Machine/Statistical Learning

Lecture 5: Unsupervised classification
K-means and Mixture models

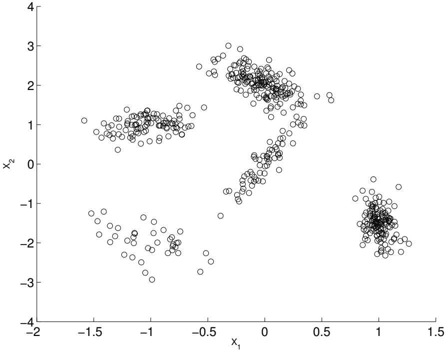
Filière SICOM, 3A

Unsupervised classification

Assumptions

- $X \in \mathbb{R}^p, \quad Y \in \{1, \dots, K\} \leftarrow K$ classes
- Training set $(x_1, \dots, x_n)$ $\leftarrow$ unknown outputs y_i

Example ($p = 2$)

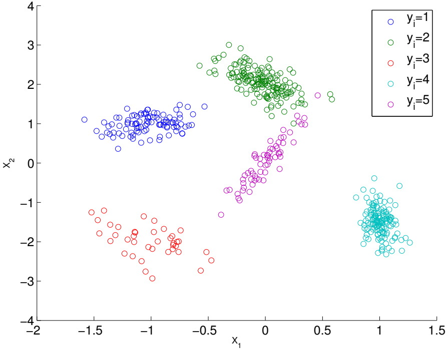


Unsupervised classification : Clustering

Objectives

- grouping similar data in the same cluster ← clustering
- ☞ For each x_i, $1 \leq i \leq n$, predict the class variable $Y_i \in \{1, \ldots, K\}$

Example ($p = 2$)



True classes y_i for data x_i ($K = 5$)

Clustering limitations

Combinatorics problem

- Number of partitions into K classes for a sized n dataset : *Stirling number* of the 2nd kind $S(n, K)$
- Number of partitions for a sized n dataset : *Bell number* $B_n = \sum_{k=1}^{n} S(n, k)$

dataset size n	2	5	10	100	200
$S(n, 2)$ ($K = 2$ classes)	1	15	511	6.3×10^{29}	8.0×10^{59}
$S(n, 4)$ ($K = 4$ classes)	0	10	34105	6.7×10^{58}	1.1×10^{119}
B_n	2	52	115975	4.8×10^{115}	6.2×10^{275}

- Remember $\simeq 10^{80}$ atoms in the Universe...

Pb : Exhaustive search (brute-force) not possible in practice

☞ local search around initial solutions/values $\rightarrow$ sub-optimal

Estimation problem and model selection

- possible parameters are unknown $\leftarrow$ estimation
- Number of classes K possibly unknown $\leftarrow$ model selection

Mixture of distributions

- Data $X_1, \dots, X_n$ assumed to be i.i.d. with pdf f
- f is modeled as a *mixture of distributions*

$$f(x) = \sum_{k=1}^{K} \pi_k \phi\left(x; \theta_k\right)$$

- $\pi_1, \dots, \pi_k$ are the relative sizes ($\sum_{k=1}^{K} \pi_k = 1$) of the classes :

$$\Pr\left(Y_i = k\right) = \pi_k$$

- density ϕ is the parametric shape of a class,
- parameters $\theta_1, \dots, \theta_K$ are the *centroids* of the classes/clusters

Latent variable

$Y \in \{1, \dots, K\}$ indicating the class of the r.v. X

- $Y \sim$ discrete distribution s.t. $\Pr\left(Y_i = k\right) = \pi_k, \quad k = 1, \dots, K$
- $X|Y = k \sim$ distribution with pdf $\phi\left(\cdot|\theta_k\right)$

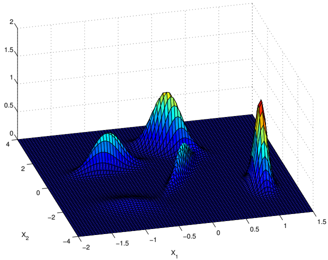
Gaussian mixture model

- Class centroid : $\theta = (\mu \leftarrow \text{mean}, \Sigma \leftarrow \text{covariance matrix})$
- Density ϕ of a class : multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ pdf

