Model-based clustering and classification

part 3: network data and co-clustering

Julien JACQUES

Université Lumière Lyon 2

Mixture model for network data

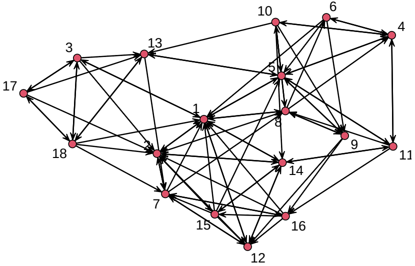
Co-clustering

Mixture model for network data

Network data

Network data arise when the relational data are collected on pairs of entities

```
library(network)
data(sampson , package="ergm")
xSampson <- as.matrix(samplike)
layout <- network.layout.fruchtermanreingold(samplike,
                                                   layout.par=NULL)
plot.network(samplike, label=1:18, mode="fruchtermanreingold",
     coord=layout)
```



Network data

Notations

- n: number of individuals, **nodes** in the graph
- x_{ij} relationship between nodes i and j
  - if $x_{ij} = 1$ if node i and j are linked, 0 otherwise, we talk about **binary** graph. But
  - but x_{ij} can be also in $\mathbb{N}, \mathbb{R}$..
- we consider **undirected** graph: $x_{ij} = x_{ji}$

Network data

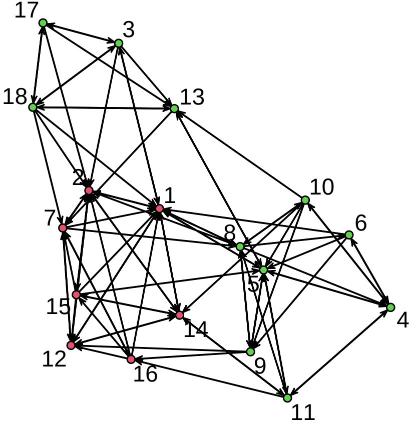
Example: Sampson's Monk data recorder interaction between 18 novice monks in a monastery.

```
head(xSampson)
```

```
##              John Bosco Gregory Basil Peter Bonaventure Bertho
## John Bosco            0       1     1     0           1
## Gregory               1       0     0     0           0
## Basil                 1       1     0     0           0
## Peter                 0       0     0     0           1
## Bonaventure           1       0     0     1           0
## Berthold              1       0     0     1           1
##              Ambrose Romauld Louis Winfrid Amand Hugh Boniface
## John Bosco         0       0     0       1     0    1        0
## Gregory            0       0     0       1     0    1        1
## Basil              0       0     0       0     1    0        0
## Peter              0       1     1       0     0    0        0
## Bonaventure        1       0     1       0     1    0        0
## Berthold           1       0     0       0     0    0        0
##              Simplicius
## John Bosco            0
```

Network clustering

The goal is to detect **communities** in the network: subsets of nodes which have a tendency to form link with each other



The Stochastik Block Model

The Stochastic Block Model (SBM) assumes:

- there is G communities/blocks in the network
- a node come from block g with probability p_g
- α_{gh}: probability that a node in block g is related to a node in block h
- $\tilde{\boldsymbol{z}}_i = (\tilde{z}_{i1}, \ldots, \tilde{z}_{iG})$: membership of node i to the blocks
- all pairwise interactions are independent

Consequently:

$$p(x_{ij} = 1 | \boldsymbol{\alpha}, \tilde{z}_{ig} = 1, \tilde{z}_{jh} = 1) = \alpha_{gh} = \tilde{\boldsymbol{z}}_i' \boldsymbol{\alpha} \tilde{\boldsymbol{z}}_j$$

with $\boldsymbol{\alpha}$ the $G \times G-$matrix of α_{gh}

Snijders, T. A. B., and Nowicki, K. 1997. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. Journal of Classification, 14(1), 75–100. 298, 300

SBM inference

We are in face to a likelihood maximization accoring to parameter $\boldsymbol{\theta} = (\boldsymbol{p}, \boldsymbol{\alpha})$ with missing variables $\underline{\tilde{\boldsymbol{z}}}$.

