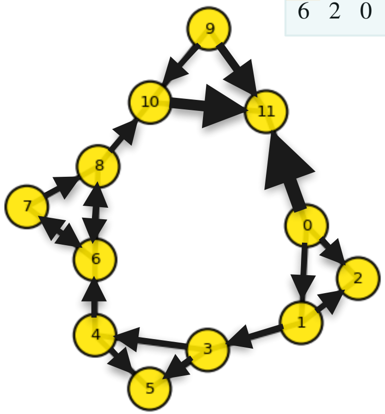
Adjacency Matrix: marginals of weighted directed graph

marginals of A => in/out weighted degrees

$$d_i^{in} = \sum_j A_{ij} \qquad d_i^{out} = \sum_j A_{ji}$$

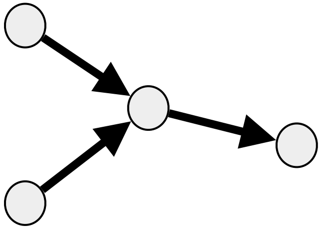
$$A \in \mathbb{R}^{N \times N}$$

A	0	1	2	3	4	5	6	7	8	9	10	11	in-degrees
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	0	0	0	0	0	2
3	0	1	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	1	0	0	0	0	0	0	0	0	1
5	0	0	0	1	1	0	0	0	0	0	0	0	2
6	0	0	0	0	1	0	0	1	1	0	0	0	3
7	0	0	0	0	0	0	1	0	0	0	0	0	1
8	0	0	0	0	0	0	1	1	0	0	0	0	2
9	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	1	0	0	2
11	4	0	0	0	0	0	0	0	0	2	3	0	9
out-degrees	6	2	0	2	2	0	2	2	2	3	3	0	

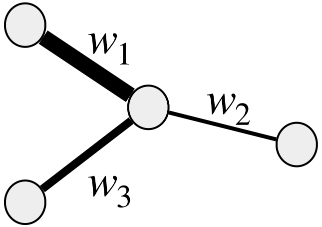


Beyond Simple Graphs

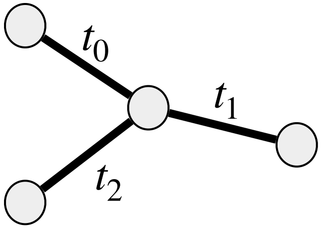
- Directions
  - E.g. who follows who at Twitter



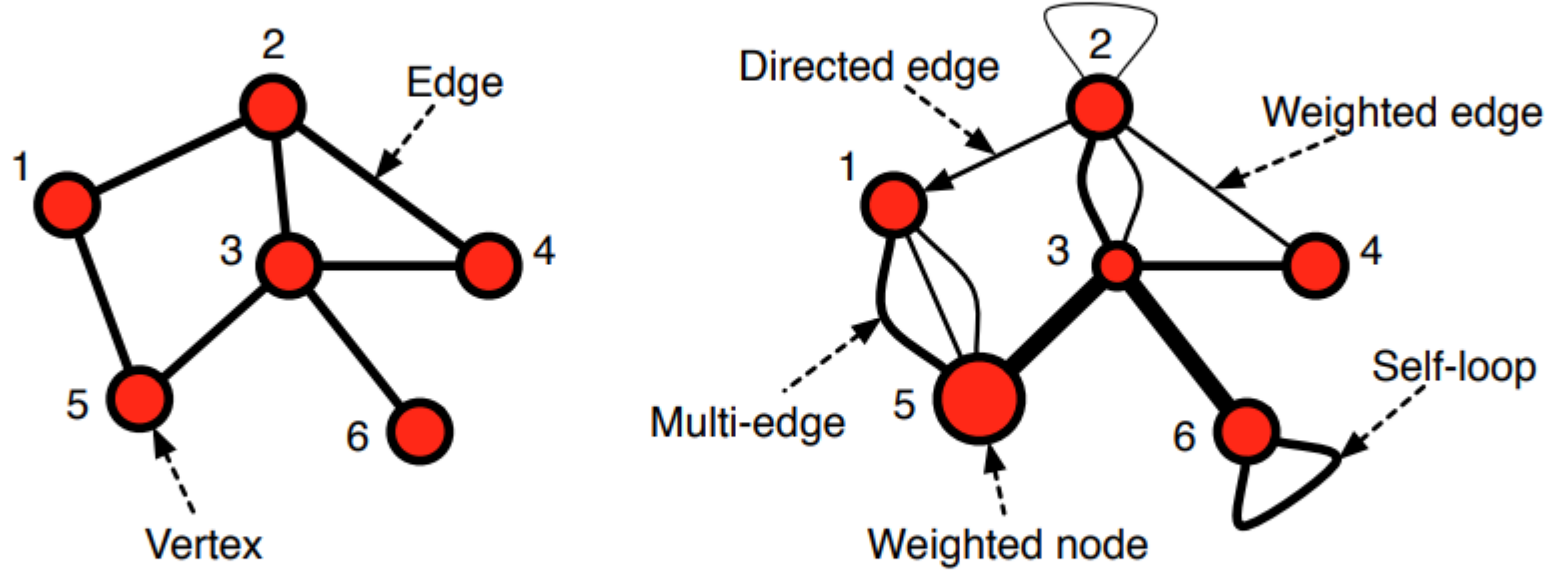
- Weights
  - E.g. friendship strength, or travel cost



- Time
  - E.g. your friendships changes
  - Triplets: (u, v, t) or tensors or graph snapshots

