Statistical analysis of graphs

Catherine Matias

CNRS - Laboratoire de Probabilités, Statistique et Modélisation, Paris
catherine.matias@math.cnrs.fr
`http://cmatias.perso.math.cnrs.fr/`

USP- October 2023

Outline

- Part I: Introduction and basics
- Part II: Random graphs models
- Part III: Community detection

Part I

Introduction and basics

Outline Part 1

Introduction

Visualisation

Descriptive statistics

Examples of graphs I

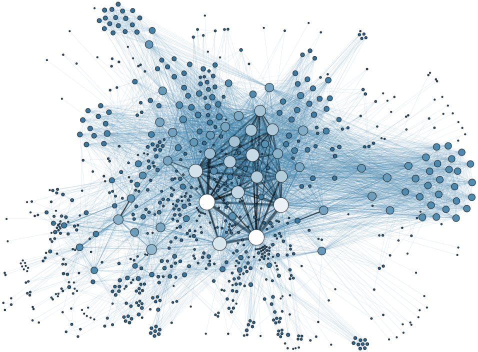


Figure: A social network.

Examples of graphs II

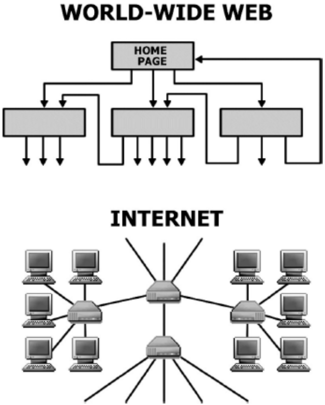


Figure: Internet and WWW. Source: [1].

Examples of graphs III

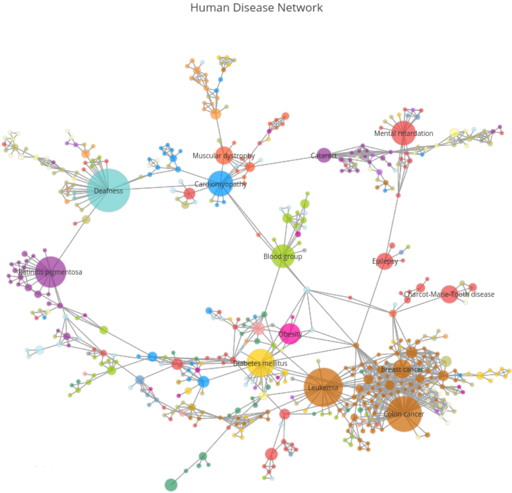


Figure: Gene regulatory network of human diseases.

Examples of graphs IV

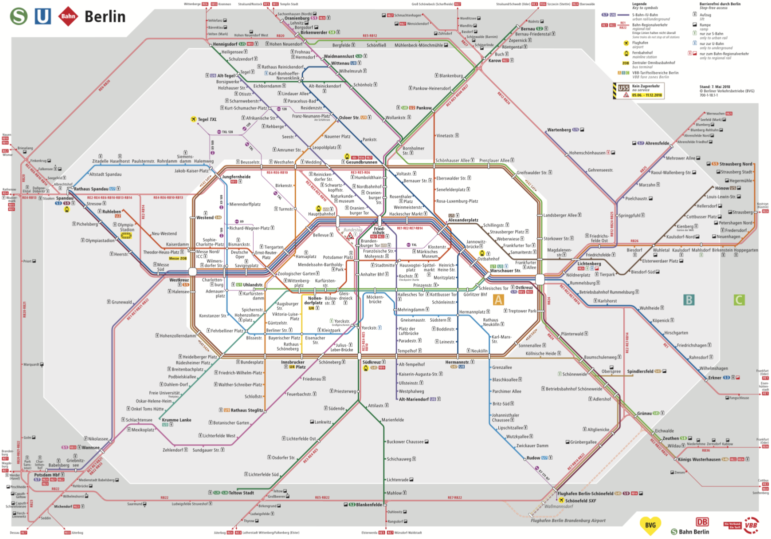


Figure: Subway network in Berlin.

Examples of graphs (foll.) I

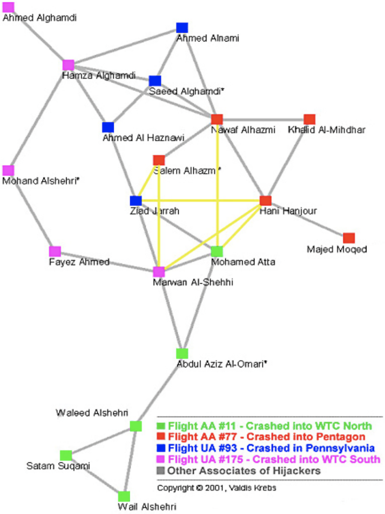


Figure: Terrorist network 09/11. Source: [5].

Examples of graphs (foll.) II

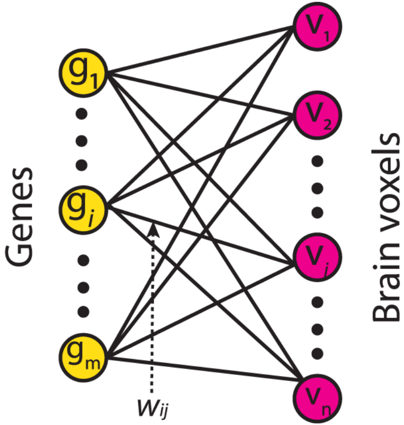


Figure: Bipartite networks of genes and brain voxels. Source: [3].

Examples of graphs (foll.) III

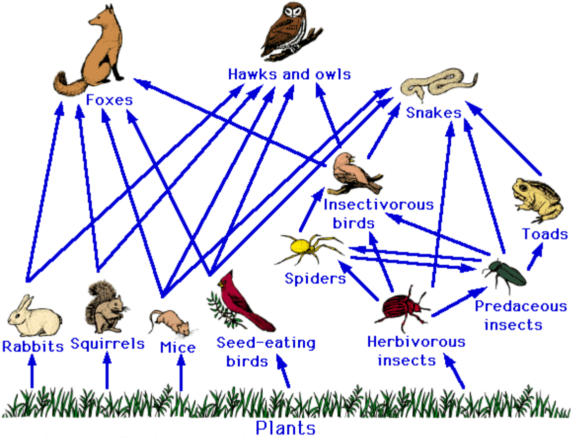


Figure: Simplified trophic network (food web). A directed link indicates who is the prey of whom.

Outline Part 1

Introduction

Visualisation

Descriptive statistics

Different visualisations of the same graph I

Warning: Visualisation can be misleading!

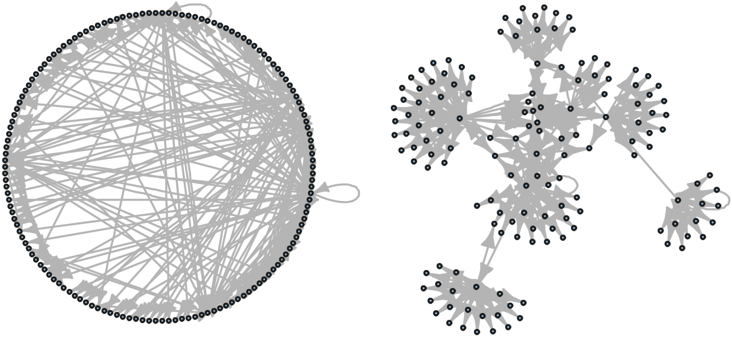


Figure: 2 representations of the same blogs network [4].