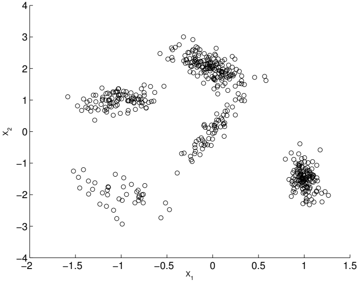
$$\phi(x;\mu,\Sigma) = \left(\det\left(2\pi\Sigma\right)\right)^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

- Mixture density $\quad f(x) = \sum_{k=1}^{K} \pi_k \phi\left(x;\mu_k,\Sigma_k\right)$

Example ($p = 2,\ K = 5$)



Mixture density f



$n = 500$ realizations

Cost based approximation : K-means

Pb : no simple expression of the Gaussian mixture parameter estimators

☞ several approximations can be conducted to obtain a simple *deterministic* cost criterion

First approximation : euclidean distance

Replace the Mahalanobis distance in the Gaussian density by the simpler euclidean one

$$(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) \to \|x-\mu_k\|^2, \quad (\text{i.e. } \Sigma_k = I_p)\,,$$

☞ cluster centroid for the kth class reduces to $\theta_k = \mu_k \leftarrow$ mean vector

Cost based approximation : K-means (Cont'd)

Pb : no straightforward expression of the Gaussian mixture parameter estimators

☞ several approximations can be conducted to obtain a simple *deterministic* cost criterion

Second approximation : hard thresholding

Binarize the posterior probabilities : for each data point x_i,

$$t_{i,k} \equiv \Pr\left(Y_i = k | x_i, \theta\right) = \begin{cases} 1 & \text{if } k = \arg\min_{1 \leq j \leq k} \|x_i - \mu_k\|, \\ 0 & \text{otherwise.} \end{cases}$$

Csq : x_i belongs with certainty to the class whose centroid is the closest

☞ hard thresholding clustering

☞ deterministic model

Cost criterion : K-means clustering

Notations

For a given clustering Y, let

- $n_k = \#\{i \mid Y_i = k\}$ is the size of the kth cluster,
- $\widehat{\mu}_k = \frac{1}{n_k}\sum_{i|Y_i=k} x_i$ is the sample mean of the points assigned in the kth cluster

Under the previous approximations, maximizing the resulting "log-likelihood" reduces to the following optimization problem :

K-means cost criterion

$$\text{Minimize} \quad J(Y) = \sum_{k=1}^{K}\sum_{i=1}^{n} t_{i,k}\|x_i - \widehat{\mu}_k\|^2,$$
$$= \sum_{k=1}^{K}\sum_{i|Y_i=k} \|x_i - \widehat{\mu}_k\|^2,$$

☞ $J(Y)$ is the sum of within-cluster dispersions

Equivalent cost criterion

(negative) Sum of between-cluster dispersions

$$J(Y) = -\sum_{k=1}^{K} n_k \|\widehat{\mu}_k - m\|^2 + \text{constant},$$

where $m = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the total mean.

- ☞ Minimizing the within-cluster dispersion $\Leftrightarrow$ Maximizing the between-cluster dispersion
- ☞ General property of clustering algorithms

Proof : let $S_T = \sum_{j=1}^{n}(x_j - m)^T(x_j - m) = \sum_{k=1}^{K}\sum_{i|Y_i=k}\|x_i - m\|^2$ be the total dispersion.

- Replace x_i by $x_i - \widehat{\mu}_k + \widehat{\mu}_k$, and expand S_T
- Show that $S_T = J(Y) + \sum_{k=1}^{K} n_k\|\widehat{\mu}_k - m\|^2$ (i.e. the cross product equals zero), and conclude by noting that S_T does not depend on Y

K-means : cost criterion optimization

Enlarged optimization problem

$$\min_{Y,\mu} J(Y,\mu) = \sum_{k=1}^{K} \sum_{i|Y_i=k} \underbrace{\|x_i - \mu_k\|^2}_{J_k},$$

- J_K is the quadratic error for the kth cluster

Remarks

- For a given Y, $\min_\mu J(Y,\mu) = J(Y,\widehat{\mu}) \equiv J(Y)$
- For a given μ, exchanging $Y_i = k$ with $Y_i^\star = l$ changes the two quadratic errors

$$\left[\begin{array}{ll} J_k^\star & = J_k - \|x_i - \mu_k\|^2, \\ J_l^\star & = J_l + \|x_i - \mu_l\|^2, , \end{array} \right.$$

