



HAL
open science

Clustering : une vision unifiée pour une utilisation éclairée - Partie 4 : Traitement de la grande dimension & co-clustering

C Biernacki

► To cite this version:

C Biernacki. Clustering : une vision unifiée pour une utilisation éclairée - Partie 4 : Traitement de la grande dimension & co-clustering. Doctoral. France. 2023. hal-04370767

HAL Id: hal-04370767

<https://inria.hal.science/hal-04370767v1>

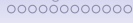
Submitted on 3 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Clustering : une vision unifiée pour une utilisation éclairée

—

Partie 4

Traitement de la grande dimension & co-clustering (leçon plus avancée)

C. Biernacki

Ateliers statistiques de la SFdS, 15 et 16 juin 2023, IHP, Paris





Outline

1 HD clustering

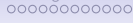
2 Modeling

3 Estimating

4 Selecting

5 Experiments

6 To go further



Motivation

High dimensional (HD) data sets are now frequent:

- Marketing: $d \sim 10^2$
- microarray gene expression: $d \sim 10^2-10^4$
- SNP data: $d \sim 10^6$
- Curves: depends on discretization but can be very high
- Text mining
- ...

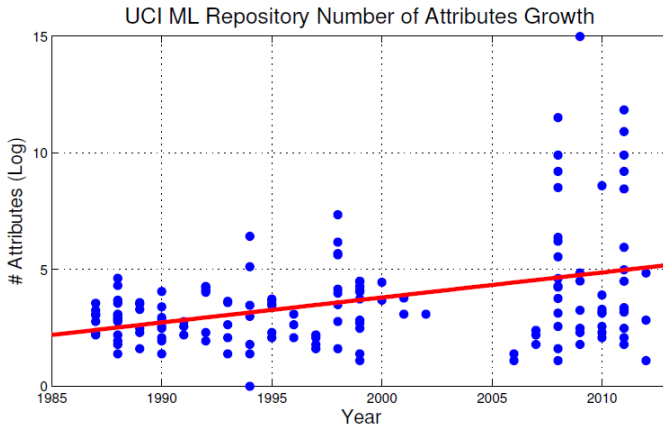
Clustering has to be applied for HD datasets for the same reasons as the lower dimensional datasets:

- Data summary
- Data exploratory
- Preprocessing for more flexibility of a forthcoming prediction step

But clustering is even more important since visualization in the HD setting can be hazardous. . .



Today': exponential growing of dimension¹



¹S. Alelyani, J. Tang and H. Liu (2013). Feature Selection for Clustering: A Review. *Data Clustering: Algorithms and Applications*, 29



HD data: definition (1/2)

An attempt in the non-parametric case

Dataset $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, \mathbf{x}_j described by d variables, where $n = o(e^d)$

Justifications:

- To approximate within error ϵ a (Lipschitz) function of d variables, about $(1/\epsilon)^d$ evaluations on a grid are required [Bellman, 61]
- Approximate a Gaussian distribution with fixed Gaussian kernels and with approximate error of about 10% [Silverman, 86]

$$\log_{10} n(d) \approx 0.6(d - 0.25)$$

For instance, $n(10) \approx 7.10^5$



HD data: definition (2/2)

An attempt in the parametric case

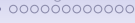
Dataset $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n)$, \mathbf{x}_j described by d variables and a model \mathbf{m} with ν parameters, where $n = o(g(\nu))$, with g a given function

Justification:

- We consider the heteroscedastic Gaussian mixture with of true parameter θ^* with K^* components. We note $\hat{\theta}$ the Gaussian MLE with K^* components. We have [linear](#) from the following result [[Michel, 08](#)]: it exists constants κ , A and C such that

$$E_x[\text{Hellinger}^2(p_{\theta^*}, p_{\hat{\theta}_{\hat{K}}})] \leq C \left[\kappa \frac{\nu}{n} \left\{ 2A \ln d + 1 - \ln \left(1 \wedge \left[\frac{\nu}{n} A \ln d \right] \right) \right\} + \frac{1}{n} \right].$$

But ν can be high since $\nu \sim d^2/2$, combined with potentially large constants.

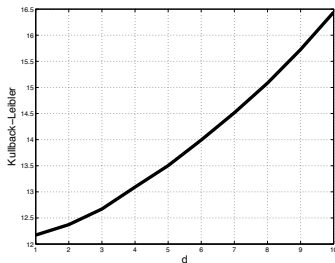
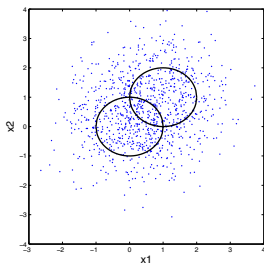


HD density estimation: curse

A two-component d -variate Gaussian mixture:

$$\pi_1 = \pi_2 = \frac{1}{2}, \quad \mathbf{X}_1 | z_{11} = 1 \sim N_d(\mathbf{0}, \mathbf{I}), \quad \mathbf{X}_1 | z_{12} = 1 \sim N_d(\mathbf{1}, \mathbf{I})$$

Components are **more and more separated** when d grows: $\|\mu_2 - \mu_1\|_1 = \sqrt{d} \dots$



... but **density estimation quality decreases** with d



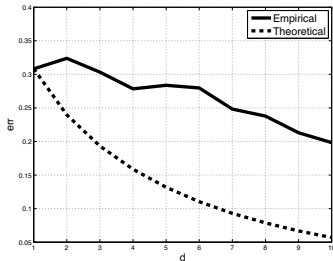
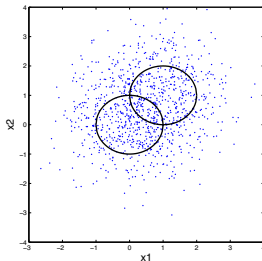
HD clustering: blessing (1/2)

A two-component d -variate Gaussian mixture:

$$\pi_1 = \pi_2 = \frac{1}{2}, \quad \mathbf{X}_1 | z_{11} = 1 \sim N_d(\mathbf{0}, \mathbf{I}), \quad \mathbf{X}_1 | z_{12} = 1 \sim N_d(\mathbf{1}, \mathbf{I})$$