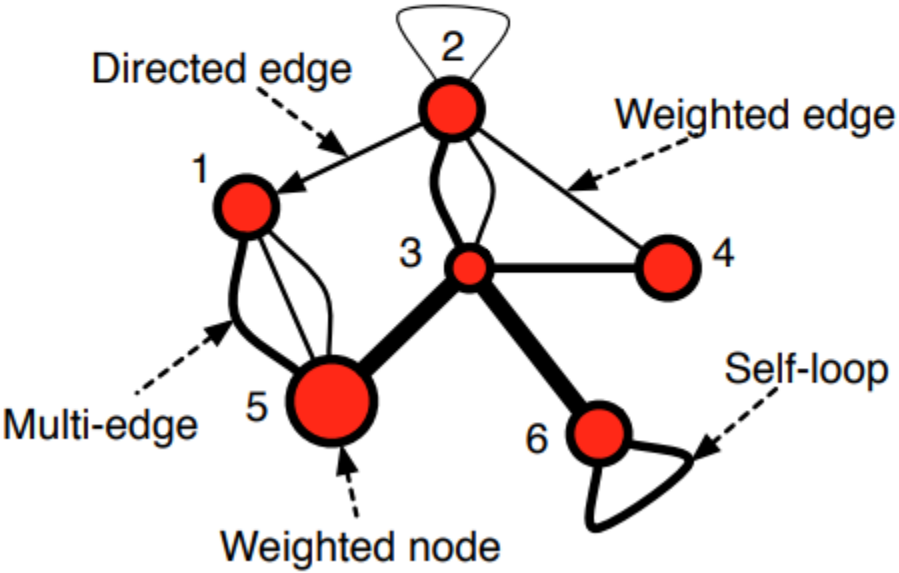


Simple and Not Simple



From Clauset's slides

Example



adjacency matrix

A	1	2	3	4	5	6
1	0	0	0	0	$\{1,1,2\}$	0
2	1	$\frac{1}{2}$	$\{2,1\}$	1	0	0
3	0	$\{2,1\}$	0	2	4	4
4	0	1	2	0	0	0
5	$\{1,1,2\}$	0	4	0	0	0
6	0	0	4	0	0	2

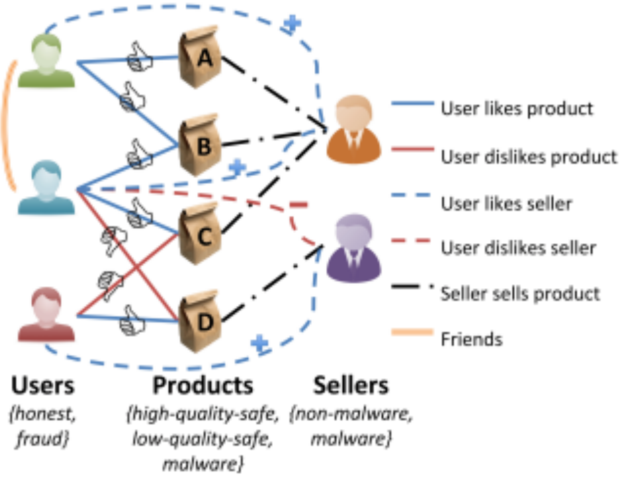
adjacency list

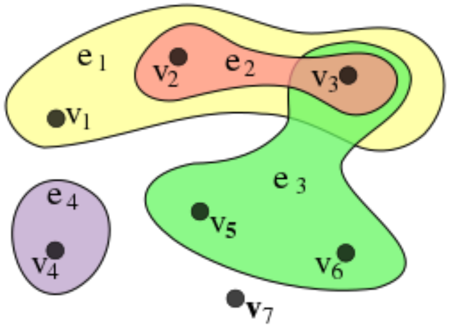
A	
1	$\rightarrow \{(5,1),(5,1),(5,2)\}$
2	$\rightarrow \{(1,1),(2,\frac{1}{2}),(3,2),(3,1),(4,1)\}$
3	$\rightarrow \{(2,2),(2,1),(4,2),(5,4),(6,4)\}$
4	$\rightarrow \{(2,1),(3,2)\}$
5	$\rightarrow \{(1,1),(1,1),(1,2),(3,4)\}$
6	$\rightarrow \{(3,4),(6,2)\}$

From Clauset's slides

Not Simple Graphs

- Multigraph: Multiple edges
  - E.g. followership & friendship
- Heterogeneous Graphs: Different Types
  - E.g. people, places, interest
- Relation between more than two nodes
  - Hypergraphs, E.g. family
- Relationships at different layers
  - Multiplex or multilayer network

