Unsupervised Learning

Graph clustering and the Stochastic Block Model

Polytechnique MAP 573, 2019 – Julien Chiquet

Autumn semester, 2019

https://github.com/jchiquet/CourseUnsupervisedLearningX

Outline

Outline

References

Statistical Analysis of Network Data: Methods and Models,
Eric Kolazcyk
Chapter 2, Section 1

Analyse statistique de graphes,
Catherine Matias
Chapitre 1

Outline

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 $\mathcal{V} = \{1, \ldots, n\}$ of vertices or nodes
- a set $\mathcal{E} = \{e_1, \ldots, e_p : e_k = (i_k, j_k) \in (\mathcal{V} \times \mathcal{V})\}$ 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, etc.

Paths, Cycles, Connected Components

Definition (Path)

In a undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a path between $i, j \in \mathcal{V}^2$ is a series of edges $e_1, \dots, e_k$ such that

- $\forall 1 \leq \ell < k$, all edges $(e_\ell, e_{\ell+1})$ share a vertex in $\mathcal{V}$
- e_1 starts from i, e_k ends to j.

Vocabulary

- A cycle is a path from i to itself.
- A connected component is a subset $\mathcal{V}' \subset \mathcal{V}$ such that there exists an path between any $i, j \in \mathcal{V}'$.
- A graph is connected when there is a path between every node pairs.

Proposition (Decomposition)

Any graph can be decomposed in a unique set of maximal connected components. The number of connected component is a least $n - |\mathcal{E}|$

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.

Outline

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.

Incidence matrix

Definition (Incidence matrix)

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

$$(\mathbf{B})_{ij} = \begin{cases} 1 & \text{if } i \text{ is incident to edge } j, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition (Relationship)

Let $\tilde{\mathbf{B}}$ be a modified signed version of $\mathbf{B}$ where $\tilde{B}_{ij} = 1/-1$ if i is incident to j as tail/head. Then

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

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

$\rightsquigarrow$ $\tilde{\mathbf{B}}\tilde{\mathbf{B}}^\intercal$ is called the Laplacian matrix and will be studied latter.

Layout and Vizualization

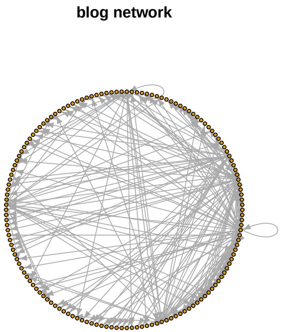
- Vizualization of large networks is a field of research in its own
- Be carefull with graphical interpretation of (large) networks

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

Layout and Vizualization

Example with circle plot

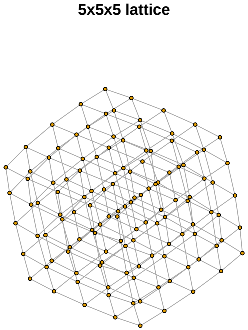
```
par(mfrow=c(1,2))
plot(GLattice, layout=layout.circle); title("5x5x5 lattice")
plot(GBlog , layout=layout.circle); title("blog network")
```

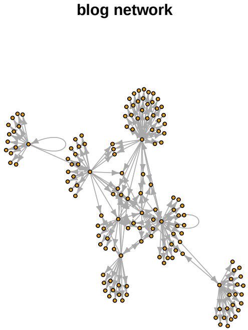


Layout and Vizualization

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")
```

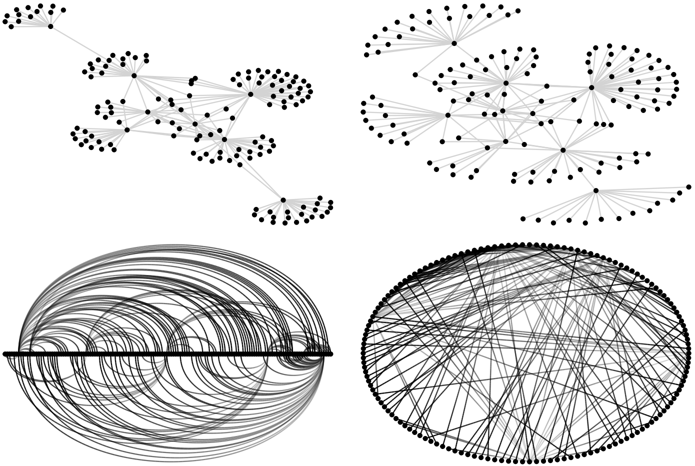




Layout and Vizualization: **ggraph** way I

