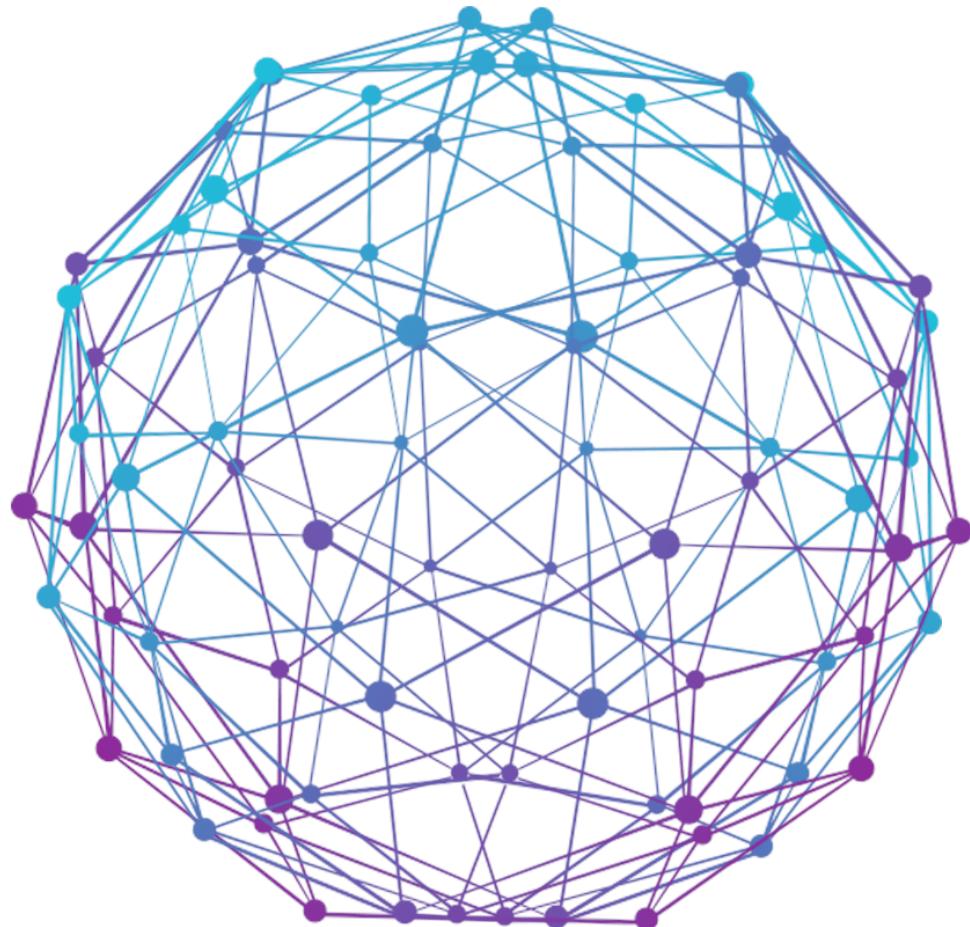


Appunti di Network Science

Analisi e Visualizzazione di Reti Complesse



Michele Lorenzo, Stefano De Rosa, Marco Santimaria

2022/2023

Contents

1	Introduction	5
1.1	Complex Systems	5
1.2	Networks	5
2	Graph theory	6
2.1	Definition	6
2.1.1	Neighbors	6
2.1.2	Degree	6
2.1.3	Average degree	7
2.1.4	Strength	7
2.1.5	Density and Sparsity	7
2.1.6	Paths	8
2.1.7	Shortest paths	8
2.2	Network types	9
2.2.1	Subnetwork	9
2.2.2	Complete	9
2.2.3	Clique	9
2.2.4	Bipartite graph (Dissimilar Network)	9
2.2.5	Multilayer	10
2.2.6	Temporal networks	10
2.2.7	Hypergraph	10
2.3	Network Representation	11
2.3.1	Adjacency Matrix	11
2.3.2	Adjacency Lists	11
2.3.3	Edge Lists (with weights)	11
2.4	Connectedness and components	11
2.5	Small world phenomenon	12
2.5.1	Short path	12
2.5.2	Clustering coefficient	12
2.6	The role of strong and weak ties in the network	12
2.6.1	Triadic closure	12
2.6.2	Bridges	13
2.6.3	Local bridges	13
2.6.4	Different level of strength	14
2.6.5	Strong triadic closure	14
2.6.6	Local bridges and weak ties	14
2.6.7	Neighborhood overlap	14
2.6.8	Embeddedness	15
2.6.9	Structural holes	15
2.7	Homophily	16
2.7.1	Definition of Homophily	16
2.7.2	Measuring Homophily	16
2.8	Assortativity	17
2.8.1	Modularity	17
2.8.2	Assortativity coefficient	18
2.8.3	Degree Assortativity	18
2.8.4	Assortativity coefficient by degree	19
2.8.5	Measuring Assortativity by neighbors	19
2.8.6	Mechanisms Underlying Homophily: Selection and Social Influence	20
2.9	Affiliation	20
2.9.1	Affiliation Networks	20
2.9.2	Co-Evolution of Social and Affiliation Networks	21
2.10	Tracking Link Formation in On-Line Data	23
2.11	Centrality Measures	23
2.11.1	Degree Centrality	23
2.11.2	Closeness Centrality	23
2.11.3	Betweenness Centrality	24

2.12	Centrality distributions	24
2.13	Ultra Small world problem and the friendship paradox	25
2.14	Robustness	26
3	Network models	26
3.1	Features of real networks	26
3.2	Erdős-Renyi random graph	27
3.2.1	Density	27
3.2.2	Degree distribution	27
3.2.3	Short paths	28
3.2.4	Clustering coefficient	28
3.3	The configuration model	28
3.3.1	Degree Sequences	29
3.4	Small world networks	30
3.4.1	The Watts-Strogatz model	30
3.5	Random networks with communities	30
3.6	Network model comparison	30
3.7	Power-Law	31
3.8	Barabasi-Albert model	31
3.8.1	Network growth	31
3.8.2	Preferential attachment	32
3.8.3	Extension of Barabasi-Albert: Attractiveness Model	32
3.8.4	Extension of Barabasi-Albert: Fitness Model	33
3.8.5	Extension of Barabasi-Albert: Random Walk Model	33
3.8.6	Extension of Barabasi-Albert: Rank Model	34
4	Epidemics	35
4.1	Basic Reproduction Number	35
4.2	SI Systems	35
4.3	SIS Systems	36
4.4	SIR System	38
4.5	Conclusions on systems	38
4.6	Network epidemics	39
4.7	Degree block approximation	39
4.8	Susceptible-Infected (SI) Model on a Network	39
4.9	SIS Model and the Vanishing Epidemic Threshold	40
4.9.1	SIS spreading in random networks	40
4.10	Temporal vs Aggregated Networks	41
4.11	Immunization	41
4.11.1	Random Immunization	41
4.11.2	Selective Immunization	41
5	Communities	42
5.1	Definition of community	42
5.1.1	Community detection	42
5.2	Cost function optimization: min-cut problem	42
5.2.1	Cut function	42
5.2.2	The ratio cut	42
5.2.3	Modularity optimization	42
5.3	Communities in structure-less graphs	43
5.4	Solve the NP hard problem of identifying communities	43
5.5	Bayesian approach: DCSBM (Degree Corrected Stochastic Block Model)	44
5.6	Data Clustering	44
5.6.1	Hierarchical Clustering	44
5.6.2	K-means	45
5.6.3	Spectral clustering	46
5.6.4	Spectral clustering with the graph Laplacian	46
5.6.5	The random walk Laplacian	47
5.6.6	Spectral clustering and the number of communities	47

5.6.7	Sparsity in spectral clustering	48
5.6.8	Node embeddings	48
5.6.9	Word2Vec	49
6	Page Rank	49
6.1	Link Analysis using Hubs and Authorities	50
6.1.1	Voting by In-Links	50
6.1.2	Hubs and Authorities	50
6.2	Page rank algorithm	51
7	Information cascading	52
7.1	Diffusion in Network	52
7.2	Modeling diffusion through networks	52
7.3	Cascading behavior	53
7.4	Cascades and Clusters	53
7.5	Diffusion, Thresholds, and the Role of Weak Ties	54
7.6	Extension of the basic cascade model	54
8	Game Theory	54
8.1	Underlying Assumptions	54
8.2	Best Responses and Dominant Strategies	55
8.3	Nash Equilibrium	55
8.4	Mixed Strategies	55
8.4.1	Nash Equilibrium with Mixed Strategies	56
8.5	Pareto-Optimality and Social Optimality	57
8.5.1	Pareto-Optimality	57
8.5.2	Social Optimality	57
9	Traffic networks	57
9.1	Traffic at Equilibrium	57
9.2	Braess's Paradox	58

1 Introduction

Tools used:

- Network Science
 - 1. NetworkX
 - 2. Graph tool (best)
 - 3. iGraph
- Data visualization
 - 1. Gephi (does not work well on big data)
 - 2. Net Logo

1.1 Complex Systems

A **complex system** is a **system** that is made up of a **number of interacting elements** (agents), whose collective behavior cannot be easily predicted by looking at the behavior of each individual component in isolation and also could leverage a pattern (i.e. “flock of birds”).

Complex systems are characterized by **emergent** properties that arise from the interactions between the components, and by nonlinear relationships. They also give rise to emergent collective phenomena (i.e. pandemic “Sick agents transmit virus to the others”), in other words, is a phenomenon that afflicts the majority of the agents. Be aware that complex is not equal to complicated.

Warning: Heterogeneous = parts or things that are very different from each other

1.2 Networks

We can think of the network as an important tool that helps us to represent interactions between agents of a complex system (i.e. railway). All networks have in common the fact that they all have some **agents** interacting with each other.

Note: Complex systems doesn't imply Complex networks, in some cases, we might have complex systems that could be represented by simple networks in a structural sense (Simple networks are described by few characteristics).

In cases where we have some complex networks, we can still find regularities in them:

- Existence of some **individuals** that are more “**central**” (a lot of connections) than others (peripherals). A central node is a particular node that, if removed, will cause a change in the network structure.
- Existence of some **hierarchical structures**
- Homophily: tendency of individuals to link with similar ones. The existence of this phenomenon leads to partitioning and also to an echo-chambers situation, thus generating clusters (also known as partitions).

Warning: there aren't trivial linear relationships but **interplay**

Important: Some nodes are very central, and the removal of some of them will change a lot of the dynamics in the network.

In general, we can also simplify the structural complexity of a network using a Gaussian hypothesis which simplifies the structure by considering a network with few nodes with many connections and the others with few connections.

Warning: Same data doesn't imply the same networks.

The node's actions affect all the other nodes. These actions could be motivated by rational behavior, in which case they could lead to a strategy.

Important: Actions are evaluated at a network level.

We can observe interactions at different levels:

- **Individual level**
- **Population level**

How we define the interactions makes a huge difference in the network structure. For instance, using different definitions of interactions we could obtain different clusters.

There are three core elements in network science:

- **System:** a number of interacting elements (ex: Italian railway system)
- **Network:** representation of the system
- **Graph:** math representation of the network

2 Graph theory

2.1 Definition

Math element composed by links and nodes: $G = (N, L)$

- Set of nodes (or vertices): $N = n_1, \dots, n_k$
- Set of links (or edges, arcs): $L = \{(i, j) | i, j \in N\}$
- Weights: (optional) define the strength of the interactions: $w_{i,j} = (i, j, w)$

Warning: If the graph is directed, the order is important.

Graphs can be classified based on different features:

Interaction direction	Weight presence
Directed (one-way directions)	Weighted
Undirected (both directions)	Unweighted

Note: In this course, we will use mostly undirected and unweighted graphs.

2.1.1 Neighbors

Neighbors of a node i can be represented as a set of nodes that can be reached with one interaction from the node i , in the case of directed graphs. Those are the closest nodes given a node (it means that a direct connection with given node exists).

Important: on directed graphs, two close nodes might be neighbor in a direction but not on the other.

2.1.2 Degree

Given a node i , the degree of that node is the size of his neighbor set and is indicated with: $k_i = |N_i|$ where N_i is the set of neighbors of the node i .

There are also other types of sets related to the degree of a node:

- P_i (predecessor set): set of nodes having an outgoing link with i
- S_i (successor set): set of nodes having an incoming link with i

Note: The degree of sets S and P are equal on undirected graphs but not necessarily on a directed graph.

Note: Usually the bigger the degree of a node is, the more important the node is.

A **singleton** is a node that has a degree of zero, ie. it has no links with other nodes in the graph, more formally $k_i = 0$. Singletons are not considered important in this course as they do not incur in any interactions with other nodes.

If the graph is directed, two other measures also exist: **in-degree** and **out-degree** which respectively measure the sum of edges “entering”, and “leaving”, the given node (which is the same as saying that it measures the size of the predecessor set and the successor set).

2.1.3 Average degree

For undirected network, we can define a measure called “Average degree”, calculated as follows:

$$\langle k \rangle = \frac{\sum_{i \in N} k_i}{|N|} = \frac{2 \cdot |L|}{|N|} \quad (\text{undirected case})$$

where k_i is the degree of the node i . The average degree is also connected to the density [1] with the following equation:

$$d = \frac{\langle k \rangle}{|N| - 1} = \frac{\langle k \rangle}{k_{max}}$$

Where k_{max} is the maximum possible degree of a network. In other words is the maximum number of possible neighbors. If we take a larger network, doesn't imply that the density increases.

2.1.4 Strength

In an indirect graph, the strength of a given node, is the sum of all the weights of the links that connect that node. More formally, it can be expressed with the following equation:

$$s_i = \sum_{j \in N_i} w_{i,j}$$

There exists also two other weight measure, that can also be applied to directed graphs:

- **In-strength:** sum of all the weights of links that arrive to a given node:

$$s_i^{in} = \sum_{j \in P_i} w_{j,i}$$

- **Out-strength:** sum of all the weights that start from a given node:

$$s_i^{out} = \sum_{j \in S_i} w_{i,j}$$

We can define the degree of a node i as $k_i = k_i^{in} + k_i^{out}$ where $k_i^{in} = |P_i|$ and $k_i^{out} = |S_i|$

2.1.5 Density and Sparsity

The network size is defined by the size of the node sets. It is indicated with $|N|$, or simply N . The same thing can be done for the number of links $|L|$, or more simply, L .

The maximum number of links, is defined by the following equations:

$$L_{max} = \binom{|N|}{2} = \frac{|N|(|N| - 1)}{2} \approx O(|N|^2) \quad (\text{undirected graph})$$

$$L_{max} = |N|(|N| - 1) \approx O(|N|^2) \quad (\text{directed graph})$$

Intuitively each node can connect to $|N| - 1$ other nodes, and there are $|N|$ of them. However with undirected graphs, that would count each pair twice, so we divide by two.

Having defined the maximum amount of links available in a graph, it is possible to define the measure of density and sparsity:

- Density:

$$d = \frac{|L|}{L_{max}} = \frac{2 \cdot |L|}{|N| \cdot (|N| - 1)} \approx O(\log(|N|)) \quad (\text{undirected graph}) \quad (1)$$

$$d = \frac{|L|}{L_{max}} = \frac{|L|}{|N| \cdot (|N| - 1)} \approx O(\log(|N|)) \quad (\text{directed graph}) \quad (2)$$

- Sparsity: There are two definitions:

1. A practical one (always used): if $d \ll 1$ then the network is sparse
2. A mathematical one (less useful): If L is in the order of N , it is sparse, otherwise it is dense

2.1.6 Paths

By definition a path is a sequence of nodes (n_1, n_2, \dots, n_k) directly connected with a link.

Whenever there is a node that repeats in a path, there is a cycle. If a path has no cycles in itself, we call it **simple path**.

If we want to explore fast the graph, simple paths are important.

2.1.7 Shortest paths

The shortest path is defined as the minimal path between two nodes. The distance between a pair of nodes is calculated as the length of the shortest path.

Note: The shortest is the path, the stronger the interaction can be.

Sometimes we are interested in calculating the highest distance between a pair of nodes i, j in a graph, also called diameter, defined as the largest shortest path:

$$\text{diameter} = l_{max} = \max_{i,j} l_{i,j}$$

Warning: l , in this context, does not indicate an edge (arc) but a path between two nodes.

If we consider all the pairs of nodes in the network and we calculate their distance, we can compute the **Average Path Length** $\langle l \rangle$.

$$\langle l \rangle = \frac{\sum_{ij} l_{ij}}{\binom{|N|}{2}} = \frac{2 \cdot \sum_{ij} l_{ij}}{|N|(|N| - 1)} \quad (\text{undirected graph})$$

$$\langle l \rangle = \frac{\sum_{ij} l_{ij}}{|N|(|N| - 1)} \quad (\text{directed graph})$$

Note: Diameter and shortest path are not defined on unconnected graphs

Important: In most cases the diameter of a network is around 5. Thus, if the network is small, we have a bigger density (because "people" have more chance to know each other) but also a smaller diameter.

Important: Complex networks generally grow exponentially. This means that $l \sim \frac{\log|N|}{\log k}$ where k is a fixed number of neighbors for each node.

2.2 Network types

There exists several types of networks:

2.2.1 Subnetwork

Subset of the network

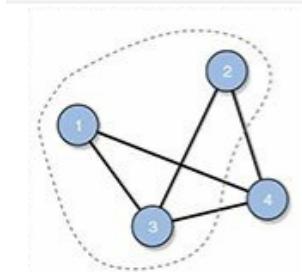


Figure 1: A subnetwork network

2.2.2 Complete

Network where all the nodes have a direct connection between each other.

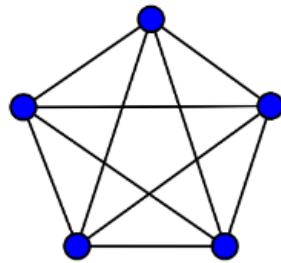


Figure 2: A Complete network

2.2.3 Clique

A complete subnetwork.

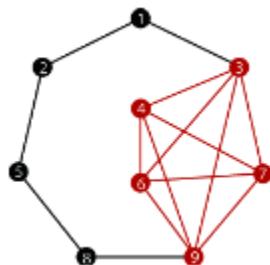


Figure 3: A clique network

2.2.4 Bipartite graph (Dissimilar Network)

There are only two types of nodes and connections only happen between opposite type nodes.

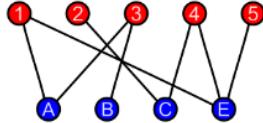


Figure 4: A bipartite graph example.

2.2.5 Multilayer

Contains different layers, where each layer represent a dimension of interaction (i.e. Language, Music).
Types of link:

- **Intra layer links:** link nodes in the same layer;
- **Inter layer links:** link nodes between different layers.

Important: If each layer is built upon the same set of nodes, the network is also called multiplex.

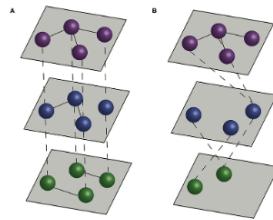


Figure 5: Two multilayer network example

2.2.6 Temporal networks

A multiplex network in which the layer (temporal snapshots) represents links at different times. It is a dynamical process on a static network. Links are defined by a triplets (i, j, t) where t is the time.

Important: Edges (Links) can exist on a given time and disappear afterwards.

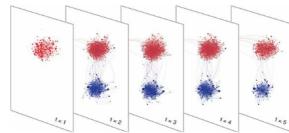


Figure 6: A temporal network example

2.2.7 Hypergraph

Generalization of a graph (Note: Graph is a hypergraph with size 2) where interactions might occur between multiple nodes simultaneously. It is represented by a sequence of edges of size > 2 called also as Hyperedges. For example $L = \{(1, 2), (2, 4, 6), (1, 5, 4, 2, 9), \dots\}$

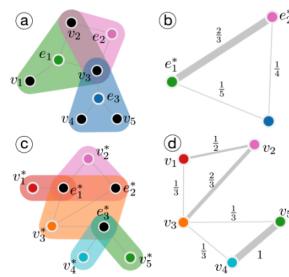


Figure 7: A Hypergraph example.

