Data anaysis and Unsupervised Learning
Clustering: model-based approaches

MAP 573, 2020 – Julien Chiquet

École Polytechnique, Autumn semester, 2020

https://jchiquet.github.io/MAP573

ACCÉLÉRATEUR & INCUBATEUR DE START-UP ÉCOLE POLYTECHNIQUE

INRA SCIENCE & IMPACT

Packages required for reproducing the slides

```r
library(tidyverse)  # opinionated collection of packages for data manipulation
library(GGally)     # extension to ggplot vizualization system
library(kernlab)    # Kernel-based methods, among which spectral-clustering
library(aricode)    # fast computation of clustering measures
library(mclust)     # gaussian mixture models
library(sbm)        # Stochastic Block Models
library(igraph)     # graph manipulation
theme_set(theme_bw()) # plots themes
```

Companion data set

Morphological Measurements on Leptograpsus Crabs

Description

The crabs data frame has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia.

```r
crabs <- MASS::crabs %>% select(-index) %>%
  rename(sex = sex,
         species         = sp,
         frontal_lob     = FL,
         rear_width      = RW,
         carapace_length = CL,
         carapace_width  = CW,
         body_depth      = BD)
crabs %>% select(sex, species) %>% summary() %>% knitr::kable("latex")
```

	sex	species
	F:100	B:100
	M:100	O:100

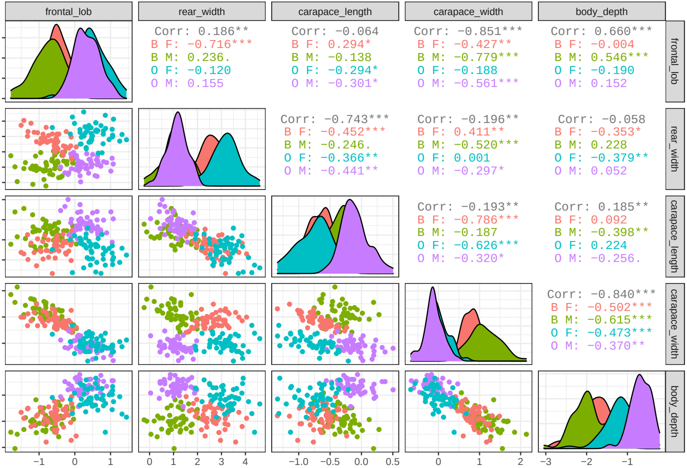
Remove size effect I

```r
attributes <- select(crabs, -sex, -species) %>% as.matrix()
u1 <- eigen(cov(attributes))$vectors[, 1, drop = FALSE]
attributes_rank1 <- attributes %*% u1 %*% t(u1)
crabs_corrected <- crabs
crabs_corrected[, 3:7] <- attributes - attributes_rank1
```

⇝ Axis 1 explains a latent effect, here the size in the case at hand, common to all attributes.

```r
ggpairs(crabs_corrected, columns = 3:7, aes(colour = paste(crabs$species, crabs$sex
```

Remove size effect II



Clustering: general goals

Objective: construct a map

$$f : \mathcal{D} = \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$$

where K is a fixed number of clusters.

Careful! classification $\neq$ clustering

- Classification presupposes the existence of classes
- Clustering labels only elements of the dataset
  - $\rightsquigarrow$ no ground truth (no given labels)
  - $\rightsquigarrow$ discovers a structure "natural" to the data
  - $\rightsquigarrow$ not necessarily related to a known classification

Motivations

- describe large masses of data in a simplified way,
- structure a set of knowledge,
- reveal structures, hidden causes,
- use of the groups in further processing,
- ...

Clustering: challenges

Clustering quality

No obvious measure to define the quality of the clusters. Ideas:

- Inner homogeneity: samples in the same group should be similar
- Outer inhomogeneity: samples in different groups should be different

Number of clusters

Choice of the number of clusters K often complex

- No ground truth in unsupervised learning!
- Several solutions might be equally good

Two general approaches

- distance-based: require a distance/dissimilarity between $\{\mathbf{x}_i\}$
- **model-based**: require assumptions on the distribution $\mathbb{P}$

Part II

Model-based method

Outline

Model-based method

1. Mixture models
    - Statistical model: latent variable
    - Expectation-Maximization algorithm
    - Example: mixture of Gaussians

2. The Stochastic Block Model (SBM)

References

- Pattern recognition and machine learning,
  Christopher Bishop
  Chapter 9: Mixture Models and EM
  http://users.isr.ist.utl.pt/~wurmd/Livros/school/

- Models with Hidden Structure with Applications in Biology and Genomics,
  Stéphane Robin
  Master MathSV Course
  https://www6.inra.fr/mia-paris/content/download/4587/42934/version/1/file/ModelsHiddenStruct-Biology.pdf

- Classification non-supervisées,
  É. Lebarbier, T. Mary-Huard
  Chapitre 3 - méthode probabiliste: le modèle de mélange
  https://www.agroparistech.fr/IMG/pdf/ClassificationNonSupervisee-AgroParisTech.pdf

Outline

Model-based method

1. Mixture models
   - Statistical model: latent variable
   - Expectation-Maximization algorithm
   - Example: mixture of Gaussians
2. The Stochastic Block Model (SBM)

Latent variable models

Definition

A latent variable model is a statistical model that relates, for $i = 1, \dots, n$ individuals,

- a set of manifest (observed) variables $\mathbf{X} = (X_i, i = 1, \dots, n)$ to
- a set of latent (unobserved) variables $\mathbf{Z} = (Z_i, i = 1, \dots, n)$.