Each variable provides **equal** and **own** separation information

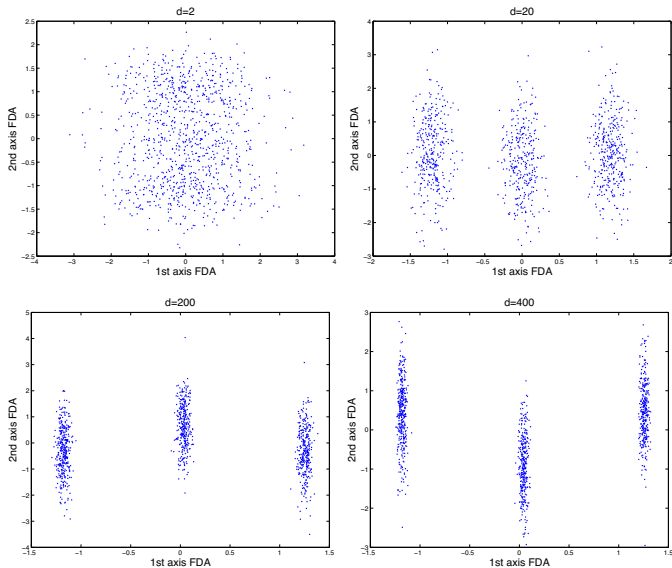
Theoretical error decreases when d grows: $\text{err}_{theo} = \Phi(-\sqrt{d}/2) \dots$

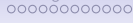


... and **empirical error rate decreases** also with d !

HD clustering: blessing (2/2)

FDA





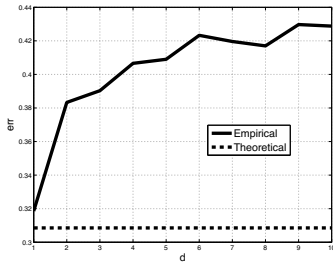
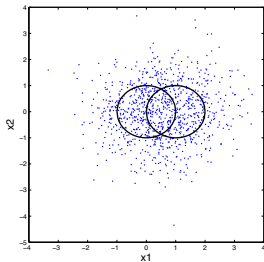
HD clustering: curse (1/2)

Many variables provide **no separation information**

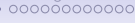
Same parameter setting except:

$$\mathbf{X}_1 | z_{12} = 1 \sim N_d((1 \ 0 \ \dots \ 0)', \mathbf{I})$$

Groups are **not separated more** when d grows: $\|\mu_2 - \mu_1\|_1 = 1 \dots$



... thus **theoretical error is constant** ($= \Phi(-\frac{1}{2})$) and **empirical error increases** with d



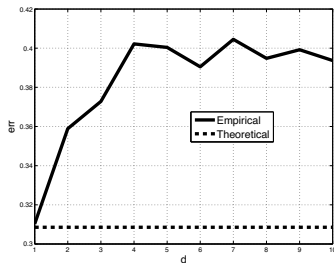
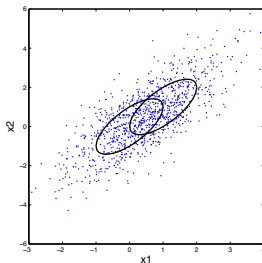
HD clustering: curse (2/2)

Many variables provide **redundant separation information**

Same parameter setting except:

$$\mathbf{x}_1^j = \mathbf{x}_1^1 + N_1(0, 1) \quad (j = 2, \dots, d)$$

Groups are **not separated more** when d grows: $\|\mu_2 - \mu_1\|_{\Sigma} = 1 \dots$



... thus err_{theo} is constant ($= \Phi(-\frac{1}{2})$) and **empirical error increases (less)** with d

The trade-off bias/variance

The fundamental statistical principle

Always minimize an error err between truth (\mathbf{z}) and estimate ($\hat{\mathbf{z}}$)

- Gap between true (\mathbf{z}) and model-based (\mathcal{Z}_p) partitions: $\mathbf{z}^* = \arg \min_{\tilde{\mathbf{z}} \in \mathcal{Z}_p} \Delta(\mathbf{z}, \tilde{\mathbf{z}})$
- Estimation $\hat{\mathbf{z}}$ of \mathbf{z}^* in \mathcal{Z}_p : any relevant method (bias, consistency, efficiency...)
- Fundamental decomposition of the observed error $\text{err}(\mathbf{z}, \hat{\mathbf{z}})$:

$$\begin{aligned} \text{err}(\mathbf{z}, \hat{\mathbf{z}}) &= \left\{ \text{err}(\mathbf{z}, \mathbf{z}^*) - \text{err}(\mathbf{z}, \mathbf{z}) \right\} + \left\{ \text{err}(\mathbf{z}, \hat{\mathbf{z}}) - \text{err}(\mathbf{z}, \mathbf{z}^*) \right\} \\ &= \left\{ \text{bias} \right\} + \left\{ \text{variance} \right\} \\ &= \left\{ \text{error of approximation} \right\} + \left\{ \text{error of estimation} \right\} \end{aligned}$$



Gaussian mixture of factor analysers (1/2)

Definition

[Ghahramani and Hinton, 97], [McLachlan *et al.*, 03]

$$\Sigma_k = \mathbf{B}_k \mathbf{B}_k' + \omega_k \Lambda_k$$

where

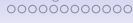
- \mathbf{B}_k is a *loadings* $d \times q$ non-square real matrix ($1 \leq q \leq q_{\max}$, $q_{\max} < d$)
- ω_k is a positive real number
- Λ_k is a $d \times d$ diagonal positive definite matrix such that $|\Lambda_k| = 1$

Interpretation $\mathbf{X}_1 \in \mathbb{R}^d$ is generated by a latent variable $\mathbf{Y}_1 \in \mathbb{R}^q$

$$\mathbf{X}_1 | \mathbf{Y}_1, Z_{1k}=1 = \mathbf{B}_k \underbrace{\mathbf{Y}_1}_{\text{factor}} + \boldsymbol{\mu}_k + \boldsymbol{\varepsilon}_k$$

where $\mathbf{Y}_1 \perp \boldsymbol{\varepsilon}_k$ and

$$\mathbf{Y}_1 \sim N_q(\mathbf{0}, \mathbf{I}) \quad \text{and} \quad \boldsymbol{\varepsilon}_k \sim N_d(\mathbf{0}, \omega_k \Lambda_k)$$



Gaussian mixture of factor analysers (2/2)

