

# Masterclass in Machine Learning

## Graph clustering and the Stochastic Bloc Model

Villiers-le-lac

January 2022

<https://jchiquet.github.io/>

# Setup and Reproducibility

```
library(tidyverse) # data manipulation
library(igraph)    # graph manipulation
library(sbm)       # stochastic bloc model
library(missSBM)   # stochastic bloc model with missing data
library(aricode)   # clustering measures comparison
```

# Outline

- ① Motivations
- ② Graph Partitionning
  - Hierarchical clustering for graph
  - Spectral methods
- ③ The Stochastic Block Model (SBM)
  - Some Graphs Models and their limitations
  - Mixture of Erdös-Rényi and the SBM
  - Statistical Inference in the SBM
  - SBM: some extensions

# Outline

- ① Motivations
- ② Graph Partitioning
- ③ The Stochastic Block Model (SBM)

## Network data

### Recommendation system: Epinion

Who-trust-whom online social network of a general consumer review site Epinions.com. Members of the site can decide whether to "trust" each other.

### Social networks in ethnobiology

A seed exchange network in Kenya is collected on a limited space area, where all the 155 farmers are interviewed. Farmers provide information about other farmers with whom they have interacted.

### Ecological networks: plant-pollinator network

Interaction network between predefined sets of plants and pollinators, by direct observation.

# Companion data set: French political Blogosphere

Single day snapshot of almost 200 political blogs automatically extracted the 14 October 2006 and manually classified by the "Observatoire Présidentielle" project.

```
data("frenchblog2007", package = "missSBM")
blog <- frenchblog2007 %>% delete_vertices(which(degree(frenchblog2007) <= 1))
summary(blog)

## IGRAPH 6fb607c UN-- 192 1431 --
## + attr: name (v/c), party (v/c)

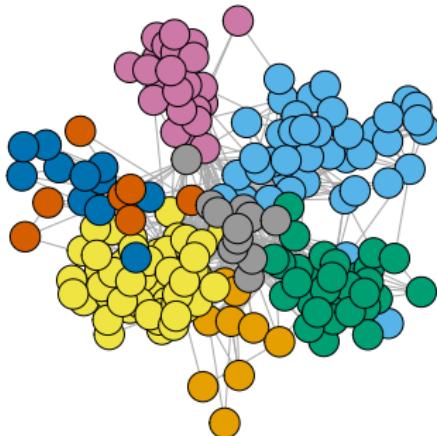
party <- V(blog)$party %>% as_factor()
party %>% table() %>% knitr::kable("latex")
```

.	Freq
green	9
right	40
center-rigth	32
left	57
center-left	11
far-left	7
liberal	25
analyst	11

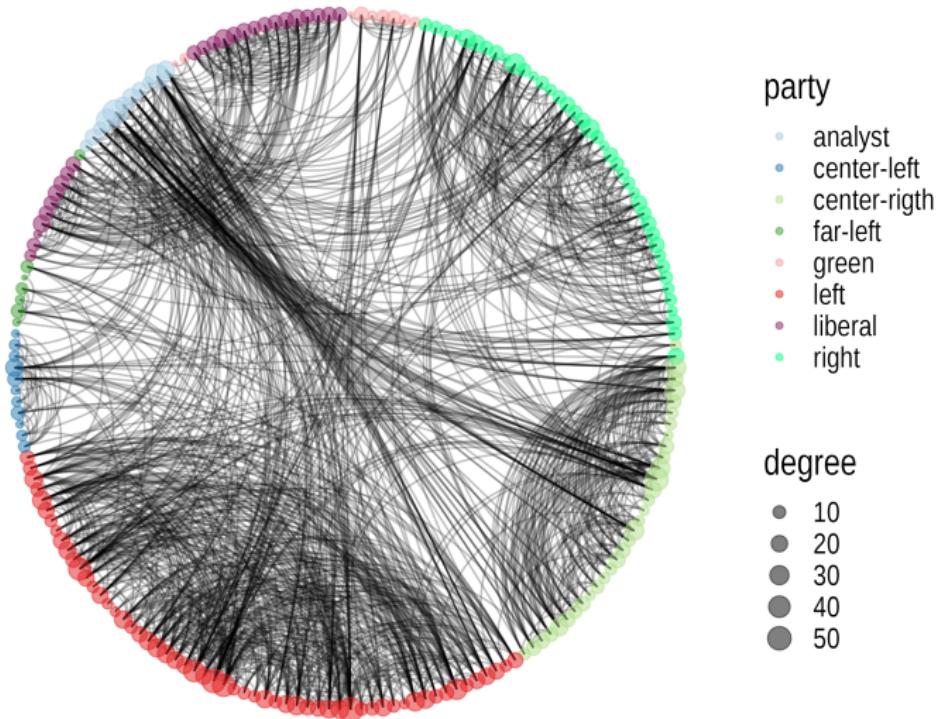
## Visualization: graph view

A visual representation of the network data with nodes colored according to the political party each blog belongs to is achieved as follows:

```
plot.igraph(blog,
  vertex.color = party,
  vertex.label = NA
)
```

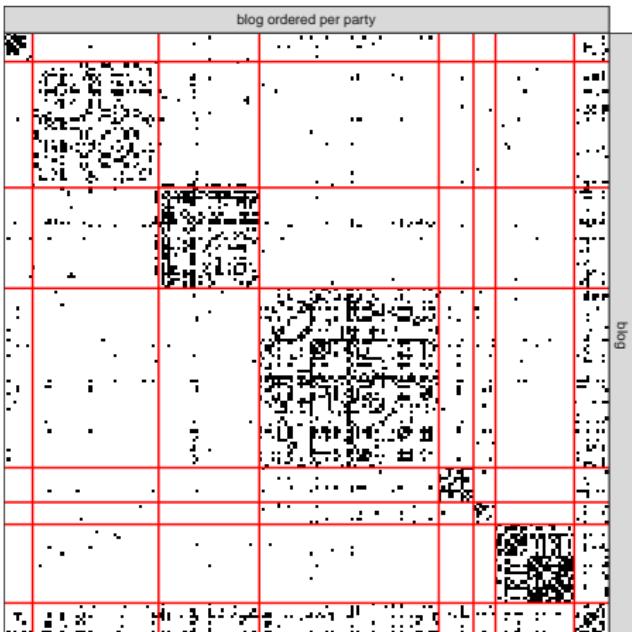


# Visualization: graph view (advanced)



## Visualization: matrix view

```
Y <- as_adj(blog, sparse = FALSE)
sbm::plotMyMatrix(
  Y, dimLabels = list('blog', "blog ordered per party"),
  clustering = list(row = party))
```



# Questions

## Remarks

- The pattern of connections between the nodes is highly related to the blog classification (political party).
- The data may support a natural grouping of the node which is not necessarily related to a predefined classification
- Same remark holds for any kind of clustering and unsupervised learning problem

## Objective: Graph clustering

Automatically find a **partitioning** of the nodes, i.e. a clustering, that groups together nodes with similar connectivity pattern.

# Network data and binary graphs: minimal notation

A **network** is a collection of interacting entities. A **graph** is the mathematical representation of a network.

## Definition

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  $|\mathcal{V}|$  is called the **order**
- The number of edges  $|\mathcal{E}|$  is called the **size**
- The neighbors of a vertex are the nodes directly connected to this vertex:

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

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

## Representation: adjacency matrix

The connectivity of a binary undirected (symmetric) graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is captured by the  $|\mathcal{V}| \times |\mathcal{V}|$  matrix  $Y$ , called the adjacency matrix

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

For a valued or weighted graph, a similar definition would be

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

where  $w_{ij}$  is the weight associated with edge  $i \sim j$ .

### 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.*

# Outline

① Motivations

② Graph Partitioning

Hierarchical clustering for graph

Spectral methods

③ The Stochastic Block Model (SBM)

## References

-  Statistical Analysis of Network Data: Methods and Models,  
Eric Kolaczyk  
**Chapiter 4, Section 4**
-  Analyse statistique de graphes,  
Catherine Matias, **Chapitre 3**
-  DS David Sontag's Lecture  
[http://people.csail.mit.edu/dsontag/courses/ml13/  
slides/lecture16.pdf](http://people.csail.mit.edu/dsontag/courses/ml13/slides/lecture16.pdf)
-  A Tutorial on Spectral Clustering,  
Ulrike von Luxburg

# Outline

① Motivations

② Graph Partitionning

Hierarchical clustering for graph

Spectral methods

③ The Stochastic Block Model (SBM)

# Principle

Form a partition of the nodes composed by "cohesive" sets, e.g.

