Statistical Analysis of Random Graphs

Maximilien Dreveton

November 7, 2019

Contents

Random graph models

References The best reference for this chapter that fits very well into this course mindset is: Barabási et al. (2016). Note that an online and interactive version is available at: http://networksciencebook.com/. Other good references for this chapter are (by order of relevance): Hofstad (2016), Durrett (2007), and Chung and Lu (2006). Other traditional books (but heavier on the maths side) written by "big names" of the field are Janson et al. (2011) and Bollobás (1998).

1.1 Some reminders

1.1.1 Probability toolbox

Proposition 1.1.1.

- *For a, b constants,* $\mathbb{E}(aX+b) = a\mathbb{E}(X) + b$
- $\mathbb{E}(X_1 + \cdots + X_m) = \mathbb{E}(X_1) + \cdots + \mathbb{E}(X_m)$*;*
- *Let X be a r.v., A an event and $1_A(X)$ the indicator that event is realized by X. Then:*
$$\mathbb{E}\big(1_A(X)\big) = \Pr(X \in A).$$

Definition 1.1.2. $\mathrm{Var}(X) = E\Big(\big(X - \mathbb{E}(X)\big)^2\Big)$.

Proposition 1.1.3.

- $\mathrm{Var}(X) = \mathbb{E}(X^2) - (E(X))^2$*;*
- *For a, b constants,* $\mathrm{Var}(aX+b) = a^2\,\mathrm{Var}(X)$
- *If $X_1, \dots, X_m$ are mutually independent r.v., then* $\mathrm{Var}(X_1 + \cdots + X_m) = \mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_m)$*.*
- *If we don't have this independence, then* $\mathrm{Var}(X_1 + \cdots + X_m) = \mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_m) + \sum_{i \neq j} \mathrm{Cov}(X_i, X_j)$*, where* $\mathrm{Cov}(X_i, X_j) = \mathbb{E}(X_i X_j) - \mathbb{E}(X_i)\mathbb{E}(X_j)$*.*

1.1.2 Basic probability laws

Definition 1.1.4. Let X be a random variable. We say X is generated around a Bernoulli law of parameter $p \in [0;1]$, and denote $X \sim \mathrm{Ber}(p)$ if:

1. X takes values in $\{0;1\}$ (almost surely);
2. $\Pr(X=1) = p$ and $\Pr(X=0) = 1-p$.

Example 1.1.5. A $\mathrm{Ber}(p)$ model the result when we biased coin toss (p is the probability of winning the coin toss).

Proposition 1.1.6. *Let* $X \sim \mathrm{Ber}(p)$. *We have* $\mathbb{E}X = p$ *and* $\operatorname{Var} X = p(1-p)$.

Definition 1.1.7. The binomial distribution with parameters and p, denoted $\mathrm{Bin}(n,p)$, is the discrete probability distribution of the number of successes in a sequence of n independent Bernoulli trials of parameters p.

Proposition 1.1.8. *If* $(X_i)_{i=1,\dots,n}$ *is a sequence of* n *i.i.d. random variable distributed according to* $\mathrm{Ber}(p)$, *then* $\sum_i X_i \sim \mathrm{Bin}(n,p)$.

Corollary 1.1.9. *Let* $X \sim \mathrm{Bin}(n,p)$. *Then:* $\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$. *Moreover,* $\mathbb{E}X = np$ *and* $\operatorname{Var} X = np(1-p)$.

Definition 1.1.10. The geometric distribution of parameter p, denoted $\mathrm{Geo}(p)$ is the probability of the number of Bernoulli trials (of parameter p) needed to get one success. In particular, if $X \sim \mathrm{Geo}(p)$, then $\Pr(X = k) = (1-p)^{k-1}p$.

Exercise 1.1.11. Let $X \sim \mathrm{Geo}(p)$. Show that $\mathbb{E}X = \frac{1}{p}$ and $\operatorname{Var} p = \frac{1-p}{p^2}$.

1.1.3 Concentration of random variables

First moment inequalities

Proposition 1.1.12 (Markov's inequality)**.** *Let* X *be a random variable, with positive values, and* $a \in \mathbb{R}$. *We have:*

$$\Pr\big(X \geq a\big) \leq \frac{\mathbb{E}X}{a}.$$

Proof.

$$\mathbb{E}X \geq \mathbb{E}\Big(X 1_{X\geq a}\Big) \geq a\mathbb{E}\Big(1_{X\geq a}\Big) = a\Pr\Big(X \geq a\Big).$$

□

Remark 1.1.13. By letting $a = t\mathbb{E}X$, we obtain $\Pr(X \geq t\mathbb{E}X) \leq \frac{1}{t}$. The convergence speed in $1/t$ is rather slow, and depending on the situations might not be strong enough.

Corollary 1.1.14 (First moment method)**.** *Let* X *be a positive, integer valued random variable. We have:*

$$\Pr(X \neq 0) \leq \mathbb{E}(X).$$

The first moment is an upper bound on the probability that a integer random variable is not equal to zero.

Proof. Since X is integer valued, we have $\Pr(X \neq 0) = \Pr(X > 0) = \Pr(X \geq 1)$, and from there we can use Markov inequality. □

Application 1.1.15 (Union bound)**.** Let $A_1, \dots, A_m$ be a collection of events. Then,

$$\Pr(A_1 \cup \cdots \cup A_n) \leq \sum_{i=1}^{m} \Pr(A_i).$$

This can be shown by using the first moment method on $X = \sum_{i=1}^m 1_{A_i}$ and observe that $\{X > 0\} = A_1 \cup \cdots \cup A_m$.

Remark 1.1.16. The first moment method is generally used when we have a sequence of integer, positive r.v. X_n such that $\mathbb{E}X_n \to 0$. In that case, $X_n \to 0$ almost surely.

Remark 1.1.17. We could think that, if $\mathbb{E}X_n \to +\infty$, then $\Pr(X_n > 0) \to 1$, but this isn't true, as the next example shows.

Example 1.1.18. Let's take X_n such that $X_n = n^2$ with probability $1/n$ and $X_n = 0$ otherwise. Then $\mathbb{E}(X_n) = n \to +\infty$ but $X_n \to 0$. Loosely speaking, this can happen because the variance of X is very large.

Second moment inequalities

Proposition 1.1.19 (Chebyshev's inequality)**.** *Let X be a random variable, and $a > 0$. We have:*

$$\Pr\Big(\big|X - \mathbb{E}X\big| \geq a\Big) \leq \frac{\operatorname{Var} X}{a^2}.$$

Proof. Markov inequality to $Y = (X - \mathbb{E}X)^2$. □

Example 1.1.20. Let X be Gaussian $\mathcal{N}(0, \sigma^2)$. Then $\mathbb{E}|X| = \sigma\sqrt{\frac{2}{\pi}}$. Markov applied to $|X|$ gives us:

$$Pr(X \geq a) \leq \sqrt{\frac{2}{\pi}}\frac{\sigma}{b},$$

while Chebyshev gives:

$$\Pr(X \geq a) \leq \left(\frac{\sigma}{a}\right)^2.$$

If a is large, Chebyshev gives a stronger bound.

Application 1.1.21 (Weak law of Large Numbers)**.** $X_1, \dots X_n$ be independent r.v. with mean μ and variance $\sigma^2 < +\infty$. Then:

$$\Pr\left(\left|\frac{X_1 + \cdots + X_n}{n} - \mu\right| > \epsilon\right) \to 0.$$

Actually, with extra work, the condition $\sigma^2 < +\infty$ isn't needed. With lot of extra work, we can show that the convergence is almost surely (and not in probability, as we have here).

Proof. Chebychev at $U_n := \dfrac{X_1 + \cdots + X_n}{n}$, which has a mean μ and variance σ^2 gives:

$$\Pr\big(|U_n| \geq \epsilon\big) \leq \frac{\sigma^2}{n\epsilon^2} \to 0$$

□

Corollary 1.1.22 (Second moment method)**.** *Let X be a positive random variable. We have:*

$$\Pr\left(X=0\right) \leq \frac{\operatorname{Var} X}{\left(\mathbb{E}X\right)^2} = \frac{\mathbb{E}(X^2)}{(\mathbb{E}X)^2} - 1.$$

Proof. Chebychev with $a = \mathbb{E}X$.

$$\Pr\left(X=0\right) \leq \Pr\left(\left|X-\mathbb{E}X\right| \geq \mathbb{E}X\right) \leq \frac{\operatorname{Var} X}{\left(\mathbb{E}X\right)^2}.$$

The first inequality comes from $\left|X-\mathbb{E}X\right| \geq \mathbb{E}X \Rightarrow X \leq 0$ or $X \geq 2\,\mathbb{E}X$. □

Remark 1.1.23. Using Cauchy-Schwarz inequality, we have $\mathbb{E}(X) \leq \mathbb{E}(X 1_{X>0}) \leq \sqrt{\mathbb{E}(X^2)}\sqrt{\Pr(X>0)}$, thus $\Pr(X=0) = 1 - \Pr(X>0) \leq \frac{\operatorname{Var}(X)}{\mathbb{E}(X^2)}$, which is a slightly stronger inequality than 1.1.22.

Concentration sum of i.i.d. random variables

Proposition 1.1.24 (Hoeffding's inequality)**.** *Let X_i be some independent random values, such that $a_i \leq X_i \leq b_i$, and $S_n = \sum_{i=1}^n X_i$. For $t > 0$, we have:*

$$\Pr\left(S_n \geq \mathbb{E}S_n + t\right) \leq \exp\left(-\frac{2t^2}{\sum_i (b_i-a_i)^2}\right),$$
$$\Pr\left(S_n \geq \mathbb{E}S_n - t\right) \leq \exp\left(-\frac{2t^2}{\sum_i (b_i-a_i)^2}\right),$$
$$\Pr\left(|S_n - \mathbb{E}S_n| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_i (b_i-a_i)^2}\right).$$

More details about concentration inequalities can be found in the chapter 2 of Vershynin (2018).

1.1.4 Exercises

Exercise 1.1.25. Let throw m balls into n bins. What can we say about the probability that one bin is empty:

1. If $m = (1+\epsilon)n\log n$;
2. If $m = (1-\epsilon)n\log n$.

Exercise 1.1.26.