Complexity

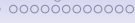
$$\nu = (K - 1) + Kd + Kq(d - (q - 1)/2) + Kd$$

Model selection

- Models in competition: $\mathbf{m} = (q, K) + 12$ parsimonious versions [McNicholas and Murphy, 10]
- Classical criteria can be used

Package

PGMM: <http://cran.r-project.org/web/packages/pgmm/index.html>



HD Gaussian models (1/2)

Definition

[Bouveyron *et al.*, 07]

$$\Sigma_k = \mathbf{D}_k \mathbf{\Delta}_k \mathbf{D}_k'$$

where

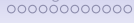
- \mathbf{D}_k is the orthogonal matrix of the eigenvectors of Σ_k
- $\mathbf{\Delta}_k$ is a diagonal matrix containing the eigenvalues of Σ_k

$$\mathbf{\Delta}_k = \left(\begin{array}{ccc|ccc} \boxed{\begin{matrix} a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{k\delta_k} \end{matrix}} & & & & & \\ & & & & \mathbf{0} & \\ \hline & & & \mathbf{0} & & \\ & & & & \boxed{\begin{matrix} b_k & & 0 \\ & \ddots & \\ 0 & & b_k \end{matrix}} & \\ & & & & & \end{array} \right) \left. \begin{array}{l} \vphantom{\left(\right)} \right\} \delta_k \\ \vphantom{\left(\right)} \right\} (d - \delta_k)$$

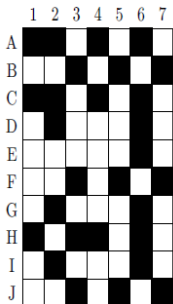
with $a_{kj} \geq b_k$, for $j = 1, \dots, \delta_k$ and $\delta_k < d$

Complexity

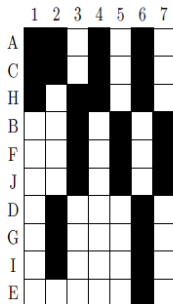
$$\nu = (K - 1) + Kd + \sum_{k=1}^K \delta_k [d - (\delta_k + 1)/2] + \sum_{k=1}^K \delta_k + 2K$$



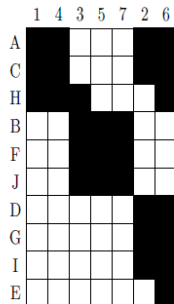
From clustering to co-clustering



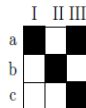
(1)



(2)



(3)



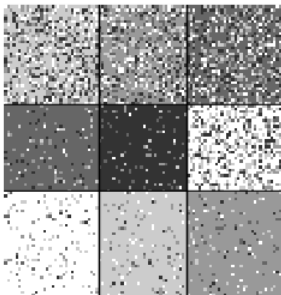
(4)

[Govaert, 2011]

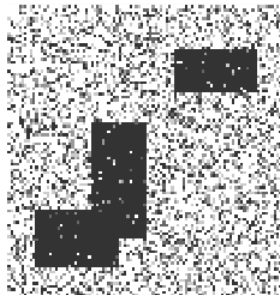
Bi-clustering

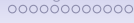
- A generalization of co-clustering
- Look for submatrices of x which are homogeneous
- We do not consider bi-clustering here

coclustering



biclustering





Link between co-clustering and PCA/kmeans²

Interpreting LBM as an MBC dimension reduction method PCA is certainly the most emblematic reduction dimension method for numerical data sets \mathbf{x} (we consider in the following that \mathbf{x} is centered in columns). It proceeds in two steps. First, it expresses each data unit \mathbf{x}_i of the initial data set \mathbf{x} in a new vector basis $(\mathbf{u}^1, \dots, \mathbf{u}^d)$, ordered in decreasing value of the preserved variance, with respective coordinates (a_i^1, \dots, a_i^d) and where each \mathbf{u}^j is defined by a linear combination of the canonical vector basis $(\mathbf{e}^1, \dots, \mathbf{e}^d)$, namely $\mathbf{u}^j = \sum_{j'=1}^d b_{j'} \mathbf{e}^{j'}$. Second, it selects a reduced number of these new coordinates only (say J), resulting in a new data set of smaller dimension ($J < d$). This PCA sequential procedure for the i -th data individual \mathbf{x}_i can be expressed as follows:

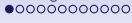
$$\mathbf{x}_i = \sum_{j=1}^d x_i^j \mathbf{e}^j = \sum_{j=1}^d a_i^j \mathbf{u}^j \approx \sum_{j=1}^J a_i^j \mathbf{u}^j. \quad (7)$$

Consider now the (very) specific co-clustering case reduced to $K = 1$, thus meaning that only a variable clustering in L column clusters is performed. Let $\varepsilon_i^j \sim \mathcal{N}(0, \sigma_{\tilde{w}_j}^2)$ in an i.i.d manner, where \tilde{w}_j is the cluster index such that $\tilde{w}_j = \ell \Leftrightarrow w_{j\ell} = 1$. We can write, with $\mathbf{v}^\ell = \sum_{\{j: \tilde{w}_j = \ell\}} \mathbf{e}^j$ and $\mathbf{r}_i = \sum_{j=1}^d \varepsilon_i^j \mathbf{e}^j$:

$$\mathbf{x}_i = \sum_{j=1}^d x_i^j \mathbf{e}^j = \sum_{j=1}^d (\mu_{\tilde{w}_j} + \varepsilon_i^j) \mathbf{e}^j = \sum_{\ell=1}^{L_r} \mu_\ell \mathbf{v}^\ell + \mathbf{r}_i \approx \sum_{\ell=1}^{L_r} \mu_\ell \mathbf{v}^\ell. \quad (8)$$

It reinforces the interest of co-clustering for the HD case...