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

## Agglomerative hierarchical clustering

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  $\{\{1, \dots, n\}\}$

## Ingredients

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

# Dissimilarity measures

## Graph-specific

- **Modularity:** *fraction of edges that fall within a given groups minus expected fraction if edges were distributed at random*

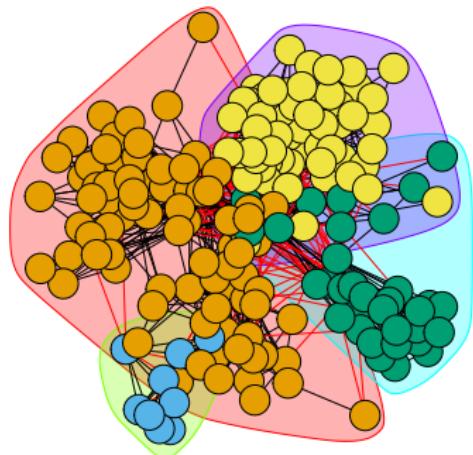
For  $\mathcal{C} = \{C_1, \dots, C_K\}$  a candidate partition and  $f_{ij}(\mathcal{C})$  the fraction of edges connecting vertices from  $C_i$  to  $C_j$

$$\text{modularity}(\mathcal{C}) = \sum_{k=1}^K (f_{kk}(\mathcal{C}) - \mathbb{E}_{H_0}(f_{kk}))^2$$

- **Betweenness:** *number of shortest paths that go through a node in a graph or network*

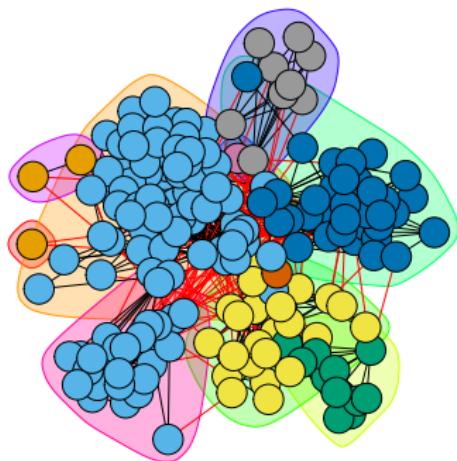
# Examples of graph partitioning I

```
hc <- cluster_fast_greedy(blog)
plot(hc, blog, vertex.label=NA)
```



## Examples of graph partitioning II

```
hc <- cluster_edge_betweenness(blog)  
plot(hc, blog, vertex.label=NA)
```



# Outline

① Motivations

② Graph Partitionning

Hierarchical clustering for graph

Spectral methods

③ The Stochastic Block Model (SBM)

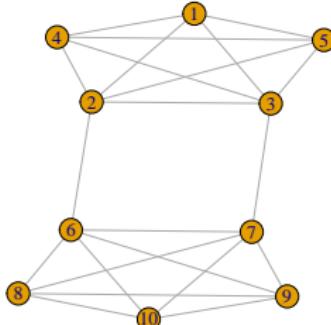
# Motivation: graph-cut

## Definition

The cut between two sets of nodes that form a partition in the graph is

$$\text{cut}(\mathcal{V}_A, \mathcal{V}_B) = \sum_{i \in \mathcal{V}_A, j \in \mathcal{V}_B} Y_{ij}, \quad \mathcal{V}_A \cup \mathcal{V}_B = \mathcal{V}$$

**Example:** The graph cut between  $\mathcal{V}_A = \{1, 2, 3, 4, 5\}$  and  $\mathcal{V}_B = \{6, 7, 8, 9, 10\}$  is 2.



## Min-cut

Idea: Find the 2-partition that minimizes the cut to form two homogeneous clusters.

### Min-cut problem

Based on this principle, the normalized cut consider the connectivity between groups relative to the volume of each groups

$$\arg \min_{\{\mathcal{V}_A, \mathcal{V}_B\}} \text{cut}^N(\mathcal{V}_A, \mathcal{V}_B),$$

where  $\text{Vol}(\mathcal{V}_S) = \sum_{i \in S} d_i$  and

$$\begin{aligned}\text{cut}^N(\mathcal{V}_A, \mathcal{V}_B) &= \frac{\text{cut}(\mathcal{V}_A, \mathcal{V}_B)}{\text{Vol}(\mathcal{V}_A)} + \frac{\text{cut}(\mathcal{V}_A, \mathcal{V}_B)}{\text{Vol}(\mathcal{V}_B)} \\ &= \text{cut}(\mathcal{V}_A, \mathcal{V}_B) \frac{\text{Vol}(\mathcal{V}_A) + \text{Vol}(\mathcal{V}_B)}{\text{Vol}(\mathcal{V}_A)\text{Vol}(\mathcal{V}_B)}\end{aligned}$$

## Solving min-cut for 2 clusters

Let

$$x = (x_i)_{i=1,\dots,n} = \begin{cases} -1 & \text{if } i \in \mathcal{V}_A, \\ 1 & \text{if } i \in \mathcal{V}_B. \end{cases}$$

Then, letting  $D$  the diagonal matrix of degrees,

$$x^\top (D - Y)x = x^\top Dx - (x^\top Dx - 2\text{cut}(\mathcal{V}_A, \mathcal{V}_B)),$$

so that

$$\text{cut}(\mathcal{V}_A, \mathcal{V}_B) = \frac{1}{2}x^\top (D - Y)x.$$

## Solving Min-cut for 2 clusters

Normalized graph-cut  $\Leftrightarrow$  integer programming problem

$$\arg \min_{\{\mathcal{V}_A, \mathcal{V}_B\}} \text{cut}^N(\mathcal{V}_A, \mathcal{V}_B)$$

$$\Leftrightarrow \arg \min_{x \in \{-1,1\}^n} \frac{x^\top (D - Y)x}{x^\top D x}, \quad \text{s.c.} \quad x^\top D \mathbf{1}_n = 0,$$

where the constraint imposes only discrete values in  $x$ .

Relax version

If we relax to  $x \in [-1, 1]^n$ , it turns to a simple eigenvalue problem

$$\arg \min_{x \in [-1,1]^n} x^\top (D - Y)x, \quad \text{s.c.} \quad x^\top D x = 1 \Leftrightarrow (D - Y)x = \lambda D x.$$

where  $\mathbf{L} = D - Y$  is called the Laplacian matrix of the graph  $\mathcal{G}$ .

## Solving Min-cut for 2 clusters

Normalized graph-cut  $\Leftrightarrow$  integer programming problem

$$\begin{aligned} & \arg \min_{\{\mathcal{V}_A, \mathcal{V}_B\}} \text{cut}^N(\mathcal{V}_A, \mathcal{V}_B) \\ \Leftrightarrow & \arg \min_{x \in \{-1,1\}^n} \frac{x^\top (D - Y)x}{x^\top Dx}, \quad \text{s.c.} \quad x^\top D \mathbf{1}_n = 0, \end{aligned}$$

where the constraint imposes only discrete values in  $x$ .

Relax version

If we relax to  $x \in [-1, 1]^n$ , it turns to a simple eigenvalue problem

$$\arg \min_{x \in [-1,1]^n} x^\top (D - Y)x, \quad \text{s.c.} \quad x^\top D x = 1 \Leftrightarrow (D - Y)x = \lambda D x.$$

where  $\mathbf{L} = D - Y$  is called the **Laplacian matrix** of the graph  $\mathcal{G}$ .

## Graph Laplacian: spectrum

Proposition (Spectrum of  $\mathbf{L}$ )

*The  $n \times n$  matrix  $\mathbf{L}$  has the following properties:*

$$\mathbf{x}^\top \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i,j} Y_{ij} (x_i - x_j)^2, \quad \forall \mathbf{x} \in \mathbb{R}^n.$$

- $\mathbf{L}$  is a symmetric, positive semi-definite matrix,
- $\mathbf{1}_n$  is in the kernel of  $L$  since  $L\mathbf{1}_n = 0$ ,
- The first normalized eigen vector with eigen value  $\lambda > 0$  is solution to the relaxed graph cut problem

The Laplacian is easily (and fastly) computed in R thanks to the **igraph** package:

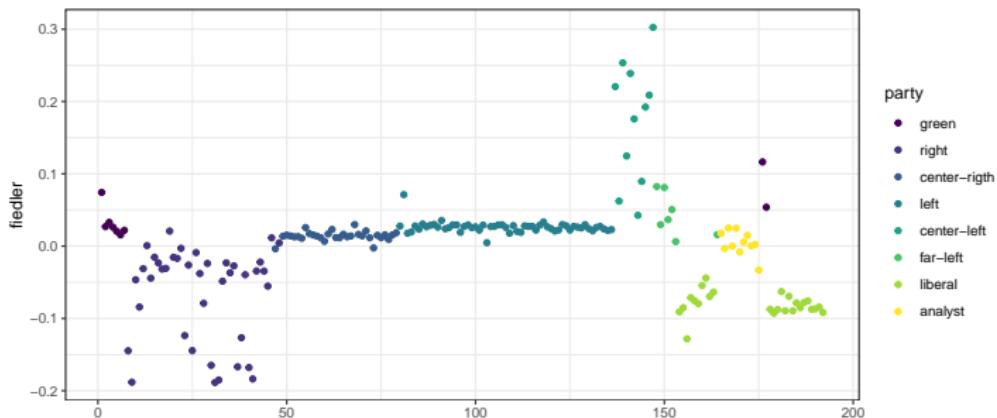
```
L <- laplacian_matrix(blog)
```

# Bi-partionning and the Fiedler vector

Fiedler vector is the named sometimes given to the normalized eigen vector associated with the smallest positive eigen-value of  $\mathbf{L}$ .