2.3 Network Representation

2.3.1 Adjacency Matrix

Square matrix used to represent a **dense network** with rows and columns labeled by network nodes. The elements of the matrix indicate whether or not pairs of nodes are adjacent in the graph (1: adjacent, 0: not).

Note: If you prefer (1: exist edge, 0: not exist edge).

Note: When we have weighted graphs, the matrix elements represent the weight of the edge.

The matrix is defined as follows:

$$\begin{aligned} A &\in [0, 1]^{|N| \times |N|} \\ A_{i,j} &= \mathbb{1}_{i,j} \in L \\ W_{i,j} &= (\mathbb{1}_{i,j} \in L) \cdot w_{i,j} \end{aligned}$$

Where $\mathbb{1}$ is an indicator, is a variable that can either be assigned to 0 or 1. Equally, the matrix can be defined as:

$$a_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in L \\ 0 & \text{otherwise} \end{cases} \quad w_{i,j} = \begin{cases} w & \text{if } (i,j) \in L \\ 0 & \text{if } (i,j) \notin L \end{cases}$$

Note: If the matrix is undirected the adjacency matrix is symmetric.

In undirected graph we can calculate the degree of a node by summing the elements of that row:

$$k_i = \sum_{j \in |N|} A_{i,j} \quad \vec{k} = A \vec{1}_n$$

The same process can be done for directed graphs, but we need to calculate the in and out degree instead, which is done as follows:

$$k_i^{out} = \sum_j A_{i,j} \quad k_i^{in} = \sum_j A_{j,i}$$

2.3.2 Adjacency Lists

Since we are working with **sparse matrices** in real scenarios, we can optimize the matrix representation that would otherwise take up $O(|N|^2)$ space by using, for example, the adjacency lists $O(|N|)$.

The space required to store the matrix in the directed graph is: $L + N$.

In the case of undirected graph, instead, the space required is: $2L + N$, because we have to store both (a,b) and (b,a).

By storing only one edge and specifying that the graph is undirected, you can optimize the amount of memory used to store the matrix.

2.3.3 Edge Lists (with weights)

A more popular way to encode the graph is the edge list. The big advantage of this representation is that can be also generalized to weighted and temporal graphs.

2.4 Connectedness and components

A graph is connected if there is a path between any pair of nodes, otherwise it is called disconnected.

A connected component is a connected sub-graph.

The largest connected component is called giant component and is usually the most important one where studies are conducted (because the contagiousness of a virus is higher in big networks).

Note: Singleton are by definition disconnected.

In the directed case, things are a bit more complicated because we have to pay attention to link directions when determining whether a node can be reached from another. We can of course ignore the link directions

and treat the links as if they were undirected. In this case we refer to the components as **weakly connected**. In a **strongly connected** component, there is at least one directed path between every pair of nodes, in both directions.

The **in-component** of a strongly connected component S is the set of nodes from which one can reach S , but that cannot be reached from S directly. The **out-component** of a strongly connected component S is the set of nodes that can be reached from S , but from which one cannot reach S directly.

2.5 Small world phenomenon

In human interactions, whenever we have three nodes, A, B, C and two links $(A, B), (B, C)$ it is extremely likely that also the link (C, A) exists. This forms a triangle (which is the smallest possible clique). See the Milgram experiment ¹

2.5.1 Short path

We say that the average path length is short when it grows very slowly with the size of the network:

$$\langle l \rangle \approx \log |N|$$

2.5.2 Clustering coefficient

The clustering coefficient, is a measure of the degree to which nodes in a graph tends to cluster together, this value is in range $[0, 1]$.

We define the clustering coefficient of node i as:

$$C_i = \frac{\tau(i)}{\tau_{max}(i)} = \frac{\tau(i)}{\binom{k_i}{2}} = \frac{2\tau(i)}{k_i(k_i - 1)} \quad (\text{undirected case})$$

where $\tau(i)$ is the number of triangles involving i and $\tau_{max}(i)$ represents the pairs of the node's neighbors that are connected to each other.

The network clustering coefficient is defined as the average of the clustering coefficients:

$$C = \frac{1}{|N|} \sum_{i=1}^{|N|} C_i$$

Note: When we have random networks, the clustering coefficient is typically low.

These triangles could be generated, for instance, from a group of people that agree on something, because they tend to interact with each other. So affinity is weakly (it's only likely to be) transitive.

2.6 The role of strong and weak ties in the network

Edges describe local interactions between nodes. A few changes in the edges can make changes of the overall behavior.

2.6.1 Triadic closure

We can define informally a **Triadic closure** as follows:

if (A, B) are friends, (B, C) also are friends, it's very likely that at some point in the future, (A, C) will also become friends.

Triadic closures are the reason why the **Clustering coefficient** exists: interpreted as the probability that two randomly selected friends of node A are friends between each other.

In general, an higher clustering coefficient (a lot of triangles) means that we have triadic closures in the network. For instance, social networks have an high clustering coefficient, whereas randomly generated networks have a low clustering coefficient, more formally: $CC(G) \gg CC(G_R)$.

Note: To analyze graphs, a common technique is to use a *randomized reference model*

In social networks, the triadic closures are explained by the following reasons:

¹Milgram experiment on Wikipedia

- **Opportunity:** if A spends time with both B and C , there is an increased chance that they will all end up knowing each other and become potentially friends at some point in time.
- **Trust** if B, C are friends with A , then they have a basis of trusting each other, in comparison to a random couple of people (i.e references in the Academic world)
- **Incentive** If A is friend with B and C , then it can become a source of stress if B and C are not friends with each other.

2.6.2 Bridges

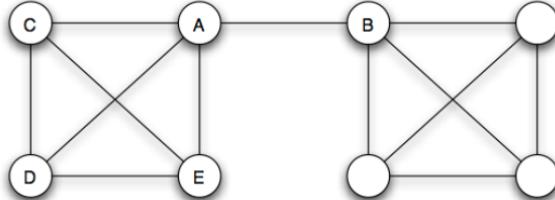


Figure 8: A Hypergraph example.

We call a bridge, simply an edge that connects two nodes A and B living in a two unconnected components in a graph, that, if deleted would cause A and B to lie in two different components.

In other words, the **bridge edge**, is the edge that links two different components. Bridges are extremely rare in real social networks.

2.6.3 Local bridges

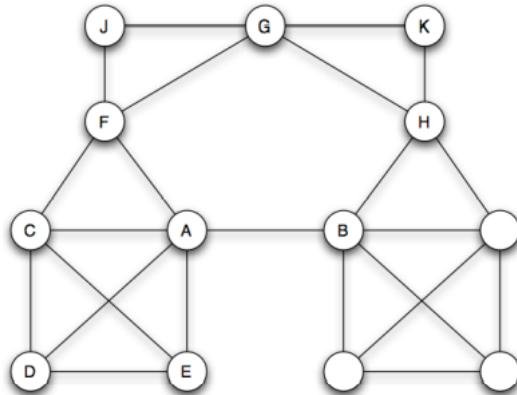


Figure 9: A Hypergraph example.

We say that an edge joining two nodes A and B in a graph is a local bridge if its endpoints A and B have no friends in common. In other words, if we delete the edge between the nodes, the distance between node A and B would increase with a size > 2 . A bridge is statistically very rare in classical scenarios. Triadic closures are mainly related to strong connections (ties).

Span of local bridges: we define the span of a local bridge (between node A , and B), the distance between the nodes if we delete the link that directly connects them (the local bridge). Thus, in figure 9, the $A - B$ edge is a local bridge with span four.

Note: The notion of triadic closure and local bridges are opposites: an edge is a local bridge whenever it does not form a side with any of the triangle in the graph.

Local bridges perform the same role as bridges in the networks: they allow a part of the network to connect to another part with a shorter path, than it would take to reach if the local bridge were not present.

2.6.4 Different level of strength

We generally avoid to define the concept of strength in a network, but we provide a general classification:

- **Strong ties:** stronger ties according to the notion that we give to the weights of the network
- **Weak ties:** weaker ties

2.6.5 Strong triadic closure

We say that a node A violates the Strong triadic Closure Property if it has strong ties to two other nodes B and C , and there is no edge at all (either a strong or weak tie) between B and C . Thus, we say that a node A satisfies the Strong Triadic Closure Property if it does not violate it.

Note: The Strong Triadic Closure Property is too extreme for us to expect it hold across all nodes of a large social network. But it is a useful step as an abstraction to reality, making it possible to reason further about the structural consequences of strong and weak ties.

2.6.6 Local bridges and weak ties

If a node A in a network satisfies the Strong triadic Closure Property and is involved in at least two strong ties, then any local bridge it is involved in must be a weak tie.

This phrase can be rephrased less formally as: "a local bridge between nodes A and B tends to be a weak tie because if it weren't, triadic closure would tend to produce short-cuts to A and B that would eliminate its role as a local bridge"

A general notation for the definition of weak ties and local bridge is:

- an edge is either a strong tie or a weak tie
- $\forall l \in L : w(e) \in [0, +\infty]$ is either a local bridge or not

Note: the strength can be converted to a numerical value on the basis of some sort of meaning we give to the strength.

Note: surprising fact that life transitions, such as a new jobs, are often rooted in contact with distant acquaintances.

The argument is that these are the social ties that connect us to new sources of information and new opportunities, and their conceptual "span" in the social network (the local bridge property) is directly related to their weakness as social ties. Weak ties have a dual role as weak connections but also valuable conduits to hard-to-reach parts of the network

2.6.7 Neighborhood overlap

We define the **Neighborhood overlap** of an edge connecting A and B to the ratio:

$$O_{AB} = \frac{|N(A) \cap N(B)|}{|N(A) \cup N(B) \setminus \{A, B\}|}$$

This ratio tell us how close an edge is to be a local bridge. to be precise:

$$O_{A,B} = 0 \iff (A, B) \text{ is a local bridge}$$

hence we can think of edges with very small neighborhood overlap as being "almost" local bridges.

The bigger is the overlap, the less important is the node's role, and the smaller is the overlap, the more important is the node's role.

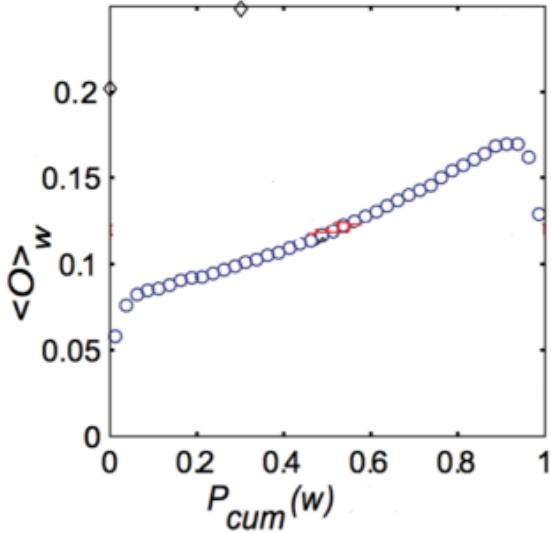


Figure 10: neighborhood overlap of edges as a function of their percentile in the sorted order of all edges by tie strength

We can observe that overlap increases with increasing tie strength. Thus, the strength of weak ties predicts that neighborhood overlap should grow as tie strength grows.

Important: if on a network, we firstly delete the strongest ties, the giant component shrinks steadily, whereas if we begin from the weakest ties, the giant component falls apart almost instantly. **This is a proof that weak ties are crucial in keeping the network connected.** The percolation is a phenomena where we have to find how many edges we need to remove before the giant component falls apart ($|\text{Giant component}| \ll |N|$). To accomplish this, we randomly select the edges and remove them.

A fundamental concept for social networks is the *passive engagement* which means to feed social information to a person without a real communication. To put it more frankly: the likes, shares, visuals and etc.. is the new way of communicating. Since we usually use informal measures we can't really grasp the context and the true meaning of the real relationships between the nodes. Unless we have the complete control of the network we can't really get much from the analytics that we run.

2.6.8 Embeddedness

If node A , is the center of a single tightly-knit group, its neighbors have been subject to considerable triadic closure. Hence A has a high clustering coefficient.

Note: we define the **embeddedness** of an edge in a network to be the number of common neighbors the two endpoints have.

The **embeddedness** is the numerator in the equation that defines the neighborhood overlap, more formally:

$$e(A, B) = |N(A) \cap N(B)|$$

Local bridges are the edges whose embeddedness is equal to zero.

Note: We can use the embeddedness to distinguish different roles of the nodes inside the network.

Note: Higher is the embeddedness, bigger is the control that we have on the local structure.

2.6.9 Structural holes

The part of the graph that is missing. They include connections that are missing to put things together. In society these nodes have a lot of advantages and have been observed to rise against the sea of mediocre employees...

2.7 Homophily

Homophily is a concept in sociology describing the tendency of individuals to associate and bond with similar others. This can be translated in a popular saying that is that *birds of a feather flock together*, which in Italian is translated in *dimmi con chi vai e ti diro chi sei*. In other words, the basic idea of homophily is that is very likely for me to be connected to people that are similar to me.

The surrounding context of homophily requires us to focus on:

- **structural properties of networks:** how nodes are linked to each other.
- **link formation processes:** how links are formed, and this allows us to introduce predictive models that allow us to predict which links are going to be formed before others.
- **personal characteristics:** we link structural properties to personal characteristics (like the bridge and the weak ties) and with this we can get at the end **similarities between individuals**.
- **surrounding contexts:** his are the factors that exist outside the nodes and the edges of a network but can have an effect on the evolution of the network itself.

2.7.1 Definition of Homophily

Homophily provides us with a first, fundamental illustration of how a network's surrounding context can drive the formation of its links. In general, a new link can be added for various reasons that are intrinsic to the network itself: we don't need to look beyond the network to understand where the link came from. But also the new link arises for an equally natural reason, but one that makes sense only when we look at the contextual factors beyond the network.

There are strong interactions between intrinsic and contextual effects on the formation of any single link: they are both operating concurrently in the same network.

For example, the principle of triadic closure — that triangles in the network tend to “close” as links form between friends of friends — is supported by a range of mechanisms that range from the intrinsic to the contextual (i.e. social contexts).

2.7.2 Measuring Homophily

To answer the question about whether the edges are “genuinely” present in the network itself, and not simply an artifact of how the network is drawn, we need to introduce a numerical measure.

We can formalize the test for Homophily with 2 categories according as following:

If the fraction of cross-category edges is significantly less than $2pq$, then there is evidence for homophily.

where p is the probability of belonging to the first category, and q is the probability for the second one (p,q are usually given and are not calculated from the graph).

In Figure 11, for example, 5 of the 18 edges in the graph are cross-category. Since $p = 2/3$ and $q = 1/3$ in this example, we should be comparing the fraction of cross-category edges to the quantity $2pq = 4/9 = 8/18$. In other words, with no homophily, one should expect to see 8 cross-category edges rather than than 5, and so this example shows some evidence of homophily.

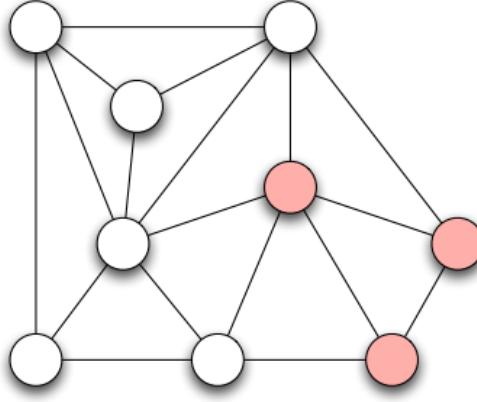


Figure 11: A network with two class of nodes.

When the characteristic can have more than two possible values, we still perform a general version of the same calculation. For this, we say that an edge is heterogeneous if it connects two nodes that are different according to the characteristic in question. We then ask how the number of heterogeneous edges compares to what we'd see if we were to randomly assign values for the characteristic to all nodes in the network (using the proportions from the real data as probabilities). In this way, even a network in which the nodes are classified into many groups can be tested for homophily using the same underlying comparison to a baseline of random mixing.

2.8 Assortativity

Assortativity, or assortative mixing is the preference for a network's nodes to connect with others that are similar. We can think of assortativity as homophily in a network context.

Intuitively we can say that a network is assortative if a significant fraction of the edges in the network run between nodes of the same type.

This fraction is 1 if all nodes belong to the same single type and 0 if there is a complete separation. We would like a measure that is large in non-trivial cases and small in trivial ones. A better measure quantifies the level on non-randomness in placement of edges in the network.

We can measure the assortativity of a network in different ways based on the variable types:

- *categorical*: comparable but not quantifiable (i.e. sex, colors, ...)
- *numerical*: comparable and sortable (i.e. age, income, ...)

2.8.1 Modularity

The *modularity* can be defined as the difference between the fraction of edges between nodes of the same type and the expected number of edges between all pairs of node of the same type. This measure is used when we have to deal with categorical features.

Important: we'll define modularity in the case of undirected graphs.

Given the group (class or type) of node i $g_i = 1..N$, we can compute the fraction of edges between nodes of the same type as follows:

$$\sum_{edges(i,j)} \delta_{g_i g_j} = \frac{1}{2m} \sum_{ij} a_{ij} \delta_{g_i g_j} \quad \delta_{g_i g_j} = \begin{cases} 1 & \text{if nodes } g_i \text{ and } g_j \text{ are of the same class} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Then we do the same for the expected number of edges between all pairs of node of the same type:

$$\frac{1}{2m} \sum_{ij} \frac{k_i k_j}{2m} \delta_{g_i g_j} \quad (4)$$

Now, we can formalize the definition of modularity as follows:

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{ij} a_{ij} \delta_{g_i g_j} - \frac{1}{2m} \sum_{ij} \frac{k_i k_j}{2m} \delta_{g_i g_j} \\ &= \frac{1}{2m} \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta_{g_i g_j} \end{aligned}$$

Then we can normalize the modularity as follows:

$$r = \frac{\frac{1}{2m} \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta_{g_i g_j}}{\sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) \delta_{g_i g_j}}$$

We can interpret the value of modularity as the extent to which similar nodes connect in a network. It is strictly less than 1 and takes positive values if there are more edges between nodes of the same type than we would expect by random chance. It can also take negative values if there are fewer such edges than we would expect by chance.

2.8.2 Assortativity coefficient

In case we have to deal with scalar features x_i for the node i , i.e. age, income, etc.; we consider the pairs of values x_i, x_j for the nodes i and j at the ends of each edge in the network and let us calculate their covariance over all edges, i.e. the joint variability of the values over the network.

$$\text{cov}(x_i, x_j) = \frac{1}{2m} \sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) x_i x_j$$

The covariance will be positive if, on balance, values at either end of an edge tend to be both large or both small, and negative if they tend to vary in opposite directions. In other words, the covariance will be positive when we have assortative mixing and negative for disassortative mixing.

If we want that the covariance takes value 1 in a perfect assortative mixing, we can normalize by the variance to get the assortative coefficient r :

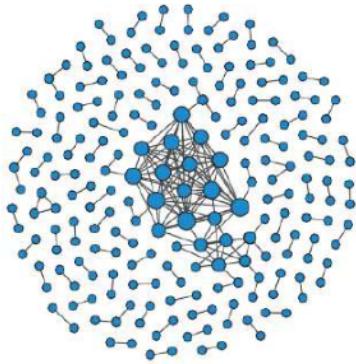
$$r = \frac{\frac{1}{2m} \sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) x_i x_j}{\sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) x_i x_j}$$

The correlation coefficient varies in value between a maximum of 1 for a perfectly assortative network and a minimum of -1 for a perfectly disassortative one (opposite of homophily, which dissimilar nodes tend to be connected, i.e. sexual sentimental networks).

