Model-based clustering and classification

part 1: basics

Julien JACQUES

Université Lumière Lyon 2

Introduction

Clustering and classification

- **clustering** (unsupervised):

to group a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)

- **classification** (discrimination, scoring / supervised):

to predict the group of a new observation from a labeled sample

Clustering and classification

Notations

- observations are described by p features $\boldsymbol{X} = (X_1, \ldots, X_p) \in E$ $(E = \mathbb{R}^p, ...)$
- $\boldsymbol{X}_i = (X_{i1}, \ldots, X_{ip})$ is the features for observation i $(1 \leq i \leq n)$
- $Z_i \in \{1, \ldots, K\}$ is the group of observation i

Clustering versus classification

Clustering

- Z_i unknown
- goal: to predict $Z_1, \ldots, Z_n$ from $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$
- $Z_1, \ldots, Z_n$ are a posteriori interpreted in order to give significance to the clusters

Classification

- Z_i observed
- goal: to build a classification rule r from $(\boldsymbol{X}_1, Z_1), \ldots, (\boldsymbol{X}_n, Z_n)$:

$$r : \boldsymbol{X} \longrightarrow r(X) = Z$$

- to use this rule in order to classify new observation for which the group is unknown

Applications

Clustering

- exploratory analysis: to give a simplified representation of data in order to understand them
- example:
  - to recognize communities in social networks
  - to extract topics from corpus of documents
  - typology of customers in CRM (Customer Relationship Management)

Classification

- predictive analysis: to predict Z (categorical) from covariates $\boldsymbol{X}$ (categorical, continuous. . .)
- example: to predict the probability (score) . . .
  - marketing: . . . for a new customer to buy a product
  - medicine: . . . for a patient to be suffering from a disease
  - finance: . . . for a firm to enter bankruptcy

Different methods

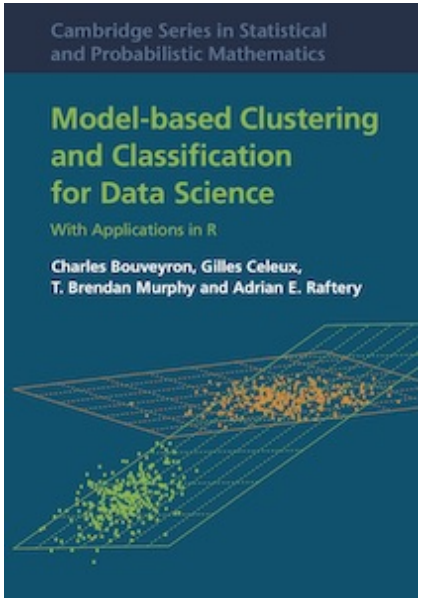
Clustering

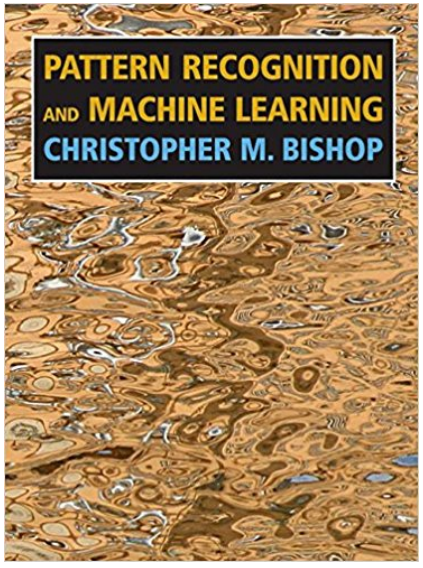
- geometric methods
  - kmeans, hierarchical clustering
- probabilistic methods
  - **mixture models**

Classification

- generative methods: estimation of $p(\boldsymbol{X}, Z)$
  - **mixture models** (linear/quadratic discriminant analysis ,. . .)
- predictive methods: estimation of $p(Z|\boldsymbol{X})$
  - logistic regression, K- nearest neighbors, classification tree, SVM, neural networks. . .

References





- G. Celeux & G. Govaert (1995), *Gaussian parsimonious clustering models*, Pattern Recognition, 28(5), 781–793.
- L. Scrucca, M. Fop, T. B. Murphy & A.E. Raftery (2016), *mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models*, The R Journal, 8(1), 289–317.