```
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 Vizualization: **ggraph** way II



Outline

References

- Statistical Analysis of Network Data: Methods and Models,
  Eric Kolazcyk
  Chapter 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
- A Tutorial on Spectral Clustering,
  Ulrike von Luxburg

Principle of graph partionning

Definition (Partition)

A decomposition $\mathcal{C} = \{C_1, \ldots, C_K\}$ 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 partionning

Form a partition of the vertices with unsupervized 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

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 $\{\{1\}, \ldots, \{n\}\}$ to $\{\{1, \ldots, n\}\}$

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:

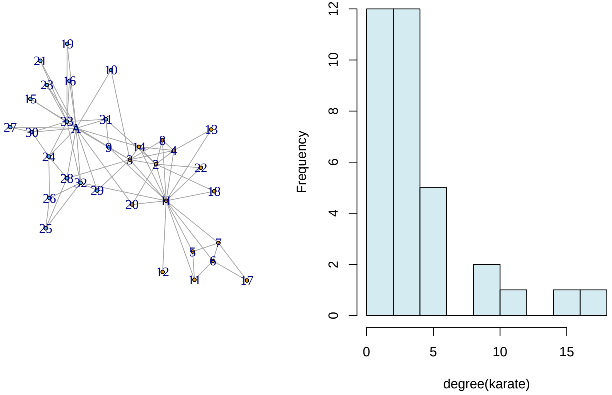
- Euclidean distance: $x_{ij} = \sqrt{\sum_{ij}(A_{ik} - A_{jk})^2}$
- Manhattan distance: $x_{ij} = \sum_{ij} |A_{ik} - A_{jk})|$
- etc. . .

Graph-specific

For instance, Modularity (studied during tutorial)

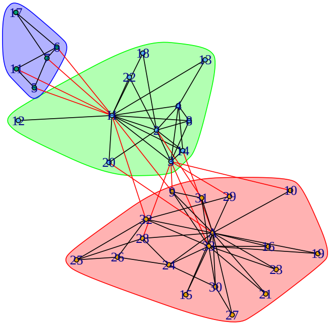
Example: karaté club

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



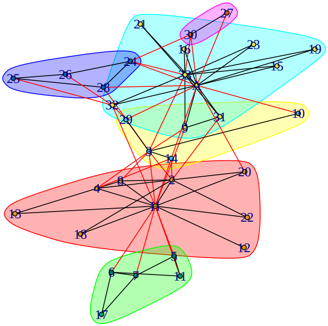
Examples of graph clustering I

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



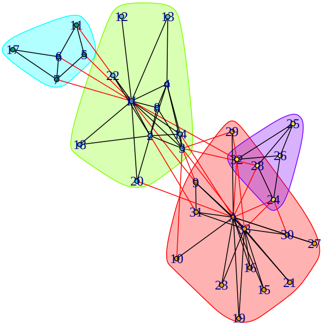
Examples of graph clustering II

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



Examples of graph clustering III

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



Outline

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

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

where $\mathbf{D} = \mathrm{diag}(d_i, i \in \mathcal{V})$ 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 $\mathcal{G}$.

Graph Laplacian: spectrum

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

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

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

- $\mathbf{L}$ *is a symmetric, positive semi-definite matrix,*
- *the smallest eigenvalue is 0 with associated eigenvector $\mathbf{1}$.*
- $\mathbf{L}$ *has n positive eigenvalues $0 = \lambda_1 < \cdots < \lambda_n$.*

Corollary (Spectrum and Graph)

- *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 eigenvalue, the higher the connectivity of $\mathcal{G}$.*

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

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

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

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 $i \in q$

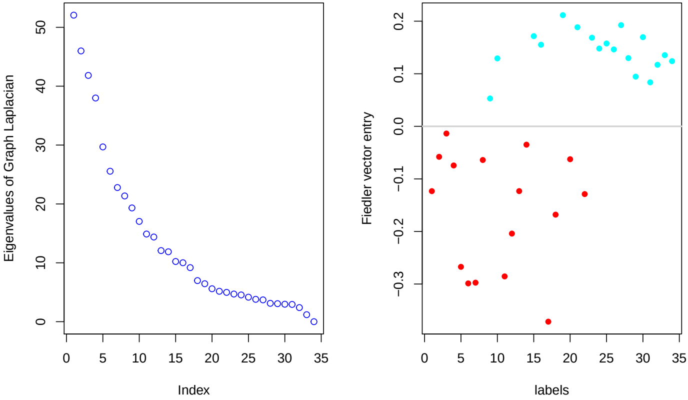
Remarks

- implemented during today's lab
- also apply to no graphical data!

