

# Model-based Statistical Learning



Pr. Charles BOUVEYRON

Professor of Statistics  
Chair of the Institut 3IA Côte d'Azur  
Université Côte d'Azur & Inria

[charles.bouveyron@univ-cotedazur.fr](mailto:charles.bouveyron@univ-cotedazur.fr)  
[@cbouveyron](https://twitter.com/cbouveyron)

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

$\Rightarrow$  if  $n$  is too small compared to  $p$ , it won't be possible to run an EM algorithm with a full-GMM!

# The curse of dimensionality

From the estimation point of view:

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

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

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

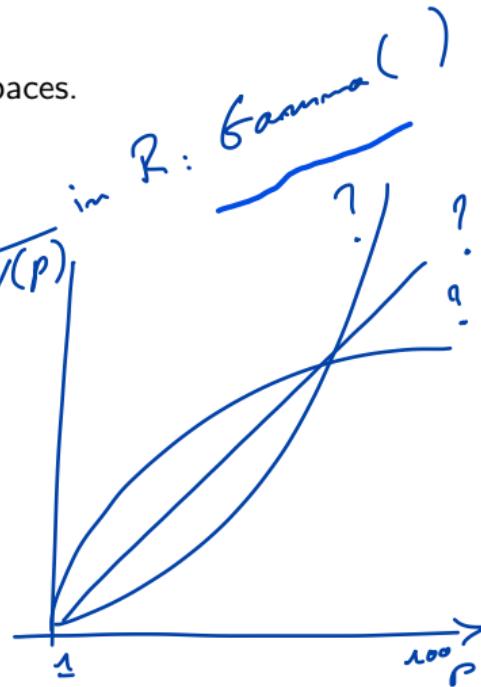
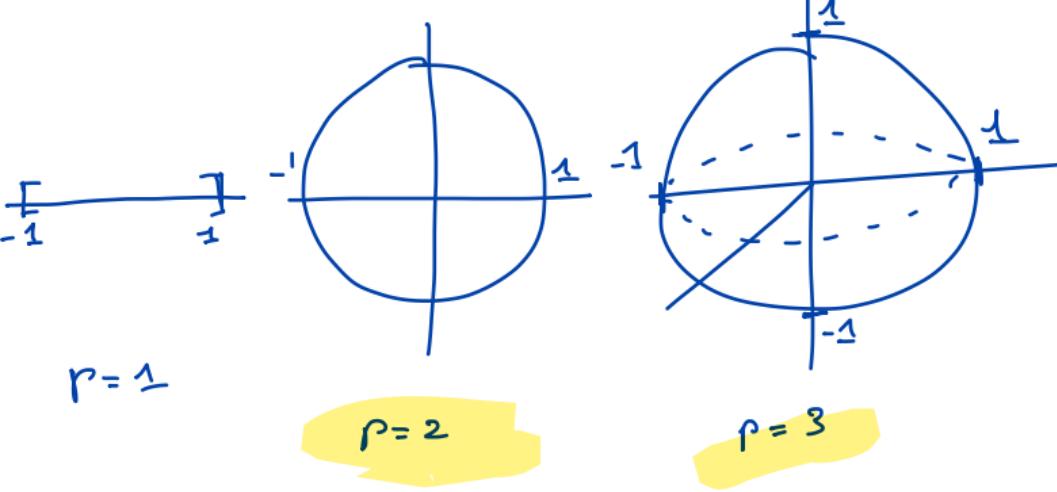
- consequently, if the ratio  $p/n \rightarrow 0$  when  $n \rightarrow +\infty$ , then  $E[\hat{\tau}(\hat{\Sigma})] \rightarrow \tau(\Sigma)$ ,
- however, if the dimension  $p$  is comparable with  $n$ , then  $E[\hat{\tau}(\hat{\Sigma})] \rightarrow c\tau(\Sigma)$  when  $n \rightarrow +\infty$ ,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