Different visualisations of the same graph II

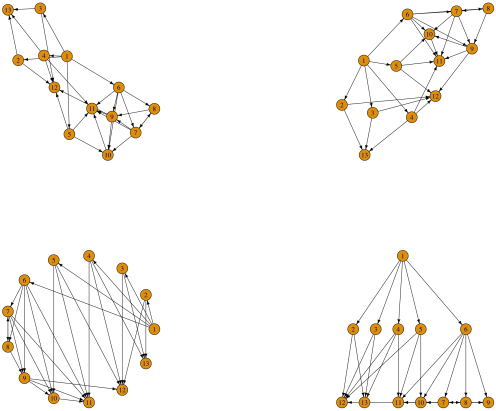


Figure: Different visualisations of the food web from Figure 7.

Different visualisations of the same graph III

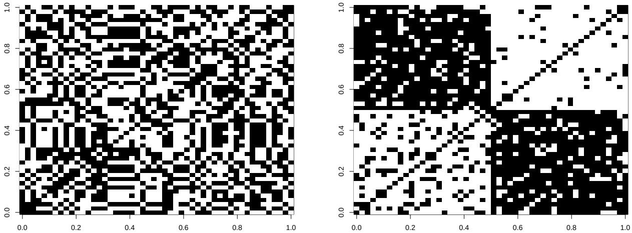
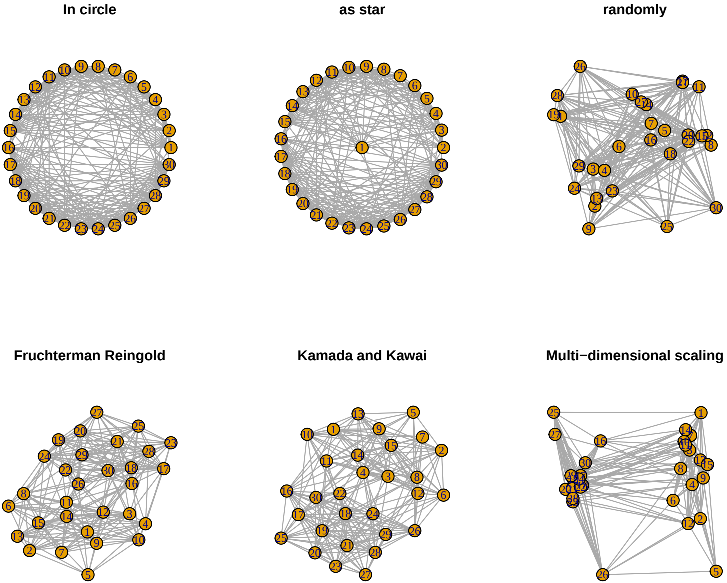


Figure: Dotplot representation of a graph: random node numbering (left) and specific permutation of the nodes (right)

Examples of representations



In circle

as star

randomly

Fruchterman Reingold

Kamada and Kawai

Multi−dimensional scaling

Outline Part 1

Introduction

Visualisation

Descriptive statistics

Vocabulary

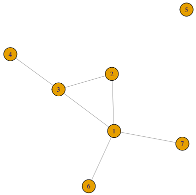
Basic definitions

- A graph $G = (V, E)$ is a set of nodes (or vertices) $V = \{1, \ldots, n\}$ and a set of edges (or links) $E \subset V^2$
- n is the order; $|E|$ is the size
- graphs can be undirected ($\{i, j\} \in E$) or directed ($(i, j) \in E$); binary (edge $\{i, j\}$ is present or absent) or weighted (present edge $\{i, j\}$ has a value w_{ij}; when $w_{ij} \in \mathbb{N}$ this is a multiplicity); with or without self-loops ($\{i, i\}$ is a self-loop);
- a node is isolated if it doesn't belong to any edge;
- a bipartite graph is s.t. $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$ and edges $e = \{u, v\} \in E$ are such that $u \in V_1, v \in V_2$ (e.g. bipartite network of genes and brain voxels)

Data structures

- Adjacency matrix $A = (A_{ij})_{i,j \in V}$ where $A_{ij} = 1\{\{i, j\} \in E\}$ (or $A_{ij} = w_{ij}$)
  - Undirected graphs have symmetric adjacency matrices
  - when graphs are sparse (ie not too many edges), this representation as a matrix is not efficient (n^2 size);
- List of interactions: this encoding is the most efficient.
  - NB: if the list of nodes is not additionally given, there cannot be isolated nodes;

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



1,2
1,3
1,6
1,7
2,3
3,4

Density

A simple binary graph has at most $\binom{n}{2}$ edges.
Its density is:

$$\mathrm{den}(G) = \frac{|E|}{\binom{n}{2}} = \frac{|E|}{n(n-1)/2}.$$

- the complete graph K_n is the undirected graph with n nodes that contains all possible $\binom{n}{2}$ edges; it has density 1.
- a clique is a complete subgraph in a graph

Neighbors and degrees I

- Neighbors of node $i \in V$ are $\mathcal{V}_i = \{j \in V, j \neq I, \{i,j\} \in E\}$: nodes connected to i in the graph
- Degree of node i is the number of its neighbours $d_i = |\mathcal{V}_i| = \sum_{j \neq i} A_{ij} = \sum_{j \neq i} A_{ji}$
- In directed graphs, one may define indegrees and outdegrees: $d_i^{out} = \sum_{j \neq i} A_{ij}$ and $d_i^{in} = \sum_{j \neq i} A_{ji}$
- Degrees are obtained as rowSums or colSums of adjacency matrix
- we have $\sum_{i=1}^{n} d_i = 2|E|$
- Mean degree $\bar{d} = n^{-1} \sum_{i=1}^{n} d_i$
- a d-regular graph has constant degree d (ex infinite grid)
- Hubs (informal) a hub is a large degree node in a graph

Neighbors and degrees II

Degree distributions only loosely characterize graphs

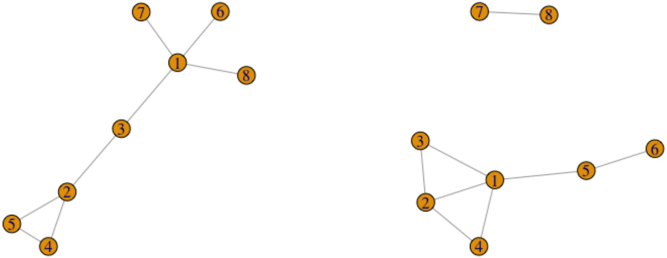
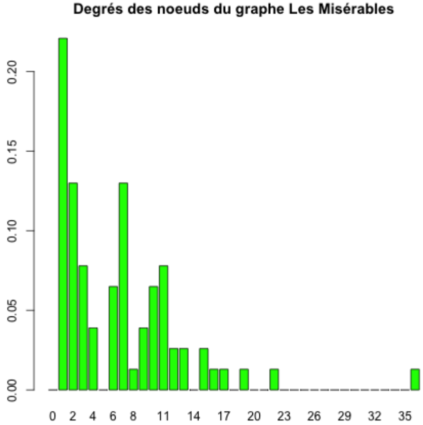


Figure: Example of 2 graphs with same degree sequence.