The complete log-likelihood is:

$$L(\boldsymbol{\theta}, \underline{\boldsymbol{x}}, \underline{\tilde{\boldsymbol{z}}}) = \prod_{i=1}^{n} \prod_{g=1}^{G} p_g^{\tilde{z}_{ig}} \times \prod_{i=1}^{n-1} \prod_{j=i+1}^{n} (\tilde{\boldsymbol{z}}_i \boldsymbol{\alpha} \tilde{\boldsymbol{z}}_j)^{x_{ij}} (1 - \tilde{\boldsymbol{z}}_i \boldsymbol{\alpha} \tilde{\boldsymbol{z}}_j)^{1-x_{ij}}$$

and its logarithm

$$\ell(\boldsymbol{\theta}, \underline{\boldsymbol{x}}, \underline{\tilde{\boldsymbol{z}}}) = \sum_{i=1}^{n} \sum_{g=1}^{G} \tilde{z}_{ig} \ln p_g + \underbrace{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij} \ln(\tilde{\boldsymbol{z}}_i \boldsymbol{\alpha} \tilde{\boldsymbol{z}}_j) + (1 - x_{ij}) \ln(1 - \tilde{\boldsymbol{z}}_i \boldsymbol{\alpha} \tilde{\boldsymbol{z}}_j)}_{A}$$

where

$$A = \sum_{i=1}^{n-1} \sum_{g=1}^{G} \sum_{j=i+1}^{n} \sum_{h=1}^{G} x_{ij} \tilde{z}_{ig} \tilde{z}_{jh} \ln(\alpha_{gh}) + (1 - x_{ij}) \ln(1 - \alpha_{gh}) \tilde{z}_{ig} \tilde{z}_{jh}$$

SBM inference

The E-step of the EM algorithm for the SBM model leads to compute:

$$E_{\boldsymbol{\theta}^{(q)}}[\tilde{z}_{ig}\tilde{z}_{jh}|x_{ij}] = p_{\boldsymbol{\theta}^{(q)}}(\tilde{z}_{ig}\tilde{z}_{jh} = 1|x_{ij})$$

which is intractable since not factorizable: $\tilde{\boldsymbol{z}}_i$ and $\tilde{\boldsymbol{z}}_j$ are not independent since it depends on x_{ij}, the fact there is or not a link between nodes i and j.

We should use alternatives: Gibbs sampling or Variational approximation

Variational approximation

Since $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ can not be calculated, we need to approximate by any $q(\boldsymbol{z})$.

For any $q(\boldsymbol{z})$, we have$^{(1)}$:

$$\begin{aligned} \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}) &\geq \ln p_{\boldsymbol{\theta}}(\boldsymbol{x}) - KL(p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}|\boldsymbol{x}), q(\tilde{\boldsymbol{z}})) \qquad (1) \\ &= E_q[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}, \tilde{\boldsymbol{z}})] - E_q[\ln q(\tilde{\boldsymbol{z}})] \\ &= \mathcal{J}_{\boldsymbol{\theta}}(q) \end{aligned}$$

Equality in (1) occur for $q(\tilde{\boldsymbol{z}}) = p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}|\boldsymbol{x})$

[(1)] *see for instance p29 of* *http://www-math.univ-poitiers.fr/~yslaoui/2019-EL-HAJ-Abir-These.pdf*

Variational EM algorithm

The principe of the Variational EM (VEM) algorithm is to maximize the lower bound $\mathcal{J}_{\boldsymbol{\theta}}(q)$

Since equality occurs when $q(\tilde{\boldsymbol{z}}) = p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}|\boldsymbol{x})$, the VEM consists in alternating 2 steps:

- VE step: find the best lower bound $\mathcal{J}_{\boldsymbol{\theta}}(q)$ according to q. This is equivalent to find $q^* = argmin_q KL(p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}|\boldsymbol{x}), q(\tilde{\boldsymbol{z}}))$
- M step: as usualy, find

$$\boldsymbol{\theta}^* = argmax_{\boldsymbol{\theta}}\, E_{q^*}[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}, \tilde{\boldsymbol{z}})] \tag{2}$$

All the interest is to restrict $q(\boldsymbol{z})$ to a familly of functions which makes (2) tractable.

Mean-field approximation

Mean-field approximation consider factorizable $q(\tilde{\boldsymbol{z}})$:

$$q_{\boldsymbol{\tau}}(\boldsymbol{z}) = \prod_i q(\tilde{z}_i, \tau_i) = \prod_i \prod_g \tau_{ig}^{\tilde{z}_{ig}}$$

Variational EM algorithm

Starting from initialization of variational parameter $\boldsymbol{\tau}^{(0)}$, alternates:

- *pseudo- E step*: update $\boldsymbol{\tau}$:

$$\boldsymbol{\tau}^{(t+1)} = \mathit{argmax}_{\boldsymbol{\tau}}\, \mathcal{J}_{\boldsymbol{\theta}^{(t)}}(q_{\boldsymbol{\tau}})$$

- *M step*: update $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{(t+1)} = \mathit{argmax}_{\boldsymbol{\theta}}\, \mathcal{J}_{\boldsymbol{\theta}}(q_{\boldsymbol{\tau}^{(t+1)}})$$

until $||\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}|| < \epsilon$

Node clustering

One the VEM algorithm has converged, the node clustering can be obtained from the variational distribution of $\boldsymbol{z}$: node i is affected to cluster g which maximizes $\hat{\tau}_{ig}$

Model selection

We have to choose the number of communities.

