

An introduction to graph analysis and modeling

Descriptive Analysis of Network Data

Julien Chiquet

September, 2019

<https://github.com/jchiquet/CourseNetworkLondon>

Statistical analysis of Networks

Different questions

Understanding the network topology

- Data = observed network
- Questions: central nodes? cluster structure? small-world property?

Inferring/Reconstructing the network

- Data = repeated signal observed at each node
- Questions: which nodes are connected?

Each to be combined with

covariates, time, heterogeneous data set, missing data, ...

Reconstruction and analysis of biological networks

E. coli regulatory network

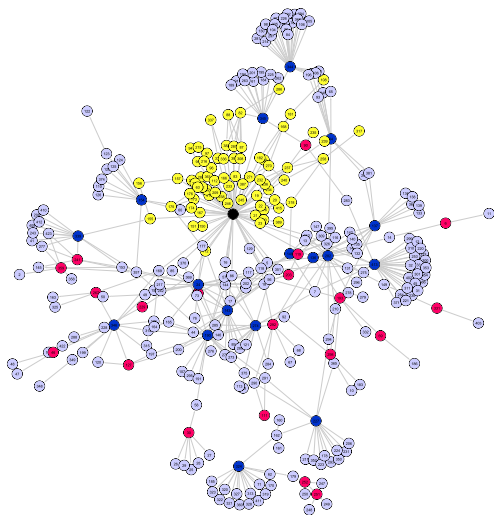
Target network

Relations between genes and their products

- highly structured
- always incomplete

Data and method

- transcriptomic data
- **Inference:** sparse Gaussian graphical model
- **Analysis:** Stochastic Block Model



Outline

- ① Basic notions on graphs and networks
- ② Descriptive statistics
- ③ Graph Partitioning
- ④ The Stochastic Block Model (SBM)

Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)

Graphs, Networks: some definitions

Definition (Network versus Graph)

- A **Network** is a collection of interacting entities
- A **Graph** is the mathematical representation of a network

Definition (Graph)

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a mathematical structure consisting of

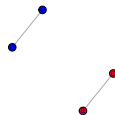
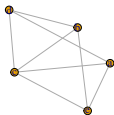
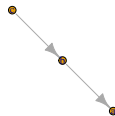
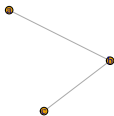
- a set $\mathcal{V} = \{1, \dots, n\}$ of **vertices** or **nodes**
- a set $\mathcal{E} = \{e_1, \dots, e_p : e_k = (i_k, j_k) \in (\mathcal{V} \times \mathcal{V})\}$ of **edges** or **links**
- The number of vertices $N_v = |\mathcal{V}|$ is called the **order**
- The number of edges $N_e = |\mathcal{E}|$ is called the **size**

Definition (Vocabulary)

subgraph, induced subgraph, (un)directed graph, weighted graph, bipartite graph, tree, DAG, path, cycle, connected components, etc.

Examples

Undirected, directed (digraph), complete, bipartite



Neighborhood, Degree

Definition (Neighborhood)

The neighbors of a vertex are the nodes directly connected to this vertex:

$$\mathcal{N}(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}.$$

Definition (Degree)

The degree d_i of a node i is given by its number of neighbors, i.e. $|\mathcal{N}(i)|$.

Remark

In digraphs, vertex degree is replaced by **in-degree** and **out-degree**.

Proposition

*In a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ the sum of the degree is given by $2|\mathcal{E}|$. Hence **this is always an even quantity**.*

Adjacency matrix and list of edges

Definition (Adjacency matrix)

The connectivity of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is captured by the $|\mathcal{V}| \times |\mathcal{V}|$ matrix \mathbf{A} :

$$(\mathbf{A})_{ij} = \begin{cases} 1 & \text{if } i \sim j, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition

The degree of \mathcal{G} are then simply obtained as the row-wise and/or column-wise sums of \mathbf{A} .

Remark

If the list of vertices is known, the only information which needs to be stored is the list of edges. In terms of storage, this is equivalent to a sparse matrix representation.

Layout and Visualization

- Visualization of large networks is a field of research in its own
- Be carefull with graphical interpretation of (large) networks

```
library(igraph)
library(sand)
GLattice <- graph.lattice(c(5,5,5))
GBlog <- aidsblog
```

Layout and Visualization

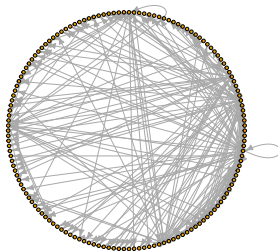
Example with circle plot

```
par(mfrow=c(1,2))  
plot(GLattice, layout=layout.circle); title("5x5x5 lattice")  
plot(GBlog , layout=layout.circle); title("blog network")
```