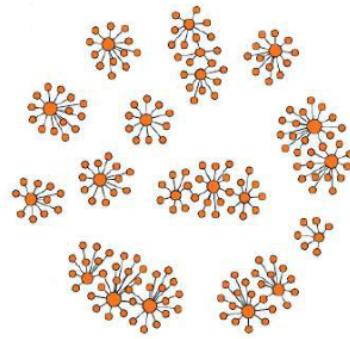
2.8.3 Degree Assortativity

Sometimes we want to measure the assortativity using scalar properties internal to the graph such as the degree. Assortativity based on degree is called degree assortativity or degree correlation: this occurs when high-degree nodes tend to be connected to other high-degree nodes, while low-degree nodes tend to have other low-degree nodes as neighbors. Networks with this property are called assortative. An example of an assortative network is shown in Figure 12a: the hubs form a densely connected core while low-degree nodes are loosely attached to each other and/or to core nodes. Therefore we say that assortative networks have a *core-periphery* structure. Social networks are often assortative. Networks where high-degree nodes tend to be connected to low-degree nodes and vice versa are called disassortative.

There are two ways to measure the degree assortativity of a network, both based on measuring the correlation between degrees of neighbor nodes.



(a) Assortative Network



(b) Disassortative Network

Figure 12: Example of assortative and disassortative networks

2.8.4 Assortativity coefficient by degree

The first one is based on the assortativity coefficient considering similar nodes with the same degree then we can substitute x_i with k_i and obtain the assortativity coefficient by degree, defined as follows:

$$r = \frac{\frac{1}{2m} \sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) k_i k_j}{\sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) k_i k_j}$$

where:

$$\text{cov}(k_i, k_j) = \frac{1}{2m} \sum_{i,j} \left(a_{i,j} - \frac{k_i k_j}{2m} \right) k_i k_j$$

When the assortativity coefficient is positive, the network is assortative, and when it is negative, the network is disassortative.

2.8.5 Measuring Assortativity by neighbors

Another way to compute the degree assortativity is by measuring the **degree correlation function**, which is the correlation between the degree and the average degree of the neighbors of nodes with that degree.

First of all we define the average degree of the neighbors of node i as follows:

$$k_{nn}(i) = \frac{1}{k_i} \sum_{j=1}^N a_{ij} k_j$$

Then we define the degree correlation function as the average of all k -degree nodes:

$$\langle k_{nn}(k) \rangle = \sum_{k'} k' P(k'|k)$$

Now, with that in mind, we can classify the networks into three types based on the degree correlation function's behavior with respect to the range of the possible degree values:

- **Assortative networks:** The networks that have a core-periphery structure with hubs in the core (like social networks). The hubs tend to link to each other and avoid linking to small-degree nodes. The increasing $k_{nn}(k)$ with k indicates that the network is assortative (Figure 13a).
- **Disassortative networks:** Networks that have hub-and-spoke structures that are disconnected from each other (like the web). The hubs tend to link to small-degree nodes. The decreasing $k_{nn}(k)$ documents the network's disassortative nature (Figure 13b).
- **Neutral Network:** don't have a precise structure (like in power grids). Nodes link to each other randomly. The horizontal $k_{nn}(k)$ indicates the lack of degree correlations, in line with our expectations for neutral networks (Figure 13c).

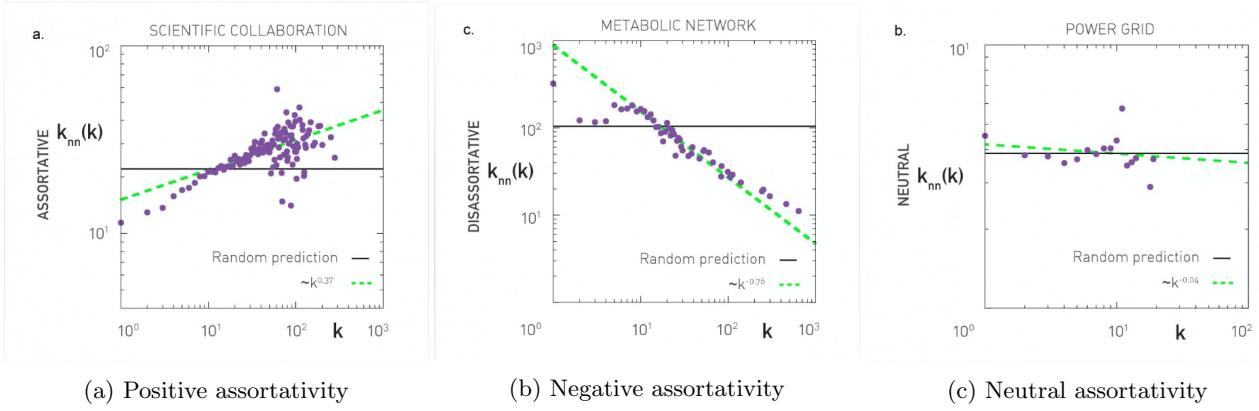


Figure 13: Three types of assortativity.

2.8.6 Mechanisms Underlying Homophily: Selection and Social Influence

In case of immutable characteristics such as race or ethnicity, the tendency of people to form friendships with others who are like them is called **selection**, in that people are selecting friends with similar characteristics. Selection may operate at several different scales, and with different levels of intentionality.

When we consider how immutable characteristics interact with network formation, the order of events is clear, but with more mutable characteristics is not that obvious: the feedback effects between people's individual characteristics and their links in the social network become significantly more complex.

Social influence can be viewed as the reverse of selection: with selection, the individual characteristics drive the formation of links, while with social influence, the existing links in the network serve to shape people's (mutable) characteristics.

When we look at a single snapshot of a network and see that people tend to share mutable characteristics with their friends, it can be very hard to sort out the distinct effects and relative contributions of selection and social influence. Have the people in the network adapted their behaviors to become more like their friends, or have they sought out people who were already like them? Such questions can be addressed using *longitudinal studies* of a social network, in which the social connections and the behaviors within a group are both tracked over a period of time. Fundamentally, this makes it possible to see the behavioral changes that occur after changes in an individual's network connections, as opposed to the changes to the network that occur after an individual changes his or her behavior.

An observation of homophily is often not an endpoint in itself, but rather the starting point for deeper questions — questions that address why the homophily is present, how its underlying mechanisms will affect the further evolution of the network, and how these mechanisms interact with possible outside attempts to influence the behavior of people in the network.

2.9 Affiliation

These surrounding contexts have been viewed, appropriately, as existing “outside” the network. But in fact, it's possible to put these contexts into the network itself, by working with a larger network that contains both people and contexts as nodes. Through such a network formulation, we will get additional insight into some broad aspects of homophily, and see how the simultaneous evolution of contexts and friendships can be put on a common network footing with the notion of triadic closure.

In this course we'll focus on how to represent the set of activities (called *foci* “focal points” of social interaction) a person takes part in, and how these affect the formation of links.

2.9.1 Affiliation Networks

We can represent the participation of a set of people in a set of foci using a graph called affiliation network, since it represents the affiliation of people (drawn on the left) with foci (drawn on the right). More generally, affiliation networks are examples of bipartite graphs.

In the case of affiliation networks, the two categories are the people and the foci, with each edge connecting a person to a focus that he or she participates in.

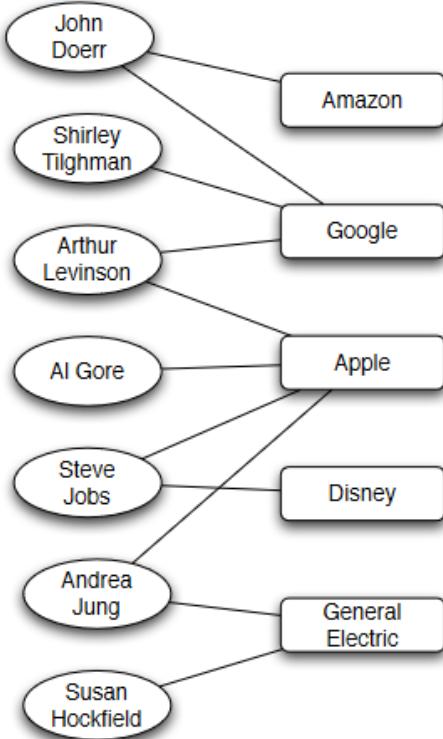


Figure 14: A small portion of a type of affiliation network that has been widely studied is the memberships of people on corporate boards of directors

2.9.2 Co-Evolution of Social and Affiliation Networks

Moreover, these changes represent a kind of co-evolution that reflects the interplay between selection and social influence: if two people participate in a shared focus, this provides them with an opportunity to become friends; and if two people are friends, they can influence each other's choice of foci.

A social-affiliation network representation slightly extends the affiliation network, simultaneously contains a social network on the people and an affiliation network on the people and foci. This type of network has two distinct kinds of edges: first kind of edge functions as an edge in a social network: it connects two people, and indicates friendship. The second kind of edge functions as an edge in an affiliation network: it connects a person to a focus, and indicates the participation of the person in the focus.

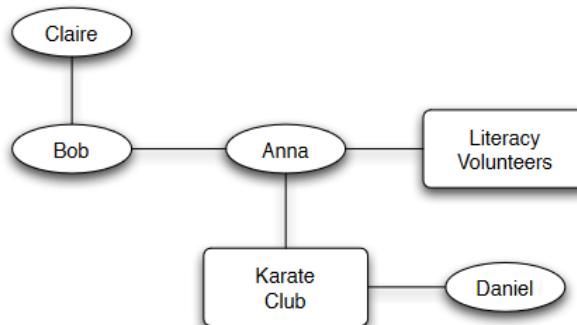


Figure 15: A social-affiliation network shows both the friendships between people and their affiliation with different social foci.

In social-affiliation networks as our representation, there is a range of different mechanisms for link formation

that can all be viewed as types of closure processes, in that they involve “closing” the third edge of a triangle in the network.

- **triadic closure**
- **focal closure:** selection
- **membership closure:** social influence

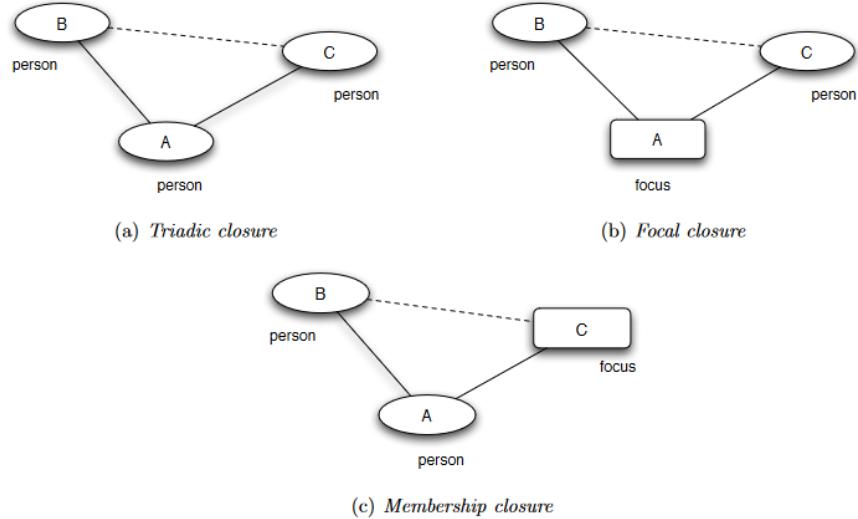


Figure 16: Each of triadic closure, focal closure, and membership closure corresponds to the closing of a triangle in a social-affiliation network.

2.10 Tracking Link Formation in On-Line Data

In general when we are extrapolating information from a network, we have to address to different questions about triadic closures, for instance:

- How much more likely is a link to form between two people in a social network if they already have a friend in common?
- How much more likely is an edge to form between two people if they have multiple friends in common?
- How much more likely is the formation of a link in the first of these two cases?

We can address these questions empirically using network data as follows:

1. We take two snapshots of the network at different times.
2. For each k , we identify all pairs of nodes who have exactly k friends in common in the first snapshot, but who are not directly connected by an edge.
3. We define $T(k)$ to be the fraction of these pairs that have formed an edge by the time of the second snapshot. This is our empirical estimate for the probability that a link will form between two people with k friends in common.
4. We plot $T(k)$ as a function of k to illustrate the effect of common friends on the formation of links.

2.11 Centrality Measures

To address the question "How important a node can be?" We need a measure that quantify the centrality of that node, in other words the importance of that node.

2.11.1 Degree Centrality

We can measure the centrality in terms of node's degree, so that the higher the degree of a node is, the more central it is.

The degree is not a good measure if we are in a small network.

2.11.2 Closeness Centrality

Another way to measure the centrality of a node is by determining how "close" it is to the other nodes. This can be done by summing the distances from the node to all others.

$$\sum_{j \neq i} l_{ij}$$

If the distances are short on average, their sum is a small number and we say that the node has high centrality. This leads to the definition of closeness centrality, which is simply the inverse of the sum of distances of a node from all others.

$$g_i = \frac{1}{\sum_{j \neq i} l_{ij}}$$

where l_{ij} is the distance from i to j and the sum runs over all the nodes of the network, except i itself.

An alternative formulation is obtained by multiplying by the constant, which is just the number of terms in the sum at the denominator:

$$\tilde{g}_i = (N - 1)g_i = \frac{N - 1}{\sum_{j \neq i} l_{ij}} = \frac{1}{\sum_{j \neq i} l_{ij}/(N - 1)}$$

In other word, the expression $\sum_{j \neq i} l_{ij}/(N - 1)$ is the average distance from the focal node i to the rest of the network and we that closeness can be expressed equivalently as the inverse of the average distance.

This way we discount the graph size and make the measure comparable across different networks.

2.11.3 Betweenness Centrality

Given two nodes h, j , there might be more than one shortest path between the two nodes². The betweenness centrality of a node i is defined as:

$$b_i = \sum_{h \neq j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}}$$

where $\sigma_{hj}(i)$ is the number of shortest path between node h and j passing through node i ³, and σ_{hj} is the number of shortest path between node h and j . This means that, the more path between two nodes pass through a node, the more central the node is. This is a common measure in diffusion processes.

2.12 Centrality distributions

Using a particular centrality measure in a small network we can interpret the results better than if we are in a large network. In this case, a better way is to understand the centrality distribution to address the question about what is the most important node of the network.

The statistical distribution of a centrality measure tells us how many elements — nodes or links — have a certain value of centrality, for all possible values.

To better understand how centrality is distributed among the many nodes in large networks, we need to take a statistical approach. In this way we can focus on classes of nodes and links sharing similar features, rather than on single elements of the network. For example, we can group together all nodes having similar values of degree centrality. The statistical distribution of a centrality measure tells us how many elements — nodes or links — have a certain value of centrality, for all possible values.

The range of the distribution also reveals the heterogeneity of the network elements with respect to a specific

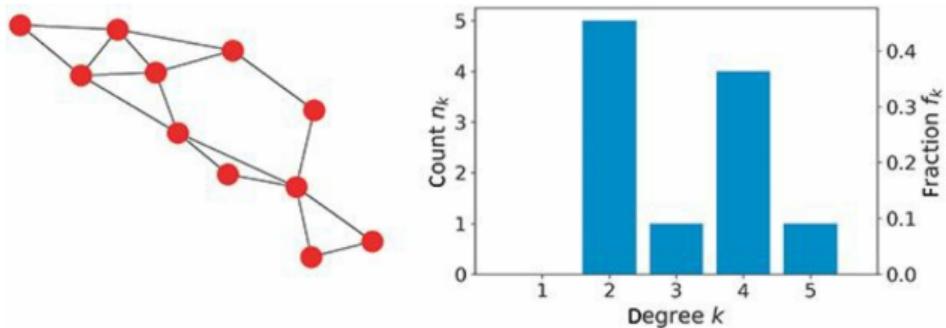


Figure 17: Example of a degree density where f_k is the frequency $f_k = \frac{n_k}{N}$ and n_k is the number of observations for the value k ; when $N \rightarrow \infty$ we can interpret f_k as a probability measure

centrality measure of interest: for example, if node degrees span many orders of magnitude, from units to millions, then the network is very heterogeneous with respect to degree.

Heavy-tailed degree distributions display a large heterogeneity in the degree values: while many of the nodes have just a few neighbors, some others have many neighbors, which gives them a prominent role in the network. These nodes are called hubs. Many natural, social, information and man-made networks have heavy-tailed degree distributions with highly connected hubs. One way to measure the breadth of the degree distribution is to compute the heterogeneity parameter, which compares the variability of the degree across nodes to the average degree.

To formally define the heterogeneity parameter κ of a network's degree distribution, we need to introduce the average squared degree, which is the average of the squares of the degrees:

$$\langle k^2 \rangle = \frac{k_1^2 + k_2^2 + \dots + k_{N-1}^2 + k_N^2}{N} = \frac{1}{N} \sum_i k_i^2$$

The heterogeneity parameter can be defined as the ratio between the average squared degree and the square of the average degree of the network:

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle^2}$$

²Shortest paths all have the same length.

³This length might be different from the length of the shortest path

For a normal or narrow distribution with a sharp peak at some value, say k_0 , the distribution of the squared degrees is concentrated around k_0^2 . Therefore $\langle k^2 \rangle \approx k_0^2$ and $\langle k \rangle \approx k_0$, yielding $\kappa \approx 1$.

For a heavy-tailed distribution with the same average degree k_0 , $\langle k^2 \rangle$ blows up because of the large degree of the hubs, so that $\kappa \gg 1$. If the degree distribution is concentrated around a typical value, there is no heterogeneity and the parameter is typically close to one. If the degree distribution is broad instead, the heterogeneity parameter is heavily inflated by the largest degrees of the hubs, and may take large values. The more hubs there are, the larger the heterogeneity. If a network is directed, like our Wikipedia and Twitter graphs, we have to consider two distributions, the in-degree and out-degree distributions, defined as the probability that a randomly chosen vertex has a given in- or out-degree, respectively. In this case, the definition of hub may refer to either the in-degree or the out-degree. For instance, a Web page may have many other pages linking to it (large in-degree), but it may itself link to just a few pages (low out-degree), or vice versa. In several directed networks the two measures are correlated, so nodes with large (small) in-degree also have large (small) out-degree.