The mixture model

Definition and notation

Idea: each group is described by its *own* probability distribution

$$\boldsymbol{X}|Z = k \sim f(\boldsymbol{x}, \theta_k) = f_k(\boldsymbol{x})$$

for instance, $f(\cdot, \theta_k)$ can be

- Continuous features: the Gaussian distrib. ($\theta_k = (\mu_k, \Sigma_k)$), the Student distribution for more heavy tails...
- Binary features: multivariate Bernoulli distrib., $\theta_k = (\alpha_{kj})_{1 \leq j \leq p}$
- Categorical features: multinomial distribution...

Definition and notation

mixing proportion

$$Z = k \Leftrightarrow \widetilde{Z} = (0, \ldots, 0, \underbrace{1}_{k-\text{th position}}, 0, \ldots, 0)$$

$$\widetilde{Z} \sim \mathcal{M}(1, p_1, \ldots, p_K)$$

where $p_k = P(Z = k) = P(\widetilde{Z}_k = 1)$ is the mixing proportion of group k

The mixture model

- marginal distribution of $\boldsymbol{X}$ (mixture density)

$$\boldsymbol{X} \sim \sum_{k=1}^{K} p_k f_k(\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}).$$

The mixture model

- marginal distribution of $\boldsymbol{X}$ (mixture density)

$$\boldsymbol{X} \sim \sum_{k=1}^{K} p_k f_k(\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}).$$

proof:

$$\begin{aligned} P(\boldsymbol{X} \in I) &= P(\boldsymbol{X} \in I \cap Z \in \{1, \ldots, K\}) \\ &= \sum_{k=1}^{K} P(\boldsymbol{X} \in I \cap Z = k) \\ &= \sum_{k=1}^{K} P(\boldsymbol{X} \in I | Z = k) P(Z = k) \end{aligned}$$

The mixture model

- conditional probability that $\boldsymbol{x}$ belongs to group k (via Bayes theorem):

$$P(Z = k|\boldsymbol{X} = \boldsymbol{x}) = \frac{p_k f_k(\boldsymbol{x})}{f_{\boldsymbol{X}}(\boldsymbol{x})} = t_k(\boldsymbol{x}).$$

The mixture model

- conditional probability that $\boldsymbol{x}$ belongs to group k (via Bayes theorem):

$$P(Z = k|\boldsymbol{X} = \boldsymbol{x}) = \frac{p_k f_k(\boldsymbol{x})}{f_{\boldsymbol{X}}(\boldsymbol{x})} = t_k(\boldsymbol{x}).$$

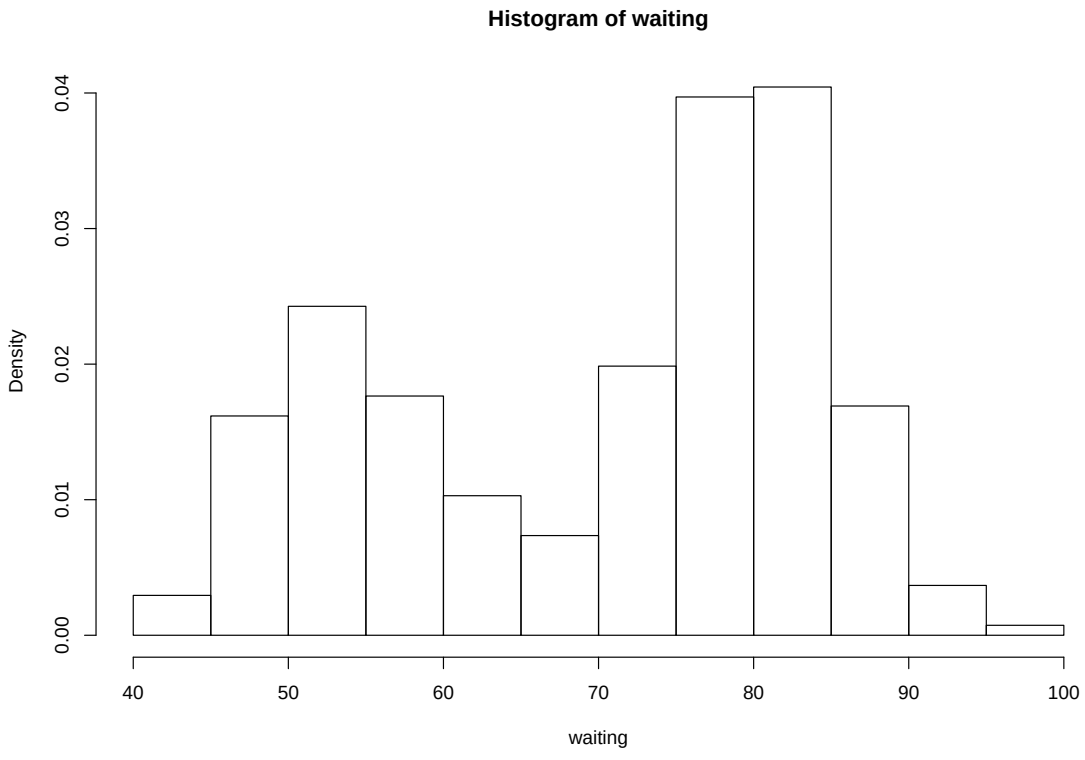
proof:

$$\begin{aligned} P(Z = k|\boldsymbol{X} = \boldsymbol{x}) &= \frac{f(\boldsymbol{x}|Z = k)P(Z = k)}{f(\boldsymbol{x})} \\ &= \frac{f(\boldsymbol{x}|Z = k)P(Z = k)}{\sum_{\ell=1}^{K} f(\boldsymbol{x}|Z = \ell)P(Z = \ell)} \end{aligned}$$

Example - Faithful

Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

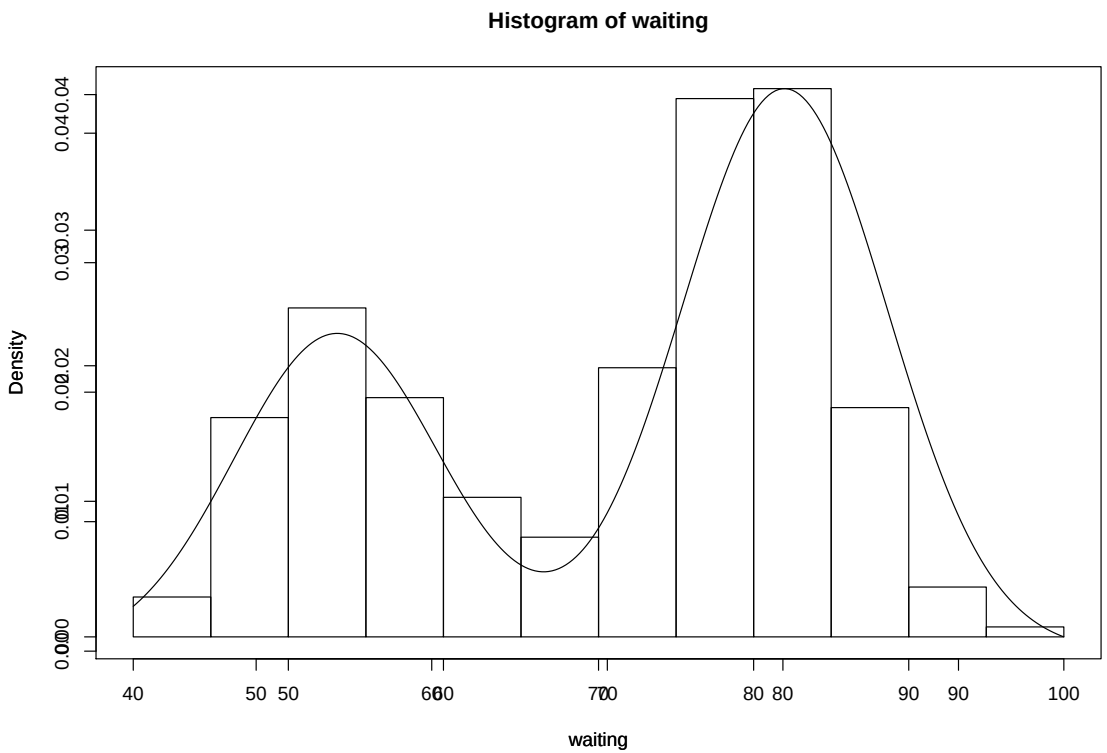
```
attach(faithful)
hist(waiting,prob=T)
```



Example - Faithful

Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

```
library(mclust)
res=Mclust(waiting)
hist(waiting,prob=T);par(new=TRUE)
plot(res,'density')
```



Clustering and classification rule

Clustering and classification rule

Let assume that all the mixture model parameters (p_k and the parameters of f_k) are known (*they will be estimated in practice from data*)