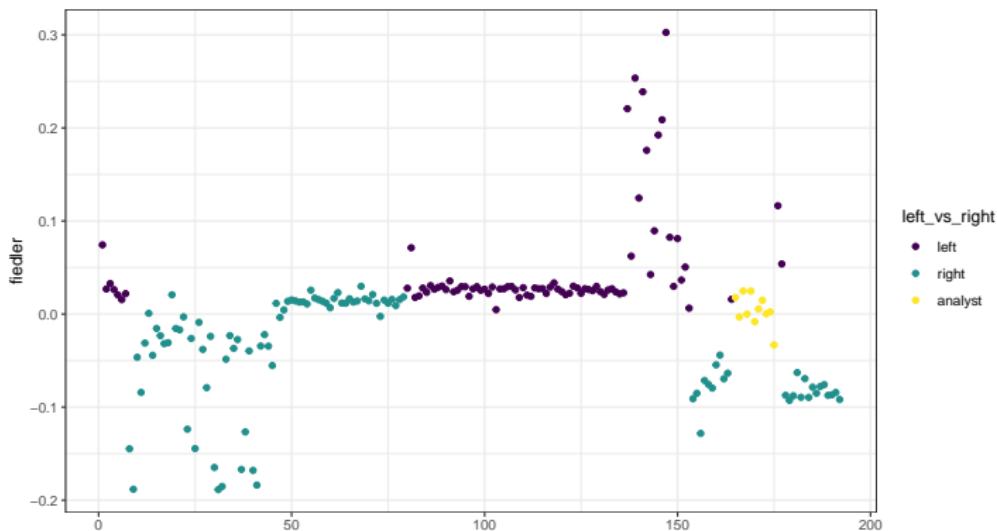
- solves the relaxed min-cut problem
- can be used to compute a bi-partition of a graph.

```
spec_L <- eigen(L); practical_zero <- 1e-12
lambda <- min(spec_L$values[spec_L$values>practical_zero])
fiedler <- spec_L$vectors[, which(spec_L$values == lambda)]
qplot(y = fiedler, colour = party) + viridis::scale_color_viridis(discrete = TRUE)
```



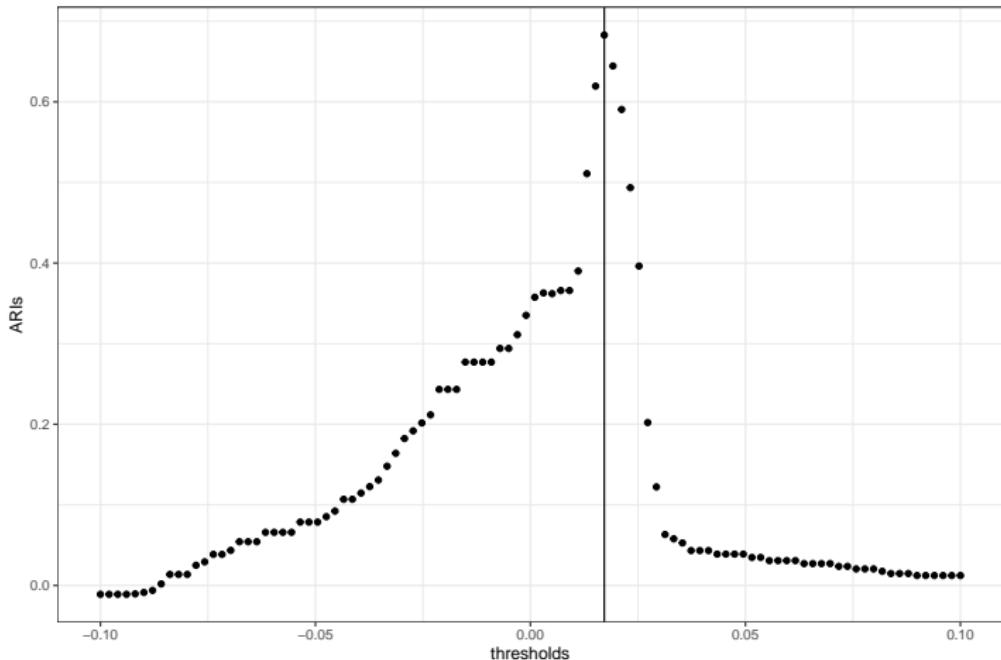
## Example on a simplified left/right view

```
left_vs_right <-
 forcats::fct_collapse(party,
  left = c("green", "left", "far-left", "center-left"),
  right = c("right", "liberal", "center-rigth"),
  analyst = "analyst"
)
qplot(y = fiedler, colour = left_vs_right) + viridis::scale_color_viridis(discrete=
```



# "Validation"

```
thresholds <- seq(-.1, .1, len = 100)
ARIs <- map_dbl(thresholds, ~ARI(left_vs_right, fiedler > .))
qplot(thresholds, ARIs) + geom_vline(xintercept = thresholds[which.max(ARIs)]) + th
```



# Spectral clustering

From the definition of the Laplacian matrix,

- The multiplicity of the first eigen value (0) of  $\mathbf{L}$  determines the number of connected components in the graph.
- The larger the second non trivial (positive) eigenvalue, the higher the connectivity of  $\mathcal{G}$ .

## General Heuristic

- ① Compute spectral decomposition of  $\mathbf{L}$  to perform clustering in the eigen space
- ② For a graph with  $K$  connected components, the first  $K$  eigen-vectors are  $\mathbf{1}$  spanning the eigenspace associated with eigenvalue 0
- ③ Applying a simple clustering algorithm to the rows of the  $K$  first eigenvectors separate the components

~~ Generalizes to graphs with a single component (tends to separates groups of nodes which are highly connected together)

## Some variants

### Definition ((Normalized) Laplacian)

The normalized Laplacian matrix  $\mathbf{L}$  is defined by

$$\mathbf{L}_N = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}.$$

### Definition ((Absolute) Graph Laplacian)

The absolute Laplacian matrix  $\mathbf{L}_{abs}$  is defined by

$$\mathbf{L}_{abs} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{L}_N,$$

with eigenvalues  $1 - \lambda_n \leq \dots \leq 1 - \lambda_2 \leq 1 - \lambda_1 = 1$ , where  $0 = \lambda_1 \leq \dots \leq \lambda_n$  are the eigenvalues of  $\mathbf{L}_N$ .

# Normalized Spectral Clustering

by Ng, Jordan and Weiss (2002)

**Input:** Adjacency matrix and number of classes  $Q$

Compute the normalized graph Laplacian  $\mathbf{L}$

Compute the eigen vectors of  $\mathbf{L}$  associated with the  $Q$  **smallest eigenvalues**

Define  $\mathbf{U}$ , the  $n \times Q$  matrix that encompasses these  $Q$  vectors

Define  $\tilde{\mathbf{U}}$ , the row-wise normalized version of  $\mathbf{U}$ :  $\tilde{u}_{ij} = \frac{u_{ij}}{\|\mathbf{U}_i\|_2}$

Apply k-means to  $(\tilde{\mathbf{U}}_i)_{i=1,\dots,n}$

**Output:** vector of classes  $\mathbf{C} \in \mathcal{Q}^n$ , such as  $C_i = q$  if  $i \in q$

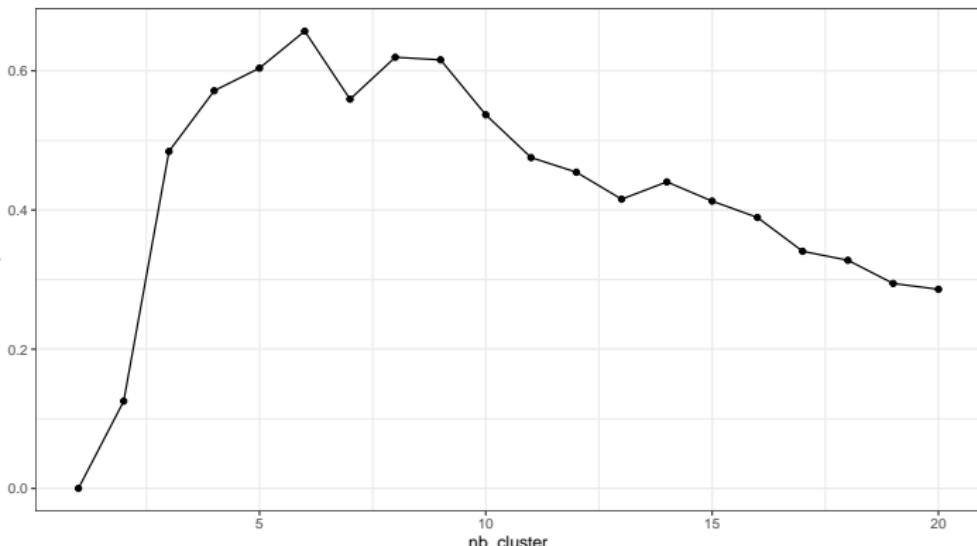
# Implementation of normalized spectral clustering

```
spectral_clustering <- function(graph, nb_cluster, normalized = TRUE) {  
  
    ## Compute Laplacian matrix  
    L <- laplacian_matrix(graph, normalized = normalized)  
    ## Generates indices of last (smallest) K vectors  
    selected <- rev(1:ncol(L))[1:nb_cluster]  
    ## Extract an normalized eigen-vectors  
    U <- eigen(L)$vectors[, selected, drop = FALSE] # spectral decomposition  
    U <- sweep(U, 1, sqrt(rowSums(U^2)), '/')  
    ## Perform k-means  
    res <- kmeans(U, nb_cluster, nstart = 40)$cl  
  
    res  
}
```