1. What's the 'typical' position of a simple random walk after n steps?
2. What's the longest run of Heads in n flips of a fair coin?
3. What is the maximum of n independent standard normal RV's?

1.1.5 Graph Theory

Definition, vocabulary

Definition 1.1.27. A graph G is a pair (V, E), where V is a (finite) set, whose elements are called nodes (or vertices, or points) and E is a set of ordered node pairs called edges (or links, lines, bonds). Some vocabulary:

- If $(ij) \in E \iff (ji) \in E$, the graph is said undirected. (It means that if there is a link going from i to j, there exists the same link in opposite direction).
- If for all nodes i, $(ii) \notin E$, we say there is no self-loops.

Definition 1.1.28.

- We call path of G of length k a sequence $e_1, \dots, e_k$ of edges $e_i = (v_{i-1}, v_i)$ where the v_i are distinct vertices.
- A k-cycle is a path of legnth k that starts and ends at the same vertex.
- Let suppose G is undirected. We say that two nodes u, v are connected if there exists a path going from u to v. We note $u \leftrightarrow v$.

Proposition 1.1.29. *$\leftrightarrow$ is an equivalence relationship. In particular, we can partition the nodes into equivalent classes, that we will call connected components.*

Proof. We have $u \leftrightarrow u$ (path of length 0), if $u \leftrightarrow v$ and $v \leftrightarrow z$, then $u \leftrightarrow z$ (by combining the two paths; transitivity). Finally, $u \leftrightarrow v$ implies $v \leftrightarrow u$ (the same path, on the opposite direction; symmetry). □

Definition 1.1.30. If G has only one equivalent class under the relation $\leftrightarrow$, we say G is connected. We say G is disconnected otherwise.

In particular, if a graph is connected, it means for every node i and j, there exists a path going from i to j.

Definition 1.1.31. Let i, j be two nodes. We call the distance between i and j, and denote $d(i, j)$ the length of the shortest path between i and j. If $i \not\leftrightarrow j$, then $d(i, j) := +\infty$.

Definition 1.1.32 (Diameter)**.** We call diameter of a graph the largest distance between any pair of connected vertices.

Graph Theory and Linear Algebra

Definition 1.1.33. We call adjacency matrix (denoted by A), the binary matrix such that $A_{ij} = 1$ iff $(ij) \in E$.

Proposition 1.1.34. *A is symmetric if and only if the graph is un-directed. Moreover, the diagonals elements of A are zeros iff the graph doesn't have self-loops.*

We can easily extend those definition to weighted graph (where the edges bear some weight).

Definition 1.1.35. We call degree matrix, denoted D, of a graph G the diagonal matrix whose element d_{ii} is the degree of node i.

Definition 1.1.36. We call Laplacian the matrix $L := D - A$.

1.1.6 Random Graph: vocabulary and notations

In the following notes, $G = (V, E)$ will denote a graph, where $V = \{1, \dots, n\}$ is the set of vertices (nodes) and E the set of edges.

When we say G is a random graph, we actually mean that G is a graph that was generated along some probability distribution. It is a slight abuse of notation, as once G is given, it is not random.

We will call $\mathbf{d} := (d_1, \dots, d_n)$ the degree sequence of nodes $1, \dots, n$.

Since G is random, the degree d_i of a node i is actually a random variable, which can be distributed along some probability distribution (binomial, etc.). When all the nodes degree are identically distributed (*i.e.*, $d_1, \dots, d_n$ are all distributed along the same probability distribution $\mathcal{D}$), we say that the degrees in the graph G are distributed along that degree distribution $\mathcal{D}$.

Most graph observed in nature exhibits a power law degree distribution Clauset et al. (2009).

Loosely speaking, the power law means that $P(d = k) := p_k \propto k^{-\tau}$. In particular, if we note $\Pr(X = k) = p_k$, we have $\log p_k = -\tau \log k + C$ for some constant C. In a log/log plot, we get a straight line of slope $-\tau$.

Definition 1.1.37 (Different power laws)**.** Let X be a random variable.

- X is said to be distributed according to a Zeta distribution of exponent τ if X takes integer values, and $\forall k \in \mathbb{N} : \Pr(X = k) = C_\tau k^{-\tau}$. Here C_τ is the normalization factor, equal to $\left(\sum_{k=1}^{\infty} k^{-\tau} \right)^{-1} = \dfrac{1}{\zeta(\tau)}$ (where ζ is the Riemann function).

- X is said to be distributed according to a Zipf law of parameter n and exponent τ if $\Pr(X = k) = C_{n,\tau} k^{-tau}$

- X is said to be distributed according to a (continuous) power law if X takes values in $[x_{min}; +\infty]$ where $x_{min} > 0$ and X has a density $f(x) = Cx^{-\tau}$

Exercise 1.1.38. Let X be a r.v. following a continuous power law distribution. Show that:

- X has a mean iff $\tau > 2$, and in that case $\mathbb{E}X = \dfrac{\tau - 1}{\tau - 2} x_{min}$;

- X has a second moment iff $\tau > 3$, and in that case $\mathbb{E}X^2 = \dfrac{\tau - 1}{\tau - 3} x_{min}^2$.

What would it be for a zeta distribution ?

Proposition 1.1.39. *Let $f(x) = ax^{-\tau}$ the density of a power law. Then f is scale invariant, that is to say $f(cx) \propto f(x)$ for any constant c.*

Because of this proposition, graphs whose degree distribution follow a power law are said to be *scale-free* (or *scale invariant*).

Small World property

Many real networks present the *small world property*, i.e., two nodes are not too far apart of each other (given the graph distance 1.1.31).

Definition 1.1.40. Let $(G_n)_{n\in\mathbb{N}}$ be a sequence of random graphs (G_n having n nodes), and let H_n be the distance among two connected nodes of G_n chosen uniformly at random.

Now what typical value should use to say that H_n is small or not?

Exercise 1.1.41.
- Consider the linear graph. What's the maximum distance between two nodes ? The typical distance H_n ?
- Consider the circular graph. Same questions.
- Same questions with the complete graph.
- The nearest-neighbors torus of width r and dimension d ?

Definition 1.1.42. We say that $(G_n)_{n\in\mathbb{N}}$ is a *small world* if there exists a constant $K < \infty$ such that:

$$\lim_{n\to+\infty} \Pr(H_n \leq K \log n) = 1$$

Moreover, we say that $(G_n)_{n\in\mathbb{N}}$ is *ultra small world* when, for every $\epsilon > 0$

$$\lim_{n\to+\infty} \Pr(H_n \leq \epsilon \log n) = 1$$

1.1.7 Miscellaneous

Lemma 1.1.43. $\left(1-\dfrac{\lambda_n}{n}\right)^n \exp(\lambda_n) = \exp(o(\lambda_n))$.

1.2 Erdős-Rényi random graphs

1.2.1 Bernoulli random graphs

Definition 1.2.1. Let $P = (p_{ij})_{i,j\in\{1,\dots,n\}} \in [0;1]^{n\times n}$ be a symmetric matrix. A Bernoulli random graph G is a (undirected, unweighted) graph G where the edges (ij) are generated independently such that $\Pr\big((ij)\in E\big) = p_{ij}$. In that case, we note $G \sim G(n,(p_{ij}))$.

Remark 1.2.2. If $G \sim G(n,(p_{ij}))$ then the adjacency matrix of A is a symmetric random matrix, whose entries are independently distributed, and $A_{ij} \sim \mathrm{Ber}(p_{ij})$.

Proposition 1.2.3. *Let $G \sim G(n,(p_{ij}))$. We have:*

$$\Pr(G) = \prod_{i<j} p_{ij}^{A_{ij}} (1-p_{ij})^{1-A_{ij}}.$$

Example 1.2.4. Suppose that $\forall i, j : p_{ij} = p$. Then, $G(n, (p_{ij}))$ is called the Erdős-Rényi model[1], and traditionally denoted $G(n,p)$ or $G_{n,p}$.

Example 1.2.5. Assume the n nodes are separated into K distinct, non-overlapping communities, *i.e.*, there exists $\sigma : V \to \{1, \dots, K\}$, such that $\sigma(i)$ denotes the community of node i.

Assume that $p_{ij} = q_{\sigma_i, \sigma_j}$ (this means that the probability of observing an edge between i and j depends only on the community of node i and j).

Then in that case, $G(n, (p_{ij}))$ is called the Stochastic Block Model (SBM), and denoted $SBM(n, \sigma, Q)$. More precisely, $\Pr\Big((ij) \in E\Big) = q_{\sigma_i \sigma_j}$ ony depends on the community assignment of nodes i and j.

Moreover, $\Pr(G|\sigma) = \prod_{i<j} q_{\sigma_i \sigma_j}^{A_{ij}} (1 - q_{ij})^{1 - A_{ij}}$.

Remark 1.2.6. The adjacency matrix of SBM(n,Q,K) can be seen as a block matrix, whose blocks are ER graphs.

1.2.2 Degree distribution

Proposition 1.2.7. *Let $G \sim G(n,p)$, and let d_i be the degree of node i. Then, the sequence $(d_1, \dots, d_n)$ is i.i.d., and the d_i's are distributed according to* $\mathrm{Bin}(n,p)$.

Proof. Indeed, the degree of i, denoted d_i, is equal to $\sum_{j=1}^n A_{ij}$, where A_{ij} are i.i.d. Bernoulli random variable, of parameter p. Since the elements $(A_{ij})_{i<j}$ are independent, so are the d_i's. □

Proposition 1.2.8. *Let G be a homogeneous SBM graph, i.e., two communities of equal size $\frac{n}{2}$, with probabilities p_{in} and p_{out} of forming intra and inter-communities edge. Then, $d_i \sim \mathrm{Bin}\left(\frac{n}{2}, p_{in}\right) + \mathrm{Bin}\left(\frac{n}{2}, p_{out}\right)$ and:*

$$\mathbb{E}\, d_i = \frac{p_{in} + p_{out}}{2}.$$

Proof. Similar. □

Remark 1.2.9. It has been observed that many real graphs have a power law degree distribution, and not a binomial one. See for example Clauset et al. (2009). An intuitive explanation is that since binomial distributions are well concentrated, a Erdős-Rényi graph does not allows for many hubs (degree much higher than the average), which we tend to see in real networks (think of a social network).