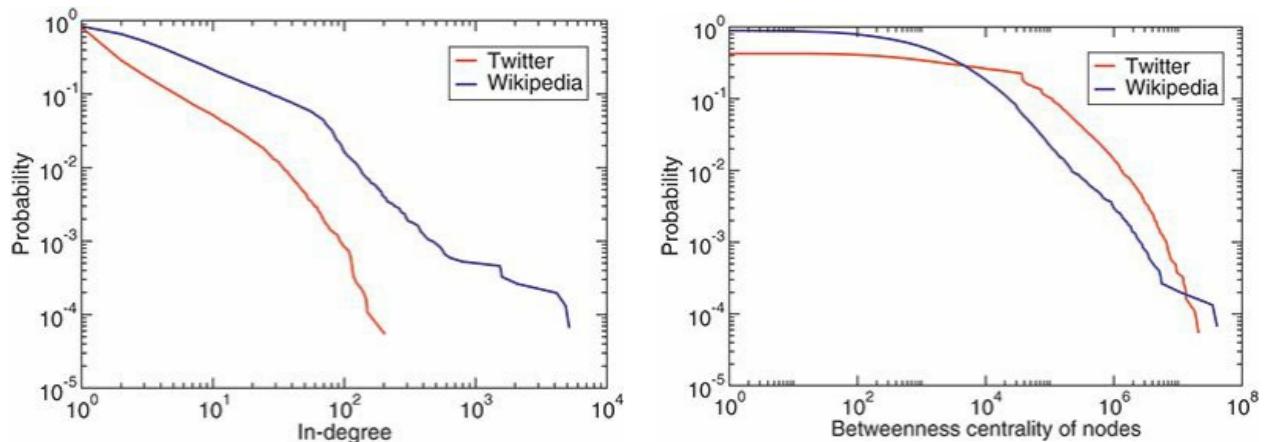
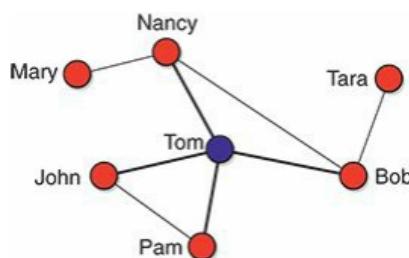


Figure 18: Degree vs Betweenness centrality distribution

Hubs, when present, are the single most important feature of a network. They are the pillars of its structure and the drivers of the processes running on it.

In practice, to have a better quality of the results we discard data that aren't "interesting" defining a threshold.

2.13 Ultra Small world problem and the friendship paradox



Suppose that you are looking for the person that has the highest number of friends, among N people. If you randomly select a person, the chance of selecting the one with most friends is $\frac{1}{N}$. If however, you ask a friend of a friend the chances are higher. this is due to the fact, that in this case, we are not choosing a node randomly, but we are choosing a link randomly; and since we want a link where one of the two edges is the target person, we have a much higher chance of selecting it. In the image below, if the target node is Tom, if we choose randomly between nodes, we have a probability of $\frac{1}{7}$, whereas if we choose to select a random link, the probability is $\frac{4}{8} = 0.5$. This is known as the **Friendship paradox**. This effect is extremely important whenever we aim to discover hubs, for example in epidemics. Thanks to this fact, hubs are extremely easy to find.

Usually, large networks have a property called **Ultra small world**: in networks with hubs, we expect that

the average distance is smaller than the average distance in the same network but with no hubs. We in fact expect that if a network has hubs, the distance between nodes is usually extremely small.

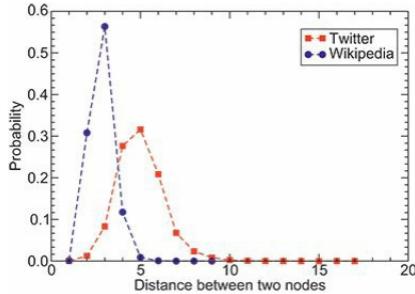


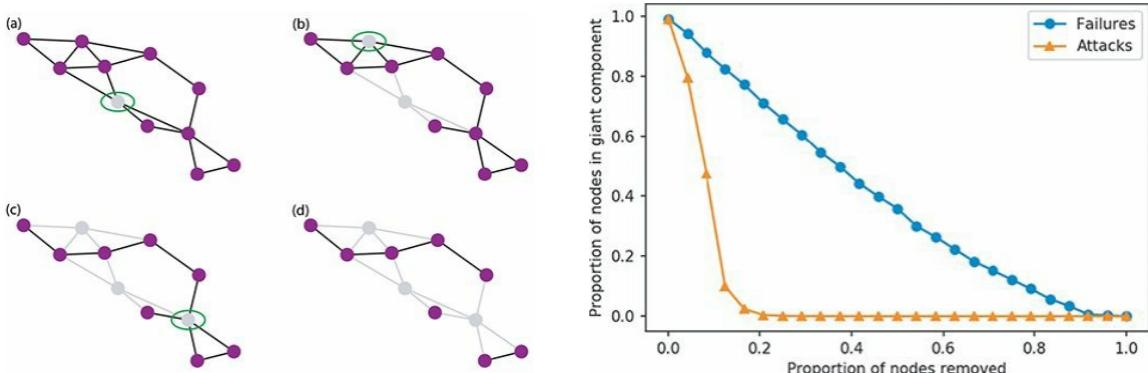
Figure 19: In this example, we see that the probability of having nodes with small distances between each other is way more higher than to have two nodes with very big distances.

2.14 Robustness

A system is **robust** if the failure of some of its components does not affect its function.

Nodes can describe a broad variety of entities (i.e. people, proteins, airports, etc.). In such a high-level representation, it is not straightforward to define the failure of a node, which depends on the specific type of network.

One way to define and measure the robustness of a network is to observe how the removal of a node and its links affects the connectedness of the system. If the system remains connected, we can assume that it will keep working fine, to some extent. However, a breakup of the network into disconnected pieces would signal severe damage that might compromise its function (Figure 20a).



(a) Network robustness. Effect of a sequence of deletions of nodes and their incident links. In each diagram the deleted node is highlighted by a circle. Deleted nodes and their incident links are colored in gray. After three nodes are deleted (d), the network breaks into three components disconnected from each other
(b) Network robustness. Fraction of nodes in the giant component as a function of the fraction of nodes removed from the OpenFlights World network. We show what happens if nodes are removed at random (random failures), or prioritized based on degree (targeted attacks)

Figure 20: Degree vs Betweenness centrality distribution

3 Network models

3.1 Features of real networks

We have seen that real networks of many different types share some common properties:

- **Small world problem** Most real world networks, are small world. This means that the average path length, $\langle l \rangle$, is usually very small.

- **High clustering coefficient** Many networks have high clustering coefficient, however, if the graph is bipartite or tree-like, the clustering coefficient is low.
- **Heterogeneous distributions** of node and link variables, like degree and weights.
- **Community structure** There are a lot of triadic closures, weak ties. Connections exists as proxy or affinity.

To understand where such properties come from, for instance how do nodes choose their neighbors? How are hubs generated? How are triangles formed? One way to study the origin of network characteristics is to formulate a model. By following the recipe of a model, we can build a network and compare it with real networks to see how they are similar or different. In this way, we can learn about the mechanisms that give rise to real-world networks.

3.2 Erdős-Renyi random graph

There are many ways to place the links between pairs of nodes, and to be more precise, we have seen two:

- **Gilbert random graph:** $G(n, p)$ is a random graph where any two nodes from n are connected to each other with probability p ;
- **Erdős-Renyi random graph:** $G(n, L)$ is a random graph, where L edges are placed at random between nodes.

The main difference between the two formulations is that in the version by Erdős and Rényi the number of links of the network is fixed, whereas in the model by Gilbert it is variable.

$G(n, p)$ and $G(n, L)$ are equivalent if:

$$L = \text{Binom} \left(\frac{N(N-1)}{2}, p \right)$$

Where Binom stands for binomial random variable, $\frac{N(N-1)}{2}$ is the number of throws and p is the probability for each throw.

3.2.1 Density

The number of links in a random graph is distributed with a binomial random variable, hence the expected number of links in a random graph is

$$\langle L \rangle = p \binom{N}{2} = \frac{pN(N-1)}{2}$$

The average degree of a network as twice the number of links divided by the number of nodes, we get the expected average degree of a random network:

$$\langle k \rangle = \frac{2\langle L \rangle}{N}$$

From the equation above we find that the expected density of a random network is $\langle d \rangle = p$.

The link probability expresses the density of a random network, real networks are usually sparse, for a random graph to be a good model of real networks, the link probability should be close to zero.

3.2.2 Degree distribution

The resulting probability distribution for the degree in a random network is a bell-shaped curve with a prominent peak concentrated around the average degree, and rapidly decaying on both sides of the peak. The degree of most nodes is close to the average degree, and large deviations from it are very unlikely.

Clearly, from the image below, we can see that a random network model does not provide a good description of the distribution: the nodes have approximately the same degree, so there are no hubs. Such discrepancy is one of the reasons why we need more sophisticated network models.

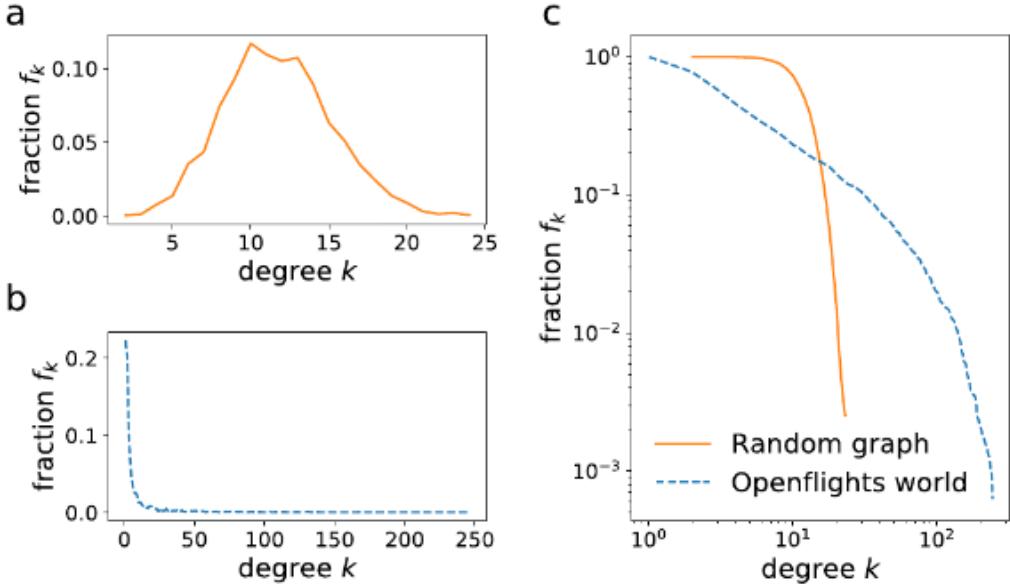


Figure 21: The degree distribution of random networks is very different from the broad distributions of most real-world networks

3.2.3 Short paths

Check whether random networks have short paths. We have seen in the previous section that the nodes have approximately the same degree.

The diameter l_{max} such that the number of nodes reachable within at most l_{max} steps from any node matches the total number of nodes N is given by $k^{l_{max}} = N$ from which we obtain:

$$l_{max} = \log_k N = \frac{\log N}{\log k}$$

It turns out that this is a good approximation of the diameter of the network, even when we consider neighborhood overlap and fluctuations in degree around. The slow logarithmic growth of l_{max} with N indicates that distances within the network are small, even when the network size is very large.

3.2.4 Clustering coefficient

The clustering coefficient of a node i can be interpreted as the probability that two neighbors of i are connected. In a random generated network, since links are placed independently from each other, the probability that two links are connected is

$$C_i = p = \frac{\langle k \rangle}{N}$$

Since k is usually a small number, the average clustering coefficient of random networks with realistic values for k and N is much smaller than the ones observed in real-world networks.

3.3 The configuration model

Let us answer the following question: *given some degree distribution, can we construct a network whose nodes have exactly that degree distribution?*

A simple solution is provided by the **configuration model**. This model actually pursues a more ambitious goal: generating a network whose nodes have an arbitrary degree sequence (Section 3.3.1). A degree sequence could be produced from a particular distribution we are interested in reproducing, or it could be taken from the nodes of a real network. Once we reproduce the sequence of all node degrees, we must have reproduced the corresponding degree distribution as well. In contrast, many degree sequences correspond to the same distribution. For example, two networks with distinct degree sequences (1,2,1) and (1,1,2) have the same degree distribution.

Suppose we have a set of nodes and their degree sequence. The first step is to assign to each node a number of stubs corresponding to the degree of the node, as in Figure 22(a). A stub is just a dangling link having the

node as one of its endpoints, but not yet connected to a neighbor. The network is then constructed by the following iterative steps:

1. A pair of stubs is selected at random.
2. The chosen stubs are joined to each other, forming a link between the nodes attached to the stubs.

This routine is repeated until all stubs are joined in pairs. Naturally, for this to happen, there needs to be an even number of stubs.

As illustrated in Figure 22(b–d), multiple networks can be created this way, depending on the sequence of pairs of stubs that are combined.

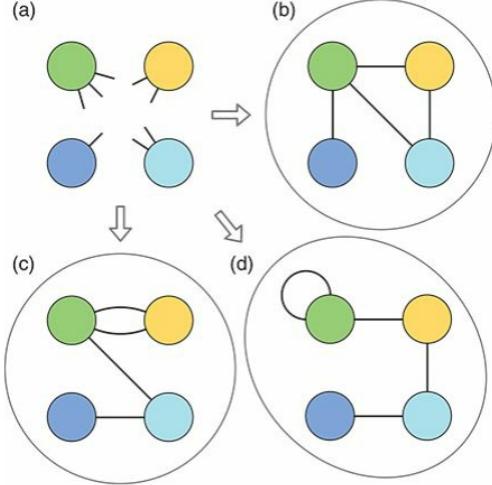


Figure 22: Configuration model. (a) We start with nodes and stubs corresponding to a given degree sequence. (b–d) We can connect the stubs in different ways, leading to different networks with the given degree sequence.

With this model, neighbors with high degree are more likely to get connected, so heterogeneity improves the clustering, however, randomness makes triangle rare, especially for large networks.

3.3.1 Degree Sequences

The degree sequence of a network is the list of degrees of its nodes, in the order of their labels. The degree sequence is a list of N numbers $(k_0, k_1, k_2, \dots, k_{N-1})$, where k_i is the degree of node i .

3.4 Small world networks

The randomness, gives to random generated graphs small diameter and low clustering coefficient. To solve this problem, we can interpolate a regular lattice (high clustering) and a random network.

We start from a lattice, then to shorten the average path length, we add shortcuts!

3.4.1 The Watts-Strogatz model

N nodes form a regular ring lattice, with even degree k . Each link is then rewired randomly with probability p . The expected number of rewired link is

$$pL = \frac{pNk}{2}$$

if $p = 0$ then no links are rewired, and we still have a lattice. if p is small, few links are rewired, the average clustering coefficient stays approximately the same, as few triangles are destroyed, but distances shrink considerably

If $p = 1$, all links are then rewired, and the network becomes a random network.

The degree after reshuffling is:

$$\begin{aligned}\bar{k} &= k - k_{out} + k_{in} \\ k_{out} &= \text{Binom}\left(\frac{k}{2}, p\right) \\ k_{in} &= \text{Binom}\left(\frac{(n-1)k}{2}, \frac{p}{n-1}\right)\end{aligned}$$

And the expected value, and variance are:

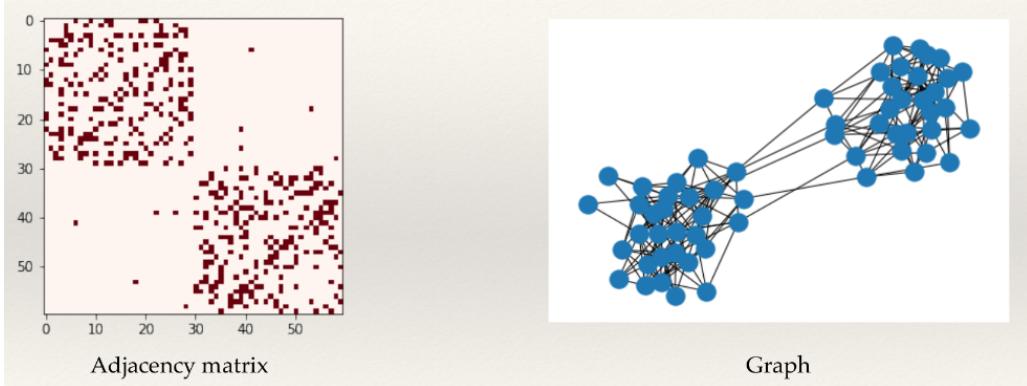
$$\mathbf{E}[\bar{k}] = k \quad \mathbf{V}[\bar{k}] \approx \frac{kp(2-p)}{2}$$

The degree distribution is peaked as most nodes have the same degree (no hubs), hence the Watts-Strogatz model fails to reproduce the broad degree distributions observed in many real-world networks

3.5 Random networks with communities

Suppose there are two communities. Each node has a label ($l = 1$ or $l = 2$). Then, each edge is generated independently at random with the following probability

$$\mathbb{P}(A_{ij} = 1) = \begin{cases} p_{in} & \text{if } l_i = l_j \\ p_{out} & \text{if } l_i \neq l_j \end{cases} \quad (5)$$



Setting $p_{in} > p_{out}$ we obtain the required community structure. This can be extended to multiple classes.

3.6 Network model comparison

Network model comparison				
Features	Erdős-Renyi	Configuration Model	Watts-Strogatz Model	Stochastic Block Model
Distances between pairs of nodes are short (small-world property)	✓	✓	✓	✓
Same clustering coefficient of real networks of the same size and average degree	✗	✗	✓	✗
Presence of hubs	✗	✓	✗	✗
Community structure exists	✗	✗	✗	✓

Table 1: Network model comparison

3.7 Power-Law

The concept of popularity is widespread present in the real world. We can define the popularity as a phenomenon characterized by extreme imbalances: while almost everyone goes through life known only to people in their immediate social circles, a few people achieve wider visibility, and a very, very few achieve global name recognition. To address the question of how we can quantify these imbalances, we need to introduce the power laws.

A powerlaw is a function k to some fixed power, for instance $\frac{1}{k^2}$, is called a power law; when used to measure the fraction of items having value k , it says, qualitatively, that it's possible to see very large values of k .

One of the first things that's worth doing is to test whether it's approximately a power law $\frac{1}{k^c}$ for some c , and if so, to estimate the exponent c .

Let $f(k)$ be the fraction of items that have value k , and suppose you want to know whether the equation $f(k) = \frac{a}{k^c}$ approximately holds, for some exponent c and constant of proportionality a . Then, if we write this as $f(k) = ak^{-c}$ and take the logarithms of both sides of this equation, we get

$$\log f(k) = \log a - c \log k$$

This says that if we have a power-law relationship, and we plot $\log f(k)$ as a function of $\log k$, then we should see a straight line: $-c$ will be the slope, and $\log a$ will be the y -intercept.

3.8 Barabasi-Albert model

Barabási-Albert model is a simple combination of **growth** and **preferential attachment**. At each step, a new node is added and connected to some existing nodes. The probability that a new node is attached to an old node is proportional to the degree of the old node.

3.8.1 Network growth

The models explored so far are static, all the nodes of the network are there from the beginning. Real networks, instead, are usually dynamic, nodes and links appear and disappear during the time. Nodes may also disappear

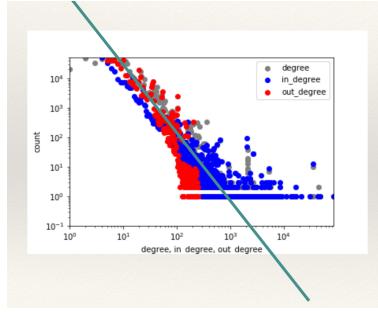


Figure 23

but the introduction of new nodes is more likely. This is the reason why realistic dynamic models typically incorporate some form of *network growth*.

Network Growth procedure: The dynamic procedure starts from an initial configuration, usually a very small clique of nodes. Then nodes are added one by one, each new node is attached to a number of old nodes based on some rule, which characteristic of the model (Figure 24).