Common assumption: conditional independence

$$\mathbb{P}((X_1, \dots, X_n)|(Z_1, \dots, Z_n)) = \prod_{i=1}^{n} \mathbb{P}(X_i|Z_i).$$

Famous examples

- $(Z_i, i \geq 1)$ is Markov chain: Markov models
- Z_i categorical and independent: mixture models

Latent variable models

Definition

A latent variable model is a statistical model that relates, for $i = 1, \dots, n$ individuals,

- a set of manifest (observed) variables $\mathbf{X} = (X_i, i = 1, \dots, n)$ to
- a set of latent (unobserved) variables $\mathbf{Z} = (Z_i, i = 1, \dots, n)$.

Common assumption: conditional independence

$$\mathbb{P}((X_1, \dots, X_n)|(Z_1, \dots, Z_n)) = \prod_{i=1}^{n} \mathbb{P}(X_i|Z_i).$$

Famous examples

- $(Z_i, i \geq 1)$ is Markov chain: Markov models
- Z_i categorical and independent: mixture models

Mixture models: the latent variables

When $(Z_1, \dots, Z_n)$ are independent categorical variables, they give a natural (latent) classification of the observations $(X_1, \dots, X_n)$ – or labels.

Notations

Let $(Z_1, \dots, Z_n)$ be *iid* categorical variables with distribution

$$\mathbb{P}(i \in q) = \mathbb{P}(Z_i = q) = \alpha_q, \quad \text{s.t.} \sum_{q=1}^{Q} \alpha_q = 1.$$

Alternative (equivalent) notation

Let $Z_i = (Z_{i1}, \dots, Z_{iq})$ be an indicator vector of label for i:

$$\mathbb{P}(i \in q) = \mathbb{P}(Z_{iq} = 1) = \alpha_q, \quad \text{s.t.} \sum_{q=1}^{Q} \alpha_q = 1.$$

By definition, $Z_i \sim \mathcal{M}(1, \boldsymbol{\alpha})$, with $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_Q)$.

Mixture models: the latent variables

When $(Z_1, \dots, Z_n)$ are independent categorical variables, they give a natural (latent) classification of the observations $(X_1, \dots, X_n)$ – or labels.

Notations

Let $(Z_1, \dots, Z_n)$ be *iid* categorical variables with distribution

$$\mathbb{P}(i \in q) = \mathbb{P}(Z_i = q) = \alpha_q, \quad \text{s.t.} \sum_{q=1}^{Q} \alpha_q = 1.$$

Alternative (equivalent) notation

Let $Z_i = (Z_{i1}, \dots, Z_{iq})$ be an indicator vector of label for i:

$$\mathbb{P}(i \in q) = \mathbb{P}(Z_{iq} = 1) = \alpha_q, \quad \text{s.t.} \sum_{q=1}^{Q} \alpha_q = 1.$$

By definition, $Z_i \sim \mathcal{M}(1, \boldsymbol{\alpha})$, with $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_Q)$.

Mixture models: the latent variables

When $(Z_1, \dots, Z_n)$ are independent categorical variables, they give a natural (latent) classification of the observations $(X_1, \dots, X_n)$ – or labels.

Notations

Let $(Z_1, \dots, Z_n)$ be *iid* categorical variables with distribution

$$\mathbb{P}(i \in q) = \mathbb{P}(Z_i = q) = \alpha_q, \quad \text{s.t.} \sum_{q=1}^{Q} \alpha_q = 1.$$

Alternative (equivalent) notation

Let $Z_i = (Z_{i1}, \dots, Z_{iq})$ be an indicator vector of label for i:

$$\mathbb{P}(i \in q) = \mathbb{P}(Z_{iq} = 1) = \alpha_q, \quad \text{s.t.} \sum_{q=1}^{Q} \alpha_q = 1.$$

By definition, $Z_i \sim \mathcal{M}(1, \boldsymbol{\alpha})$, with $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_Q)$.

