Model-based clustering and classification

part 2: some specific mixture models

Julien JACQUES

Université Lumière Lyon 2

Mixture model for Gaussian data

Model selection

Mixture model for non Gaussian data

Mixture model for Gaussian data

The Gaussian Mixture Model

The density of group k is

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x} - \mu_k)^t \Sigma_k^{-1}(\boldsymbol{x} - \mu_k)\}$$

where

- μ_k is the mean vector
- Σ_k the covariance matrix of group k
- $|\Sigma_k|$ denotes the determinant of Σ_k

Complexity of the Gaussian Mixture Model

Number of model parameters

- Σ_k: $K \times p(p+1)/2$
- μ_k: $K \times p$
- p_k: $K - 1$ (since $\sum_{k=1}^{K} p_k = 1$)
- total: $K(p(p+1)/2 + p + 1) - 1$
- example:
  - $K = 3$, $p = 10 \Rightarrow 197$
  - $K = 6$, $p = 100 \Rightarrow 30905$

There is a need to **reduce the number of parameters** in order to avoid over-fitting

Parsimonious Gaussian Mixture Model

Most of the parameters are dedicated to the variance matrices Σ_k.

In order to reduce the number of parameters, we can for instance assume:

- $\Sigma_k = \Sigma \quad \forall 1 \leq k \leq K$
- impact:
  - $K = 3$, $p = 10 \Rightarrow 87$ parameters (vs 197)
  - $K = 6$, $p = 100 \Rightarrow 5655$ parameters (vs 30905)

This model is knwon as **Linear Discriminant Analysis** (LDA). (*warning: $\neq$ Fisher linear discriminant analysis* which looks for a discriminative subspace)

The full Gaussian mixture with Σ_k free is knwon as **Quadratic Discriminant Analysis** (QDA).

LDA/QDA separating surface for 2 classes

The Bayes optimal **separating surface** is $g(\boldsymbol{x}) = \frac{C(2,1)t_1(\boldsymbol{x})}{C(1,2)t_2(\boldsymbol{x})} = 1$

For **QDA** ($\Sigma_1 \neq \Sigma_2$), $g(\boldsymbol{x}) = 1$ is equivalent to

$$\begin{aligned}
\ln g(\boldsymbol{x}) &= \ln \frac{C(2,1)p_1 f_1(\boldsymbol{x})}{C(1,2)p_2 f_2(\boldsymbol{x})} \\
&= \ln \frac{f_1(\boldsymbol{x})}{f_2(\boldsymbol{x})} + \underbrace{\ln \frac{C(2,1)p_1}{C(1,2)p_2}}_{s} \\
&= \frac{1}{2}\left(\ln \frac{|\Sigma_2|}{|\Sigma_1|} + (\boldsymbol{x} - \mu_2)^t \Sigma_2^{-1}(\boldsymbol{x} - \mu_2) - (\boldsymbol{x} - \mu_1)^t \Sigma_1^{-1}(\boldsymbol{x} - \mu_1)\right) + s.
\end{aligned}$$

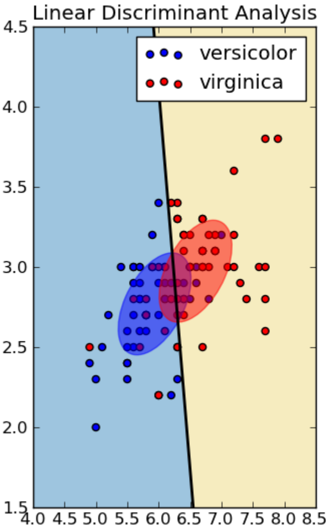
which is **quadratic** in $\boldsymbol{x}$.

For **LDA** ($\Sigma_1 = \Sigma_2 = \Sigma$):

$$\ln g(\boldsymbol{x}) = (\mu_1 - \mu_2)^t \Sigma^{-1}(\boldsymbol{x} - \frac{\mu_1 + \mu_2}{2}) + s,$$

which is **linear** in $\boldsymbol{x}$.

LDA/QDA separating surface for 2 classes



Parsimonious Gaussian Mixture Model

More parsimonious GMM have been introduced by considering the spectral decomposotion of Σ_k:

- Banfield, J. D. and Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. Biometrics, 49(3), 803–821.
- Celeux, G. and Govaert, G. (1995). Gaussian parcimonious models. Pattern Recognition, 28(5), 781–793.

Parsimonious Gaussian Mixture Model

Spectral decomposition

$$\Sigma_k = \lambda_k D_k A_k D_k^t$$

where

- λ_k: largest eigenvalue
- D_k: orthogonal matrix of eigenvectors
- A_k: diagonal matrix of normalized eigenvalues, such that $A_k = diag(a_{1k}, \ldots, a_{pk})$ with $1 = a_{1k} \geq \ldots \geq a_{pk}$

Interpretation:

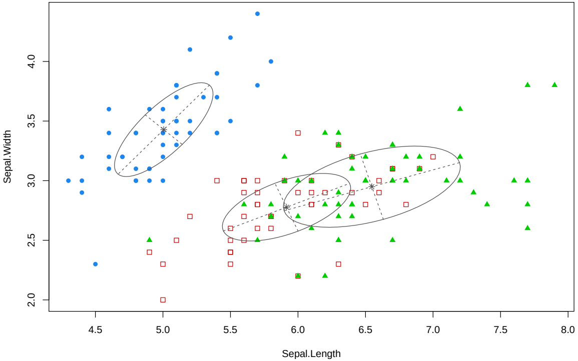
- λ_k: volume of (*space occupied by*) group k
- D_k: orientation of group k
- A_k: shape of group k

Restrictions on λ_k, D_k, $A_k \Rightarrow$ parsimonious models

Parsimonious Gaussian Mixture Model

Example of GMM with equal shape (A_k) and different volume and orientation (λ_k,D_k)

```
library(mclust)
mod1 <- Mclust(iris[,1:4],G=3,modelNames = "VEV")
plot(mod1,"classification",dimens = 1:2)
```



Parsimonious GMM - mclust

`Mclust` model names:

Model	$\boldsymbol{\Sigma}_k$	Distribution	Volume	Shape	Orientation
EII	$\lambda \boldsymbol{I}$	Spherical	Equal	Equal	—
VII	$\lambda_k \boldsymbol{I}$	Spherical	Variable	Equal	—
EEI	$\lambda \boldsymbol{A}$	Diagonal	Equal	Equal	Coordinate axes
VEI	$\lambda_k \boldsymbol{A}$	Diagonal	Variable	Equal	Coordinate axes
EVI	$\lambda \boldsymbol{A}_k$	Diagonal	Equal	Variable	Coordinate axes
VVI	$\lambda_k \boldsymbol{A}_k$	Diagonal	Variable	Variable	Coordinate axes
EEE	$\lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$	Ellipsoidal	Equal	Equal	Equal
EVE	$\lambda \boldsymbol{D}\boldsymbol{A}_k\boldsymbol{D}^\top$	Ellipsoidal	Equal	Variable	Equal
VEE	$\lambda_k \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$	Ellipsoidal	Variable	Equal	Equal
VVE	$\lambda_k \boldsymbol{D}\boldsymbol{A}_k\boldsymbol{D}^\top$	Ellipsoidal	Variable	Variable	Equal
EEV	$\lambda \boldsymbol{D}_k\boldsymbol{A}\boldsymbol{D}_k^\top$	Ellipsoidal	Equal	Equal	Variable
VEV	$\lambda_k \boldsymbol{D}_k\boldsymbol{A}\boldsymbol{D}_k^\top$	Ellipsoidal	Variable	Equal	Variable
EVV	$\lambda \boldsymbol{D}_k\boldsymbol{A}_k\boldsymbol{D}_k^\top$	Ellipsoidal	Equal	Variable	Variable
VVV	$\lambda_k \boldsymbol{D}_k\boldsymbol{A}_k\boldsymbol{D}_k^\top$	Ellipsoidal	Variable	Variable	Variable

Table 3: Parameterisations of the within-group covariance matrix $\boldsymbol{\Sigma}_k$ for multidimensional data available in the **mclust** package, and the corresponding geometric characteristics.

From *Scrucca L., Fop M., Murphy T. B. and Raftery A. E. (2016) mclust 5: clustering, classification and density estimation using Gaussian finite mixture models, The R Journal, 8/1, pp. 205-233.*

Parsimonious GMM - mclust

`Mclust` model names:

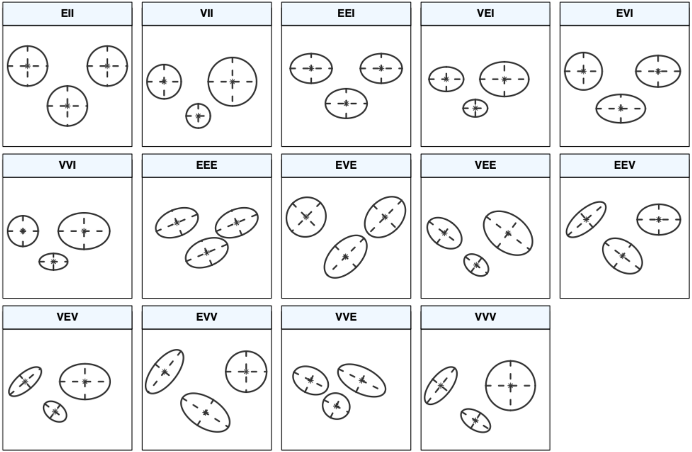


Figure 2: Ellipses of isodensity for each of the 14 Gaussian models obtained by eigen-decomposition in case of three groups in two dimensions.

From *Scrucca L., Fop M., Murphy T. B. and Raftery A. E. (2016) mclust 5: clustering, classification and density estimation using Gaussian finite mixture models, The R Journal, 8/1, pp. 205-233.*

Parsimonious GMM

There exist a lot of other parsimonious GMM for high dimensional data.

For a survey, have a look to:

C. Bouveyron and C. Brunet, Model-based clustering of high-dimensional data: A review, Computational Statistics and Data Analysis, vol. 71, pp. 52-78, 2014.

We will see one of them, described in:

C. Bouveyron, S. Girard and C. Schmid, High-Dimensional Data Clustering, Computational Statistics and Data Analysis, vol. 52 (1), pp. 502-519, 2007.

C. Bouveyron, S. Girard and C. Schmid, High Dimensional Discriminant Analysis, Communications in Statistics: Theory and Methods, vol. 36 (14), pp. 2607-2623, 2007.

High-Dimensional Data Clustering (HHDC)

Let's go back to the spectral decomposition, whitout factoring by λ_k

$$\Sigma_k = D_k A_k D_k^t$$

where

- D_k: orthogonal matrix of eigenvectors
- A_k: diagonal matrix of eigenvalues, in which the $p - d_k$ latest eignevalues are assumed to be equal

$$A_k = \left(\begin{array}{cc} \begin{array}{|ccc|} \hline a_{k1} & & 0 \\ & \ddots & \\ 0 & & a_{kd_k} \\ \hline \end{array} & \mathbf{0} \\ \mathbf{0} & \begin{array}{|ccc|} \hline b_k & & 0 \\ & \ddots & \\ 0 & & b_k \\ \hline \end{array} \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c} a \\ \ddots \\ a \end{array}}\right\} d_k \\ \left.\vphantom{\begin{array}{c} b \\ \ddots \\ b \end{array}}\right\} (m - d_k) \end{array}$$

High-Dimensional Data Clustering (HHDC)

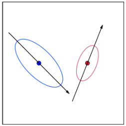
With this assumption:

- the last $p - d_k$ dimensions are assumed to correspond to noise, and consequently modelled with only 1 parameter
- allows to avoir computing the last $p - d_k$ columns of D_k

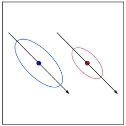
Several submodels are defined:

Model	Number of parameters
$[a_{kj}b_kQ_kd_k]$	$\rho + \bar{\tau} + 2K + D$
$[a_{kj}bQ_kd_k]$	$\rho + \bar{\tau} + K + D + 1$
$[a_kb_kQ_kd_k]$	$\rho + \bar{\tau} + 3K$
$[ab_kQ_kd_k]$	$\rho + \bar{\tau} + 2K + 1$
$[a_kbQ_kd_k]$	$\rho + \bar{\tau} + 2K + 1$
$[abQ_kd_k]$	$\rho + \bar{\tau} + K + 2$
$[a_{kj}b_kQ_kd]$	$\rho + K(\tau + d + 1) + 1$
$[a_{kj}bQ_kd]$	$\rho + K(\tau + d) + 2$
$[a_kb_kQ_kd]$	$\rho + K(\tau + 2) + 1$
$[ab_kQ_kd]$	$\rho + K(\tau + 1) + 2$
$[a_kbQ_kd]$	$\rho + K(\tau + 1) + 2$
$[abQ_kd]$	$\rho + K\tau + 3$
$[a_jbQd]$	$\rho + \tau + d + 2$
$[abQd]$	$\rho + \tau + 3$

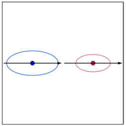
High-Dimensional Data Clustering (HHDC)



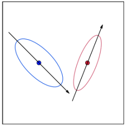
modèle $[a_i b_i Q_i d]$



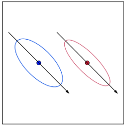
modèle $[a_i b_i Q d]$



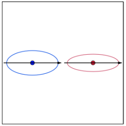
modèle $[a_i b_i I_2 d]$



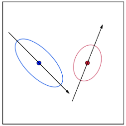
modèle $[a b_i Q_i d]$



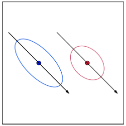
modèle $[a b_i Q d]$



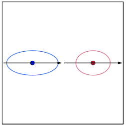
modèle $[a b_i I_2 d]$



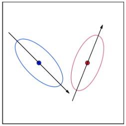
modèle $[a_i b Q_i d]$



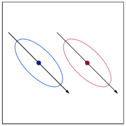
modèle $[a_i b Q d]$



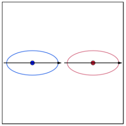
modèle $[a_i b I_2 d]$



modèle $[a b Q_i d]$



modèle $[a b Q d]$



modèle $[a b I_2 d]$

High-Dimensional Data Clustering (HHDC)

```
library(HDclassif)
hddc(iris[,1:4],model="all")
```

```
## HIGH DIMENSIONAL DATA CLUSTERING
## MODEL: AKBKQKD
##   Posterior probabilities of groups
##          1     2     3
##     0.301 0.333 0.366
##         Intrinsic dimensions of the classes:
##          1 2 3
##   dim: 1 1 1
##           1     2     3
## Ak: 0.508 0.232 0.745
##            1      2      3
## Bk: 0.0366 0.0238 0.0659
## BIC:  -588.5989
```

Model selection

Model selection

We have defined several parsimonious models, and we have to choose between them.

- In classification, model selection can be done by:
  - training/test,
  - cross-validation.
- In clustering, it is harder since no partition is known
  - likelihood can not be used since it increases with model complexity
  - model selection criterion can be used: AIC, BIC, ICL

E. Lebarbier and T. Mary-Huard (2004), Le critère BIC: fondements théoriques et interprétation, Rapport de Recherche Inria n°0249-6399

Model selection in classification

In (supervised) classification, variable selection is of primary interest.

So we have to model selection problems:

- choosing among parsimonious models
- choosing which variable to introduce in the model

Since a labeled sample is available $(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}})$, model selection can be done by:

- training / test
- cross validation

For variables selection problem, iterative algorithms should be employed in order to test only a reduced number of combinations (forward / backward / stepwise)

Model selection in clustering

In (unsupervised) clustering, the model selection task is harder since there is no labeled sample.

We have to select:

- the **number of clusters**
- the **parsimonious model** to use
- the variable to introduce in the model

For variable selection, which is of *secondary interest* in clustering, refer to chapter 5 of the MBCC book 

The two first tasks can be seen as a **model selection** problem:

$$m = \{f(\boldsymbol{x}; \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$$

Biais-variance tradeoff

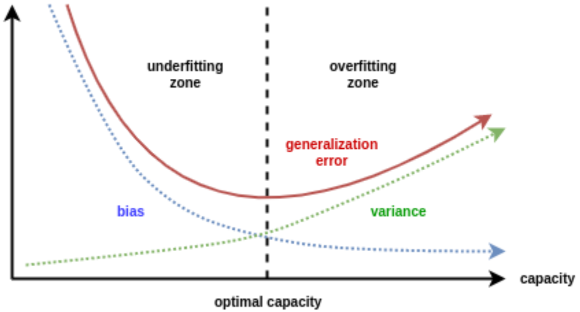
A too simple model will have:

- low variance but large biais

A too complex model will have:

- low biasis but large variance

We have to select a model with the **best biais-variance tradeoff**



Model selection using hypothesis testing

We can use **maximum likelihood ratio test**:

$H_0 : \boldsymbol{\theta} \in \boldsymbol{\Theta}_0$ versus $H_1 : \boldsymbol{\theta} \in \boldsymbol{\Theta}_1 = \boldsymbol{\Theta} \setminus \boldsymbol{\Theta}_0$

for which the reject region is:

$$W = \{\underline{\boldsymbol{x}} : -2 \ln \frac{\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta_0}} l(\boldsymbol{\theta}, \underline{\boldsymbol{x}})}{\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta_1}} l(\boldsymbol{\theta}, \underline{\boldsymbol{x}})} > \chi^2_{p-p_o,1-\alpha}\}$$

where p and p_0 are the dimension of $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}_0$

Example: $\boldsymbol{\Theta}$ is a full GMM, $\boldsymbol{\Theta}_0$ is a constrained GMM.

Limitation of this approach:

- difficult to test more than 2 models (tests are not transitive)
- need to embedded models

Frequentist model selection

Idea: to select model $f_{\hat{\boldsymbol{\theta}}}$ minimizing entropic cost (or **deviance** D):

$$E_{\underline{\boldsymbol{x}}}[KL(f, f_{\hat{\boldsymbol{\theta}}})] = \frac{1}{2} D$$

where KL the Kullback-Leibler divergence:

$$KL(f, f_{\hat{\boldsymbol{\theta}}}) = \int \ln\left(\frac{f(\underline{\boldsymbol{y}})}{f_{\hat{\boldsymbol{\theta}}}(\underline{\boldsymbol{y}})}\right) f(\underline{\boldsymbol{y}}) d\underline{\boldsymbol{y}}$$

This criterion is intractable since it depends on f which is unknown.