²C. Biernacki, J. Jacques, C. Keribin (2023). A Survey on Model-Based Co-Clustering: High Dimension and



Outline

1 HD clustering

2 Modeling

3 Estimating

4 Selecting

5 Experiments

6 To go further



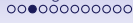
Notations

- \mathbf{z}_i : the cluster of the row i
- \mathbf{w}_j : the cluster of the column j
- $(\mathbf{z}_i, \mathbf{w}_j)$: the **block** of the element \mathbf{x}_{ij} (row i , column j)

- $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_n)$: partition of individuals in K clusters of rows
- $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_d)$: partition of variables in L clusters of columns
- (\mathbf{z}, \mathbf{w}) : **bi-partition** of the whole data set \mathbf{x}
- Both space partitions are respectively denoted by \mathcal{Z} and \mathcal{W}

Restriction

All variables are of the same kind (see discussion at the end)



A geometric approach

- Example in the continuous case: $\mathbf{x} \in \mathbb{R}^{n \times d}$
- It could be possible to define a **within-block** inertia criterion

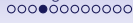
$$W(\mathbf{z}, \mathbf{w}) = \underbrace{\sum_{k=1}^K \sum_{l=1}^L \sum_{i=1}^n \sum_{j=1}^d}_{\sum_{i,j,k,l}} z_{ik} w_{jl} \|x_{ij} - \mu_{kl}\|^2$$

with μ_{kl} the **center of the block** (k, l)

$$\mu_{kl} = \frac{1}{n_{kl}} \sum_{i,j} z_{ik} w_{jl} x_{ij}$$

where $n_{kl} = \sum_{i,j} z_{ik} w_{jl}$ is the **sample size of the block** (k, l)

But we know now that it hides some model-based assumptions. . .



The latent block model (LBM)

- Generalization of some existing non-probabilistic methods
- Extend the latent class principle of local (or conditional) independence
- Thus x_{ij} is assumed to be independent once z_i and w_j are fixed ($\alpha = (\alpha_{kl})$):

$$p(\mathbf{x}|\mathbf{z}, \mathbf{w}; \alpha) = \prod_{i,j} p(x_{ij}; \alpha_{z_i w_j})$$

- $\pi = (\pi_k)$: vectors of proba. π_k that a row belongs to the k th row cluster
- $\rho = (\rho_l)$: vectors of proba. ρ_l that a row belongs to the l th column cluster
- Independence between all z_i and w_j
- Extension of the traditional mixture model-based clustering ($\alpha = (\alpha_{kl})$):

$$p(\mathbf{x}; \theta) = \sum_{(\mathbf{z}, \mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} \prod_{i,j} \pi_{z_i} \rho_{w_j} p(x_{ij}; \alpha_{z_i w_j})$$



Distribution for different kinds of data

[Govaert and Nadif, 2014] The pdf $p(\cdot; \alpha_{z_i w_j})$ depends on the kind of data x_{ij} :

- **Binary** data: $x_{ij} \in \{0, 1\}$, $p(\cdot; \alpha_{kl}) = \mathcal{B}(\alpha_{kl})$
- **Categorical** data with m levels:
 $\mathbf{x}_{ij} = \{x_{ijh}\} \in \{0, 1\}^m$ with $\sum_{h=1}^m x_{ijh} = 1$ and $p(\cdot; \alpha_{kl}) = \mathcal{M}(\alpha_{kl})$ with $\alpha_{kl} = \{\alpha_{kjh}\}$
- **Count** data: $x_i^j \in \mathbb{N}$, $p(\cdot; \alpha_{kl}) = \mathcal{P}(\mu_k \nu_l \gamma_{kl})^3$
- **Continuous** data: $x_i^j \in \mathbb{R}$, $p(\cdot; \alpha_{kl}) = \mathcal{N}(\mu_{kl}, \sigma_{kl}^2)$

³The Poisson parameter is here split into μ_k and ν_l the effects of the row k and the column l respectively and γ_{kl} the effect of the block kl . Unfortunately, this parameterization is not identifiable. It is therefore not possible to estimate simultaneously μ_k , ν_l and γ_{kl} without imposing further constraints. Constraints $\sum_k \pi_k \gamma_{kl} = \sum_l \rho_l \gamma_{kl} = 1$ and $\sum_k \mu_k = 1, \sum_l \nu_l = 1$ are a possibility.



Extreme parsimony ability

Model	Number of parameters
Binary	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + KL$
Categorical	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + KL(m - 1)$
Contingency	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + KL$
Continuous	$\dim(\boldsymbol{\pi}) + \dim(\boldsymbol{\rho}) + 2KL$

Very parsimonious so well suitable for the (ultra) HD setting

$$\text{nb. param.}_{\text{HD}} = \text{nb. param.}_{\text{classic}} \times \frac{L}{d}$$

Other advantage: stay in the canonical space thus meaningful for the end-user



Binary illustration: easy interpretation

[Govaert, 2011]

	<i>abcdefghij</i>
y1	1010001101
y2	0101110011
y3	1000001100
y4	1010001100
y5	0111001100
y6	0101110101
y7	0111110111
y8	1100111011
y9	0100110000
y10	1010101101
y11	1010001100
y12	1010000100
y13	1010001101
y14	0010011100
y15	0010010100
y16	1111001100
y17	0101110011
y18	1010011101
y19	1010001000
y20	1100101100

Raw data

	<i>a c g h</i>	<i>b d e f i j</i>
y2	0 0 0 0	1 1 1 1 1 1
y6	0 0 0 1	1 1 1 1 0 1
y7	0 1 0 1	1 1 1 1 1 1
y8	1 0 1 0	1 0 1 1 1 1
y9	0 0 0 0	1 0 1 1 0 0
y17	0 0 0 0	1 1 1 1 1 1
y1	1 1 1 1	0 0 0 0 0 1
y3	1 0 1 1	0 0 0 0 0 0
y4	1 1 1 1	0 0 0 0 0 0
y5	0 1 1 1	1 1 0 0 0 0
y10	1 1 1 1	0 0 1 0 0 1
y11	1 1 1 1	0 0 0 0 0 0
y12	1 1 0 1	0 0 0 0 0 0
y13	1 1 1 1	0 0 0 0 0 1
y14	0 1 1 1	0 0 0 1 0 0
y15	0 1 0 1	0 0 0 1 0 0
y16	1 1 1 1	1 1 0 0 0 0
y18	1 1 1 1	0 0 0 1 0 1
y19	1 1 1 0	0 0 0 0 0 0
y20	1 0 1 1	1 0 1 0 0 0