Mixture models: the manifest variables

A mixture model represents the presence of subpopulations within an overall population as follows:

$$\mathbb{P}(X_i) = \sum_{z_i \in \mathcal{Z}_i} \mathbb{P}(X_i, Z_i) = \sum_{Z_i \in \mathcal{Z}_i} \mathbb{P}(X_i | Z_i)\mathbb{P}(Z_i).$$

Conditional distribution of the manifest variables

We assume a parametric distribution of X in each subpopulation

$$X_i | \{Z_i = q\} \sim \mathbb{P}_{\theta_q} \qquad \left(\Leftrightarrow X_i | \{Z_{iq}\} = 1 \sim \mathbb{P}_{\theta_q} \right)$$

The specificity of each class is handled by $\{\boldsymbol{\theta}_q\}_{q=1}^{Q}$.

Mixture models: likelihoods

The complete-data likelihood

It is the join distribution of (X_i, Z_i):

$$\mathbb{P}(X_i, Z_i) = \alpha_{Z_i} \mathbb{P}_{\boldsymbol{\theta}_{Z_i}}(X_i)$$

The incomplete-data likelihood

It is the marginal distribution of X_i once Z_i integrated:

$$\mathbb{P}(X_i) = \sum_{q=1}^{Q} \mathbb{P}(X_i, Z_i = q) = \sum_{q=1}^{Q} \alpha_q \mathbb{P}_{\boldsymbol{\theta}_q}(X_i)$$

$\rightsquigarrow$ A mixture model is a sum of distributions weigthed by the proportion of each subpopulation.

Mixture models: likelihoods

The complete-data likelihood

It is the join distribution of (X_i, Z_i):

$$\mathbb{P}(X_i, Z_i) = \alpha_{Z_i} \mathbb{P}_{\boldsymbol{\theta}_{Z_i}}(X_i)$$

The incomplete-data likelihood

It is the marginal distribution of X_i once Z_i integrated:

$$\mathbb{P}(X_i) = \sum_{q=1}^{Q} \mathbb{P}(X_i, Z_i = q) = \sum_{q=1}^{Q} \alpha_q \mathbb{P}_{\boldsymbol{\theta}_q}(X_i)$$

$\rightsquigarrow$ A mixture model is a sum of distributions weigthed by the proportion of each subpopulation.

Outline

Model-based method

1. Mixture models
    - Statistical model: latent variable
    - Expectation-Maximization algorithm
    - Example: mixture of Gaussians
2. The Stochastic Block Model (SBM)

Intractability of the Likelihood

Maximum Likelihood Estimator

The MLE aims to maximize the (marginal) likehood of the observations:

$$L(\boldsymbol{\theta}; \mathbf{X}) = \mathbb{P}_{\boldsymbol{\theta}}((X_1, \dots, X_n)) = \int_{\mathbf{Z} \in \mathcal{Z}} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}) \mathrm{d}\mathbf{Z}$$

Integrations are summation over $\{1, \dots, Q\}$: we have Q^n terms !

Intractable summation

With mixture models, for $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_Q)$ we have

$$\log L(\boldsymbol{\theta}; \mathbf{X}) = \sum_{i=1}^{n} \log \left\{ \sum_{q=1}^{Q} \alpha_q \mathbb{P}_{\boldsymbol{\theta}_q}(X_i) \right\}.$$

$\rightsquigarrow$ Direct maximization of the likelihood is impossible in practice

Intractability of the Likelihood

Maximum Likelihood Estimator

The MLE aims to maximize the (marginal) likehood of the observations:

$$L(\boldsymbol{\theta}; \mathbf{X}) = \mathbb{P}_{\boldsymbol{\theta}}((X_1, \dots, X_n)) = \int_{\mathbf{Z} \in \mathcal{Z}} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}) \mathrm{d}\mathbf{Z}$$

Integrations are summation over $\{1, \dots, Q\}$: we have Q^n terms !

Intractable summation

With mixture models, for $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_Q)$ we have

$$\log L(\boldsymbol{\theta}; \mathbf{X}) = \sum_{i=1}^{n} \log \left\{ \sum_{q=1}^{Q} \alpha_q \mathbb{P}_{\boldsymbol{\theta}_q}(X_i) \right\}.$$

$\leadsto$ Direct maximization of the likelihood is impossible in practice

Bayes decision rule / Maximum *a posteriori*

Principle

Affect an individual i to the subpopulation which is the most likely according to the data:

$$\tau_{iq} = \mathbb{P}(Z_{iq} = 1 | X_i = x_i)$$

This is the posterior probability for $i \in q$.

Application of the Bayes Theorem

It is straightforward to show that

$$\tau_{iq} = \frac{\alpha_q \mathbb{P}_{\theta_q}(x_i)}{\sum_{q=1}^{Q} \alpha_q \mathbb{P}_{\theta_q}(x_i)}$$

Principle of the EM algorithm

If $\boldsymbol{\theta}$ were known

. . . estimating the posterior probability $\mathbb{P}(Z_i|\mathbf{X})$ of $\mathbf{Z}$ should be easy
By means of the Bayes decision rule

If $\mathbf{Z}$ were known. . .

