Network Science Cheatsheet

UNIVERSITÉ DE LYON

Made by
Remy Cazabet

Random Graphs

Many elements of this course are inspired by the excellent classes by Aaron Clauset, than can be found online:
`http://tuvalu.santafe.edu/~aaronc/courses/5352/`

Synthetic networks usages

Using synthetic networks is essential in network science for several reasons. In particular, they allow to:

- Study some properties in a **controlled environment**. *What happens if we increase property X, while keeping all other properties constant?*
- Compare an observed network with a **randomized** version of it. *I observed property X in my data, is it something remarkable, or would I observe the same thing on a random network similar to my graph?*
- Explain a phenomenon. *Property X seems exceptional. It can be reproduced in random networks by simple mechanism Y.*
- Generate synthetic datasets, for instance to test the same algorithm on multiples variations of the same network.

Synthetic networks types

There are three main types of synthetic networks:

- **Deterministic models** are instances of famous graphs or, more commonly, repeated regular patters. e.g., *Caveman graph, grids, lattices.*
- **Generative models** assign to each pair of nodes a probability of having an edge according to their properties (degree, label, etc.). e.g., *Erdős Rényi, Configuration model, etc.*
- **Mechanistic models** create networks by following a set of rules, a process defined by an algorithm. e.g., *Preferential attachment, Forest fire, etc.*

Regular lattices

Regular lattices are defined as repetition of the same pattern a given (potentially infinite) number of times. Nodes all have the same degree. The pattern can be in 1, 2 or more dimensions.
The **clustering coefficient** depends on the structure, it can be large if the structure is made of triangles, for instance. It is the same for all nodes (except potentially nodes at the boundaries).
The **average distance** grows quickly with n, if $k \ll n$

Erdős-Rényi (ER) model

The **Erdős-Rényi (ER)** model is the simplest random graph model. Assuming that we know the number of nodes and the number of edges, and no other information, then edges are simply put between randomly chosen pairs of nodes.
ER models can be defined in two ways:

- in the $G(n, L)$ formulation, the number of edges of the generated graph is set to exactly L, and thus L random pairs of nodes are chosen among the set of all existing node pairs (sharp constraint, microcanonical ensemble).
- in the $G(n, p)$ formulation, an edge is added between any set of node with a probability p. (soft constraint, canonical ensemble).

Properties of both model are similar when the number of edges (defined by L or p) is large.

Random version of observed graph

When one wants to compare a real network with a **randomized** version of it (also called a **rewired** network), the usual way is not to start from the original network and to actually rewire it edge by edge, but instead to generate a new ER random graph keeping the same number of nodes and the same number of edges (or the same density) as the observed network. Properties of the observed network can then be compared with the generated network. Note that it does note make sense to compare the properties of any particular node in both networks, since nodes in the random graph have no identity. For many applications, there is not need to actually generate a random graph: one can simply compare properties of the real network with theoretical properties of the random graph.

Soft ER

In the soft ER, the number of edges is not known in advance. The distribution of the number of edges in the soft ER is described by the **binomial distribution** $\mathbb{B}(L^{max}, p)$
From the known properties of the Binomial distribution, it can be shown that:

- The **expected number of edges** is $\langle L \rangle = pL^{max}$,
- The **variance of the number of edges** is $\sigma^2 = L^{max}p(1-p)$

Binomial distribution

The **Binomial distribution** $\mathbb{B}(N_b, p_b)$ is a discrete distribution modeling the number of successes x in a sequence of N_b independent experiments with success probability p_b. For instance, it models how many times (x) one will obtain a 6 (*success*) if they throw a dice N_b times and that the probability to obtain a 6 is $\frac{1}{6}$. It is defined as $P(x) = \binom{N_b}{x} p^x (1-p_b)^{N-x}$. $\binom{N}{x}$ is the binomial coefficient, describing the number of ways, disregarding order, that x elements can be chosen among N_b.