Example: Karate club and Fielder vector and eigenvalue I

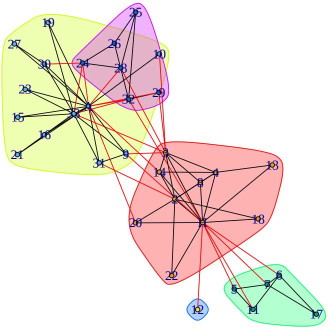
```
eigen_karate <- graph.laplacian(karate) %>% eigen()

fielder_vector <- eigen_karate$vectors[, igraph::vcount(karate) - 1]
faction <- as.character(vcount(karate))
faction[V(karate)$Faction == 1] <- "red"
faction[V(karate)$Faction == 2] <- "cyan"

par(mfrow = c(1,2))
plot(eigen_karate$values, col = "blue", ylab = "Eigenvalues of Graph Laplacian")
plot(fielder_vector, pch = 16, xlab = "labels",
   ylab = "Fiedler vector entry", col = faction)
abline(0, 0, lwd = 2, col = "lightgray")
```

Example: Karate club and Fielder vector and eigenvalue II



Clustering based on the first non null eigenvalue

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



Outline

References

- Statistical Analysis of Network Data: Methods and Models
  Eric Kolazcyk
  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

- Analyse statistique de graphes,
  Catherine Matias
  Chapitre 4, Section 4

Motivations

Last section: find an underlying organization in a observed network

Spectral or hierachical clustering for network data

$\rightsquigarrow$ 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.

$$\begin{array}{c} \text{hierarchical/kmeans clustering} \leftrightarrow \text{Gaussian mixture models} \\ \Updownarrow \\ \text{hierarchical/spectral clustering for network} \leftrightarrow \text{Stochastic block model} \end{array}$$

Outline

A mathematical model: Erdös-Rényi graph

Definition

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

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

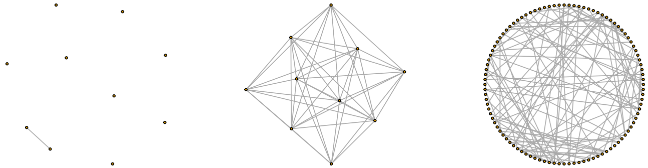
Proposition (degree distribution)

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

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

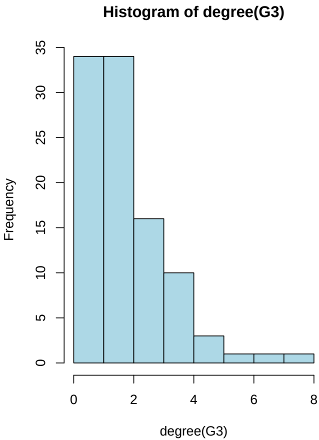
Erdös-Rényi - example

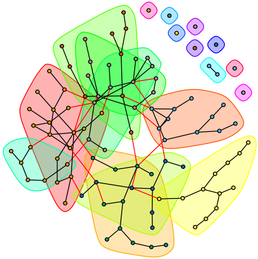
```
G1 <- igraph::sample_gnp(10, 0.1)
G2 <- igraph::sample_gnp(10, 0.9)
G3 <- igraph::sample_gnp(100, .02)
par(mfrow=c(1,3))
plot(G1, vertex.label=NA) ; plot(G2, vertex.label=NA)
plot(G3, vertex.label=NA, layout=layout.circle)
```