For the same reason that the EM is not tractable, the likelihood and then the BIC criterion are not tractable.

But the ICL criterion, based on the complete likelihood is tractable, by approximating $\underline{\boldsymbol{z}}$ by $\hat{\underline{\boldsymbol{z}}}$.

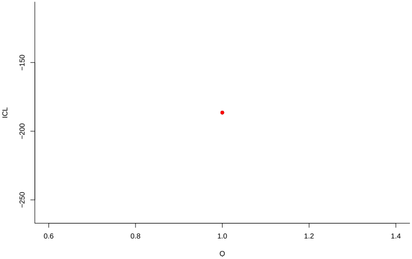
$$ICL = \ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}}, \hat{\underline{\boldsymbol{z}}}) - \frac{\nu}{2} \ln 2\pi$$

where $\nu = G - 1 + G^2$ is the total number of SBM parameter.

SBM in R

```
library(blockmodels)
my_model <- BM_bernoulli("SBM",xSampson)
my_model$estimate()
```

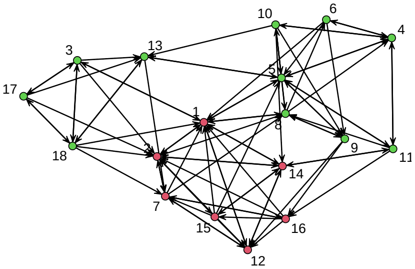
```
## -> Estimation for 1 groups
##                  -> 1 initializations provided
##                  -> 0 initializations already used
##              -> Estimation with 1 initializations
## Executing 1 jobs in parallel
##                      -> new ICL: -186.453270386301
##                      -> old ICL: NA
```



```
## -> Computation of eigen decomposition used for initalizations
```

SBM in R

```r
G=which.max(my_model$ICL)
z <- my_model$memberships[[G]]
plot.network(samplike, label=1:18, mode="fruchtermanreingold",
     coord=layout,vertex.col=max.col(z$Z)+1)
```



Exercice

The **PoliticalBlogs** data set shows the linking structure in online blogs which commentate on French political issues; the data were collected by Observatoire Presidentielle in October 2006

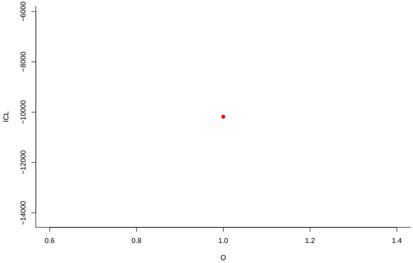
Let use SBM to detect communities in the PoliticalBlogs network

```
library(MBCbook)
data(PoliticalBlogs)
```

Exercice

```
library(MBCbook)
data(PoliticalBlogs)
xPoliticalBlogs=as.matrix(PoliticalBlogs)
my_model <- BM_bernoulli("SBM",xPoliticalBlogs)
my_model$estimate();
```

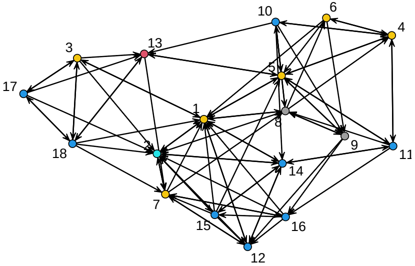
```
## -> Estimation for 1 groups
##                 -> 1 initializations provided
##                 -> 0 initializations already used
##             -> Estimation with 1 initializations
## Executing 1 jobs in parallel                                              -> B
##                     -> new ICL: -10180.2078002613
##                     -> old ICL: NA
```



```
## -> Computation of eigen decomposition used for initalizations
##
```

Exercice

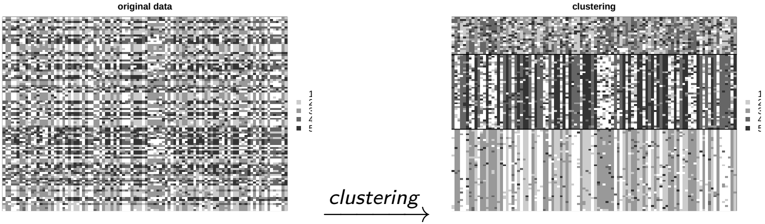
```
G=which.max(my_model$ICL)
z <- my_model$memberships[[G]]
plot.network(samplike, label=1:18, mode="fruchtermanreingold",
     coord=layout,vertex.col=max.col(z$Z)+1)
```