. . . estimating the best set of parameter $\boldsymbol{\theta}$ should be easy
This is close to usual maximum likelihood estimation

EM principle

Maximize the marginal likelihood iteratively:

1. Initialize θ
2. Compute the probability of $\mathbf{Z}$ given θ
3. Get a better θ with the new $\mathbf{Z}$
4. Iterate until convergence

Principle of the EM algorithm

If $\boldsymbol{\theta}$ were known

…estimating the posterior probability $\mathbb{P}(Z_i|\mathbf{X})$ of $\mathbf{Z}$ should be easy
By means of the Bayes decision rule

If $\mathbf{Z}$ were known…

…estimating the best set of parameter $\boldsymbol{\theta}$ should be easy
This is close to usual maximum likelihood estimation

EM principle

Maximize the marginal likelihood iteratively:

1. Initialize $\boldsymbol{\theta}$
2. Compute the probability of $\mathbf{Z}$ given $\boldsymbol{\theta}$
3. Get a better $\boldsymbol{\theta}$ with the new $\mathbf{Z}$
4. Iterate until convergence

Formal algorithm

Initialization: start from a good guess either of $\mathbf{Z}$ or $\boldsymbol{\theta}$, then iterate 1-2

1. Expectation step

Calculate the expected value of the loglikelihood under the current $\boldsymbol{\theta}$

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right) = \mathbb{E}_{\mathbf{Z}|\mathbf{X};\boldsymbol{\theta}^{(t)}}\left[\log L(\boldsymbol{\theta};\mathbf{X},\mathbf{Z})\right] \qquad (\textit{needs } \mathbb{P}_{\boldsymbol{\theta}^{(t)}}(\mathbf{Z}|\mathbf{X}))$$

2. Maximization step

Find the parameters that maximize this quantity

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}\right)$$

Stop when $\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\| < \varepsilon$ or $\|Q^{(t+1)} - Q^{(t)}\| < \varepsilon$

(Basic) Convergence analysis

Theorem

At each step of the EM algorithm, the loglikelihood increases. EM thus reaches a local optimum.

Proof.

On board. □

Choosing the number of component

Reminder: Bayesian Information Criterion

The BIC is a model selection criterion which penalizes the adjustement to the data by the number of parameter in model $\mathcal{M}$ as follows:

$$\mathrm{BIC}(\mathcal{M}) = \log L(\hat{\boldsymbol{\theta}}; \mathbf{X}) - \frac{1}{2} \log(n) \mathrm{df}(\mathcal{M}).$$

Integrated Classification Criterion

It is an adaptation working with the complete-data likelihood:

$$\begin{aligned} \mathrm{ICL}(\mathcal{M}) &= \log L(\hat{\boldsymbol{\theta}}; \mathbf{X}, \hat{\mathbf{Z}}) + \frac{1}{2} \log(n) \mathrm{df}(\mathcal{M}) \\ &= \mathrm{BIC} - \mathcal{H}(\mathbb{P}(\hat{\mathbf{Z}} | \mathbf{X}), \end{aligned}$$

where the entropy $\mathcal{H}$ measures the separability of the subpopulations.

$\leadsto$ We choose $\mathcal{M}(Q)$ that maximizes either BIC or ICL

Choosing the number of component

Reminder: Bayesian Information Criterion

The BIC is a model selection criterion which penalizes the adjustement to the data by the number of parameter in model $\mathcal{M}$ as follows:

$$\mathrm{BIC}(\mathcal{M}) = \log L(\hat{\boldsymbol{\theta}}; \mathbf{X}) - \frac{1}{2}\log(n)\mathrm{df}(\mathcal{M}).$$

Integrated Classification Criterion

It is an adaptation working with the complete-data likelihood:

$$\begin{aligned} \mathrm{ICL}(\mathcal{M}) &= \log L(\hat{\boldsymbol{\theta}}; \mathbf{X}, \hat{\mathbf{Z}}) + \frac{1}{2}\log(n)\mathrm{df}(\mathcal{M}) \\ &= \mathrm{BIC} - \mathcal{H}(\mathbb{P}(\hat{\mathbf{Z}}|\mathbf{X}), \end{aligned}$$

where the entropy $\mathcal{H}$ measures the separability of the subpopulations.

⇝ We choose $\mathcal{M}(Q)$ that maximizes either BIC or ICL

Choosing the number of component

Reminder: Bayesian Information Criterion

The BIC is a model selection criterion which penalizes the adjustement to the data by the number of parameter in model $\mathcal{M}$ as follows:

$$\mathrm{BIC}(\mathcal{M}) = \log L(\hat{\boldsymbol{\theta}}; \mathbf{X}) - \frac{1}{2}\log(n)\mathrm{df}(\mathcal{M}).$$

Integrated Classification Criterion

It is an adaptation working with the complete-data likelihood:

$$\begin{aligned}\mathrm{ICL}(\mathcal{M}) &= \log L(\hat{\boldsymbol{\theta}}; \mathbf{X}, \hat{\mathbf{Z}}) + \frac{1}{2}\log(n)\mathrm{df}(\mathcal{M}) \\ &= \mathrm{BIC} - \mathcal{H}(\mathbb{P}(\hat{\mathbf{Z}}|\mathbf{X}),\end{aligned}$$

where the entropy $\mathcal{H}$ measures the separability of the subpopulations.