ER: Degree distribution

Since each node has an independent probability to be connected with each other node, the degree distribution of the ER model is modeled as a binomial distribution $\mathbb{B}(N-1, p)$, i.e., the probability to have a given degree knowing that we have a probability p to have a link with each of the other nodes in the graph. From the properties of the Binomial distribution, we know that:

- The **expected average degree** is $\langle k \rangle = p(N-1)$
- The **variance of the degree** is $\sigma_k^2 = p(N-1)(1-p)$

We can note that the distribution becomes increasingly narrow as the network size increases, i.e., we are increasingly confident that the degree of a node is in the vicinity of $\langle k \rangle$:

$$\frac{\sigma_k}{\langle k \rangle} = \frac{1}{(N-1)^{1/2}}$$

ER: Approximation of degree distribution by a Poisson Distribution

When the number of nodes N is large and the average degree $\langle k \rangle$ is small, the degree distribution can be approximated by a Poisson distribution $\text{Pois}(\langle k \rangle)$. From the properties of Poisson distributions, we approximate that for a network with average degree $\langle k \rangle$:

- The **variance of the degree** is $\sigma_k^2 \approx \langle k \rangle$

Poisson distribution

The **Poisson distribution** $\text{Pois}(\delta)$ is a discrete distribution modeling the probability of observing exactly x occurrences of an event in a period of duration Δ_t if this event occurs randomly and that there are in average δ occurrences of it during a period Δ_t. Working with the Poisson distribution is convenient because it depends only on a single parameter δ.

ER: Clustering Coefficient

The **Global Clustering Coefficient** of a network is defined as the fraction of closed triads among all triads. Since any edge (u, v) has a fix probability to exist p independently of the existence of any other edge in the network, the probability of having edge $(a, c) \in E$ for a triad $[a, b, c]$ such as $(a, b), (b, c) \in E$ tends towards p for large graphs.
Thus, the clustering coefficient of an ER graph is $C^g \approx p$. Since we know that most real networks are sparse, p is small, thus C^g is small. A similar reasoning can be used to show that the average clustering coefficient $\langle C \rangle$ is small too.

ER: Average Distance

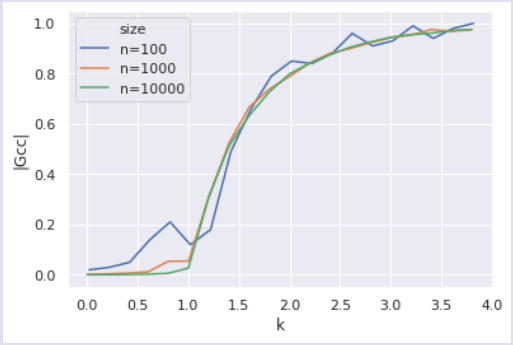
We can intuitively estimate the order of the **Average Distance** of an ER random graph as follows:
We know that the clustering coeffient of an ER graph is small. Therefore, we can approximate the graph as having a tree-like structure. As a consequence, the number of nodes located at distance d of a node u increases as $\langle k \rangle^d$. From this approximation, the relation between distance and number of nodes is $N = \langle k \rangle^d$ hops, thus the order of ℓ is $\log_{\langle k \rangle} n = \frac{\log N}{\log \langle k \rangle}$.
We can thus say that the order of the average distance of a sparse ER graph relatively to its size is $\mathcal{O}(logN)$, and thus that: **ER graphs have a short average distance**.

Order of magnitude

The notation $\mathcal{O}$ is used to represent the **order of magnitude** of a value. It roughly indicates how this value is related to another one, ignoring any constant. For instance, $\mathcal{O}(x) = \mathcal{O}(10x) = \mathcal{O}(x/10)$. Typical orders of magnitude are $\mathcal{O}(\log x)$, $\mathcal{O}(x)$, $\mathcal{O}(x^2)$ and $\mathcal{O}(2^x)$.

ER: Largest connected component

The largest connected component of a graph is a way to measure its connectivity. On random networks, the relation between the density (or average degree) of a graph and the size of its largest connected component is known to undergo a *phase transition* phenomenon, i.e., a rapid change when a threshold is crossed. More precisely, as long as $\langle k \rangle < 1$, several connected components of similar sizes exist in the network, while, when $\langle k \rangle > 1$, the graph has a single *giant component* with high probability.



An intuitive way to understand this phenomenon is to use the same observation of the graph being tree-like as previously. Since the number of nodes N that can be reached after d hops can be estimated to grow as $\langle k \rangle^d$, a value of $\langle k \rangle < 1$ leads to an impossibility to reach all nodes even for a large d, while $\langle k \rangle > 1$ leads to arbitrarily large N for long enough d. Proper demonstration and more details can be found in the original paper[a].

You can explore this property using this interactive *explorable*: `https://www.complexity-explorables.org/explorables/the-blob/`

[a] Erdős and Rényi 1960.