Clustering: each observation $\boldsymbol{x_i}$ is classified into the group k maximizing the conditional probability $t_k(\boldsymbol{x_i}) = P(Z = k|\boldsymbol{X} = \boldsymbol{x_i})$:

$$Z = argmax_k \, t_k(\boldsymbol{x_i})$$

Classification: it will depend of the cost of misclassification (not necessary symetric)

Classification rule

To define a classification rule

$$r : \boldsymbol{x} \in \mathbb{E} \rightarrow r(\boldsymbol{x}) \in \{1, \ldots, K\}.$$

is equivalent to divide $\mathbb{E}$ into K subsets Ω_k s.t.

$$\begin{aligned} \Omega_1 \cup \ldots \cup \Omega_K &= \mathbb{E}, \\ \Omega_k \cap \Omega_\ell &= \emptyset \\ \text{and} \quad \boldsymbol{x} \in \Omega_k \Leftrightarrow r(\boldsymbol{x}) &= k. \end{aligned}$$

Probability of misclassification

Probability of classifying an observation of group G_k into G_ℓ ($\ell \neq k$) with r:

$$e_{k\ell}(r) = P(r(\boldsymbol{X}) = \ell | Z = k) = \int_{\Omega_\ell} f_k(\boldsymbol{x}) d\boldsymbol{x}.$$

Probability of misclassification of an observation of G_k with r:

$$e_k(r) = P(r(\boldsymbol{X}) \neq k | Z = k) = \sum_{\ell \neq k} e_{k\ell}(r) = \int_{\mathbb{E} \setminus \Omega_k} f_k(\boldsymbol{x}) d\boldsymbol{x}.$$

Global probability of misclassification (global misclassification error):

$$e(r) = \sum_{k=1}^{K} p_k e_k(r).$$

Misclassification cost

Cost of misclassifying an observation of G_ℓ in G_k:

$$C : (k, \ell) \in \{1, \ldots, K\} \times \{1, \ldots, K\} \to C(k, \ell) \in \mathbb{R}^+,$$

with $C(k, k) = 0$.

Remarks:

- $C(k, \ell)$ generally not symmetric
- to be defined with practician (or fixed to 1 if you have no information)

Examples of misclassification cost

Medecine:

- a test allows to detect if a patient is ill (G_1) or not (G_2)
- $C(1,2)$: cost of classifying a *healthy* patient (G_2) as *ill* (G_1) $\Rightarrow$ cost of carrying out more exam on a healthy patient
- $C(2,1)$: cost of classifying a *ill* patient as a *healthy* one $\Rightarrow$ cost that an ill patient go back home without treatment

Finance:

- a consumer finance company predicts if the customer will have (G_1) problem in repaying loan or not (G_2)
- $C(1,2)$: cost of classifying a *good* customer (G_2) as a *bad* one (G_1)
- $C(2,1)$: cost of classifying a *bad* customer as a *good* one
- the company probably has to choose $C(2,1) >> C(1,2)$

Bayes optimal classification rule

Conditional risk associated to $\boldsymbol{x}$: average cost of misclassification of $\boldsymbol{x}$

$$R(r(\boldsymbol{X})|\boldsymbol{X}=\boldsymbol{x}) = E[C(r(\boldsymbol{X}), Z)|\boldsymbol{X}=\boldsymbol{x}] = \sum_{k=1}^{K} C(r(\boldsymbol{x}), k) t_k(\boldsymbol{x}),$$

Average risk

$$R(r) = E_{\boldsymbol{X}}[R(r(\boldsymbol{X})|\boldsymbol{X}=\boldsymbol{x})] = \sum_{k=1}^{K} p_k \sum_{\ell=1}^{K} C(\ell, k) \int_{\Omega_\ell} f_k(\boldsymbol{x}) d\boldsymbol{x}.$$

Proofs: exercice.

Bayes optimal classification rule

We look for the optimal rule r^* which minimize the average risk, which is equivalent to minimize the conditional risk since:

$$R(r^*) = \min_r E_{\boldsymbol{X}}[R(r(\boldsymbol{X})|\boldsymbol{X} = \boldsymbol{x})] \geq E_{\boldsymbol{X}}[\min_r R(r(\boldsymbol{X})|\boldsymbol{X} = \boldsymbol{x})].$$

The optimal rule classifies $\boldsymbol{x}$ into G_k if

$$R(r(\boldsymbol{X}) = k|\boldsymbol{X} = \boldsymbol{x}) < R(r(\boldsymbol{X}) = \ell|\boldsymbol{X} = \boldsymbol{x}) \qquad \forall \ell \neq k.$$