# 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  $V(p) = \frac{\pi^{p/2}}{\Gamma(p/2+1)}$ ,

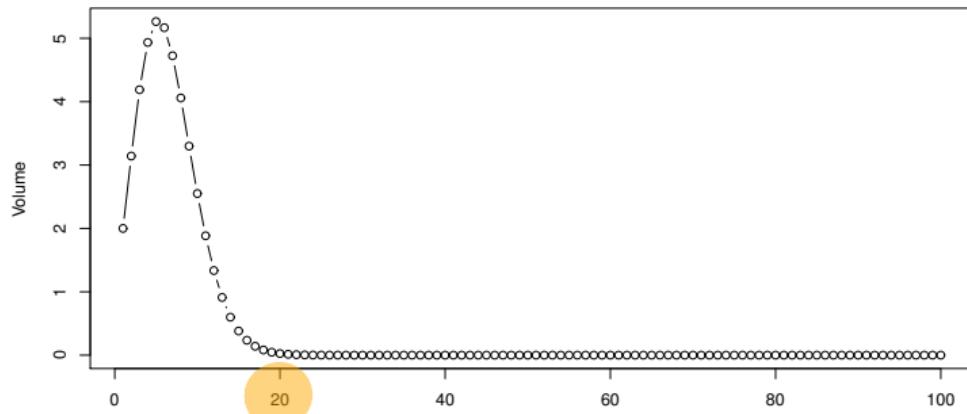


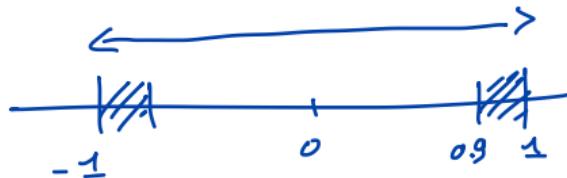
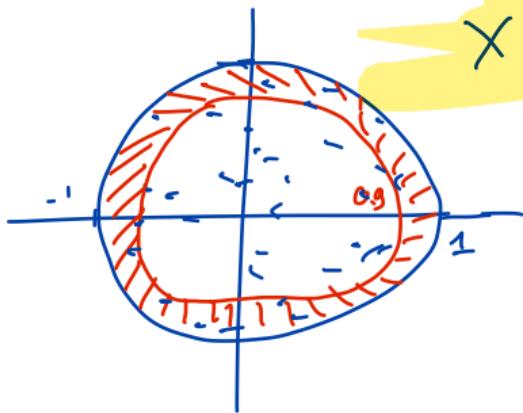
Fig. Volume of a sphere of radius 1 regarding to the dimension  $p$ .

## 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

$$P(X \in S_{0.9}(p)) = 1 - 0.9^p \xrightarrow{p \rightarrow \infty} 1$$

$X \sim U(\text{HS}1)$

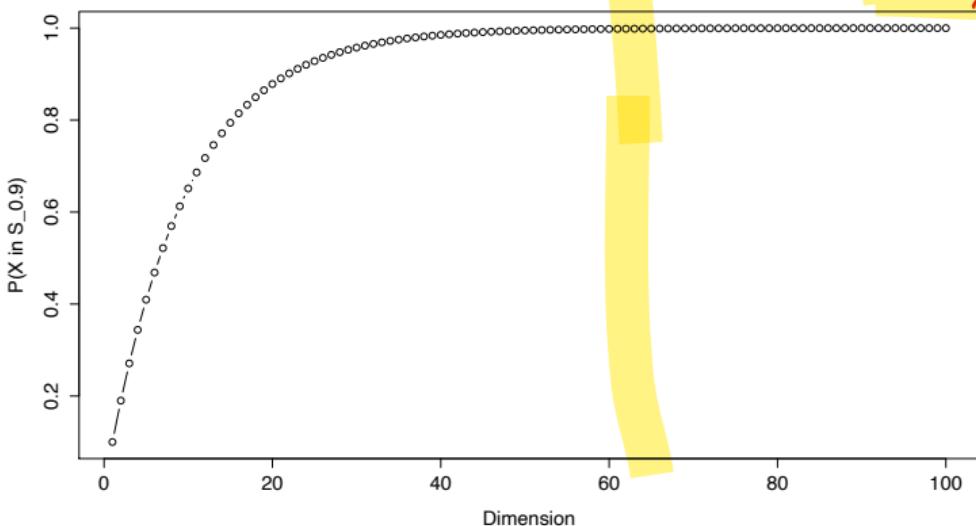


$$P(X \in S_{0.9}) = 0.1$$

# 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

$$P(X \in S_{0.9}(p)) = 1 - 0.9^p \xrightarrow{p \rightarrow \infty} 1$$



A) The message here is that HD spaces are mostly empty. The data are located in subspaces of lower dimensionalities.  
⇒ the "empty space phenomenon"

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

# The blessings of dimensionality

---

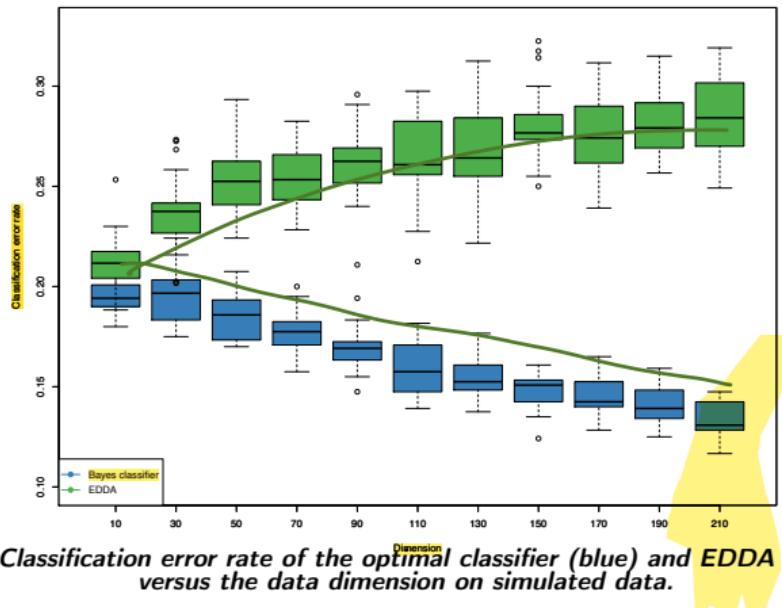
## Third example:

- since high-dimensional spaces are almost empty,
- it should be easier to separate groups in high-dimensional space with an adapted classifier,
- a way to observe this is to look at the Bayes classifier behaviour.