# Application to the French blogosphere (1)

Perform spectral clustering on the blogosphere for various numbers of group:

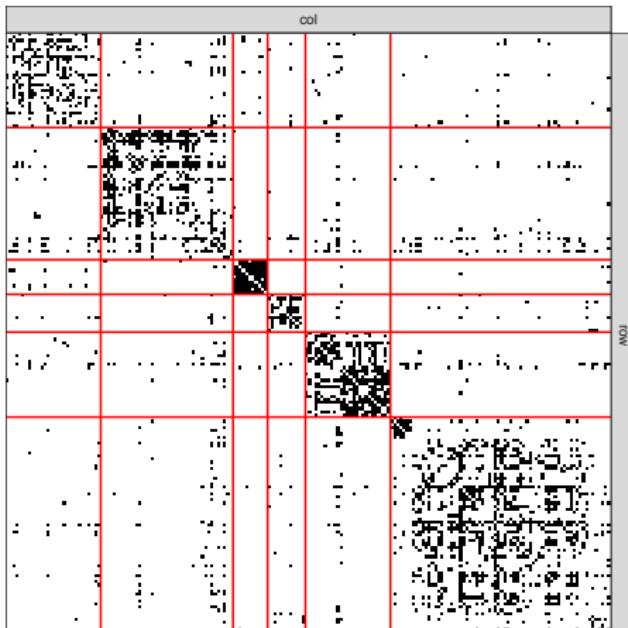
```
nb_cluster <- 1:20
map(nb_cluster, ~spectral_clustering(blog, .)) %>%
  map_dbl(ARI, party) %>% qplot(nb_cluster, y = .) + geom_line() + theme_bw()
```



## Application to the French blogosphere (2)

Once reorder according to the best clustering (obtained  $k = 6$ ) groups, the orginal data matrix looks as follows

```
plotMyMatrix(as_adj(blog, sparse = FALSE),  
            clustering = list(row = spectral_clustering(blog, 6)))
```



# Outline

- ① Motivations
- ② Graph Partitionning
- ③ The Stochastic Block Model (SBM)

Some Graphs Models and their limitations  
Mixture of Erdös-Rényi and the SBM  
Statistical Inference in the SBM  
SBM: some extensions

# References

-  Statistical Analysis of Network Data: Methods and Models  
Eric Kolaczky  
**Chapters 5 and 6**
-  Mixture model for random graphs, Statistics and Computing  
Daudin, Robin, Picard  
[pbil.univ-lyon1.fr/members/fpicard/franckpicard\\_fichiers/pdf/DPR08.pdf](http://pbil.univ-lyon1.fr/members/fpicard/franckpicard_fichiers/pdf/DPR08.pdf)
-  Analyse statistique de graphes,  
Catherine Matias  
**Chapitre 4, Section 4**

# Motivations

Last section: find an underlying organization in a observed network

Spectral or hierarchical clustering for network data

~ Not model-based, thus no statistical inference possible

Now: 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,
- the variational EM algorithm used to infer SBM from network data.

hierarchical/kmeans clustering  $\leftrightarrow$  Gaussian mixture models



hierarchical/spectral clustering for network  $\leftrightarrow$  Stochastic block model

# Outline

- ① Motivations
- ② Graph Partitionning
- ③ The Stochastic Block Model (SBM)
  - Some Graphs Models and their limitations
    - Mixture of Erdös-Rényi and the SBM
    - Statistical Inference in the SBM
    - SBM: some extensions

# A mathematical model: Erdös-Rényi graph

## Definition

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

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

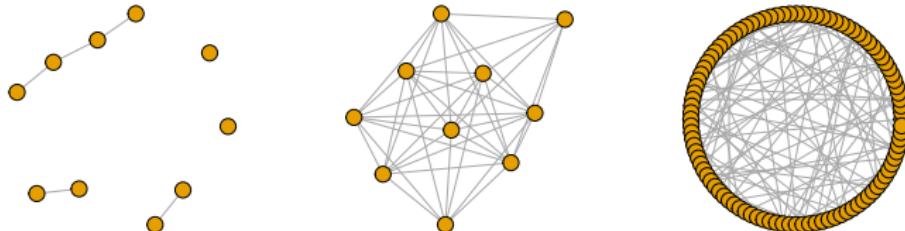
## Proposition (degree distribution)

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

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

## Erdös-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)
```

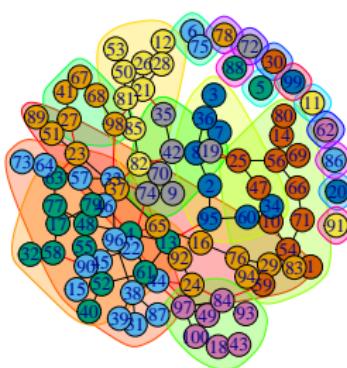
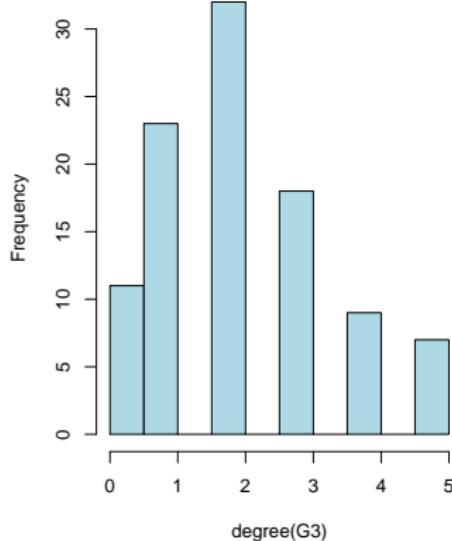


# Erdös-Rény - limitations: very homogeneous

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

```
## [1] 5.649385  
## [1] 13
```

Histogram of degree(G3)



# Mechanism-based model: preferential attachment

The graph is defined dynamically as follows

## Definition

Start from a initial graph  $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ , then for each time step,

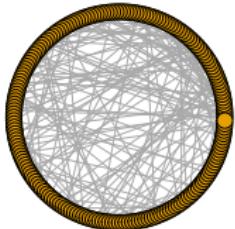
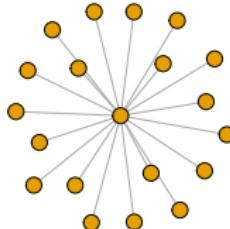
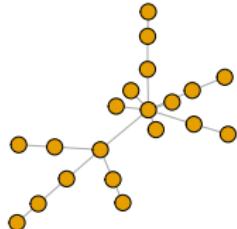
- ① At  $t$  a new node  $V_t$  is added
- ②  $V_t$  is connected to  $i \in V_{t-1}$  with probability

$$D_i^\alpha + \text{cst.}$$

~~ Nodes with high degree get more connections thus **richers get richers**

## Preferential attachment - example

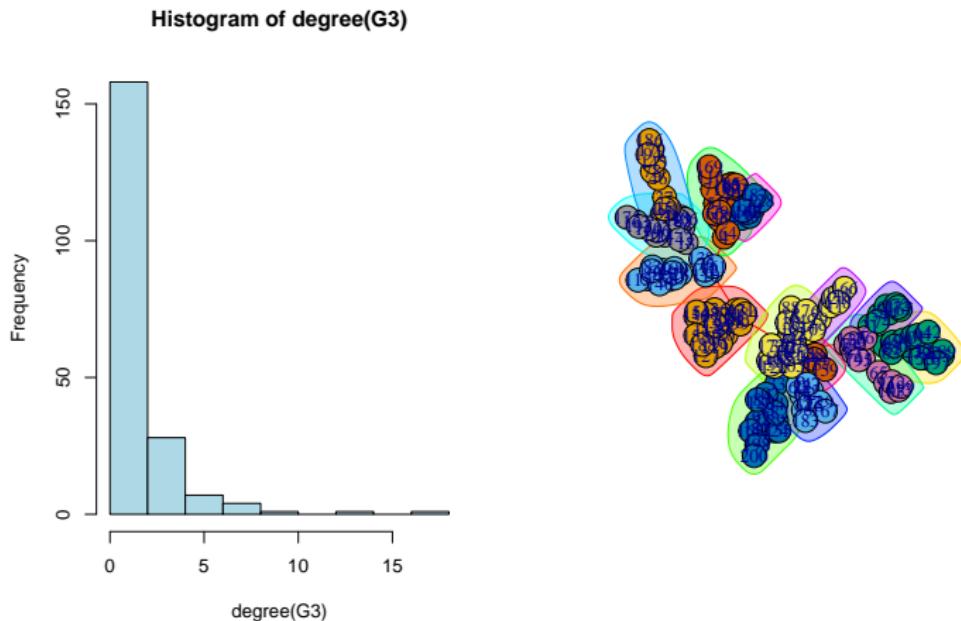
```
G1 <- igraph::sample_pa(20, 1, directed=FALSE)
G2 <- igraph::sample_pa(20, 5, directed=FALSE)
G3 <- igraph::sample_pa(200, directed=FALSE)
par(mfrow=c(1,3))
plot(G1, vertex.label=NA) ; plot(G2, vertex.label=NA)
plot(G3, vertex.label=NA, layout=layout.circle)
```



## Preferential attachment - limitations

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

