Clustering of high-dimensional and functional data

(with some applications in Astrophysics)

Charles BOUVEYRON

Professor of Statistics & Chair in Artificial Intelligence
Director of the Institut 3IA Côte d'Azur

Maasai research team
Centre Inria d'Université Côte d'Azur

✉ charles.bouveyron@univ-cotedazur.fr
✉ charles.bouveyron@inria.fr
@cbouveyron

Preamble

“Essentially, all models are wrong but some are useful”

George E.P. Box

Introduction

Introduction

Statistical learning is nowadays an unavoidable field:

- it aims to model a phenomenon and predict its future behavior,
- classification is one of the most active topic in this field.

A big challenge is to learn from modern data which are:

- high-dimensional (p large),
- big or as stream (n large),
- evolutive (evolving phenomenon),
- heterogeneous (categorical, functional, networks, ...)

The understanding of the results is essential:

- in many applications, practitioners are very interested in visualizing the processed data,
- and to understand what are the relevant original variables for interpretation.

The Maasai research team (team.inria.fr/maasai/)

A research team in "core AI", created in 2020:

- 6 permanent researchers, 25 Ph.D. students and postdocs, and 4 engineers,
- located at the Centre Inria of Université Côte d'Azur, in Sophia-Antipolis,

The team focuses on the Models and Algorithms of Artificial Intelligence:

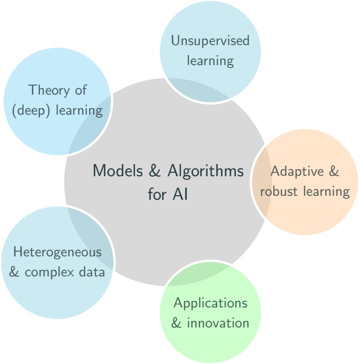

Unsupervised learning
Theory of (deep) learning
Models & Algorithms for AI
Adaptive & robust learning
Heterogeneous & complex data
Applications & innovation

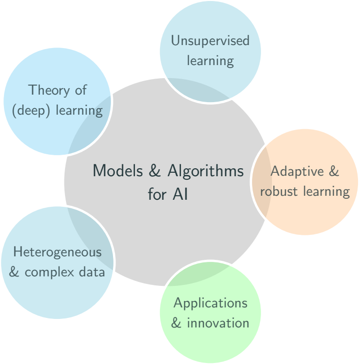
Figure 1: Scientific objectives of Maasai.

The Maasai research team (team.inria.fr/maasai/)

A research team in "core AI", created in 2020:

- 6 permanent researchers, 25 Ph.D. students and postdocs, and 4 engineers,
- located at the Centre Inria of Université Côte d'Azur, in Sophia-Antipolis,

The team focuses on the Models and Algorithms of Artificial Intelligence:



Unsupervised learning

Theory of (deep) learning

Models & Algorithms for AI

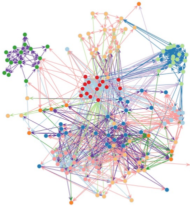
Adaptive & robust learning

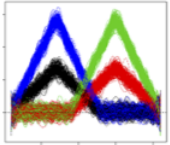
Heterogeneous & complex data

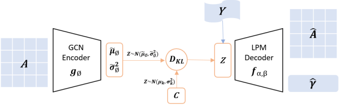
Applications & innovation

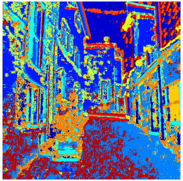
Figure 1: Scientific objectives of Maasai.

A summary of our topics:






Y
A
GCN Encoder g_ϕ
$\tilde{\mu}_\phi$
$\tilde{\sigma}^2_\phi$
$Z \sim N(\tilde{\mu}_\phi, \tilde{\sigma}^2_\phi)$
D_{KL}
$Z \sim N(\mu_k, \sigma^2_k)$
C
Z
LPM Decoder $f_{\alpha,\beta}$
$\hat{A}$
$\hat{Y}$



The challenges of high-dimensional data

Among the AI challenges, learning with HD data is still only partially solved:

- we have to deal with the "curse of dimensionality",
- high-dimensional spaces are quite "special"!
- the "ultra-high dimensional" case ($n < p$) is even more difficult.

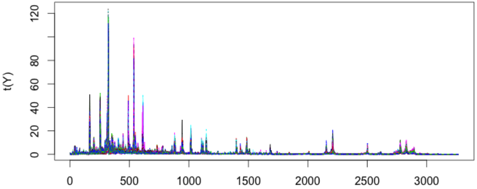
Functional data / time series is a special case:

- they can be seen as infinite dimensional data,
- such data are more and more present (smart sensors, high frequency recordings, ...)

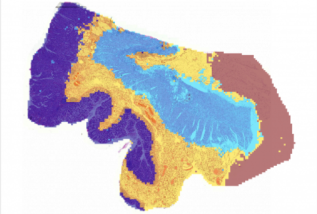
A motivating example: mass spectrometry

Mass spectrometry:

- it is a recent analytical technique that measures the mass-to-charge ratio of charged particles and which aims is to identify the elemental composition of a sample,
- It exist two types of mass spectrometry data:
  - multi-array data which aims to analyze serums or tissue fragments



  - MALDI images which are 2D or 3D MS images of tissues or organs



A motivating example: mass spectrometry

Classification is useful in this context:

- it is used in Medicine for disease diagnostic from blood samples:
  - a supervised classifier is learned from blood samples of healthy and sick patients,
  - the classifier is then used to classify new blood samples.
- a combination of supervised and unsupervised classification can be used to detect errors in the labels

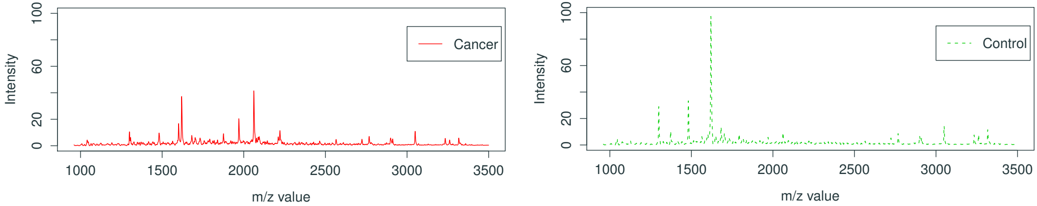


FIGURE: Control and cancer (colorectal) mass spectrometry spectra.

A motivating example: hyperspectral imaging

Hyperspectral imaging:

- it is an imaging technique which collects information from across the electromagnetic spectrum,
- as a consequence, the result is an image where each pixel is a high-dimensional spectrum,
- among the application fields, we can cite: agriculture, mineralogy, environment, security, astronomy.

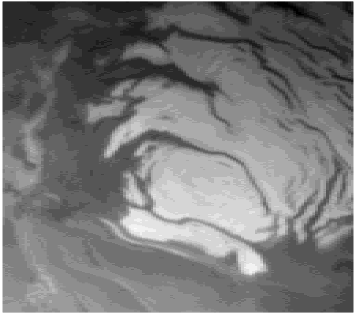


FIGURE: Image of the studied zone (south pole) of planet Mars.

A motivating example: hyperspectral imaging

The data from IPAG:

- a 300 × 128 hyperspectral image of the south pole of Mars,
- each "pixel" is described by a 256-dimensional spectrum.

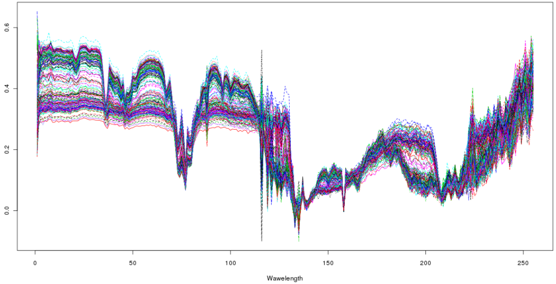


FIGURE: A few spectra of the studied zone.

Classification is useful in this context:

- for the segmentation of the studied zones -> ground nature classification,
- for selecting the discriminative spectral bands which allows the ground nature determination.

Problems and challenges in classification & clustering

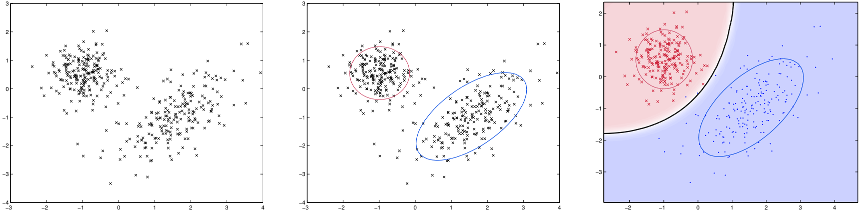
The classification problem

Classification is a two-headed problem:

- unsupervised classification which is also known as clustering,
- supervised classification which is also known as discriminant analysis.

The clustering problem consists in:

- organizing a set of n observations $y_1, ..., y_n \in \mathcal{Y}$ into K classes,
- *i.e.* associating the labels $z_1, ..., z_n \in \{1, ..., K\}$ to the data.

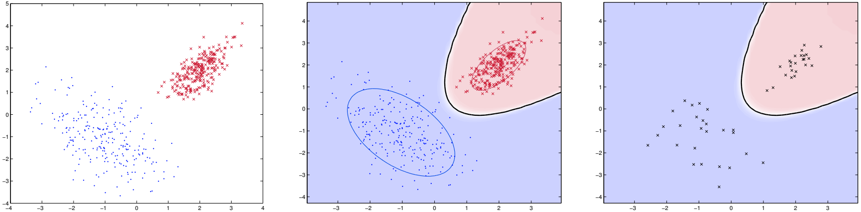


The classification problem

The discriminant analysis problem aims to:

- on the basis of a complete set $\{(y_1, z_1), ..., (y_1, z_1)\} \in \mathcal{Y} \times \{1, ..., K\}$, learn a classifier δ,
- which can predict the class z of a new observation y:

$$\begin{aligned} \delta : \mathcal{Y} &\rightarrow \{1, ..., K\}, \\ y &\rightarrow z. \end{aligned}$$



- The optimal rule δ^* is the one which assigns x to the class with the highest posterior probability (called the MAP rule):

$$\delta^*(x) = \mathrm{argmax}_{k=1,...,K} P(Z = k | X = x).$$

The mixture model

The mixture model:

- the observations $x_1, ..., x_n$ are assumed to be independent realizations of a random vector $X \in \mathcal{X}^p$ with a density:
$$f(x) = \sum_{k=1}^{K} \pi_k f(x, \theta_k),$$
    - K is the number of classes,
    - π_k are the mixture proportions,
    - $f(x, \theta_k)$ is a probability density with its parameters θ_k.

The Gaussian mixture model:

- among all mixture models, the Gaussian mixture model is certainly the most used in the classification context,
- in this case, $f(x, \theta_k)$ is the Gaussian density $\mathcal{N}(\mu_k, \Sigma_k)$ with $\theta_k = \{\mu_k, \Sigma_k\}$.