Neighbors and degrees III

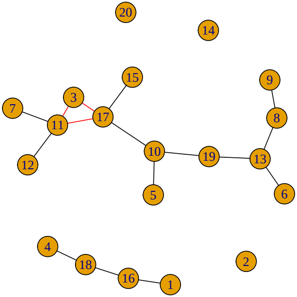
Graphs often show degree distributions with heavy tails, such as scale-free distributions



Paths, connectivity, diameter I

Paths

- a path between nodes $i, j \in V$ is a sequence of edges $e_1, \ldots, e_k \in E$ such that e_t and e_{t+1} share a node, $i \in e_1$ and $j \in e_k$;
- Its length is k;
- a cycle is a path that connects a node to itself;



Paths, connectivity, diameter II

Connectivity

- a connected component (cc) is a nodes subset $C = \{v_1, \ldots, v_k\} \in V$ such that for any $v_i, v_j \in C$ there exist a path that connects them;
- C is a maximal cc (mcc) if $C = V$ or if for any $v \in V \setminus C$, the subset $C \cup \{v\}$ is not a cc;
- an isolated node forms a mcc;
- any graph may be decomposed into a unique collection of mcc;
- there are at most $n - |E|$ such cc;
- the graph is connected when it has a unique mcc;
- Giant component (informal): In a sequence of graphs G_n each with n nodes, let C_n be the largest mcc in G_n. We say that C_n is a giant component if its relative size $|C_n|/n$ does not tend to 0 as n increases;

Paths, connectivity, diameter III

Diameter

- the distance ℓ_{ij} between 2 nodes $i, j \in V$ is the length of the shortest path between i, j (and $+\infty$ if the nodes are not in the same cc)
- the mean distance in the graph is $\bar{\ell} = 1/(n(n-1)) \sum_{i,j} \ell_{ij}$
- diameter $\mathrm{diam}(G) = \max\{\ell_{ij}; i, j \in V\}$;
- It's finite only if the graph is connected;
- Small-world property (informal): a graph has the small-world property whenever $\bar{\ell}$ is of the order of $\log(n)$;
- see the small-world experiment by Stanley Milgram; and its modern version: three and a half degrees of separation [2]

Clustering coefficients, transitivity, centrality I

Friends of my friends are my friends …

- Let H_i be the subgraph induced by the neighbors of node $i \in V$, $i.e. H_i = (\mathcal{V}_i, E_i)$ where $\mathcal{V}_i$ is the set of neighbors and E_i set of edges $\{j, k\} \in E$ st $j, k \in \mathcal{V}_i$.
- Clustering coefficient C_i of node i is

$$C_i = \begin{cases} \frac{2|E_i|}{d_i(d_i-1)} & \text{if } d_i \geq 2, \\ 0 & \text{otherwise} \end{cases}$$

- It is the local density of the subgraph H_i and thus $C_i \in [0, 1]$
- the mean clustering coefficient is $\bar{C} = \frac{1}{|V|} \sum_{i \in V} C_i$
- transitivity

$$T = \frac{\sharp \text{ triangles}}{\sharp \text{ triplets of connected nodes}}$$

Clustering coefficients, transitivity, centrality II

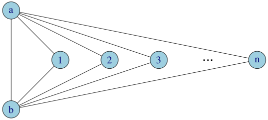


Figure: Here $C_i = 1$ for all nodes except a, b and thus $\bar{C}$ tends to 1. However T tends to 0.

Clustering coefficients, transitivity, centrality III

Centrality

- Degree centrality $C_D(i) = d_i$
- Closeness centrality $C_P(i) = \left(\sum_{j \in V} \ell_{ij}\right)^{-1}$, where ℓ_{ij} is the distance between i, j
- Betweenness centrality $C_B(i) = \sum_{j,k:j \neq k \neq i} \frac{g_{jk}(i)}{g_{jk}}$, where g_{jk} is the number of shortest paths from j to k, and $g_{jk}(i)$ is the number of shortest paths from j to k that go through i;

Motifs I

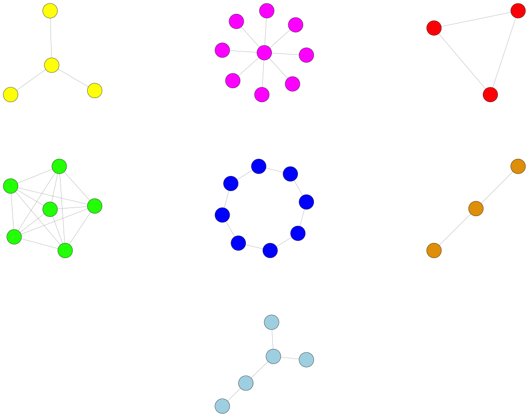


Figure: Examples of motifs: stars (k-stars with $k = 3$ and $k = 8$), cliques (K_3 or triangle and K_6), cycle of length 8, . . .

Motifs II

- Counting frequencies of small sizes motifs may be a way to characterize the topology of the graph;
- When the size of the motif becomes large, enumerating all occurrences of a motif becomes a computationally difficult problem;
- with a null model, one can test the hypothesis that the observed frequencies of a motif are too large or too small wrt to some expected value;

References for part 1 - I

[1] Albert, R. and A.-L. Barabási.
Statistical mechanics of complex networks.
Rev. Mod. Phys. 74, 47–97, 2002.

[2] Bhagat, S., M. Burke, C. Diuk, I. O. Filiz, and S. Edunov (2016).
Three and a half degrees of separation.
facebook research blog `https://research.fb.com/three-and-a-half-degrees-of-separation/`.

[3] Ji, S., W. Zhang, and R. Li
A probabilistic latent semantic analysis model for coclustering the mouse brain atlas.
IEEE/ACM Transactions on Computational Biology and Bioinformatics 10(6), 1460–1468, 2014.

References for part 1 - II

[4] Kolaczyk, E. D. and G. Csárdi (2014).
Statistical analysis of network data with R.
Use R! Springer, New York.

[5] V. Krebs.
Unloaking terrorist networks.
Connections, 24(3), 2001.

Part II

Some random graph models

Outline Part 2

The Erdős-Rényi model

Modeling the degree distribution

Preferential attachment

Exponential random graph model

Latent position model

Stochastic Block Model

Random graph models

A random graph model is simply a (finite or countable) collection $\mathcal{G}$ of graphs together with a probability distribution $\mathbb{P}$ on this collection.