```
## [1] 6.117387  
## [1] 14
```



# Limitations

- Erdős-Rényi

The ER model does not fit well real world network

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

- Preferential attachment

- Is defined through an algorithm so performing statistics is complicated
- Is stucked to the power-law distribution of degrees

## The Stochastic Block Model

The SBM<sup>1</sup> 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

---

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

# Outline

- ① Motivations
- ② Graph Partitionning
- ③ The Stochastic Block Model (SBM)

Some Graphs Models and their limitations

Mixture of Erdös-Rényi and the SBM

Statistical Inference in the SBM

SBM: some extensions

# 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

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

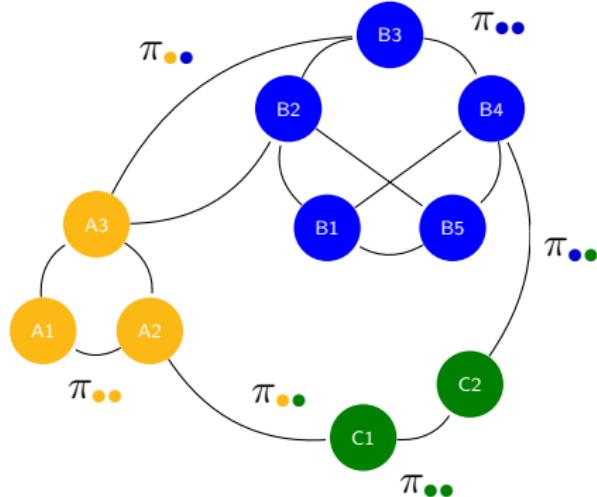
## The conditional distribution of the edges

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  $\pi$  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

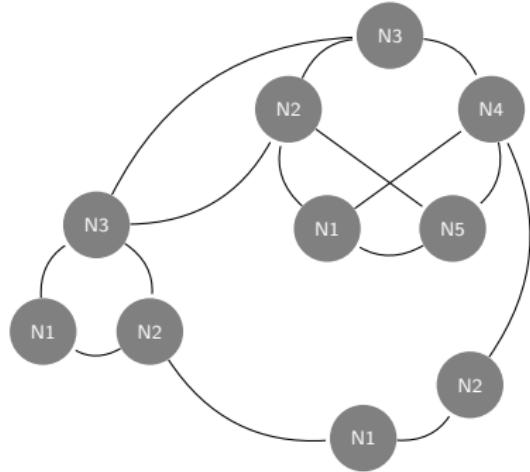
Let  $n$  nodes divided into

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

$$Z_i = \mathbf{1}_{\{i \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \alpha), \quad \forall \bullet \in \mathcal{Q},$$

$$X_{ij} \mid \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \mathcal{B}(\pi_{\bullet\bullet})$$

# Stochastic Block Model: unknown parameters



Stochastic Block Model

Let  $n$  nodes divided into

- $\mathcal{Q} = \{\bullet\text{, } \bullet\text{, } \bullet\}$ ,  $\text{card}(\mathcal{Q})$  known
- $\alpha_\bullet = ?$ ,
- $\pi_{\bullet\bullet} = ?$

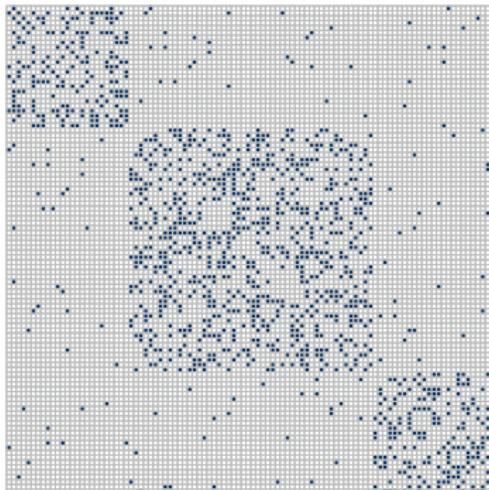
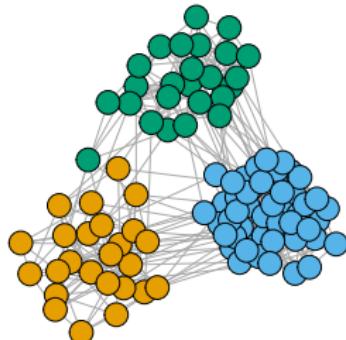
$$Z_i = \mathbf{1}_{\{i \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \alpha), \quad \forall \bullet \in \mathcal{Q},$$

$$X_{ij} \mid \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \mathcal{B}(\pi_{\bullet\bullet})$$

# Stochastic block models – examples of topology

## Community network

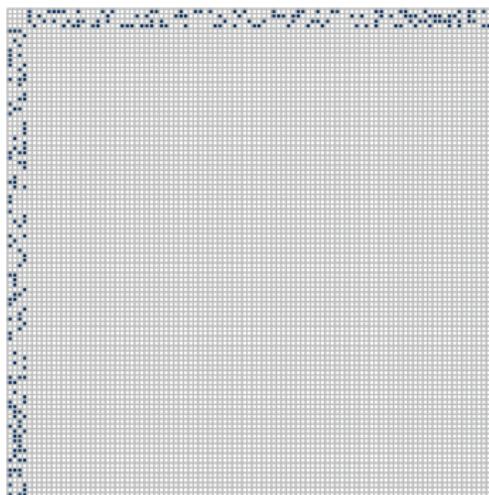
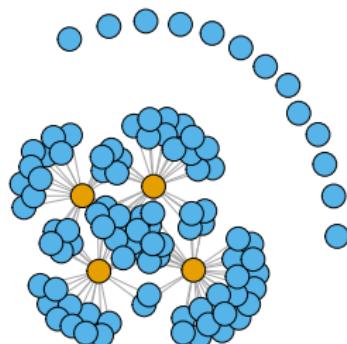
```
pi <- matrix(c(0.3,0.02,0.02,0.02,0.3,0.02,0.02,0.02,0.3),3,3)
communities <- igraph::sample_sbm(100, pi, c(25, 50, 25))
par(mfrow = c(1,2))
plot(communities, vertex.label=NA, vertex.color = rep(1:3,c(25, 50, 25)))
corrplot(as_adj(communities, sparse =FALSE), tl.pos = "n", cl.pos = 'n')
```



# Stochastic block models – examples of topology

## Star network

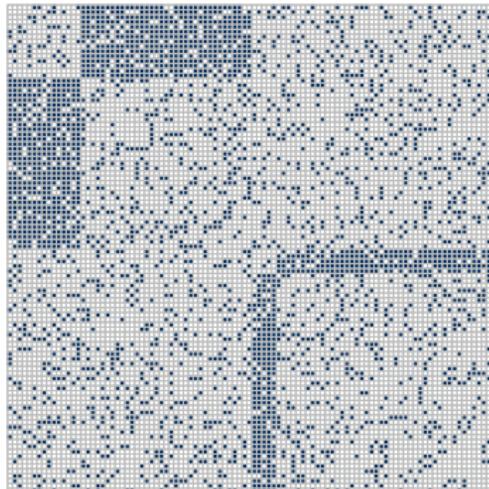
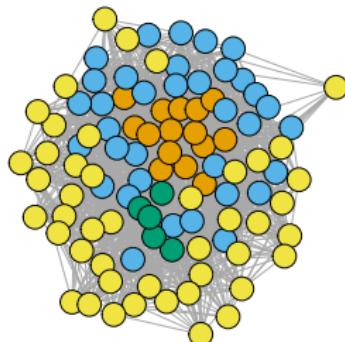
```
pi <- matrix(c(0.05,0.3,0.3,0),2,2)
star <- igraph::sample_sbm(100, pi, c(4, 96))
par(mfrow = c(1,2))
plot(star, vertex.label=NA, vertex.color = rep(1:2,c(4,96)))
corrplot(as_adj(star, sparse =FALSE), tl.pos = "n", cl.pos = 'n')
```



# Stochastic block models – examples of topology

## Bipartite network