Since

$$\begin{aligned} R(r(\boldsymbol{X}) = k|\boldsymbol{X} = \boldsymbol{x}) &= E[C(k, Z)|\boldsymbol{X} = \boldsymbol{x}] \\ &= \sum_{\ell=1}^{K} C(k, \ell) t_\ell(\boldsymbol{x}) = \sum_{\ell \neq k}^{K} C(k, \ell) t_\ell(\boldsymbol{x}), \end{aligned}$$

the **optimal Bayes classification rule** is:

$$r^*(\boldsymbol{x}) = k \qquad \text{if } \sum_{\ell \neq k}^{K} C(k, \ell) t_\ell(\boldsymbol{x}) < \sum_{\ell \neq k'}^{K} C(k', \ell) t_\ell(\boldsymbol{x}) \quad \forall k' \neq k.$$

Bayes optimal rule for equal costs

If $C(k, \ell) = c \; \forall k \neq \ell$, the conditional risk is

$$R(r(\boldsymbol{X}) = k | \boldsymbol{X} = \boldsymbol{x}) = c \sum_{\ell \neq k}^{K} t_\ell(\boldsymbol{x}) = c(1 - t_k(\boldsymbol{x})),$$

and thus $r^*(\boldsymbol{x}) = k$ if $c(1 - t_k(\boldsymbol{x})) < c(1 - t_{k'}(\boldsymbol{x})) \quad \forall k' \neq k$ or equivalently

$$r^*(\boldsymbol{x}) = k \qquad \text{if } t_k(\boldsymbol{x}) > t_{k'}(\boldsymbol{x}) \quad \forall k' \neq k.$$

$\Rightarrow \boldsymbol{x}$ is classified into the group which has the greater posterior probability **maximum a posteriori**.

Bayes optimal rule for equal costs

If moreover $c = 1$, the average risk is

$$\begin{aligned} R(r) &= \sum_{k=1}^{K} p_k \sum_{\ell \neq k} \int_{\Omega_\ell} f_k(\boldsymbol{x}) d\boldsymbol{x} \\ &= \sum_{k=1}^{K} p_k \int_{\bar{\Omega}_\ell} f_k(\boldsymbol{x}) d\boldsymbol{x} \\ &= \sum_{k=1}^{K} p_k e_k(r) \\ &= e(r) \end{aligned}$$

Bayes optimal rule for 2 groups

For 2 groups, we have

$$\begin{aligned} & r^*(\boldsymbol{x}) = 1 \quad \text{if} \quad C(1,2)t_2(\boldsymbol{x}) < C(2,1)t_1(\boldsymbol{x}), \\ \text{and} \quad & r^*(\boldsymbol{x}) = 2 \quad \text{if} \quad C(2,1)t_1(\boldsymbol{x}) < C(1,2)t_2(\boldsymbol{x}), \end{aligned}$$

and by noting $g(\boldsymbol{x}) = \frac{C(2,1)t_1(\boldsymbol{x})}{C(1,2)t_2(\boldsymbol{x})}$, the Bayes optimal rule is:

$$\begin{aligned} & r^*(\boldsymbol{x}) = 1 \quad \text{if} \quad g(\boldsymbol{x}) > 1, \\ \text{and} \quad & r^*(\boldsymbol{x}) = 2 \quad \text{if} \quad g(\boldsymbol{x}) < 1. \end{aligned}$$

$g(\boldsymbol{x}) = 1$ is the equation of the **separating surface**.