Erdös-Rény - limitations: very homegeneous

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

## [1] 5.834998
## [1] 14
```





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,

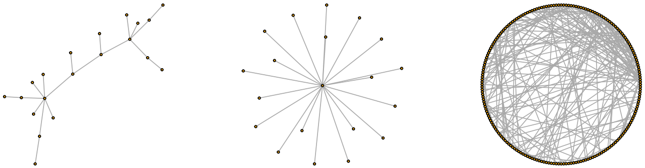
1. At t a new node V_t is added
2. V_t is connected to $i \in V_{t-1}$ with probability

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

$\rightsquigarrow$ Nodes with high degree get more connections thus richers get richers

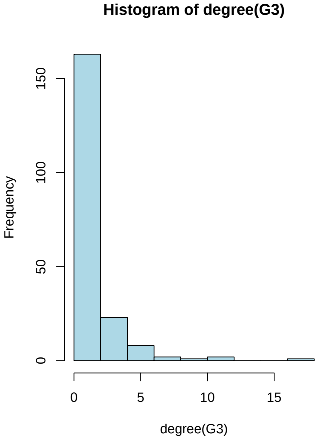
Preferential attachment - example

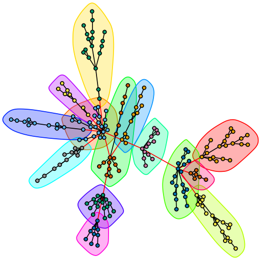
```
G1 <- igraph::sample_pa(20, 1, directed=FALSE)
G2 <- igraph::sample_pa(20, 5, directed=FALSE)
G3 <- igraph::sample_pa(200, directed=FALSE)
par(mfrow=c(1,3))
plot(G1, vertex.label=NA) ; plot(G2, vertex.label=NA)
plot(G3, vertex.label=NA, layout=layout.circle)
```



Preferential attachment - limitations

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

## [1] 6.137487
## [1] 14
```





Limitations

- Erdös-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[1] 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

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

Outline

Stochastic Block Model: definition

Mixture model point of view: mixture of Erdös-Rényi

Latent structure

Let $\mathcal{V} = \{1, .., n\}$ 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.

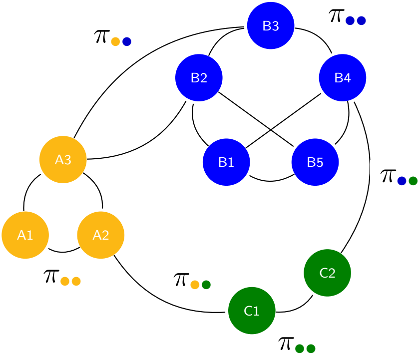
The conditional distribution of the edges

Connexion probabilities depend on the node class belonging:

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

The $Q \times Q$ matrix $\boldsymbol{\pi}$ 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

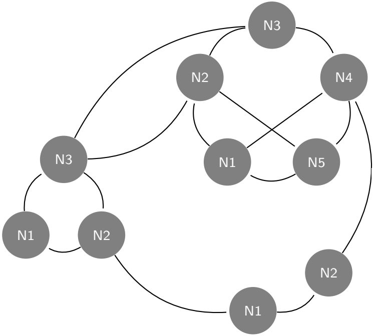
Let n nodes divided into

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

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

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

Stochastic Block Model: unknown parameters



Stochastic Block Model

Let n nodes divided into

- $\mathcal{Q} = \{\textcolor{orange}{\bullet}, \textcolor{blue}{\bullet}, \textcolor{green}{\bullet}\}$, card($\mathcal{Q}$) known
- $\alpha_{\bullet} = ?$,
- $\pi_{\textcolor{orange}{\bullet}\textcolor{blue}{\bullet}} = ?$

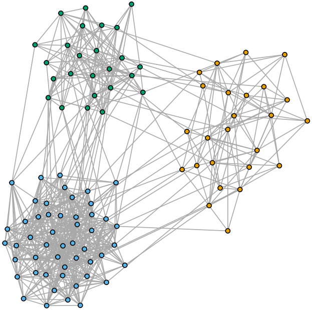
$$Z_i = \mathbf{1}_{\{i \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \alpha), \quad \forall \bullet \in \mathcal{Q},$$