Erdős-Rényi model I

- Introduced at the end of the 50's, it's the simplest random graph model
- 2 variants, called $\mathcal{G}(n, M)$ and $\mathcal{G}(n, p)$
- $\mathcal{G}(n, M)$ is the collection of all simples graphs (binary, undirected, with no self-loops nor multiple edges) with n nodes and M edges, together with the uniform distribution $\mathbb{P}$ on that collection.
- $\mathcal{G}(n, M)$ contains $\binom{N}{M}$ different graphs, where $N = n(n-1)/2$ and the occurrence probability of each of these graphs is $1/\binom{N}{M}$.
- $\mathcal{G}(n, p)$ is the collection of all simple graphs generated with 2 parameters: n is the number of nodes and $p \in (0, 1)$ the probability of connection of any 2 nodes,
- each graph in $\mathcal{G}(n, p)$ is such that its adjacency matrix $A = (A_{ij})$ contains iid random variables $\sim \mathcal{B}(p)$ (for $1 \leq i < j \leq n$).

Erdős-Rényi model II

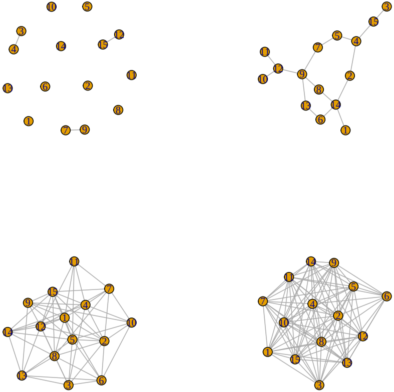


Figure: Erdős-Rényi graphs $\mathcal{G}(n, p)$ with $n = 15$ and $p \in \{0.05, 0.2, 0.6, 0.9\}$.

Erdős-Rényi model III

Exercise

Let n and p be fixed. Consider a graph generated under the model $\mathcal{G}(n,p)$:

1. *Give the average number of edges in this graph.*
2. *Let $M \geq 0$. What is the probability that the graph is of size M?*
3. *Give the law of the random variable D_i which denotes the degree of the node i in the model $\mathcal{G}(n,p)$. Deduce the expectation of the degree D_i.*
4. *Study the convergence of $\bar{D}_n/(n-1)$ when $n \to \infty$, where $\bar{D}_n$ denotes the average degree of the graph under $\mathcal{G}(n,p)$.*
5. *Give an approximation of the law of D_i when n is large.*

Erdős-Rényi model IV

Simulation of large graphs

- In principle, it is enough to generate $n(n-1)/2$ rv with distribution $\mathcal{B}(p)$;
- When n is large and $p = p_n$ is the order of $1/n$, this procedure is very inefficient (the expected degree of a node is finite and therefore most variables are 0).
- A more efficient procedure is to simulate the number M of edges present in the graph according to a binomial law $Bin(n(n-1)/2, p)$, and then draw the positions of the edges, i.e. draw without replacement M positions among the $n(n-1)/2$ possible ones;
- the computational complexity is then $O(n + |E|)$ instead of $O(n^2)$ (See [4], section 6.2.3 for more details).

Erdős-Rényi model V

Properties in $G(n, p)$

- The degree D_i of the node i satisfies $D_i \sim Bin(n-1, p)$
- (LLN): $\bar{D}/(n-1)$ converges to $\mathbb{E}(A_{ij}) = p$
- In particular, $\mathbb{E}(D_i) = (n-1)p = pn(1+o(1))$ (when n large and p small)
- When $n \to +\infty$ and $p \to 0$ with $np \to \lambda > 0$ then the law $Bin(n-1, p)$ is approximated by a Poisson law $\mathcal{P}(\lambda)$.
- Binomial and Poisson distribution are light-tailed distributions; thus Erdős-Rényi model does not fit well real networks

Outline Part 2

The Erdős-Rényi model

Modeling the degree distribution

Preferential attachment

Exponential random graph model

Latent position model

Stochastic Block Model

Scale-free (or power law) distribution

Many real world networks are such that their degree distribution is a power-law:

$$f_{D_i}(k) := \mathbb{P}(D_i = k) = \frac{c}{k^\gamma},$$

where c is a normalising constant and $\gamma > 0$ is the exponent of the power law.

Configuration models I

We can define random graph models using only the distribution of the degrees D_i of the nodes:

1. Power law of degrees: We consider random graphs with n nodes such that $D_1, \ldots, D_n$ are i.i.d according to a power law
2. Model with fixed degrees: Let $\underline{d} = (d_1, \ldots, d_n)$ be a (possible) sequence of degrees of nodes and $FD(\underline{d})$ the collection of all the graphs on n nodes which have exactly the sequence of degrees $\underline{d}$, provided with uniform probability.
3. Model with variable degrees: Let $\underline{d} = (d_1, \ldots, d_n)$ be a (possible) sequence of degrees of nodes and $RD(\underline{d})$ the random graph model on n nodes such that all edges A_{ij} are independent, with law $\mathcal{B}(p_{ij})$ with $p_{ij} = d_i d_j / C$ where C positive constant such that $0 \leq p_{ij} \leq 1$ (for example $C = \max_{i \neq j} d_i d_j$).

Configuration models II

Rems on the sequence of degrees

- In the $FD(\underline{d})$ model, all graphs have exactly the sequence of degrees $\underline{d}$.
- In the power law model, we start by drawing a sequence $\underline{d}$ of degrees according to this power law, then we consider a graph which has this sequence of fixed degrees.
- In the $RD(\underline{d})$ model, the degrees are only approximately equal to $\underline{d}$. Indeed in that case

$$\mathbb{E}(D_i) = \sum_{j \neq i} \mathbb{E}(A_{ij}) = \sum_{j \neq i} p_{ij} = \frac{d_i}{C} \sum_{j \neq i} d_j = \frac{d_i(2|E| - d_i)}{C}.$$

By taking d_i not too large and $C \simeq 2|E|$ we obtain $\mathbb{E}(D_i) \simeq d_i$.

Simulation

- The power law model of degrees is not constructive nor simply generative: if we draw a sequence of D_i as indicated, it is unlikely that the realization satisfies the conditions of the Erdős-Gallai theorem and therefore is achievable as a sequence of degrees of a graph.
- The simulation of graphs in the random degree RD($\underline{d}$) model is direct since it suffices to draw the A_{ij} from independent (not identically distributed) Bernoulli distributions.
- To generate graphs in $FD(\underline{d})$, we use either a matching algorithm; either a rewiring or switching algorithm.

Matching algorithm

- Input: $\underline{d} = (d_1, \ldots, d_n)$; Output: list of edges.
- Initialization: Edge.List ← (); Node.List ←()
- Create fake Node.List: For $i \in \{1, \ldots, n\}$,
    - While $d_i \geq 1$,
        - Node.list ← concatenate(Node.List,i)
        - $d_i \leftarrow d_i - 1$
- Create Edge.List: While (Node.List is not empty)
    - Draw i, j uniformly and without replacement in Node.List
    - Edge.List ← concatenate(Edge.List, $\{i, j\}$)
- Simple graph test: If Edge.List contains loops or multiple edges, the output is invalid. Let's start again.