⇝ We choose $\mathcal{M}(Q)$ that maximizes either BIC or ICL

Outline

Model-based method

1. Mixture models
   - Statistical model: latent variable
   - Expectation-Maximization algorithm
   - Example: mixture of Gaussians
2. The Stochastic Block Model (SBM)

Popular model: Gaussian Multivariate mixture models

The distribution of X_i conditional on the label of i is assumed to be a multivariate Gaussian distribution with unknown parameters:

$$X_i | i \in q \sim \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$$

Complete Likelihood $(\mathbf{X}, \mathbf{Z})$

The model complete loglikelihood is

$$\log L(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \mathbf{X}, \mathbf{Z}) = \sum_{i=1}^{n} \sum_{q=1}^{Q} Z_{iq} \left(\log \alpha_q - \frac{1}{2} \log \det(\boldsymbol{\Sigma}_q) - \frac{1}{2} \|\mathbf{x}_i - \boldsymbol{\mu}_q\|^2_{\boldsymbol{\Sigma}_q^{-1}}) \right) + c$$

$\leadsto$ Implementation of the univariate case during the labs.

Mixture of Gaussians

Calculs in the univariate case: complete likelihood

The distribution of X_i conditional on the label of i is assumed to be a univariate Gaussian distribution with unknown parameters:

$$X_i | Z_{iq} = 1 \sim \mathcal{N}(\mu_q, \sigma_q^2)$$

complete Likelihood $(\mathbf{X}, \mathbf{Z})$

The model complete loglikelihood is

$$\log L(\boldsymbol{\mu}, \boldsymbol{\sigma}^2; \mathbf{X}, \mathbf{Z}) = \sum_{i=1}^{n} \sum_{q=1}^{Q} Z_{iq} \left(\log \alpha_q - \log \sigma_q - \log(\sqrt{2\pi}) - \frac{1}{2\sigma_q^2}(x_i - \mu_q)^2 \right)$$

Gaussian mixture model in `R` I

The package `Mclust` is a great reference

See `https://cran.r-project.org/web/packages/mclust/vignettes/mclust.html`

Gaussian mixture model in R II

```
GMM <- crabs_corrected %>%
  select(-sex, -species) %>%
  Mclust(modelNames = c("EII", "EEI"))

## Error in Mclust(., modelNames = c("EII", "EEI")):  could not find function
"Mclust"

plot(GMM, 'BIC')

## Error in h(simpleError(msg, call)):  error in evaluating the argument ’x’ in
selecting a method for function ’plot’:  object ’GMM’ not found

aricode::ARI(GMM$classification, classes)

## Error in sortPairs(c1, c2):  object ’GMM’ not found

aricode::ARI(GMM$classification, clusters)

## Error in sortPairs(c1, c2):  object ’GMM’ not found

aricode::ARI(GMM$classification, clusters_ward)

## Error in sortPairs(c1, c2):  object ’GMM’ not found

plot(GMM, 'classification')
```

Mixture of Gaussians

Calculs in the univariate case: complete likelihood

The distribution of X_i conditional on the label of i is assumed to be a univariate Gaussian distribution with unknown parameters:

$$X_i | Z_{iq} = 1 \sim \mathcal{N}(\mu_q, \sigma_q^2)$$

complete Likelihood $(\mathbf{X}, \mathbf{Z})$

The model complete loglikelihood is

$$\log L(\boldsymbol{\mu}, \boldsymbol{\sigma}^2; \mathbf{X}, \mathbf{Z}) = \sum_{i=1}^{n} \sum_{q=1}^{Q} Z_{iq} \left(\log \alpha_q - \log \sigma_q - \log(\sqrt{2\pi}) - \frac{1}{2\sigma_q^2} (x_i - \mu_q)^2 \right)$$

Mixture of Gaussians

Calculs in the univariate case: E-step

E-step

For fixed values of μ_q, σ_q^2 and α_q, the estimates of the posterior probabilities $\hat{\tau}_{iq} = \mathbb{P}(Z_{iq} = 1 | X_i)$ are

$$\hat{\tau}_{iq} = \frac{\alpha_q \mathcal{N}(x_i; \mu_q, \sigma_q^2)}{\sum_{q=1}^{Q} \alpha_q \mathcal{N}(x_i; \mu_q, \sigma_q^2)},$$

where $\mathcal{N}$ is the density of the normal distribution.