The mixture model

The MAP decision rule becomes in the mixture model framework:

$$\begin{aligned} \delta^*(x) &=_{k=1,\ldots,K} P(Z=k|X=x), \\ &=_{k=1,\ldots,K} P(Z=k)P(X=x|Z=k), \\ &=_{k=1,\ldots,K} H_k(x), \end{aligned}$$

where H_k is defined by $H_k(x) = -2\log(\pi_k f(x,\theta_k))$.

The building of the decision rule consists in:

1. estimate the parameters θ_k of the mixture model,
2. calculate the value of $H_k(x)$ for each new observation x.

Gaussian mixtures for classification

Gaussian model Full-GMM (QDA in discrimination):

$$H_k(x) = (x-\mu_k)^t\Sigma_k^{-1}(x-\mu_k) + \log(\det \Sigma_k) - 2\log(\pi_k) + C^{st}.$$

Gaussian model Com-GMM which assumes that $\forall k$, $\Sigma_k = \Sigma$ (LDA in discrimination):

$$H_k(x) = \mu_k^t\Sigma^{-1}\mu_k - 2\mu_k^t\Sigma^{-1}x - 2\log(\pi_k) + C^{st}.$$

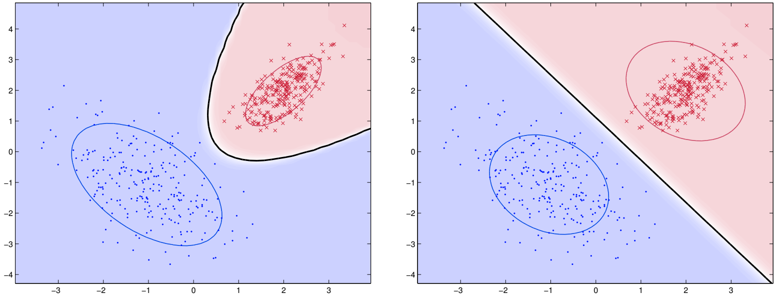


Fig. Decision boundaries for Full-GMM (left) and Com-GMM (right).

The curse of dimensionality

The curse of dimensionality:

- this term was first used by R. Bellman in the introduction of his book "Dynamic programming" in 1957:

  All [problems due to high dimension] *may be subsumed under the heading "the curse of dimensionality". Since this is a curse,* [...], *there is no need to feel discouraged about the possibility of obtaining significant results despite it.*

- he used this term to talk about the difficulties to find an optimum in a high-dimensional space using an exhaustive search,
- in order to promotate dynamic approaches in programming.

The curse of dimensionality

In the mixture model context:

- the building of the data partition mainly depends on:

$$H_k(x) = -2\log(\pi_k f(x,\theta_k)),$$

- model Full-GMM:

$$H_k(x) = (x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k) + \log(\det \Sigma_k) - 2\log(\pi_k) + \gamma.$$

Consequently:

- it is necessary to invert Σ_k which have a number of parameters proportional to p^2,
- if n is small compared to p^2, the estimates of Σ_k are ill-conditionned or singular and it will be difficult or impossible to invert Σ_k.

The curse of dimensionality

From the estimation point of view:

- let us consider the normalized trace $\tau(\Sigma) = \mathrm{tr}(\Sigma^{-1})/p$ of the inverse covariance matrix Σ^{-1} of a multivariate Gaussian distribution $\mathcal{N}(0, \Sigma)$,
- the estimation of τ from a sample of n observations $\{x_1, ..., x_n\}$ conduces to:

$$\tau(\hat{\Sigma}) = \tau(\hat{\Sigma}) = \frac{1}{p}\mathrm{tr}(\hat{\Sigma}^{-1}),$$

$$E[\tau(\hat{\Sigma})] = \left(1 - \frac{p}{n-1}\right)^{-1} \tau(\Sigma).$$

- consequently, if the ratio $p/n \to 0$ when $n \to +\infty$, then $E[\tau(\hat{\Sigma})] \to \tau(\Sigma)$,
- however, if the dimension p is comparable with n, then $E[\tau(\hat{\Sigma})] \to c\tau(\Sigma)$ when $n \to +\infty$, where $c = \lim_{n\to+\infty} p/n$.

The blessings of dimensionality

As Bellman thought:

- all is not bad in high-dimensional spaces (hopefully!)
- there are interesting things which happen in high-dimensional spaces.

First example: volume of the unit sphere is $V(p) = \frac{\pi^{p/2}}{\Gamma(p/2+1)}$,

The blessings of dimensionality

As Bellman thought:

- all is not bad in high-dimensional spaces (hopefully!)
- there are interesting things which happen in high-dimensional spaces.

First example: volume of the unit sphere is $V(p) = \frac{\pi^{p/2}}{\Gamma(p/2+1)}$,

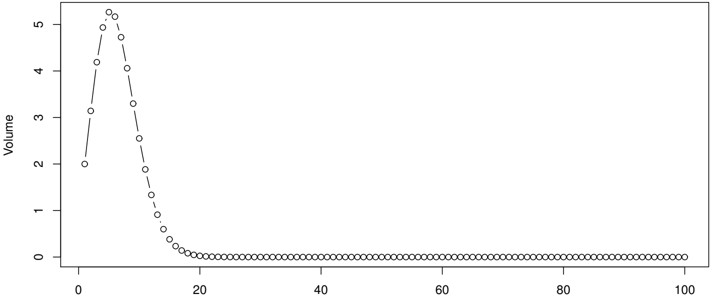


Fig. Volume of a sphere of radius 1 regarding to the dimension *p*.

The blessings of dimensionality

Second example: probability that a uniform variable on the unit sphere belongs to the shell between the spheres of radius 0.9 and 1 is

$$P(X \in S_{0.9}(p)) = 1 - 0.9^p \underset{p \to \infty}{\longrightarrow} 1$$

The blessings of dimensionality

Second example: probability that a uniform variable on the unit sphere belongs to the shell between the spheres of radius 0.9 and 1 is

$$P(X \in S_{0.9}(p)) = 1 - 0.9^p \underset{p \to \infty}{\longrightarrow} 1$$

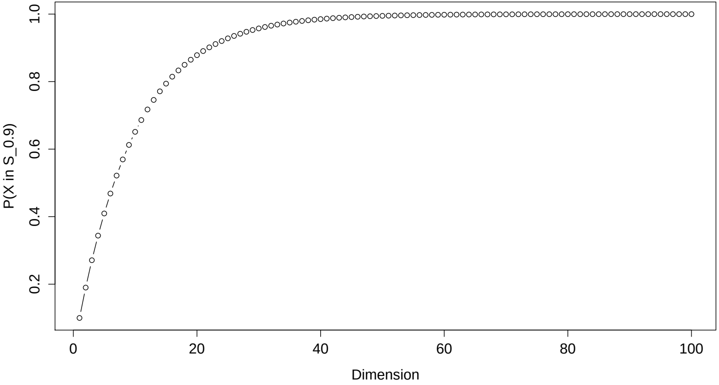


Fig. Probability that X belongs to the shell $S_{0.9}$ regarding to the dimension p.

The blessings of dimensionality

Third example:

- since high-dimensional spaces are almost empty,
- it should be easier to separate groups in high-dimensional space with an adapted classifier.

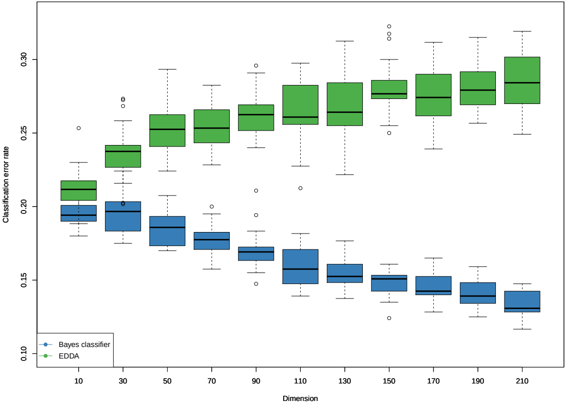


Fig. Correct classification rate of the optimal classifier versus the data dimension on simulated data.

Classical ways to avoid the curse of dimensionality

Dimension reduction:

- the problem comes from that p is too large,
- therefore, reduce the data dimension to $d \ll p$,
- such that the curse of dimensionality vanishes!

Regularization:

- the problem comes from that parameter estimates are unstable,
- therefore, regularize these estimates,
- such that the parameter are correctly estimated!

Parsimonious models:

- the problem comes from that the number of parameters to estimate is too large,
- therefore, make restrictive assumptions on the model,
- such that the number of parameters to estimate becomes more "decent"!

Recent approaches for clustering

In the past decade, several innovative approaches were proposed:

- subspace clustering:
  - several key works: Tipping & Bishop (Mixt. PPCA), McLachlan *et al.* (MFA), Bouveyron *et al.* (HDDC), McNicholas & Murphy (PGMM), Beak *et al.* (MCFA), ...
  - clustering in low-dimensional subspaces has shown a high efficiency but their result are difficult to interpret,

Recent approaches for clustering

In the past decade, several innovative approaches were proposed:

- subspace clustering:
  - several key works: Tipping & Bishop (Mixt. PPCA), McLachlan *et al.* (MFA), Bouveyron *et al.* (HDDC), McNicholas & Murphy (PGMM), Beak *et al.* (MCFA), ...
  - clustering in low-dimensional subspaces has shown a high efficiency but their result are difficult to interpret,
- variable selection for clustering:
  - Dean & Raftery and Maugis *et al.* proposed a Bayesian framework to iteratively select the relevant variables for model-based clustering,
  - these approaches successfully identify the relevant variables for the clustering but are time-consuming.

Recent approaches for clustering

In the past decade, several innovative approaches were proposed:

- subspace clustering:
  - several key works: Tipping & Bishop (Mixt. PPCA), McLachlan *et al.* (MFA), Bouveyron *et al.* (HDDC), McNicholas & Murphy (PGMM), Beak *et al.* (MCFA), ...
  - clustering in low-dimensional subspaces has shown a high efficiency but their result are difficult to interpret,
- variable selection for clustering:
  - Dean & Raftery and Maugis *et al.* proposed a Bayesian framework to iteratively select the relevant variables for model-based clustering,
  - these approaches successfully identify the relevant variables for the clustering but are time-consuming.
- sparsity:
  - Pan & Shen and Galimberti *et al.* proposed ℓ_1-penalized maximum likelihood approaches to select the relevant variables,
  - Witten & Tibshirani recently proposed a ℓ_1-penalized approach for k-means and hierarchical clustering,
  - these methods are also very efficient but time-consuming and difficult to parametrize.

Subspace clustering: HDDC

Objectives of subspace clustering

Our objectives:

- clustering efficiency: the methodology should match the performance standard of classical clustering techniques from both the clustering and the computing points of view,
- modeling: the methodology should provide a probabilistic modeling of each group and should be able to automatically choose the number of groups,
- visualization: the methodology should provide a comprehensive low-dimensional representation of the clustered data,

Objectives of subspace clustering

Our objectives:

- clustering efficiency: the methodology should match the performance standard of classical clustering techniques from both the clustering and the computing points of view,
- modeling: the methodology should provide a probabilistic modeling of each group and should be able to automatically choose the number of groups,
- visualization: the methodology should provide a comprehensive low-dimensional representation of the clustered data,

Our proposal:

- a subspace clustering method which models and clusters the data in low-dimensional subspaces.