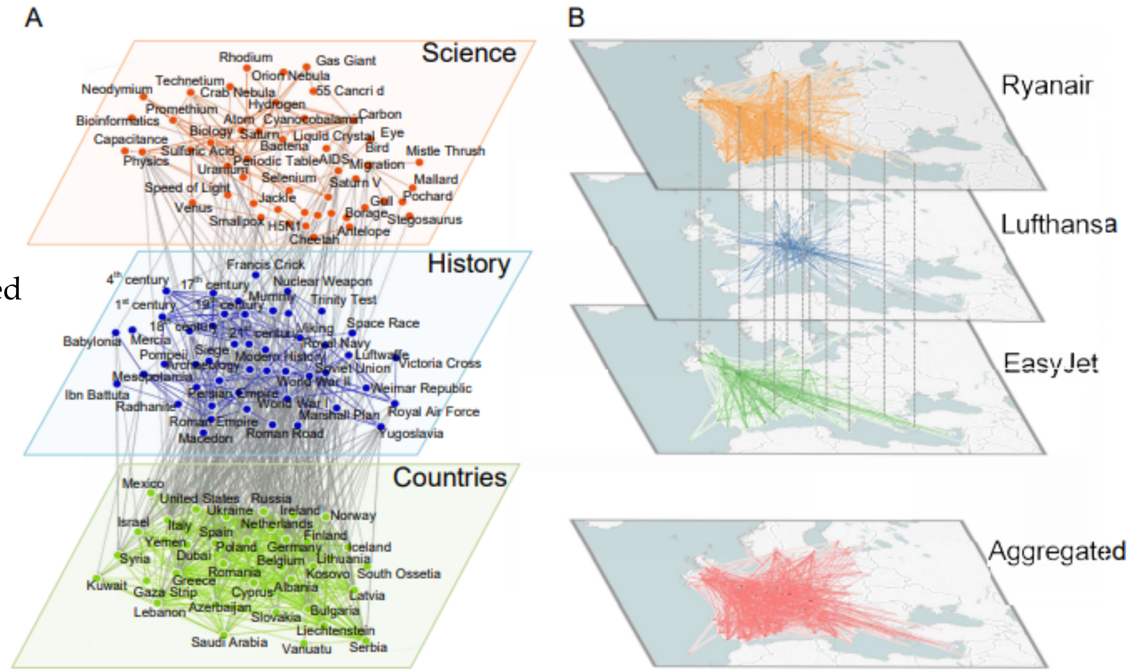




Multilayer Networks

different sets of nodes

E.g. wiki pages layered by subject



Multiplex: same set of nodes

different types of connections

E.g. flights layered by airlines

https://arxiv.org/pdf/1708.07763.pdf

Incidence Matrix

- Adjacency Matrix:
  - $A_{ij} = 1$ if node i is connected to node j & 0 otherwise
- Incidence Matrix:
  - $B_{ik} = 1$ if node i is incident to **edge** k & 0 otherwise
- If a simple graph G has n nodes and m edges what are the dimensions of A & B ?
- How many non-zero elements are in A & B?
- If simple graph, we have 2 ones in each column
  - What is the row marginal of B?
  - $BB^T = A + D$

A

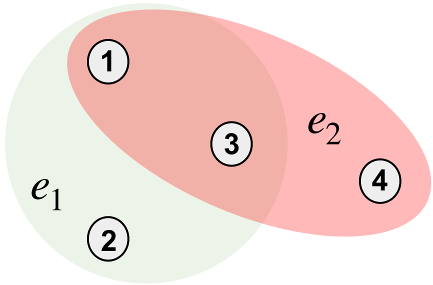
	0	1	2	3	4	5	6	7	8	9	10	11	
0	0	1	1	0	0	0	0	0	0	0	0	1	3
1	1	0	1	1	0	0	0	0	0	0	0	0	3
2	1	1	0	0	0	0	0	0	0	0	0	0	2
3	0	1	0	0	1	1	0	0	0	0	0	0	3
4	0	0	0	1	0	1	1	0	0	0	0	0	3
5	0	0	0	1	1	0	0	0	0	0	0	0	2
6	0	0	0	0	1	0	0	1	1	0	0	0	3
7	0	0	0	0	0	0	1	0	1	0	0	0	2
8	0	0	0	0	0	0	1	1	0	0	1	0	3
9	0	0	0	0	0	0	0	0	0	0	1	1	2
10	0	0	0	0	0	0	0	0	1	1	0	1	3
11	1	0	0	0	0	0	0	0	0	1	1	0	3
	3	3	2	3	3	2	3	2	3	2	3	3	

B

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3
1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	3
2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	3
4	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	3
5	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	2
6	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	3
7	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	2
8	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	3
9	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	2
10	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	3
11	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	3
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	

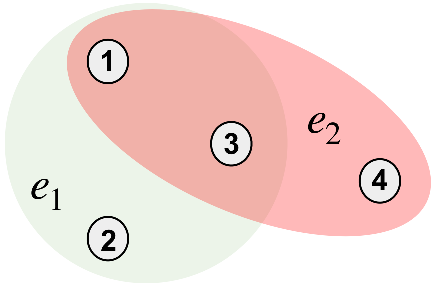
Incidence Matrix

- Adjacency Matrix:
  - $A_{ij} = 1$ if node i is connected to node j & 0 otherwise
- Incidence Matrix:
  - $B_{ik} = 1$ if node i is incident to **edge** k & 0 otherwise
- If a simple graph G has n nodes and m edges what are the dimensions of A & B ?
- How many non-zero elements are in A & B?
- If simple graph, we have 2 ones in each column
  - What is the row marginal of B?
  - $BB^T = A + D$
- Can be used for hypergraphs

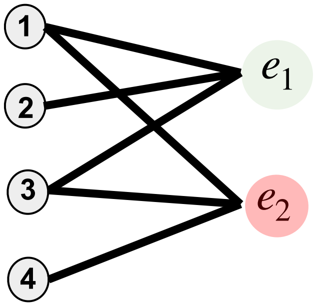


B	$\mathbf{e_1}$	$\mathbf{e_2}$
1	1	1
2	1	0
3	1	1
4	0	1

Incidence Matrix



- Can be used for hypergraphs
  - hyper-edges with more than one node
- Can be used for **bipartite** graphs
  - Two sets of nodes
  - Edges only between them