1.2.3 Phase transition phenomena

Theorem 1.2.10 (Phase transition for giant component – constant degree regime)**.** *Let $G \sim G(n, p_n)$ be a Erdős-Rényi graph, with $d_n = np_n$ is a constant. We have:*

(a) If $d_n < 1$: a.s no connected component of size larger than $O(\log n)$;

(b) If $d_n = 1$: a.s there is one large component of size $O(n^{2/3})$;

(c) If $d_n > 1$ (d constant) : one giant component of size $O(n)$.

[1] But was introduced by Gilbert, in 1959.

Theorem 1.2.11 (Phase transition for connectivity – logarithmic degree regime)**.** *Let $G \sim G(n, p_n)$ be a Erdős-Rényi random graph, with $d_n = np_n$. We have:*

(a) If there exists $\omega_n \to +\infty$ such that $d_n < \log n - \omega_n$, then G is a.s. non connected (we in fact have a bit stronger: the graph contains a.s. isolated nodes);

(b) If there exists $\omega_n \to +\infty$ such that $d_n > \log n + \omega_n$, then G is a.s. connected.

We also say that the function $t(n) = \log n$ is a threshold function (of the degree) for the property "the graph is connected".

Example 1.2.12. Assume $d_n = \log n + \log\log n$, and let $G_n \sim G(n, p_n)$. Asymptotically, will G be connected ? Same question for $d_n = 1.00001 \log n$, $d_n = 0.9999 \log n$ and $d_n = \log n + 25$.

1.2.4 Beginning of a proof

Theorem 1.2.13 (Isolated nodes)**.** *The probability that a Erdős-Rényi graph G_n on node set $[n]$ with link probability p_n contains isolated nodes satisfies*

$$\Pr\big(\exists \text{ isolated node}\big) \to \begin{cases} 0 & \text{if} \quad p_n \geq \dfrac{\log n + \omega_n}{n} \text{ for some } \omega_n \to +\infty \\ 1 & \text{if} \quad p_n \leq \dfrac{\log n - \omega_n}{n} \text{ for some } \omega_n \to +\infty. \end{cases} \tag{1.2.1}$$

In particular, this result implies that if $p_n \leq \dfrac{\log n + \omega_n}{n}$, then the graph is a.s. not connected. This correspond to point (a) of Theorem 1.2.11.

Proof. To shorten notations, let $p_n^{\pm} := \dfrac{\log n \pm \omega_n}{n}$ (we deal with both case at once).

Let A_i be the event "node i is isolated", and let $I_n := \sum_{i=0}^{n} 1(A_i)$ be the number of isolated nodes. Recall we note $d_n = np_n$ the mean degree. We have:

$$\begin{aligned} \Pr(A_i) &= (1 - p_n)^n \\ &= \Big(1 - \frac{d_n}{n}\Big)^n \\ &\sim \exp\big(-d_n\big) \\ &\sim \frac{1}{n}\exp(\mp\omega_n) \end{aligned}$$

and thus

$$\begin{aligned} \mathbb{E}\big(I_n\big) &= \sum_{i=0}^{n} \Pr(A_i) \\ &= n \Pr(A_1) \\ &\sim \mathrm{e}^{\mp\omega_n}. \end{aligned}$$

So,

1. If $d_n = \log n + \omega_n$, we have $\mathbb{E}\big(I_n\big) \sim \mathrm{e}^{-\omega_n} \to 0$. Since the expected number of isolated nodes goes to 0, this imply (a) by the first moment method.

2. If $d_n = \log n - \omega_n$, we have $\mathbb{E}\big(I_n\big) \sim \mathrm{e}^{+\omega_n} \to +\infty$. the expected number of isolated nodes goes to infinity. Unfortunately, this isn't enough to imply (b), and we will need the second moment method.

Recall (a) is straightforward using Markov inequality. Assume $d_n = \log n - \omega_n$. Then:

$$\begin{aligned} \Pr\big(\exists \text{ isolated node}\big) &= \Pr(I_n \geq 1\big) \\ &\leq \frac{\mathbb{E}I_n}{1} \to 0. \end{aligned}$$

Now assume $d_n = \log n - \omega_n$. To get (b), we have to show that the random variable I_n is well-concentrated around its mean. Since its means diverges to infinity, the result will follow. For that, we will use Chebyshev's inequality (second moment method). We have:

$$\mathrm{Var}(I_n) = \mathbb{E}\big(I_n^2\big) - \big(\mathbb{E}I_n\big)^2.$$

Note that

$$\begin{aligned} \mathbb{E}\big(I_n^2\big) &= \mathbb{E}\Big(\sum_i \sum_j 1(A_i)1(A_j)\Big) \\ &= \sum_i \sum_j \Pr\big(A_i, A_j\big) \\ &= n \Pr\big(A_1\big) + n(n-1)\Pr\big(A_1, A_2\big). \end{aligned}$$

Here we need to be careful, since A_1 and A_2 are not independent. Indeed, knowing that node 1 is isolated means that there is no edge between 1 and 2, and thus increases a bit (weakly) the probability that 2 is isolated. We have:

$$\begin{aligned} \Pr\big(A_1, A_2\big) &= \Pr\big(A_1 \mid A_2\big)\Pr\big(A_2\big) \\ &= (1-p_n)^{n-1}(1-p_n)^n \\ &= (1-p_n)^{n-1}\Pr\big(A_1\big) \\ &= \frac{1}{1-p_n}\Pr\big(A_1\big), \end{aligned}$$

since $\Pr\big(A_1\big) = (1-p_n)^n$. Lastly,

$$\begin{aligned} \big(\mathbb{E}I_n\big)^2 &= \Big(\sum_i \Pr(A_i)\Big)^2 \\ &= \sum_i \sum_j \Pr(A_i)\Pr(A_j) \\ &= \sum_i \sum_j \Pr(A_1)^2 \\ &= n^2 \Pr(A_1)^2. \end{aligned}$$

By putting all pieces together, it leads to:

$$\begin{aligned}
\mathrm{Var}(I_n) &= n \Pr\big(A_1\big) + n(n-1)\Pr\big(A_1 \cup A_2\big) - n^2 \Pr(A_1)^2 \\
&= n \Pr\big(A_1\big) + \frac{n(n-1)}{1-p_n} \Pr\big(A_1\big)^2 - n^2 \Pr\big(A_1\big)^2 \\
&\leq n \Pr\big(A_1\big) + n^2 \Pr\big(A_1\big)^2 \frac{1}{1-p_n} - n^2 \Pr\big(A_1\big)^2 \\
&= n \Pr\big(A_1\big) + n^2 \Pr\big(A_1\big)^2 \Big(\frac{1}{1-p_n} - 1\Big) \\
&= \mathbb{E}(I_n) + \big(\mathbb{E} I_n\big)^2 \frac{p_n}{1-p_n}.
\end{aligned}$$

Thus:

$$\begin{aligned}
\Pr(I_n = 0) &\leq \frac{\mathrm{Var}(I_n)}{\Big(\mathbb{E}(I_n)\Big)^2} \\
&\leq \frac{1}{\mathbb{E}(I_n)} + \frac{p_n}{1-p_n}
\end{aligned}$$

and this last quantity goes to zero when n goes to infinity (when $p_n = (1-\epsilon)\log n$, we already saw that $\mathbb{E} I_n \to \infty$). □

1.3 Other models

1.3.1 Configuration model

Consider a sequence $d = (d_1, \dots, d_n)$. We aim to construct a graph with n vertices, where node i has a degree d_i. Few remarks:

- We can suppose $d_i \geq 1$, as $d_i = 0$ means the node i is isolated and we can remove it.
- It is not obvious that there should exist a graph whose degree are given by a fixed sequence. In fact, such a graph does not necessarily exist. For example, if we assume the graph unweighted, then $\sum_{i=1} d_i$ should be even (since this sum corresponds to counting two times the edges).
- Even by adding the restraint $\sum_{i=1} d_i$, constructing such a graph isn't always possible.
- To avoid those issues, we will allow self loops and multi-edges.

Example 1.3.1. If $d_1 = \cdots = d_n = d$, then we obtain a random d-regular graph (*i.e.*, a graph for which all node have the same degree d).

Example 1.3.2. If the d_i follows a $\mathrm{Bin}(n,p)$, then when $n \to +\infty$, we recover a Erdős-Rényi graph.

Algorithm 1.3.3 (Configuration Model)**.** *Let $\boldsymbol{d} := (d_1, \dots, d_n)$ a sequence such that $\ell_n := \sum_{i=1}^n d_i$ is even.*

At each node $i \in \{1, \dots, n\}$, we attach d_i half-edges. We thus have ℓ_n half-edges, that we can number in an arbitrary order. We pair the first half edge with another one, chosen uniformly at random with the $\ell_n - 1$: this pair gives us our first edge.

We then iterate the procedure, until all the half edges are connected.

*The resulting graph is called the configuration model with degree sequence **d**, abbreviated in $\mathrm{CM}_n(\boldsymbol{d})$.*

Remark:

- As discussed ealier, this model allow multi-edges and self-loops.
- By convention, a self-loop counts for two in the degree of a node.
- The model does not depend on the numbering of the half-edge (since the procedure or pairing half edges is exchangeable).

Proposition 1.3.4 (The law of the Configuration Model)**.** *Let $G = (x_{ij})_{i,j\in[n]}$ be a multigraph on the vertices $[n]$ such that:*

$$d_i = x_{ii} + \sum_{j=1}^n x_{ij}$$

Then,

$$\Pr\Big(\mathrm{CM}_n(\boldsymbol{d}) = G\Big) = \frac{1}{(\ell_n - 1)!!} \frac{\prod_{i=1}^n d_i!}{\prod_{i=1}^n 2^{x_{ii}} \prod_{1 \leq i \leq j \leq n} x_{ij}!}. \tag{1.3.1}$$

Proof. The total number of configurations is equal to $(\ell_n - 1)!!$. By construction of Algorithm 1.3.3, each configuration has equal probability, hence

$$Pr\Big(\mathrm{CM}_n(\mathbf{d}) = G\Big) = \frac{1}{(\ell_n - 1)!!} N(G) \tag{1.3.2}$$

where $N(G)$ is the number of configuration given rise to the same multigraph G, up to a permutation of the vertices labels.