Permuted data
(rows/columns)

mode

0	1
1	0

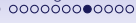
Summary

0.86	0.79
0.83	0.86

Homogeneity

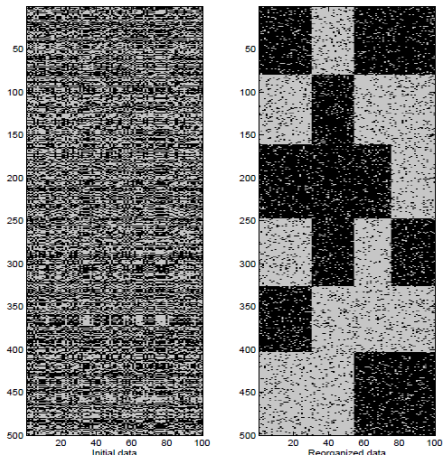
proba=mode

iid Bin(0.83)

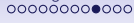


Binary illustration: user-friendly visualization

[Govaert, 2011]



$$n = 500, d = 10, K = 6, L = 4$$



Other kind of data: ordinal

[Jacques and Biernacki, 2018]

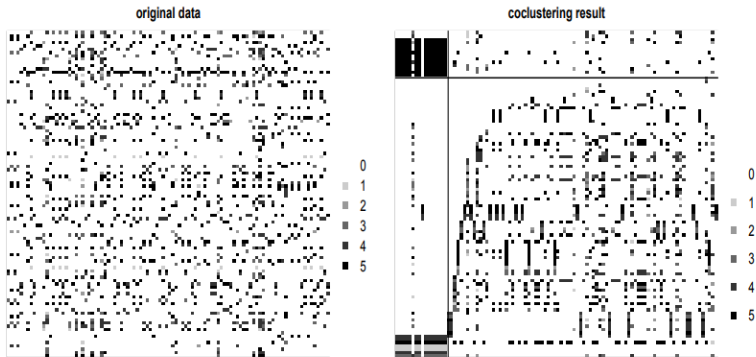
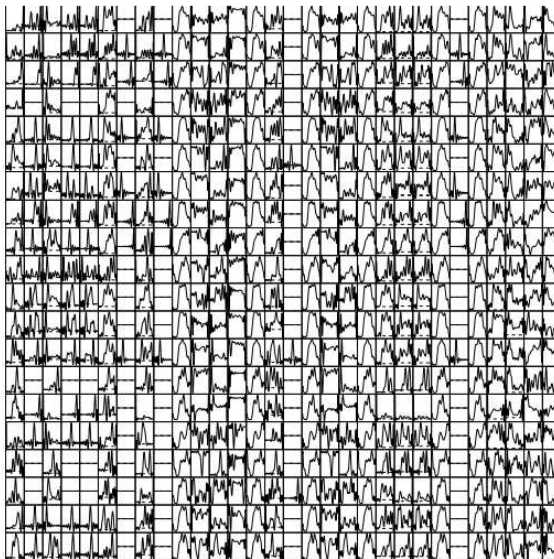
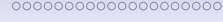
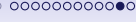


Figure 11: Top 100 Amazon Fine Food Review data (left) and co-clustering result (right).

Other kind of data: functional

[Jacques, 2016]



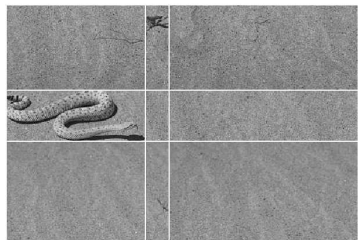


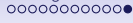
Other kind of data: image

Original Data



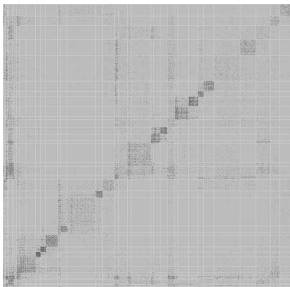
Co-Clustered Data

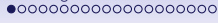




Particular case: graph clustering

Stochastic Block Model (SBM): adjacency matrix with $n = d$ and $K = L$





Outline

1 HD clustering

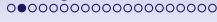
2 Modeling

3 Estimating

4 Selecting

5 Experiments

6 To go further



MLE estimation: log-likelihood(s)

- Remember Lesson 3: first estimate θ , then deduce estimate of (z, \mathbf{w})
- Observed log-likelihood: $\ell(\theta; \mathbf{x}) = \ln p(\mathbf{x}; \theta)$
- MLE:

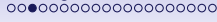
$$\hat{\theta} = \arg \max_{\theta} \ell(\theta; \mathbf{x})$$

- Complete log-likelihood:

$$\begin{aligned} \ell_c(\theta; \mathbf{x}, \mathbf{z}, \mathbf{w}) &= \ln p(\mathbf{x}, \mathbf{z}, \mathbf{w}; \theta) \\ &= \sum_{i,k} z_{ik} \log \pi_k + \sum_{k,l} w_{jl} \log \rho_l + \sum_{i,j,k,l} z_{ik} w_{jl} \log p(x_i^j; \alpha_{kl}) \end{aligned}$$

Be careful with asymptotics. . .

If $\ln(d)/n \rightarrow 0$, $\ln(n)/d \rightarrow 0$ when $n \rightarrow \infty$ and $d \rightarrow \infty$, then the MLE is consistent
 [Brault et al., 2017]



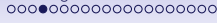
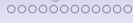
MLE estimation: EM algorithm

- **E-step** of EM (iteration q):

$$\begin{aligned}
 Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) &= E[\ell_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}, \mathbf{w}) | \mathbf{x}; \boldsymbol{\theta}^{(q)}] \\
 &= \sum_{i,k} \underbrace{p(z_i = k | \mathbf{x}; \boldsymbol{\theta}^{(q)})}_{t_{ik}^{(q)}} \ln \pi_k + \sum_{j,l} \underbrace{p(w_j = l | \mathbf{x}; \boldsymbol{\theta}^{(q)})}_{s_{jl}^{(q)}} \ln \rho_l \\
 &\quad + \sum_{i,j,k,l} \underbrace{p(z_i = k, w_j = l | \mathbf{x}; \boldsymbol{\theta}^{(q)})}_{e_{ijkl}^{(q)}} \ln p(x_{ij}; \alpha_{kl})
 \end{aligned}$$

- **M-step** of EM (iteration q): classical. For instance, for the Bernoulli case, it gives

$$\pi_k^{(q+1)} = \frac{\sum_i t_{ik}^{(q)}}{n}, \quad \rho_l^{(q+1)} = \frac{\sum_j s_{jl}^{(q)}}{d}, \quad \alpha_{kl}^{(q+1)} = \frac{\sum_{i,j} e_{ijkl}^{(q)} x_{ij}}{\sum_{i,j} e_{ijkl}^{(q)}}$$



MLE: intractable E step

$e_{ijkl}^{(q)}$ is usually intractable...