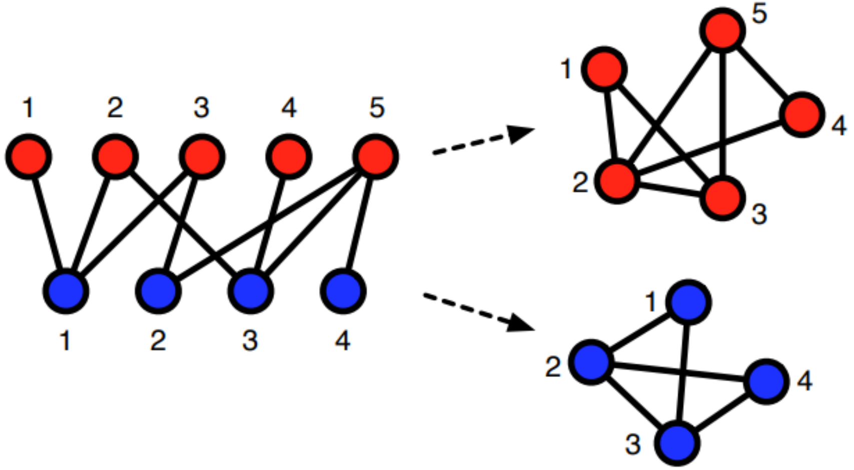


B	$\mathbf{e_1}$	$\mathbf{e_2}$
1	1	1
2	1	0
3	1	1
4	0	1

$$V = A \cup B \mid A \cap B = \varnothing \; and \; \forall (i, j) \in E, ((i \in A) \wedge (j \in B)) \vee ((i \in B) \wedge (j \in A))$$

Bipartite Networks

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



$B^T B$

BB^T

authors & papers	people & locations (checkins)
actors & movies/scenes	metabolites & reactions
musicians & albums	genes & substrings
people & online groups	words & documents
people & corporate boards	plants & pollinators

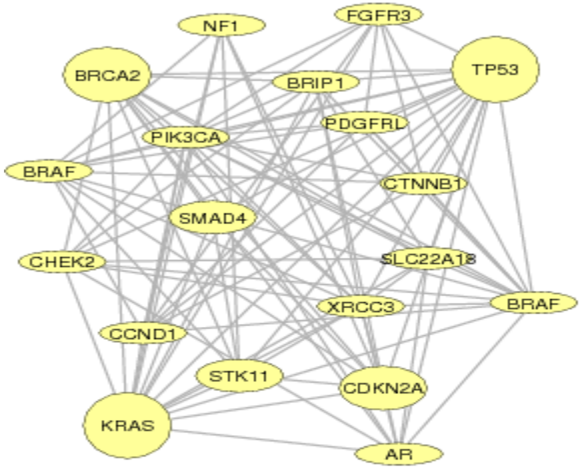
No within edges
& Two possible
one-mode projections

Make the graph to show connections between only one type of node

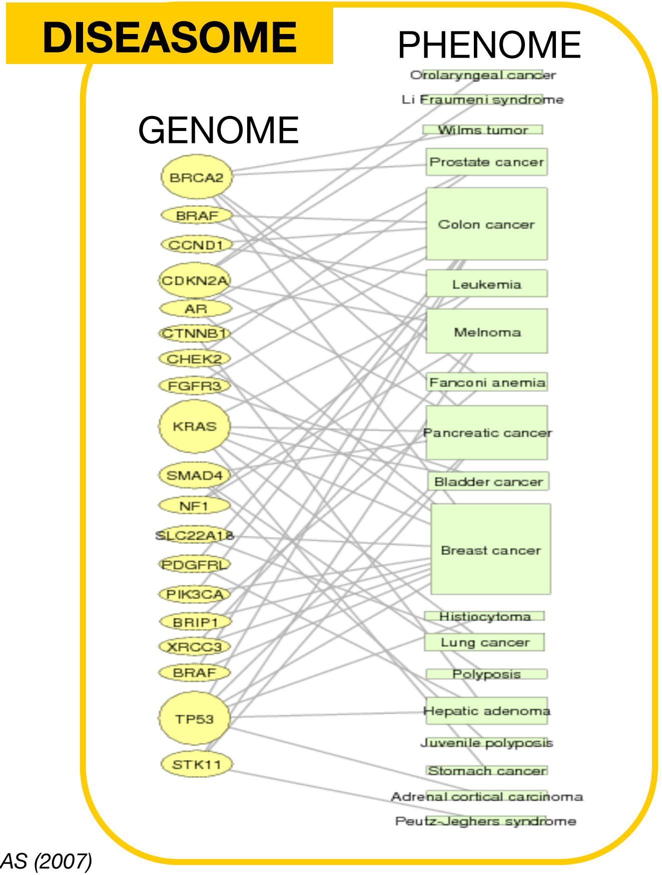
What are the one-mode projection of actors & movies graph?