Co-clustering

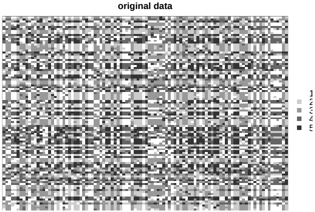
Co-clustering

When the number of features becomes large... cluster interpretation is impossible

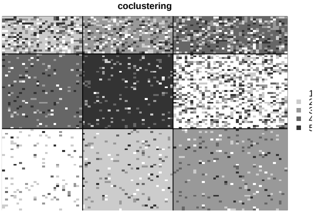


Co-clustering

Co-clustering clusterizes both the observations (row) and the features (colums)



$\xrightarrow{co-clustering}$



Co-clustering notations

- K: number of row clusters
- L: number of column clusters
- row-cluster partitions: $\boldsymbol{v} = (v_{ik})_{i,k}$,
  s.t. $v_{ik} = 1$ if row i belong to row-cluster k
- column-cluster partitions: $\boldsymbol{w} = (w_{h\ell})_{h,\ell}$,
  s.t. $w_{h\ell} = 1$ if column h belong to row-cluster ℓ
- α_k: row-cluster propotion
- β_ℓ: column-cluster propotion
- $\pi_{k\ell}$: the parameter of the distribution in block $k\ell$ (the distribution depends on the nature of the data)
- $\boldsymbol{\theta} = (\pi_{k\ell}, \alpha_k, \beta_\ell)$: the whole set of LBM parameter

Latent Block Model (LBM)

LBM Assumption 1:

- $\boldsymbol{v}$ and $\boldsymbol{w}$ are independent

Consequently, LBM likelihood is

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{v}\in V} \sum_{\boldsymbol{w}\in W} p(\boldsymbol{v};\boldsymbol{\theta})p(\boldsymbol{w};\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{w};\boldsymbol{\theta})$$

with

- V (W) set of possible partitions of rows (columns) into K (L) groups,
- $p(\boldsymbol{v};\boldsymbol{\theta}) = \prod_{ik} \alpha_k^{v_{ik}}$ and $p(\boldsymbol{w};\boldsymbol{\theta}) = \prod_{h\ell} \beta_\ell^{w_{h\ell}}$

Latent Block Model (LBM)

LBM Assumption 2:

- once the row $\boldsymbol{v}$ and column $\boldsymbol{w}$ partitions are fixed, the $n \times p$ random variables $\boldsymbol{x}$ are assumed to be independent, idenditically distributed in each block by $p(\cdot;\pi_{k\ell})$

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{v}\in V}\sum_{\boldsymbol{w}\in W}\prod_{ik}\alpha_k^{v_{ik}}\prod_{h\ell}\beta_\ell^{w_{h\ell}}\prod_{ihk\ell}p(x_{ih};\pi_{k\ell})^{v_{ik}w_{h\ell}}$$

LBM parcimony

Summarizing the $n \times p$ data matrix with univariate distributions in the $K \times L$ groups in very parsimonious, in comparision in a clustering in which the distribution in the K clusters would be p-variate.

Even if we are not interested in co-clustering, but only in clustering, LBM model can be seen as a data-driven parsimonious clustering model

Example in the Gaussian case:

- with $n = p = 100$
- clustering only with GMM: with $K = 3$, the number of parameters is $(3-1) + 3 \times 100 + 3 \times 100 \times 101/2 = 15452$
- LBM co-clustering: with $K = L = 3$, the number of parameters is $(3-1) + (3-1) + 3 \times 3 \times (1+1) = 22$

LBM likelihood

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{v}\in V} \sum_{\boldsymbol{w}\in W} p(\boldsymbol{v};\boldsymbol{\theta}) p(\boldsymbol{w};\boldsymbol{\theta}) p(\boldsymbol{x}|\boldsymbol{v},\boldsymbol{w};\boldsymbol{\theta})$$

The likelihood is not computationally tractable due to the double missing structure ($\boldsymbol{v}$ and $\boldsymbol{w}$):

- $\#V = K^n$,
- $\#W = L^p$
- $\Rightarrow \sum_{\boldsymbol{v},\boldsymbol{w}}$ has $K^n L^p$ terms

LBM inference

The aim is to estimate θ by maximizing the observed log-likelihood.