5x5x5 lattice



blog network

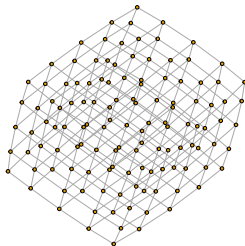


Layout and Vizualization

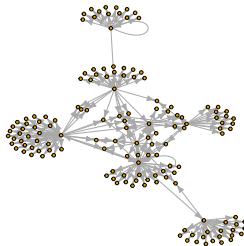
Example with Fruchterman and Reingold

```
par(mfrow=c(1,2))  
plot(GLattice, layout=layout.fruchterman.reingold); title("5x5x5 lattice")  
plot(GBlog , layout=layout.fruchterman.reingold); title("blog network")
```

5x5x5 lattice



blog network



Layout and Visualization: **ggraph** way I

```
library(ggraph)
library(gridExtra)
g1 <- ggraph(GBlog, layout = "fr") +
  geom_edge_link(color = "lightgray") + geom_node_point() + theme_void()

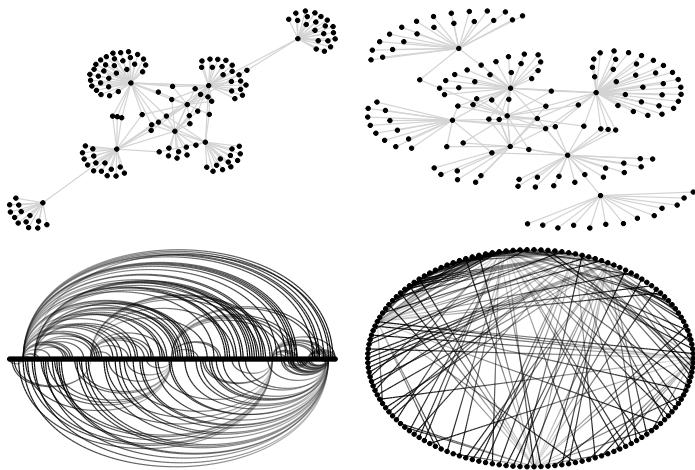
g2 <- ggraph(GBlog, layout = "kk") +
  geom_edge_link(color = "lightgray") + geom_node_point() + theme_void()

g3 <- ggraph(GBlog, layout = "linear") +
  geom_edge_arc(aes(alpha=..index..), show.legend = FALSE) +
  geom_node_point() + theme_void()

g4 <- ggraph(GBlog, layout = "linear", circular = TRUE) +
  geom_edge_link(aes(alpha=..index..), show.legend = FALSE) +
  geom_node_point() + theme_void()

grid.arrange(g1, g2, g3, g4, nrow = 2, ncol = 2)
```

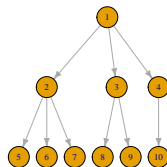
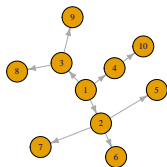
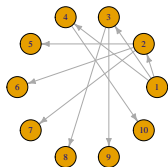
Layout and Visualization: **ggraph** way II



Layout and Vizualization

Do not be fooled by the plot

```
g.tree <- graph.formula(1-+2,1-+3,1-+4,2-+5,2-+6,2-+7, 3-+8,3-+9,4-+10)
par(mfrow=c(1, 3))
igraph.options(vertex.size=30, edge.arrow.size=0.5, vertex.label=NULL)
plot(g.tree, layout=layout.circle)
plot(g.tree, layout=layout.reingold.tilford(g.tree, circular=T))
plot(g.tree, layout=layout.reingold.tilford)
```



Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
 - Vertex characteristics
 - Local measurements
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)

Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
 - Vertex characteristics
 - Local measurements
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)

Vertex degree

Definition (Degree distribution)

In a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, recall that d_i counts the number of incident edges in \mathcal{E} to i . Define f_d to be the fraction of vertices $i \in \mathcal{V}$ with degree $d_i = d$. The collection $\{f_d, d \geq 0\}$ is called the **degree distribution** of \mathcal{G} .

Property

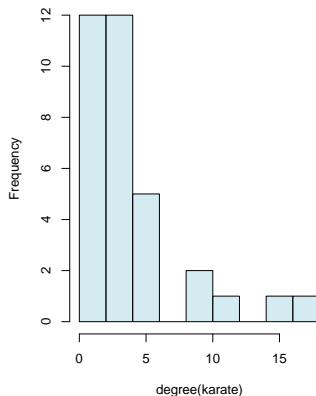
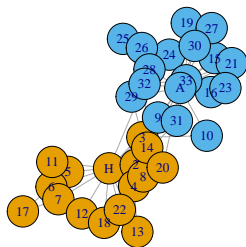
Many real world networks have a degree distribution fitting well power law distributions:

$$f_{d_i}(d) = \mathbb{P}(d_i = d) = \frac{c}{d^\gamma}, \quad c \in \mathbb{R}, \gamma > 0.$$

Those heavy-tail distributions describe few vertices with very high degrees.