Now, if we permutate the half-edges incident to a vertex, the resulting graph remains unchanged, but the configuration is ifferent. The factor $\prod_{i=1}^n d_i!$ accounts for this (it is the number of ways to permute the half-edges incident to all vertices).

Some of these permutations give the same configuration: this is taken into account by the factor $x_{ij}!$ (multiple edges between vertices i, j). The last factor $2^{x_{ii}}$ compensates the fact that pairing kl and lk gives the same overall configuration. □

Proposition 1.3.5. *Let $d_1, \dots, d_n$ be i.i.d., distributed according to some distribution d, and let $\gamma = \dfrac{\mathbb{E}(D(D-1))}{\mathbb{E}(D)}$. Then:*

- *the expected number of self loops is smaller than $\dfrac{\gamma}{2}$;*
- *the expected number of multi-edges is smaller than $\dfrac{\gamma}{4}$.*

Proof. We will admit it here. □

1.3.2 Preferential attachment model

Previous models are statics, in the sense that the number of nodes is fixed. Moreover, they don't really explain how interesting properties (degree distribution, etc.) can arise in the real graphs we observe. This section provides an example of random graphs where the nodes are added over time.

A first possibility is to imagine that G_n is an Erdős-Rényi graph $G(n,p)$, and a new node $n+1$ is added, and edges $(i,n+1)$ (for $i=1,\dots,n$) are added independently with probability p. The new graph G_{n+1} is thus a $G(n+1,p)$. (Note that G_n is a subgraph of G_{n+1}). The problem is that the degree sequence is binomial, hence doesn't fit what we observe in most of real networks.

The *preferential attachment paradigm* offers an intuitive explanation behind the power law degree distribution that we seem to observe in reality. Indeed, a new node $n+1$ will be connected to the n previous nodes by some additional edges. Theses edges $(i,n+1)$ are drawn independently, but with a probability proportional to the degree of the vertex i at that time. Thus, the new node $n+1$ will be more likely to be linked to vertices with large degrees.

Definition 1.3.6 (Preferential attachment - Informal definition)**.** At time t, new node will be connected to an old node i with a probability proportional to the degree $d_i(t)$ of the old node (at time t).

With that definition, we can draw few remarks:

- The old nodes will tend to have higher degrees than the new ones;
- *The rich get richer.* Indeed, new nodes tend to be attached to high degree old nodes, thus we expect the formation of hubs.

The fact that the graph will have hubs tend to make us think that the degree distribution will not be binomial, but will exhibit a power law. We will show that in the next section, just after giving a proper definition of the model.

Remark 1.3.7. The term *preferential attachment* comes from Barabási and Albert (1999), who proposed a model (but not totally well defined). The model is actually close to earlier work of Yule (1925) For a more rigorous treatment, one can see Bollobás et al. (2001) (and of course Hofstad (2016)).

Model definition

Definition 1.3.8. A sequence of graph $\Big\{G_t=(V_t,E_t), t\in\mathbb{N}\Big\}$ is said to be drawn under the preferential attachment model if:

- $|V_1|=1$ and $|E_1|=1$: at time $t=1$, we have one node v_1 with a single self-loop;
- At timestep $t+1$, we add one node (let us call him v_{t+1}) to the graph. This node will be linked to ONE node. The probability that the new node is connected to node v_i is given by:

$$\Pr\Big((v_{t+1},v_i)\in E_{t+1}\big|G_t\Big)=\begin{cases}\dfrac{1}{2t+1} & \text{if } v_i=v_{t+1}\\ \dfrac{D_i(t)}{2t+1} & \text{otherwise.}\end{cases} \tag{1.3.3}$$

Here $D_i(t)$ is the degree of node v_i at time t (recall that by convention, a self-loop increases the degree by 2).

Lemma 1.3.9. *After t timestep, the PA algorithm results in a network with $N = t$ nodes and t edges. In particular, the equation (1.3.3) defines a probability.*

Proof. Indeed, at each time step, we add one node, so $|V_t| = t$. Moreover, we add only one edge per time step, □

Remark 1.3.10. A more general version is given in Hofstad (2016). The one we gave here corresponds to the case $m = 1, \delta = 0$.

Study of the degree

Proposition 1.3.11. *Whent $t \to +\infty$, the Preferential Attachment model exhibits a power law degree distribution, of exponent 3.*

Proof. Let $p(k,s,t)$ be the probability that a vertex s has degree k at time t. The evolution of $p(k,s,t)$ is described by the *master equation*

$$p(k,s,t+1) = \frac{k-1}{2t+1} p(k-1,s,t) + \Big(1 - \frac{k}{2t+1}\Big) p(k,s,t) \tag{1.3.4}$$

with initial condition $p(k, s = 1, 1, t = 1) = \delta_{k,1}$ and boundary $p(k,t,t) = \delta_{k,1}$. Here the term $\dfrac{k-1}{2t+1}$ represents the probability that the new edge is linked to node s at time t (thus increasing the degree of s by 1), and $\big(1 - \dfrac{k}{2t+1}\big)$ the probability that the new edge is not linked to node s.

Let us sum over all nodes $s = 1, \dots, t$ in the networks at time t. By denoting $P(k,t)$ the total degree distribution of the entire network, *i.e.*:

$$P(k,t) = \frac{1}{t} \sum_{s=1}^{t} p(k,s,t), \tag{1.3.5}$$

we get:

$$(t+1)P(k,t+1) = \frac{k-1}{2t+1} tP(k-1,t) + \Big(1 - \frac{k}{2t+1}\Big) tP(k,t). \tag{1.3.6}$$

Thus the evolution of $P(k,t)$ can be written as:

$$(t+1)P(k,t+1) - tP(k,t) = \frac{t}{2t+1}\Big((k-1)P(k-1,t) - kP(k,t)\Big) + \delta_{k,1}. \tag{1.3.7}$$

When $t \to +\infty$, this equation for the stationary distribution reduces to

$$P(k) + \frac{1}{2}\Big(kP(k) - (k-1)P(k-1)\Big) = \delta_{k,1}, \tag{1.3.8}$$

where $P(k)$ would be $\lim\limits_{t \to +\infty} P(k,t)$. This last equation look like the differential equation

$$P(k) + \frac{1}{2}\frac{\mathrm{d}kP(k)}{\mathrm{d}k} = 0 \tag{1.3.9}$$

whose solution is

$$P(k) = Ck^{-3} \tag{1.3.10}$$

where C is a normalization factor, such that $\sum_k P(k) = 1$ ($C = \zeta(3)$ where ζ is the Riemann function). □

Remark 1.3.12. The previous proof is not totally rigorous, as it involved some approximations. One can refer to Hofstad (2016) for a more mathematically involved (but rigorous) proof, as well as deeper results on the Preferntial Attachment Model. Lastly, a nice note on the differential equation method can be found in Warnke (2019) (and references therein).

1.3.3 Random geometric graphs

Motivations : base station localisation in wireless and sensors networks ; features in a feature space.

Definition 1.3.13. Let $V = [n]$ be the set of vertices. Given a dimension d, for each vertex $u \in V$, we assign a vector $X_u \in \mathbb{R}^d$ chosen uniformly in $\mathcal{S}^{d-1}$ (the $d-1$ dimensional sphere of $\mathbb{R}^d$).

Then we assign an edge between two nodes i and j if an only if the distance $d(x_i, x_j)$ is less than some threshold $r > 0$.

Extension : Spatially Embedded Random Networks (SERN) - Barnett et al. (2007)

Definition 1.3.14 (SERN model)**.** A SERN ensemble of nodes is specified by the following:

1. A number n of nodes;
2. A metric space $(\mathcal{S}, d)$, where $\mathcal{S}$ is the space where the network resides;
3. A node distribution random variable X taking values in $\mathcal{S}$. X is to represent the location in $\mathcal{S}$ of a randomly (not necessarily uniformly) situated node. Note that the position of the nodes are not necessarily uniform;
4. A connectivity decay function $\gamma : \mathbb{R}^+ \to [0,1]$. $\gamma(s)$ represents the probability of assigning an edge to a pair of nodes at a distance s apart.

Given a n independent realizations $(x_1, \dots, x_n)$ of the random variable X, we assign independently an edge between two nodes i and j with a probability $\gamma(d(x_i, x_j))$.

Example 1.3.15. If $\mathcal{S}$ is the sphere, X is uniformly distributed along $\mathcal{S}$ and $\gamma = 1_{d(s) \le r}$ (for some $r \geq 0$), then we recover the Random Geometric Graphs model).

1.4 Exercises session

1. Propose a basic function (complexity in $O(n^2)$), who takes as parameter n and p, to generate the adjacency matrix of an Erdős-Rényi graph $G(n, p)$.
2. Now, propose an efficient algorithm (in complexity $O(|E|)$ where $|E|$ is the total number of edges) to generate large ER graph.
3. Find experimentally for an Erdős-Rényi graph the threshold of connectivity (for $n = 100$, $n = 1000$). You can use the *networkx* package to get basic function on graphs (such as *is_ connected*).

4. Write a function that generate a configuration model (given input parameters a degree sequence).

5. Simulate a graph generated according to Preferential Attachment Model, and find experimentally the exponent of the power law.

6. Simulate a Random Geometric Graph, find experimentally threshold of connectivity (consider the cases $d = 2$, $d = 3$).

7. Draw some Erdős-Rényi and Random Geometric Graphs using the package *networkx*.

Bibliography

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Barabási, A.-L. et al. (2016). *Network science*. Cambridge University Press.

Barnett, L., Di Paolo, E., and Bullock, S. (2007). Spatially embedded random networks. *Physical Review E*, 76(5):056115.

Bollobás, B. (1998). *Random Graphs*, pages 215–252. Springer New York, New York, NY.

Bollobás, B., Riordan, O., Spencer, J., and Tusnády, G. (2001). The degree sequence of a scale-free random graph process. *Random Structures & Algorithms*, 18(3):279–290.