# 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,
- a way to observe this is to look at the Bayes classifier behaviour.



## 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"!

# Outline

---

1. Introduction
2. Reminder on the learning process
3. Learning in high-dimensions
4. Dimension reduction
5. Clustering and classification

# Dimension reduction

A common phantasm about dimension reduction:

- believe that dimension reduction helps for classification,
- this is not true because, most of the time, dimension reduction implies an information loss which would be discriminative.

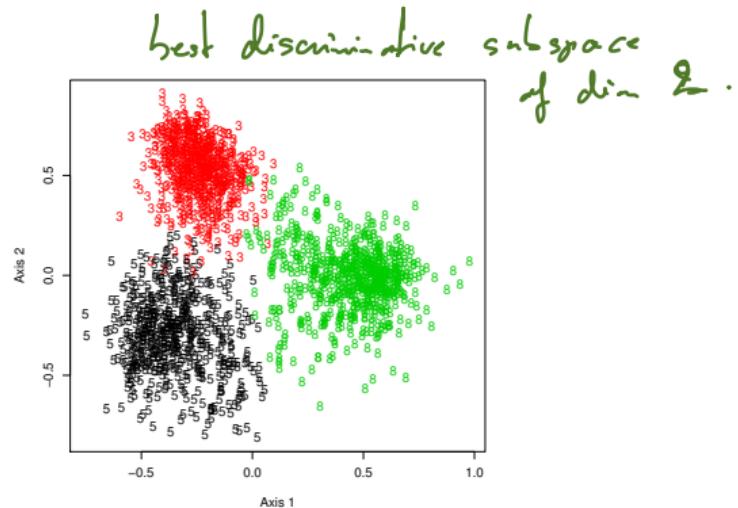
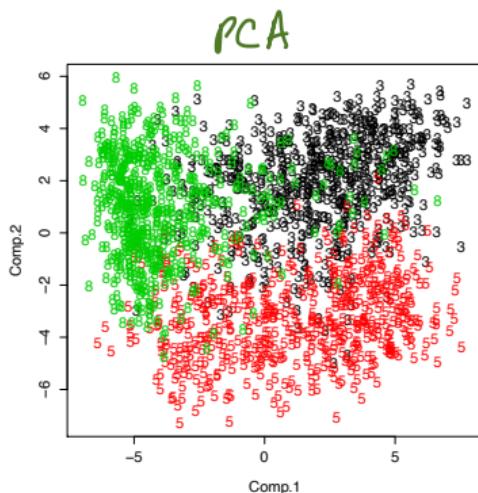
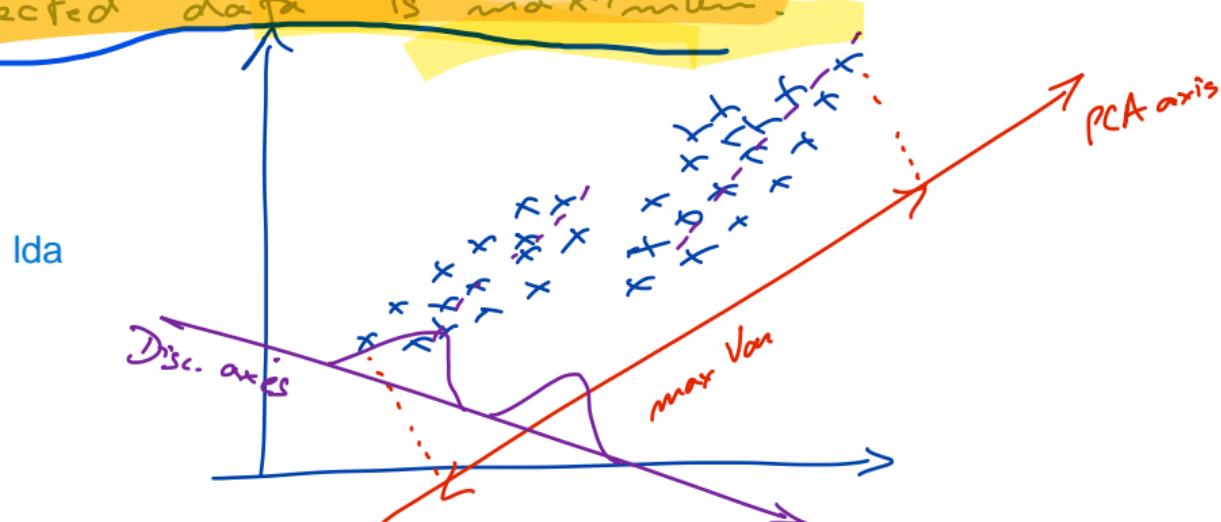


Figure: Projection of the 256-dimensional USPS data with PCA (left, unsupervised) and FDA (right, supervised).