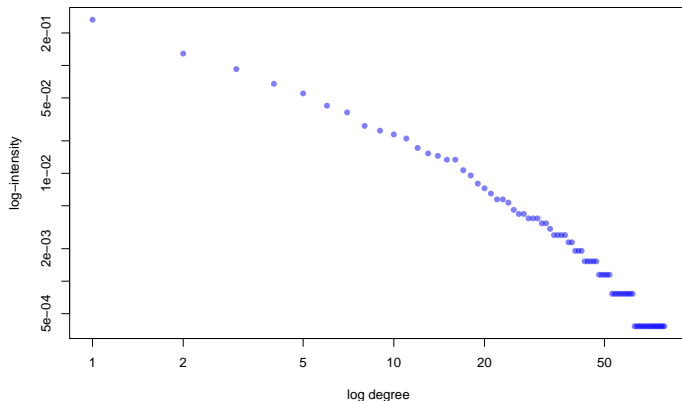
Vertex degree: example I

```
library(sand) data(karate)
par(mfrow=c(1,2)) plot(karate)
hist(degree(karate), col=adjustcolor("lightblue", alpha.f = 0.5), main="")
```



Vertex degree: example II

```
library(igraphdata) data(yeast)
degrees.yeast <- rev(sort(degree.distribution(yeast)))
plot(degrees.yeast[degrees.yeast!=0], log="xy", col=adjustcolor("blue", alpha.f =
0.5), pch=16, xlab="log degree", ylab="log-intensity")
```



Distance and diameter I

Definition (distance)

- **The Length** of a path e_1, \dots, e_k is the number of edges enterin the path (here k).
- If two nodes i, j are connected in G , then **the distance** ℓ_{ij} is the length of the shortest path between i and j . If the two nodes are not connected then $\ell_{ij} = \infty$.

Definition (diameter)

The diameter of \mathcal{G} is the greatest distance between two nodes:

$$\text{diameter}(\mathcal{G}) = \max_{(i,j) \in \mathcal{V} \times \mathcal{V}} (\ell_{ij})$$

Distance, Diameter: example I

```
library(Matrix)
data(ppi.CC)
diameter(ppi.CC)

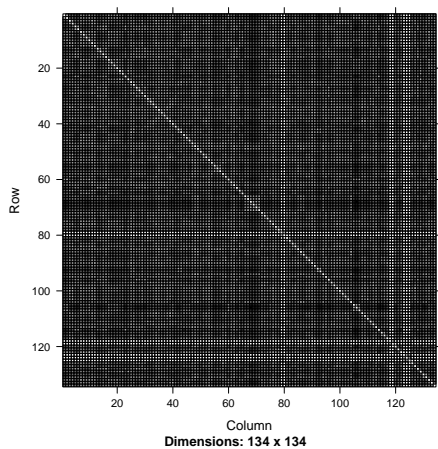
## [1] 12

average.path.length(ppi.CC)

## [1] 4.448039

image(Matrix(distances(ppi.CC)))
```

Distance, Diameter: example II



Vertex centrality: closeness

Question

How important is the node/vertice in the network?

Definition (Farness, Closeness)

Farness is the sum of the length of the shortest paths between the node and all other nodes in the graph. Closeness is defined as its reciprocal:

$$C(x) = \frac{1}{\sum_y d(y, x)}.$$

\rightsquigarrow *The more central a node is, the closer it is to all other nodes.*

Vertex centrality: betweenness

Question

How important is the node/vertex in the network?

Definition (Betweenness)

For every pairs of vertices, there exists at least one shortest path between the vertices such that the number of edges that the path passes through is minimized. The betweenness centrality for each vertex is the number of these shortest paths that pass through the vertex:

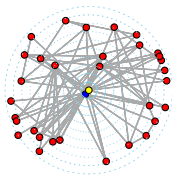
$$g(i) = \sum_{j \neq i \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}}$$

where σ_{jk} is the total number of shortest paths from node j to node k and $\sigma_{jk}(i)$ the number of those paths that pass through i .