```
pi <- matrix(c(.2,1-.2,.2,.2,1-.2,.2,.2,.2,.2,.2,.2,.2,.2, .2,1-.2,.2,.2,1-.2,.2),4,4)
bipar <- igraph::sample_sbm(100, pi, c(15, 35, 5, 45))
par(mfrow = c(1,2))
plot(bipar, vertex.label=NA, vertex.color = rep(1:4,c(15, 35, 5, 45)))
corrplot(as_adj(bipar, sparse =FALSE), tl.pos = "n", cl.pos = 'n')
```



## Degree distributions

### Conditional degree distribution

The conditional degree distribution of a node  $i \in q$  is

$$D_i | i \in q \sim b(n - 1, \bar{\pi}) \approx \mathcal{P}(\lambda_q), \quad \bar{\pi}_q = \sum_{\ell=1}^Q \alpha_\ell \pi_{q\ell}, \quad \lambda_q = (n - 1) \bar{\pi}_q$$

### Conditional degree distribution

The degree distribution of a node  $i$  can be approximated by a mixture of Poisson distributions:

$$\mathbb{P}(D_i = k) = \sum_{q=1}^Q \alpha_q \exp \{-\lambda_q\} \frac{\lambda_q^k}{k!}$$

# Outline

- ① Motivations
- ② Graph Partitionning
- ③ The Stochastic Block Model (SBM)

Some Graphs Models and their limitations

Mixture of Erdös-Rényi and the SBM

**Statistical Inference in the SBM**

SBM: some extensions

# Likelihoods

Complete data likelihood

$$\begin{aligned}\ell_c(\mathbf{Y}, \mathbf{Z}; \theta) &= p(\mathbf{Y}|\mathbf{Z}; \boldsymbol{\alpha})p(\mathbf{Z}; \boldsymbol{\pi}) = \prod_{i,j} f_{\alpha_{Z_i, Z_j}}(Y_{ij}) \times \prod_i \pi_{Z_i} \\ &= \prod_{i,j} \alpha_{Z_i, Z_j}^{Y_{ij}} (1 - \alpha_{Z_i, Z_j})^{1-Y_{ij}} \prod_i \pi_{Z_i}\end{aligned}$$

Marginal likelihood ( $\mathbf{Y}$ )

$$\log \ell(\mathbf{Y}; \theta) = \log \sum_{\mathbf{Z} \in \mathcal{Z}} \ell_c(\mathbf{Y}, \mathbf{Z}; \theta).$$

$\mathcal{Z} = \{1, \dots, K\}^n$ : impossible to compute when  $K$  and  $n$  increase.

Standard tool to maximize the likelihood when latent variables involved :  
EM algorithm.

# From EM to variational EM

## Standard EM

At iteration  $(t)$  :

- **Step E:** compute

$$Q(\theta|\theta^{(t-1)}) = \mathbb{E}_{\mathbf{Z}|\mathbf{Y},\theta^{(t-1)}} [\log \ell_c(\mathbf{Y}, \mathbf{Z}; \theta)]$$

- **Step M:**

$$\theta^{(t)} = \arg \max_{\theta} Q(\theta|\theta^{(t-1)})$$

With SBM,

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

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

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

## The EM strategy does not apply directly for SBM

Ouch: another intractability problem

- the  $Z_{iq}$  are not independent conditional on  $(X_{ij}, i < j) \dots$
- we cannot compute  $\eta_{ijq\ell} = \mathbb{P}(Z_{iq}Z_{j\ell} = 1 | \mathbf{X}) = \mathbb{E}[Z_{iq}Z_{j\ell} | \mathbf{X}]$ ,
- the conditional expectation  $Q(\theta)$ , i.e. the main EM ingredient, is intractable.

Solution: mean field approximation

Approximate  $\eta_{ijq\ell}$  by  $\tau_{iq}\tau_{j\ell}$ , i.e., assume conditional independence between  $Z_{iq}$

↔ This can be formalized in the variational framework

# Revisiting the EM algorithm I

## Proposition

Consider a distribution  $\mathbb{Q}$  for the  $\{Z_{iq}\}$ . We have

$$\log L(\boldsymbol{\theta}; \mathbf{X}) = \mathbb{E}_{\mathbb{Q}}[\log L(\boldsymbol{\theta}, \mathbf{X}, \mathbf{Z})] + \mathcal{H}(\mathbb{Q}) + \text{KL}(\mathbb{Q} \mid \mathbb{P}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})),$$

where  $\mathcal{H}$  is the entropy and  $\text{KL}(\cdot \mid \cdot)$  is the Kullback-Leibler divergence:

$$\mathcal{H}(\mathbb{Q}) = - \sum_z \mathbb{Q}(z) \log \mathbb{Q}(z) = -\mathbb{E}_{\mathbb{Q}}[\log \mathbb{Q}(Z)]$$

$$\text{KL}(\mathbb{Q} \mid \mathbb{P}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})) = \sum_z \mathbb{Q}(z) \log \frac{\mathbb{Q}(z)}{\mathbb{P}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})} = \mathbb{E}_{\mathbb{Q}} \left[ \log \frac{\mathbb{Q}(z)}{\mathbb{P}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta})} \right]$$

## Revisiting the EM algorithm II

Let

$$J(\mathbb{Q}, \boldsymbol{\theta}) \triangleq \mathbb{E}_{\mathbb{Q}} (\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})) + \mathcal{H}(\mathbb{Q})$$

The steps in the EM algorithm may be viewed as:

Expectation step : choose  $\mathbb{Q}$  to maximize  $J(\mathbb{Q}; \boldsymbol{\theta}^{(t)})$

The solution is  $\mathbb{P}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(t)})$

Maximization step : choose  $\boldsymbol{\theta}$  to maximize  $J(\mathbb{Q}^{(t)}; \boldsymbol{\theta})$

The solution maximizes  $\mathbb{E}_{\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(t)}} (\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}))$

# Variational approximation for SBM

Problem for SBM

$\mathbb{P}(\mathbf{Z}|\mathbf{X}; \boldsymbol{\theta}^{(t)})$  cannot be computed thus the E-step cannot be solved.

Idea

Choose  $\mathbb{Q}$  in a class of function so that the E-step can be solved.

Family of distribution that factorizes

We chose  $\mathbb{Q}$  the multinomial distribution so that

$$\mathbb{Q}(\mathbf{Z}) = \prod_{i=1}^n \mathbb{Q}_i(Z_i) = \prod_{i=1}^n \prod_{q=1}^Q \tau_{iq}^{Z_{iq}},$$

where  $\tau_{iq} = \mathbb{Q}_i(Z_i = q) = \mathbb{E}_{\mathbb{Q}}(Z_{iq})$ , with  $\sum_q \tau_{iq} = 1$  for all  $i = 1, \dots, n$ .

## Variational EM for SBM: the criterion

Lower bound of the loglikelihood

Since  $\mathbb{Q}$  is an approximation of  $\mathbb{P}(\mathbf{Z}|\mathbf{X})$ , the Kullback-Leibler divergence is non-negative and

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

For the SBM,

$$J(\mathbb{Q}, \boldsymbol{\theta}) = \sum_{i,q} \tau_{iq} \log \alpha_q + \sum_{i < j, q, \ell} \tau_{iq} \tau_{j\ell} \log b(X_{ij}; \pi_{q\ell}) - \sum_{i,q} \tau_{iq} \log(\tau_{iq}),$$

we optimize the loglikelihood lower bound  $J(\mathbb{Q}, \boldsymbol{\theta}) = J(\boldsymbol{\tau}, \boldsymbol{\theta})$  in  $(\boldsymbol{\tau}, \boldsymbol{\theta})$ .

## E and M steps for SBM

### Variational E-step

Maximizing  $J(\tau)$  for fixed  $\theta$ , we find a fixed-point relationship:

$$\hat{\tau}_{iq} \propto \alpha_q \prod_j \prod_\ell b(X_{ij}, \pi_{q\ell})^{\hat{\tau}_{j\ell}}$$

### M-step

Maximizing  $J(\theta)$  for fixed  $\tau$ , we find,

$$\hat{\alpha}_q = \frac{1}{n} \sum_i \hat{\tau}_{iq}, \quad \hat{\pi}_{q\ell} = \frac{\sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{j\ell} X_{ij}}{\sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{j\ell}}.$$

## Model selection

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

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

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

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

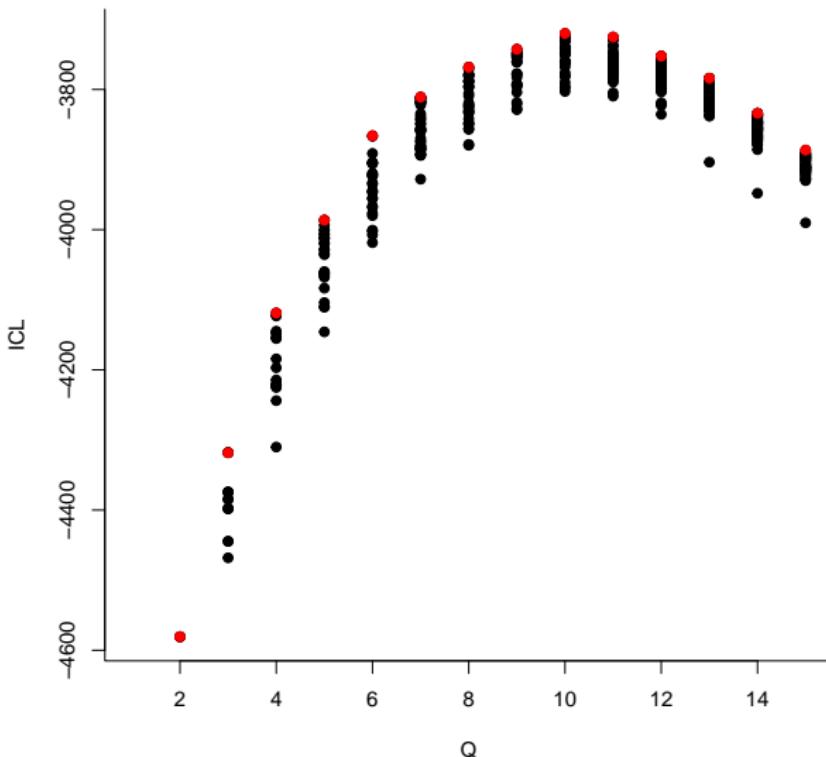
## Example: French political blogosphere