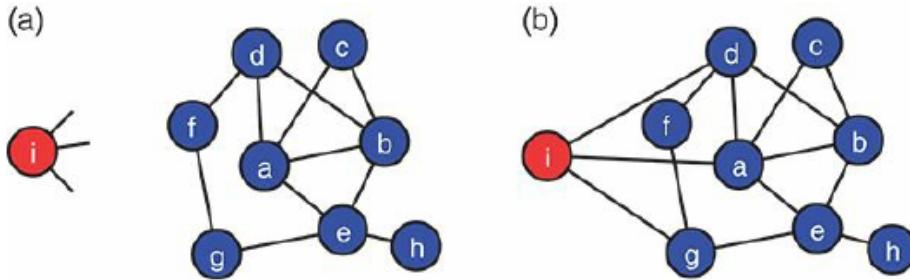


Figure 24: Network growth

3.8.2 Preferential attachment

Another limitation of the network models considered until this point is that they cannot explain the existence of hubs. To be more precise, the configuration model can generate hubs, but only by deciding a priori the degree of the nodes. The random network and small-world models do not give rise to hubs. The main reason is that in both cases the linking rules are basically egalitarian, the nodes choose their neighbors totally at random. This way, it is extremely unlikely for any one node to have an advantage over the other nodes and end up with many more neighbors than the rest. If we want to recover the hubs, it is thus necessary to introduce a mechanism that favors some nodes over others. Such a mechanism is called *preferential attachment*: a popular node is one with high degree, high-degree nodes have a high probability of receiving new links.

3.8.3 Extension of Barabasi-Albert: Attractiveness Model

What happens if a node has no neighbors ($K_i = 0$)? It will never get connections from other nodes. This is not a problem for standard initial condition, as the initial sub graph is a clique (so it is complete, and hence every node has nonzero degree). But what if the network is directed and the linking probability is proportional to the in-degree? Each new node has in-degree equal zero, so it will never be linked by future nodes...

We need to introduce a new procedure to solve the problem:

1. Start with a group of m_0 nodes, usually fully connected (clique)
2. At each step a new node i is added to the system, and sets m links with some of the older nodes ($m \leq m_0$)
3. The probability that the new node i chooses an older node j as neighbor is proportional to the sum of the

degree k_j of j and an attractiveness A , indicating the intrinsic appeal:

$$\text{Probability of } i \text{ linking with } j = \mathbb{P}(i \leftrightarrow j) = \frac{A + k_j}{\sum_l (A + k_l)}$$

At this point, if $A = 0$, we get the standard Barabasi Albert random graph. For every value of A we get networks with heavy-tailed degree distributions. The pattern of the distribution changes with A , so it is possible to match distributions of real-world networks, unlike the BA model.

3.8.4 Extension of Barabasi-Albert: Fitness Model

With the standard, or the Attractiveness Barabasi-Albert model, a problem arises: **Pitfall of preferential attachment:** Hubs are the oldest nodes. This means that the oldest nodes are hubs. This is extremely unrealistic.

The fitness model is introduced to solve this issue, and the steps are:

1. Start with a group of m_0 nodes, usually fully connected (clique)
2. At each step a new node i is added to the system, and sets m links with some of the older nodes ($m \leq m_0$)
3. The probability that the new node i chooses an older node j as neighbor is proportional to the product of the degree k_j of j with a fitness η_j , indicating the intrinsic appeal of j :

$$\text{Probability of } i \text{ linking with } j = \mathbb{P}(i \leftrightarrow j) = \frac{\eta_j k_j}{\sum_l (\eta_l k_l)}$$

The fitness values are extracted from a distribution $\rho(\eta)$ and assigned to each new node. Difference with attractiveness model are that the fitness enters as a factor in the link probability, not as a summed, and that the fitness is characteristic of each node, it is not a constant. The result of the model are the following:

- If the fitness distribution $\rho(\eta)$ has finite support (the fitness is distributed over a finite range of values), the degree distribution of the network is heavy tailed
- If the fitness distribution $\rho(\eta)$ has infinite support (the fitness is distributed over an infinite range of values), the node with largest fitness attracts a big fraction of all links
- Nodes with large fitness can acquire a large degree even if they are introduced late in the system (Which is a good thing)

3.8.5 Extension of Barabasi-Albert: Random Walk Model

The standard Barabasi Albert model have a problem: there are too few triangles. This is because to close a triangle we need to set a link between two neighboring nodes, whereas in the BA model links are set based on degree, regardless of whether the future neighbors have common neighbors.

We need to introduce a mechanism to implement the triadic closure:

1. Start with a group of m_0 nodes, usually fully connected (clique)
2. At each step a new node i is added to the system, and sets m links with some of the older nodes ($1 < m \leq m_0$)
3. The first link targets a randomly chosen node j
4. From the second link onwards:
 - With probability p the link is set with a neighbor of j , chosen at random
 - With probability $1 - p$ the link is set with a randomly chosen node

As a result:

- The degree distribution is heavy-tailed;
- The average clustering coefficient is much higher than in BA networks (the larger, the greater the probability p of triadic closure)

- When the triadic closure probability p is sufficiently high that many triangles are formed, the network has community structure (it is made of cohesive groups of nodes, loosely connected to each other).

If links are set at random, as it seems, how can the model generate hubs? The triadic closure mechanism of the random walk model induces effective preferential attachment, as choosing a random node and a random neighbor of the node is like choosing a link at random, and the probability that the endpoint(s) of a randomly selected link have a given degree is proportional to the degree.

3.8.6 Extension of Barabasi-Albert: Rank Model

In realistic settings, it is more common to have a perception of the relative value of things, rather than of their absolute value. The idea at the basis of the *rank model* is that in real world scenarios we are able to rank the nodes of a network based on a specific variable (e.g. degree or age). Doing this we keep the nodes ranked by one of their properties, say the degree. Then we select nodes to receive new links with probability proportional to some inverse power of their rank. The top node will have the highest chance of receiving a link, followed by the node ranked second, third, and so on. How the link probability decays with the rank is determined by an exponent parameter.

The rank model differs from the previously defined models for the formula used to determine the probability to have a link between a pair of nodes (i, j) :

$$\mathbb{P}(i \leftrightarrow j) = \frac{R_j^{-\alpha}}{\sum_k R_l^{-\alpha}}$$

Where the exponent α is a parameter and R_j is the rank of the node j .

From the formula above, we can clearly see that the top-ranked node (small rank) is more likely to receive a new link than a poorly ranked one (large rank). If the variable used for the ranking is degree, this means that high-degree nodes have higher chances of attracting new links than low-degree nodes, as in the BA model.

As it turns out, the rank model generates networks with heavy-tailed degree distributions, for any property used to rank the nodes and any value of the exponent parameter.

By tuning the exponent, it is possible to vary the shape of the distribution, and to reproduce the empirical distributions observed in many real-world networks.

Hubs are created even if nodes have partial information on the system, in that they are aware of the existence of only a fraction of the nodes.

4 Epidemics

Epidemiology has developed a robust analytical and numerical framework to model the spread of pathogens. This framework relies on two fundamental hypotheses:

- **Compartmentalization:** Epidemic models classify each individual based on the stage of the disease affecting them. The simplest classification assumes that an individual can be in one of three states or compartments:
 - Susceptible (S): Healthy individuals who have not yet contacted the pathogen
 - Infectious (I): Contagious individuals who have contacted the pathogen and hence can infect others.
 - Recovered (R): Individuals who have been infected before, but have recovered from the disease, hence are not infectious.
- **Homogeneous Mixing:** The homogenous mixing hypothesis (also called fully mixed or mass-action approximation) assumes that each individual has the same chance of coming into contact with an infected individual. This hypothesis eliminates the need to know the precise contact network on which the disease spreads, replacing it with the assumption that anyone can infect anyone else.

4.1 Basic Reproduction Number

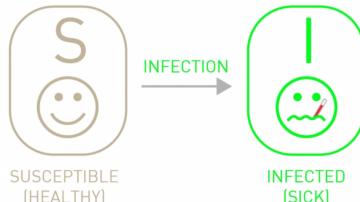
The R_0 index, also known as Basic reproduction number, is an index that express the evolution of a pandemic. For each of the below defined systems, R_0 , has a given formula but its value tells us the following things:

- If $R_0 > 1$, then the epidemic is in the endemic state;
- If $R_0 = 1$, then the pandemic will remain the same in size;
- If $R_0 < 1$, then the pandemic will shrink, and eventually disappear

This index is not static, and it is defined as a function of time (see below), and as such it will change during a pandemic, either increasing or shrinking. The higher R_0 is, the faster the pandemic will spread.

4.2 SI Systems

In this model, we are under the hypothesis that once a person gets infected, it cannot be recovered.



Denote with $S(t)$ the number of individuals who are susceptible (healthy) at time t and $I(t)$ the number individuals that have been already infected.

Let us assume that a typical individual has $\langle k \rangle$ contacts and that the likelihood that the disease will be transmitted from an infected to a susceptible individual in a unit time is β . From that we ask the following: "How change the number of infected people over time t ?" Within the homogeneous mixing hypothesis the probability that the infected person encounters a susceptible individual is $\frac{S(t)}{N}$. Therefore the infected person comes into contact with $\langle k \rangle \frac{S(t)}{N}$ susceptible individuals in a unit time. Since $I(t)$ infected individuals are transmitting the pathogen, each at rate β , the number of new infections $dI(t)$ during a timeframe dt is:

$$\frac{dI(t)}{dt} = \beta\langle k \rangle \frac{S(t)I(t)}{N}$$

$$= \beta\langle k \rangle s \cdot i \quad \left(s(t) = \frac{S(t)}{N}, \quad i(t) = \frac{I(t)}{N} \right)$$

$$= \beta\langle k \rangle i(1 - i) \quad (s + i = 1)$$

$$\frac{di}{i} + \frac{di}{(1-i)} = \beta\langle k \rangle dt$$

$$\int \frac{di}{i} + \frac{di}{(1-i)} = \int \beta\langle k \rangle dt$$

$$\ln i - \ln(1-i) + C = \beta\langle k \rangle t$$

$$i(t) = \frac{i_0 e^{\beta\langle k \rangle t}}{1 - i_0 + i_0 e^{\beta\langle k \rangle t}}$$

■

The characteristic time required to reach an $1/e$ fraction of all susceptible individuals is $\tau = \frac{1}{\beta\langle k \rangle}$.

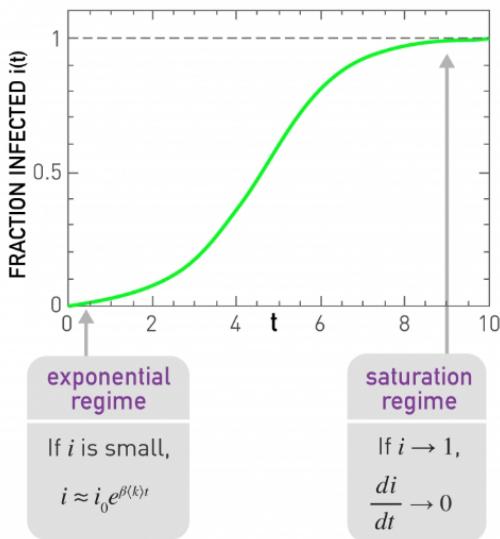


Figure 25: SI system

From image 25 we can see that the number of infected people grows exponentially at first, and then for $t \rightarrow \infty$, we get a *saturation regime*: $i(\infty) = 1$. This means that at a certain point, all the population will be infected. Also in this model, there is no **Epidemic threshold** so the number of susceptible people, will tend to zero at $t = \infty$.

For this model \mathbf{R}_0 is calculated as:

$$\mathbf{R}_0 = \frac{\beta}{N}$$

4.3 SIS Systems

To capture the fact that most pathogens are eventually defeated by the immune system or by treatment, we define the SIS model which has the same two states as the SI model, susceptible and infected. The difference

is that now infected individuals recover at a fixed rate μ , becoming susceptible again.

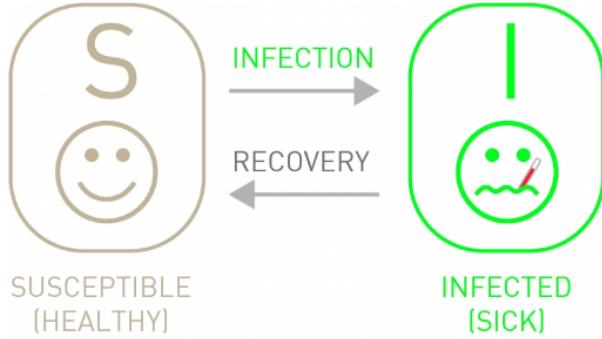


Figure 26: SIS system

The equation describing the dynamics of this model is an extension of section 4.2:

$$\frac{di}{dt} = \beta \langle k \rangle i(1 - i) - \mu i$$

where μ is the recovery rate and the μi term captures the rate at which the population recovers from the disease. Following the same steps we did in the SI model we finally obtain the equation of the infected by time:

$$i(t) = \left(1 - \frac{\mu}{\beta \langle k \rangle}\right) \frac{Ce^{(\beta \langle k \rangle - \mu)t}}{1 + Ce^{(\beta \langle k \rangle - \mu)t}} \quad (6)$$

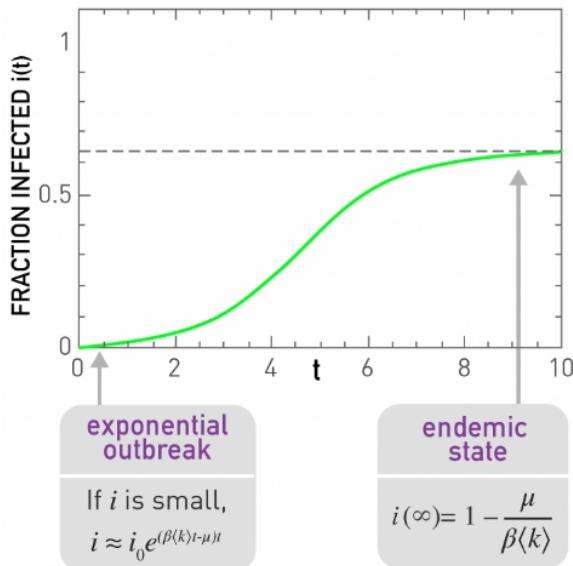


Figure 27: SIS system: fraction of infected

While in the SI model eventually everyone becomes infected, 6 predicts that in the SIS model the epidemic has two possible outcomes:

- Endemic state: For low recovery rate the fraction of infected individuals, i , follows a logistic curve similar to the one observed for the SI model. Yet, not everyone is infected, but i reaches a constant $i(\infty) < 1$ value, in particular we have that:

$$i(\infty) = 1 - \frac{\mu}{\beta \langle k \rangle}$$

- Disease-free State: For a sufficiently high recovery rate the exponent in (6) is negative. Therefore, i decreases exponentially with time, indicating that an initial infection will die out exponentially. This is because in this state the number of individuals cured per unit time exceeds the number of newly infected individuals. Therefore with time the pathogen disappears from the population.

For this model \mathbf{R}_0 is calculated as:

$$\mathbf{R}_0 = \frac{\beta\langle k \rangle}{\mu}$$

Recall that the reproductive number provides the number of individuals an infectious individual infects if all its contacts are susceptible. For $R_0 < 1$ the pathogen naturally dies out, as the number of recovered individuals exceeds the number of new infections. If $R_0 > 1$ the pathogen will spread and persist in the population. The higher is R_0 , the faster is the spreading process.

4.4 SIR System

For many pathogens, like most strains of influenza, individuals develop immunity after they recover from the infection. Hence, instead of returning to the susceptible state, they are “removed” from the population. These recovered individuals do not count any longer from the perspective of the pathogen as they cannot be infected, nor can they infect others. The SIR model, captures the dynamics of this process.

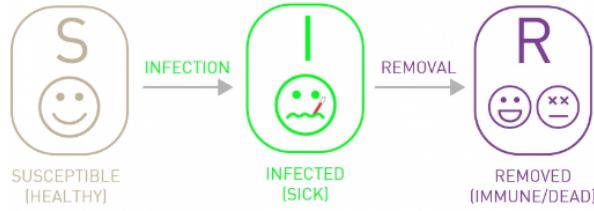


Figure 28: SIR system

We model the behavior of the system accordingly with the following differential equations:

$$\frac{ds}{dt} = -\beta\langle k \rangle i [1 - r - i] \quad \frac{di}{dt} = -\mu i + \beta\langle k \rangle i [1 - r - i] \quad \frac{dr}{dt} = \mu i$$

As we can see in image 29 we have that everyone recovers at the end.

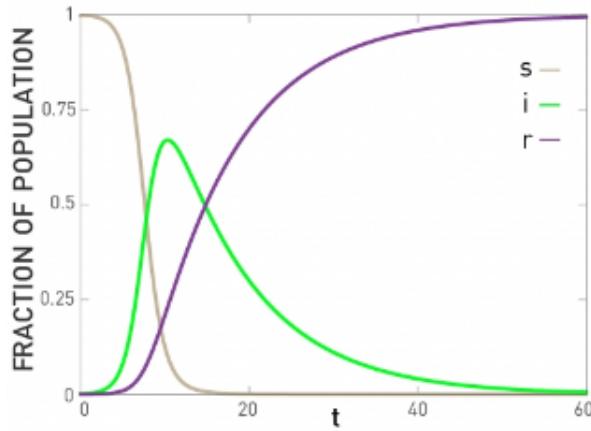


Figure 29: SIR system: fraction of infected

For this model \mathbf{R}_0 is calculated as:

$$\mathbf{R}_0 = \frac{\beta\langle k \rangle}{\mu}$$

4.5 Conclusions on systems

In summary, depending on the characteristics of a pathogen, we need different models to capture the dynamics of an epidemic outbreak.

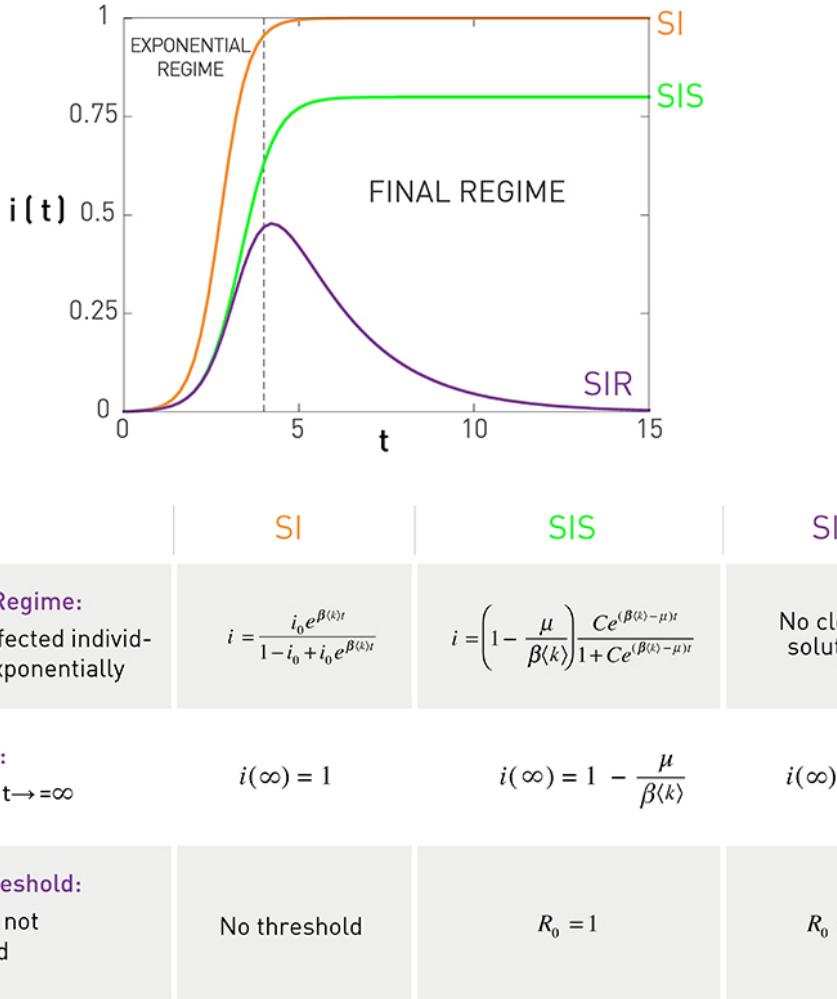


Figure 30: Summary of SI, SIS and SIR models

4.6 Network epidemics

The epidemic models previously defined fails to model real world epidemics because they assume that any individual can come into contact with any other individual (homogenous mixing hypothesis) and that all individuals have comparable number of contacts, $\langle k \rangle$. Both assumptions are false: Individual can transmit a pathogen only to those they come into contact with, hence pathogens spread on a complex contact network. Furthermore, these contact networks are often scale-free, hence $\langle k \rangle$ is not sufficient to characterize their topology.