AIC criterion

But a 2-order Taylor development allows to additivelly separate f and $\hat{\boldsymbol{\theta}}$:

$$D = 2\{\ln f(\underline{\boldsymbol{x}}) - \ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}})\} + 2\nu + Op(\sqrt{n})$$

where ν is the number of model parameter (*if the true model belong to the tested ones...*)

Since $\ln f(\underline{\boldsymbol{x}})$ is constant for every models, minimizing the deviance is approximately equivalent to minimizing:

$$AIC = -2\ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}}) + 2\nu$$

Theoretical properties:

- Cross-validation criterion $\sum_i \ln f(x_i, \hat{\boldsymbol{\theta}}_{-i})$ tends to AIC when $n \to \infty$
- if we compare 2 models $M_1 \subset M_2$ and if M_2 is the true one, AIC will choose it.

Bayesian model selection

Idea: to select model M maximizing its posterior probability:

$$p(M|\underline{\boldsymbol{x}}) \propto p(\underline{\boldsymbol{x}}|M)p(M)$$

Assuming that all models are *a priori* equivalent, $p(M)$ is constant.

Bayesian model selection leads to maximize the **integrated likelihood**:

$$p(\underline{\boldsymbol{x}}|M) = \int_{\boldsymbol{\Theta}} p(\underline{\boldsymbol{x}}|M, \boldsymbol{\theta})p(\boldsymbol{\theta}|M)d\boldsymbol{\theta}$$

where $p(\underline{\boldsymbol{x}}|M, \boldsymbol{\theta}) = f(\underline{\boldsymbol{x}}, \boldsymbol{\theta})$ is the likelihood (over model M).

Computing this criterion leads to select the prior distribution $p(\boldsymbol{\theta}|M)$...

BIC criterion

To avoid choosing $p(\boldsymbol{\theta}|M)$, a Laplace approximation can be used to approximate the intregrated likelihood:

$$\ln p(\underline{\boldsymbol{x}}|M) = \ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}}) - \frac{\nu}{2} \ln n + 0_p(1)$$

where $\ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}})$ is the log-likelihood.

Maximizing the intregrated likelihood is equivalent to maximizing the **BIC criterion**:

$$BIC = \ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}}) - \frac{\nu}{2} \ln n$$

Theoretical properties:

- if we compare 2 models $M_1 \in M_2$, BIC will asymptotically choose the true one.
- BIC is consitant for choosing the number K of clusters

BIC in practice

In order to mimic the AIC expression, the previous BIC expression is sometimes multiplyied by 2 or -2.

- for instance in Mclust: $BIC = 2\ell(\hat{\boldsymbol{\theta}}, \underline{\boldsymbol{x}}) - \nu \ln n$ and should be maximized

So, before to use BIC provided by any package, check how it is defined...

Another Bayesian information criterion

AIC and BIC aim to look for the **true model**

"All models are wrong, but some are useful" G. Box, 1976

In clustering, we look for well separeted clusters.

On way to achieve this is to maximze the **integrated complete Likelihood**:

$$p(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}|M) = \int_{\boldsymbol{\Theta}} p(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}|M, \boldsymbol{\theta}) p(\boldsymbol{\theta}|M) d\boldsymbol{\theta}$$

where $p(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}|M, \boldsymbol{\theta})$ is the complete likelihood.

ICL criterion

As BIC, using a Laplace approximation, $\ln p(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}|M)$ can be approximated by the ICL criterion:

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

Remark:

- ICL penalized the BIC criterion by the mean entropy:

$$ICL = BIC - \sum_i \sum_k t_{ik}(\hat{\boldsymbol{\theta}}) \ln t_{ik}(\hat{\boldsymbol{\theta}})$$

consequently, it leads to select more separated clusters.

Mixture model for non Gaussian data

Categorical nominal features: the multinomial mixture

The Multinomial Model

- each categorical feature X_j is coded as follows:

$$X_j = (X_j^1, \dots, X_j^{m_j})$$

  with $X_j^h = 1$ if the j-th categorical feature takes the h-th category, 0 otherwise.
- the full multinomial model (for group k) is defined by probabilities:

$$f_k(\boldsymbol{x}) = p(x_1^{h_1} = 1, \dots, x_p^{h_p} = 1 | Z = k) = \alpha_k^{h_1 \dots h_p}$$

- number of parameters per cluster: $\prod_{j=1}^{p} m_j - 1$
  - ex: 10 features with 5 categories $\Rightarrow 5^{10} - 1$ (per cluster)
- this model is never used due to its too large number of parameters

The Latent Class Model

The Latent Class Model assumes that the categorical feature are independent conditionally to Z

$$
\begin{aligned}
f_k(\boldsymbol{x}) &= p(x_1^{h_1} = 1, \ldots, x_p^{h_p} = 1 | Z = k) \\
&= \prod_{j=1}^{p} p(x_j^{h_j} = 1 | Z = k) \\
&= \prod_{j=1}^{p} \prod_{h=1}^{m_j} (\alpha_k^{jh})^{x_j^h}
\end{aligned}
$$

- number of parameters per cluster: $\sum_{j=1}^{p}(m_j - 1)$
  - 10 features with 5 categories $\Rightarrow$ 40 parameters (per cluster)
- the marginal distribution is:

$$
f(\boldsymbol{x}) = \sum_{k=1}^{K} p_k \prod_{j=1}^{p} \prod_{h=1}^{m_j} (\alpha_k^{jh})^{x_j^h}
$$

The Latent Class Model

- more parsimonious models can be considered that for each X_j, only the probability of the majority category is free (all the others categories are assumed to be equally distributed)

Latent Class Model estimation in classification

Latent Class Model assumes that the categorical features are independent conditionally to Z:

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} p_k \prod_{j=1}^{p} \prod_{h=1}^{m_j} (\alpha_k^{jh})^{x_j^h}$$

Maximum likelihood estimation:

$$\hat{\alpha}_k^{jh} = \frac{1}{n_k} \sum_{i=1}^{n} \tilde{z}_{ik} x_{ij}^h$$

Exercice 1

Prove the expression of the maximum likelihood estimator for the Latent Class model.