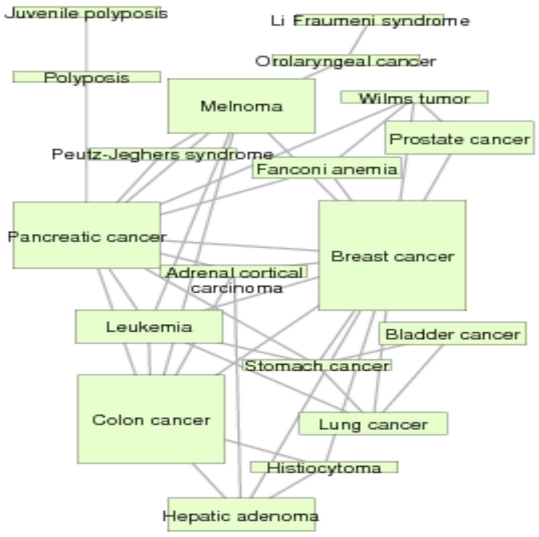
From Clauset's slides

Bipartite Networks example



Gene network



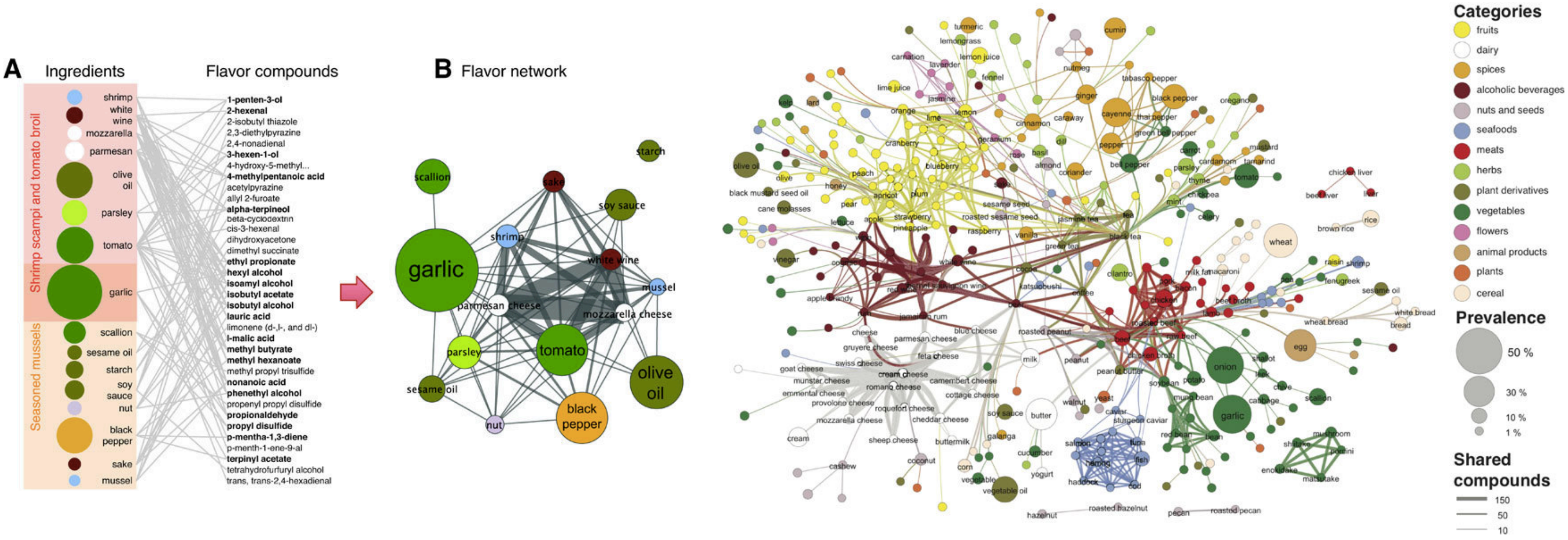


Disease network

Goh, Cusick, Valle, Childs, Vidal & Barabási, PNAS (2007)

[From Barbasi's slides]

Bipartite Networks examples



Ingredient-Flavor Network

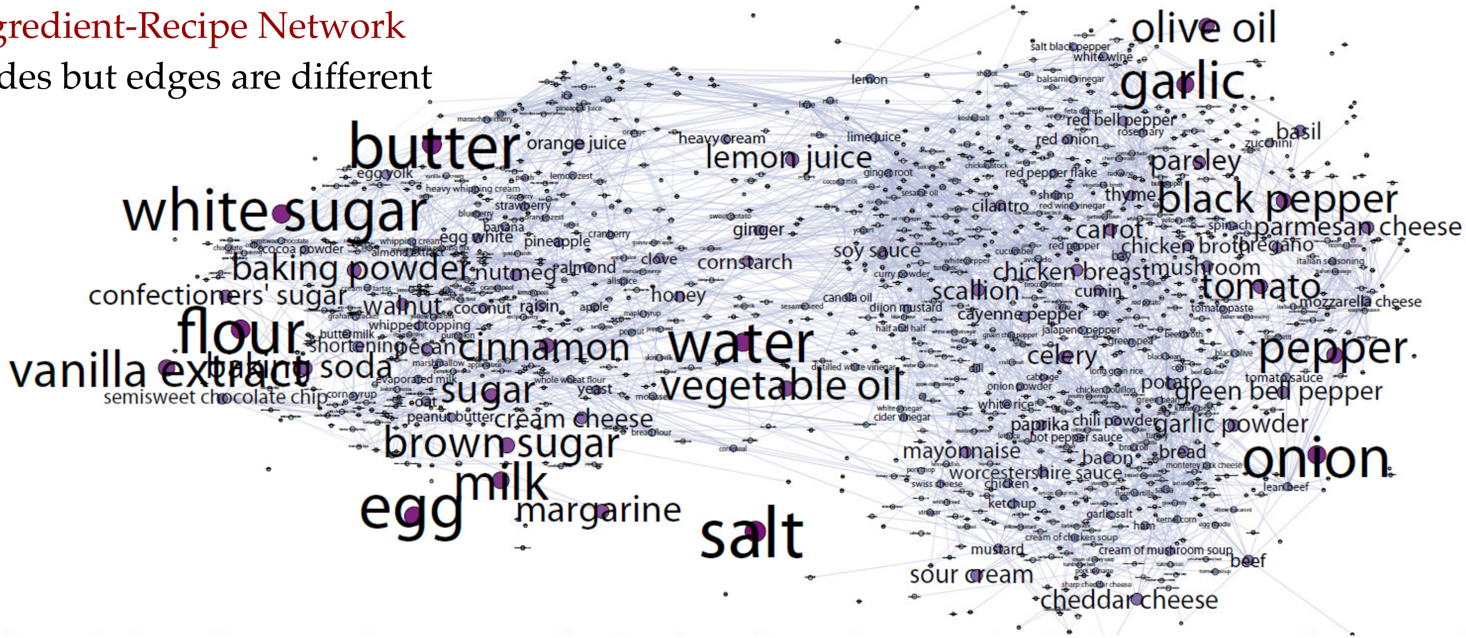
[From Barbasi's slides]

Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási Flavor network and the principles of food pairing , Scientific Reports 196, (2011).