4.7 Degree block approximation

To account for the role of the network topology, we use a different approach called the degree block approximation. This method adds another set of compartments, placing all nodes that have the same degree into the same block. In other words, we assume that nodes with the same degree behave similarly.

4.8 Susceptible-Infected (SI) Model on a Network

If a pathogen spreads on a network, individuals with more links are more likely to be in contact with an infected individual, hence they are more likely to be infected. Therefore the mathematical formalism must consider the degree of each node as an implicit variable. This is achieved by the degree block approximation, that distinguishes nodes based on their degree and assumes that nodes with the same degree are statistically equivalent. We define the following equations:

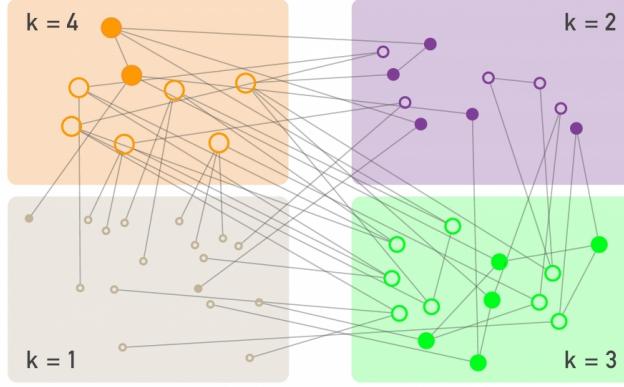


Figure 31: Degree block approximation

$$i_k = \frac{I_k}{N_k} \quad i = \sum_k p_k i_k \quad \frac{di_k}{dt} = \beta(1 - i_k)k\Theta_k \quad (7)$$

Where i_k is the fraction of nodes with degree k that are infected among all N_k degree- k nodes in the network; i is the total fraction of infected nodes; p_k is the degree distribution value for degree k . Θ_k is the density function representing the fraction of infected neighbors of a susceptible node k . $\frac{di_k}{dt}$ is the rate equation of infected members that have degree k

Given the equation (7), we explore the early time behavior of i_k (useful i.e. for developing cures for a new pathogen). In other words, we want to analyze the number of individual infected in the early stages of the epidemic.

4.9 SIS Model and the Vanishing Epidemic Threshold

The continuum equation describing the dynamics of the SIS model on a network is a straightforward extension of the SI model discussed previously:

$$\frac{di_k}{dt} = \beta(1 - i_k)k\Theta_k(t) - \mu i_k \quad (8)$$

The difference between (7) and (8) is the presence of the recovery term $-\mu i_k$. This change the characteristic time of the epidemic to:

$$\tau^{SIS} = \frac{\langle k \rangle}{\beta \langle k^2 \rangle - \mu \langle k \rangle}$$

For sufficiently large μ the characteristic time is negative, hence i_k decays exponentially. The condition for the decay depends not only on the recovery rate and $\langle k \rangle$, but also on the network heterogeneity, through $\langle k^2 \rangle$. To predict when a pathogen persists in the population we define the spreading rate:

$$\lambda = \frac{\beta}{\mu}$$

where β is the transmission probability and μ is the recovery rate. The higher is λ , the more likely that the disease will spread. Yet, the number of infected individuals does not increase gradually with λ . Rather, the pathogen can spread only if its spreading rate exceeds an epidemic threshold λ_c .

4.9.1 SIS spreading in random networks

If a pathogen spreads on a random network, we can calculate and obtain that a pathogen persists on a population if:

$$\lambda > \frac{1}{\langle k \rangle + 1}$$

And we also get that the spreading rate is:

$$\lambda > \frac{1}{\langle k \rangle + 1} \quad \lambda_c = \frac{1}{\langle k \rangle + 1}$$

Where λ_c is the epidemic threshold, that when the epidemic values are less than λ_c , there is no epidemic, otherwise, there is an epidemic where the values of infected at time inf is dictated by λ .

4.10 Temporal vs Aggregated Networks

Most interactions in a network are not continuous, but have a finite duration. We must therefore view the underlying networks as temporal networks.

From the image above we can see the timeline of the interactions between four individuals. Each vertical line marks the moment when two individuals come into contact with each other. If A is the first to be infected, the pathogen can spread from A to B and then to C , eventually reaching D . If, however, D is the first to be infected, the disease can reach C and B , but not A . This is because there is a temporal path from A to D . The resulting network obtained by merging the temporal interactions is called *aggregated network*. If we only have access to this aggregated representation, the pathogen can reach all individuals, independent of its starting point.

Bursty interactions are observed in a number of contact processes of relevance for epidemic phenomena, from email communications to call patterns and sexual contacts. To be specific, power law interevent times increase the characteristic time T . The number of infected individuals decays slower than predicted by a random contact pattern.

4.11 Immunization

Immunization strategies specify how vaccines, treatments or drugs are distributed in the population. They are guided by monitoring the pathogen's spreading rate λ . These strategies are classified in two categories:

- Random immunization
- Selective immunization

4.11.1 Random Immunization

Let us also consider the situation when a randomly selected g fraction of individuals are immunized in a population, only the remaining $(1 - g)$ fraction of the nodes can contact and spread the disease consequently, the effective degree of each susceptible node changes from $\langle k \rangle$ to $\langle k \rangle(1 - g)$.

The spreading rate $\lambda = \beta/\mu$ decreases to $\lambda' = \lambda(1 - g)$. If the pathogen spreads on a random network, for a sufficiently high g_c the spreading rate λ' could fall below the epidemic threshold λ_c as follows:

$$\lambda_c = \lambda(1 - g_c) = (1 - g_c) \frac{\beta}{\mu} \simeq \frac{1}{\langle k \rangle}$$

obtaining $g_c \simeq 1 - \frac{\mu}{\beta \langle k \rangle}$. Thus let us with the following consideration: if we have a fraction of immunized individuals above g_c , our spreading rate will be under the epidemic threshold: the pathogen will die out.

In the case where the pathogen spreads on a network with $\kappa \gg 1$, random immunization will allow us to determine the critical g_c as follows:

$$\lambda_c = \lambda(1 - g_c) = (1 - g_c) \frac{\beta}{\mu} \simeq \frac{\langle k \rangle}{\langle k^2 \rangle}$$

obtaining $g_c \simeq 1 - \frac{\mu \langle k \rangle}{\beta \langle k^2 \rangle}$ and in heterogeneous networks $\langle k^2 \rangle \gg \langle k \rangle^2$. In scale-free networks, $\langle k^2 \rangle \rightarrow \infty$, hence $g_c \rightarrow 1$: we need to immunize virtually all nodes to stop the epidemics.

4.11.2 Selective Immunization

Random immunization is unable to eradicate a disease, but targeting the hubs, can restore a finite critical threshold, helping us eradicate the disease. We must find a way to increase the epidemic threshold: reduce the variance $\langle k^2 \rangle$ of the underlying contact network, so the focus is to immunize the hubs (detected by the friendship paradox).

5 Communities

5.1 Definition of community

There are two main features that defines communities:

- **High cohesion:** communities have many internal links, so their nodes stick together;
- **High separation:** communities are connected to each other by few links

5.1.1 Community detection

Given a definition of community and a graph, a community detection algorithm is an unsupervised algorithm that outputs a node partition, assigning a community label to each node.

5.2 Cost function optimization: min-cut problem

To identify communities, we can define a function that tells us how good a node partition is, and perform community detection by simply optimizing that function. This means that the cost function defines the concept of communities, so it must be done carefully.

Ideally we would like to see that in a community there are more links inside the community than links that starts from inside and go outside.

The minimization of the cut problem is known as the min-cut problem.

5.2.1 Cut function

Let N_1, N_2, \dots, N_q be a note partition in q communities. Each node of the graph belongs to one and only one community. We can define the graph cut as the number of edges that are between two different communities:

$$cut(N_1, \dots, N_q) = \frac{1}{2} \sum_{a=1}^q \sum_{i \in N_a} \sum_{b \neq a} \sum_{j \in N_b} A_{ij} = L - \sum_{i \in N} \sum_{j \in N} A_{ij} \delta_{l(i), l(j)} \quad (9)$$

Where $l(i)$ is the community of node i, and $l(j)$ is the community of node j. The $\delta_{l(i), l(j)}$ is a Kronecker delta, which in this case is a function that evaluates 1 if both nodes are in the same community, and zero otherwise. Unfortunately, this first approximation of a function, is not the most efficient function in creating partitions, as it tends to isolate single nodes from the rest of the network.

5.2.2 The ratio cut

The ratio cut is defined by counting the average number of connections each node has with nodes in another community:

$$Rcut(N_1, \dots, N_q) = \frac{1}{2} \sum_{a=1}^q \frac{1}{|N_a|} \sum_{i \in N_a} \sum_{b \neq a} \sum_{j \in N_b} A_{ij} \quad (10)$$

This definition would appear to be a better one, unfortunately, to calculate the ratio cut of a network, we would need to check all possible configuration, making it a NP complete problem. Moreover, the ratio cut tends to favor configurations in which all communities have the same average size.

5.2.3 Modularity optimization

The ratio cut however generates a set of partitions, but how can we evaluate the goodness of the solution retrieved by the ratio cut algorithm? To answer this question we can use a measure called modularity defined in Section 2.8.1.

The modularity is obtained by comparing the cut with expected cut in a random graph ($\mathbb{E}[cut']$) as follows:

$$mod(N_1, \dots, N_q) = \frac{cut'(N_1, \dots, N_q) - cut(N_1, \dots, N_q)}{2L} = \frac{1}{4L} \sum_{i,j \in N} \left(A_{ij} - \frac{k_i k_j}{2L} \right) \delta_{l(i), l(j)}$$

Where the expected cut is formally defined like below:

$$\mathbb{E}[(cut'(N_1, \dots, N_q))] = L - \sum_{i \in N} \sum_{j \in N} \mathbb{E}[A'_{ij}] \delta_{l(i), l(j)} = L - \sum_{i \in N} \sum_{j \in N} \frac{k_i K_j}{2L} \delta_{l(i), l(j)} \quad (11)$$

Empirically, high modularity values (approx 0,3 - 0,4) corresponds to "good" partitions.

The optimization of the modularity is one of the most popular approaches to perform community detection on graphs. This method has, however, several known problems that we here summarize:

- The modularity optimization is capable of finding communities in a structureless graph. Simply looking at the modularity value is dangerous.
- The most common way of optimizing the modularity considers the number of communities q as a variable of the optimization. This is an improvement over the cut approach. It can be shown however that the modularity is not strictly comparable for different values of q and this approach may lead to obtaining too many or too few communities.
- This optimization problem is also NP hard. There exist approximate but efficient methods to optimize it, but, in typical settings, the landscape of the cost function is very rough and it is nearly impossible to find the actual minimum.

5.3 Communities in structure-less graphs

The approach on which the modularity maximization is based on relies in comparing a signal (everything you extract from data) (the community partition on the graph) on a random realization thereof. This is a standard and solid statistical method, but if we optimize over it, we risk overfitting and obtain a node partition that is "over optimistic".

The modularity is not directly comparable for different values of q and this typically leads to be overconfident and split the graph in many small pieces. However, the opposite can happen as well and it is known as the **resolution limit of modularity maximization** where clusters with size $< \sqrt{2L}$ aren't considered. This problem is solved introducing a regularization parameter called resolution parameter γ :

$$mod(N_1, \dots, N_q) = \frac{1}{4L} \sum_{i,j} \left(A_{ij} - \gamma \frac{K_i K_j}{2L} \right) \delta_{l(i), l(j)}$$

5.4 Solve the NP hard problem of identifying communities

One of the most popular (and fast) solutions to optimize the modularity is the Louvain algorithm (more recently improved by the Leiden algorithm). We give each node a different community label. We then iterate two steps until convergence:

1. For all nodes, consider the modularity gain obtained by moving that node to the community of one of its neighbors. After that, change its community label to the class that produces the highest gain of the modularity.
2. After this operation is done for all nodes, merge all nodes in the same community reweighing the edges and start over with the smaller graph.

This method is very fast and gives a solution in $O(n \log(n))$ steps. This strategy is, however largely suboptimal and it is very likely to end up in a local extreme of the optimization function. To be more quantitative, depending on the graph structure, this may become exponentially likely. As a consequence, different runs of the same algorithm on the same graph may produce node partitions that have almost nothing in common, but that have very similar (and potentially high) modularity values. The Louvain algorithm is hence powerful and fast, but is must be used with great care.

Defining communities as the solution of an optimization problem is probably the most straightforward strategy. Among the major pitfalls associated to this class of algorithms we recall that:

- The role played by the number of communities q is not clear and may lead to over or down partitioning
- We only introduced few cost functions but none of them is problem-free. A typical approach consists in changing the cost function to prevent bad behaviors. But in the end, what is the good function to optimize.
- In most of the cases, the cost function to optimize is NP hard and, even though efficient approximations exist, they are only approximations

If your goal is truly to optimize a cost function, then this is undeniably a good, fast and well studied approach. But be aware of all the problems that may come with it. The inference algorithm can still be applied, but the most common issue is associated to convergence problems. This is due to the fact that the DCSBM has a low clustering coefficient while real world networks have a lot of triangles. The consequence is that, even if we have provable guarantees of convergence of the algorithm on a DCSBM graph, for an arbitrary initial condition there may be convergence problems on a real graph.

5.5 Bayesian approach: DCSBM (Degree Corrected Stochastic Block Model)

Sometimes (see the "let's play a game example on the slides) a randomized approach yields better results than optimizations approaches. We use the Bayesian inference to generate a new model called Bayesian stochastic block model that associate a probability that each node belongs to a given community. Consequently, if we look for q communities in an Erdos-Renyi random graph, we will obtain for each node probabilities that are very close to $\frac{1}{q}$ to belong to any cluster. Bayes inference is optimal when the parameters of the model (p_{in} , p_{out} , $q\dots$) are known. This is not a reasonable assumption in general, but there are ways to learn them or to choose them well to perform "mismatched inference". The probabilities can be estimated in polynomial time with respect to n (i.e. Montecarlo algorithm). In general, the optimization approach is a "corner case" of the Bayesian approach, hence, it is likely to give worse results. However, in practical settings, one must keep speed and stability into account and be able to interpret the results of the algorithm.

5.6 Data Clustering

Community detection is a special version of the much more general problem of data clustering. There are two main classes of algorithms for data clustering:

- **hierarchical clustering**: which delivers a nested series of partitions,
- **partitional clustering**: which yields only one partition.

Hierarchical clustering is much more frequently adopted in network community detection than partitional clustering.

The *similarity measure* between nodes is fundamental to data clustering. Such a measure may be derived from specific properties of the nodes. If the nodes can be embedded in a geometric space, the distance between the points corresponding to a pair of nodes can be used as a dissimilarity measure for the nodes, so that points that are nearer to each other indicate more similar nodes. Alternatively, similarity measures can be derived from the structure of the network alone. A classic example is structural equivalence, which expresses the similarity between the neighborhoods of a pair of nodes.

The similarity S_{ij}^{SE} of a pair of nodes i and j via structural equivalence can be defined as:

$$S_{ij}^{SE} = \frac{\text{number of neighbors shared by } i \text{ and } j}{\text{total number of nodes neighboring only } i, \text{ only } j, \text{ or both}}$$

The next step is to define the similarity between two groups of nodes. This can be done in several ways. The most popular approaches are:

- **Single linkage** uses the maximum pairwise similarity: $S_{G_1 G_2} = \max_{i,j} S_{ij};$
- **Complete linkage** uses the minimum pairwise similarity: $S_{G_1 G_2} = \min_{i,j} S_{ij};$
- **Average linkage** uses the average pairwise similarity: $S_{G_1 G_2} = \langle S_{ij} \rangle_{i,j}.$

In these procedures, the similarity between two groups is determined via the similarity scores of pairs of nodes, where each pair consists of one node in each group.

5.6.1 Hierarchical Clustering

Hierarchical clustering techniques are:

- **agglomerative**: if partitions are generated by iteratively merging groups of nodes;
- **divisive**: if partitions are generated by iteratively splitting groups of nodes.

Agglomerative hierarchical clustering starts from the trivial partition into N groups, where each group consists of a single node. At each step, the pair of groups with the largest similarity are merged. This is repeated until all nodes are in the same group. Since at each step the number of groups decreases by one, the procedure yields a series of N partitions, which can be represented via a dendrogram, or hierarchical tree (Figure 32).

To single out one of the partitions, we cut the dendrogram with a horizontal line, as shown in the Figure 32. The vertical lines severed by the cut indicate the clusters of the partition. High cuts yield partitions into a few larger groups, whereas low cuts yield partitions into many smaller groups. The partitions are hierarchical by construction: if we take any two partitions, every cluster of the one lying higher in the dendrogram is a merger of clusters of the lower one.

Hierarchical clustering summary	
Pros	Cons
<p>Real graph are typically structured into hierarchical communities and this approach allows one to unveil this structure.</p> <p>This concept of distance leverages on the relation between weak ties and local bridges.</p>	<p>It delivers as many partitions as there are nodes, without providing a criterion that helps to choose which ones are meaningful for the network under study.</p> <p>The results usually depend on the similarity measure and on the criterion adopted to compute the similarity of the groups.</p> <p>The algorithms are rather slow, and networks with millions or more nodes are out of reach</p>

Table 2: Hierarchical clustering summary, pros and cons.

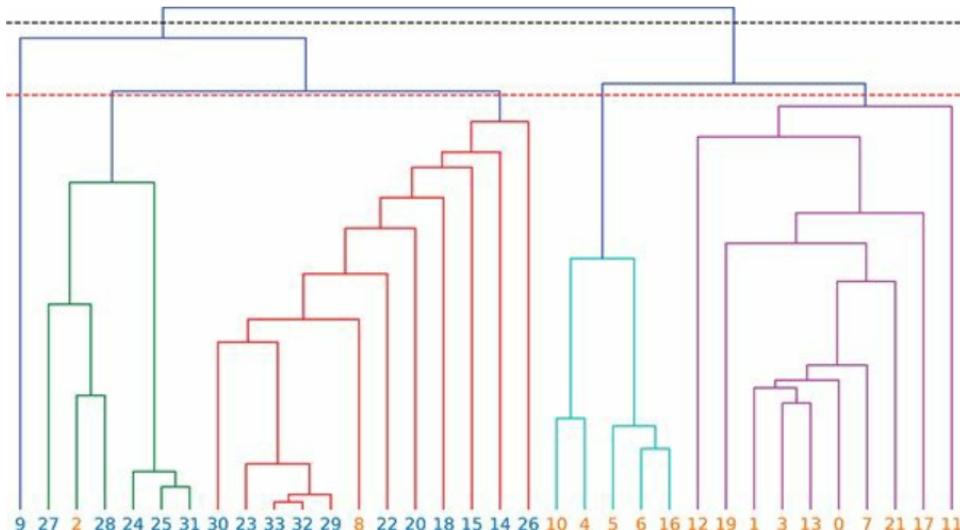


Figure 32: Hierarchical clustering illustrated by a dendrogram of hierarchical partitions of Zachary's karate club network. Horizontal cuts single out partitions of the network.