Latent Class Model estimation in clustering

EM algorithm:

- E step: computation of

$$E[\tilde{z}_{ik}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(q)}] = t_k^{(q)}(\boldsymbol{x}_i) = \frac{p_k \prod_{j=1}^{p} \prod_{h=1}^{m_j} (\alpha_k^{jh(q)})^{x_{ij}^h}}{\sum_{\ell=1}^{K} p_\ell \prod_{j=1}^{p} \prod_{h=1}^{m_j} (\alpha_\ell^{jh(q)})^{x_{ij}^h}}$$

- M step: maximisation of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ according to $\boldsymbol{\theta}$:
  - $\hat{p}_k = \frac{n_k^{(q)}}{n}$ with $n_k = \sum_{i=1}^{n} t_k^{(q)}(\boldsymbol{x}_i)$
  - $\alpha_k^{jh(q+1)} = \frac{1}{n_k} \sum_{i=1}^{n} t_k^{(q)}(\boldsymbol{x}_i) x_{ij}^h$

Exercice 2

Implement an EM algorithm for estimating the latent class model.

Test it for the clustering of simulated categorical data set.

Latent Class Model with R

The Credit data set has 66 rows and 11 columns, describing customers who took out loans from a credit company described with 11 categorical or ordinal variables.

```
library(Rmixmod)
library(FactoMineR)
library(MBCbook)
data(credit)
X = credit
X$Age = as.factor(X$Age)
```

Latent Class Model with R

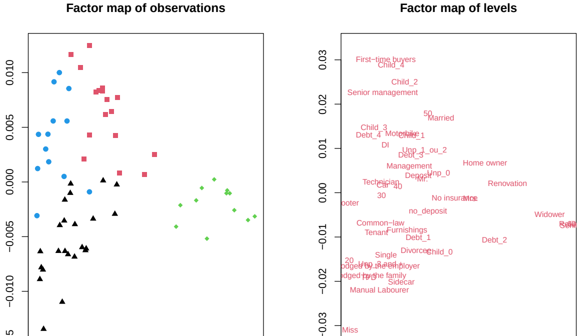
Clustering with LCM through Rmixmod

```
res = mixmodCluster(X,nbCluster=1:8,dataType="qualitative",
  model=mixmodMultinomialModel(listModels="Binary_pk_Ekj"),
  criterion = c("BIC","ICL"))
```

Latent Class Model with R

Visualization of the best result according to BIC (in the first factorial plane of MCA)

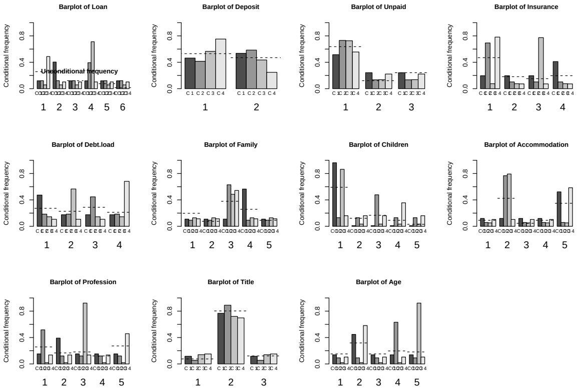
```
par(mfrow=c(1,2))
lbl = res@results[[1]]@partition
acm = mca(X,abbrev = TRUE)
plot(predict(acm,X),col=lbl,
     pch=c(17,15,18,19)[lbl],xlab='',ylab='',
     main="Factor map of observations")
plot(acm,rows=F,cex=0.75,main="Factor map of levels")
```



Latent Class Model with R

Barplot of cluster-conditional level frequency

```r
par(mfrow=c(1,1))
barplot(res)
```



Categorical ordinal features

Ordinal data

An **ordinal** variable μ takes values among m *full ordered* categories

$$\mu \in \{1, \ldots, m\} \text{ with } 1 < \ldots < m$$

Model for ordinal data

Ordinal data are often considered as:

- nominal data (cf. the example with the previous credit data): information order is lost
- continous data, by given arbitrary integer to each category: arbitrary distance is introduce

It is preferable to use specific model for ordinal data:

- BOS model
- Latent variable model

J.Jacques and C.Biernacki (2016), Model-based clustering of multivariate ordinal data relying on a stochastic binary search algorithm, Statistics and Computing, 26 [5], 929-943.

D.McParland and C.Gormleu (2016), Model-based clustering for mixed data: clustMD. Advances in Data Analysis and Classification, 10, 155-169.

The BOS model

The BOS model has been defined on the basis that an ordinal data results from a *dichotomic search algorithm* in $\{1,\dots,m\}$

- Principle: relies on comparisons $\{<,=,>\}$

$$e_1=\{1,\dots,m\}\to y_1\to e_2\to\dots\to y_{m-1}\to e_m=\{\mu\}.$$

- Example: search for $\mu=3$ in $\{1,2,3,4\}$

Step	search interval	middle value	comparisons
1	$\underbrace{\boxed{1}\boxed{2}\boxed{3}\boxed{4}}_{e_1}$	$\boxed{1}\ \underbrace{\boxed{2}}_{y_1}\ \boxed{3}\boxed{4}$	$\mu \overset{?}{<} \boxed{2}$ $\mu \overset{?}{=} \boxed{2}$ $\underbrace{\mu \overset{?}{>} \boxed{2}}_{\text{good}}$
2	$\underbrace{\boxed{3}\boxed{4}}_{e_2}$	$\underbrace{\boxed{3}}_{y_2}\ \boxed{4}$	$\mu \overset{?}{<} \boxed{3}$ $\underbrace{\mu \overset{?}{=} \boxed{3}}_{\text{good}}$ $\mu \overset{?}{>} \boxed{3}$