- Consequence of dependency between x_{ij} s (link between rows and columns)
- Involve $K^n L^d$ calculus (number of possible blocks)
- Example: if $n = d = 20$ and $K = L = 2$ then 10^{12} blocks
- Example (cont'd): 33 years with a computer calculating 100,000 blocks/second

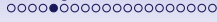
Alternatives to EM

- **Variational EM** (numerical approx.): conditional independence assumption

$$p(\mathbf{z}, \mathbf{w} | \mathbf{x}; \theta) \approx p(\mathbf{z} | \mathbf{x}; \theta) p(\mathbf{w} | \mathbf{x}; \theta)$$

- **SEM-Gibbs** (stochastic approx.): replace E-step by a S-step approx. by Gibbs

$$\mathbf{z} | \mathbf{x}, \mathbf{w}; \theta \quad \text{and} \quad \mathbf{w} | \mathbf{x}, \mathbf{z}; \theta$$



MLE: variational EM (1/2)

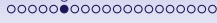
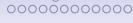
- Use a general variational result from [\[Hathaway, 1985\]](#)
- Maximizing $\ell(\boldsymbol{\theta}; \mathbf{x})$ on $\boldsymbol{\theta}$ is equivalent to maximize $\tilde{\ell}_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{e})$ on $(\boldsymbol{\theta}, \mathbf{e})$

$$\tilde{\ell}_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{e}) = \sum_{i,k} t_{ik} \ln \pi_k + \sum_{j,l} s_{jl} \ln \rho_l + \sum_{i,j,k,l} e_{ijkl} \ln p(x_{ij}; \boldsymbol{\alpha}_{kl})$$

where $\mathbf{e} = (e_{ijkl})$, $e_{ijkl} \in \{0, 1\}$, $\sum_{k,l} e_{ijkl} = 1$, $t_{ik} = \sum_{j,l} e_{ijkl}$, $s_{jl} = \sum_{i,k} e_{ijkl}$

- Of course maximizing $\ell(\boldsymbol{\theta}; \mathbf{x})$ or $\tilde{\ell}_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{e})$ are both intractable
- **Idea:** restriction on \mathbf{e} to obtain tractability $\mathbf{e}_{ijkl} = t_{ik}s_{jl}$
- New variables are thus now $\mathbf{t} = (t_{ik})$ and $\mathbf{s} = (s_{jl})$
- As a consequence, it is a maximization of a **lower bound** of the max. likelihood

$$\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; \mathbf{x}) \geq \max_{\boldsymbol{\theta}, \mathbf{t}, \mathbf{s}} \tilde{\ell}_c(\boldsymbol{\theta}; \mathbf{x}, \mathbf{e})$$



MLE: variational EM (2/2)

Approximated E-step

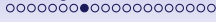
$$Q(\theta, \theta^{(q)}) \approx \sum_{i,k} t_{ik}^{(q)} \ln \pi_k + \sum_{j,l} s_{jl}^{(q)} \ln \rho_l + \sum_{i,j,k,l} t_{ik}^{(q)} s_{jl}^{(q)} \ln p(x_{ij}; \alpha_{kl})$$

- We called it now VEM
- Also known as **mean field** approximation
- **Consistency** of the variational estimate [Brault *et al.*, 2017]



MLE: local maxima

- More local maxima than in classical mixture models
- It is a consequence of many more latent variables (blocks)
- Thus: either many VEM runs, or use the SEM-Gibbs algorithm



MLE: SEM-Gibbs

- We have already seen the SEM algorithm in Lesson 3 (thus we do not detail more)
- It limits dependency to starting point, so it limits local maxima
- The S-step: a draw $(\mathbf{z}^{(q)}, \mathbf{w}^{(q)}) \sim p(\mathbf{z}, \mathbf{w} | \mathbf{x}; \boldsymbol{\theta}^{(q)})$ instead an expectation
- But it is still intractable, thus use a Gibbs algorithm to approx. this draw

Approximated S-step

Two easy draws

$$\mathbf{z}^{(q)} \sim p(\mathbf{z} | \mathbf{w}^{(q-1)}, \mathbf{x}; \boldsymbol{\theta}^{(q)})$$

and

$$\mathbf{w}^{(q)} \sim p(\mathbf{w} | \mathbf{z}^{(q)}, \mathbf{x}; \boldsymbol{\theta}^{(q)})$$

- Rigorously speaking, many draws within the S-step should be performed
- Indeed, Gibbs has to reach a stochastic convergence
- In practice it works well while saving computation time

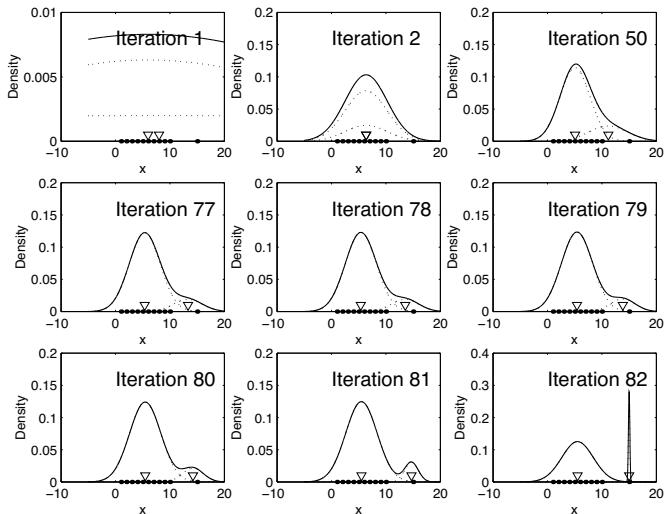


MLE: degeneracy

- More degenerate situations than in classical mixture models
- It is again a consequence of many more latent variables (blocks)
- The Bayesian regularization (instead MLE) can be an answer