5.6.2 K-means

K-means clustering is a method of vector quantization that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid). The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum.

The most common algorithm uses an iterative refinement technique. Due to its ubiquity, it is often called "the

k-means algorithm”; it is also referred to as Lloyd’s algorithm, particularly in the computer science community. It is sometimes also referred to as ”naive k-means”, because there exist much faster alternatives. It consists of the following steps:

1. Initialize the label assignment at random
2. Iterate until convergence:
 - Compute the center of each cluster
 - Compute the distance of each node from each cluster center

Given a set of observations (x_1, x_2, \dots, x_n) , where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k (\leq n)$ sets $S = S_1, S_2, \dots, S_k$ to minimize the within-cluster sum of squares (i.e. variance). Formally, the objective is to find:

$$\begin{aligned} \arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 &= \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \operatorname{Var} S_i \\ \boldsymbol{\mu}_i &= \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}, \end{aligned}$$

$|S_i|$ is the size of S_i , and $\|\cdot\|$ is the usual L2 norm. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \frac{1}{|S_i|} \sum_{\mathbf{x}, \mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2$$

The equivalence can be deduced from identity $|S_i| \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \frac{1}{2} \sum_{\mathbf{x}, \mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2$. Since the total variance is constant, this is equivalent to maximizing the sum of squared deviations between points in different clusters (between-cluster sum of squares, BCSS).

5.6.3 Spectral clustering

This method consists in creating a node embedding using the eigenvectors of appropriate graph matrix representations. The algorithm is made of three steps:

1. Build the matrix M
2. Compute q eigenvectors of M and store them in a matrix X , where each row of X corresponds to a node.
3. Use the k-means on X

5.6.4 Spectral clustering with the graph Laplacian

Be A the adjacency matrix of a connected graph, and D the diagonal degree matrix. We define the graph Laplacian matrix L as:

$$L = D - A$$

Property: consider an arbitrary vector y , then:

$$y^T L y = \frac{1}{2} \sum_{i,j} A_{ij} (y_i - y_j)^2$$

The projection of y over L tells us how fast y changes on the graph, hence if y is the community assignment it should change slowly.

But what is the relation with the community detection? if we recall the ratio cut problem, by doing the Spectral clustering with the graph Laplacian, we are minimizing the ratio cut.

1. Build the matrix L from the graph
2. Compute the q eigenvectors associated to the q smallest eigenvalues of L

3. Store them in a matrix of size (n,q)
4. Perform k-means on the rows of X
5. Return the labels

This algorithm runs in $\mathcal{O}(Lq^2)$ operations.

5.6.5 The random walk Laplacian

Let us take another approach to introduce a popular spectral clustering algorithm. Suppose that v_i is a value associated to node i. At each iteration all nodes update their value v_i with a value proportional to the average of their neighbors and iterate until convergence. Mathematically this can be written as:

$$v_i^{(t+1)} = \frac{\lambda}{K_i} \sum_{j \in \mathbb{N}} A_{ij} v_j^t = \lambda \left(D^{-1} A v^{(t)} \right)$$

The convergence imposes that $v_i^{t+1} = v_i^t$ and so that v is an eigenvector of the random walk Laplacian.

1. Build the matrix L_{rw} from the graph
2. Compute the q eigenvectors associated to the q largest eigenvalues of L
3. Store them in a matrix of size (n,q)
4. Perform k-means on the rows of X
5. Return the labels

This algorithm runs in $\mathcal{O}(Lq^2)$ operations.

5.6.6 Spectral clustering and the number of communities

Spectral clustering is a way to evaluate the number of clusters in a network. It is based on two definitions:

- **Isolated eigenvalue:** the distance from the closest eigenvalue is independent from the graph size
- **Bulk eigenvalue:** the distance from the closest eigenvalue vanishes with the graph size.

If we were to plot the distribution of the eigenvalues, we get this graph, in which the pink bars are bulk eigenvalues and black are isolated eigenvalues.

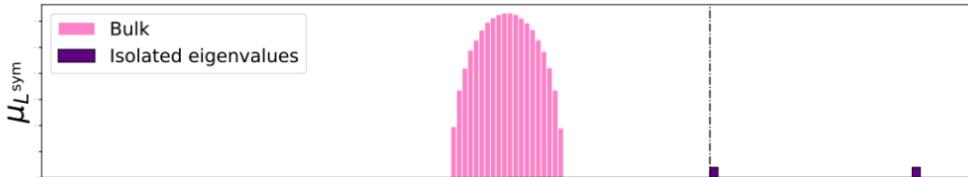


Figure 33

As a rule of thumb **the number of isolated eigenvalues, is the number of communities in the network.** This is due to this equation:

$$M = Q + \Omega$$

where

- M is the matrix used to do spectral clustering
- Q is the expectation of M . This is a matrix of rank q (number of communities) that contains all the information of the graph structure. This matrix give rises to the isolated eigenvalues.
- Ω is the noisy part of the spectrum and generates the bulk eigenvalues, which we can ignore.

Spectral clustering summary	
Pros	Cons
<p>This type of algorithms does not require any iterative process and eigenvectors can be computed exactly</p> <p>Several choices of M exist and they are connected to both optimization and Bayesian approaches, providing fast and good approximations</p> <p>They have a very solid theoretical background and are well understood, They provide ways to estimate the number of communities and They are model-free</p>	<p>The complexity scales with q^2 and they are unsuited for graphs with a lot of communities.</p> <p>They do not provide a hierarchical solution, They hardly are optimal: there is typically some (slower) algorithm that performs better</p> <p>Most of the known results are studied for synthetic graphs. On real-world graphs things are still well defined, but more complicated, and You do not output a node partition but an embedding, hence you need kmeans</p>

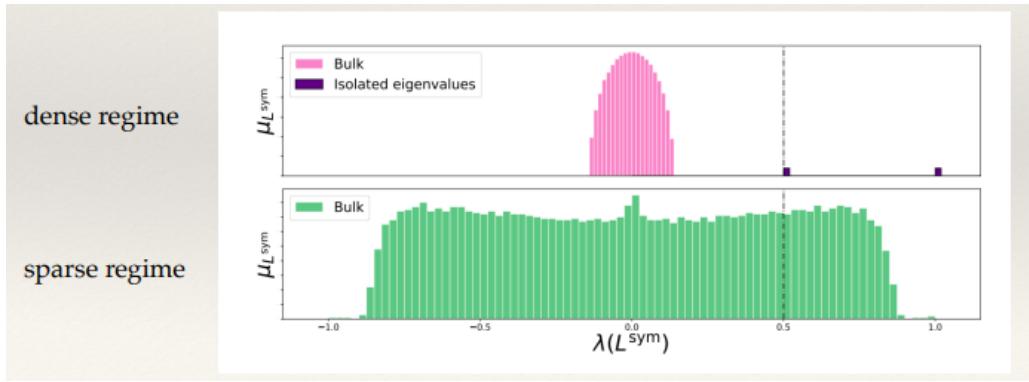


Figure 34: Spectral clustering for sparse and dense networks.

5.6.7 Sparsity in spectral clustering

Sparsity has always been the Achille heels for spectral clustering: without enough edges, the noise swallows the informations contained inside the matrix Q , making the graphs absolutely useless. To solve this problem, a peculiar matrix is used, known as random walk Laplacian matrix, defined as:

$$L_\tau = D_\tau^{-1} A \quad D_\tau = D + \tau I_n \quad \tau = \frac{\langle k^2 \rangle}{\langle k \rangle} - 1$$

5.6.8 Node embeddings

Node embeddings are useful in community detection where we initially represent each node of the graph as a vector in d dimensions $i \rightarrow x_i \in \mathbb{R}^d$, then we cluster the points in the d dimensional space with (for instance) k -means. Mapping the graph to a text and exploits a word embedding algorithm is what the DeepWalk algorithm does.

Once the embedding has been obtained one can do several things, among which, community detection by simply performing clustering on the embedded space. We can summarize the basics of the DeepWalk algorithm as follows:

1. Create a several random walk on the graph. Each of them is a sentence of the text
2. Given the "text" use Word2Vec to obtain the embedding of the words (aka the nodes)

5.6.9 Word2Vec

Word2Vec is a technique used to mathematically encode words, the basic idea of that is to define a set of context words, for each word, in the text and train an algorithm so to obtain similar representations for words that are in similar contexts.

Word2Vec is based on the skip-gram model, where: given a window size, all words within a distance from a central word are said to be in its context and the central word is "skipped".

1. We define an input embedding x and an output (context) embedding y for each word of the vocabulary.
2. For each word in the text we obtain its context words and define an objective function that aligns the input embedding of that word (i) with the output embedding of the context words (k). Recall the sigmoid function.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad c_1 = \log \sigma(x_i^T y_k)$$

3. Now, draw a number of negative samples at random from the dictionary and define an objective function that prevents the embeddings alignment. This approach recalls the one of the modularity in which we maximize over the actual realization and minimize over the randomization.

$$c_2 = \sum_{t=1}^q \mathbb{E}_{k \sim P_{neg}} [\log \sigma(-x_i^T y_k)]$$

4. Optimize over all the words in text using stochastic gradient descent
5. Use the embedding X obtained from the previous step to represent words

An alternative version of DeepWalk exists and is called Node2Vec. In this case we have two parameters p, q that allow us to performed biased random walks on the network. For $p = q = 1$ the Node2Vec algorithm behave like the DeepWalk algorithm.

DeepWalk summary	
Pros	Cons
It has a strong connection to variational Bayesian inference and, in general, it does not lead to overfitting It does not have major convergence issues on real graphs It highly parallelize The complexity to produce the embedding is independent of the number of communities and it can be used for very large graphs	The number of communities must not be specified before, so it has to be figured out from the embedding. Higher embedding dimensions generally give better results but at a higher computational cost Even if it is a "competing" algorithm, it is not optimal in terms of performance

6 Page Rank

PageRank is an algorithm, or procedure, to compute a centrality measure that aims to capture the prestige or importance of each node; it is typically used in directed networks. It is also the name we give to the centrality measure itself. So when applied to the Web, the algorithm assigns each page a PageRank value. The ranking algorithm of a search engine can then use this value, in combination with many other factors, such as the match between query and page text, to sort the results of a query. A page with high PageRank is considered prestigious or important, and is given a boost by the ranking algorithm: other things being equal, pages with larger PageRank are ranked higher.

6.1 Link Analysis using Hubs and Authorities

6.1.1 Voting by In-Links

This is the first approach to page ranking. The basic idea of the "in-links" approach is that if a page receives many links from other relevant pages, then it receives a kind of collective endorsement.

So, we can use the in-links to determinate the authority of a page on a topic, through the implicit endorsements that other pages on the topic confer through their links to it. Of course, each individual link may have many possible meanings: it may be off-topic; it may convey criticism rather than endorsement; it may be a paid advertisement. It is hard for search engines to automatically determinate the intent of each link.

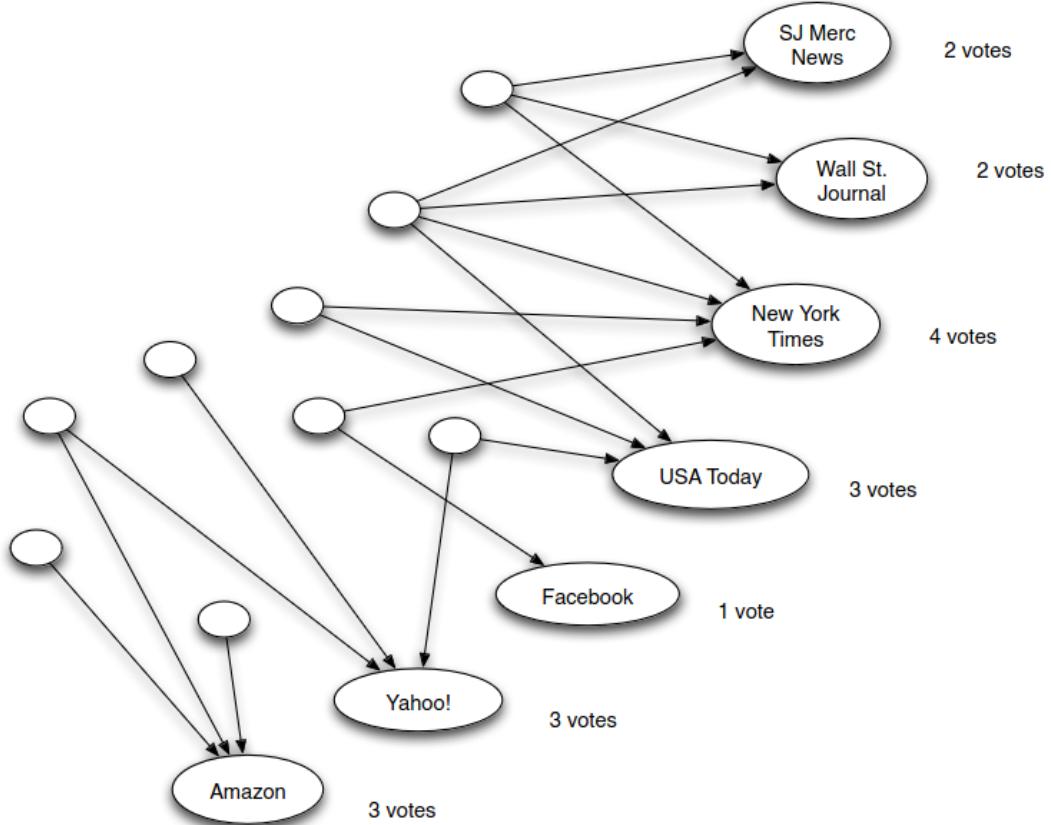


Figure 35: Counting in-links to pages for the query “newspapers”.

6.1.2 Hubs and Authorities

Before discussing the HITS algorithm, let us define Hubs and Authorities as follows:

- **Hubs:** Hubs can be thought of as web pages that serve as valuable resources or directories, providing links to other relevant pages on a specific topic. Hubs are pages that tend to contain a large number of outgoing links, pointing to other pages that are considered authoritative or informative within the same subject area. They act as gateways to further information and resources. Hubs help users navigate the web by directing them to authoritative sources.
- **Authority:** Authority refers to the quality and trustworthiness of a web page within a given topic. An authoritative page is one that is highly regarded and recognized as a reliable source of information. Authority is determined by the number and quality of incoming links from other pages within the same subject area. The more incoming links a page receives from reputable sources or hubs, the higher its authority score becomes. Pages with high authority are considered to have valuable and reliable content related to the topic.

For each page p , we are trying to estimate its value as a potential authority and as a potential hub, and so we assign it two numerical scores: $auth(p)$ and $hub(p)$. Each of these starts out with a value equal to 1. Update rules:

- **Authority Update Rule:** For each page p , update $auth(p)$ to be the sum of the hub scores of all pages that point to it.
- **Hub Update Rule:** For each page p , update $hub(p)$ to be the sum of the authority scores of all pages that it points to.

HITS algorithm:

- We start with all hub scores and all authority scores equal to 1.
- We choose a number of steps k .
- We then perform a sequence of k hub-authority updates. Each update works as follows:
 - First apply the Authority Update Rule to the current set of scores.
 - Then apply the Hub Update Rule to the resulting set of scores.
- At the end, the hub and authority scores may involve numbers that are very large. But we only care about their relative sizes, so we can normalize to make them smaller: we divide down each authority score by the sum of all authority scores, and divide down each hub score by the sum of all hub scores.

For larger and larger values of k the results stabilize so that continued improvement leads to smaller and smaller changes in the values we observe. The studies about this argument shows that we reach the same limiting values no matter what we choose as the initial hub and authority values, provided only that all of them are positive. In other words, the limiting hub and authority values are a property purely of the link structure, not of the initial estimates we use to start the process of computing them.

6.2 Page rank algorithm

Endorsement is best viewed as passing directly from one prominent page to another — in other words, a page is important if it is cited by other important pages. And it is this mode of endorsement that forms the basis for the PageRank measure of importance.

As with hubs and authorities, the intuition behind PageRank starts with simple voting based on in-links, and refines it using the Principle of Repeated Improvement. In particular, the Principle is applied here by having nodes repeatedly pass endorsements across their out-going links, with the weight of a node's endorsement based on the current estimate of its PageRank: nodes that are currently viewed as more important get to make stronger endorsement.

Page rank is computed from the link structure of a web graph, with an iterative process called the power method. The algorithm works as follows:

- The rank of each node i , is initialized to $\frac{1}{N}$, where N in the number of pages.
- At each step, the rank is updated until convergence (that is the value does not change from iteration i and iteration $i - 1$) wit the following equation:

$$R_t(i) = \frac{\alpha}{N} + (1 - \alpha) \sum_{j \in pred(i)} \frac{R_{t-1}(j)}{k_j^{out}} \quad (12)$$

We assume that random jumps occur with a probability expressed with a parameter α , called the **teleportation factor**. The parameter $(1 - \alpha)$ is called damping factor and it is associated with the random walk process.

- According to the page rank algorithm, at every step, with probability α a user jumps to a node selected at random among all pages, and with probability $1 - \alpha$ the user continues to browse the internet jumping to another page from a link in the current page;

Another possible formulation of the page rank algorithm exploits matrices and eigenvalues, and works as follows:

- A matrix $N^{n \times n}$, where n is the number of nodes in the graph, is initialized as follows:

$$N_{i,j} = \begin{cases} 0 & \text{if no link is present from node } i \text{ to node } j \\ \frac{1}{k_i^{out}} & \text{otherwise} \end{cases}$$

- We then now represent the page rank, with a vector r , where $r_i = \text{Page rank of node } i$.
- With this notation, we can calculate the page rank at time t as:

$$r^{(t)} = N^T \cdot r^{(t-1)}$$

- In case we want to use the scaled page rank, the update rule can be expressed with a slightly different notation of the matrix N :

$$\tilde{N}_{i,j} = sN_{i,j} + \frac{1-s}{n}$$

Where s is the dumping factor (usually chosen between 0.8 and 0.9), instead n , is the number of nodes in the graph. The page rank update rule at time t , becomes:

$$r^{(t)} = \tilde{N}^T r^{(t-1)}$$

Notice that the total PageRank in the network will remain constant as we apply these steps: since each page takes its PageRank, divides it up, and passes it along links, PageRank is never created nor destroyed, just moved around from one node to another. As a result, we don't need to do any normalizing of the numbers to prevent them from growing, the way we had to with hub and authority scores.

The basic definition of PageRank suffers of cycles that cause "slow-leak", to get around to this problem, one natural way is to modify the definition of PageRank using a scaling factor (also called dumping factor). The basic idea is defined as follows: we pick a scaling factor s that should be strictly between 0 and 1. We then replace the Basic PageRank Update Rule with the following:

Scaled PageRank Update Rule: First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of s . This means that the total PageRank in the network has shrunk from 1 to s . We divide the residual $1 - s$ units of PageRank equally over all nodes, giving $\frac{(1-s)}{n}$ to each. Usually the value of s is between 0.8 and 0.9.

We now describe an equivalent formulation of PageRank using the random walk analogy.

The algorithm starts by choosing a page at random, picking each page with equal probability. It then follows links for a sequence of k steps: in each step, picks a random out-going link from their current page, and follow it to where it leads. Such an exploration of nodes performed by randomly following links is called a random walk on the network.

The analysis in terms of random walks provides some additional intuition for PageRank as a measure of importance: the PageRank of a page X is the limiting probability that a random walk across hyperlinks will end up at X , as we run the walk for larger and larger numbers of steps.

As we did before for the classic page rank definition, we can also formulate the Scaled PageRank Update Rule in terms of random walks.