Bipartite Networks example

From Ingredient-Recipe Network

Same nodes but edges are different



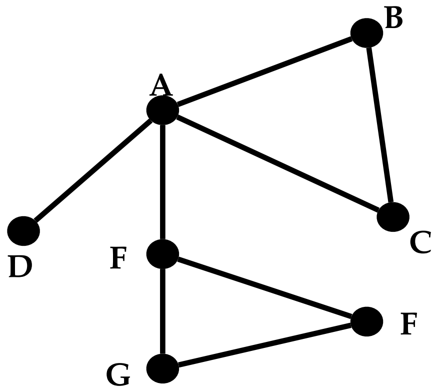
https://arxiv.org/pdf/1111.3919.pdf

https://studentwork.prattsi.org/infovis/labs/visualizing-ingredient-networks/ browse for visualizarions and project ideas

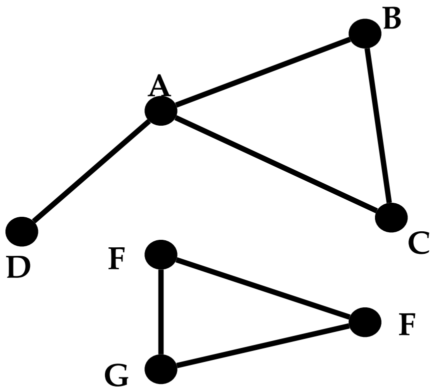
Connectivity

Connected (undirected) graph: any two vertices can be joined by a path

A **disconnected** graph is made up by two or more connected components



Connected



Not Connected

From Barbasi's slides

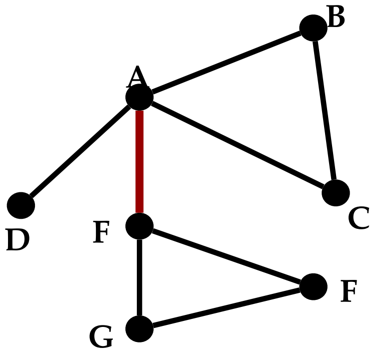
Connectivity: GCC & bridges

Connected (undirected) graph: any two vertices can be joined by a path

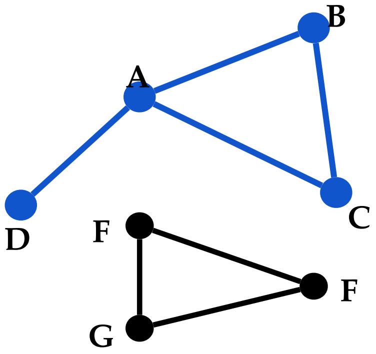
A disconnected graph is made up by two or more connected components

Largest Component is referred to as the **giant connected component (GCC)**

Bridge edges are those that if erased, the graph becomes disconnected



Connected



Not Connected

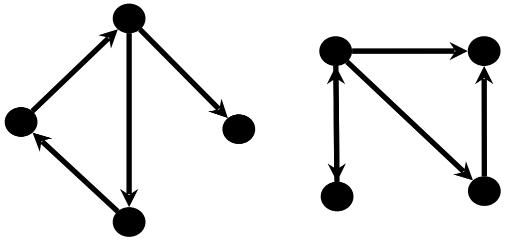
From Barbasi's slides

Connectivity in directed graphs

- **Strongly** connected component
  - has a path from each node to every other node and vice versa
    - e.g. A to B path and B to A path
- **Weakly** connected component
  - it is connected if we disregard the edge directions

How many scc do we have in this example graph?

How many wcc do we have in this example graph?



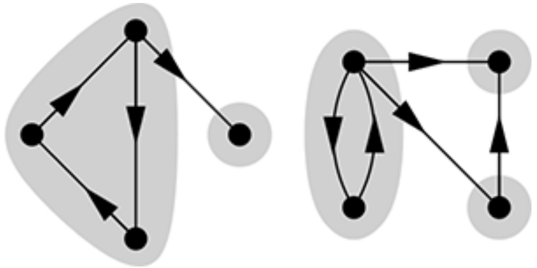
From Barbasi's slides &
From newman's book

Connectivity in directed graphs

- **Strongly** connected component
  - has a path from each node to every other node and vice versa
    - e.g. A to B path and B to A path
- **Weakly** connected component
  - it is connected if we disregard the edge directions

How many scc do we have in this example graph? 5

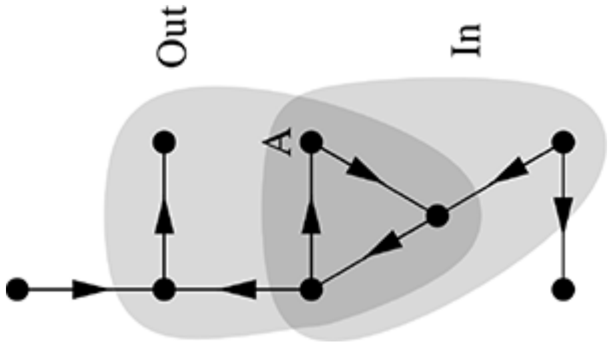
How many wcc do we have in this example graph? 2