Classical approach is to use an EM algorithm, which alternates

- E step: compute

$$E_{\boldsymbol{\theta}^{(q)}}[\ell_{\boldsymbol{c}}(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w})|\boldsymbol{x}] = E_{\boldsymbol{\theta}^{(q)}}[\ln p(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w}; \boldsymbol{\theta})|\boldsymbol{x}].$$

- M step: update $\boldsymbol{\theta}$ by

$$\boldsymbol{\theta}^{(q+1)} = argmax_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}^{(q)}}[\ln p(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{w}; \boldsymbol{\theta})|\boldsymbol{x}]$$

But for the same reason that the likelihood is not tractable, the **E step is also not tractable**.

Stochastic EM

Let remind that the **Stochastic EM** algorithm is a variant of the EM algorithm, obtained by *generating* the missing variable at the SE step:

- SE step:
  - generate missing data $(\boldsymbol{v}, \boldsymbol{w})$ cond. on $\boldsymbol{x}$ and $\boldsymbol{\theta}^{(q)}$
- M step:
  - update $\theta^{(q)}$ in $\theta^{(q+1)}$ maximizing the complete likelihood

After a given number of iterations, $\hat{\boldsymbol{\theta}}$ is obtained from the sampling distribution (after a burn-in period)

Stochastic EM within Gibbs

Since the distribution of $(\boldsymbol{v}, \boldsymbol{w})$ is not tractable, the idea is to generate them using a Gibbs sampler, which alternates on few iterations:

- SE-Gibbs step:
  - generate $\boldsymbol{v}$ cond. on $(\boldsymbol{x}, \boldsymbol{w})$ and $\boldsymbol{\theta}^{(q)}$
  - generate $\boldsymbol{w}$ cond. on $(\boldsymbol{x}, \boldsymbol{v})$ and $\boldsymbol{\theta}^{(q)}$
- M step:
  - update $\theta^{(q)}$ in $\theta^{(q+1)}$ maximizing the complete likelihood

After a given number of iterations, $\hat{\boldsymbol{\theta}}$ is obtained from the sampling distribution (after a burn-in period)

Partition estimation $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{w}})$ can be obtained with a Gibbs sampler using $\hat{\boldsymbol{\theta}}$.

Model selection

Since the likelihood is not tractable, BIC is also not tractable.

We can use ICL which is based on the complete likelihood, by approximating it using $(\hat{\boldsymbol{v}}, \hat{\boldsymbol{w}})$:

$$ICL = \ln p(\boldsymbol{x}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{w}}; \boldsymbol{\theta}) - \frac{K-1}{2} \log n - \frac{L-1}{2} \log d - \frac{KL\nu}{2} \log(nd)$$

where ν is the number of model parameter per block.

LBM in practice

```
library(blockcluster)
data(binarydata)
out<-coclusterBinary(binarydata,nbcocluster=c(2,3))
```

```
## Co-Clustering successfully terminated!
```

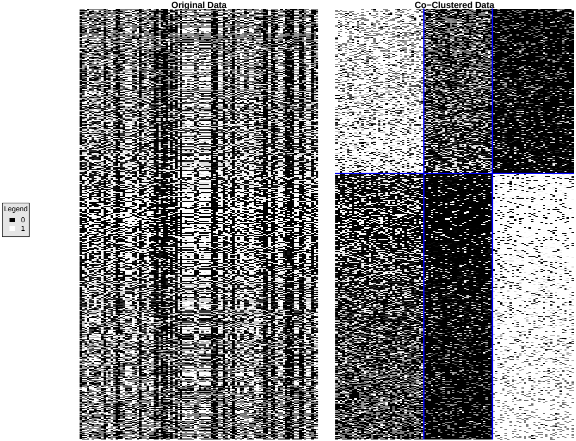
LBM in practice

```
summary(out)
```

```
## *************************************************************
## Model Family : Bernoulli Latent block model
## Model Name : pik_rhol_epsilonkl
## Co-Clustering Type : Unsupervised
## ICL value:  -45557.07
##
## Model Parameters..
##
## Class Mean:
##       [,1]  [,2]  [,3]
## [1,]  TRUE FALSE FALSE
## [2,] FALSE FALSE FALSE
##
## Class Dispersion:
##           [,1]       [,2]      [,3]
## [1,] 0.4131605 0.09584329 0.1027146
## [2,] 0.4684641 0.30207689 0.3085856
##
## Row proportions:  0.382 0.618
```

LBM in practice

```
plot(out)
```



Exercice

The goal is to provide a analysis of the Young People Survey available at https://www.kaggle.com/cardot/se-young-people-survey/data with the `ordinalClust` package.