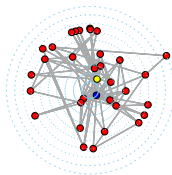
Example for karate club data set

administrator and instructor are in blue and yellow

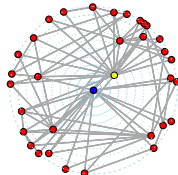
Degree



Closeness



Betweenness



Jaccard Coefficient

Definition (Jaccard Coefficient or Jaccard Index)

The Jaccard coefficient **measures similarity between finite sample sets**, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Example

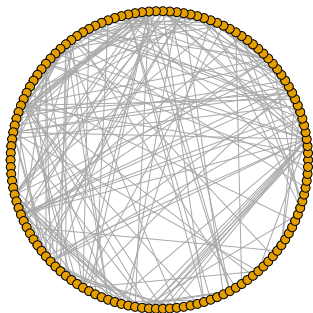
It can be used to compared two sets of egdes. For instance

- for two networks \mathcal{G} and \mathcal{H} defined on the same set of node, we can compare the sets $\mathcal{E}_{\mathcal{G}}$ and $\mathcal{E}_{\mathcal{H}}$.
- for a networks \mathcal{G} we can compute similarity between nodes with the Jaccard index and use it to define a weighted graph of similarity.

Jaccard Coefficient: example

Plot the yeast PPI interaction network

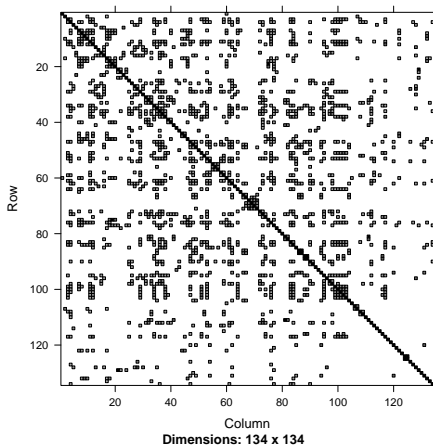
```
library(sand)
library(igraph)
plot(ppi.CC, vertex.size=6, vertex.label=NA, layout=layout_in_circle)
```



Jaccard Coefficient: example II

Compute Jaccard similarity between vertices and give a image of this

```
library(Matrix)
image(Matrix(igraph::similarity(ppi.CC, method = "jaccard")))
```



Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
 - Vertex characteristics
 - Local measurements**
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)

Density

Question

Is the network locally **dense** in some sense?

Definition (Clique)

In an undirected graph, a clique is a subset of the vertices such that **every two distinct vertices are adjacent**.

Definition (Density)

The density of a (sub)-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by

$$\text{density}(\mathcal{G}) = \frac{2|\mathcal{E}|}{|\mathcal{V}|(|\mathcal{V}| - 1)} = \frac{\bar{D}}{|V| - 1},$$

where \bar{D} is the mean degree of the network: how much \mathcal{G} is close to a clique?

Clustering

Question

Is the network locally **dense** in some sense?

Definition (Triangle)

Triplets of vertices in the graph that are connected through a triangle. They correspond to transitive relationships. We let

- $\tau_{\Delta}(i)$ be the number of triangles in \mathcal{G} where i falls.
- $\tau_3(i)$ be the number of triplets in \mathcal{G} where i falls.

Definition (Clustering coefficient)

$$\text{clustering}(\mathcal{G}) = \frac{1}{\mathcal{V}_2} \sum_{i \in \mathcal{V}_2} \tau_{\Delta}(i) / \tau_3(i),$$

where \mathcal{V}_2 is the set of vertices whose degree is greater or equal to 2.

Transitivity

Question

Is the network locally **dense** in some sense?

Definition (Triangle)

Triplet of vertices in the graph that are connected through a triangle. They correspond to transitive relationships. We let

- $\tau_{\Delta}(i)$ be the number of triangle in \mathcal{G} where i falls.
- $\tau_3(i)$ be the number of triplet in \mathcal{G} where i falls.

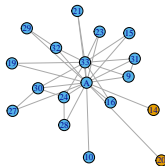
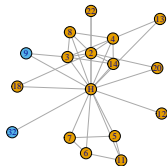
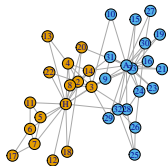
Definition (Transitivity)

$$\text{transitivity}(\mathcal{G}) = \frac{\sum_{\mathcal{V}} \tau_{\Delta}(i)}{\sum_{\mathcal{V}} \tau_3(i)},$$

Local density: example

Create ego graphs around teacher and instructor

```
data(karate)
ego.instr <- igraph::induced_subgraph(karate, neighborhood(karate, 1, 1)[[1]])
ego.admin <- igraph::induced_subgraph(karate, neighborhood(karate, 1, 34)[[1]])
```



Local density: example (II)

Look for graph density and transitivity/clustering either globally or locally

```
graph.density(karate)
```

```
## [1] 0.1390374
```

```
graph.density(ego.instr)
```

```
## [1] 0.25
```

```
graph.density(ego.admin)
```

```
## [1] 0.2091503
```

```
transitivity(karate)
```

```
## [1] 0.2556818
```

```
transitivity(karate, "local", vids = c(1,34))
```

```
## [1] 0.1500000 0.1102941
```

Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning**
 - Hierarchical clustering
 - Spectral Clustering
- 4 The Stochastic Block Model (SBM)

Principle of graph partitionning

Definition (Partition)

A decomposition $\mathcal{C} = \{C_1, \dots, C_K\}$ of the vertices \mathcal{V} such that

- $C_k \cap C_{k'} = \emptyset$ for any $k \neq k'$
- $\bigcup_k C_k = \mathcal{V}$

Goal of graph partitionning

Form a partition of the vertices with unsupervised approach where the \mathcal{C} is composed by "cohesive" sets of vertices, for instance,

- ① vertices well connected among themselves
- ② well separated from the remaining vertices

Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning**
 - Hierarchical clustering**
 - Spectral Clustering
- 4 The Stochastic Block Model (SBM)

Principle

Input: n individuals with p attributes)

1. Compute the dissimilarity between groups
2. Regroup the two most similar elements

Iterate until all element are in a single group

Output: n nested partitions from $\{\{1\}, \dots, \{n\}\}$ to $\{\{1, \dots, n\}\}$

Algorithm 1: Agglomerative hierarchical clustering

Ingredients

- ① a dissimilarity measure between singleton
- ② a distance measure between sets

Dissimilarity measures

Standards

Use standard distances on adjacency matrix:

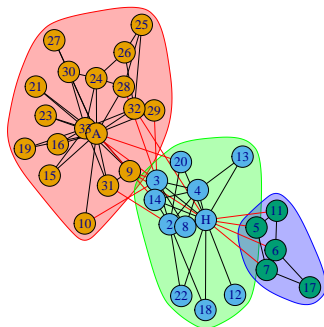
- Euclidean distance: $x_{ij} = \sqrt{\sum_{ik} (A_{ik} - A_{jk})^2}$
- Manhattan distance: $x_{ij} = \sum_{ik} |A_{ik} - A_{jk}|$
- etc. . .

Graph-specific

For instance, modularity, betweenness, etc.

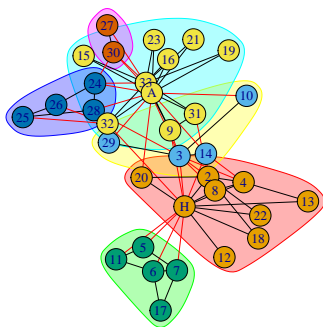
Examples of graph clustering I

```
hc <- cluster_fast_greedy(karate)  
plot(hc, karate)
```



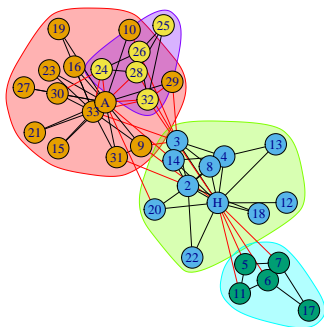
Examples of graph clustering II

```
hc <- cluster_edge_betweenness(karate)  
plot(hc, karate)
```



Examples of graph clustering III

```
hc <- cluster_walktrap(karate)  
plot(hc, karate)
```



Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning**
 - Hierarchical clustering
 - Spectral Clustering**
- 4 The Stochastic Block Model (SBM)

Graph Laplacian

Definition ((Un-normalized) Laplacian)

The Laplacian matrix \mathbf{L} , resulting from the modified incidence matrix $\tilde{\mathbf{B}}$ $\tilde{B}_{ij} = 1 / -1$ if i is incident to j as tail/head, is defined by

$$\mathbf{L} = \tilde{\mathbf{B}}\tilde{\mathbf{B}}^T = \mathbf{D} - \mathbf{A},$$

where $\mathbf{D} = \text{diag}(d_i, i \in \mathcal{V})$ is the diagonal matrix of degrees.

Remark

- \mathbf{L} is called Laplacian by analogy to the second order derivative (see below).
- Spectrum of \mathbf{L} has much to say about the structure of the graph \mathcal{G} .

Spectral Clustering

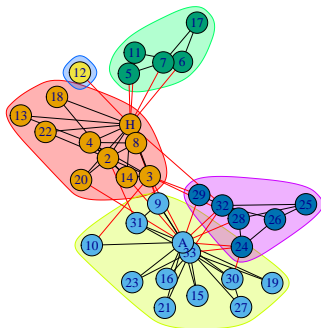
Principle

- ① Use the spectral property of \mathbf{L} to perform clustering in the eigen space
- ② If the network have K connected components, the first K eigenvectors are $\mathbf{1}$ span the eigenspace associated with eigenvalue 0
- ③ Applying a simple clustering algorithm to the rows of the K first eigenvectors separate the components

↪ This principle generalizes to a graph with a single component: spectral clustering tends to separates groups of nodes which are highly connected together