Chung, F. and Lu, L. (2006). *Complex Graphs and Networks (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society, Boston, MA, USA.

Clauset, A., Shalizi, C., and Newman, M. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

Durrett, R. (2007). Random graph dynamics. *Random Graph Dynamics*.

Hofstad, R. v. d. (2016). *Random Graphs and Complex Networks*, volume 1 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press.

Janson, S., Luczak, T., and Rucinski, A. (2011). *Random graphs*, volume 45. John Wiley & Sons.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press.

Warnke, L. (2019). On Wormald's differential equation method. *arXiv e-prints*, page arXiv:1905.08928.

Yule, G. U. (1925). A mathematical theory of evolution, based on the conclusions of dr. j. c. willis, f. r. s. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213(402-410):21–87.

Community detection of networks

2.1 Reminders

2.1.1 Linear Algebra

Theorem 2.1.1 (Spectral Theorem)**.** *If M is symmetric, real valued, there exists an orthonormal basis consisting of eigenvectors of M. Moreover, the eigenvalues of M are real.*

Counterexample 2.1.2 (False if M has complex entries)**.** $M = \begin{pmatrix} 1 & i \\ i & -1 \end{pmatrix}$ is symmetric but not diagonalizable (indeed, from a direct computation of characteristic polynom, we can see that its only eigenvalue is 0).

Definition 2.1.3. A symmetric matrix M is said positive semidefinite (PSD) (resp. positive definite - PD) if $\forall x \in \mathbb{R}^n : x^T M x \geq 0$ (resp. $x^T M x > 0$).

Example 2.1.4. For all $M \in \mathbb{R}^{n \times n}$, the matrix $M^T M$ is symmetric definite positive.

Lemma 2.1.5. *Let M be a symmetric matrix, and $\lambda_1, \dots, \lambda_n$ its (real) eigenvalues. M is positive semidefinite (resp. positive definite) iff $\lambda_i \geq 0$ (resp $\lambda_i > 0$).*

2.1.2 Norms

Definition 2.1.6. Let E be a vector space. A function $N : E \to \mathbb{R}$ is a norm if:

1. (positivity) $\forall x \in E : N(x) \geq 0$;

2. (definite) $N(x) = 0 \Rightarrow x = 0_E$;

3. (homogeneity) $\forall x \in E, t \in \mathbb{R} : N(tx) \leq |t| N(x)$;

4. (triangle inequality) $\forall x, y \in E : N(x + y) \leq N(x) + N(y)$.

Vector norms

Proposition 2.1.7. *Let $E = \mathbb{R}^n$ and $p > 0$, we define the p-norms as: $||x||_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$.*

Proof. $||.||$ verify the first three rules (easy to check). The triangle inequality holds thanks to Minkowski's inequality. □

Example 2.1.8. Let $X = (x_1, \dots, x_n)^T \in \mathbb{R}^n$.

1. For $p = 1$: $||X||_1 := \sum_{i=1}^{n} |x_i|$.
2. For $p = 2$, $||X||_2 := \sqrt{\sum_{i=1}^{n} |x_i|^2}$. This is the Euclidian norm. If we introduce the scalar product : $< X, Y >= X^T Y$ we have $||X||_2^2 = X^T X$.
3. $||X||_\infty := \lim_{p \to +\infty} ||X||_p = \max\{|x_1|, \dots, |x_n|\}$.

Matrix norms Serre (2010)

Definition 2.1.9. Let $||.||$ be a norm on $\mathbb{R}^n$, we define the operator norm $|||.|||$ on $\mathbb{R}^{n\times n}$ induced by $||.||$ as:

$$|||A||| = \sup_{X \neq 0_n} \frac{||AX||}{||X||}.$$

Very often, by abuse of notation, we denote $||.||$ the operator norm (instead of $|||.|||$).

Lemma 2.1.10. $|||A||| = \sup_{||X||=1} ||AX|| = \sup_{||X||\leq 1} ||AX|| = \max_{||X||\leq 1} ||AX||$.

Example 2.1.11. Let $A \in \mathbb{R}^{n\times n}$.

1. $||A||_1 = \sup_{||X||_1=1} ||AX||_1 = \max_{j=1\dots n} \sum_{i=1}^{n} |A_{ij}|$ (max column-sum);
2. $||A||_\infty = \sup_{||X||_\infty=1} ||AX||_\infty = \max_{i=1\dots n} \sum_{j=1}^{n} |A_{ij}|$ (max row-sum);
3. $||A||_2 = \sup_{X^TX=1} \sqrt{X^T A^T A X} = \sqrt{\lambda_{\max}(A^T A)}$.
4. If A is invertible, $||A^{-1}||_2 = \dfrac{1}{\lambda_{\min}(A^T A)}$.

Proposition 2.1.12. *Let $||.||$ be a matrix norm. Then* $\forall A, B \in \mathbb{R}^{n\times n} : ||AB|| \leq ||A||.||B||$.

Counterexample 2.1.13. False in general (if the norm isn't induced from a vector norm). Let $N(A) = \max_{i,j} |a_{ij}|$ (to not be confused by $|||.|||_\infty$), and $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. Then, $N(A^2) = 2 > N(A)N(A) = 1$.

Definition 2.1.14. We denote $||A||_F := \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |A_{ij}|^2}$ the Frobenius norm

2.1.3 Optimization

Proposition 2.1.15. *Let M be a $n \times n$ symmetric matrix, and $\lambda_1 \leq \cdots \leq \lambda_n$ eigenvalues of A, with associated eigenvectors $v_1, \dots, v_n$.*

$$\lambda_1 = \min_{\substack{X \in \mathbb{R}^n \\ ||X||=1}} X^T M X = \min_{\substack{X \in \mathbb{R}^n \\ X \neq 0_n}} \frac{X^T M X}{X^T X} \tag{2.1.1}$$

$$\lambda_2 = \min_{\substack{X \in \mathbb{R}^n \\ ||X||=1 \\ X \perp v_1}} X^T M X = \min_{\substack{X \in \mathbb{R}^n \\ X \neq 0_n \\ X \perp v_1}} \frac{X^T M X}{X^T X} \tag{2.1.2}$$

$$\lambda_n = \max_{\substack{X \in \mathbb{R}^n \\ ||X||=1}} X^T M X = \max_{\substack{X \in \mathbb{R}^n \\ X \neq 0_n}} \frac{X^T M X}{X^T X} \tag{2.1.3}$$

Moreover, the respective $\arg\min$ *are v_1, v_2 and v_n (properly normalized).*

Proof. Let us give two proof, one by diagonalizing the matrix M, and one using calculus (Lagrange minimizers).

- M being symmetric, we can write $M = P^T D P$ Let $Y = PX$ Note that $||Y|| = ||X||$, thus the constraint $||X|| = 1$ becomes $\sum_{i=1}^n y_i^2 = 1$. Since $X^T M X = Y^T D Y = \sum_{i=1}^n \lambda_i y_i^2$, this expression is minimized (given the constraint) when all y_i are zeros except for $y_1 = 1$, and maximized when $y_n = 1$ and all others y_i are 0.

- Let us use the Lagrange multipliers. The Lagrangian associated to the minimization problem (2.1.1) (or (2.1.3)) is $\mathcal{L}(X, \lambda) = X^T M X - \lambda\Big(X^T X - 1\Big)$. Note that letting $\dfrac{\partial \mathcal{L}}{\partial \lambda} = 0$ gives back the constraint $||X|| = 1$. Moreover: $\dfrac{\partial L}{\partial X} = 2MX - 2\lambda X = 0 \Rightarrow MX = \lambda X$, and thus X is an eigenvector of M and λ the corresponding eienvalue. As equation (2.1.1) is a minimization problem, the solution it the smallest eigenvalue. Similarly, the solution of equation (2.1.3) is the largest eigenvalue.

□

Proposition 2.1.16. *Let $M \in \mathbb{R}^{n \times n}$ a symmetric matrix with $v_1, \dots, v_n$ an orthonormal basis of eigenvectors associated to $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The solution of:*

$$\arg\min_{\substack{H \in \mathbb{R}^{n \times K} \\ H^T H = I_K}} = \mathrm{Tr}\left(H^T M H\right) \tag{2.1.4}$$

is given by $H = [v_1, \dots, v_K]$.

Proof. Let the Lagrangian $\mathcal{L}(H, \Lambda) := \mathrm{Tr}\left(H^T M H\right) - \mathrm{Tr}\Big(\Lambda^T\left(H^T H - I_K\right)\Big)$ where $\Lambda \in \mathbb{R}^{K \times K}$ a diagonal matrix whose entries are the Lagrange multipliers.

We have: $\dfrac{\partial \mathcal{L}}{\partial H} = 2MH - 2H\Lambda = 0$ lead to $MH = H\Lambda$. Thus, the colums of H are indeed eigenvectors of M, and the diagonals elements of Φ the corresponding eigenvalues. □

2.1.4 Graph Laplacian(s)

In the following, we will call G a graph, on vertex set $V = \{1, \dots, n\}$. Let A be the adjacency matrix, and D the degree matrix of G.

Definition 2.1.17 (Graph Laplacian)**.** We define:

- The Laplacian $L = D - A$;
- The normalized Laplacian $\mathcal{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$.

Remark 2.1.18. Note that $D^{-1/2}$ is not well define if there is a isolated node (degree 0). We can either assume there is no such node in our graph, or by convention let $D_{ii}^{-1/2} = 0$ if i isolated.