The model $[a_{kj} b_k Q_k d_k]$

Bouveyron & Girard (2007) proposed to consider the Gaussian mixture model:

$$f(x) = \sum_{k=1}^{K} \pi_k f(x, \theta_k),$$

where $\theta_k = \{\mu_k, \Sigma_k\}$ for each $k = 1, ..., K$.

Based on the spectral decomposition of Σ_k, we can write:

$$\Sigma_k = Q_k \, \Delta_k \, Q_k^t,$$

where:

- Q_k is an orthogonal matrix containing the eigenvectors of Σ_k,
- Δ_k is diagonal matrix containing the eigenvalues of Σ_k.

The model $[a_{kj} b_k Q_k d_k]$

We assume that Δ_k has the following form:

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

where:

- $a_{kj} \geq b_k$, for $j = 1, \ldots, d_k$ and $k = 1, \ldots, K$,
- and $d_k < p$, for $k = 1, \ldots, K$.

The model $[a_{kj} b_k Q_k d_k]$

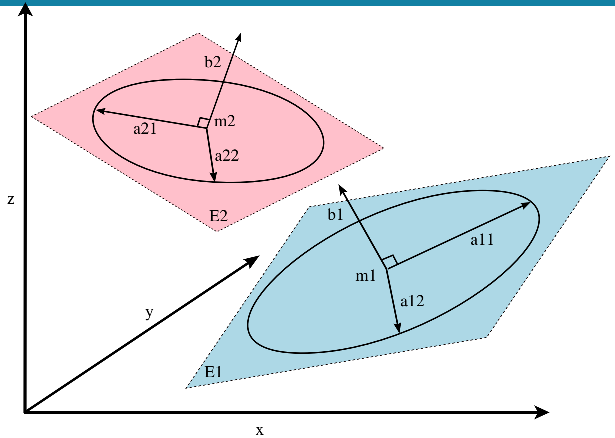


Fig. The subspace $\mathbb{E}_k$ and its supplementary $\mathbb{E}_k^\perp$.

We also define:

- the affine space $\mathbb{E}_k$ generated by eigenvectors associated to the eigenvalues a_{kj} and such that $\mu_k \in \mathbb{E}_k$,
- the affine space $\mathbb{E}_k^\perp$ such that $\mathbb{E}_k \oplus \mathbb{E}_k^\perp = \mathbb{R}^p$ and $\mu_k \in \mathbb{E}_k^\perp$,
- the projectors P_k and $P_k^\perp$ respectively on $\mathbb{E}_k$ and $\mathbb{E}_k^\perp$.

The model $[a_{kj} b_k Q_k d_k]$ and its submodels

We thus obtain a re-parameterization of the Gaussian model:

- which depends on a_{kj}, b_k, Q_k and d_k,
- the model complexity is controlled by the subspace dimensions.

We obtain increasingly regularized models:

- by fixing some parameters to be common within or between the classes,
- from the most complex model to the simplest model.

Our family of GMM contains 28 models and can be splitted into three branches:

- 14 models with free orientations,
- 12 models with common orientations,
- 2 models with common covariance matrices.

The model $[a_{kj} b_k Q_k d_k]$ and its submodels

Model	Nb of prms, $K = 4$ $d = 10, \ p = 100$	Classifier type
$[a_{kj} b_k Q_k d_k]$	4231	Quadratic
$[a_{kj} b_k Q d_k]$	1396	Quadratic
$[a_j bQd]$	1360	Linear
Full-GMM	20603	Quadratic
Com-GMM	5453	Linear

Table. Properties of the sub-models of $[a_{kj} b_k Q_k d_k]$

Construction of the classifiers

In the supervised context:

- the classifier has been named HDDA,
- the estimation of parameters is direct since we have complete data,
- parameters are estimated by maximum likelihood.

In the unsupervised context:

- the classifier has been named HDDC,
- the estimation of parameters is not direct since we do not have complete data,
- parameters are estimated through a EM algorithm which iteratively maximizes the likelihood.

HDDC: the E step

In the case of the model $[a_k b_k Q_k d_k]$:

$$H_k(x) = \frac{1}{a_k}\|\mu_k - P_k(x)\|^2 + \frac{1}{b_k}\|x - P_k(x)\|^2 + d_k \log(a_k) + (p - d_k)\log(b_k) - 2\log(\pi_k).$$

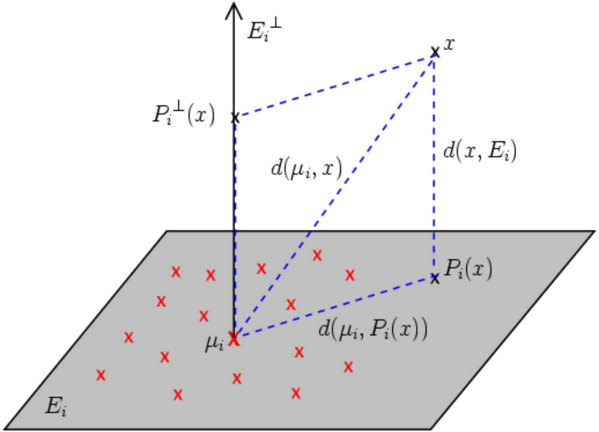


Fig. The subspaces $\mathbb{E}_k$ and $\mathbb{E}_k^{\perp}$ of the kth mixture composant.

HDDC: the M step

The ML estimators for the model $[a_{kj} b_k Q_k d_k]$ are closed forms:

- Subspace $\mathbb{E}_k$: the d_k first columns of Q_k are estimated by the eigenvectors associated to the d_k largest eigenvalues λ_{kj} of the empirical covariance matrix S_k of the kth class.

- Estimator of a_{kj}: the parameters a_{kj} are estimated by the d_k largest eigenvalues λ_{kj} of S_k.

- Estimator of b_k: the parameter of b_k is estimated by:

$$\hat{b}_k = \frac{1}{(p - d_k)} \left((S_k) - \sum_{j=1}^{d_k} \lambda_{kj} \right).$$

HDDC: hyper-parameter estimation

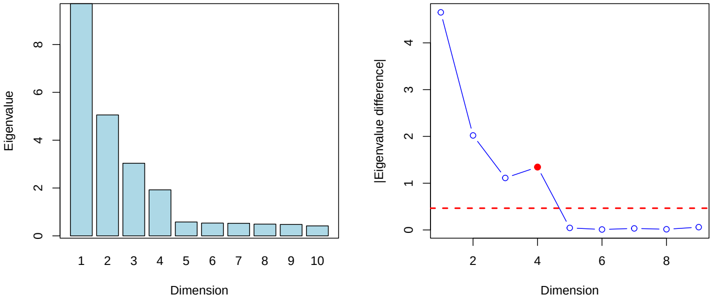


Fig. The scree-test of Cattell based on the eigenvalue scree.

Estimation of the intrinsic dimensions d_k :

- we use the *scree-test* of Cattell [Catt66],
- it allows to estimate the K parameters d_k in a common way.

Estimation of the nomber of groups K :

- in the supervised context, K is known,

Numerical considerations

- Numerical stability : the decision rule of HDDC does not depend on the eigenvectors associated with the smallest eigenvalues of W_k.

- Reduction of computing time : there is no need to compute the last eigenvectors of $S_k \rightarrow$ reduction of computing time with a designed procedure ($\times 60$ for $p = 1000$).

- Particular case $n < p$: from a numerical point of view, it is better to compute the eigenvectors of $\bar{X}_k \bar{X}_k^t$ instead of $S_k = \bar{X}_k^t \bar{X}_k$ ($\times 500$ for $n = 13$ and $p = 1000$).

HDDC: an EM-based algorithm

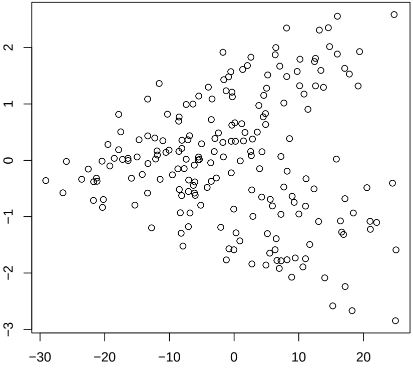


Fig. Projection of the «Crabs» data on the first principal axes.

«Crabs» data:

- 200 observations in a 5-dimensional space (5 morphological features),
- 4 classes: BM, BF, OM and OF.

HDDC: an EM-based algorithm

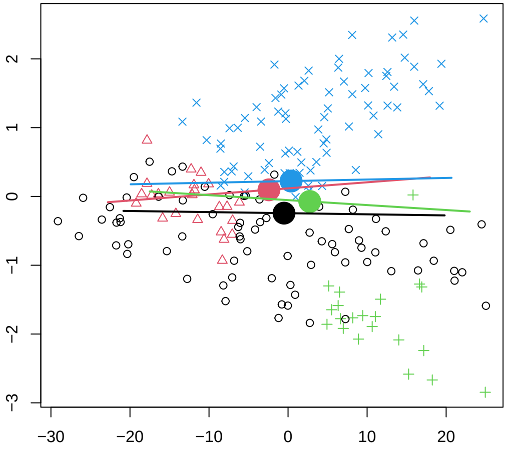


Fig. Step n° 1 of HDDC on the «Crabs» data.

HDDC: an EM-based algorithm

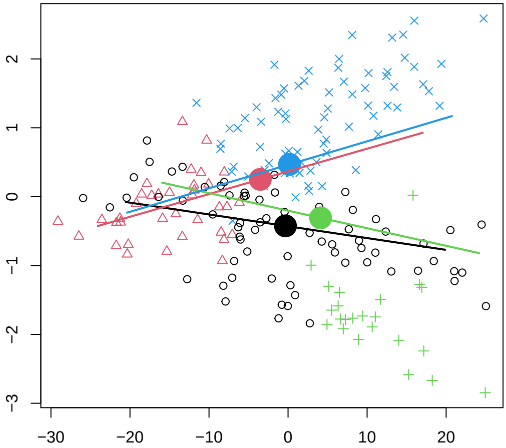


Fig. Step n° 2 of HDDC on the «Crabs» data.

HDDC: an EM-based algorithm

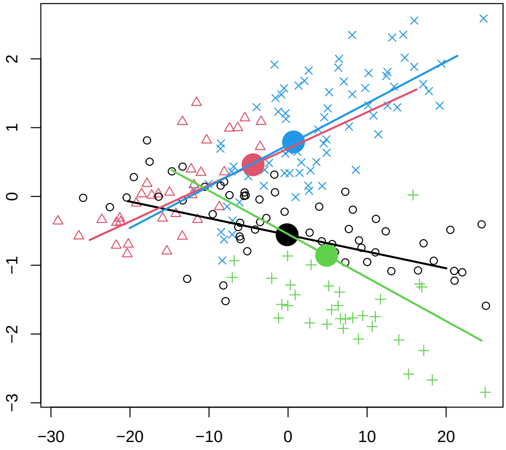


Fig. Step n° 3 of HDDC on the «Crabs» data.