Clustering based on the first non null eigenvalue

```
hc <- cluster_leading_eigen(karate)  
plot(hc, karate)
```



Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)**
 - Some Graphs Models and their limitations
 - Mixture of Erdős-Rényi and the SBM

Motivations

Previous Section: find an underlying organization in a observed network

Spectral or hierarchical clustering for network data

⇒ Not model-based, thus no statistical inference possible

This Section: clustering of network based on a probabilistic model of the graph

Become familiar with

- the stochastic block model, a random graph model tailored for clustering vertices

hierarchical clustering \leftrightarrow Gaussian mixture models



hierarchical/spectral clustering for network \leftrightarrow Stochastic block model

Motivations

Previous Section: find an underlying organization in a observed network

Spectral or hierarchical clustering for network data

~> Not model-based, thus no statistical inference possible

This Section: clustering of network based on a probabilistic model of the graph

Become familiar with

- the stochastic block model, a random graph model tailored for clustering vertices

hierarchical clustering \leftrightarrow Gaussian mixture models



hierarchical/spectral clustering for network \leftrightarrow Stochastic block model

Motivations

Previous Section: find an underlying organization in a observed network

Spectral or hierachical clustering for network data

⇒ Not model-based, thus no statistical inference possible

This Section: clustering of network based on a probabilistic model of the graph

Become familiar with

- the stochastic block model, a random graph model tailored for clustering vertices

hierarchical clustering \leftrightarrow Gaussian mixture models



hierarchical/spectral clustering for network \leftrightarrow Stochastic block model

Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)
Some Graphs Models and their limitations
Mixture of Erdős-Rényi and the SBM

A mathematical model: Erdős-Rényi graph

Definition

Let $\mathcal{V} = 1, \dots, n$ be a set of fixed vertices. The (simple) Erdős-Rényi model $\mathcal{G}(n, \pi)$ assumes random edges between pairs of nodes with probability π . In other word, the (random) adjacency matrix \mathbf{X} is such that

$$X_{ij} \sim \mathcal{B}(\pi)$$

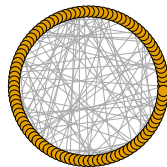
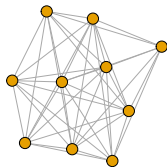
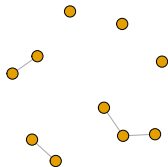
Proposition (degree distribution)

The (random) degree D_i of vertex i follows a binomial distribution:

$$D_i \sim b(n - 1, \pi).$$

Erdős-Rényi - example

```
G1 <- igraph::sample_gnp(10, 0.1)
G2 <- igraph::sample_gnp(10, 0.9)
G3 <- igraph::sample_gnp(100, .02)
par(mfrow=c(1,3))
plot(G1, vertex.label=NA) ; plot(G2, vertex.label=NA)
plot(G3, vertex.label=NA, layout=layout.circle)
```



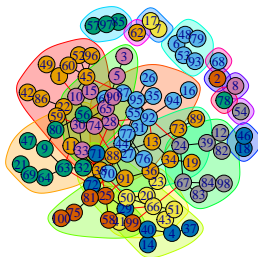
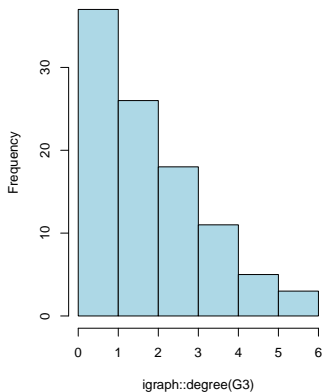
Erdős-Rényy - limitations: very homogeneous

```
average.path.length(G3); diameter(G3)
```

```
## [1] 4.642086
```

```
## [1] 10
```

Histogram of `igraph::degree(G3)`



Limitations

- Erdős-Rényi

The ER model does not fit well real world network

- As can be seen from its degree distribution
- ER is generally too homogeneous

The Stochastic Block Model

The SBM¹ generalizes ER in a mixture framework. It provides

- a statistical framework to adjust and interpret the parameters
- a flexible yet simple specification that fits many existing network data

¹Other models exist (e.g. exponential model for random graphs) but less popular.

Outline

- 1 Basic notions on graphs and networks
- 2 Descriptive statistics
- 3 Graph Partitioning
- 4 The Stochastic Block Model (SBM)
 - Some Graphs Models and their limitations
 - Mixture of Erdős-Rényi and the SBM

Stochastic Block Model: definition

Mixture model point of view: mixture of Erdős-Rényi

Latent structure

Let $\mathcal{V} = \{1, \dots, n\}$ be a fixed set of vertices. We give each $i \in \mathcal{V}$ a **latent label** among a set $\mathcal{Q} = \{1, \dots, Q\}$ such that

- $\alpha_q = \mathbb{P}(i \in q), \quad \sum_q \alpha_q = 1;$
- $Z_{iq} = \mathbf{1}_{\{i \in q\}}$ are independent hidden variables.