## PCA: principal component analysis

The goal of PCA is to find  $d$  new axes build on the  $p$  original axes by linear combinations such that the variance of the projected data is maximum.



## PCA: the principle

The goal of PCA :

$$\underset{U \text{ pxd}}{\operatorname{arg\,max}} \operatorname{Var}(\underline{\underline{X}}U)$$

← projection of  
 $\underline{\underline{X}}$  in the d-  
dimensional space

Solution : the d PCA axes  $u_1, \dots, u_d$  are the  
eigen vectors of  $\underline{\underline{X}}^t \underline{\underline{X}} = S$  associated with the largest  
eigenvalues  $\lambda_1, \dots, \lambda_d$  of  $S$ .

In practice : 1) compute  $S = \underline{\underline{X}}^t \underline{\underline{X}}$  where  $\underline{\underline{X}}$  are the centered  
data  
2) do the eigen decomposition of  $S$ .  
3) keep the eigenvectors of  $S$  associated with the largest  $\lambda_j$ .

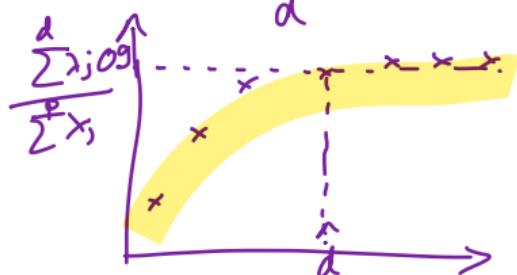
## PCA: the principle

How to choose  $d$ ?:

1) the "elbow" method: plot the eigenvalue scree and select  $d$  in the place of the elbow in the scree

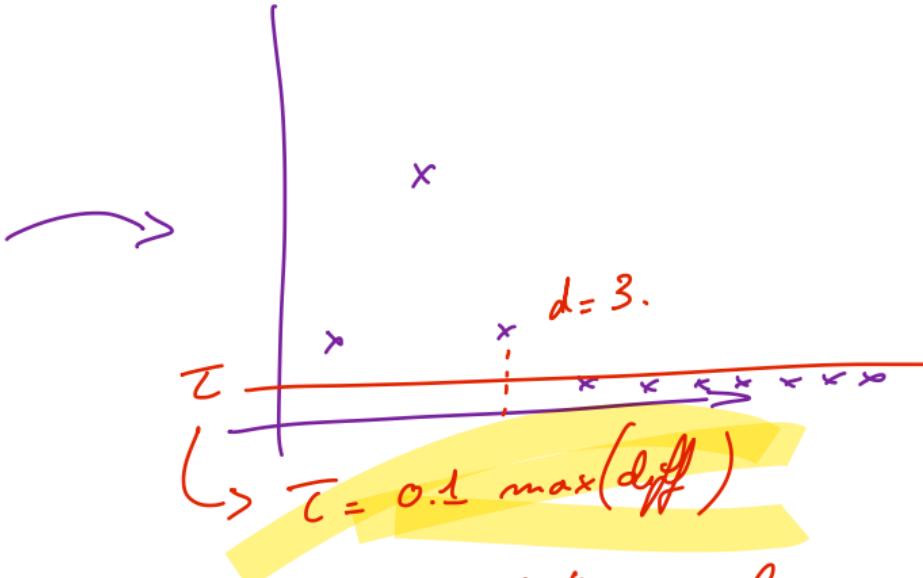
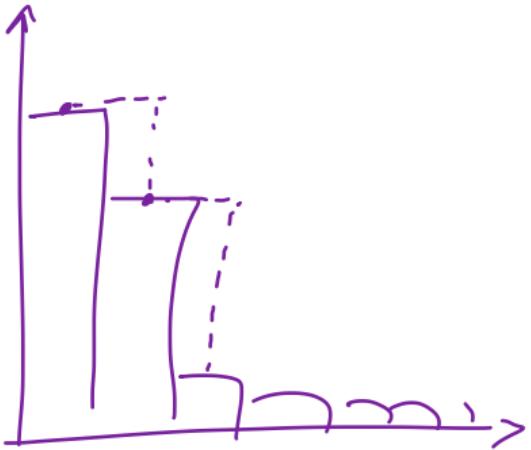


2) the "90%" rule:

$$\hat{d} = \underset{d}{\operatorname{argmin}} \frac{\sum_{j=1}^d \lambda_j}{\sum_{j=1}^p \lambda_j} / \frac{\sum_{j=1}^p \lambda_j}{\sum_{j=1}^p \lambda_j} \geq 90\%.$$


3) the scree-test of Cattell

Compute the differences between consecutive eigenvalues and select  $d$  when all differences after  $d$  are smaller than a threshold



A general remark: in case of hesitation, it is better to keep too much dimensions than too few!