Continuous features: the Gaussian mixture

The Gaussian Mixture Model

The density of group k is

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

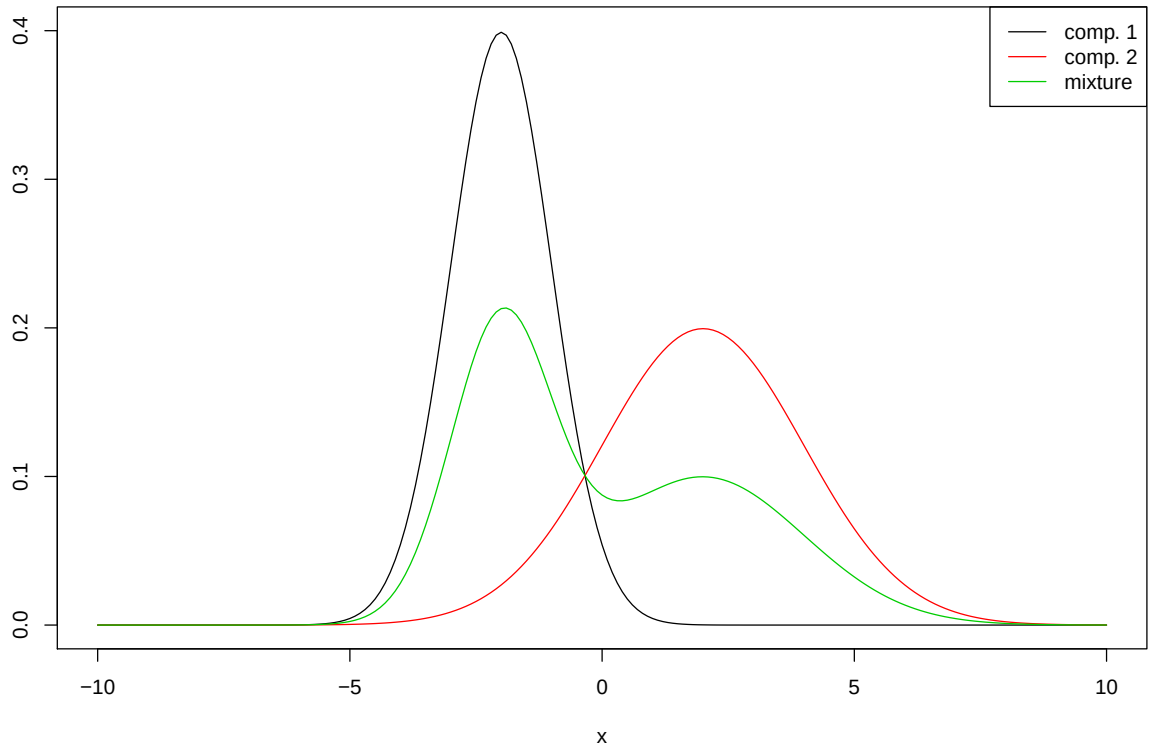
where

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

The Gaussian Mixture Model

An example of Gaussian mixture

```
x=seq(-10,10,.1)
plot(x,dnorm(x,-2,1),type='l',col=1,ylab='',ylim=c(0,0.4))
lines(x,dnorm(x,2,2),col=2)
lines(x,.5*dnorm(x,-2,1)+.5*dnorm(x,2,2),col=3)
legend('topright',legend=c('comp. 1','comp. 2','mixture'),
       col=1:3,lty=1)
```



Mixture model estimation in classification

Maximum likelihood estimation

- Estimation of the classification rule r^* is obtained by estimating θ by **maximum likelihood**
- likelihood in the classification context ($\underline{x}, \underline{z}$ available):

$$\begin{aligned} p(\underline{x}, \underline{z}) &= \prod_i p(x_i, z_i) \\ &= \prod_i p(Z = z_i) f(x_i | Z = z_i) \\ &= \prod_i \prod_k (p(Z = k) f(x_i | Z = k))^{\tilde{z}_{ik}} \\ &= \prod_i \prod_k p_k^{\tilde{z}_{ik}} f_k(x_i)^{\tilde{z}_{ik}} \end{aligned}$$

- Log-likelihood

$$\ell(\underline{x}, \underline{z}; \theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} \tilde{z}_{ik} \left(\ln p_k - \frac{p}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (x_i - \mu_k)^t \Sigma_k^{-1} (x_i - \mu_k) \right)$$

Parameter estimation

Maximization leads to the usual empirical estimates:

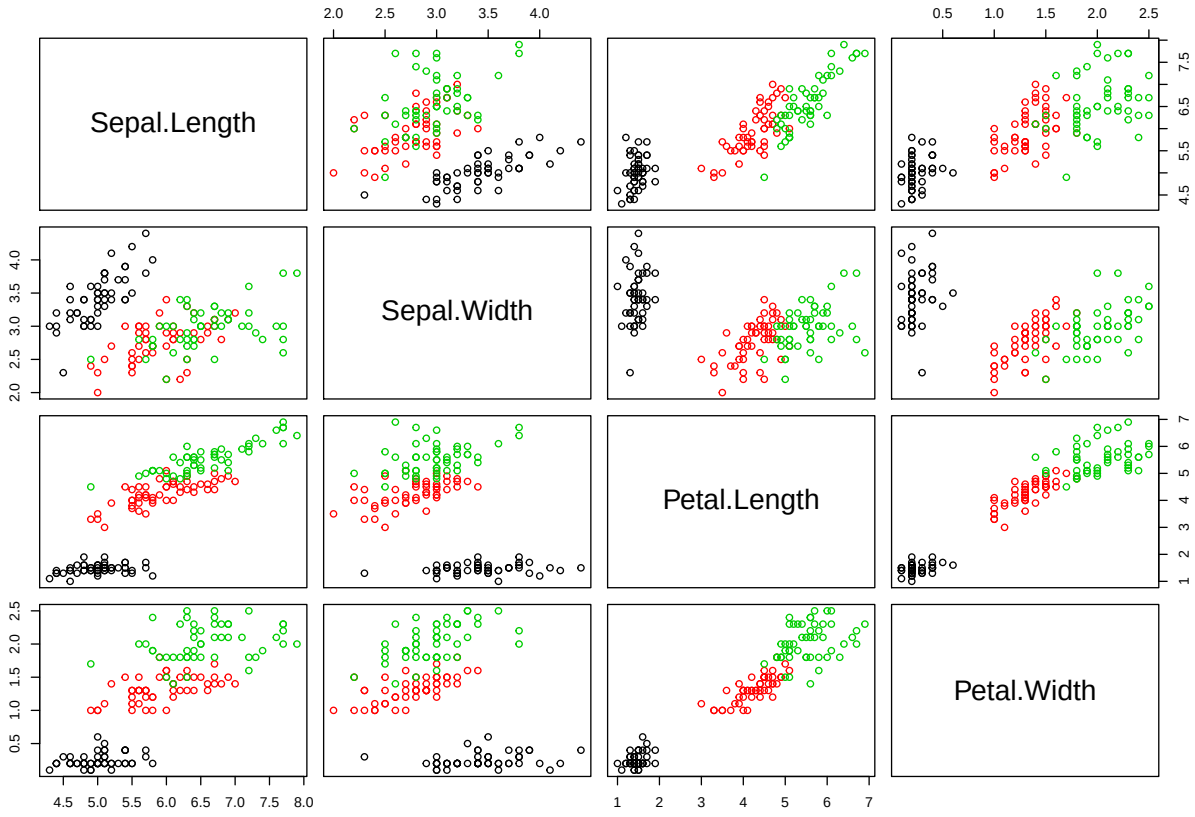
- $\hat{p}_k = \frac{n_k}{n}$ with $n_k = \sum_{i=1}^{n} \tilde{z}_{ik}$ the number of observations of group k
- $\hat{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{n} \tilde{z}_{ik} \boldsymbol{x}_i$
- $\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n} \tilde{z}_{ik} (\boldsymbol{x}_i - \hat{\mu}_k)^t (\boldsymbol{x}_i - \hat{\mu}_k)$

Exercice 1

Prove the expression of the previous estimators in the case of $p = 1$.
Use Lagrange multipliers for the constraint $\sum_{k=1}^{K} p_k = 1$.

Application on iris dataset

```
plot(iris[,1:4],col=iris$Species)
```



Application on iris dataset

Split into train and test data sets

```
set.seed(2)
tmp=sample(1:150,50)
X.train <- iris[-tmp,-5]
Class.train <- iris[-tmp,5]
X.test <- iris[tmp,-5]
Class.test <- iris[tmp,5]
```

Application on iris dataset

Estimation of a mixture of Gaussians (*model 'XXX' is the usual Gaussian model in MclustDA*)

```
irisMclustDA <- MclustDA(X.train, Class.train,
                         modelNames = "XXX")
summary(irisMclustDA)
```

```
## ------------------------------------------------
## Gaussian finite mixture model for classification
## ------------------------------------------------
##
## MclustDA model summary:
##
##  log-likelihood   n df       BIC
##       -113.9301 100 42 -421.2774
##
## Classes       n  % Model G
##   setosa     35 35   XXX 1
##   versicolor 35 35   XXX 1
##   virginica  30 30   XXX 1
##
```

Application on iris dataset

Evaluation of the prediction

```
tmp=summary(irisMclustDA, newdata = X.test,
            newclass = Class.test)
print(tmp$tab.newdata)
```

```
##              Predicted
## Class         setosa versicolor virginica
##   setosa          15          0         0
##   versicolor       0         14         1
##   virginica        0          0        20
```