HDDC: an EM-based algorithm

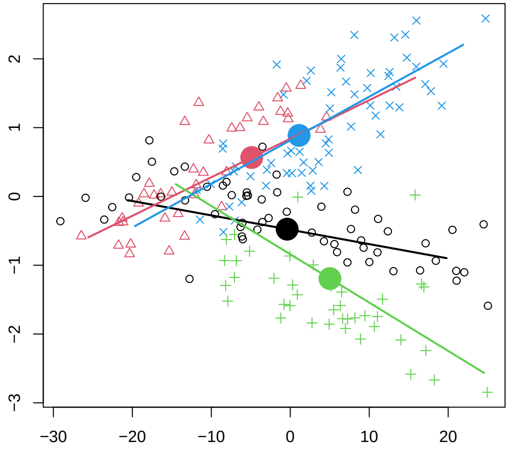


Fig. Step n° 4 of HDDC on the «Crabs» data.

HDDC: an EM-based algorithm

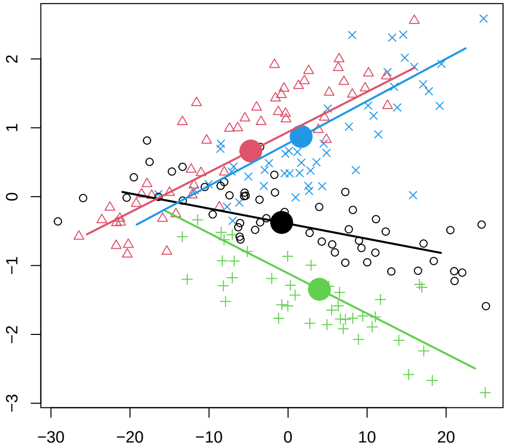


Fig. Step n° 5 of HDDC on the «Crabs» data.

HDDC: an EM-based algorithm

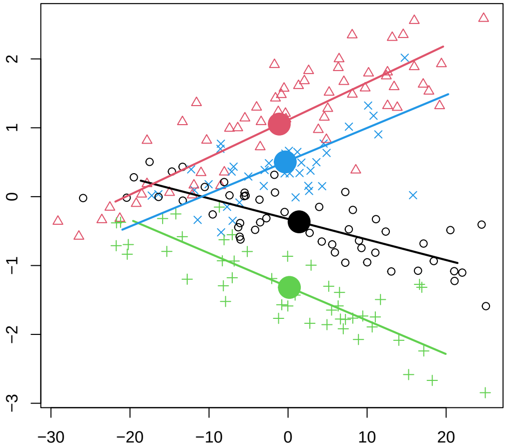


Fig. Step n° 10 of HDDC on the «Crabs» data.

HDDC: an EM-based algorithm

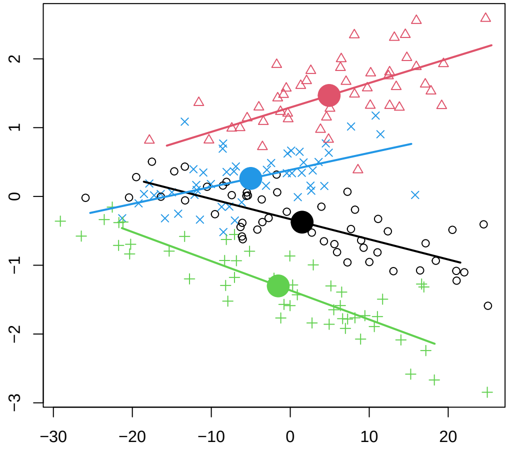


Fig. Step n°15 of HDDC on the «Crabs» data.

Discriminative clustering: Fisher-EM

Objectives of discriminative clustering

Our objectives:

- clustering efficiency: the methodology should match the performance standard of subspace clustering techniques from both the clustering and the computing points of view,
- modeling: the methodology should provide a probabilistic modeling of each group and should be able to automatically choose the number of groups,
- visualization: the methodology should provide a unique and comprehensive low-dimensional representation of the clustered data,
- interpretation: the methodology should allow to select the discriminative variables which may have specific meanings (biology, economics, ...)

Objectives of discriminative clustering

Our objectives:

- clustering efficiency: the methodology should match the performance standard of subspace clustering techniques from both the clustering and the computing points of view,
- modeling: the methodology should provide a probabilistic modeling of each group and should be able to automatically choose the number of groups,
- visualization: the methodology should provide a unique and comprehensive low-dimensional representation of the clustered data,
- interpretation: the methodology should allow to select the discriminative variables which may have specific meanings (biology, economics, ...)

Our proposal:

- a subspace clustering method which models and clusters the data in a common and discriminative low-dimensional subspace.

The DLM model... at a glance!

The observed random vector $Y \in \mathbb{R}^p$ is linked to a latent random vector $X \in \mathbb{E}$ (supposed to be the most discriminative) by:

$$Y = UX + \varepsilon,$$

where U is a $p \times d$ orthogonal matrix $(U^T U = I_d)$ and $d < p$.

Distribution assumptions, for $k = 1, ..., K$.:

$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \Psi),$$

$$X_{|Z=k} \sim \mathcal{N}(\mu_k, \Sigma_k),$$

The marginal distribution of Y is then:

$$f(y) = \sum_{k=1}^{K} \pi_k \phi(y; m_k, S_k),$$

where $m_k = U\mu_k$ and $S_k = U\Sigma_k U^T + \Psi_k$.

The DLM model... at a glance!

We finally assume that the noise covariance matrix Ψ_k is such that $\Delta_k = W^T S_k W$ has the following form:

$$\Delta_k = \left(\begin{array}{cc} \boxed{\quad \Sigma_k \quad} & \mathbf{0} \\ \mathbf{0} & \boxed{\begin{matrix} \beta_k & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \beta_k \end{matrix}} \end{array} \right) \begin{array}{l} \left.\vphantom{\begin{matrix} a \\ b \end{matrix}}\right\} \; d \leq K-1 \\ \left.\vphantom{\begin{matrix} a \\ b \\ c \\ d \end{matrix}}\right\} \; (p-d) \end{array}$$

where $W = [U, V]$.

This model is referred to by $\mathrm{DLM}_{[\Sigma_k \beta_k]}$ and 11 submodels can be obtained by constraining parameters within or between groups.

The DLM model

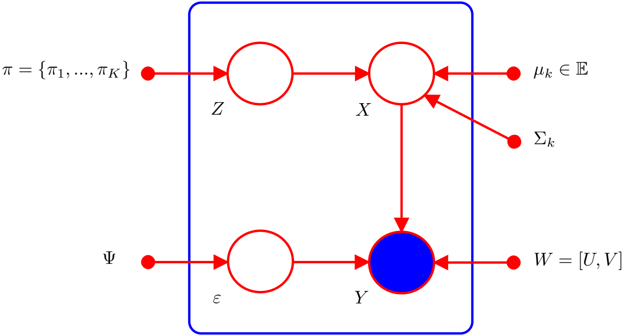


Figure 2: Graphical summary of the $\mathrm{DLM}_{[\Sigma_k\beta]}$ model

The Fisher-EM algorithm

The inference of mixture models:

- is usually done with the EM algorithm since likelihood maximization in intractable,
- however, we can not make use of the EM algorithm here since the subspace has to be discriminant.

We therefore proposed the Fisher-EM algorithm for inferring the DLM models:

- **a E step** which, roughly speaking, determines the current data partition through the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the current partition of the data,
- **a M step** which updates the mixture parameters conditionally to U and t_{ik}.

Looking back in the past: Fisher's criterion

We based our F step on the idea of Fisher's discriminant analysis (1936):

- knowing a partition of the data, Fisher's objectives were to find a low-dimensional subspace such that:
  - the groups are well separated → large between-class variance S_B
  - the groups are homogeneous → small within-class variance S_W
- since $S = S_W + S_B$, the usual Fisher criterion writes as follows:

$$\max_U \quad \mathrm{tr}\left((U^T S U)^{-1} U^T S_B U\right),$$

- the solution of this optimization problem are the $d = K - 1$ eigenvectors of the matrix $S^{-1} S_B$

Looking back in the past: Fisher's criterion

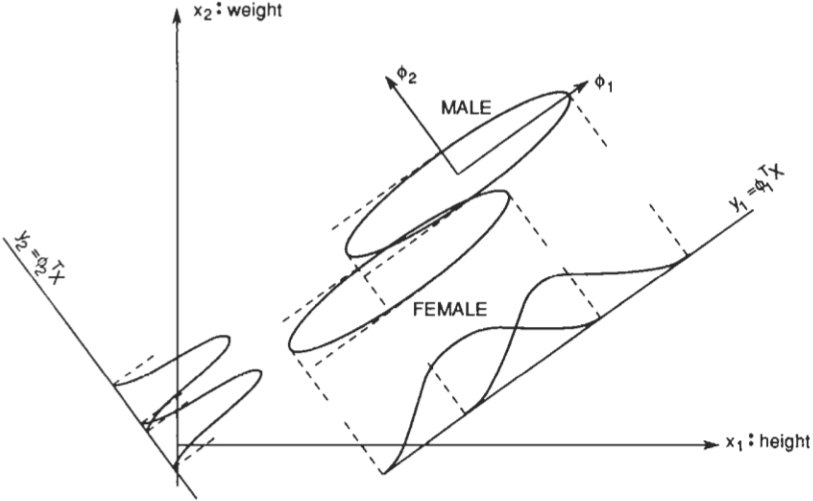


Figure 3: Discriminative axis vs. principal axis (Fukunaga, 1990)

The F step of the Fisher-EM algorithm

The F step of Fisher-EM:

- determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving the unsupervised counterpart of Fisher's criterion:

$$\left\{ \begin{array}{ll} \max\limits_{U} & tr\left((U^T SU)^{-1} U^T S_B^{(q)} U\right), \\ \text{wrt} & u_j^T u_l = 0, \quad \forall j \neq l \in \{1, \ldots, d\}, \end{array} \right. \tag{1}$$

  where:

  - $S_B^{(q)} = \frac{1}{n} \sum_{k=1}^{K} n_k^{(q)} (\hat{m}_k^{(q)} - \bar{y})^T (\hat{m}_k^{(q)} - \bar{y})$,
  - $n_k^{(q)} = \sum_{i=1}^{n} t_{ik}^{(q)}$, $\hat{m}_k^{(q)} = \frac{1}{n} \sum_{i=1}^{n} t_{ik}^{(q)} y_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$.

- we proposed a Gramm-Schmidt procedure to solve this constrained optimization problem.

The Fisher-EM algorithm... at work!

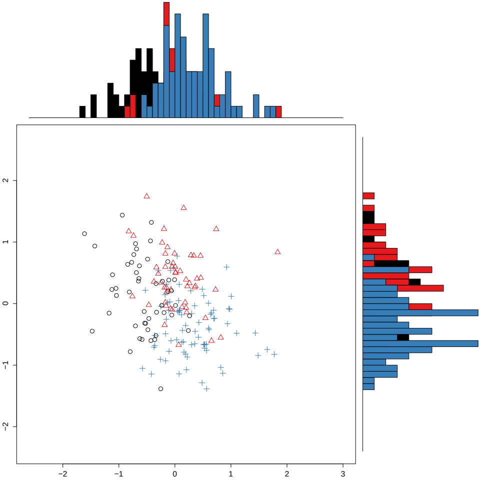


Figure 4: Step 0 of the Fisher-EM algorithm on the Wine data.