Rewiring algorithm

- Input: Edge.List; Nb.iter
- Output: Edge.List
- While(Nb.iter $\geq$ 1)
    - Choose $e_1 = \{u_1, v_1\}$ and $e_2 = \{u_2, v_2\}$ uniformly in Edge.List
    - Propose the creation of $e'_1 = \{u_1, v_2\}, e'_2 = \{u_2, v_1\}$
    - If no loop or multiple edge created: replace e_1, e_2 with e'_1, e'_2
    - Nb.iter $\leftarrow$ Nb.iter -1

Matching versus rewiring

- matching algorithm does not necessarily create a simple graph (before the final test, because possibility of loops and multiple edges)
- If the produced graph is not simple, it must be discarded and we draw a new one. Very inefficient algorithm!
- a naive correction of this algorithm, which verifies that $i \neq j$ or that the edge $\{i, j\}$ does not yet exist may either not converge, or give biased sampling from all the possible graphs
- rewiring algorithm is more efficient but it works only from an already existing graph which has the sequence of degrees that we have set for ourselves.
- In addition, it requires to set the number of iterations. Empirically, around 100 times the number of edges in the graph.

Application to testing

The behavior of the model $FD(\underline{d})$ can be studied using (expensive) numerical simulations.

- In order to test significance of a statistic T^{obs} measured on the observed graph, one can define the null model H_0 as the hypothesis $FD(\underline{d})$, where $\underline{d}$ is the observed degree sequence
- By simulating a large number of graphs under H_0 (using the rewiring algorithm), one obtains a sequence of values for the random variable T and thus its empirical distribution under H_0
- Comparing T^{obs} with the quantiles of that distribution, one can test wether T is too large or too small wrt to $FD(\underline{d})$.

Outline Part 2

The Erdős-Rényi model

Modeling the degree distribution

Preferential attachment

Exponential random graph model

Latent position model

Stochastic Block Model

Preferential attachment I

Rich get richer

Principle

- Start with a small initial graph $G_0 = (V_0, E_0)$ with degrees sequence $(d_{1,0}, \ldots, d_{|V_0|,0})$;
- construct an increasing (in the number of nodes) sequence of graphs $G_t = (V_t, E_t)$.

Preferential attachment II

Iterations

At each time step $t \geq 1$,

- a new node i_t with degree $m \geq 1$ is added $V_t = V_{t-1} \cup \{i_t\} = V_0 \cup \{i_1, \dots, i_t\}$.
- This new node connects with m existing nodes chosen with probability $d_{j,t-1}/(2|E_{t-1}|)$ where $d_{j,t}$ is the degree of node j at time t and $2|E_t|$ is the total number of edges at time t (preferential attachment to large degree nodes),
- Update degrees $d_{j,t}$ for all nodes $j \in V_t$.

At final timestep T, the graph has $|V_0| + T$ nodes and $|E_0| + Tm$ edges.

Preferential attachment III

(Dis)-advantages

- ☺ This is a dynamic generative model that can be used to simulate data
- ☺ (Under certain conditions) the degree distribution of the graph follows a power law.
- ☹ Problem with the choice of parameters G_0, m, T_{final}. Impact of this choice on the graph obtained?
- ☹ From a statistical point of view, this is not a model that can be fitted to the data (with one exception, see [2])

Outline Part 2

- The Erdős-Rényi model
- Modeling the degree distribution
- Preferential attachment
- Exponential random graph model
- Latent position model
- Stochastic Block Model

Exponential random graph model (ERGM) I

- $n \geq 1$ an integer
- $\mathcal{A}_n$ the set of binary adjacency matrices (symmetric or not) with size $n \times n$
- for any $A \in \mathcal{A}_n$, let $S(A) \in \mathbb{R}^p$ denote a vector of statistics on the graph

ERGM(S) is defined by the probability $\{\mathbb{P}_\theta\}_{\theta \in \mathbb{R}^p}$ over $\mathcal{A}_n$:

$$\forall \theta \in \mathbb{R}^p, \forall A \in \mathcal{A}_n, \quad \mathbb{P}_\theta(A) = \frac{1}{c(\theta)} \exp\left(\theta^\top S(A)\right),$$

where $c(\theta) = \sum_{A \in \mathcal{A}_n} \exp(\theta^\top S(A))$ is a normalizing constant.

Exponential random graph model (ERGM) II

Comments

- $S(A)$ automatically becomes a vector of exhaustive statistics of the model: All graphs having the same observed value of S have the same probability of occurrence under ERGM(S).
- In practice, $S(A)$ can contain the number of edges, triangles, k-stars, … or even covariates of the model.

Examples I

the simplest

- If $S_0(A) = vec((A_{ij})_{1\leq i<j\leq n})$ then ERGM(S_0) is such that
$$\mathbb{P}_\theta(A) \propto \exp(\textstyle\sum_{i<j} \theta_{ij}A_{ij}),$$
where $\propto$ means 'proportional to'.
- the rv A_{ij} are independent and non identically distributed with $A_{i,j} \sim \mathcal{B}(p_{ij})$ and $p_{ij} = \exp(\theta_{ij})/(1+\exp(\theta_{ij}))$.
- the model has as many parameters as observations!

Examples II

a particular case

When we moreover impose the constraint $\theta_{ij} = \theta$ for all i, j, then we obtain Erdős-Rényi model

$$\mathbb{P}_\theta(A) \propto \exp(\theta S_1(A)),$$

where $S_1(A) = \sum_{i,j} A_{ij}$ number of edges and $\hat{p} = S_1(A)/[n(n-1)/2]$ (this is the MLE).

Examples III

a less elementary example

- If $S(A) = (S_1(A), S_2(A))$ with S_1 as above and $S_2(A) = \sum_{i,j,k} A_{ij}A_{ik}$ then the rv $(A_{ij})_{i<j}$ are non independent and there is no analytical expression for a MLE
- For $k \geq 1$ let $S_k(A)$ be the number of k-stars and $T(A) = \sum_{ijk} A_{ij}A_{ik}A_{jk}$ the number of triangles. Markov random graphs rely on $S = (S_1, \ldots, S_{n-1}, T)$.
- In practice, $k = n - 1$ is much too large and we use $k << n - 1$ for most ERGMs.

Problems with ERGM

- The constant $c(\theta)$ may not be computed. Parameter estimation methods based on MCMC with for instance Gibbs sampling can get rid of that issue;
- Maximisation of the likelihood in ERGMs is a very difficult pbm, and it is ill-posed. These models often degenerate in the sense that either they concentrate the mass of the distribution on the complete graph K_n, or on the empty graph, or on a mixture of these 2 extremes [1, 5]

I do not recommend using ERGMs.

Outline Part 2

The Erdős-Rényi model

Modeling the degree distribution

Preferential attachment

Exponential random graph model

Latent position model

Stochastic Block Model

Latent position model I

Principe

- Suggested to study social networks in [3];
- Latent variables $\{Z_i\}_{1 \leq i \leq n}$ (ie unobserved) associated with each node live in the space $\mathbb{R}^2$ which represents a *social space*
- The proximity of individuals in this space induces a greater probability of connection in the graph. Thus, only the relative position of the latent variables between them is important for the model (and not their absolute position).