Configuration Model (CM)

The **Configuration Model** is another classic random graph model in which the degree of each node –or the degree distribution– is preserved. In general terms, a configuration model is defined by the number of nodes in the graph, the number (or probability) of edges, and a distribution of degrees of nodes.
This degree distribution can either be chosen *a priori*, for instance following a *Poisson* or a *Power-law* distribution, or by taking the observed distribution of a real network we would like to obtain a randomized-version of.
Note that in the later case, nodes can be considered to retain their identity: one can compare the local properties of the node of highest degree between the two graphs, for instance.

Why the configuration model

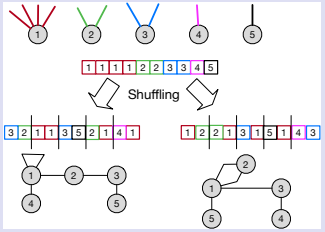
For many real graphs, nodes represent real entities, and the degree of those nodes is due to an intrinsic property of those nodes, which is known in advance and should be taken into account. For instance, let's consider a network representing flight connections between airports: each node represents an airport, and there is an edge between two airports if a direct flight exist between them. *JFK* international airport in New-York will likely be a Hub in this network, having a very large degree. This large degree is a consequence of the properties of the city it belongs to: large population, touristic attraction, etc. So, *if connections between airports were random*, it could nevertheless be relevant to keep the degree of this node.
Furthermore, the degree distribution itself is also a characteristic of the network: the fact that hubs *do exist* in the network change its properties, compared with a network in which such nodes do not exist.

Approximate/Soft Configuration model

In the approximate version of the Configuration model, each pair of node is connected by an edge with a given probability, which depends on their **objective degrees**.
More precisely, the probability of having an edge (i, j) is defined as $p_{uv} = \frac{k_u k_v}{2L}$. Note that this is a well defined probability only if $\max(k_u)^2 < 2m$, otherwise it can be higher than 1. p_{uv} should therefore rather be understood as the *expected number of edges* in a multigraph.
Intuitively, this definition can be understood as follows: each node u has k_u stubs. The total number of stubs in the graph is $2L$. Knowing that node v has k_v stubs, the probability for each stub of u to connect to a stub of v is $\frac{k_v}{2L}$.
Note that this model is defined such as self-loops can exist.

Rewired exact configuration model

When the objective of a configuration model is to obtain a randomized version of an observed graph, a common approach is to fix the exact degree of each node, and to connect *stubs* randomly. An efficient way to do so is to use the following algorithm:
1) Create a list s such as it contains k_u times the index of node u - 2) Randomize s - 3)For each i in $[0, L]$, create an edge between nodes of index s_{2i} and s_{2i+1}.



Note that this method can create self-loops and multiple links between the same nodes, even if the original network was a *simple graph*. However, the number of multiple links and self-links decreases when the number of nodes increases, for sparse graphs. The probability of an edge to exist between two nodes depends on their degree, and is the same as in the soft CM.
For more details on configuration models with fixed degree sequences, see[a].

[a] Fosdick et al. 2018.

CM: Clustering Coefficient

The clustering coefficient of the configuration model can also be studied theoretically. Its derivation is beyond the scope of this class and can be found in the literature[a]. Intuitively, we can use the same reasoning as for the ER model: the probability of having edge $(a, c) \in E$ for a triad $[a, b, c]$ such as $(a, b), (b, c) \in E$ is $\frac{k_a k_c}{2L}$. However, the probability of observing (a, b) and (b, c) and thus to have such a triad also depends on k_a, k_b, k_c. In the end, the clustering coefficient is

$$C = \frac{1}{L} \frac{[\langle k^2 \rangle - \langle k \rangle]^2}{\langle k \rangle^3}$$

where $\langle k^2 \rangle$ correspond to the expected *variance* (second moment) of the degree.
Since the right part of the equation is a constant depending only on the average degree, the order of the clustering coefficient is $\mathcal{O}(1/L)$, and thus small for large graphs. This is true as long as $\langle k^2 \rangle$ is definite, which might not be the case if the degree distribution is a *power law*.

[a] M. Newman 2018.