## PCA: projection

$$Y = X \cdot \hat{U}$$

$m \times d$     $m \times p$     $p \times d$

will contain the projected data points on the d-dimensional PCA subspace.

↑  
principal scores

the loading matrix -

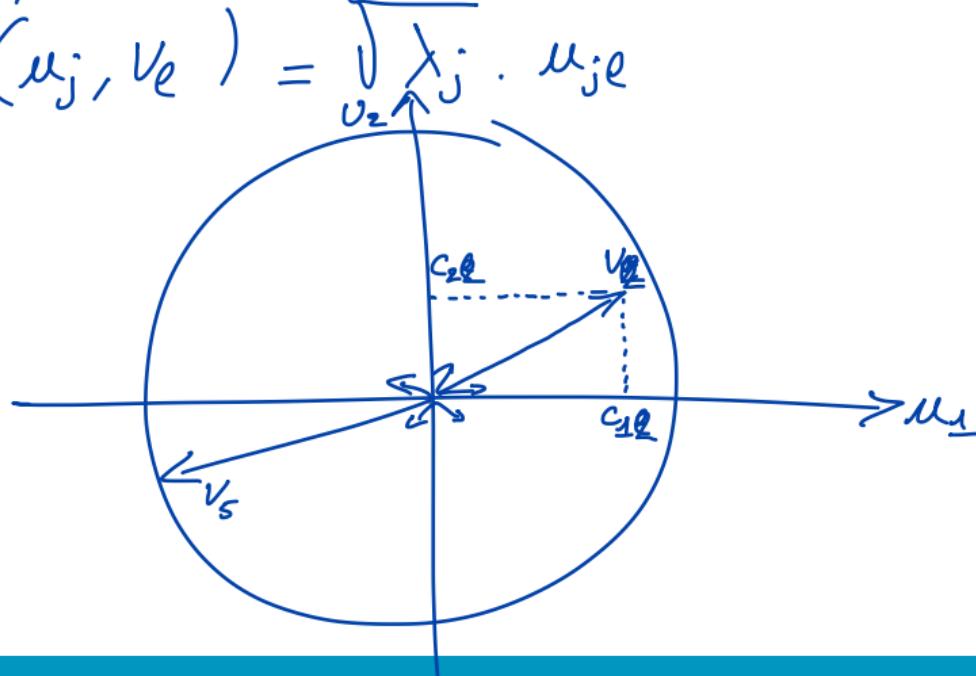
## PCA: how many axes?

## PCA: how many axes?

## PCA: correlation circle

It is easy to relate the PCA axes with the original variables by calculating the correlation between them:

$$c_{je} = \text{Cor}(u_j, v_e) = \sum u_{je}$$



## PCA: analysis of the marathon data

## PPCA: a probabilistic version of PCA

Probabilistic PCA (Tipping & Bishop, 1999) extends PCA in a statistical framework. PPCA justifies PCA under the Gaussian assumption.

PPCA assumes a linear link between  $X$  the observed variable and  $Y$  a **latent** low-dimensional variable:

$$X = U^t Y + \varepsilon$$

where  $X \in \mathbb{R}^P$   
 $Y \in \mathbb{R}^d$  with  $d \ll P$   
 $\varepsilon \in \mathbb{R}^P$  is a noise term.  
 $U$  is a  $p \times d$  matrix

## PPCA: a probabilistic version of PCA

PPCA adds some statistical assumptions:

$$\epsilon \sim N(0, \sigma^2 I_p)$$

$$y \sim N(\mu, I_d) \quad \text{with } \mu \in \mathbb{R}^d$$

Interestingly, the marginal distribution of  $x$  is:

$$x \sim N(U\mu + UU^T\sigma^2 I_p)$$

Let's introduce  $\Theta = \{\mu, \sigma^2, U\}$ , the set of model parameters

Thanks to this statistical model, we have now a new way to perform PCA: estimate the model parameters of PPCA from the data.

performing PCA ( $\Rightarrow$  estimate the models of PPCA).

Maximum Likelihood can be used for that, through the EM algorithm.

Theorem: the estimation of  $U$  through the EM algo is the eigenvectors of  $S = \bar{X}^t \bar{X}$  associated with the largest eigenvalues of  $S$ .

## PPCA: why is it interesting?

- i) PPCA offers a theoretical justification of PCA under the Gaussian assumption.
- ii) it also helps to understand the limitation of PCA
- iii) we can rely on model selection (AIC, BIC, ...) to select  $d$
- iv) new statistical models can be elaborated from PPCA.