The Fisher-EM algorithm... at work!

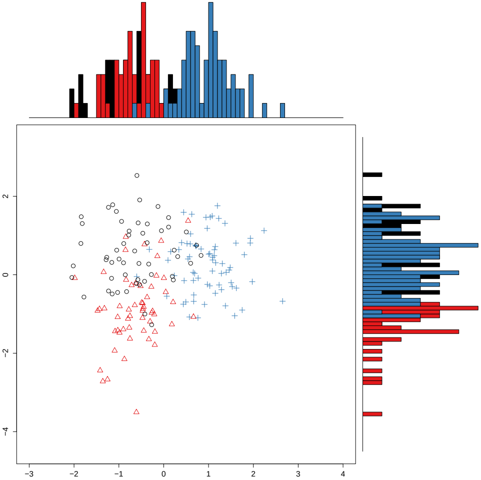


Figure 5: Step 1 of the Fisher-EM algorithm on the Wine data.

The Fisher-EM algorithm... at work!

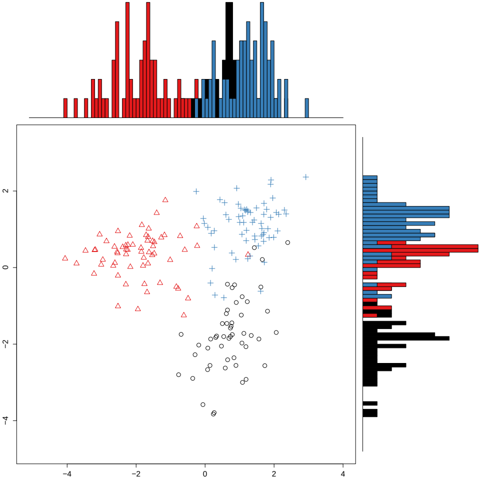


Figure 6: Step 2 of the Fisher-EM algorithm on the Wine data.

The Fisher-EM algorithm... at work!

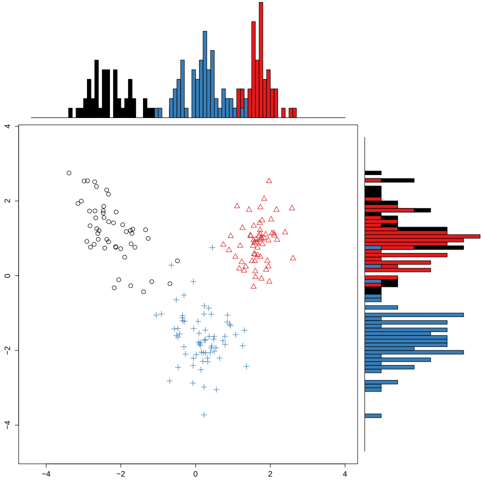


Figure 7: Step 3 of the Fisher-EM algorithm on the Wine data.

The Fisher-EM algorithm... at work!

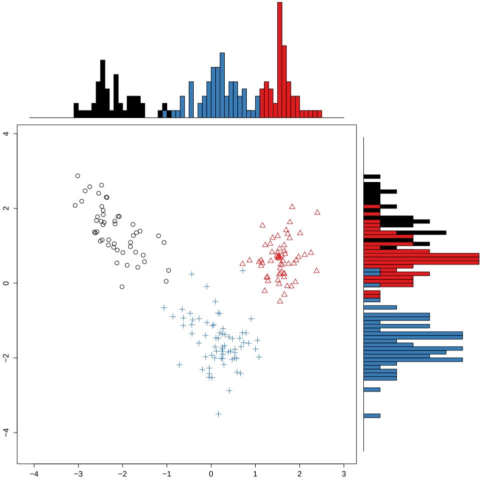


Figure 8: Step 5 of the Fisher-EM algorithm on the Wine data.

The Fisher-EM algorithm... at work!

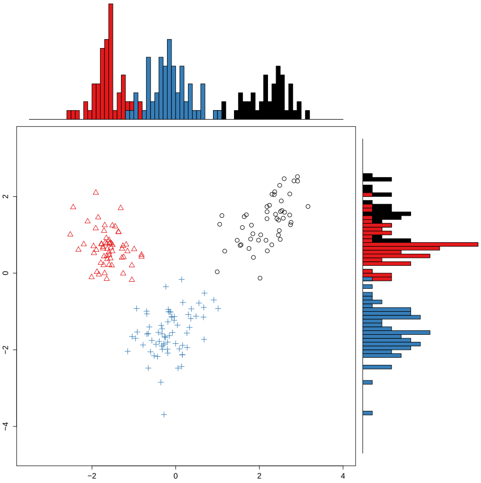


Figure 9: Step 10 of the Fisher-EM algorithm on the Wine data.

The Fisher-EM algorithm... at work!

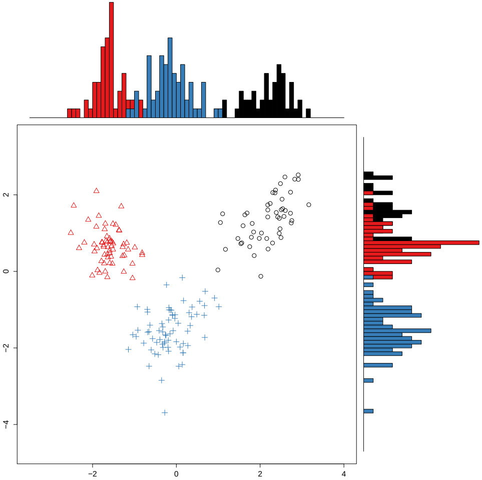


Figure 10: Step 20 of the Fisher-EM algorithm on the Wine data.

Experimental results: benchmark

Method	iris	wine	chiro	zoo	glass	satimage	usps358
$\text{DLM}_{[\Sigma_k\beta_k]}$	86.8±7.3†	97.8±0.0*	91.2±6.1	80.1±5.7	48.5±2.6	69.6±0.0*	81.1±5.4*†
$\text{DLM}_{[\Sigma_k\beta]}$	92.6±11	89.3±0.0	**98.2**±3.4	-	47.9±2.7	64.5±0.0	77.4±9.1
$\text{DLM}_{[\Sigma\beta_k]}$	80.5±3.4	93.8±1.1	94.7±4.2	72.6±5.3	49.4±2.9	65.7±1.3	73.7±7.4
$\text{DLM}_{[\Sigma\beta]}$	79.1±2.9	89.8±0.8	85.2±3.2	79.6±5.6	48.6±3.6	65.5±1.6	76.4±9.9
$\text{DLM}_{[\alpha_{kj}\beta_k]}$	87.8±0.5*	97.2±0.0†	85.0±1.4	71.8±6.6†	49.6±2.6†	**70.1**±0.0	**82.3**±4.7
$\text{DLM}_{[\alpha_{kj}\beta]}$	**97.8**±0.1	95.2±1.6	98.1±5.2	71.4±8.0	**51.1**±2.1*	61.7±0.2	73.2±9.5
$\text{DLM}_{[\alpha_k\beta_k]}$	92.8±2.1	**98.9**±0.0	85.5±14*†	71.8±6.9*	48.5±2.2	68.8±0.0	70.9±13.6
$\text{DLM}_{[\alpha_k\beta]}$	95.8±7.3	97.1±0.9	97.8±5.0	71.0±6.4	49.5±2.4	68.8±0.0	68.3±11.2
$\text{DLM}_{[\alpha_j\beta_k]}$	81.6±4.5	91.6±0.5	93.8±4.1	68.5±6.7	49.3±1.8	62.9±0.0†	76.1±11.0
$\text{DLM}_{[\alpha_j\beta]}$	73.6±6.7	89.8±0.9	89.7±4.1	79.1±4.9	47.4±1.2	67.6±2.8	77.4±10.7
$\text{DLM}_{[\alpha\beta_k]}$	80.1±6.9	91.4±3.2	89.3±1.9	70.1±6.5	48.9±1.3	68.7±1.9	80.5±6.0
$\text{DLM}_{[\alpha\beta]}$	66.8±0.0	89.5±1.0	89.2±5.7	**80.2**±5.3	47.0±1.7	62.1±0.0	69.9±14.2
Full-GMM	79.0±5.7	60.9±7.7	44.8±4.1	-	38.3±2.1	35.9±3.1	-
Com-GMM	57.6±18.3	61.0±14.9	51.9±10.9	59.9±10.3	38.3±3.1	26.1±1.5	38.2±1.1
Mixt-PPCA	89.1±4.2	63.1±7.9	56.3±4.5	50.9±6.5	37.0±2.3	40.6±4.7	53.1±9.6
Diag-GMM	93.5±1.3	94.6±2.8	92.1±4.2	70.9±12.3	39.1±2.4	60.8±5.2	45.9±9.1
Sphe-GMM	89.4±0.4	96.6±0.0	85.9±9.9	69.4±5.4	37.0±2.1	60.2±7.5	78.7±11.2
PCA-EM	66.9±9.9	64.4±5.7	66.1±4.0	61.9±6.2	39.0±1.7	56.2±4.2	67.6±11.2
k-means	88.7±4.0	95.9±4.0	92.9±6.0	68.0±7.4	41.3±2.8	66.6±4.1	74.9±13.9
MCFA ($q=3$)	80.6±12.6	92.9±8.2	75.4±7.8	-	47.7±6.9	67.9±8.8	54.2±8.7
PGMM	96.7±0.0	97.1±0.0	97.9±0.0	65.3±0.0	41.6±0.0	58.7±0.0	55.5±0.0
Mclust	96.7	97.1	97.9	65.3	41.6	58.7	55.5
Model name	*(VEV)*	*(VVI)*	*(EEE)*	*(EII)*	*(VEV)*	*(VVV)*	*(EEE)*

Application to SDSS galaxy spectra

The Sloan Digital Sky Survey (SDSS) dataset:

- spectra of 702248 galaxies and quasars (with redshift smaller than 0.25),
- 3 850 dimensions (points in the wavelength range, $\lambda = 3800$ to 9250Å),
- the spacing is uniform in resolution ($\delta\lambda/\lambda = 1/4342$),
- taking the redshift into account, the range common to all the spectra goes from 3806 to 7371Å with 2874 wavelengths.

Our objectives:

- experiment modern clustering tools for HD data in astrophysics,
- have a new look at existing classifications of galaxies / quasars.

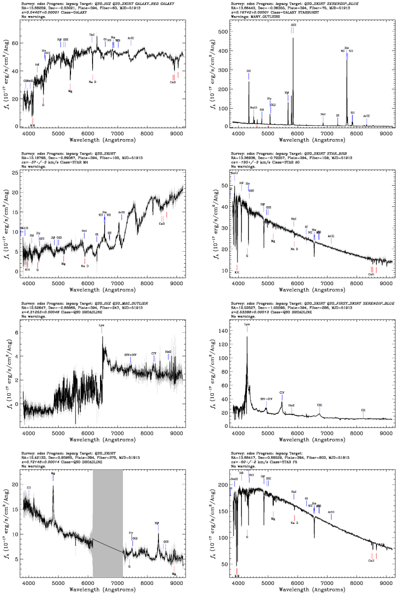
Application to SDSS galaxy spectra

The Sloan Digital Sky Survey (SDSS) dataset:

- spectra of 702248 galaxies and quasars (with redshift smaller than 0.25),
- 3 850 dimensions (points in the wavelength range, $\lambda = 3800$ to 9250Å),
- the spacing is uniform in resolution ($\delta\lambda/\lambda = 1/4342$),
- taking the redshift into account, the range common to all the spectra goes from 3806 to 7371Å with 2874 wavelengths.