Mixture of Gaussians

Calculs in the univariate case: M-step

M-step

For fixed values of τ_{iq}, the estimates of the model parameters are

$$\hat{\alpha}_q = \frac{\sum_{i=1}^n \tau_{iq}}{\sum_{i=1}^n \sum_{q=1}^Q \tau_{iq}} \quad \hat{\mu}_q = \frac{\sum_i \tau_{iq} x_i}{\sum_i \tau_{iq}} \quad \hat{\sigma}_q^2 = \frac{\sum_{i=1}^n \tau_{iq}(x_i - \mu_q)^2}{\sum_{i=1}^n \tau_{iq}}$$

R code: auxiliary functions

We start by defining functions to compute the complete model loglikelihood, perform the E step and the M step.

```r
get.cloglik <- function(X, Z, theta) {
  alpha <- theta$alpha; mu <- theta$mu; sigma <- theta$sigma
  xs <- scale(matrix(X,length(x),length(alpha)),mu,sigma)
  return(sum(Z*(log(alpha)-log(sigma)-.5*(log(2*pi)+xs^2))))
}

M.step <- function(X, tau) {
  n <- length(X); Q <- ncol(tau)
  alpha  <- colMeans(tau)
  mu     <- colMeans(tau * matrix(X,n,Q)) / alpha
  sigma  <- sqrt(colMeans(tau*sweep(matrix(X,n,Q),2,mu,"-")^2)/alpha)
  return(list(alpha=alpha, mu=mu, sigma=sigma))
}

E.step <- function(X, theta) {
  tau <- mapply(function(alpha, mu, sigma) {
      alpha*dnorm(X,mu,sigma)
    }, theta$alpha, theta$mu, theta$sigma)
  return(tau / rowSums(tau))
}
```

R code: EM for univariate mixture

```r
EM.mixture <- function(X, Q,
                       init.cl=sample(1:Q,n,rep=TRUE), max.iter=100, eps=1e-5) {
    n <- length(X); tau <- matrix(0,n,Q); tau[cbind(1:n,init.cl)] <- 1
    Eloglik <- vector("numeric", max.iter)
    iter <- 0; cond <- FALSE

    while (!cond) {
        iter <- iter + 1
        ## M step
        theta <- M.step(X, tau)
        ## E step
        tau <- E.step(X, theta)
        ## check consistency
        Eloglik[iter] <- get.cloglik(X, tau, theta)
        if (iter > 1)
            cond <- (iter>=max.iter) | Eloglik[iter]-Eloglik[iter-1] < eps
    }

    return(list(alpha = theta$alpha,  mu = theta$mu,  sigma = theta$sigma,
                tau   = tau, cl = apply(tau, 1, which.max),
                Eloglik = Eloglik[1:iter]))
}
```

Example: data generation

We first generate data with 4 components:

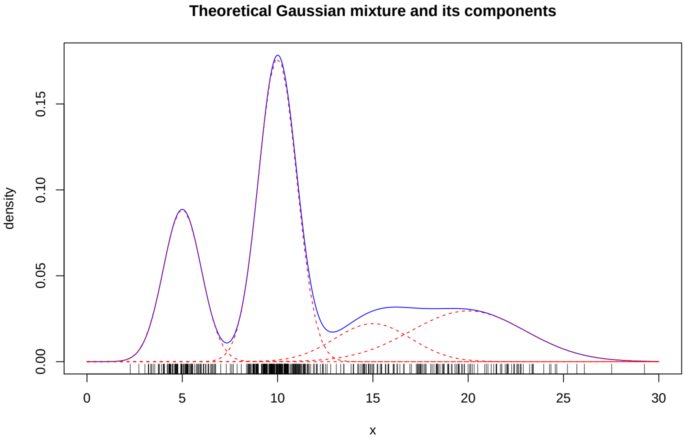
```r
mu1 <- 5   ; sigma1 <- 1; n1 <- 100
mu2 <- 10  ; sigma2 <- 1; n2 <- 200
mu3 <- 15  ; sigma3 <- 2; n3 <- 50
mu4 <- 20  ; sigma4 <- 3; n4 <- 100
cl <- rep(1:4,c(n1,n2,n3,n4))
x <- c(rnorm(n1,mu1,sigma1),rnorm(n2,mu2,sigma2),
       rnorm(n3,mu3,sigma3),rnorm(n4,mu4,sigma4))
n <- length(x)

## we randomize the class ordering
rnd <- sample(1:n)
cl <- cl[rnd]
x  <- x[rnd]

alpha <- c(n1,n2,n3,n4)/n
```