Illustration of a degenerate situation





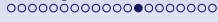
Bayesian estimation: pitch

- Everything passes by the **posterior distribution of θ**

$$p(\theta|\mathbf{x}) \propto \underbrace{p(\mathbf{x}|\theta)}_{\text{log-likelihood}} \underbrace{p(\theta)}_{\text{prior}}$$

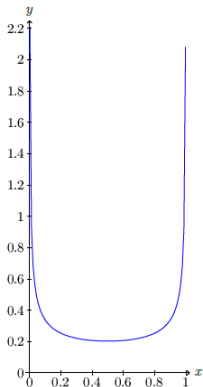
- Then, take (for instance) the **MAP** as a θ estimate (use a VEM like algo...)

$$\hat{\theta} = \arg \max_{\theta} p(\theta|\mathbf{x})$$

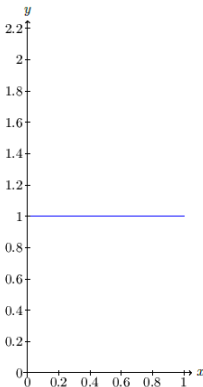


Bayesian estimation: prior overview

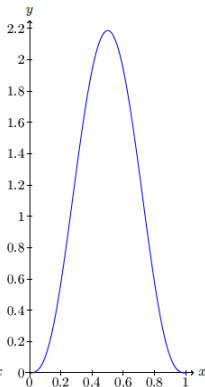
$$Be\left(\frac{1}{2}, \frac{1}{2}\right)$$

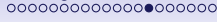


$$Be(1, 1)$$



$$Be(4, 4)$$





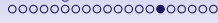
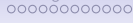
Block estimation: estimate

- Once we have a parameter estimate $\hat{\theta}$, we need to have an block estimate (\hat{z}, \hat{w})
- But MAP not directly available because of the following maximization difficulty

$$(\hat{z}, \hat{w}) = \arg \max_{(z, w)} \underbrace{p(z, w | x; \hat{\theta})}_{\text{intractable}}$$

- Instead the following (easily, as classical mixtures) estimates are usually retained

$$\hat{z} = \arg \max_z p(z | x; \hat{\theta}) \quad \text{and} \quad \hat{w} = \arg \max_w p(w | x; \hat{\theta})$$

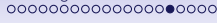
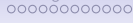


Block estimation: evaluation

- **Empirical error rate** between blocks:

$$\text{err}_{\text{blocks}}\left(\underbrace{(\mathbf{z}, \mathbf{w})}_{\text{"True" blocks}}, \underbrace{(\hat{\mathbf{z}}, \hat{\mathbf{w}})}_{\text{Estimated blocks}}\right) = \text{err}(\mathbf{z}, \hat{\mathbf{z}}) + \text{err}(\mathbf{w}, \hat{\mathbf{w}}) - \text{err}(\mathbf{z}, \hat{\mathbf{z}}) \times \text{err}(\mathbf{w}, \hat{\mathbf{w}})$$

- **Rand index** between blocks: it exists also a recent definition...

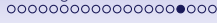
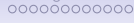


Block estimation: consistency

[Mariadassou and Matias, 12]

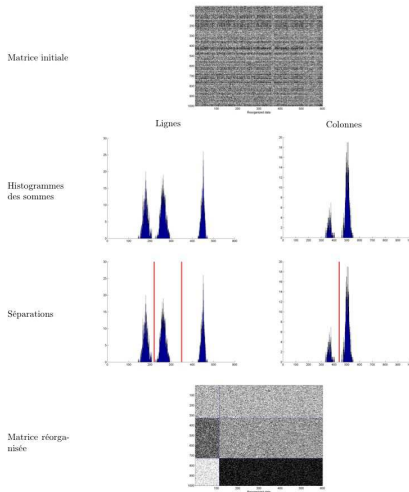
$$\underbrace{\hat{\theta} \xrightarrow{n, d \rightarrow \infty} \theta^*}_{\text{we have seen that...}} \Rightarrow \underbrace{p(\hat{\mathbf{z}} = \mathbf{z}^*, \hat{\mathbf{w}} = \mathbf{w}^* | \mathbf{x}; \hat{\theta}) \xrightarrow{n, d \rightarrow \infty} 1}_{\text{exact bi-partition retrieval!}}$$

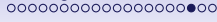
Thus we retrieve the HD clustering blessing...



Block estimation: non asymptotic properties (1/2)

Binary case: marginals seems so **simple mixtures!** [Brault, 14]





Block estimation: non asymptotic properties (2/2)

[Brault, 14]

- Probability of x_{ij} with no regard to the column membership is Bernoulli

$$p(x_{ij} = 1 | z_{ik} = 1) = \tau_k = \sum_{l=1}^L \alpha_{kl} \rho_l$$

- Thus marginal distribution of x_{ij} is a mixture (indep. of x_{ij} cond. $z_{ik} = 1$)

$$\left(\sum_j x_{ij} \right) | z_{ik} = 1 \sim B(d, \tau_k)$$

- Control of error on this partition mixture estimate $\hat{\mathbf{z}}^{mix}$ of binomial distributions

$$p(\hat{\mathbf{z}}^{mix} \neq \mathbf{z}^*) \leq 2n \exp \left\{ - \frac{1}{8} d \underbrace{\left[\min_{k \neq k'} |\tau_k - \tau_{k'}| \right]}_{\text{overlap}} \right\} + K(1 - \min_k \pi_k)^n$$

- We retrieve also consistency for very high dimension with constraint

$$\ln(n) = o(d)$$

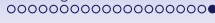
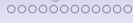
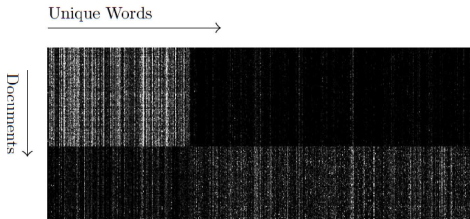


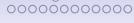
Illustration: document clustering (2/2)



Results with 2×2 blocks

	Medline	Cranfield
Medline	1033	0
Cranfield	0	1398

Experiment illustrates previous theory: HD clustering is blessing



Outline

1 HD clustering

2 Modeling

3 Estimating

4 Selecting

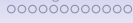
5 Experiments

6 To go further



Models in competition

$\mathbf{m} = (K, L)$ typically, but not restricted to



BIC criterion: two difficulties

- **Difficult 1:** which BIC definition because of the double asymptotic on n and d ?
- **Difficult 2:** the observed log-likelihood value is intractable

$$\ell(\boldsymbol{\theta}; \mathbf{x}) = \sum_{(\mathbf{z}, \mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} p(\mathbf{x}, \mathbf{z}, \mathbf{w}; \boldsymbol{\theta})$$