Rather than simply following a random edge in each step, the walker performs a "scaled" version of the walk as follows: With probability s , the walker follows a random edge as before; and with probability $1 - s$, the walker jumps to a random node anywhere in the network, choosing each node with equal probability.

7 Information cascading

7.1 Diffusion in Network

Let's consider how new behaviors, practices, opinions, conventions, and technologies spread from person to person through a social network, as people influence their friends to adopt new ideas.

These studies were done in the 20th century and are called "diffusion of innovation".

7.2 Modeling diffusion through networks

We are under the assumption that individuals make decisions based on the choices of their neighbors (as such we focus on links). This model have been introduced by Stephen Morris in 2000.

In our model, each node has a choice between two possible behaviors, and players have an incentive to adopt the same behavior: for each node, let p be the fraction of neighbors of node i that choose behavior A, $1 - p$ the fraction of neighbors of node i that chose behavior B, d is the number of neighbors of node i , then the node will choose behavior A if

$$pda \geq (1-p)db$$

or, rearranging terms, if:

$$p \geq \frac{b}{a+b} \quad (\text{called threshold})$$

where pda and $(1-p)db$ is the payoff.

In practice, this is actually simpler: for each node, that node will choose the behavior that most of its neighbors have.

Of course a problem arises: in this very simple model, since the equation has a greater than equal symbol, one opinion will always be prevalent over the other, and no equilibrium will be reached: either all nodes will choose A or all other nodes will choose B.

7.3 Cascading behavior

In the previous illustrated model only two equilibrium exists:

- One in which everyone adopts A
- One in which everyone accepts B

we would like to understand how easy is to get one of these equilibrium, and also understand if other intermediate equilibrium exists and how they look like. We make the following assumptions from now on:

- Everyone is using B at the beginning;
- A: is a set of Small initial adopters of A;

Will the spread of A, make everyone to switch to the new technology, or will the spread stop? The answer is actually simple: it depends on the network structure, the choice of nodes in S, and the value of q .

Consider a set of initial adopters who start with a new behavior A, while every other node starts with behavior B. Nodes then repeatedly evaluate the decision to switch from B to A using a threshold of q . If the resulting cascade of adoptions of A eventually causes every node to switch from B to A, then we say that the set of initial adopters causes a complete cascade at threshold q .

7.4 Cascades and Clusters

Homophily can serve as a barrier to diffusion: it is hard for innovation to arrive from outside densely connected communities, more formally:

Definition 7.1 *A cluster of density p is a set of nodes C where each node in the set has at least p fraction of edges in C.*

Each node in a cluster has a prescribed fraction of its friends residing in the cluster: cohesion. Nodes in the same clusters do not necessarily have much in common. Any network is a cluster of density. The union of two clusters of density p is still a cluster of density p . In fact, clusters in networks can exist simultaneously at different scales.

A cascade stops if and only if it runs into a dense cluster: clusters are natural obstacles to cascades.

Let S be the set of initial adopters of A, with a threshold of q :

- if the remaining network contains a cluster of density $> (1 - q)$, then S cannot cause a complete cascade
- if S fails to cause a cascade, then there is a cluster of density $> (1 - q)$ in the remaining network

Threshold models highlight some important implications of "the strength of weak ties" theory: they receive very fresh ideas from other communities. Bridges and weak ties are great for spreading rumors or jokes across the network, but not for diffusion of innovation or social mobilization. Strong ties can have more significant role for others in the community to take actions.

7.5 Diffusion, Thresholds, and the Role of Weak Ties

One of the fundamental things we learn from studying diffusion is that there is a crucial difference between learning about a new idea and actually deciding to adopt it.

Centola and Macy and Siegel make the interesting observation that threshold models for diffusion thus highlight an interesting subtlety in the strength-of-weak-ties theory. Weak ties provide access to sources of information that reside in parts of the network we otherwise wouldn't have access to. But things look very different if we consider the spread of a new behavior that requires not just awareness, but an actual threshold for adoption. They are powerful ways to convey awareness of new things, but they are weak at transmitting behaviors that are in some way risky or costly to adopt (behaviors where you need to see a higher threshold of neighbors doing it before you do it as well).

7.6 Extension of the basic cascade model

We have been keeping the underlying model of individual behavior as simple as possible: everyone has the same payoffs, and the same intensity of interaction with their network neighbors. We can make these assumptions more general while still preserving the structure of the model and the close connection between cascades and clusters.

As the main generalization we consider, suppose that each person in the social network values behaviors A and B differently.

Almost all the previous analysis carries over with only small modifications. For each node v , we define a payoff a_v , labeled so that it is specific to v , that it receives when it coordinates with someone on behavior A, and we define a payoff b_v that it receives when it coordinates with someone on behavior B. If v has d neighbors, of whom a p fraction have behavior A and a $(1 - p)$ fraction have behavior B, then the payoff from choosing A is pda_v while the payoff from choosing B is $(1 - p)db_v$. Thus, A is the better choice if:

$$p \geq \frac{b_v}{a_v + b_v}$$

Again, clusters block cascading behavior, but in this case we have a slightly different definition. Cluster in heterogeneous case is defined as follows:

Definition 7.2 (Blocking Cluster) *Given a set of node thresholds, let's say that a blocking cluster in the network is a set of nodes for which each node v has more than a $1 - q_v$ fraction of its friends also in the set.*

8 Game Theory

Game theory is concerned with situations in which decision-makers interact with one another, and in which the happiness of each participant with the outcome depends not just on his or her own decisions but on the decisions made by everyone.

A game is any situation with the following three aspects.

1. There is a set of participants, whom we call the players.
2. Each player has a set of options for how to behave; we will refer to these as the player's possible strategies.
3. For each choice of strategies, each player receives a payoff that can depend on the strategies selected by everyone. The payoffs will generally be numbers, with each player preferring larger payoffs to smaller payoffs.

8.1 Underlying Assumptions

In order to make this question tractable, we will make a few assumptions:

1. We assume everything that a player cares about is summarized in the player's payoffs.
2. We also assume that each player knows everything about the structure of the game. To begin with, this means that each player knows his or her own list of possible strategies.
3. We suppose that each individual chooses a strategy to maximize her own payoff, given her beliefs about the strategy used by the other player. This model of individual behavior, which is usually called *rationality*, actually combines two ideas:

- (a) Each player wants to maximize her own payoff. Since the individual's payoff is defined to be whatever the individual cares about, this hypothesis seems reasonable.
- (b) Each player actually succeeds in selecting the optimal strategy.

8.2 Best Responses and Dominant Strategies

The first strategy is characterized by two approaches:

1. **Best response:** it is the best choice of one player, given a belief about what the other player will do. We can make this precise with a bit of notation, as follows. If S is a strategy chosen by Player 1, and T is a strategy chosen by Player 2, then there is an entry in the payoff matrix corresponding to the pair of chosen strategies (S, T) . We will write $P_1(S, T)$ to denote the payoff to Player 1 as a result of this pair of strategies, and $P_2(S, T)$ to denote the payoff to Player 2 as a result of this pair of strategies. Now, we say that a strategy S for Player 1 is a best response to a strategy T for Player 2 if S produces at least as good a payoff as any other strategy paired with T :

$$P_1(S, T) \geq P_1(S', T)$$

for all other strategies S' of Player 1. There is a completely symmetric definition for Player 2.

Notice that this definition allows for multiple different strategies of Player 1 to be tied as the best response to strategy T . This can make it difficult to predict which of these multiple different strategies Player 1 will use. We can emphasize that one choice is uniquely the best by saying that a strategy S of Player 1 is a strict best response to a strategy T for Player 2 if S produces a strictly higher payoff than any other strategy paired with T :

$$P_1(S, T) > P_1(S', T)$$

for all other strategies S' of Player 1. When a player has a strict best response to T , this is clearly the strategy she should play when faced with T .

2. **Strictly dominant strategy:** we can formulate its definition in terms of best responses as follows:

- (a) We say that a dominant strategy for Player 1 is a strategy that is a best response to every strategy of Player 2.
- (b) We say that a strictly dominant strategy for Player 1 is a strategy that is a strict best response to every strategy of Player 2.

If a player has a strictly dominant strategy, then we can expect him or her to use it. The notion of a dominant strategy is slightly weaker, since it can be tied as the best option against some opposing strategies. As a result, a player could potentially have multiple dominant strategies, in which case it may not be obvious which one should be played.

8.3 Nash Equilibrium

When neither player in a two-player game has a strictly dominant strategy, we need some other way of predicting what is likely to happen. In 1950, John Nash proposed a simple but powerful principle for reasoning about behavior in general games, and its underlying premise is the following: *even when there are no dominant strategies, we should expect players to use strategies that are best responses to each other.*

More precisely, suppose that Player 1 chooses a strategy S and Player 2 chooses a strategy T . We say that this pair of strategies (S, T) is a Nash equilibrium if S is a best response to T , and T is a best response to S .

This is not a concept that can be derived purely from rationality on the part of the players; instead, it is an equilibrium concept. The idea is that if the players choose strategies that are best responses to each other, then no player has an incentive to deviate to an alternative strategy, so the system is in a kind of equilibrium state, with no force pushing it toward a different outcome.

8.4 Mixed Strategies

We extend the set of strategies to include the possibility of randomization; once players are allowed to behave randomly, one of John Nash's main results establishes that there are always equilibria.

Let H and T be two possible strategies that each player has.

The simplest way to introduce randomized behavior is to say that each player is not actually choosing strategy H or T directly, but rather is choosing a probability with which she will play H . So in this model, the possible

strategies for Player 1 are numbers p between 0 and 1; a given number p means that Player 1 is committing to play H with probability p , and T with probability $1 - p$. Similarly, the possible strategies for Player 2 are numbers q between 0 and 1, representing the probability that Player 2 will play H .

Since a game consists of a set of players, strategies, and payoffs, we should notice that by allowing randomization, we have actually changed the game. It no longer consists of two strategies by each player, but instead a set of strategies corresponding to the interval of numbers between 0 and 1. We will refer to these as mixed strategies, since they involve “mixing” between the options H and T . Notice that the set of mixed strategies still includes the original two options of committing to definitely play H or T ; these two choices correspond to selecting probabilities of 1 or 0 respectively, and they are called *pure strategies* in the game.

With this new set of strategies, we also need to determine the new set of payoffs.

In order to rank random payoffs numerically, we will attach a number to each distribution that represents how attractive this distribution is to the player. Once we have done this, we can then rank outcomes according to their associated number. The number we will use for this purpose is the expected value of the payoff.

For instance, we have this payoff matrix:

		Player 2	
		H	T
		-1,+1	+1,-1
Player 1	H	-1,+1	+1,-1
	T	+1,-1	-1,+1

Table 3: Example of payoff matrix for mixed strategies.

if Player 1 chooses the pure strategy H while Player 2 chooses a probability of q , as above, then the expected payoff to Player 1 is:

$$(-1)(q) + (1)(1 - q) = 1 - 2q$$

Similarly, if Player 1 chooses the pure strategy T while Player 2 chooses a probability of q , then the expected payoff to Player 1 is:

$$(1)(q) + (-1)(1 - q) = 2q - 1.$$

8.4.1 Nash Equilibrium with Mixed Strategies

We define a Nash equilibrium for the mixed-strategy version: it is a pair of strategies (now probabilities) so that each is a best response to the other.

First, let's observe that no pure strategy can be part of a Nash equilibrium. Suppose, for example, that the pure strategy H in Table 3 (i.e. probability $p = 1$) by Player 1 were part of a Nash equilibrium. Then Player 2's unique best response would be the pure strategy H as well (since Player 2 gets +1 whenever he matches). But H by Player 1 is not a best response to H by Player 2, so in fact this couldn't be a Nash equilibrium. Analogous reasoning applies to the other possible pure strategies by the two players. So we reach the natural conclusion that in any Nash equilibrium, both players must be using probabilities that are strictly between 0 and 1.

Next, let's ask what Player 1's best response should be to the strategy q used by Player 2. In Section 8.4, we determined that the expected payoff to Player 1 from the pure strategy H in this case is:

$$1 - 2q$$

while the expected payoff to Player 1 from the pure strategy T is:

$$2q - 1.$$

Now, here's the key point: if $1 - 2q \neq 2q - 1$, then one of the pure strategies H or T is in fact the unique best response by Player 1 to a play of q by Player 2. This is simply because one of $1 - 2q$ or $2q - 1$ is larger in this case, and so there is no point for Player 1 to put any probability on her weaker pure strategy. But we already established that pure strategies cannot be part of any Nash equilibrium for Table 3, and because pure strategies are the best responses whenever $1 - 2q \neq 2q - 1$, probabilities that make these two expectations unequal cannot be part of a Nash equilibrium either.

So we've concluded that in any Nash equilibrium for the mixed-strategy version of Table 3, we must have

$$1 - 2q = 2q - 1$$

or in other words, $q = 1/2$. The situation is symmetric when we consider things from Player 2's point of view, and evaluate the payoffs from a play of probability p by Player 1. We conclude from this that in any Nash equilibrium, we must also have $p = 1/2$.

Thus, the pair of strategies $p = 1/2$ and $q = 1/2$ is the only possibility for a Nash equilibrium. We can check that this pair of strategies in fact do form best responses to each other.

8.5 Pareto-Optimality and Social Optimality

In a Nash equilibrium, each player's strategy is a best response to the other player's strategies. In other words, the players are optimizing individually. But this doesn't mean that, as a group, the players will necessarily reach an outcome that is in any sense good.

8.5.1 Pareto-Optimality

Definition 8.1 (Pareto-Optimality) *A choice of strategies, one by each player, is Pareto-optimal if there is no other choice of strategies in which all players receive payoffs at least as high, and at least one player receives a strictly higher payoff.*

To see the intuitive appeal of Pareto-optimality, let's consider a choice of strategies that is not Pareto-optimal. In this case, there's an alternate choice of strategies that makes at least one player better off without harming any player. In basically any reasonable sense, this alternate choice is superior to what's currently being played. If the players could jointly agree on what to do, and make this agreement binding, then surely they would prefer to move to this superior choice of strategies.

8.5.2 Social Optimality

A stronger condition that is even simpler to state is social optimality.

Definition 8.2 (Social Optimality) *A choice of strategies, one by each player, is a social welfare maximizer (or socially optimal) if it maximizes the sum of the players' payoffs.*

Outcomes that are socially optimal must also be Pareto-optimal: if such an outcome weren't Pareto-optimal, there would be a different outcome in which all payoffs were at least as large, and one was larger — and this would be an outcome with a larger sum of payoffs. On the other hand, a Pareto-optimal outcome need not be socially optimal.

9 Traffic networks

9.1 Traffic at Equilibrium

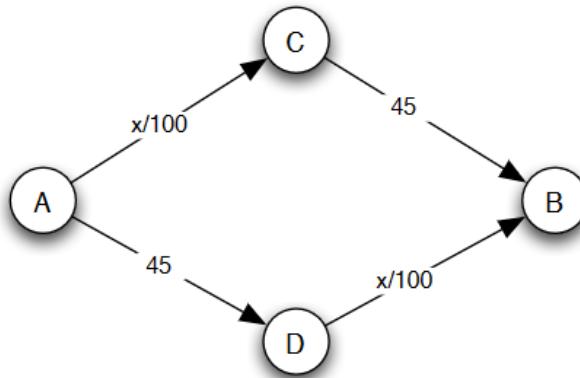


Figure 36: A highway network, with each edge labeled by its travel time (in minutes) when there are x cars using it.

We represent a transportation network by a directed graph: we consider the edges to be highways, and the nodes to be exits where you can get on or off a particular highway. There are two particular nodes, which we'll

call A and B , and we'll assume everyone wants to drive from A to B . For example, we can imagine that A is an exit in the suburbs, B is an exit downtown, and we're looking at a large collection of morning commuters. Finally, each edge has a designated travel time that depends on the amount of traffic it contains.

To make this concrete, consider the graph in Figure 36. The label on each edge gives the travel time (in minutes) when there are x cars using the edge. In this simplified example, the (A, D) and (C, B) edges are insensitive to congestion: each takes 45 minutes to traverse regardless of the number of cars traveling on them. On the other hand, the (A, C) and (D, B) edges are highly sensitive to congestion: for each one, it takes $x/100$ minutes to traverse when there are x cars using the edge.

Now, suppose that 4000 cars want to get from A to B as part of the morning commute. There are two possible routes that each car can choose: the upper route through C , or the lower route through D . For example, if each car takes the upper route (through C), then the total travel time for everyone is 85 minutes, since $4000/100 + 45 = 85$. The same is true if everyone takes the lower route. On the other hand, if the cars divide up evenly between the two routes, so that each carries 2000 cars, then the total travel time for people on both routes is $2000/100 + 45 = 65$.

The traffic model we've described is really a game in which the players correspond to the drivers, and each player's possible strategies consist of the possible routes from A to B . The payoff for a player is the negative of his or her travel time (we use the negative since large travel times are bad). In this traffic game, there is generally not a dominant strategy; for example, in Figure 36 either route has the potential to be the best choice for a player if all the other players are using the other route. However, the game does have Nash equilibria that which is achieved by balancing the two paths.

We just observe that with an even balance between the two routes, no driver has an incentive to switch over to the other route. So, consider a list of strategies in which x drivers use the upper route and the remaining $4000 - x$ drivers use the lower route. Then if x is not equal to 2000, the two routes will have unequal travel times, and any driver on the slower route would have an incentive to switch to the faster one. Hence any list of strategies in which x is not equal to 2000 cannot be a Nash equilibrium; and any list of strategies in which $x = 2000$ is a Nash equilibrium.

9.2 Braess's Paradox

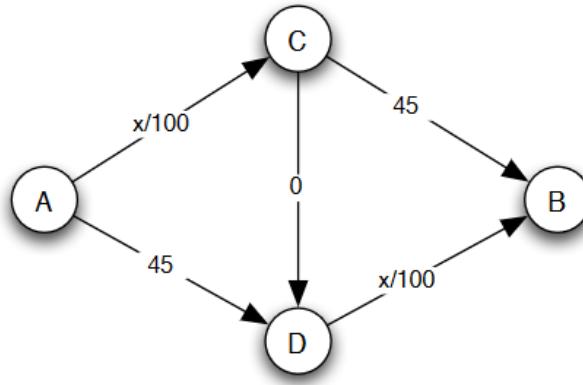


Figure 37: The highway network from the previous figure, after a very fast edge has been added from C to D .

In Figure 36, everything works out very cleanly: self-interested behavior by all drivers causes them, at equilibrium, to balance perfectly between the available routes. But with only a small change to the network, we can quickly find ourselves in truly counterintuitive territory: suppose that the city government decides to build a new, very fast highway from C to D , as indicated in Figure 37. To keep things simple, we'll model its travel time as 0, regardless of the number of cars on it, although the resulting effect would happen even with more realistic (but small) travel times. It would stand to reason that people's travel time from A to B ought to get better after this edge from C to D is added.

Does it?

There is a unique Nash equilibrium in this new highway network, but it leads to a worse travel time for everyone. At equilibrium, every driver uses the route through both C and D ; and as a result, the travel time for every driver is 80 (since $4000/100 + 0 + 4000/100 = 80$). To see why this is an equilibrium, note that no driver can

benefit by changing their route: with traffic snaking through C and D the way it is, any other route would now take 85 minutes. And to see why it's the only equilibrium, you can check that the creation of the edge from C to D has in fact made the route through C and D a dominant strategy for all drivers: regardless of the current traffic pattern, you gain by switching your route to go through C and D .

In the new network there is no way, given individually self-interested behavior by the drivers, to get back to the even-balance solution that was better for everyone.

This phenomenon, that adding resources to a transportation network can sometimes hurt performance at equilibrium, is known as Braess's Paradox.