Our objectives:

- experiment modern clustering tools for HD data in astrophysics,
- have a new look at existing classifications of galaxies / quasars.



Application to SDSS galaxy spectra

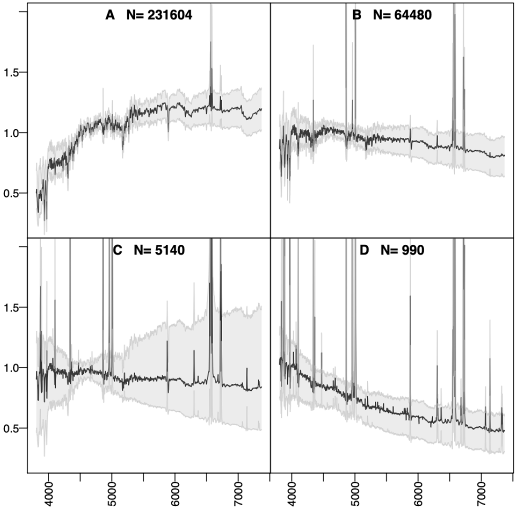


Figure 11: The 4-group clustering of the SDSS data.

Application to SDSS galaxy spectra

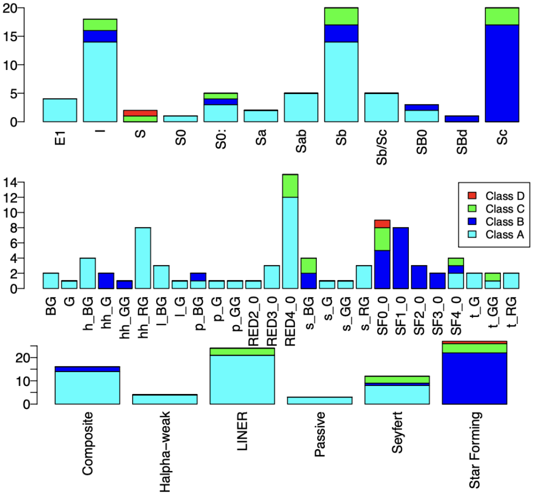


Figure 12: Comparison of the obtained segmentations with the ones of Kennicutt (1992), Dobos et al. (2012) and Wang et al. (2018).

Application to SDSS galaxy spectra

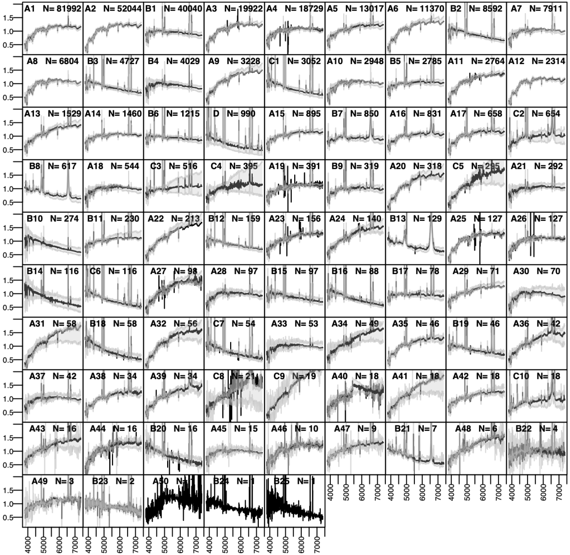


Figure 13: The 86 final groups of the sub-clustering of the 4 initial groups.

Discriminative variable selection by ℓ_1 penalization

Discriminative variable selection

Clustering is a data analysis tool and result interpretation is important. Unfortunately, the loading matrix U is usually difficult to interpret:

variable	axis 1	axis 2
sepal length	-0.203	-0.062
sepal width	-0.324	-0.697
petal length	0.519	0.404
petal width	0.763	-0.588

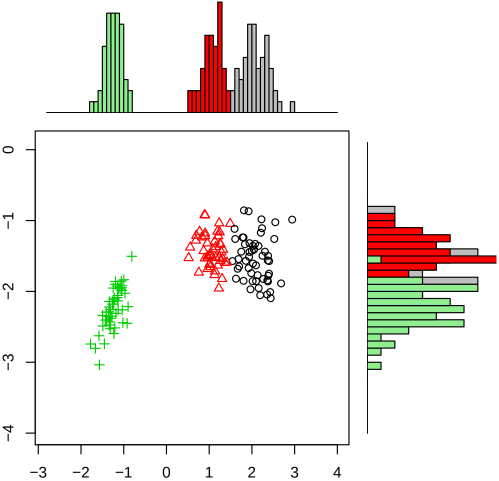
Discriminative variable selection

Clustering is a data analysis tool and result interpretation is important. Unfortunately, the loading matrix U is usually difficult to interpret:

variable	axis 1	axis 2
sepal length	-0.203	-0.062
sepal width	-0.324	-0.697
petal length	0.519	0.404
petal width	0.763	-0.588

And we would prefer:

variable	axis 1	axis 2
sepal length	0	0
sepal width	0	-1
petal length	0	0
petal width	1	0



Three ways to introduce sparsity

We chose to introduce sparsity within the F step:

- we want to identify the original variables which best discriminate the groups,
- which amounts to estimate the orientation matrix U with, as much as possible, only 0 or ± 1,
- a popular way to do that is to use a ℓ_1 penalty (lasso).

We identified three different ways to introduce sparsity:

1. SparseFEM$_1$: classical F step + sparsity step,
2. SparseFEM$_2$: F step as a ℓ_1-penalized regression problem,
3. SparseFEM$_3$: sparse SVD on the matrix $S^{-1}S_B^{(q)}$.

SparseFEM$_2$: ℓ_1-penalized regression problem

Defining the matrices $H_W^{(q)}$ and $H_B^{(q)}$ such that $H_W^{(q)} H_W^{(q)t} = S_W^{(q)}$ and $H_B^{(q)} H_B^{(q)t} = S_B^{(q)}$, we obtained:

Proposition
The best sparse approximation at the level λ of the solution of (1) is the solution $\hat{B}$ of the following penalized regression problem:

$$\min_{A,B} \sum_{k=1}^{K} \left\| R_W^{(q)-t} H_{B,k}^{(q)} - AB^t H_{B,k}^{(q)} \right\|_F^2 + \rho \sum_{j=1}^{d} \beta_j^t S_W^{(q)} \beta_j + \lambda \sum_{j=1}^{d} \|\beta_j\|_1 ,$$

such that $A^t A = \mathbf{I}_d$ and where $R_W^{(q)} \in \mathbb{R}^{p \times p}$ is such that $S_W^{(q)} = R_W^{(q)t} R_W^{(q)}$, $A = [\alpha_1, ..., \alpha_d]$, $B = [\beta_1, ..., \beta_d]$, $H_{B,k}^{(q)}$ is the kth column of $H_B^{(q)}$ and $\rho > 0$ is a ridge-type regularization parameter.

Remark : we proposed an iterative procedure based on the LARS algorithm to solve this problem.

Selection of the sparsity parameter

The selection of the hyper-parameter λ:

- this problem has received very few attention in the unsupervised context,
- a natural way in the model-based clustering context is to use the BIC criterion,
- but, the degree of freedom of the model has to be updated in order to take into account the sparsity!

Selection of the sparsity parameter

The selection of the hyper-parameter λ:

- this problem has received very few attention in the unsupervised context,
- a natural way in the model-based clustering context is to use the BIC criterion,
- but, the degree of freedom of the model has to be updated in order to take into account the sparsity!

[Zou07, Kachour11] have shown:

- that the number of non zero coefficients is a consistent estimator of the degree of freedom of the model,
- we finally get for the model of SparseFEM:

$$BIC_{pen}(\mathcal{M}) = -2\log(\mathcal{L}(\hat{\theta})) - \gamma_e \log(n),$$

where $\gamma_e = (K-1) + Kd + (d[p-(d+1)/2] - \mathbf{d_e}) + Kd(d+1)/2 + K$.

Comparison with variable selection

	iris	wine	chiro	zoo	glass	satimage	usps358
	$(p=4,K=3)$	$(p=13,K=3)$	$(p=17,K=3)$	$(p=16,K=7)$	$(p=9,K=7)$	$(p=36,K=6)$	$(p=256,K=3)$
Approaches	$(n=150)$	$(n=178)$	$(n=178)$	$(n=101)$	$(n=214)$	$(n=4435)$	$(n=1726)$
Fisher-EM	88.3 ± 1.0	97.8 ± 0.0	89.5 ± 13.0	71.8 ± 6.9	51.1 ± 2.1	69.6 ± 0.0	82.3 ± 4.7
sparseFEM-1	96.5±0.3	97.8±0.2	84.2±11	71.4±8.5	50.2±1.9	69.6±0.1	84.7±3.2
	(2.0±0.0*)*	*(2.0*±0.0*)*	*(2.3*±0.5*)*	*(13*±2.5*)*	*(6.0*±1.0*)*	*(36*±0.0*)*	*(5.5*±0.7*)*
sparseFEM-2	89.9±0.4	98.3±0.0	84.8±12	70.1±12.2	48.4±3.0	67.5±1.6	82.8±9.1
	(4.0±0.0*)*	*(4.0*±0.0*)*	*(2.0*±0.6*)*	*(14*±3.6*)*	*(6.6*±0.7*)*	*(36*±.0.0*)*	*(15.5*±16*)*
sparseFEM-3	96.5±0.3	97.8±0.0	82.9±12	72.0±4.3	48.2±2.7	71.8±2.3	79.1±7.4
	(2.0±0.3*)*	*(2.0*±0.0*)*	*(2.0*±0.0*)*	*(10*±2.8*)*	*(7.0*±0.0*)*	*(36*±0.0*)*	*(6.0*±1.3*)*
sparse-kmeans	90.7	94.9	95.3	79.2	52.3	71.4	74.7
	(4.0)	*(13.0)*	*(17.0)*	*(16.0)*	*(6.0)*	*(36.0)*	*(213)*
Clustvarsel	96.0	92.7	71.1	75.2	48.6	58.7	48.3
	(3.0)	*(5.0)*	*(6.0)*	*(3.0)*	*(3.0)*	*(19.0)*	*(6.0)*
Selvarclust	96.0	94.4	92.6	92.1	43.0	56.4	36.7
	(3.0)	*(5.0)*	*(8.0)*	*(5.0)*	*(6.0)*	*(22.0)*	*(5.0)*

Figure 14: Clustering accuracies and their standard deviations on 7 UCI datasets (averaged on 20 trials, models and λ selected by BIC).