```
my_sbm <-
  blog %>% as_adj(sparse = FALSE) %>%
    sbm::estimateSimpleSBM(estimOptions = list(plot = FALSE))
```

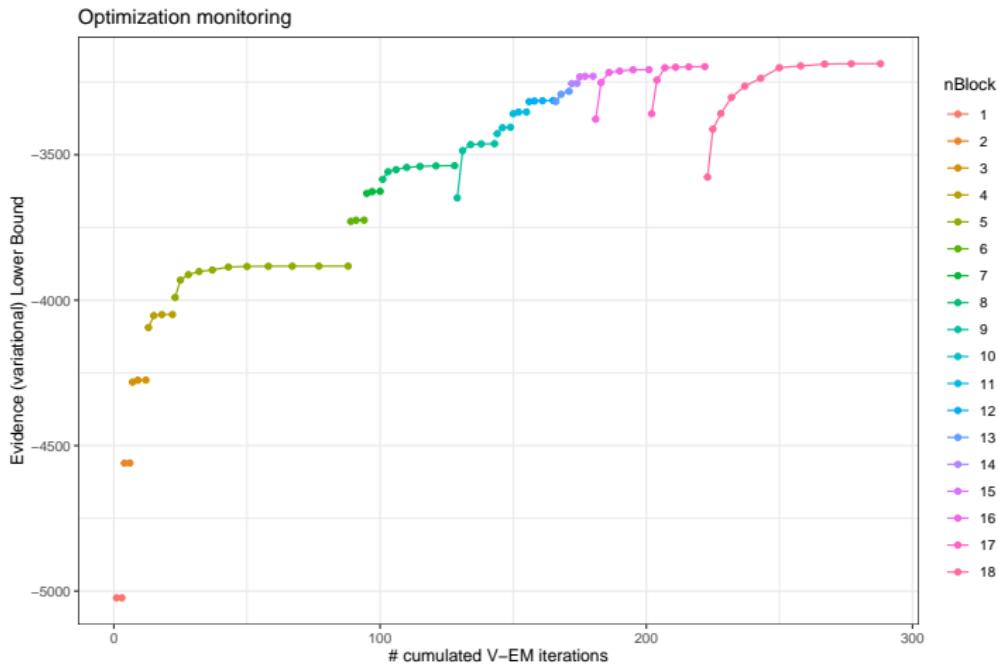
```
my_sbm
```

```
## Fit of a Simple Stochastic Block Model -- bernoulli variant
## =====
## Dimension = ( 192 ) - ( 10 ) blocks and no covariate(s).
## =====
## * Useful fields
##   $nbNodes, $modelName, $dimLabels, $nbBlocks, $nbCovariates, $nbDyads
##   $blockProp, $connectParam, $covarParam, $covarList, $covarEffect
##   $expectation, $indMemberships, $memberships
## * R6 and S3 methods
##   $rNetwork, $rMemberships, $rEdges, plot, print, coef
## * Additional fields
##   $probMemberships, $loglik, $ICL, $storedModels,
## * Additional methods
##   predict, fitted, $setModel, $reorder
```

## Example: model exploration (vICL)

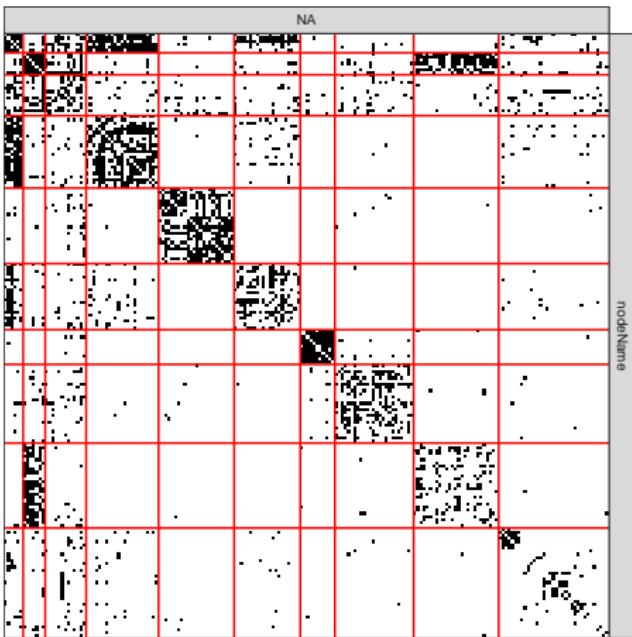


# Example: monitoring convergence (ELBO)



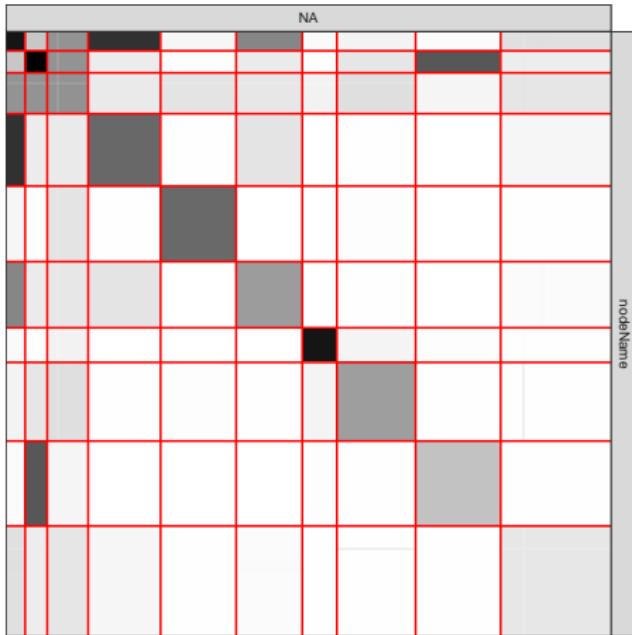
## Vizualisation: matrix view

```
plot(my_sbm, dimLabels = list(row = "blogs", col = "blogs"))
```



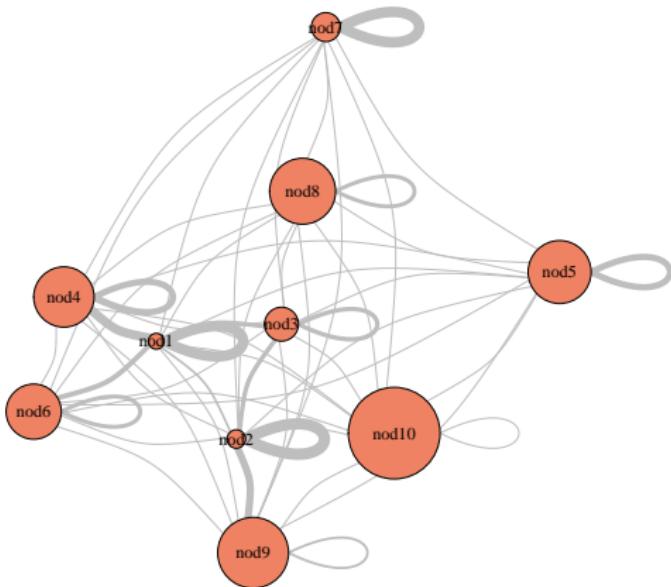
## Vizualisation: expected value