Lemma 2.1.19. *If i and j are two neighboring nodes, we note $i \sim j$. Assume the graph doesn't have self-loops. We have:*

- $L_{ij} = \begin{cases} d_i & \text{if} \quad i = j, \\ -1 & \text{if} \quad i \sim j, \\ 0 & \text{otherwise.} \end{cases}$
- $\mathcal{L}_{ij} = \begin{cases} 1 & \text{if} \quad i = j, \\ -\dfrac{1}{\sqrt{d_i d_j}} & \text{if} \quad i \sim j, \\ 0 & \text{otherwise.} \end{cases}$

Proof. Direct from the definitions of L and $\mathcal{L}$. □

Proposition 2.1.20 (Properties of the Laplacian)**.** *$L := D - W$ satisfies the following properties.*

1. *For any vector $X \in \mathbb{R}^n$, $X^T L X = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (x_i - x_j)^2$;*
2. *L is symmetric, positive semi-definite;*
3. *The smallest eigenvalue of L is 0, corresponding to the eigenvector 1_n;*
4. *L has n non-negative, real valued eigenvalues $0 = \lambda_1 \leq \cdots \leq \lambda_n$.*

Proof. 1. Recall $d_i = \sum_{j=1}^{n} w_{ij}$. We have:

$$\begin{aligned} \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (x_i - x_j)^2 &= \frac{1}{2} \Big(\sum_{i,j} w_{ij} x_i^2 - 2 \sum_{i,j} w_{ij} x_i x_j + \sum_{i,j} w_{ij} x_j^2 \Big) \\ &= \frac{1}{2} \Big(\sum_{i=1}^{n} d_i x_i^2 - 2 \sum_{i,j=1}^{n} x_i x_j w_{ij} + \sum_{j=1}^{n} d_j x_j^2 \Big) \\ &= \sum_{i=1}^{n} d_i x_i^2 - \sum_{i,j=1}^{n} x_i x_j w_{ij} \\ &= X^T D X - X^T W X \\ &= X^T L X. \end{aligned}$$

2. L symmetric because D and W are. From point 1, we have $X^T LX \geq 0$, so L is positive semi-definite.

3. $L1_n = 0$.

4. L is symmetric, so its eigenvalues are real. It is positive semi-definite, so its eigenvalues are non-negative.

□

Definition 2.1.21 (Indicator vector of a set)**.** Let W be a subset of the node set V. We define 1_W as the $n \times 1$ vector such that $1_W)_i = 1$ if $i \in W$, and 0 if $i \notin W$. We let 1_n be the vector of all ones.

Lemma 2.1.22. *$LX = 0 \Leftrightarrow X$ is constant on each connected component of G.*

Proof. Let $V_1, \ldots, V_K$ be the connected components of G.

- Assume $LX = 0$. Then $X^T LX = 0$, and from formula of previous proposition it comes (after a few steps) that $\forall i, j \in V_k, x_i = x_j$. We can conclude that $LX = 0 \Rightarrow X$ is constant on each connected component of the graph.

- Reciprocally, we can see from direct computation that if X is constant on each V_k, then $LX = 0$.

K: number of eigenvalue which are equal to zero it means how many connected component we have!

□

Proposition 2.1.23 (Number of connected components)**.** *Let G be an undirected graph with non-negative weights. Then, the multiplicity k of the eigenvalue 0 of L is equal to the number of connected components $V_1, \ldots, V_k$. Moreover, the eigenspace of eigenvalue 0 (Ker L) is spanned by the indicator vectors $1_{V_1}, \ldots, 1_{V_k}$.*

Proof.

- If $k = 1$, it means the only eigenvector of 0 is $X = 1_n$, and the graph is connected.

- Now suppose $k > 1$. We can assume that the vertices are ordered according to the connected components they belong to. Thus, $L = \text{diag}(L_1, \ldots, L_k)$ where L_i is the Laplacian of the $i\!-\!th$ connected component. Each L_i has eigenvalue 0 with multiplicity 1, and the corresponding eigenvector is the constant one vector. Thus, $L1_{A_i} = L_i 1_n = 0$, and each 1_{1_i} is eigenvector of L associated to 0.

□

Example 2.1.24. Assume the graph is connected. Then there is only one connected component ($k = 1$), so $\dim \text{Ker}\, L = 1$, and the corresponding eigenspace is spanned by 1_n.

2.2 First examples and motivation

- Although community structures are quite common in real networks, there is no clear definition of what a community is;

- Computationally difficult;

- Problem in the evaluation of algorithms (still open question ?).

2.2.1 What is a community

No clear definition, but a few leads.

- **Definitions Based on Node Similarity**: communities are groups of nodes that are similar to each other. One would then choose a similarity measure, for example the commute time between nodes, as a similarity measure.

- **Local Definitions**: for example nodes that interact a lot. Communities would then be groups of nodes that are relatively densely connected to each other, but sparsely connected to the rest of the graph.

- **Global Definitions**: We will evaluate the quality of a graph partition into disjoint communities using a quantity called modularity. The most popular modularity is proposed by Newman and Girvan, which compares the number of edges inside the community to the expected number of internal edges in the null model.

2.2.2 First basic example

Assume the graph is disconnected into three connected components V_1, V_2 and V_3. Then, L has eigenvalue 0 with multiplicity 3, and eigenvectors $1_{V_1}, 1_{V_2}$ and 1_{V_3}.

Although this example is too easy, it highlights the fact that the eigenvectors of L can be used to find the communities.

Now assume G is connected, the nodes are partitioned into 3 communities, and there isn't so many edges accross communities. Then the Laplacian can be expressed as $L = \tilde{L} + \delta L$ where $\tilde{L}$ correspond to the graph were we took off the out-community edges, and δL the perturbation term (due to the out-community edges). Then the three first eigenvectors of $\tilde{L}$ are the indicator vectors of the community. Of course, we do not observe $\tilde{L}$, but L. If the perturbation term is small, we should expect that the eigenvectors of L would be close to the one of $\tilde{L}$, and will help us to recover the community structure.

This is for now a lot of hand-waving, and will be studied more formally in the next section.

2.2.3 How to compare those methods ?

Real networks performances

Not many of them since need the ground truth Popular networks include:

- Zachary Karate Club;

- Dolphins network;

- Political Blog Dataset;

- Other networks are available on Mark Newman[1] personal webpage: http://www-personal.umich.edu/~mejn/netdata/

- For larger database, one may consult *Stanford Large Network Dataset Collection* at https://snap.stanford.edu/data/

[1] There is several homonyms called Mark Newman.

Synthetic networks

Due to the fact that not many real world networks (with ground truth) are available, we can also use synthetic networks. A popular model of random graph with community structure (although not realistic) is the Stochastic Block Model (SBM).

Definition 2.2.1 (SBM)**.** Let $G = (V, E)$ be a graph constructed as follow.

- Input : number of nodes, number of communities K, rate matrix $P \in \mathbb{R}^{K\times K}$ (symmetric matrix).
- Output: a random graph with n nodes and a community assignment vector (ground truth) $\sigma \in \{0, \dots, K\}^n$.
- For each node $i \in V$, assign a it to a community σ_i with probability α_i.
- Given the community assignment σ

Other models exists, like the Degree Corrected Stochastic Block Model (which, like its name indicates, correct the degree, for example to have a power law degree distribution), , etc.

2.3 Cut-based methods [Von Luxburg (2007)]

2.3.1 Graph bisection (two same size clusters)

Consider an undirected but possibly weighted graph, $w_{ij} = w_{ji} \geq 0$. Define the degree d_i of a node $i \in V$ as $\sum_{j=1}^n w_{ij}$.

Intuition: we want to separate the nodes into two groups, such that inside each group, there is lot of edges, and between two groups there is not so much edges.

More formally, we want to partition the node set V into two subsets A_1, A_2 such that $A_1 \cap A_2 = \emptyset$ (non overlapping clusters) and $A_1 \cup A_2 = V$. Remark : $A_2 = A^c$ (complementary of A).

Definition 2.3.1. For two sets of nodes A, B, we define $\operatorname{cut}(A, B) = \sum_{i\in A, j\in B} w_{ij}$ as the number of edges going from A to B.

Thus, we want to solve

$$\operatorname*{arg\,min}_{A\subset V} \sum_{i\in A, j\in A^c} w_{ij}. \tag{2.3.1}$$

The solution of this minimization problem is $A = V$ (and $A = \emptyset$). Even by imposing $A \neq V$ and $A \neq \emptyset$, very often, this would lead to a solution where almost all the node are in one cluster, apart for a few in the other cluster.

So, we need to penalize for such imbalanced solutions, by imposing that the size of the sets A and A^c are roughly of the same size. In a first time, consider the minimization problem over the set $|A| = |V|/2$. This is called the *graph bisection problem*. The general case will be done directly in the subsection with K clusters.

Another problem : the minimization is over all subset of V (of size $n/2$); there is $2^{n/2}$ of them, and was shown to be NP-hard (Wagner and Wagner (1993), Garey et al. (1974)). Thus, we will use a relaxation approximation, by using the next Theorem.

Theorem 2.3.2. *For $A \subset V$ such that $|A| = |V|/2$, define the cut vector χ_A such that $(\chi_A)_i = 1$ if $i \in A$, $(\chi_A)_i = -1$ otherwise. We have:*

$$\underset{A \subset V : |A| = |V|/2}{\arg\min} \ \mathrm{cut}(A, A^c) = \underset{A \subset V : |A| = |V|/2}{\arg\min} \ \chi_A^T L \chi_A. \qquad (2.3.2)$$

Moreover, $\chi_A \perp 1_n$ and $||\chi_A||^2 = n$.

Proof. $\chi_A \perp 1_n$ and $||\chi_A||^2 = n$: left as exercise. Then, let us see that

$$\begin{aligned} \mathrm{cut}(A, A^c) &= \sum_{i \in A, j \in A^c} w_{ij} \\ &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \Big((\chi_A)_i - (\chi_A)_j \Big)^2 \\ &= \frac{1}{4} \chi_A^T L \chi_A. \end{aligned}$$

The factor $\frac{1}{4}$ doesn't change the minimization solution, and this settles the proof. □

Thus, minimizing $\mathrm{cut}(A, A^c)$ is equivalent of minimizing $\chi^T L \chi$, where $\chi \in \{-1; 1\}^n$ and $\chi \perp 1_n$. Thus, the relaxed problem is:

$$\widehat{X} := \underset{\substack{X \in \mathbf{R}^n; \\ ||X||_2^2 = n; \\ X \perp 1_n}}{\arg\min} X^T L X.$$

By relaxation, we mean we went from $\chi \in \{-1; 1\}^n$ to a real value vector $X \in \mathbb{R}^n$ (with proper constraints added). This allow us to use calculus method to solve the arg min problem (cf. following Lemma). Once $\widehat{X}$ computed, we can assign cluster according to the sign of $\widehat{X}_i$. This leads to Spectral Clustering algorithm.