## PPCA in practice...

- 1) Let's do PCA as before !
- $\hat{U} = \text{eigen}(\text{cov}(X))$
- $\hat{\mu} = \text{mean}(X\hat{U})$
- $\hat{\Sigma} = \text{mean}(\text{diag}(\text{cov}(X - X\hat{U}\hat{U}^T)))$
- 2) Use BIC to select d:

$$\text{BIC}(d) = \log L(X; \hat{\theta}_{\text{PPCA}}) - \frac{\gamma(d)}{2} \log(n)$$

where  $\gamma(d) = d + \frac{1}{\hat{\sigma}^2} + p \frac{(d+1)}{2}$

$\begin{matrix} \uparrow & \uparrow & \uparrow \\ \mu & \hat{\sigma}^2 & \hat{U} \\ & & p \times d \end{matrix}$

# Outline

---

1. Introduction
2. Reminder on the learning process
3. Learning in high-dimensions
4. Dimension reduction
5. Clustering ~~and classification~~ in HD spaces

## The limits of GMM in high dimensional spaces

- ⊕ Empty HD spaces  $\Rightarrow$  good for clustering
  - ⊖ Model-based approaches are usually over-parametrized  
which is problem when learning!
- $\Rightarrow$  The solution: adapt the statistical techniques  
to HD spaces by taking into account the  
properties of those spaces.

# Parsimonious models for GMM

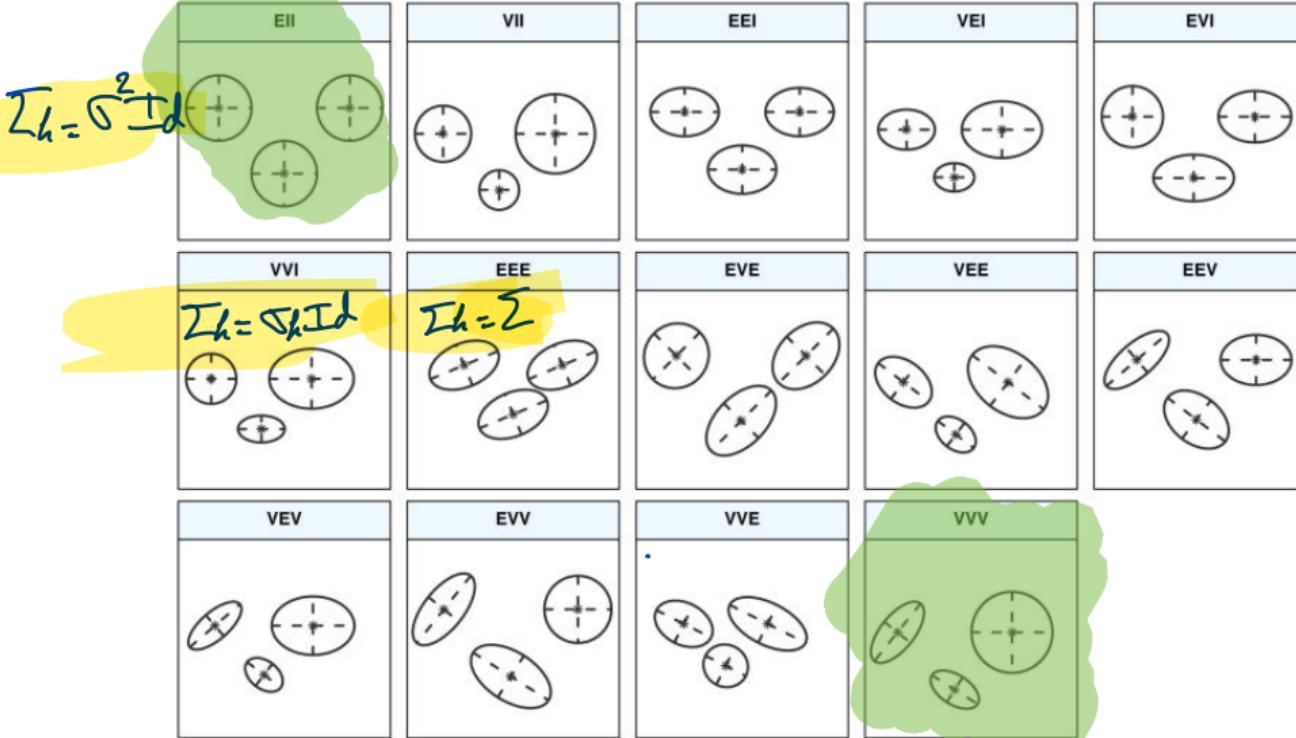
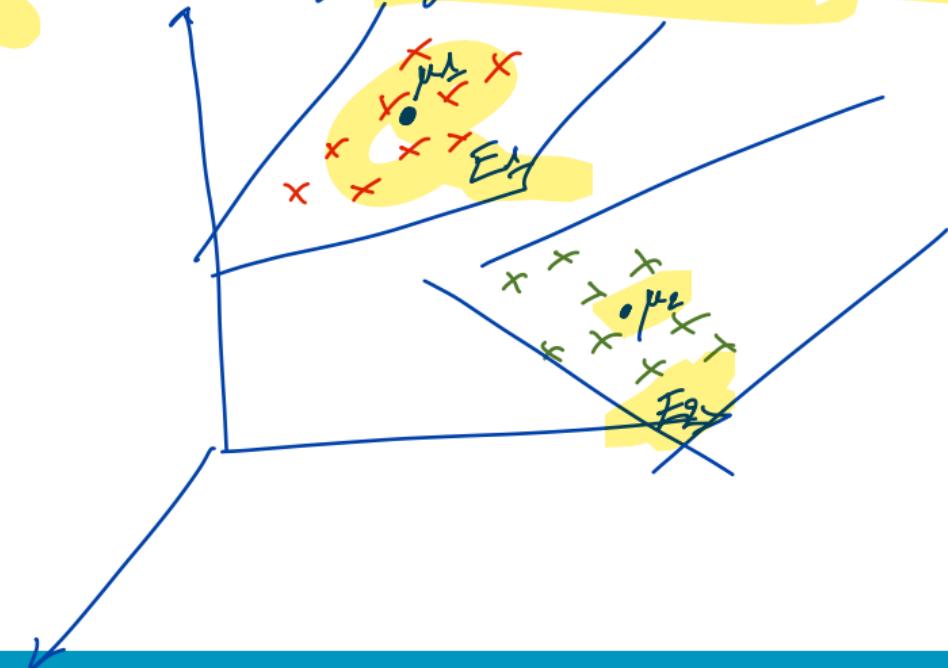