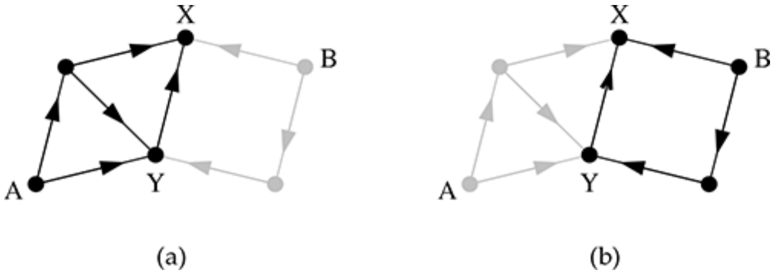
[From Barbasi's slides](#) &
From newman's book

In/Out components

In-component: nodes that can reach the scc
Out-component: nodes that can be reached from the scc



in/out-component of a specific node: set of nodes reachable by directed paths to/from that node

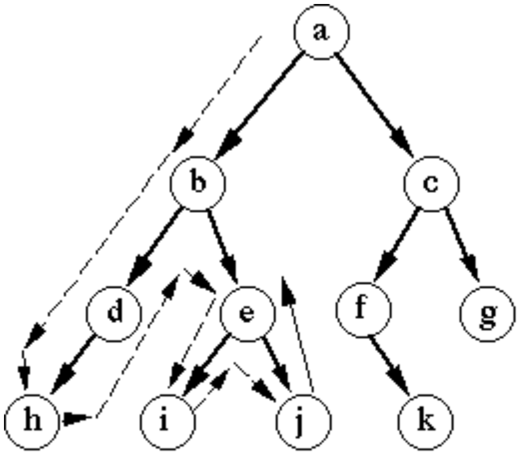


out-component of node A   out-component of node B

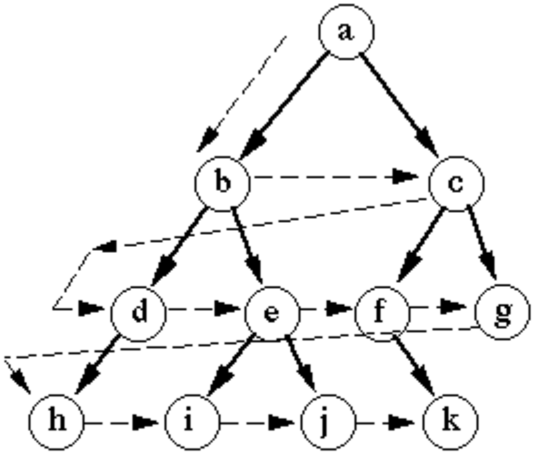
From newman's book

How to check connectivity?

Start from one node, traverse the graph and record the nodes you reach. If the size of this reached set of nodes is equal to all the nodes in the graph, then the graph is connected. If not, this is one component and continue until all nodes have been reached to get all the components.



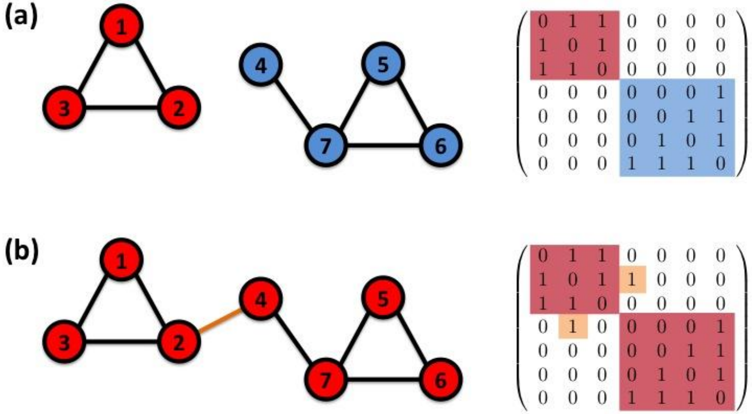
Depth-first search



Breadth-first search

Connectivity & Adjacency Matrix

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



From Barbasi's slides

How can we use this to see if the graph is connected based on A?

Representing Graphs with Laplacian Matrix

marginals of A => degree

$$d_i = \sum_j A_{ij}$$

Laplacian Matrix

$$\in \mathbb{R}^{N \times N}$$

$$L = D - A$$

diagonal degree matrix

Eigenvalues of Graph laplacian tells us about the connectivity of the graph

A

	0	1	2	3	4	5	6	7	8	9	10	11	
0	0	1	1	0	0	0	0	0	0	0	0	1	3
1	1	0	1	1	0	0	0	0	0	0	0	0	3
2	1	1	0	0	0	0	0	0	0	0	0	0	2
3	0	1	0	0	1	1	0	0	0	0	0	0	3
4	0	0	0	1	0	1	1	0	0	0	0	0	3
5	0	0	0	1	1	0	0	0	0	0	0	0	2
6	0	0	0	0	1	0	0	1	1	0	0	0	3
7	0	0	0	0	0	0	1	0	1	0	0	0	2
8	0	0	0	0	0	0	1	1	0	0	1	0	3
9	0	0	0	0	0	0	0	0	0	0	1	1	2
10	0	0	0	0	0	0	0	0	1	1	0	1	3
11	1	0	0	0	0	0	0	0	0	1	1	0	3
	3	3	2	3	3	2	3	2	3	2	3	3	

D: diagonal matrix of degrees

	0	1	2	3	4	5	6	7	8	9	10	11	
0	3	0	0	0	0	0	0	0	0	0	0	0	3
1	0	3	0	0	0	0	0	0	0	0	0	0	3
2	0	0	2	0	0	0	0	0	0	0	0	0	2
3	0	0	0	3	0	0	0	0	0	0	0	0	3
4	0	0	0	0	3	0	0	0	0	0	0	0	3
5	0	0	0	0	0	2	0	0	0	0	0	0	2
6	0	0	0	0	0	0	3	0	0	0	0	0	3
7	0	0	0	0	0	0	0	2	0	0	0	0	2
8	0	0	0	0	0	0	0	0	3	0	0	0	3
9	0	0	0	0	0	0	0	0	0	2	0	0	2
10	0	0	0	0	0	0	0	0	0	0	3	0	3
11	0	0	0	0	0	0	0	0	0	0	0	3	3
	3	3	2	3	3	2	3	2	3	2	3	3	