Lemma 2.3.3. *The solution of $\widehat{X} := \underset{\substack{X \in \mathbf{R}^n; \\ ||X||_2^2 = n; \\ X \perp 1_n}}{\arg\min} X^T L X$ is the second eigenvector of L (normalized so its norm-2 is n).*

Proof. Indeed, 1_n is the first eigenvector of L, and we conclude by the Courant-Fischer theorem. □

```
Algorithm 1: Standard Spectral Clustering – 2 clusters.
  Input: Laplacian $L$.
  Output: Clustering assignment $Z \in \{0; 1\}^n$.

  Spectral Step:
    • Let $v_2$ be the second eigenvector of $L$ (associated to second smallest
      eigenvalue)
    • for $i = 1...n$, assign $Z_i := 1$ if $(v_2)_i > 0$, and $Z_i = 0$ otherwise.
```

2.3.2 General case: more than two clusters

Let $A_1, \dots, A_K$ be a partition of V into K (non-overlapping) clusters. We saw that we need to penalize for unbalanced solutions. To measure the size of the clusters A_k, we define the two following metrics:

$$\begin{aligned} |A_k| &:= \text{the number of nodes in } A_k; \\ \mathrm{vol}(A_k) &:= \sum_{i \in A_k} d_i. \end{aligned}$$

and instead of minimizing directly the cut, we will minimize on of those two quantitites:

$$\mathrm{ratioCut}(A, A^c) = \sum_{k=1}^{K} \frac{\mathrm{cut}(A_k, A_k^c)}{|A_k|}; \tag{2.3.3}$$

$$\mathrm{Ncut}(A, A^c) = \sum_{k=1}^{K} \frac{\mathrm{cut}(A_k, A_k^c)}{\mathrm{vol}(A_k)}. \tag{2.3.4}$$

Similarly to the previous part, minimizing those two quantities for all partitions $(A_1, \dots, A_k)$ is NP-hard, and we will solve a relaxed version of the problem. Let us define the matrix $H = (h_{ik}) \in \mathbb{R}^{n \times k}$ by:

$$i \in \{1, \dots, n\}, k \in \{1, \dots, K\} : \quad h_{ik} = \begin{cases} \dfrac{1}{\sqrt{|A_k|}} & \text{if} \quad v_i \in A_k \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.5}$$

H is a matrix containing the K indicators vectors as column, where the size of each set A_k is used as penalization term. Similarly, let us define $N = (n_{ij}) \in \mathbb{R}^{n \times K}$ as:

$$i \in \{1, \dots, n\}, k \in \{1, \dots, K\} : \quad n_{ij} = \begin{cases} \dfrac{1}{\sqrt{\mathrm{vol}(A_k)}} & \text{if} \quad v_i \in A_k \\ 0 & \text{otherwise.} \end{cases} \tag{2.3.6}$$

Here we used the volume of each set A_k as a penalization term. We have the following proposition.

Proposition 2.3.4.

- *$\mathrm{ratioCut}(A_1, \dots, A_K) = \mathrm{Tr}(H^T L H)$;*
- *$\mathrm{Ncut}(A_1, \dots, A_K) = \mathrm{Tr}(N^T L N)$;*
- *$H^T H = I_K$ and $N^T D N = I_K$.*

Proof. This proposition follows from the fact that:

- $(H^T L H)_{kk} = h_{.k}^T L h_{.k} = \dfrac{\mathrm{cut}(A_k, A_k^c)}{|A_k|}$;
- $(N^T L N)_{kk} = N_{.k}^T L N_{.k} = \dfrac{\mathrm{cut}(A_k, A_k^c)}{\mathrm{vol}(A_k)}$.

For example, let us highlight the first point.

$$
\begin{aligned}
h_i^T L h_i &= \frac{1}{2} \sum_{i,j} a_{ij} \big(h_{ik} - h_{jk} \big)^2 \\
&= \frac{1}{2} \Big(\sum_{i,j \in A_k} a_{ij} + \sum_{i,j \notin A_k} a_{ij} + \sum_{i \in A_k j \notin A_k} a_{ij} + \sum_{i \notin A_k j \in A_k} a_{ij} \Big) \\
&= \frac{1}{2} \Big(\sum_{i \in A_k, j \notin A_k} a_{ij} + \sum_{i \notin A_k, j \in A_k} a_{ij} \Big) \frac{1}{|A_k|} \\
&= \frac{1}{2} 2 \operatorname{cut}(A_k, A_k^c) \frac{1}{|A_k|}.
\end{aligned}
$$

The second lines comes from the fact that $h_{ik} = h_{jk}$ if $i, j \in A_k$ or $i, j \notin A_k$. □

Proposition 2.3.5. *Minimizing the ratioCut can be rewritten as:*

$$\underset{A_1,\dots,A_k}{\arg\min} \operatorname{Tr}(H^T L H), \tag{2.3.7}$$

where H is defined in equation (2.3.5). Similarly, minimizing the normalized Cut can be rewritten as:

$$\underset{A_1,\dots,A_k}{\arg\min} \operatorname{Tr}(U^T \mathcal{L} U), \tag{2.3.8}$$

where $U := D^{1/2} N$ and N is defined in equation (2.3.6).

Note that $U^T U = I_K$. Thus, the relaxation of those two equations give respectively Spectral Clustering and Normalized Spectral Clustering.

Proposition 2.3.6. *The solution of* $\arg\min Tr(H^T L H)$ *where $H \in \mathbb{R}^{n \times K}$ is subject to $H^T H = I$ is given by the matrix H whose columns are the first K eigenvectors of L.*

To reconvert the real values solution matrix to a discrete partition, the standard way is to apply K-means algorithm on the rows of H. One could simply assign i to $\arg\max_{k \in \{1,\dots,K\}} \phi_{ik}$, but experimentally we observe that this extra K-means step improves (a bit) the performances. For speed purposes, we can even initialize K-means using the cluster given by the simple detection rule.

Algorithm 2: (Normalized) Spectral Clustering.

```
Input: Laplacian $L$ (resp. normalized Laplacian $\mathcal{L}$), number of clusters $K$.
Output: predicted clusters $A_1, \dots, A_K$.

Spectral Step:
    • Compute $v_1, \dots, v_K$ the first $K$ eigenvectors of $L$ (resp. $\mathcal{L}$);
    • Let $V \in \mathbb{R}^{n \times K}$ be the matrix whose column $i$ is $v_i$;
Clustering Step:
    • For $i \in \{1, \dots, n\}$, let $\phi_i$ be the $i$-th row of $V$;
    • Cluster the points $(y_i)_{i=1,\dots,n}$ using $K$-means algorithm, giving $C_1, \dots, C_K$;
    • Return $A_1, \dots, A_K$, such that $A_k := \{i : \phi_i \in C_k\}$.
```

2.3.3 Discussion / Heuristics

Complexity

We need to compute the K eigenvectors of the matrix, which can be done in $O(Kn^3)$ using standard methods (Gauss, or some matrix factorization like QR / LU / Cholesky). See for example Serre (2010).

Using iterative algorithm (*e.g.,* Lanczos algorithm, which is the one implemented in the library *scipy.sparse.linalg.eigsh*), one can achieve (usually) good in $O(Km)$ where m is the number of non-zeros element of the matrix considered.

Drawbacks

- Need to know the number of clusters K in advance. And it is not clear how the method will perform if we plus an estimated K.
- Tend to fail for sparse graphs, and most graphs are sparse.
- When it fails, it usually fail big, *i.e.,* put almost all nodes in one cluster, other clusters being almost empty. Thus, it is easy to spot when the method fails/

Why does it fail ?

Among the small eigenvectors, some might associated to something totally different than the correct clusters.

- If we take the standard Laplacian, those noisy eigenvectors will be concentrated around high degree nodes. Since real graph tend to have "a lot" of hubs[2], it is a problem, and people prefer using the normalized Laplacian.
- But, for the normalized Laplacian, the correct eigenvector order breaks because of low degree nodes, and especially because of dangling trees.

There is two way to solve this problem.

- Don't look at the second eigenvectors, but at the 3rd, 4rd, 5th, etc one (see for example *Political Blog dataset*, where the correct clusters information is in the 3rd eigenvector). But it can be non trivial to determine which eigenvector one should look at (especially if $K > 2$ clusters).
- *Regularization technique*: add a perturbation term. Consider $A_\tau := A + \frac{\tau}{n} 1_n^T 1_n$ instead of A, and perform Spectral Clustering on $\mathcal{L}_\tau := I - D^{-1/2} A_\tau D^{-1/2}$, with τ an extra (small) parameters (typically $\tau = 1$ or $\tau = d$ where d is the average degree of the graph). This new adjacency matrix corresponds to the graph where we added an edge of weight $\frac{\tau}{n}$ between all nodes pairs. This tend to bring back the dangling trees to the rest of the graph, hence restoring order in the eigenvectors.

[2] Remember the power law degree distribution?

Note that the regularization technique can also partially solve the failure of Spectral Clustering in sparse graph, as adding the extra term makes the graph less sparse (of course this has a limit, as the noise term should be low enough to not destroy all the structure of the original graph). For some theoretical results about that fact, see Le et al. (2017).

2.4 Modularity-based methods

2.4.1 Definition

We will define a measure, called modularity, that aimed to compare the density of links of our cluster assignation with the one one would get if the graph was random. By optimizing the modularity measure over the space of all partitions, one expects to identify groups of nodes that are more densely connected to each other than one would expect according to a statistical null model of the graph. This statistical null model is commonly chosen to be the configuration model with the graph degree sequence.

Definition 2.4.1 (Girvan-Neuman modularity)**.** Given a vector $C \in \{1; \dots; K\}^n$ such that C_i is the predicted community for node i, the modularity of C is defined as:

$$\mathcal{M}(C) := \frac{1}{2m} \sum_{i,j} \big(A_{ij} - P_{ij}\big) \delta_{C_i C_j}, \tag{2.4.1}$$

where $m = |E|$ (number of edges) and $P_{ij} = \dfrac{d_i d_j}{2m}$.