Latent position model II

Definition

- Consider an undirected binary graph $(A_{ij})_{1\leq i,j\leq n}$
- (possibly) covariate vectors $\mathbf{x}_{ij} \in \mathbb{R}^s$ on each relation (i, j).
- We use a logistic regression model

$$\begin{aligned}\operatorname{logit}(\mathbb{P}(A_{ij} = 1|Z_i, Z_j, \mathbf{x}_{ij})) &= \frac{\mathbb{P}(A_{ij} = 1|Z_i, Z_j, \mathbf{x}_{ij})}{1 - \mathbb{P}(A_{ij} = 1|Z_i, Z_j, \mathbf{x}_{ij})} \\ &= \alpha + \beta^\intercal \mathbf{x}_{ij} - \|Z_i - Z_j\|,\end{aligned}$$

where $\|\cdot\|$ is the Euclidean norm in the latent space $\mathbb{R}^2$.

Latent position model III

Remarks

- The model parameters are $(\alpha, \beta) \in \mathbb{R} \times \mathbb{R}^s$. The α parameter adjusts the density of the graph.
- We can replace the Euclidean norm with any distance.
- The variables $\{Z_i\}_i$ can only be reconstructed up to rotation, axial symmetry and translation.

The R package `latentnet` proposes a Bayesian inference procedure in this model.

Comments and extensions of the latent position model

Comments

- Originally, the model is proposed with $Z_i \in \mathbb{R}^q$ but there is no statistical way to choose the dimension q
- the software `latentnet` deals with $q = 2, 3$ but there is no reason to believe that this is a "good" choice. It's only that then you can solve the problem.

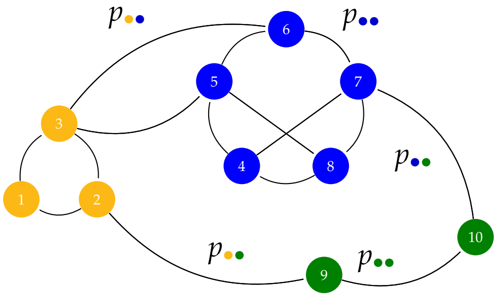
Extensions

- You can mix the latent position model with a clustering approach: assume that the latent positions Z_i follow a mixture of multivariate Gaussian variables, then you will obtain a clustering of the nodes in your graph;
- There is also a directed version of the model where you replace the distance $\|Z_i - Z_j\|$ by a normalized scalar product $Z_i^\intercal Z_j/\|Z_i\|$

Outline Part 2

- The Erdős-Rényi model
- Modeling the degree distribution
- Preferential attachment
- Exponential random graph model
- Latent position model
- Stochastic Block Model

Stochastic block model I



$n = 10, Z_{5\bullet} = 1$

$A_{12} = 1, A_{15} = 0$

- K groups (=colors ●●●).
- $\{Z_i\}_{1 \leq i \leq n}$ i.i.d. vectors $Z_i = (Z_{i1}, \ldots, Z_{iK}) \sim \mathcal{M}(1, \boldsymbol{\pi})$, with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ groups proportions. Z_i not observed (latent).
- Observations: presence/absence of an edge $\{A_{ij}\}_{1 \leq i < j \leq n}$,
- Conditional on $\{Z_i\}$'s, the r.v. A_{ij} are independent $\mathcal{B}(p_{Z_i Z_j})$.

Stochastic block model II

Comments

- SBM represents a compromise between 2 extremes cases:
  - in Erdős-Rényi graphs, the rv A_{ij} where iid $\mathcal{B}(p)$ with the same value p for every edge; this was too homogeneous;
  - in RD($\underline{d}$), the rv A_{ij} where independent with $\mathcal{B}(p_{ij})$, each one having its own value p_{ij} (no estimation possible!)
- Besides, the rv A_{ij} are not independent here (only conditionally independent), which is more interesting
- SBM automatically gives you a clustering of the nodes: groups of nodes that have the same probability of connection to the other groups.
- For all its advantages, it's a widely used model.

References for part 2 - I

[1] Chatterjee, Sourav and Diaconis, Persi (2013).
Estimating and understanding exponential random graph models.
Ann. Statist. 41(5), 2428–2461.

[2] Gao, F and van der Vaart, A (2022).
Statistical Inference in Parametric Preferential Attachment Trees.
ArXiV preprint https://arxiv.org/abs/2111.00832.

[3] Hoff, P., A. Raftery, and M. Handcock (2002).
Latent space approaches to social network analysis.
J. Amer. Statist. Assoc. 97(460), 1090–98.

[4] Kolaczyk, E. D. (2009).
Statistical Analysis of Network Data: Methods and Models.
Springer.

References for part 2 - II

[5] Schweinberger, Michael and Handcock, Mark S. (2015).
Local dependence in random graph models: characterization, properties and statistical inference.
JRSSB 77(3), 647–676.

Part III

Community detection

Outline Part 3

Introduction

Similarity graphs

Laplacian matrices of graphs

Spectral clustering algorithms

Practical comments

Introduction I

Partition the nodes of the graph into a finite number of groups.

Interest

- Summarize the information of a graph;
- Find classes of homogeneous nodes, i.e. individuals with similar behavior.

Introduction II

Group types

- Several existing clustering methods;
- In this part we limit ourselves to the detection of communities: i.e. groups of nodes that are strongly connected to each other and poorly connected to individuals in other groups.
- More general groups can be considered with other techniques (see SBM).
- In this course, we also limit ourselves to partitions: the groups have an empty intersection (and cover all nodes). There exist methods that do *overlapping clustering*.

Introduction III

Community detection methods

- Techniques based on modularity;
- Random walk methods (type `InfoMap`);
- Probabilistic methods (ex SBM), often more general than communities;
- Spectral clustering (detailed here);
- ...

Modularity based methods I

Principle

- We give ourselves a measure of the quality of a partition of the nodes of the network into groups;
- We seek to optimize this quality measure;
- As the number of possible partitions is too large to be explored exhaustively, heuristics are necessary to find the optimum.

Modularity based methods II

Modularities

The most popular is the Newman & Girvan [2] modularity. For a fixed partition C into K classes, we set

$$Q(C) = \frac{1}{2|E|} \sum_{i,j;C(i)=C(j)} \left(A_{ij} - \frac{d_i d_j}{2|E|} \right)$$

where $|E|$ is the total number of edges in the graph, $C(i)$ is the class of node i in the partition C and d_i is the degree of node i. Equivalently

$$Q(C) = \sum_{k=1}^{K} \left(\frac{\sum_{i,j;C(i)=C(j)=k} A_{ij}}{2|E|} \right) - \left(\frac{\sum_{i;C(i)=k} d_i}{2|E|} \right)^2 .$$

Modularity based methods III

Implementations

- Louvain algorithm [1] is the most famous one;
- The R package `igraph` contains a function `cluster_louvain`
- see also variants [5] and `https://github.com/vtraag/louvain-igraph`

Disadvantages

- Very unstable: partitions vary a lot from one try to another one
- No guarantee of reaching the optimum.

Methods based on random walks

Principle

- We consider a walker which starts from a node (taken at random) and which visits the graph by drawing uniformly at random a neighbour of its current position.
- If the graph is organized into communities, the walker will spend a lot of time in one of them (he remains stuck in the community). By repeating the process, we can thus determine subgroups of nodes strongly connected.