- More efficient than a sequential search: $\{1,\dots,m\}$ ordered

Randomized search algorithm

Idea 1: random in comparisons

Wrong comparisons results are possible in the search algorithm:
Step j:

- $z_j = 1$: exact comparisons (as exact algo.)
- $z_j = 0$: choose randomly the value e_{j+1} (no comparison)

$\Rightarrow z_j \sim \mathcal{B}(\pi)$

Idea 2: random in middle value

Choose uniformly y_j in e_j (and not only the middle)

Associated algorithm

- Idea 1 + Idea 2
- Principle:

$e_1 = \{1, \dots, m\} \rightarrow y_1 \rightarrow z_1 \rightarrow e_2 \rightarrow \dots \rightarrow y_{m-1} \rightarrow z_{m-1} \rightarrow e_m = \{x\}$

Marginal probabilities

- Marginal on z_j's

  $$p(e_{j+1}|e_j, y_j; \mu, \pi) = \pi p(e_{j+1}|y_j, e_j, z_j = 1; \mu) + (1-\pi) p(e_{j+1}|y_j, e_j, z_j = 0)$$

- Marginal on y_j's

  $$p(e_{j+1}|e_j; \mu, \pi) = \sum_{y_j \in e_j} p(e_{j+1}|e_j, y_j; \mu, \pi) p(y_j|e_j)$$

- Marginal on e_j's ($x \in \{1, \dots, m\}$)

  $$\begin{aligned} p(x; \mu, \pi) &= \sum_{e_{m-1}, \dots, e_1} p(e_m, e_{m-1}, \dots, e_1; \mu, \pi) \\ &= \sum_{e_{m-1}, \dots, e_1} \prod_{j=1}^{m-1} p(e_{j+1}|e_j; \mu, \pi) p(e_1) \end{aligned}$$

$$= \underbrace{\sum_{e_{m-1}} \{p(e_m|e_{m-1}; \mu, \pi) \underbrace{\sum_{e_{m-2}} \{p(e_{m-1}|e_{m-2}; \mu, \pi) \dots \underbrace{\sum_{e_1} \{p(e_2|e_1; \mu, \pi) p(e_1)\}\}\}}_{p(e_2; \mu, \pi)}}_{p(e_{m-1}; \mu, \pi)}}$$

Polynomial expression of the probabilities

Example for $m = 5$, $\mu = 2$:

$$p(1;\mu,\pi) = \frac{1}{5} - \frac{49}{600}\pi - \frac{263}{2400}\pi^2 - \frac{31}{3600}\pi^3 - \frac{1}{7200}\pi^4$$

$$p(2;\mu,\pi) = \frac{1}{5} + \frac{17}{30}\pi + \frac{379}{1800}\pi^2 + \frac{1}{45}\pi^3 + \frac{1}{1800}\pi^4$$

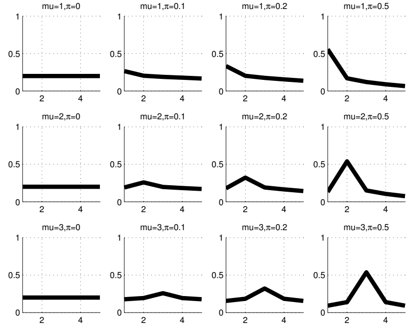
$$p(3;\mu,\pi) = \frac{1}{5} - \frac{1}{75}\pi - \frac{31}{200}\pi^2 - \frac{109}{3600}\pi^3 - \frac{1}{720}\pi^4$$

$$p(4;\mu,\pi) = \frac{1}{5} - \frac{33}{200}\pi - \frac{457}{7200}\pi^2 + \frac{2}{75}\pi^3 + \frac{13}{7200}\pi^4$$

$$p(5;\mu,\pi) = \frac{1}{5} - \frac{23}{75}\pi + \frac{47}{400}\pi^2 - \frac{1}{100}\pi^3 - \frac{1}{1200}\pi^4$$

Properties of $p(x;\mu,\pi)$

- μ: **position** parameter (unique mode if $\pi > 0$)
- **monotonic decrease** around μ
- π: **precision** parameter:
  - $p(\mu;\mu,\pi)$ increases with π
  - $p(\mu;\mu,\pi) - p(x;\mu,\pi)$ increases with π ($x \neq \mu$)
  - uniform distribution if $\pi = 0$
  - Dirac in μ if $\pi = 1$
- identifiability (if $\pi = 0$)



ML estimation of (μ, π)

- $(x_1, \ldots, x_n) \overset{iid}{\sim} p(.; \mu, \pi)$
- Model with **latent variables**

$$c_i = \{e_{ij}, y_{ij}, z_{ij}\}_{j=1,\ldots,m}$$

- Maximum likelihood can be performed by an **EM algorithm**
  - **E Step**: for all $c_i \in C_i$ $(i = 1, \ldots, n)$

$$p(c_i | x_i; \mu^{(q)}, \pi^{(q)}) = p(c_i, x_i; \mu^{(q)]}, \pi^{(q)}) / p(x_i; \mu^{(q)}, \pi^{(q)}).$$

  - **M Step**: maximization s.t. $\mu^{[q+1]} \in \{1, \ldots, m\}$ of the expected conditional completed log-likelihood

$$\sum_{i=1}^{n} \sum_{c_i \in C_i} p(c_i | x_i; \mu^{[q]}, \pi^{[q]}) \ln p(x_i, c_i; \mu^{[q+1]}, \pi^{[q+1]})$$

where $\pi^{[q+1]} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m-1} p(z_{ij}=1 | x_i; \mu^{[q]}, \pi^{[q]})}{n(m-1)}$.