Figure: The parsimonious models of Mclust.

## The solutions : Subspace clustering

The idea of subspace clustering is to assume that each group leaves in a specific low-dimensional subspace.



# Model-based Statistical Learning



Pr. Charles BOUVEYRON

Professor of Statistics  
Chair of the Institut 3IA Côte d'Azur  
Université Côte d'Azur & Inria

[charles.bouveyron@univ-cotedazur.fr](mailto:charles.bouveyron@univ-cotedazur.fr)  
[@cbouveyron](https://twitter.com/cbouveyron)

## The mixture of PPCA

Tipping & Bishop proposed in 1996 the mixture of PPCA

$$Z \sim \mathcal{H}(1; \pi) \quad \pi = (\pi_1, \dots, \pi_K)$$

$$X_{|Z=h} = \gamma U_h + \epsilon_h \quad \text{where } U_h \text{ is a } p \times d \text{ matrix}$$

$$\epsilon_h \sim N(0, \Sigma_h \mathbb{I}_p)$$

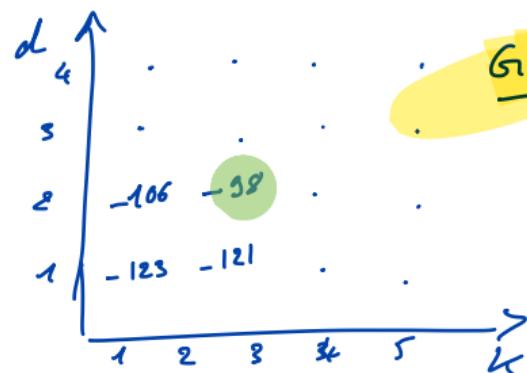
$$Y|Z=h \sim N(\mu_h, I_d)$$

$$\Rightarrow X \sim \sum_{h=1}^K \pi_h \phi(\mu_h U_h, U_h^t U_h + \Sigma_h^{-1})$$

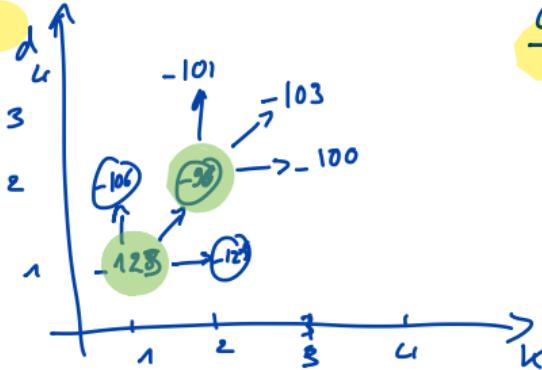
## The mixture of PPCA

The inference of MPPCA can be done using the EM algorithm.

The choice of both  $K$  and  $d$  can be done using model selection, with BIC for instance.



Grid search



Gibbs alg<sup>o</sup>

## HDDC (High Dimensional Data Clustering, Bouveyron 06)

The idea of HDDC is to extend NPPCA on 2 directions:

- 1) allows the intrinsic dimensions of the subspaces to be different :  $d \rightarrow d_k$ .
- 2) allows sub-models of NPPCA to fit in different situations with fewer data.