Spectral clustering I

A very good (but quite old) tutorial on spectral clustering is [6].

Beyond graphs

- Spectral clustering is used more largely than just for graph clustering
- Usable on a classic data table, for example as an alternative to the *k*-means algorithm, by constructing a data similarity graph.

Spectral clustering II

Characteristics of spectral clustering

- Classification adapted to the search of communities (almost exclusively);
- Which is not based on a probabilistic model;
- But which has the advantage of working on very large graphs.
- Technique limited to undirected graphs (edges = similarities or distances, therefore symmetrical). Binary or valued.

Outline Part 3

Introduction

Similarity graphs

Laplacian matrices of graphs

Spectral clustering algorithms

Practical comments

Spectral clustering on a dataframe I

Data structure

- Dataframe of size $n \times p$, i.e. n observations $x_1, \ldots, x_n$ with $x_i \in \mathbb{R}^p$ has dimension p.
- We will do (unsupervised) classification of this set of n points.
- Usual techniques: k-means or hierarchical classification. Based on similarity $s_{ij} \geq 0$ (inversely proportional to distance) between pairs of observations x_i, x_j.

Spectral clustering on a dataframe II

Principle for the construction of a similarity graph

- We construct $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ set of nodes of the graph and $e = \{v_i, v_j\}$ is an edge of the graph if the similarity s_{ij} between x_i, x_j is greater than a certain threshold.
- For a binary graph: $s_{ij} \geq s \implies \{v_i, v_j\} \in E$ and $s_{ij} < s \implies \{v_i, v_j\} \notin E$;
- For a valued graph: $s_{ij} \geq s \implies \{v_i, v_j\} \in E$ and the edge carries the value s_{ij}, otherwise the edge is not present.
- In practice, several constructions are possible (see below).

Spectral clustering on a dataframe III

Link between data clustering and graph communities

The problem of clustering points $x_1, \ldots, x_n$ can be reformulated as a problem of partitioning the similarity graph where we look for groups of nodes such that within-group connections are large (i.e. the corresponding vectors x_i in the same group are very similar to each other) and such that the connections between-groups are small (i.e. little similarity between the vectors x_i which correspond to nodes in different groups).

Different similarity graphs I

Dense (or complete) similarity graph

- The neighborhoods in $\mathbb{R}^p$ (ie the distance between points) define the similarity.
  Ex: $\forall i \neq j$ we set $s_{ij} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$ for a certain $\sigma^2 > 0$ which controls the size of neighborhoods in $\mathbb{R}^p$ and $s_{ii} = 0$.
- Valued graph, each edge (i, j) being weighted by $s_{ij} > 0$.
- As the similarities are strictly positive, we obtain a dense (complete) valued graph (all edges are present).

ϵ-neighborhood graph

We set a threshold $\epsilon > 0$ and we connect all the nodes v_i, v_j such that $s_{ij} \geq \epsilon$ (i.e. distance between the vectors x_i, x_j below a threshold). The graph thus constructed is binary.

Different similarity graphs II

k-nearest neighbor graph

- We start by defining a directed graph $\tilde{G} = (V, \tilde{E})$. If x_j is one of the k nearest neighbors of x_i (i.e. d_{ij} is among the k smallest elements of $\{d_{il}; l \neq i\}$ or s_{ij} is among the k largest elements of $\{s_{il}; l \neq I\}$), then we create an (oriented) edge from v_i to v_j, i.e. $(v_i, v_j) \in \tilde{E}$.
- From this directed graph $\tilde{G}$, we can define undirected $G = (V, E)$ in two different ways:
  - Let $\{v_i, v_j\} \in E$ as soon as $(v_i, v_j) \in \tilde{E}$ or $(v_j, v_i) \in \tilde{E}$ (graph of the nearest k neighbors) ;
  - Let $\{v_i, v_j\} \in E$ as soon as $(v_i, v_j) \in \tilde{E}$ and $(v_j, v_i) \in \tilde{E}$ (graph of the nearest k mutual neighbors).
- The edges are then given their weight s_{ij} to form a valued graph.

Different similarity graphs III

Comments

- The graph of k-nearest neighbors is a sort of compromise between the dense graph and the graph of ϵ-neighborhood: both have a thresholding step which reduces the noise, but in the former, we keep the values of the largest similarities s_{ij} (unlike ϵ-neighborhood and like dense graph).
- The choice of the similarity graph between the vectors x_i influences the result of the partitioning that we get on points. But we don't know which choice is better a priori.
- In the following, we have a graph (binary or valued), which is already constructed and which defines the relationships between our entities. We will apply spectral clustering on this graph.

Outline Part 3

Introduction

Similarity graphs

Laplacian matrices of graphs

Spectral clustering algorithms

Practical comments

Notation I

- G undirected valued graph, A its valued adjacency matrix (size $n \times n$) with positive entries $A_{ij} \geq 0$ (similarities).
- $D = diag(d_1, \ldots, d_n)$ diagonal matrix (of size $n \times n$) where d_i = valued degree of node i in G, i.e. $d_i = \sum_j A_{ij} = \sum_j A_{ji}$ (sum of the weights of the edges from i).
- There are several definitions of the Laplacian matrix of a graph. The interest of these matrices lies in the properties of their spectrum (=eigenvalues and eigenvectors).

Definitions of Laplacian matrices

Let G be a graph, we define

- a non-normalized Laplacian $L = D - A$;
- a normalized Laplacian $L_N = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$,
- an absolute Laplacian $L_{\text{abs}} = D^{-1/2} A D^{-1/2} = I - L_N$.

Notation II

Remarks

- Since D is a diagonal matrix, $D^{-1/2}$ and D^{-1} are the matrices whose diagonal elements are equal to $1/\sqrt{d_i}$ and $1/d_i$ respectively (this is not true if D is not diagonal!).
- Left (resp. right) multiplication by diagonal matrix = multiply row (resp. columns) vectors. Thus, $D^{-1/2}A$ is the matrix whose rows are the $d_i^{-1/2}A_{i\bullet}$ (it is a stochastic matrix) while $D^{-1/2}AD^{-1/2}$ is the matrix with entries i, j equal to $A_{ij}/\sqrt{d_i d_j}$.

why is this useful? I

Laplacian spectrum

- The spectra of these matrices are connected to the maximal connected components (mcc) of the graph G.
- L, L_N are real symmetric matrices which have n positive real eigenvalues (counted with multiplicity), denoted $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.
- In particular, 0 is always an eigenvalue of L, L_N and its multiplicity is equal to the number of mcc in the graph G.
- L_{abs} has the same eigenvectors as L_N but its eigenvalues are $1 - \lambda_n^{L_n} \leq \ldots \leq 1 - \lambda_2^{L_n} \leq 1 - \lambda_1^{L_n} = 1$.

why is this useful? II

From mcc to clusters

- If we denote $C_1, \ldots, C_k \subset \{v_1, \ldots, v_n\}$ these (unique) mcc and $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$ the indicator vectors of the mcc (defined by $\mathbf{1}_{C_l}(i) = 1$ if $v_i \in C_l$ and $\mathbf{1}_{C_l}(i) = 0$ otherwise), then the eigenspace associated with the eigenvalue 0 (for L, L_N) is generated by $\mathbf{1}_{C_1}, \ldots, \mathbf{1}_{C_k}$.
- In practice, we look at graphs with a single connected component (otherwise we study them separately). So 0 is an eigenvalue of multiplicity 1 and the associated eigenspace is generated by the vector $\mathbf{1}$: not very interesting. The study of the spectrum brings nothing more here!
- However, a community is almost a mcc. So we use spectral decomposition to find communities.

why is this useful? III

In practice

- Spectral clustering consists in focusing on the first k eigenvectors of the Laplacians (i.e. the k eigenvectors corresponding to the k smallest eigenvalues of L, L_N) to find k communities.
- Attention: in L, L_N it is the small eigenvalues which contain the interesting information whereas for L_{abs} we will see that it is the *large eigenvalues, in absolute value* !
- NB: These eigenvectors are different for the 3 Laplacians (L and L_N or L_{abs}) so the resulting clusterings will a priori also be different!