Mixture of BOS model

A BOS mixture

$$f(\boldsymbol{x}) = \sum_{k=1}^{K} p_k f_{BOS}(x_i, \mu_k, \pi_k)$$

can be estimated through an EM algorithm:

- E step:
  - compute usual t_{ik}
- M step:
  - run the previous EM algo. for estimating (μ, π)
    - E step: compute $p(c_i | x_i; \mu^{(q)}, \pi^{(q)})$
    - M step: update $\mu^{[q+1]}$ and $\pi^{[q+1]}$

Application to AERES bachelor degree evaluation, 2011

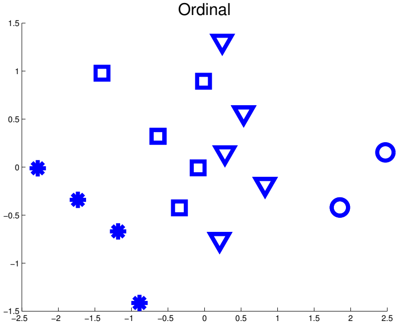
University	PT	EP	SS	EFS
Bordeaux 1	A	A	A	B
Bordeaux 2	A+	A	A+	A
Bordeaux 3	B	A	B	B
Bordeaux 4	B	A	A+	A
Pau	C	B	B	C
Toulouse 1	B	B	B	B
Toulouse 2	B	B	A	B
Toulouse 3	A	A	A+	A
Champollion	A	B	B	B
Lyon 1	A	A+	A	A
Lyon 2	B	A	B	B
Lyon 3	B	A+	B	B
St Etienne	A	B	A	B
Montpellier 1	B	A	A	B
Montpellier 2	A	A	A	B
Montpellier 3	B	B	A	B
Nîmes	C	B	C	C
Perpignan	B	B	B	B
Grenoble 1	B	B	A+	A
Grenoble 2	A	A	B	B
Grenoble 3	C	B	B	C
Savoie	A	A	A	B

The ordinalClust package

```r
library(ordinalClust)
res=bosclust(data,k=4)
```

Application to AERES bachelor degree evaluation, 2011

BIC selected 4 clusters:



- Cluster 1: $\hat{\mu}_1 =$ (A, A, A, B) *homogeneous high score*
- Cluster 2: $\hat{\mu}_2 =$ (B, A, A+, C) *contrasted score*
- Cluster 3: $\hat{\mu}_3 =$ (B, B, B, B) *homogeneous middle score*
- Cluster 4: $\hat{\mu}_4 =$ (C, B, B, C) *lower score*

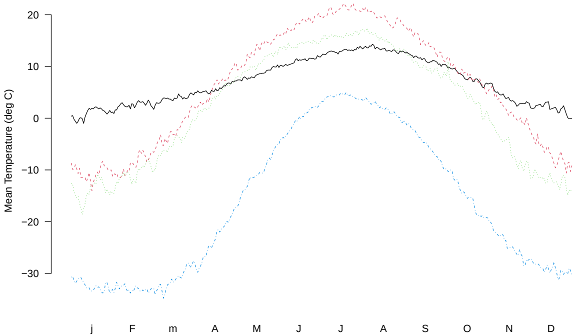
Functional features

Functional data

Functional data are curves, surfaces or anything else varying over a continuum:

$$(x_i(t))_{1 \le i \le n}, \qquad t \in [0, T]$$

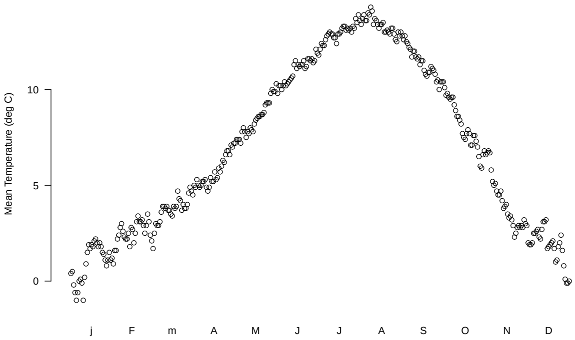
```
library(fda)
data("CanadianWeather")
stations <- c("Pr. Rupert", "Montreal", "Edmonton", "Resolute")
matplot(day.5, CanadianWeather$dailyAv[,stations , "Temperature.C"],
    type="l",axes=FALSE,xlab="",ylab="Mean Temperature (deg C)")
axis(2, las=1)
axis(1, monthMid, monthLetters, tick=FALSE)
```



Functional data

- functional data $x_i(t)$ are sample paths of a stochastic process $X \in L_2([0, T])$
- in practice $x_i(t)$ are not totaly observed, but only at some time points:

```
plot(day.5, CanadianWeather$dailyAv[,stations[1] , "Tempera
    type="p",lty=1,axes=FALSE,xlab="",ylab="Mean Temperatur
axis(2, las=1)
axis(1, monthMid, monthLetters, tick=FALSE)
```



Functional data

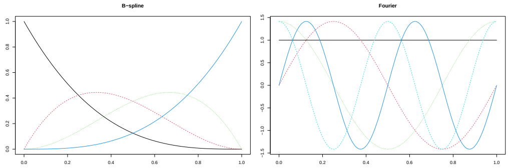
Functional Data Analysis (FDA) often start by reconstructing the functional form of data.

This can be done by assuming that the curves can be decomposed in a finite dimensional space:

$$x_i(t) = \sum_{r=1}^{R} c_{ir}\phi_r(t) = \boldsymbol{c}_i'\phi(t)$$

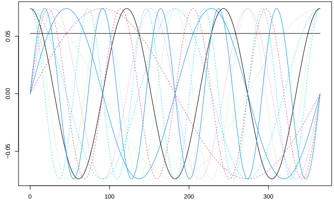
where the basis of functions $\phi(t)$ can be:

- Fourier basis when curve are periodic
- spline basis

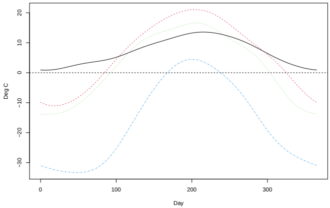


Functional data reconstruction

```
daybasis65 <- create.fourier.basis(c(0, 365), nbasis=10, period=365)
plot(daybasis65)
```




```
daytempfd=smooth.basis(day.5,CanadianWeather$dailyAv[,stations,"Temperature.C"]
                        daybasis65,fdnames=list("Day", "Station", "Deg C"))
plot(daytempfd$fd)
```




```
## [1] "done"
```

Functional data clustering

Several model-based clustering algorithms have been defined on the basis on distribution assumption for the basis expansion coefficients $(\boldsymbol{c}_i)_i$.