Could be estimated by harmonic mean but time consuming and high variance



ICL criterion: consistency

- We can obtain a BIC expression from ICLbic

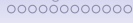
$$\begin{aligned} \text{BIC} &= \text{ICLbic} - \ln p(\hat{\mathbf{z}}, \hat{\mathbf{w}} | \mathbf{x}; \hat{\boldsymbol{\theta}}) \\ &= \underbrace{\ell(\hat{\boldsymbol{\theta}}; \mathbf{x})}_{\text{difficult}} - \frac{K-1}{2} \ln(n) - \frac{L-1}{2} \ln(d) - \frac{KL(m-1)}{2} \ln(nd) \end{aligned}$$

- [Brault *et al.*, 2017] establish that asymptotically on n and d

$$“\ell(\hat{\boldsymbol{\theta}}; \mathbf{x}) = \ell_c(\hat{\boldsymbol{\theta}}; \mathbf{x}, \hat{\mathbf{z}}, \hat{\mathbf{w}})”$$

- Thus, since BIC is consistent, ICL is also consistent

Again the HD clustering blessing is here!



Strategy to smart browsing of (K, L)

[Robert, 2017] Algorithm Bi-KM1

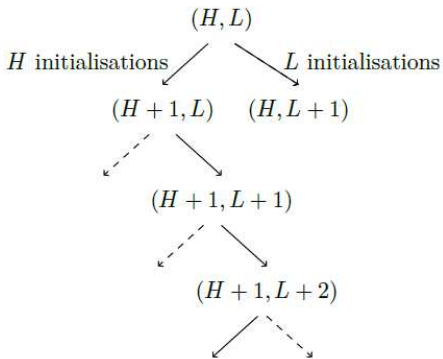




Illustration: discuss the dimension (1/2)

- SPAM E-mail Database⁴
- $n = 4601$ e-mails composed by 1813 “spams” and 2788 “good e-mails”
- $d = 48 + 6 = 54$ continuous descriptors⁵
 - 48 percentages that a given **word** appears in an e-mail (“make”, “you’...)
 - 6 percentages that a given **char** appears in an e-mail (“;”, “\$”...)
- Transformation of continuous descriptors into **binary descriptors**

$$x_{ij} = \begin{cases} 1 & \text{if word/char } j \text{ appears in e-mail } i \\ 0 & \text{otherwise} \end{cases}$$

⁴<https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/>

⁵There are 3 other continuous descriptors we do not use

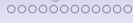
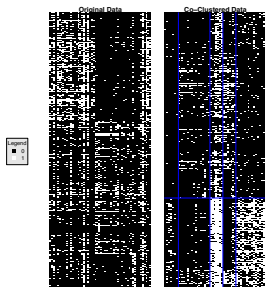


Illustration: discuss the dimension (2/2)

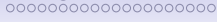
- Perform **co-clustering** with $K = 2$ and $L = 5$: ICL=-92,682, err=0.1984



- Perform **clustering**⁶ with $K = 2$: ICL=-89,433, err=0.1837

Thus use preferably co-clustering in the HD setting

⁶Equivalent to co-clustering with $L = 54$



Outline

1 HD clustering

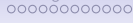
2 Modeling

3 Estimating

4 Selecting

5 Experiments

6 To go further



A synthetic example

Blockcluster basic example

Data

Load the CSV data file as dataframe.

```
In [3]: data <- as.matrix(read.table("blockcluster-example.csv", sep = ","))
load(data)
```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	...	V91	V92	V93	V94	V95	V96	V97	V98	V99	V100	
A matrix: 6 × 100																						
of type int	1	1	1	0	1	1	1	1	1	0	...	0	0	0	0	0	0	1	0	1	1	
	0	0	1	0	0	0	0	0	0	1	...	0	1	0	0	1	1	1	1	0	0	
	0	1	1	0	1	1	1	1	1	0	...	0	0	1	1	1	0	1	0	0	0	
	0	0	0	0	1	0	1	1	0	1	...	1	1	1	0	0	1	1	0	0	0	
	0	0	0	1	1	1	1	1	1	0	...	0	0	0	0	1	1	0	0	0	1	
	1	1	1	1	0	0	1	1	0	1	...	1	1	0	0	1	1	1	0	0	0	

Clustering with Blockcluster

Launch the BlockCluster package.

```
In [4]: library(blockcluster)
```

```
Loading required package: rtkore
Loading required package: Rcpp
Attaching package: 'rtkore'

The following object is masked from 'package:Rcpp':

  LdFlags

blockcluster version 4.4.3 loaded
-----
Copyright (C) 2008-2018 team INRIA, Lille & U.M.R.S. C.N.R.S. 6599 Neudislayc, UTC
Please post questions and bugs at: <https://gforge.inria.fr/forum/forum.php?forum_id=11190&group_id=3479>
```

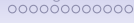
Define the strategy.

Specify the number of groups/clusters to create in rows and columns.

```
In [5]: rowClusters <- 2
columnClusters <- 3
```

Run the coclustering.

```
In [6]: res <- cocluster(data, dstatype = "binary", nbrowcluster = c(rowClusters, columnClusters))
Co-Clustering successfully terminated!
```



A synthetic example

Output's Analysis

Model criterion

This chart represents the criterion value for each model that was built. The lower the value (close to 0) the better the model.

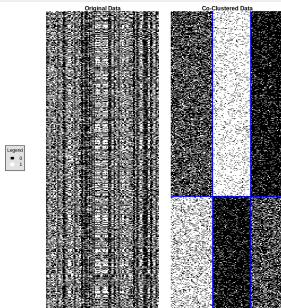
```
In [7]: res#ICLvalue
```

```
-45557.0741834614
```

Cluster Plot

Draw the original data matrix, and the matrix obtained after performing co-clustering

```
In [8]: plot(res)
```





Outline

1 HD clustering

2 Modeling

3 Estimating

4 Selecting

5 Experiments

6 To go further