CM: Friendship paradox

An interesting property of the Configuration Model with heterogeneous degree distribution arises when we study the **average degree of random neighbors**. Let's call p_k the probability to pick a node of degree k when we pick a node at random. This probability represents the *degree distribution* chosen for the configuration model. Now, if we choose one node at random, and then pick one of its neighbors at random, what is $p_{neighb,k}$, the degree distribution of *random neighbors*? It is different, because nodes with a higher degree have, by definition, a higher probability of being chosen. More formally,

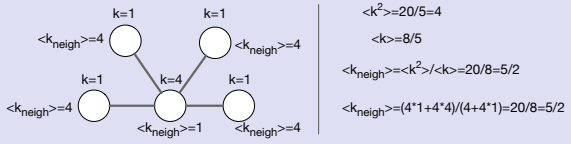
$$p_{neighb,k} = \frac{k}{2m} n p_k = \frac{k p_k}{\langle k \rangle}$$

because np_k is the number of nodes of degree k in the graph, and $\frac{k}{2m}$ is the probability to pick at random a *stub* of a particular node of degree k among all stubs.

We can now compute the **average degree of neighbors** of a node chosen at random, as:

$$\langle k_{neighb} \rangle = \sum_k k p_{neighb,k} = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

Thus if all degrees are the same (*homogeneous*), $\langle k_{neighb} \rangle = \langle k \rangle$, but if it is **heterogeneous**, $\langle k_{neighb} \rangle > \langle k \rangle$ due to the comparatively larger influence of high degrees.



CM: Average distance

We use the same logic as for the ER model of the graph being locally tree-like due to the low Clustering Coefficient to show intuitively that the *average distance* is short. This property is verified experimentally.

Examples of differences in Clustering and average path length for a few real graphs, compared with randomized versions of it.

graph	N	L	k	C_g	$\langle \ell \rangle$	ER-C_g	ER-$\langle \ell \rangle$	CM-C_g	CM-$\langle \ell \rangle$
karate	34	77	4.53	**0.26**	2.42	*0.14*	2.42	*0.14*	2.55
football	115	613	10.66	**0.41**	2.51	*0.10*	2.25	*0.07*	2.28
wiki-science	687	6523	18.99	**0.47**	3.43	*0.03*	2.55	*0.08*	2.65
euroroad	1174	1417	2.41	0.03	**18.40**	0.00	7.66	0.00	9.55

Differences btw. Real & Random networks

When comparing real networks to ER and CM networks of similar properties, we observe that they tend to disagree on one of two key properties: on real graphs, usually, the graph has a high clustering coefficient and a short average distance (or sometimes the opposite).
On the contrary, random networks have both a low clustering coefficient and a short average distance.

Watts-Strogatz (WS) Model

The Watts-Strogatz model was introduced[a] to show how a simple phenomenon could create networks having both a large clustering coefficient and a short average distance.
The model has 3 parameters:

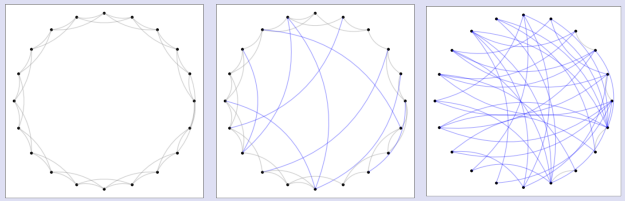
- N: number of nodes
- K: initial number of neighbors
- p: rewiring probability

The network is created following a 2-step processes: first N nodes are disposed on a ring, and each node is connected to its K closest neighbors. Then each edge is replaced by a random edge with probability p. It can be interpreted as a network combining the properties of a (1-dimentional) **regular lattice** and of an **ER network**.

[a] Watts and Strogatz 1998.

WS - Illustration

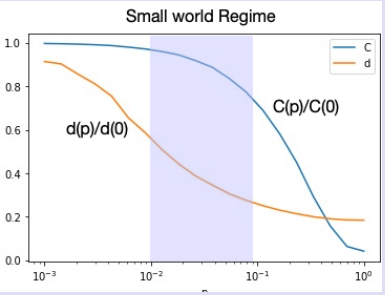
From left to right: WS graphs when increasing the probability of rewiring. $N = 20, K = 4$



(a) $p = 0$ Regular | (b) $p = 0.3$ Small world | (c) $p = 1$ Random

WS - Small World Regime

If p is small, the network has properties similar to a regular lattice, and if p is large, properties of an ER graph.
We can observe this transition by comparing how the Clustering (C) and average distance (d) change when varying p, compared with the network when $p = 0$, i.e., a regular lattice.



Example with $N = 200, K = 6$.

WS - Clustering

Properties of the WS model are not as simple to study theoretically as previous random graphs, so most details are not presented here. It can be shown however, that the global clustering coefficient can be approximated by:

$$C^g = \frac{3(K-2)}{4(K-1)+8Kp+4Kp^2}$$

which is independent of N, thus can be large even for large graphs.