Example: data generation - plot I

Let us plot the data and the theoretical mixture.

```
curve(alpha[1]*dnorm(x,mu1,sigma1) +
      alpha[2]*dnorm(x,mu2,sigma2) +
      alpha[3]*dnorm(x,mu3,sigma3) +
      alpha[4]*dnorm(x,mu4,sigma4),
      col="blue", lty=1, from=0,to=30, n=1000,
      main="Theoretical Gaussian mixture and its components",
      xlab="x", ylab="density")
curve(alpha[1]*dnorm(x,mu1,sigma1), col="red", add=TRUE, lty=2)
curve(alpha[2]*dnorm(x,mu2,sigma2), col="red", add=TRUE, lty=2)
curve(alpha[3]*dnorm(x,mu3,sigma3), col="red", add=TRUE, lty=2)
curve(alpha[4]*dnorm(x,mu4,sigma4), col="red", add=TRUE, lty=2)
rug(x)
```

Example: data generation - plot II



Outline

Model-based method

1. Mixture models

2. The Stochastic Block Model (SBM)
   - Some Graphs Models and their limitations
   - Mixture of Erdös-Rényi and the SBM
   - Inference in SBM with variational EM

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.

hierarchical/kmeans clustering ↔ Gaussian mixture models

⇕

hierarchical/spectral clustering for network ↔ Stochastic block model

Outline

Model-based method

1. Mixture models

2. The Stochastic Block Model (SBM)
   - Some Graphs Models and their limitations
   - Mixture of Erdös-Rényi and the SBM
   - Inference in SBM with variational EM

A mathematical model: Erdös-Rényi graph

Definition

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

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

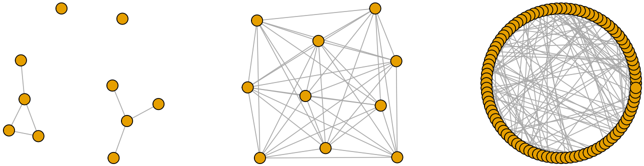
Proposition (degree distribution)

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

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

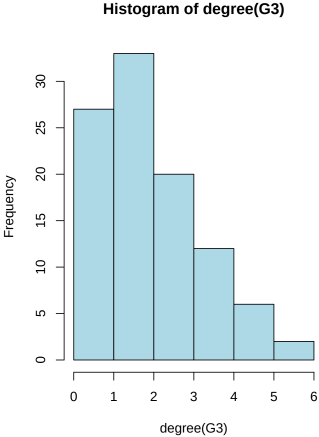
Erdös-Rényi - example

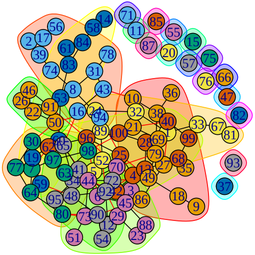
```r
G1 <- igraph::sample_gnp(10, 0.1)
G2 <- igraph::sample_gnp(10, 0.9)
G3 <- igraph::sample_gnp(100, .02)
par(mfrow=c(1,3))
plot(G1, vertex.label=NA) ; plot(G2, vertex.label=NA)
plot(G3, vertex.label=NA, layout=layout.circle)
```