Outline Part 3

Introduction

Similarity graphs

Laplacian matrices of graphs

Spectral clustering algorithms

Practical comments

Algorithms I

Preliminaries

- There are several variants of algorithms (just as there are several Laplacians).
- We will only see 2: a normalized spectral clustering algorithm which uses L_N [3] and an absolute spectral clustering algorithm based on L_{abs} [4]
- In the following, A is the valued adjacency matrix of an undirected graph with positive entries.

Algorithms II

Normalized spectral clustering [3]

- Input: A of size $n \times n$, symmetric with positive entries; number k of clusters
- Output: Clusters $C_1, \dots, C_k$ that partition $\{1, \dots, n\}$
- Procedure:
  - Compute the normalized Laplacian matrix L_N
  - Compute the k eigenvectors $u_1, \dots, u_k$ associated with k smallest eigenvalues of L_N
  - Form the matrix U of size $n \times k$ whose **columns** are $u_1, \dots, u_k$
  - **Form the matrix T of size $n \times k$ by normalizing the rows of U to have a Euclidean norm 1** (i.e. $t_{ij} = u_{ij} / \sqrt{\sum_k u_{ik}^2}$)
  - Create clusters $C_1, \dots, C_k$ on the n **lines** of T by k-means algorithm with k clusters.

Algorithms III

Absolute spectral clustering [4]

- Input: A of size $n \times n$, symmetric with positive entries; number of clusters k
- Output: Clusters $C_1, \dots, C_k$ partitioning $\{1, \dots, n\}$
- Procedure:
  - Compute the absolute Laplacian L_{abs}
  - Compute the first k eigenvectors $u_1, \dots, u_k$ of L_{abs} associated to the k **largest eigenvalues in absolute value**
  - Form the matrix U with size $n \times k$ whose **columns** are $u_1, \dots, u_k$
  - Create clusters $C_1, \dots, C_k$ on the n **rows** of U by using k-means algorithm with k clusters.

Comments

- Principle of spectral clustering: transforming the obs. $x_i \in \mathbb{R}^p, 1 \leq i \leq n$ in a new set $y_i \in \mathbb{R}^k, 1 \leq i \leq n$ (=the rows of the matrix U), by using a similarity graph, its associated Laplacian and its first k eigenvectors. You can see this as an embedding
- Properties of Laplacian matrices imply that this new set $\{y_i\}_{1 \leq i \leq n}$ is easily clustered into k groups (By simple k-means).
- the k groups obtained tend to form communities = gps of nodes with a high within-group connection probability and low between-group connection probability
- Absolute spectral clustering finds not only communities but also bipartite or dis-assortative structures

Outline Part 3

Introduction

Similarity graphs

Laplacian matrices of graphs

Spectral clustering algorithms

Practical comments

Spectral clustering in practice I

About the similarity graph

- Choice of similarity (when starting with ordinary dataset) should depend on the data type.
- Difference between ϵ-neighbourhood and k-nearest neighbour graphs (simple or mutual) = local adaptation to the neighbourhood of the latter. Neighbourhood sizes are different depending on the regions of space (larger in sparse regions, smaller in denser regions).
- *mutual* k-nearest neighbours tends to connect between them points which are in regions of constant density (like the *simple* version) but it does not connect regions with (very) different densities which are close together. ⇒ compromise between ϵ-neighbourhood and simple k-nearest neighbours.

Spectral clustering in practice II

About the similarity graph (cont.)

- *k*-nearest neighbour graphs are easier to handle than Gaussian similarity graph (which is dense/complete). Preferable; but be careful about the loss of information! We can by ex. have more mcc in these graphs than desired number of clusters!

Spectral clustering in practice III

Choice of the parameters of the similarity function

Empirical recommandations:

- take k of the order of $\log(n)$ for the graph of simple k-nearest neighbours and something large (no explicit rule) for the mutual one. In any case, we must look at the number of mcc obtained, compare it to the number of desired clusters and adjust accordingly.
- take ϵ such that the resulting ϵ-neighbourhood graph is connected.
- there is no rule for choosing σ in the Gaussian similarity.

Spectral clustering in practice IV

Maximal connected components

- If the graph has p mcc, then the eigenspace associated with the eigenvalue 0 (for L, L_N) has dimension p and is generated by the cluster indicators.
- However, the output of a spectral decomposition algo is any orthogonal basis of eigenvectors of this space (i.e. not necessarily the basis of the indicator vectors but a basis resulting from a linear combination thereof).
- Nonetheless, the k-means algo on these vectors allows us to simply obtain the clusters.

Spectral clustering in practice V

Choosing the number of eigenvectors

- Choice of the number of clusters k = recurrent problem in clustering.
- Here, no probabilistic model therefore no BIC type criterion or one based on likelihood;
- But we can use other ad-hoc criteria, for eg looking at within-group and between-group similarity.
- A common technique is to use the heuristic 'eigengap': we choose the number of clusters k that realises the largest difference $\lambda_{k+1} - \lambda_k$

Spectral clustering in practice VI

Laplacian choice

- To choose which Laplacian matrix to use, it is recommended to look at the degree distribution of the graph. If this is homogeneous, so the choice of Laplacian has little impact on the result. On the other hand, if the degrees are very different, this is no longer the case.
- In general, the degrees are not homogeneous at all.

References for part 3 - I

[1] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre.
Fast unfolding of communities in large networks.
Journal of Statistical Mechanics : Theory and Experiment, 2008.

[2] M. Newman and M. Girvan.
Finding and evaluating community structure in networks.
Physical Review E, 2004.

[3] Ng, A. Y., M. I. Jordan, and Y. Weiss (2001).
On spectral clustering: Analysis and an algorithm.
In *Advances in neural information processing systems*, pp. 849–856. MIT Press.

[4] Rohe, K., S. Chatterjee, and B. Yu (2011).
Spectral clustering and the high-dimensional stochastic blockmodel.
Annals of Statistics 39(4), 1878–1915.

References for part 3 - II

[5] V. Traag.
Faster unfolding of communities : Speeding up the louvain algorithm.
Physical Review E, 2015.

[6] von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and Computing 17(4), 395–416.