WS - Average Path length

The average path length of the WS model has been studied through approximations and numerical simulations[a] and can be shown to become small quickly with the increase in p.

[a] M. E. Newman 2000.

WS - Degree distribution

Without entering into details, it can be shown the the degree distribution range from a fixed degree for all nodes to a Poisson distribution, since each rewired edge is decreasing the degree of some nodes and increasing the degree of some others in a random way.

Barabási-Albert (BA) Model

The **Barabási-Albert** model of random graphs was introduced[a] to illustrate how a simple mechanism could explain a common property of real graphs, the **power-law degree distribution**. This mechanism is though to somewhat mimic what is happening in real life, at least for some networks. It is often called **preferential attachment**, and mimic the **rich get richer phenomena**: nodes that already have a large degree are more *attractive*, and thus are more likely to become connected with other nodes creating links.

[a] Barabási and Albert 1999.

BA - Preferential attachment

The preferential attachment process has two parameters, the number of edges to create at each step m and the initial number of nodes m_0, with $m \leq m_0$. It is defined by the following iterative process:

- Start with a connected graph with m_0 nodes
- At each step, add a new node and m links connecting it to m other nodes chosen randomly proportionnaly to their degree, i.e., with probability $p_i = \frac{k_i}{\sum_j k_j}$

BA - Degree distribution

The degree distribution created by the preferential attachment mechanism is a power law of exponent $\alpha = 3$. The exponent of the distribution does not depend on parameters m and m_0. The degree exponent is *stationary in time*, i.e., it stays the same while we add new nodes and edges.

Nodes degree increase with time: the earlier a node was added, the larger its degree tends to be.

BA - Average Path Length

Networks generated by the BA process have a power-law degree distribution of exponent $\alpha = 3$. It is known that such networks have a short average path length, more formally:

$$\langle \ell \rangle = \frac{\ln N}{\ln \ln N}$$

BA - Clustering Coefficient

Although the demonstration is beyond the scope of this class[a], it can be shown that the clustering coefficient of BA graphs is:

$$C = \frac{L}{4}\frac{(\ln N)^2}{N}$$

This is more than for a random network, but still decreases with the network size, and tends toward 0 for large graphs. It is thus considered a **small** clustering coefficient.

[a] Barabási and Albert 1999.

Other random graph models

Many other graph models have been proposed in the literature, either *mechanistic models* to mimic common properties of some graphs, as with BA and WS models, or *statistical models* to generate random graphs with imposed constraints, as the Configuration model does with degree distributions.
Some examples of mechanistic models:

- Vertex copying model (J. M. Kleinberg et al. 1999)
- Tunable-clustering scale-free model (Holme and Kim 2002)
- Forest fire model (Leskovec, J. Kleinberg, and Faloutsos 2005)

Some examples of statistical models:

- Exponential Random Graphs (Robins et al. 2007)
- Stochastic Block Models (Peixoto 2019)
- A survey on the topic (Orbanz and Roy 2014)

References

[1] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[2] Paul Erdős and Alfréd Rényi. "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pp. 17–60.

[3] Bailey K Fosdick et al. "Configuring random graph models with fixed degree sequences". In: *Siam Review* 60.2 (2018), pp. 315–355.

[4] Petter Holme and Beom Jun Kim. "Growing scale-free networks with tunable clustering". In: *Physical review E* 65.2 (2002), p. 026107.

[5] Jon M Kleinberg et al. "The web as a graph: measurements, models, and methods". In: *International Computing and Combinatorics Conference*. Springer. 1999, pp. 1–17.

[6] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. "Graphs over time: densification laws, shrinking diameters and possible explanations". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 177–187.

[7] Mark Newman. *Networks*. Oxford university press, 2018.

[8] Mark EJ Newman. "Models of the small world". In: *Journal of Statistical Physics* 101.3-4 (2000), pp. 819–841.

[9] Peter Orbanz and Daniel M Roy. "Bayesian models of graphs, arrays and other exchangeable random structures". In: *IEEE transactions on pattern analysis and machine intelligence* 37.2 (2014), pp. 437–461.

[10] Tiago P Peixoto. "Bayesian stochastic blockmodeling". In: *Advances in network clustering and blockmodeling* (2019), pp. 289–332.

[11] Garry Robins et al. "An introduction to exponential random graph (p*) models for social networks". In: *Social networks* 29.2 (2007), pp. 173–191.

[12] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), pp. 440–442.