Erdös-Rény - limitations: very homegeneous

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

## [1] 4.859664
## [1] 13
```





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,

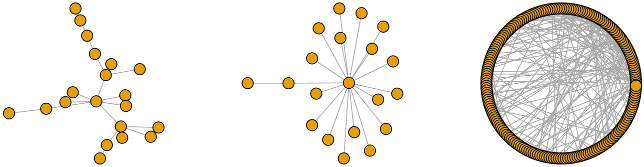
1. At t a new node V_t is added
2. V_t is connected to $i \in V_{t-1}$ with probability

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

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

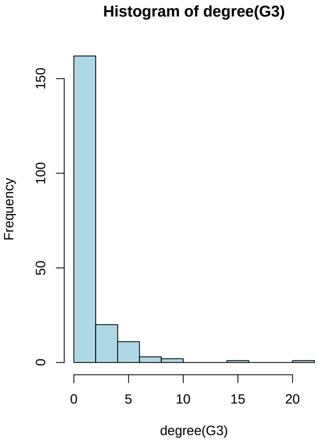
Preferential attachment - example

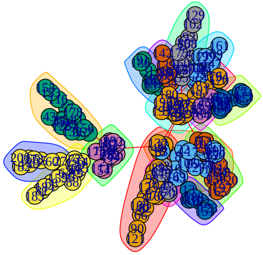
```r
G1 <- igraph::sample_pa(20, 1, directed=FALSE)
G2 <- igraph::sample_pa(20, 5, directed=FALSE)
G3 <- igraph::sample_pa(200, directed=FALSE)
par(mfrow=c(1,3))
plot(G1, vertex.label=NA) ; plot(G2, vertex.label=NA)
plot(G3, vertex.label=NA, layout=layout.circle)
```



Preferential attachment - limitations

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

## [1] 5.520854
## [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

Model-based method

1. Mixture models

2. The Stochastic Block Model (SBM)
   - Some Graphs Models and their limitations
   - Mixture of Erdös-Rényi and the SBM
   - Inference in SBM with variational EM

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.

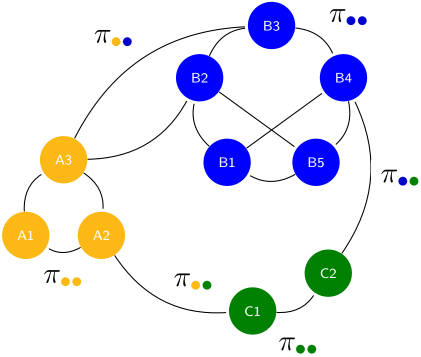
The conditional distribution of the edges

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

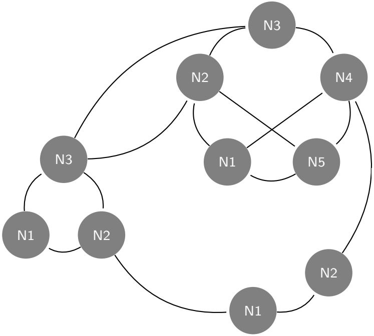
Let n nodes divided into

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

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

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

Stochastic Block Model: unknown parameters



Stochastic Block Model

Let n nodes divided into

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

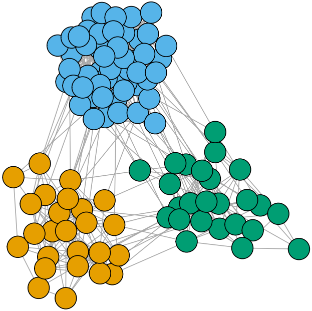
$$Z_i = \mathbf{1}_{\{i \in \bullet\}} \sim^{\text{iid}} \mathcal{M}(1, \alpha), \quad \forall \bullet \in \mathcal{Q},$$

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

Stochastic block models – examples of topology

Community network

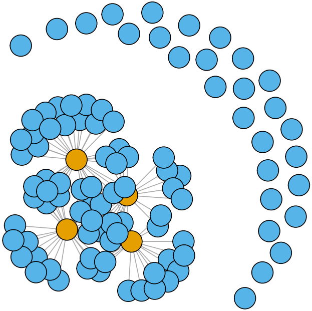
```
pi <- matrix(c(0.3,0.02,0.02,0.02,0.3,0.02,0.02,0.02,0.3),3,3)
communities <- igraph::sample_sbm(100, pi, c(25, 50, 25))
plot(communities, vertex.label=NA, vertex.color = rep(1:3,c(25, 50, 25)))
```



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:

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

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

Model-based method

1. Mixture models

2. The Stochastic Block Model (SBM)
   - Some Graphs Models and their limitations
   - Mixture of Erdös-Rényi and the SBM
   - Inference in SBM with variational EM

The EM strategy does not apply directly for SBM

Ouch: another intractability problem

- the Z_{iq} are not independent conditional on $(X_{ij}, i < j)$ ...
- 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 conditional 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:*

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

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

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 $\mathbb{Q}$ 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 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

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

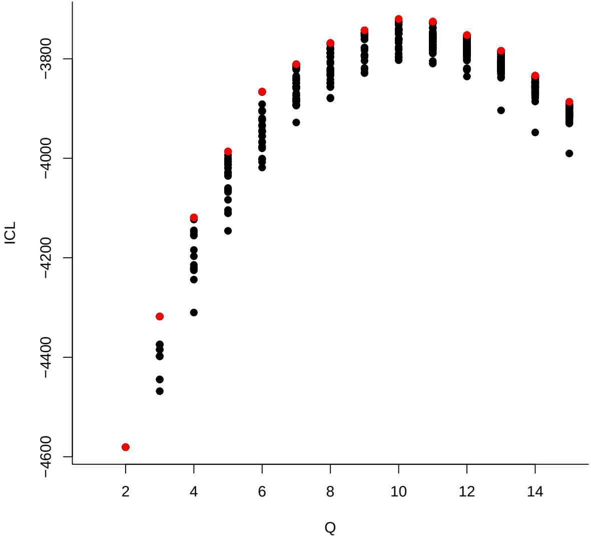
The variational BIC is just

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

Example on the French blogsphere (I)

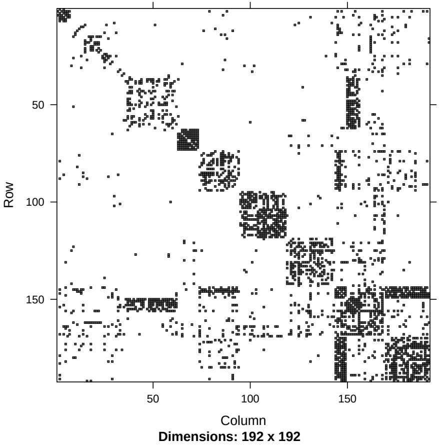
```r
library(blockmodels)
library(sand)

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)

```r
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

```
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