The conditional distribution of the edges

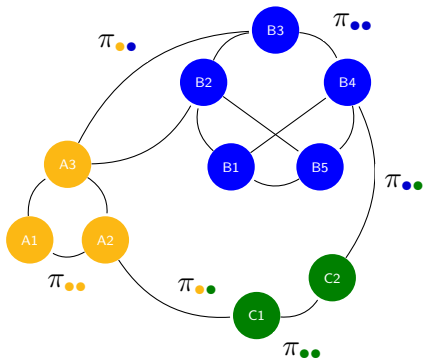
Connexion probabilities depend on the node class belonging:

$$X_{ij} | \{i \in q, j \in \ell\} \sim \mathcal{B}(\pi_{q\ell}) \quad \left(\Leftrightarrow X_{ij} | \{Z_{iq}Z_{j\ell} = 1\} \sim \mathcal{B}(\pi_{q\ell}). \right)$$

The $Q \times Q$ matrix π gives for all couple of labels

$$\pi_{q\ell} = \mathbb{P}(X_{ij} = 1 | i \in q, j \in \ell).$$

Stochastic Block Model: the big picture



Stochastic Block Model

Let n nodes divided into

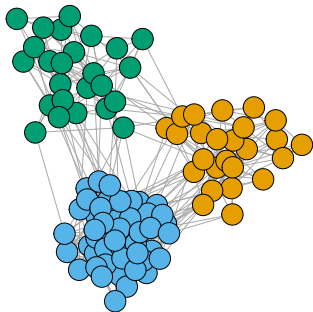
- $\mathcal{Q} = \{\bullet, \bullet, \bullet\}$ classes
- $\alpha_{\bullet} = \mathbb{P}(i \in \bullet), \bullet \in \mathcal{Q}, i = 1, \dots, n$
- $\pi_{\bullet\bullet} = \mathbb{P}(i \leftrightarrow j | i \in \bullet, j \in \bullet)$

$$Z_i = \mathbf{1}_{\{i \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \alpha), \quad \forall \bullet \in \mathcal{Q},$$
$$X_{ij} | \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \mathcal{B}(\pi_{\bullet\bullet})$$

Stochastic block models – examples of topology

Community network

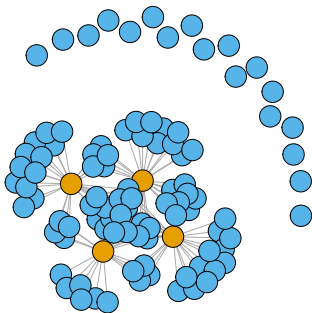
```
pi <- matrix(c(0.3,0.02,0.02,0.02,0.3,0.02,0.02,0.02,0.3),3,3)
communities <- igraph::sample_sbm(100, pi, c(25, 50, 25))
plot(communities, vertex.label=NA, vertex.color = rep(1:3,c(25, 50, 25)))
```



Stochastic block models – examples of topology

Star network

```
pi <- matrix(c(0.05,0.3,0.3,0),2,2)
star <- igraph::sample_sbm(100, pi, c(4, 96))
plot(star, vertex.label=NA, vertex.color = rep(1:2,c(4,96)))
```



Likelihoods for Expectation Maximization

Complete-data loglikelihood

$$\log L(\mathbf{X}, \mathbf{Z}) = \sum_{i,q} Z_{iq} \log \alpha_q + \sum_{i < j, q, \ell} Z_{iq} Z_{j\ell} \log \pi_{q\ell}^{X_{ij}} (1 - \pi_{q\ell})^{1-X_{ij}}.$$

Conditional expectation of the complete-data loglikelihood

$$\mathbb{E}_{\mathbf{Z}|\mathbf{X}}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})] = \sum_{i,q} \tau_{iq} \log \alpha_q + \sum_{i < j, q, \ell} \eta_{ijq\ell} \log \pi_{q\ell}^{X_{ij}} (1 - \pi_{q\ell})^{1-X_{ij}}$$

where $\tau_{iq}, \eta_{ijq\ell}$ are the posterior probabilities:

- $\tau_{iq} = \mathbb{P}(Z_{iq} = 1 | \mathbf{X}) = \mathbb{E}[Z_{iq} | \mathbf{X}]$.
- $\eta_{ijq\ell} = \mathbb{P}(Z_{iq} Z_{j\ell} = 1 | \mathbf{X}) = \mathbb{E}[Z_{iq} Z_{j\ell} | \mathbf{X}]$.