## HDDC

the model of HDDC,

$$Z \sim \text{cl}(1, \pi)$$

$$X|_{Z=h} = YU_h + \varepsilon_h$$

where  $U_h$  is a  $p \times d_h$  orthonormal matrix

$$\varepsilon_h \sim N(0, \Sigma_h^2 I_p)$$

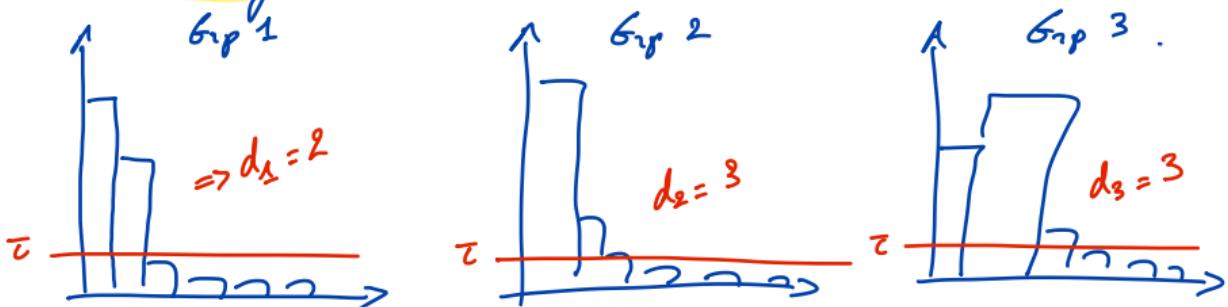
$$Y \sim N(\mu_h, \alpha_h I_{d_h}) \quad \text{where } \alpha_h \in \mathbb{R}^{d_h}$$

$$\Rightarrow X \sim \sum_{h=1}^n \pi_h N\left(\mu_h U_h, (\alpha_h U_h)(\alpha_h U_h)^T \Sigma_h^2 I_p\right)$$

## HDDC

Inference this HDDC model is still possible with ER, but model selection is now very difficult because we have  $(K+1)$  discrete hyper-parameters to choose.

A solution to avoid this combinatorial problem: the scree-test of Cattell



$\Rightarrow$  Model selection here resumes to choosing  $K$  and  $T$ .

In HDDC, it is also possible to get more parsimonious models by constraining some model parameters to be common between or within groups.

$$\forall h, d_h = d \Rightarrow \text{PPCA}$$

$$\forall h, \sigma_h^2 = \sigma^2 \Rightarrow [a_{hj} \in Q_h d_h]$$

$$a_{hj} = a_h \Rightarrow [a_h b_h Q_h d_h]$$

$$\sigma_h^2 = \sigma^2 \& a_{hj} = a_h \Rightarrow [a_h b \in Q_h d_h]$$

(...)

$$\Rightarrow [ab \in Qd]$$

$\Rightarrow$  the R Library `HDDC` classif implements it.

$\Rightarrow$  the Python package `HDDA` for Python [github.com/mfauvel/HDDA](https://github.com/mfauvel/HDDA)

}

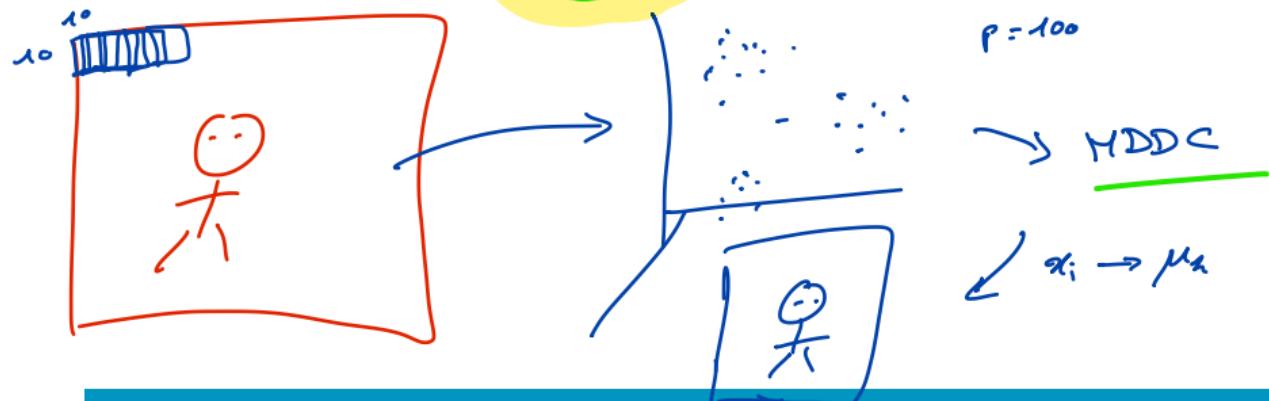
28 submodels.

which can be selected thanks to model selection.

## In the supervised case: HDDA

In the supervised case, HDDA implements all the 28 models (including MPPCA) for supervised classification.

A remark: HDDC for image denoising turns out to be extremely performant: HDTI method implements it.



## Conclusion on subspace clustering / classif.:

- ⊕ extremely performant models for HD data.
- ⊖ no easy way to visualize the data in the latent subspaces.

↳ The Fisher-EP algorithm aims to solve this visualization problem while keeping the idea of subspace clustering.