Thus $J(Y,\mu)$ is decreased if

$$\begin{array}{lll} & J_l^\star - J_l & \leq J_k - J_k^\star \\ \Leftrightarrow & \|x_i - \mu_l\|^2 & \leq \|x_i - \mu_k\|^2, \\ \Leftrightarrow & x_i \text{ is closer} & \text{(euclidean distance) from the class } l \text{ center,} \end{array}$$

K-means algorithm

- **Require :** K the number of clusters,
- **Initialization :** Set the centroid μ_k, $1 \leq k \leq K$, to a starting value $\mu_k^{(0)}$,
- **For** $t = 1 \to \dots$ **until convergence** $\left(\text{i.e. } \mu_k^{(t)} = \mu_k^{(t-1)}\right)$
  1. **Assignment step :** assign x_i to the class of the closest center
  $$Y_i^{(t)} = \arg \min_{k=1,\dots,K} \left\| x_i - \mu_k^{(t-1)} \right\|^2, \quad \text{for } i = 1, \dots, n$$
  2. **Update step :** update the centroids μ_k, for $k = 1, \dots, K$
  $$\mu_k^{(t)} = \arg\min_{\mu_k} \sum_{i \mid Y_i^{(t)} = k} \|x_i - \mu_k\|^2 = \frac{1}{n_k^{(t)}} \sum_{i \mid Y_i^{(t)} = k} x_i,$$
  i.e. $\mu_k^{(t)}$ is the sample mean of the kth cluster

Convergence of K-means algorithm

Convergence

- each step decreases the criterion,
- there is a (huge) finite number of partitions,

☞ the algorithm converges to a solution (in a finite number of steps)

But no guaranty of the solution optimality (depend on the initialization)...

Stopping criterion

K-means usually very fast for a small/moderate number of clusters K, but

- running time increases with the number of clusters K
- in the worst case, can be very slow to converge even for $K = 2$,

Thus, to shorten the computational time, the algorithm can be stopped when the cost criterion does not decrease significantly

Variants/Improvements of K-means algorithm

Initialization heuristics

- Forgy method
  - pick randomly K observations from the dataset as initial centers,
  - run K-means algorithm with these starting values
  - repeat these 2 steps several times and retain the best (cost sense) clustering
- lot of variants : **`Random partitions`**, **`k-means++`**, **`power init.`**

☞ may lower the computation time of one run,

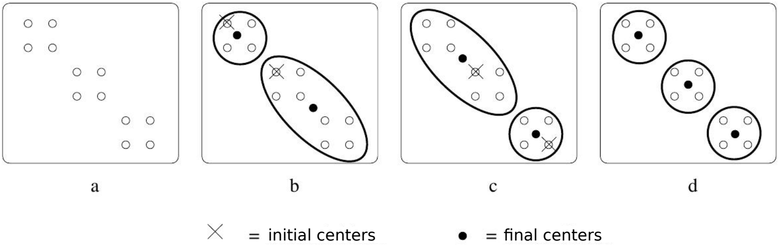
☞ can give some guaranties that the solution is competitive w.r.t. to the optimal one.

Choice of the distance

- Standard K-means based on the squared ℓ_2 (euclidean) distance.
- Other distance can be considered : e.g. using ℓ_1 distance yields the K-medians algorithm where the cluster centroid becomes the median (*Exercice : show this*, cf 2015-16 exam statement)

K-means initilization

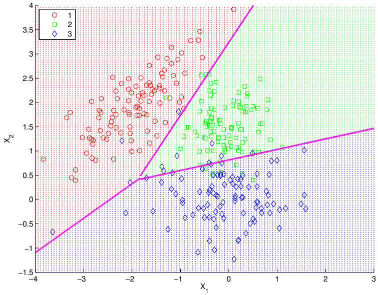
Sensitivity to initialization/data geometry/number of classes



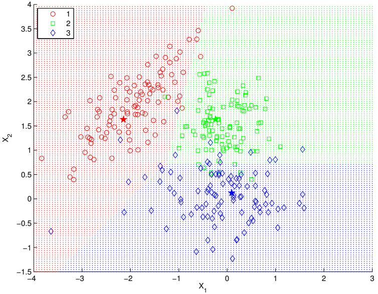
a) set of points $x_i \in \mathbb{R}^p$ ($p = 2$) to classify, b) and c) two clusterings in $K = 2$ classes with different initial centers, d) clustering in $K = 3$ classes.