Inference in the SBM

The EM strategy does not apply

Ouch: another intractability problem

- the Z_{iq} are **not independent** in the SBM framework. . .
- we cannot compute $\eta_{ijql} = \mathbb{P}(Z_{iq}Z_{jl} = 1|\mathbf{X}) = \mathbb{E}[Z_{iq}Z_{jl}|\mathbf{X}]$,
- the conditional expectation $Q(\boldsymbol{\theta})$, i.e. the main EM ingredient, is **intractable**.

Solution: mean field approximation (variational inference)

Approximate η_{ijql} by $\tau_{iq}\tau_{jl}$, i.e., **assume independence between Z_{iq}**
 \rightsquigarrow This can be formalized in the variational framework

Model selection: the number of blocks/clusters

We use our lower bound of the loglikelihood to compute an approximation of the ICL

$$\begin{aligned} \text{vICL}(Q) = \mathbb{E}_{\hat{\mathbb{Q}}}[\log L(\hat{\boldsymbol{\theta}}; \mathbf{X}, \mathbf{Z})] \\ - \frac{1}{2} \left(\frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2} + (Q-1) \log(n) \right), \end{aligned}$$

where

$$\mathbb{E}_{\hat{\mathbb{Q}}}[\log L(\hat{\boldsymbol{\theta}}; \mathbf{X}, \mathbf{Z})] = J(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\theta}}) - \mathcal{H}(\hat{\mathbb{Q}}).$$

The variational BIC is just

$$\text{vBIC}(Q) = J(\hat{\boldsymbol{\tau}}, \hat{\boldsymbol{\theta}}) - \frac{1}{2} \left(\frac{Q(Q+1)}{2} \log \frac{n(n-1)}{2} + (Q-1) \log(n) \right).$$

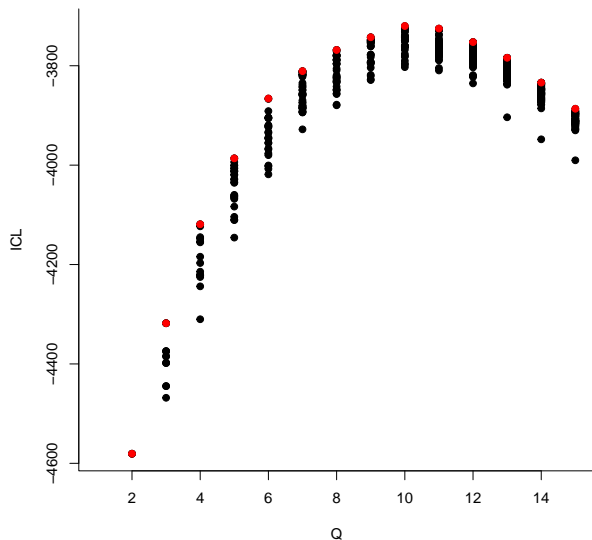
Example on the French blogosphere (I)

```
library(blockmodels)
library(sand)

adj_blog <- upgrade_graph(fbblog) %>%
  as_adjacency_matrix() %>%
  as.matrix()

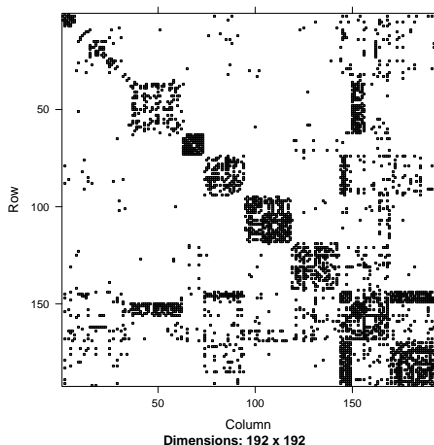
mySBM_collection <- BM_bernoulli(
  "SBM_sym",
  adj_blog, verbosity = 0,
  plotting = "figures/ICL_fbblog.pdf"
)
mySBM_collection$estimate()
```

Example on the French blogosphere (II)



Example on the French blogosphere (III)

```
clusters <-  
  apply(mySBM_collection$memberships[[10]]$Z, 1, which.max)  
image(Matrix(adj_blog[order(clusters), order(clusters)]))
```



Example on the French blogosphere (IV) I

```
library(RColorBrewer); pal <- brewer.pal(10, "Set3")

g <- graph_from_adjacency_matrix(adj_blog, mode = "undirected", weighted = TRUE, di
V(g)$class <- clusters
V(g)$size <- 5
V(g)$frame.color <- "white"
V(g)$color <- pal[V(g)$class]
V(g)$label <- ""
E(g)$arrow.mode <- 0

par(mar = c(0,0,0,0))
plot(g, edge.width=E(g)$weight)
```

Example on the French blogosphere (IV) II