A comparative example: the USPS358 dataset

We first considered the USPS358 dataset:

- which contains 1756 handwritten digits (3, 5 and 8),
- and each 16 × 16 grayscale image has been transformed as a 256-dimensional vector.

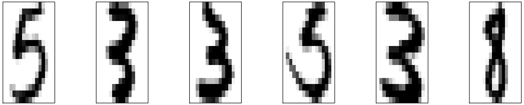


Figure 15: Sample from the USPS358 dataset.

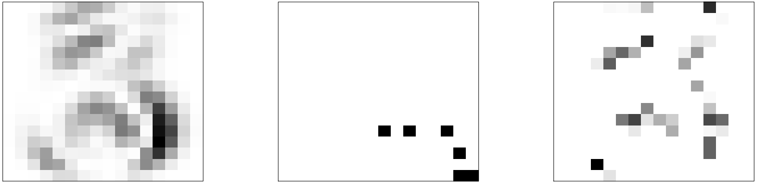
	Clustering	Non-zero	Elapsed
Approaches:	accuracy	variables	time in sec.
Fisher-EM	82.3 ± 4.7	256 ± 0.00	218.8 ± 1.5
SparseFEM$_1$	82.69 ± 6.82	5.6 ± 0.97	967.8 ± 1.1
SparseFEM$_2$	81.42 ± 6.77	16.0 ± 0.00	325.3 ± 1.0
SparseFEM$_3$	80.62 ± 8.06	10.1 ± 4.63	58.3 ± 2.6

Table 2: Clustering accuracies and computing times for the 3 versions of the sparseFEM algorithm on the 256-dimensional dataset USPS358 ($\lambda = 0.1$).

A comparative example: the USPS358 dataset

Method	Computing time	Method	Computing time
SparseFEM$_1$	967.8±1.1 sec.	Sparse k-means	1 783 sec.
SparseFEM$_2$	325.3±1.0 sec.	ClustVarSel	4 602 sec.
SparseFEM$_3$	58.3±2.6 sec.		

Table 3: Computing time on the USPS358 dataset.



(a) Sparse k-means

(b) ClustVarSel

(c) SparseFEM2

Figure 16: Variable selection obtained with the 3 sparse algorithms on the USPS358 dataset.

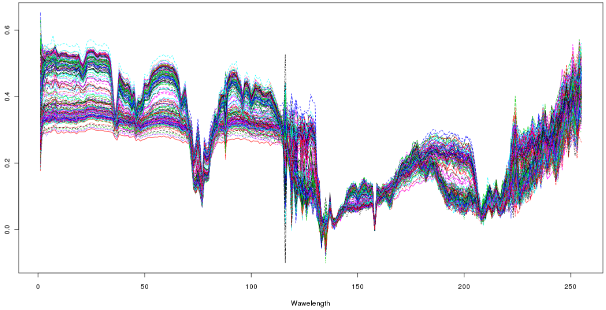
Application to hyper-spectral image analysis

The Mars Express data set:

- hyper-spectral images of the planet Mars taken in 2004,
- we considered the analysis of an image of the south pole of Mars,
- the data are 300×128 pixels described by 256 spectral variables.

We used sparseFEM to analyze this data set:

- the sparsity level λ was fixed to 0.1 to ensure to select a few discriminative variables,
- the whole process took 18 hours on a 2.6 Ghz computer.



Application to hyper-spectral image analysis

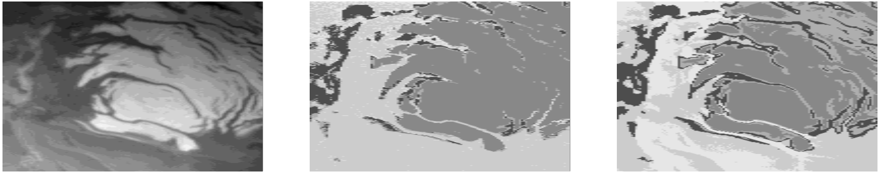


Figure 17: Segmentation results: original image (left), expert segmentation (center) and sparseFEM segmentation (right).

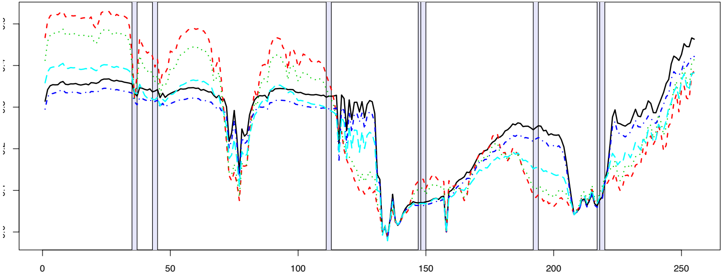


Figure 18: Selection of the discriminative spectral variables by sparseFEM.

The DFM model for the clustering of functional data

Transformation of the observed curves

Let us first assume that the observed curves $\{x_1, ..., x_n\}$ are independent realizations of a L_2-continuous stochastic process $X = \{X(t)\}_{t\in[0,T]}$.

Let us also assume that the stochastic process X admits the following basis expansion:

$$X(t) = \sum_{j=1}^{p} \gamma_j(X)\psi_j(t), \tag{2}$$

where:

- $\{\psi_1, \ldots, \psi_p\}$ is a basis of functions,
- $\Gamma = (\gamma_1(X), ..., \gamma_p(X))$ is a random vector in $\mathbb{R}^p$.

The DFM model

Let $F[0, T]$ be a latent subspace of $L_2[0, T]$ assumed to be:

- the most discriminative subspace for the K groups,
- spanned by a basis of d basis functions $\{\varphi_j\}_{j=1,...,d}$ with $d < K < p$.

The basis $\{\varphi_j\}_{j=1,...,d}$ is obtained from $\{\psi_j\}_{j=1,...,p}$ through a linear transformation

$$\varphi_j = \sum_{\ell=1}^{p} u_{j\ell}\psi_\ell,$$

such that the $p \times d$ matrix $U = (u_{j\ell})$ is orthogonal.

Let $\Lambda = \{\lambda_1, ..., \lambda_n\}$ be the basis expansion coefficients of the stochastic process $X(t)$ in the basis $\{\varphi_j\}_{j=1,...,d}$.

The DFM model

The previous modeling implies that Γ and Λ are linked by:

$$\Gamma = U\Lambda + \varepsilon, \tag{3}$$

where $\varepsilon \in \mathbb{R}^p$ is an independent and random noise term.

Distribution assumptions, for $k = 1, ..., K$.:

$$\Lambda_{|Z=k} \sim \mathcal{N}(\mu_k, \Sigma_k),$$

$$\varepsilon \sim \mathcal{N}(\mathbf{0}, \Xi),$$

The marginal distribution of Γ is then:

$$f(\gamma) = \sum_{k=1}^{K} \pi_k \phi(y; m_k, S_k),$$

where $m_k = U\mu_k$ and $S_k = U\Sigma_k U^T + \Xi$.

The DFM model

We finally assume that the noise covariance matrix Ξ is such that $\Delta_k = W^T S_k W$ has the following form:

$$\Delta_k = \left(\begin{array}{cc} \boxed{\Sigma_k} & \mathbf{0} \\ \mathbf{0} & \boxed{\begin{array}{cccc} \beta_k & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & \beta_k \end{array}} \end{array} \right) \begin{array}{l} \left.\vphantom{\Sigma_k}\right\} \quad d \leq K-1 \\ \left.\vphantom{\begin{array}{c} \beta_k \\ \ddots \\ \ddots \\ \beta_k \end{array}}\right\} \quad (p-d) \end{array}$$

where $W = [U, V]$.

This model is referred to by $\mathrm{DFM}_{[\Sigma_k \beta]}$ and 11 submodels can be obtained by constraining parameters within or between groups.

The DFM model

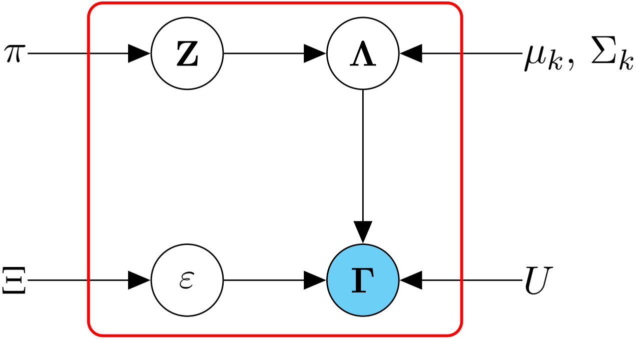


Figure 19: Graphical representation of the model $\mathrm{DFM}_{[\Sigma_k \beta]}$.

The DFM models

Model	$\boldsymbol{\Sigma_k}$	$\boldsymbol{\beta_k}$	Nb. of variance parameters
$\mathrm{DFM}_{[\Sigma_k \beta_k]}$	Free	Free	$(K-1)(p-K/2)+K^2(K-1)/2+K$
$\mathrm{DFM}_{[\Sigma_k \beta]}$	Free	Common	$(K-1)(p-K/2)+K^2(K-1)/2+1$
$\mathrm{DFM}_{[\Sigma \beta_k]}$	Common	Free	$(K-1)(p-K/2)+K(K-1)/2+K$
$\mathrm{DFM}_{[\Sigma \beta]}$	Common	Common	$(K-1)(p-K/2)+K(K-1)/2+1$
$\mathrm{DFM}_{[\alpha_{kj} \beta_k]}$	Diagonal	Free	$(K-1)(p-K/2)+K^2$
$\mathrm{DFM}_{[\alpha_{kj} \beta]}$	Diagonal	Common	$(K-1)(p-K/2)+K(K-1)+1$
$\mathrm{DFM}_{[\alpha_k \beta_k]}$	Spherical	Free	$(K-1)(K-1)(p-K/2)+2K$
$\mathrm{DFM}_{[\alpha_k \beta]}$	Spherical	Common	$(K-1)(p-K/2)+K+1$
$\mathrm{DFM}_{[\alpha_j \beta_k]}$	Diagonal & Common	Free	$(K-1)(p-K/2)+(K-1)+K$
$\mathrm{DFM}_{[\alpha_j \beta]}$	Diagonal & Common	Common	$(K-1)(p-K/2)+(K-1)+1$
$\mathrm{DFM}_{[\alpha \beta_k]}$	Spherical & Common	Free	$(K-1)(p-K/2)+K+1$
$\mathrm{DFM}_{[\alpha \beta]}$	Spherical & Common	Common	$(K-1)(p-K/2)+2$

Figure 20: The 11 submodels of the model $\mathrm{DFM}_{[\Sigma_k \beta]}$.

Inference: the FunFEM algorithm

We propose to rely for inference on an EM algorithm:

- the EM algorithm iteratively maximize the model likelihood by maximizing a lower bound $\mathcal{L}(q(Z);\theta)$:
  $$\log(p(X|\theta)) = \mathcal{L}(q(Z);\theta) + KL(q(Z)||p(Z|X,\theta)),$$
  where:
  - $\mathcal{L}(q(Z);\theta) = \sum_Z q(Z)\log(p(X,Z|\theta)/q(Z))$,
  - $KL(q(Z)||p(Z|X,\theta)) = -\sum_Z q(Z)\log(p(X|Z,\theta)/q(Z))$ is the KL divergence between $q(Z)$ and $p(Z|X,\theta)$.