K-means

Prediction vs Clustering



LDA (supervised approach)



K-means with $K = 3$ classes

- the points $x_1, \ldots, x_n$ are grouped according to the color of the regions
- Prediction : performance on *new* data is what matters
- Clustering : performance on *current* data is what matters

EM (Expectation-Maximization) algorithm

EM method is a general and important tool of statistical analysis :

- method for finding maximum likelihood (ML) or maximum a posteriori (MAP) estimates of parameters in statistical models, by maximizing iteratively the log-likelihood
- introduction of unobserved latent variables Z to decompose the optimization problem in simpler sub-problems in an iterative way
- EM iteration alternates between performing an expectation (E) step, and a maximization (M) step

EM (Expectation-Maximization) principle

- Z is a latent variable,
- Objective : maximize $\ell(\theta) = \log p(x|\theta)$

Sketch of EM algorithm

- **E step** : compute the expectation of the completed log-likelihood function evaluated using the current estimate for the parameter

$$\begin{aligned} Q\left(\theta, \theta^{(t-1)}\right) &= E_{Z|X,\theta^{(t-1)}}\left[\log p(x, z|\theta)\right], \\ &= \int p(z|x, \theta^{(t-1)}) \log p(x, z|\theta) dz \end{aligned}$$

- **M step** : compute parameters maximizing the expected log-likelihood

$$\theta^{(t)} = \arg\max_{\theta} Q\left(\theta, \theta^{(t-1)}\right),$$

- Repeat until convergence of the $\theta^{(t)}$ sequence

Application of EM to mixture models : E step

Introducing the latent variables Y_i, or equivalently, the binary variables

$$z_{ik} = \begin{cases} 1 & \text{if } Y_i = k, \\ 0 & \text{otherwise,} \end{cases}$$

the likelihood completed with the r.v. z_{ik} reads

$$p(x_1, \ldots, x_n, z|\theta) = \prod_{i=1}^{n} p(x_i, z|\theta) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k \phi(x_i|\theta_k)^{z_{ik}},$$

$$\begin{aligned} \Rightarrow \quad \log p(x_1, \ldots, x_n, z|\theta) &= \textstyle\sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \left[\pi_k \phi(x_i|\theta_k)\right], \\ \Rightarrow \quad Q\left(\theta, \theta^{(t-1)}\right) &= \textstyle\sum_{i=1}^{n} \sum_{k=1}^{K} \underbrace{E\left[z_{ik}|x_i, \theta^{(t-1)}\right]}_{t_{ik}^{(t-1)}} \log\left(\pi_k \phi(x_i|\theta_k)\right) \end{aligned}$$

$$\text{where} \quad t_{ik}^{(t-1)} = \Pr\left(Y_i = k \,\middle|\, x_i, \theta^{(t-1)}\right) = \frac{\pi_k^{(t-1)} \phi\left(x_i|\theta^{(t-1)}\right)}{\sum_{k=1}^{K} \pi_k^{(t-1)} \phi\left(x_i|\theta^{(t-1)}\right)}$$

Gaussian mixture models : M step

Find $\theta \equiv \theta^{(t)}$ maximizing $Q\left(\theta, \theta^{(t-1)}\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik}^{(t-1)} \log\left[\pi_k \phi(x_i|\theta_k)\right]$

- For any mixture model (i.e. $\forall\phi$) :

$$\pi_k^{(t)} = \frac{1}{n}\sum_{i=1}^{n} \theta^{(t-1)}$$

- For a Gaussian mixture model $\theta = \{\mu_k, \Sigma_k\}$ and

$$\mu_k^{(t)} = \frac{\sum_{i=1}^{n} t_{ik}^{(t-1)} x_i}{\sum_{i=1}^{n} t_{ik}^{(t-1)}},$$

$$\Sigma_k^{(t)} = \frac{\sum_{i=1}^{n} t_{ik}^{(t-1)} \left(x_i - \mu_k^{(t)}\right)\left(x_i - \mu_k^{(t)}\right)^T}{\sum_{i=1}^{n} t_{ik}^{(t-1)}},$$