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

Stochastic block models – examples of topology

Community network

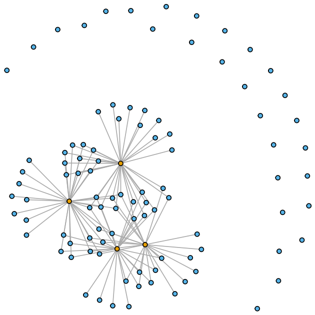
```
pi <- matrix(c(0.3,0.02,0.02,0.02,0.3,0.02,0.02,0.02,0.3),3,3)
communities <- igraph::sample_sbm(100, pi, c(25, 50, 25))
plot(communities, vertex.label=NA, vertex.color = rep(1:3,c(25, 50, 25)))
```



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)))
```



Degree distributions

Conditional degree distribution

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

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

Conditional degree distribution

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

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

Likelihoods

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

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

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

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

Outline

The EM strategy does not apply directly for SBM

Ouch: another intractability problem

- the Z_{iq} are not independent in the SBM framework...
- we cannot compute $\eta_{ijq\ell} = \mathbb{P}(Z_{iq}Z_{j\ell} = 1|\mathbf{X}) = \mathbb{E}\left[Z_{iq}Z_{j\ell}|\mathbf{X}\right]$,
- the conditional expectation $Q(\boldsymbol{\theta})$, 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 independence between Z_{iq}
$\rightsquigarrow$ This can be formalized in the variational framework

Revisting the EM algorithm I

Proposition

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

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

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

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

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

Revisting the EM algorithm II

Let

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

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

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 $\mathbb{Q}$ so as the Z_{iq} are marginally independents:

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

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

Variational EM for SBM: the criterion

Lower bound of the loglikehood

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,

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

$\rightsquigarrow$ 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 $J(\boldsymbol{\tau})$ for fixed $\boldsymbol{\theta}$, we find a fixed-point relationship:

$$\hat{\tau}_{iq} \propto \alpha_q \prod_j \prod_\ell b(X_{ij}, \pi_{q\ell})^{\hat{\tau}_{j\ell}} \tag{1}$$

M-step

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

$$\hat{\alpha}_q = \frac{1}{n} \sum_i \hat{\tau}_{iq}, \quad \hat{\pi}_{q\ell} = \frac{\sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{j\ell} X_{ij}}{\sum_{i \neq j} \hat{\tau}_{iq} \hat{\tau}_{j\ell}}. \tag{2}$$

Model selection

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

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

where

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

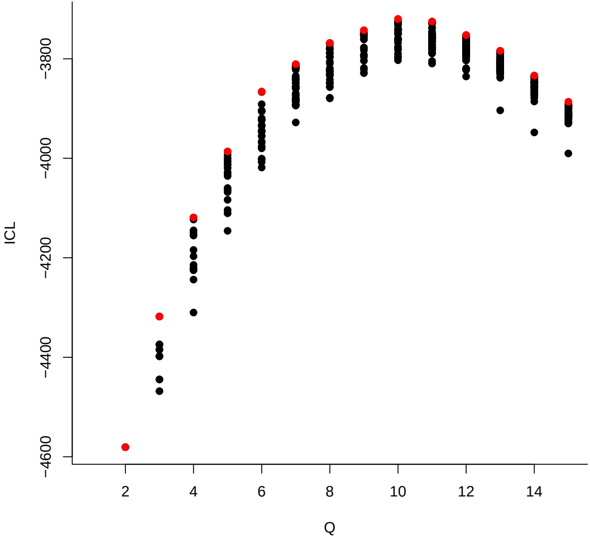
The variational BIC is just

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

Example on the French blogsphere (I)

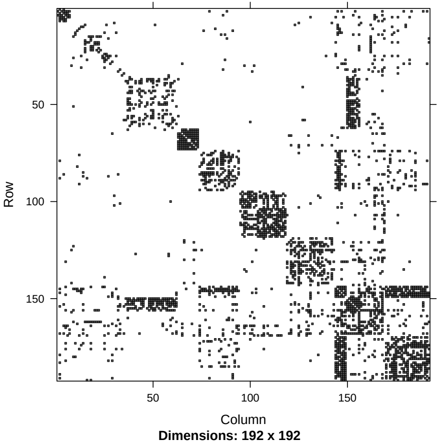
```r
library(blockmodels)
library(sand)

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

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

Example on the French blogsphere (II)



Example on the French blogsphere (III)

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



Example on the French blogsphere (IV) I

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

g <- graph_from_adjacency_matrix(
  adj_blog,
  mode = "undirected",
  weighted = TRUE,
  diag = FALSE
)
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 blogsphere (IV) II