We can also write $\mathcal{M}(C) = \dfrac{1}{2m} \sum_{k=1}^{K} \sum_{i,j : C_i = C_j = k} \big(A_{ij} - P_{ij}\big)$.

Remark 2.4.2.

- The factor $1/(2m)$ is here so that $\mathcal{M}(C) \in [-1; 1]$.
- P_{ij} is the expected number of edges between i and j if the graph was drawn from a configuration model. Indeed, node i has d_i outgoing edges, and the probability that one of this edge goes to node j is $d_j/(2m)$, $2m$ being the total number of end of edges in the network.
- Assume all nodes are in one community. Then $\mathcal{M}(C) = 0$.
- Assume all nodes are in there own community. Then $\mathcal{M}(C) \leq 0$.
- Good values for modularity typically lies in the range $[0.3; 0.7]$. Unfortunatelly, optimizing the modularity is NP-complete Brandes et al. (2007).

Additional notes on modularity

Lemma 2.4.3. $\mathcal{M}(C) = \sum_{c=1}^{K} \left(\frac{\ell_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right)$ *where* ℓ_c *is the number of edges in cluster* c*,* K *the number of cluster and* d_c *the sum of degrees in cluster* c*.*

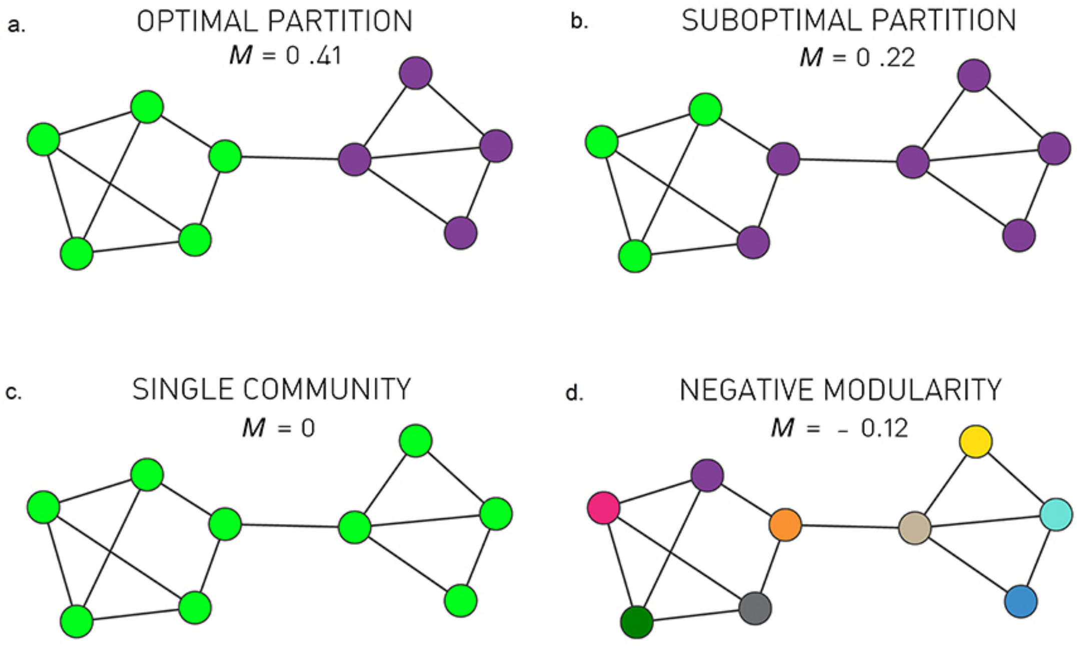

Figure 2.1: Modularity $\mathcal{M}$ defined in 2.4.1 for several partitions of a network with two obvious communities. The figure is taken from Barabási et al. (2016).

Previous Lemma can be rewritten as $\mathcal{M}(C) = \sum_{c=1}^{K} e_{cc} - a_c^2$, where $e_{cc'}$ is the fraction of edges that connect nodes in cluster c to nodes in c', i.e., $e_{cc'} = \frac{1}{m} \sum\limits_{i:C_i=c} \sum\limits_{j:C_j=c'} a_{ij}$ and $a_c = \sum\limits_{c'=1}^{K} e_{cc'} = \frac{d_c}{2}$ is the number of edges inside cluster c.

Lemma 2.4.4. *The change of modularity upon joining two communities, say c and c', is given by $\Delta Q = e_{cc'} + e_{c'c} - 2a_c a_{c'} = 2(e_{cc'} - a_c a_{c'})$, which can be calculated in constant time.*

2.4.2 Greedy algorithm

The first modularity maximization algorithm, proposed by Newman (2004), and reproduced here (Algorithm 3) iteratively joins pairs of communities if the move increases the partition's modularity. Some extension have been proposed, see for example Clauset et al. (2004), but those have been completely outperformed by Louvain algorithm (see Subsection 2.4.3).

Proposition 2.4.5. *Greedy Algorithm 3 runs in $O\big((|E| + n)n\big)$.*

Proof. By Lemma 2.4.4, the computation $\Delta\mathcal{M}$ is done in constant time. At the initial update step, we have $|E|$ of such computation to do (and then each update step, less than $|E|$ since we merge the communities). Then, after identifying the max $\Delta\mathcal{M}$ (which is done during the computations of all the $\Delta\mathcal{M}$), we need to recompute the adjacency matrice, and this can take up to $O(n)$ operations. Finally, we need to do the update step $n-1$ times, so the complexity is of the order $n-1$ times $|E| + n$. □

Algorithm 3: Greedy algorithm for modularity minimization.

```
Input: Adjacency matrix $A$.
Output: Node labeling $\mathbf{z} = (z_1, \dots, z_n)$.

Initialize: Assign each node to a community of its own, starting with $n$
            communities of single nodes (in other words, set $z_i = i$).

Update:
for each community pair connected by at least one edge do
    Compute the modularity difference $\Delta\mathcal{M}$ obtained if we merge the two
      communities.
    Identify the community pair for which $\Delta\mathcal{M}$ is the largest and merge
      these two communities. (Modularity is always calculated for the full
      network. Note that $\Delta\mathcal{M}$ can be negative)
Repeat the Update step, recording $\mathcal{M}$ at each step.
Stop when all nodes are merged into a single community.
Return the partition for which $\mathcal{M}$ is maximal
```

2.4.3 Louvain algorithm

Algorithm 4: Louvain algorithm for modularity minimization. Based on Blondel et al. (2008).

Input: Adjacency matrix A.
Output: Node labeling $\mathbf{z} = (z_1, \dots, z_n)$.

Step I:

- Assign each node to a community of its own, starting with n communities of single nodes (in other words, set $z_i = i$).
- For each node i, evaluate the gain in modularity if we place node i in the community of one of its neighbors j.
- Move node i in the community for which the modularity gain is the largest, but only if this gain is positive. If no positive gain is found, i stays in its original community.
- Apply this process to all nodes until no further improvement can be achieved. **In particular, each node can be moved several time.**

Step II: Construct a network whose nodes are the communities identified in Step I, and where:

- the weight between two communities is the sum of the weights of the links between the nodes in the corresponding communities;
- The link between nodes of the same community lead to weighted self-loops.

Step II being completed, repeat **Step I** and then **Step II** (we call it a **pass**). Each pass decreases the number of communities. The passes are repeated until there are no more changes and the maximum modularity is attained.

Remark 2.4.6. The most time consuming pass is the first one, where we have $|E|$ change of modularity to compute. The following passes are faster, as they will deal with much smaller graph. Thus the complexity is $O(|E|)$.

Remark 2.4.7. The method is called Louvain because the authors, at that time, were from Louvain University, in Belgium. See the original paper: Blondel et al. (2008).

2.4.4 Discussion

- No need to know K in advance: big advantage over Spectral Methods.
- In practice, the speed difference make the greedy method (Algorithm 3) completely out of the competition. Plus, it has been observed that Louvain actually gives better clusters (*i.e.,* clusters with higher modularity)!
- The result should depend on the numbering order of the nodes. In practice, not so much, and if we worry about that, we can run several time Louvain algorithm to see if the result change a lot or not.
- A good and easy to read paper about the Louvain method: Good et al. (2010).
- A paper about the usage of Louvain method for content recommendations in Reddit: Jamonnak et al. (2015).

Nontheless, Louvain is not guarantee to give the global minima of the modularity (since this is NP-hard!). Moreover, even if we get the global minima, it is not guarantee to correspond to the correct cluster assignment. In particular, the modularity metric tend to merge small communities into larger ones. This is known as the *resolution limit.* (Note this is a drawback of the modularity metric, not of Louvain algorithm).

- From Louvain, we can look not at the last pass result, but at intermediate steps. This gives a *hierarchical clustering.*
- We can add a *resolution parameter* γ, i.e., change the modularity into $\mathcal{M} = \frac{1}{2m}\sum_{i,j}\delta_{C_i,C_j}(A_{ij} - \gamma P_{ij})$. See for example Reichardt and Bornholdt (2006) and Arenas et al. (2008).

Bibliography

Arenas, A., Fernandez, A., and Gomez, S. (2008). Analysis of the structure of complex networks at different resolution levels. *New journal of physics*, 10(5):053039.

Barabási, A.-L. et al. (2016). *Network science.* Cambridge University Press.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2007). On modularity clustering. *IEEE transactions on knowledge and data engineering*, 20(2):172–188.

Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Garey, M. R., Johnson, D. S., and Stockmeyer, L. (1974). Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM.

Good, B. H., De Montjoye, Y.-A., and Clauset, A. (2010). Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106.

Jamonnak, S., Kilgallin, J., Chan, C.-C., and Cheng, E. (2015). Recommenddit: A recommendation service for reddit communities. In *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 374–379. IEEE.

Le, C. M., Levina, E., and Vershynin, R. (2017). Concentration and regularization of random graphs. *Random Structures & Algorithms*, 51(3):538–561.

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133.

Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1):016110.

Serre, D. (2010). *Matrices, Theory and Applications*. Springer-Verlag New York.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

Wagner, D. and Wagner, F. (1993). Between min cut and graph bisection. In *International Symposium on Mathematical Foundations of Computer Science*, pages 744–750. Springer.