- empirical averages weighted by the posterior probability in $\theta^{(t-1)}$, $t_{ik}^{(t-1)} \equiv \Pr\left(Y_i = k \,|\, x_i, \theta^{(t-1)}\right)$

☞ soft-thresholding algorithm

EM algorithm for Gaussian mixture models

EM clustering

- Initialize $\pi_k^{(0)}$, $\mu_k^{(0)}$, $\Sigma_k^{(0)}$, for $k = 1, \dots, K$
- For $t = 1, \dots$ until convergence

  (E) for $i = 1, \dots, n$, $k = 1, \dots, K$, compute $t_{ik}^{(t-1)} \equiv \Pr\left(Y_i = k \,|\, x_i, \theta^{(t-1)}\right)$

  (M) for $k = 1, \dots, K$, compute $\pi_k^{(t)}$, $\mu_k^{(t)}$, $\Sigma_k^{(t)}$

Prediction/Correction structure

- E step $\Leftrightarrow$ prediction step
- M step $\Leftrightarrow$ update/correction step

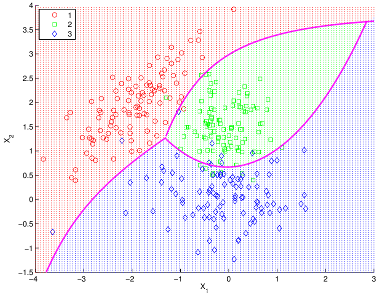
Convergence

- EM : convergence toward a local maximum of the log-likelihood

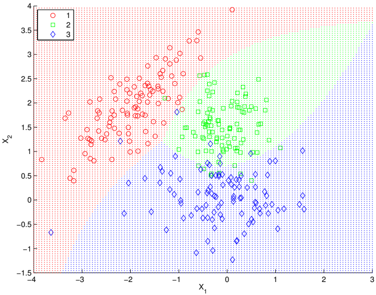
☞ no guaranty of convergence toward the optimal solution (depend on the initial values)..

Gaussian mixture model and EM algorithm

Prediction vs Clustering



QDA (supervised approach)

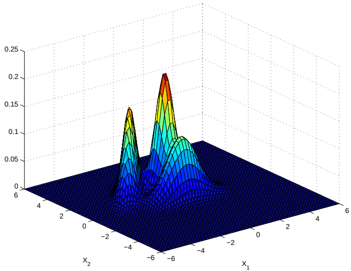


EM with $K = 3$ classes

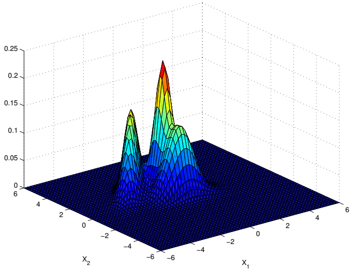
- the points $x_1, \ldots, x_n$ are grouped according to the color of the regions
- Prediction : performance on *new* data is what matters
- Clustering : performance on *current* data is what matters

Gaussian mixture model and EM algorithm

Estimation of the mixture density f



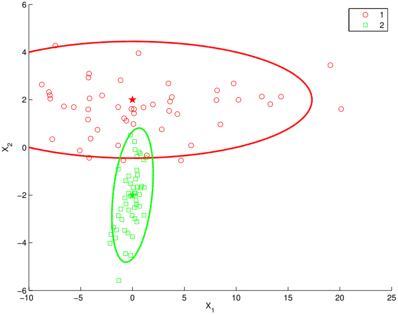
True density of the data points $x_1, \ldots, x_n$



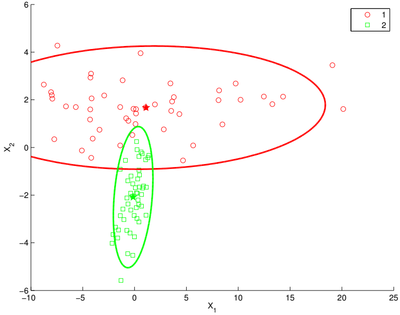
Estimated density with EM ($K = 3$ classes)

Comparison K-means vs Algo EM

2 classes with overlapping and very different dispersions (covariances Σ_k)