Inference: the FunFEM algorithm

We propose to rely for inference on an EM algorithm:

- the EM algorithm iteratively maximize the model likelihood by maximizing a lower bound $\mathcal{L}(q(Z);\theta)$:

$$\log(p(X|\theta)) = \mathcal{L}(q(Z);\theta) + KL(q(Z)||p(Z|X,\theta)),$$

  where:
  - $\mathcal{L}(q(Z);\theta) = \sum_Z q(Z)\log(p(X,Z|\theta)/q(Z))$,
  - $KL(q(Z)||p(Z|X,\theta)) = -\sum_Z q(Z)\log(p(X|Z,\theta)/q(Z))$ is the KL divergence between $q(Z)$ and $p(Z|X,\theta)$.

The EM algorithm:

- E step: θ is fixed and $\mathcal{L}$ is maximized over $q \Rightarrow q^*(Z) = p(Z|X,\theta)$
- M step: $\mathcal{L}(q^*(Z),\theta^{old})$ is now maximized over θ

$$\begin{aligned}\mathcal{L}(q^*(Z),\theta^{old}) &= \sum_Z p(Z|X,\theta^{old})\log(p(X,Z|\theta)/p(Z|X,\theta^{old}))\\ &= E[\log(p(X,Z|\theta)|\theta^{old}] + c.\end{aligned}$$

The FunFEM algorithm

The FunFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,

The FunFEM algorithm

The FunFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving:

$$\max_U \frac{Var\left[E\left[\omega(X)|Z\right]\right]}{Var\left[\omega(X)\right]}, \text{ wrt } \int u_j(t)u_l(t)dt = 0, \quad \forall j \neq l$$

where $\omega(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the function u.

The FunFEM algorithm

The FunFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving:

$$\max_U \frac{Var\left[E\left[\omega(X)|Z\right]\right]}{Var\left[\omega(X)\right]}, \text{ wrt } \int u_j(t)u_l(t)dt = 0, \quad \forall j \neq l$$

where $\omega(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the function u.

Proposition: U is solution of the generalized eigenproblem

$$\mathbf{\Gamma}'\mathbf{T}\mathbf{T}'\mathbf{\Gamma}W\nu = \eta\mathbf{\Gamma}'\mathbf{\Gamma}W\nu,$$

where $\mathbf{\Gamma} = (\gamma_{ij})_{i,j}$, $\mathbf{T} = \left(\frac{t_{ik}^{(q-1)}}{\sqrt{n_k^{(q-1)}}}\right)_{i,k}$ and $\mathbf{W} = \int_{[0,T]} \Psi(s)\Psi'(s)ds$.

The FunFEM algorithm

The FunFEM algorithm alternates over:

- **a E step** which computes the posterior probabilities $t_{ik} = E[z_{ik} = 1|y_i]$,
- **a F step** which determines the orientation matrix U according to the $t_{ik}^{(q)}$ by solving:

$$\max_U \frac{Var\left[E\left[\omega(X)|Z\right]\right]}{Var\left[\omega(X)\right]}, \text{ wrt } \int u_j(t)u_l(t)dt = 0, \quad \forall j \neq l$$

where $\omega(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the function u.

Proposition: U is solution of the generalized eigenproblem

$$\mathbf{\Gamma}'\mathbf{T}\mathbf{T}'\mathbf{\Gamma}W\nu = \eta\mathbf{\Gamma}'\mathbf{\Gamma}W\nu,$$

where $\mathbf{\Gamma} = (\gamma_{ij})_{i,j}$, $\mathbf{T} = \left(\frac{t_{ik}^{(q-1)}}{\sqrt{n_k^{(q-1)}}}\right)_{i,k}$ and $\mathbf{W} = \int_{[0,T]} \Psi(s)\Psi'(s)ds$.

- **a M step** which updates the mixture parameters.

Model selection

Model selection:

- there remains two open problems that can be solved through model selection:
  - choosing the number K of groups,
  - choosing the most appropriate model of the DFM family.

Model selection criteria:

- the most popular criterion is probably BIC (Schwarz, 1978):

$$\mathrm{BIC}(\mathcal{M}) = \ell(\hat{\theta}) - \frac{\xi(\mathcal{M})}{2} \log(n),$$

- a more recent and “data driven” criterion is the slope heuristic (Birgé & Massart, 2004):

$$\mathrm{SHC}(\mathcal{M}) = \ell(\hat{\theta}) - 2\,\hat{s}\,\xi(\mathcal{M}),$$

where $\hat{s}$ is estimated from the data.

The data

The data set:

- 1 month of station occupancy data collected on the Paris' Velib system,
- the data were collected every 1 hour over 5 weeks (February, 24 - March, 30, 2014),
- we normalized the number of available bikes by the station size and get a loading profile for each station,
- the final data set contains 3230 loading profiles sampled at 1448 time points,
- the curves were finally smoothed on a basis of 41 Fourrier functions.

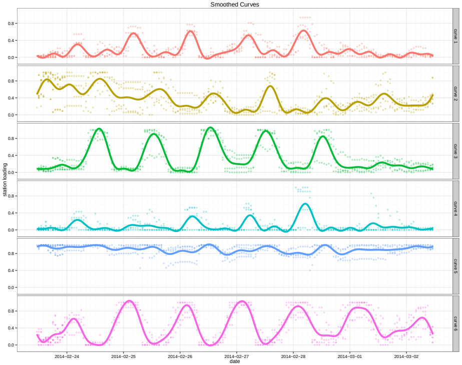


Figure 21: Some examples of the smoothed station profiles, with the corresponding observations.

The Vélib system

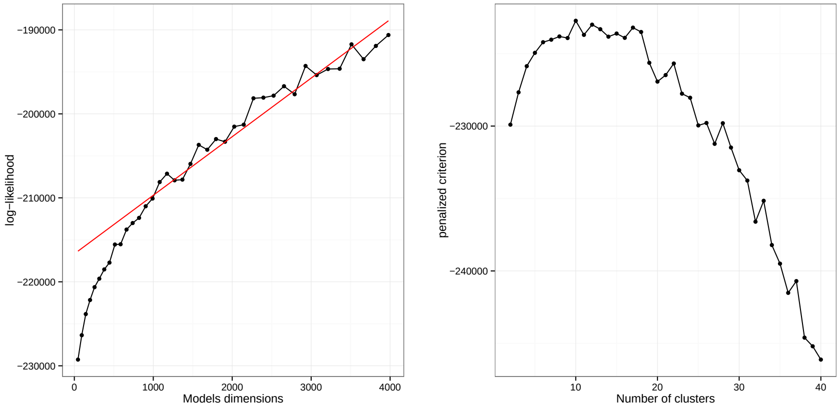


Figure 22: Model selection for Paris data: log-likelihood with respect to model dimensionality and its estimated linear part (left), slope heuristic criterion with respect to K (right).

The Vélib system

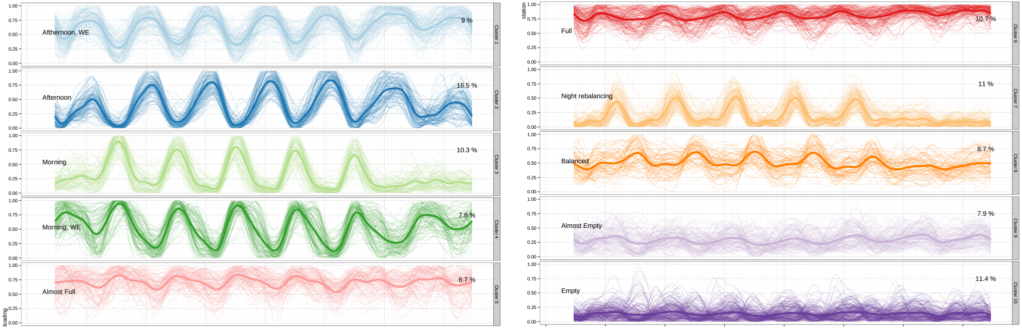


Figure 23: Cluster mean profiles together with 1000 randomly sampled curves.

The Vélib system

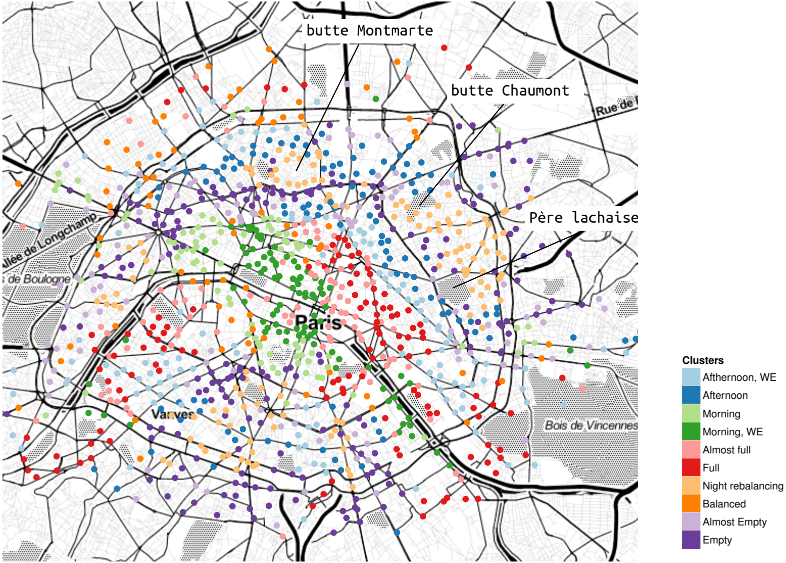


Figure 24: Map of the clustering results.

Conclusion

Conclusion

Model-based classification for HD or functional data:

- is an efficient and flexible tool for classification / clustering,
- it provides in addition information about the classification risk.

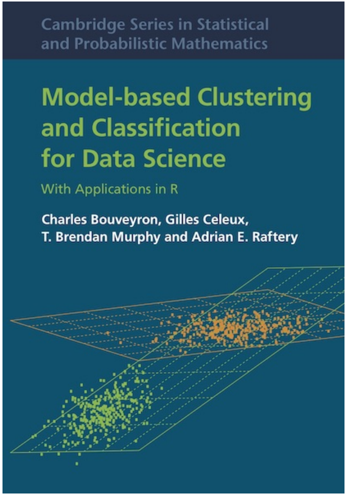
Our contributions:

- we proposed two models adapted to the classification of HD data and their associated inference algorithms,
- they model and cluster the data in low-dimensional (and discriminative) subspaces,
- they usually performs better than other clustering methods while providing a useful visualizations,
- they allow in addition to identify the original variables which are discriminative.

Software (R packages):

- package `HDclassif` for the HDDA and HDDC methods,
- package `FisherEM` for the Fisher-EM algorithm,
- package `funFEM` for the fun-FEM algorithm.

References



Cambridge Series in Statistical and Probabilistic Mathematics

Model-based Clustering and Classification for Data Science

With Applications in R

Charles Bouveyron, Gilles Celeux,
T. Brendan Murphy and Adrian E. Raftery

https://math.unice.fr/~cbouveyr/MBCbook/