A. Schmutz, J. Jacques, C. Bouveyron, L. Chèze and P. Martin (2020). Clustering multivariate functional data in group-specific functional subspaces, Computational Statistics, 35, 1101-1131.

C. Bouveyron, E. Côme and J. Jacques (2015), The discriminative functional mixture model for the analysis of bike sharing systems, Annals of Applied Statistics, 9[4], 1726-1760.

C. Bouveyron and J. Jacques (2011), Model-based Clustering of Time Series in Group-specific Functional Subspaces, Advances in Data Analysis and Classification, 5[4], 281-300.

FunHDDC

In particular, funHDDC model is the extansion of HDDC model to functional data:

$$\boldsymbol{c}_i | z_{ik} = 1 \sim \mathcal{N}(U_k \mu_k, U_k \Sigma_k U_k^t + \Xi_k)$$

where

- U_k projects the $\boldsymbol{c}_i$ into a low dimensional subspace for cluster k
- (μ_k, Σ_k): (mean,variance) into the low-dimensional subspace,
- Ξ_k the noise covariance $m \times m$-matrix s.t.:

$$Q_k^t (U_k \Sigma_k U_k^t + \Xi_k) Q_k = \left(\begin{array}{cc} \boxed{\begin{matrix} s_{k1} & & 0 \\ & \ddots & \\ 0 & & s_{kd} \end{matrix}} & \mathbf{0} \\ \mathbf{0} & \boxed{\begin{matrix} b_k & & 0 \\ & \ddots & \\ 0 & & b_k \end{matrix}} \end{array} \right) \begin{array}{l} \left.\vphantom{\begin{matrix} s \\ \ddots \\ s \end{matrix}}\right\} d \\ \left.\vphantom{\begin{matrix} b \\ \ddots \\ b \end{matrix}}\right\} (m-d) \end{array}$$

with $s_{kj} > b_k$ for all $j = 1, ..., d$.

FunHDDC

Canadian temperature curves clustering with funHDDC

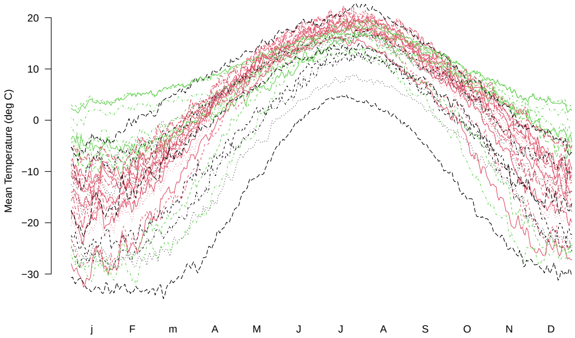
```
library(funHDDC)
daytempfd=smooth.basis(day.5,
    CanadianWeather$dailyAv[,,"Temperature.C"],
    daybasis65,fdnames=list("Day", "Station", "Deg C"))
res=funHDDC(daytempfd$fd,K=3,model="AkjBkQkDk",init="random")
```

```
## funHDDC:
##       model K threshold complexity       BIC
## 1 AKJBKQKDK 3       0.2         71 -3,037.64
##
## SELECTED: model  AKJBKQKDK  with  3  clusters.
## Selection Criterion: BIC.
```

FunHDDC

Canadian temperature curves clustering with funHDDC

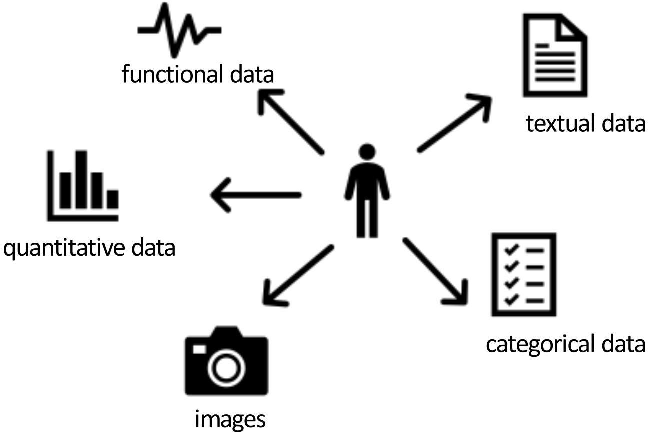
```
matplot(day.5, CanadianWeather$dailyAv[,, "Temperature.C"],
    type="l",axes=FALSE,xlab="",ylab="Mean Temperature (deg C)",
    col=res$class)
axis(2, las=1)
axis(1, monthMid, monthLetters, tick=FALSE)
```



Mixed type features

Mixed type data

Modern data are often of mixed type:



The challenge for model-based clustering is to defined a **pdf for mixed type data**.

A latent class model

Let assume that continuous and categorical features are available

- $X_1, \ldots, X_c$: categorical
- $X_{c+1}, \ldots, X_p$: continuous

bad idea

- to discretize continuous feature into categorical ones $\Rightarrow$ information loss

simple but good idea

- assume that continuous and categorical features are independent conditionally to $Z = k$

A latent class model

Under the assumption that continuous and categorical features are independent conditionally to $Z = k$

$$f_k(\boldsymbol{x}) = \underbrace{\prod_{j=1}^{c}\prod_{h=1}^{m_j}(\alpha_k^{jh})^{x_j^h}}_{f_k^{categ.}(x_1,\ldots,x_c)} \times \underbrace{\frac{1}{(2\pi)^{(p-c)/2}|\Sigma_k|^{1/2}}\exp\{-\frac{1}{2}(\tilde{\boldsymbol{x}}-\mu_k)^t\Sigma_k^{-1}(\tilde{\boldsymbol{x}}-\mu_k)\}}_{f_k^{contin.}(x_{c+1},\ldots,x_p)}$$

with $\tilde{\boldsymbol{x}} = (x_{c+1},\ldots,x_p)$

Exercice 3

Implement an EM algorithm for estimating the latent class model for mixed data (continuous + categorical).

Test it for the clustering of simulated categorical data set.

Mixed data clustering with R

The following library allows to perform (co-)clustering for mixed-type data

```
library(mixedClust)
```

Have a look to this package and its vignette.