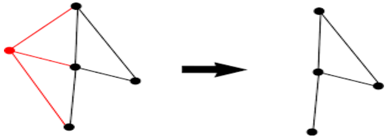
L

	0	1	2	3	4	5	6	7	8	9	10	11	
0	3	−1	−1	0	0	0	0	0	0	0	0	−1	0
1	−1	3	−1	−1	0	0	0	0	0	0	0	0	0
2	−1	−1	2	0	0	0	0	0	0	0	0	0	0
3	0	−1	0	3	−1	−1	0	0	0	0	0	0	0
4	0	0	0	−1	3	−1	−1	0	0	0	0	0	0
5	0	0	0	−1	−1	2	0	0	0	0	0	0	0
6	0	0	0	0	−1	0	3	−1	−1	0	0	0	0
7	0	0	0	0	0	0	−1	2	−1	0	0	0	0
8	0	0	0	0	0	0	−1	−1	3	0	−1	0	0
9	0	0	0	0	0	0	0	0	0	2	−1	−1	0
10	0	0	0	0	0	0	0	0	−1	−1	3	−1	0
11	−1	0	0	0	0	0	0	0	0	−1	−1	3	0
	0	0	0	0	0	0	0	0	0	0	0	0	

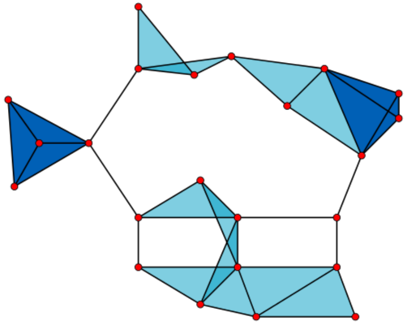
Connectivity & Laplacian Matrix

- smallest eigenvalue of **L** is always zero
- **second-smallest eigenvalue** of **L** is called Algebraic connectivity or Fiedler value and is nonzero only if graph is connected
- **number of zero eigenvalues** of **L** gives the **number of connected components**

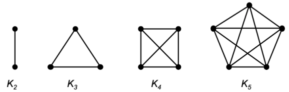
Subgraphs, cliques and k-cores



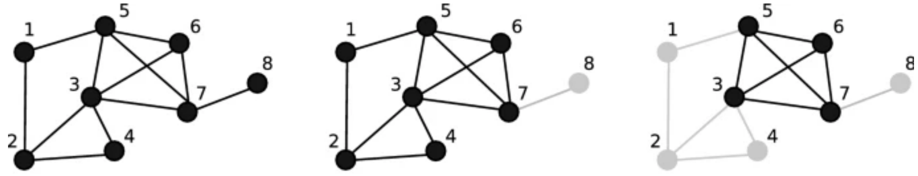
- Induced subgraph:
  - Edges between a subset of nodes in the Graph



- Clique: a.k.a. complete subgraphs



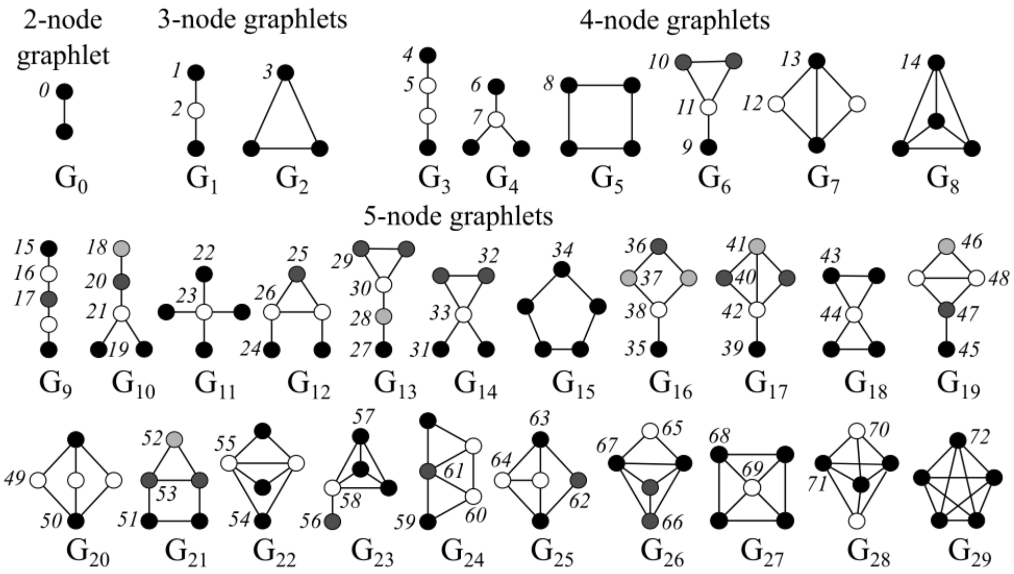
  - A subgraph where every two nodes are adjacent
    - How many [4-vertex cliques] do we see here?
- K-core:
  - Maximal subgraph where degree of each node is at least k



(a) 1-core **(b)** 2-core **(c)** 3-core

Graphlets & Motifs

- Graphlets
    - small, connected, and non-isomorphic induced subgraphs
- Motifs
    - Statistically over- or under-represented graphlets



- There are 73 different graphlets up to 5 nodes