```
plot(my_sbm, "expected", dimLabels = list(row = "blogs", col = "blogs"))
```



## Vizualisation: mesoscopic view

```
plot(my_sbm, "meso")
```

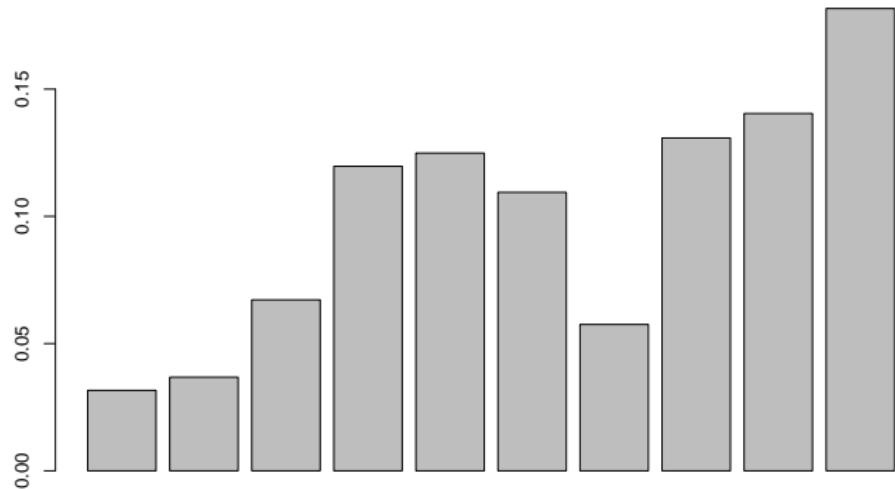


## Accessing field I

```
aricode::ARI(my_sbm$memberships, party)  
## [1] 0.4650112
```

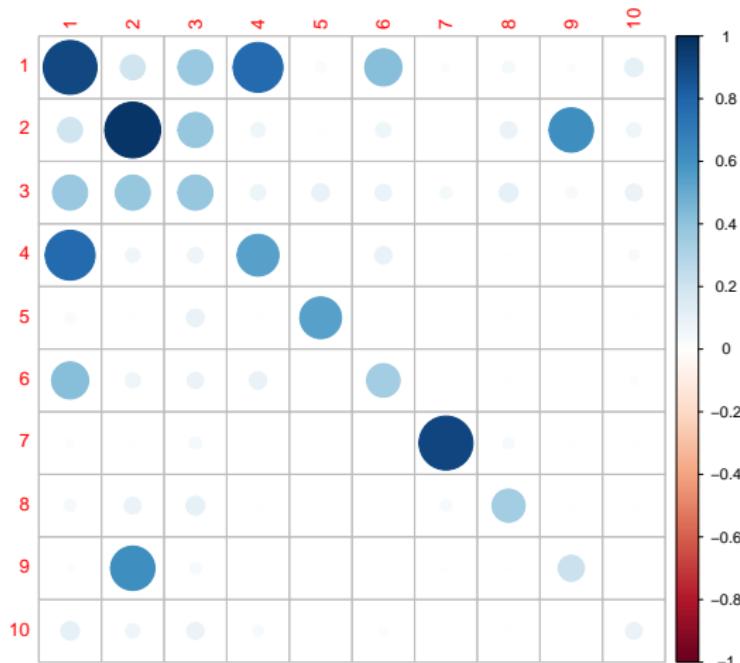
```
barplot(my_sbm$blockProp)
```

## Accessing field II



```
corrplot(my_sbm$connectParam$mean)
```

# Accessing field III



etc... see documentation and website

# Outline

- ① Motivations
- ② Graph Partitionning
- ③ The Stochastic Block Model (SBM)

Some Graphs Models and their limitations

Mixture of Erdös-Rényi and the SBM

Statistical Inference in the SBM

**SBM: some extensions**

## SBM with covariates

- As before :  $(Y_{ij})$  be an adjacency matrix
- Let  $x^{ij} \in \mathbb{R}^p$  denote covariates describing the pair  $(i, j)$

Latent variables : as before

- The nodes  $i = 1, \dots, n$  are partitioned into  $K$  clusters
- $Z_i$  independent variables  $\mathbb{P}(Z_i = k) = \pi_k$

Conditionally to  $(Z_i)_{i=1,\dots,n}$ ...

$(Y_{ij})$  independent and

$$Y_{ij}|Z_i, Z_j \sim \text{Bern}(\text{logit}(\alpha_{Z_i, Z_j} + \theta \cdot x_{ij})) \quad \text{if binary data}$$

$$Y_{ij}|Z_i, Z_j \sim \mathcal{P}(\exp(\alpha_{Z_i, Z_j} + \theta \cdot x_{ij})) \quad \text{if counting data}$$

If  $K = 1$  : all the connection heterogeneity is explained by the covariates.

# Valued-edge networks

## Values-edges networks

Information on edges can be something different from presence/absence.

It can be:

- ① a count of the number of observed interactions,
- ② a quantity interpreted as the interaction strength,

## Natural extensions of SBM and LBM

- ① Poisson distribution:  $Y_{ij} \mid \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \mathcal{P}(\lambda_{\bullet\bullet})$ ,
- ② Gaussian distribution:  $Y_{ij} \mid \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \mathcal{N}(\mu_{\bullet\bullet}, \sigma^2)$ , [?]
- ③ More generally,

$$Y_{ij} \mid \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \mathcal{F}(\theta_{\bullet\bullet})$$

# Latent Block Models aka Bipartite SBM

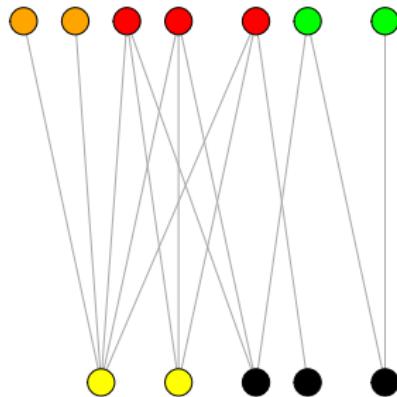
Let  $Y_{ij}$  be a bi-partite network. Individuals in row and cols are not the same.

Latent variables : bi-clustering

- Nodes  $i = 1, \dots, n_1$  partitionned into  $K_1$  clusters, nodes  $j = 1, \dots, n_2$  partitionned into  $K_2$  clusters
- $Z_i^1 = k$  if node  $i$  belongs to cluster (block)  $k$   
 $Z_j^2 = \ell$  if node  $j$  belongs to cluster (block)  $\ell$
- $Z_i^1, Z_j^2$  independent variables

$$\mathbb{P}(Z_i^1 = k) = \pi_k^1, \quad \mathbb{P}(Z_j^2 = \ell) = \pi_\ell^2$$

# Latent Block Model : illustration



## Latent Block Model

- $n_1$  row nodes  $\mathcal{K}_1 = \{\bullet, \textcolor{red}{\bullet}, \textcolor{green}{\bullet}\}$  classes
- $\pi_{\bullet}^1 = \mathbb{P}(i \in \bullet), \bullet \in \mathcal{K}_1, i = 1, \dots, n$
- $n_2$  column nodes  $\mathcal{K}_2 = \{\bullet, \textcolor{yellow}{\bullet}\}$  classes
- $\pi_{\bullet}^2 = \mathbb{P}(j \in \bullet), \bullet \in \mathcal{K}_2, j = 1, \dots, m$
- $\alpha_{\bullet\bullet} = \mathbb{P}(i \leftrightarrow j | i \in \bullet, j \in \bullet)$

$$Z_i^1 = \mathbf{1}_{\{i \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \boldsymbol{\pi}^1), \quad \forall \bullet \in \mathcal{Q}_1,$$

$$Z_j^2 = \mathbf{1}_{\{j \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \boldsymbol{\pi}^2), \quad \forall \bullet \in \mathcal{Q}_2,$$

$$Y_{ij} \mid \{i \in \bullet, j \in \bullet\} \sim^{\text{ind}} \text{Bern}(\alpha_{\bullet\bullet})$$

## To go further...

- Group GroßBM : <https://github.com/GrossSBM/sbm>;
- Documentation of package sbm:  
<https://grosssbm.github.io/sbm/>
- missSBM SBM with missing data  
<https://github.com/GrossSBM/misssbm>  
Slides : <https://grosssbm.github.io/slideshow-missSBM/slides.html>