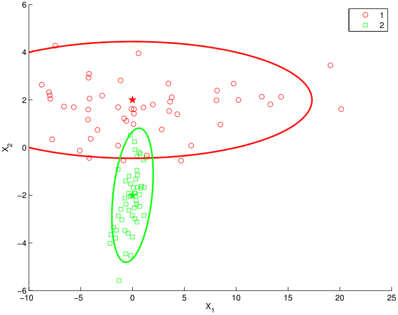
Data $x_1, \ldots, x_n$, classes and true 95% confidence regions



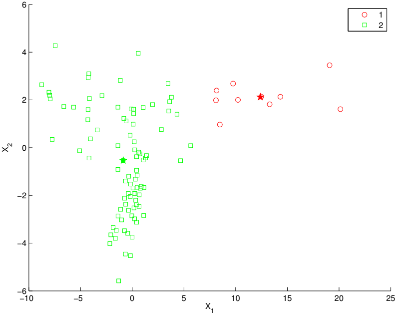
Clustering with EM ($K = 2$) and estimated 95% confidence regions

Comparison K-means vs Algo EM

2 classes with overlapping and very different dispersions (covariances Σ_k)



Data $x_1, \ldots, x_n$, classes and true 95% confidence regions



Classification with K-means ($K = 2$)

Model selection : estimation of K

Minimization of a penalized log-likelihood criterion

$$C(K) = -\hat{l}(x; K) + \text{pen}(K, n)$$

- $\hat{l}(x; K) \equiv l(x; \hat{\theta}_K, K)$ with $\hat{\theta}_K$ the MLE of the model parameters with K classes (profile log-likelihood w.r.t K)

Trade-off between two terms to minimize

- $-\hat{l}(x; K)$: fidelity to the data (likelihood)
- $\text{pen}(K, n)$: low complexity of the model

Model selection : BIC criterion

Bayesian Information Criterion (BIC)

Asymptotic ($n \gg m_K$) criterion for Bayesian models (i.e. with a prior on the model parameters)

$$\text{pen}(K, n) = \frac{1}{2} m_K \log(n)$$

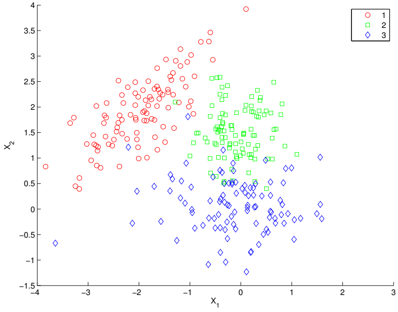
- n is the size of the data
- m_K is the effective number of parameters for the K class model

Equivalent to minimize the following criterion

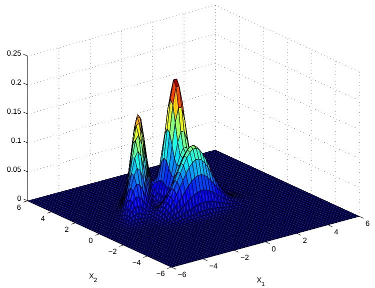
$$\text{BIC}(K) = -2\hat{l}(x; K) + m_K \log(n)$$

Model selection : estimation of K

Example of synthetic data generated according to a mixture of $K = 3$ Gaussians



Dataset $x_1, \ldots, x_n$ ($n = 500$ realizations)

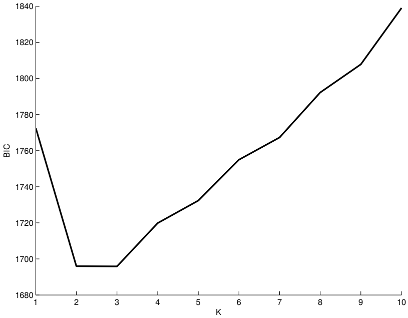


True density f

Model selection : estimation of K

Gaussian mixture : $m_K = \underbrace{K-1}_{\pi_1,\ldots,\pi_{K-1}} + K \times \underbrace{p}_{\mu_k} + K \times \underbrace{\frac{p(p+1)}{2}}_{\Sigma_k}$

BIC criterion w.r.t. K



$\Rightarrow$ $\hat{K} = 2$ or $\hat{K} = 3$ (true value $K = 3$)