Mixture model in classification

In comparison with other classification methods:

- MM has the advantage of interpretability
- but the classification power suffer from its assumption (each class should be Gaussian)

More flexilibility can be introduce by considering mixture of mixture:

- each class can be itself a mixture

Mixture of mixture on iris dataset

Estimation of a mixture of Gaussians with selection by BIC of the number of mixture components per class (*option modelType = "MclustDA"*)

```
irisMclustDA <- MclustDA(X.train, Class.train,
                         modelType = "MclustDA")
summary(irisMclustDA)
```

```
## ------------------------------------------------
## Gaussian finite mixture model for classification
## ------------------------------------------------
##
## MclustDA model summary:
##
##  log-likelihood   n df       BIC
##       -70.80528 100 63 -431.7363
##
## Classes       n  % Model G
##   setosa     35 35   VEV 2
##   versicolor 35 35   XXX 1
##   virginica  30 30   VVE 2
##
```

Application on iris dataset

Evaluation of the prediction (not necessary better for the simple iris data set)

```
tmp=summary(irisMclustDA, newdata = X.test,
            newclass = Class.test)
print(tmp$tab.newdata)
```

```
##              Predicted
## Class         setosa versicolor virginica
##   setosa          15          0         0
##   versicolor       0         14         1
##   virginica        0          0        20
```

Exercice 2

Implement your own maximum likelihood estimation for a Gaussian Mixture Model. Your function should be able to predict the class of a new observation.

Test it on simulated data.

Mixture model estimation in clustering

Maximum likelihood estimation

- Mixture model estimation for clustering is done by **maximum likelihood**
- likelihood in the clustering context (only $\underline{\boldsymbol{x}}$ available):

$$\begin{aligned} p(\underline{\boldsymbol{x}}) &= \prod_i p(\boldsymbol{x}_i) \\ &= \prod_i \sum_k p_k f_k(\boldsymbol{x}_i) \end{aligned}$$

- Log-likelihood

$$\ell(\underline{\boldsymbol{x}}; \boldsymbol{\theta}) = \sum_{i=1}^{n} \ln \left(\sum_{k=1}^{K} p_k \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \mu_k)^t \Sigma_k^{-1} (\boldsymbol{x}_i - \mu_k)\} \right)$$

Not as easy to maximise as in the classification context !

The EM algorithm

The idea of Expectation-Maximization algorithm:

- to maximize the **completed**-likelihood (by the **unobserved data** $\underline{z}$) is easier than the observed-data likelihood:

$$\ell_c(\underline{x}, \underline{z}, \theta) = \sum_{k=1}^{K} \sum_{i=1}^{n} \tilde{z}_{ik} \left(\ln p_k - \frac{p}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (x_i - \mu_k)^t \Sigma_k^{-1} (x_i - \mu_k) \right)$$

- since $\underline{z}$ is *unobserved*, the q-th iteration of the EM algorithm consists in alternating:
  - initialization: randomly choose $\theta^{(0)}$
  - at iteration (q):
    - E step: computation of

$$Q(\theta, \theta^{(q)}) = E[\ell_c(\underline{x}, \underline{z}, \theta) | \underline{x}, \theta^{(q)}]$$

    - M step: maximisation of $Q(\theta, \theta^{(q)})$ according to θ:

$$\theta^{(q+1)} = argmax_\theta Q(\theta, \theta^{(q)})$$

until convergence of the log-likelihood: $|\ell(\underline{x}, \theta^{(q+1)}) - \ell(\underline{x}, \theta^{(q)})| < \epsilon$

EM algorithm - E step

Computation of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = E[\ell_c(\underline{\boldsymbol{x}}, \underline{\boldsymbol{z}}, \boldsymbol{\theta})|\underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(q)}]$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} E[\tilde{z}_{ik}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(q)}] \left(\ln p_k - \frac{p}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (\boldsymbol{x}_i - \mu_k)^t \Sigma_k^{-1} (\boldsymbol{x}_i - \mu_k) \right)$$

with

$$\begin{aligned}
E[\tilde{z}_{ik}|\underline{\boldsymbol{x}}, \boldsymbol{\theta}^{(q)}] &= 1 \times P(\tilde{z}_{ik} = 1|\boldsymbol{x}_i, \boldsymbol{\theta}^{(q)}) + 0 \times P(\tilde{z}_{ik} = 0|\boldsymbol{x}_i, \boldsymbol{\theta}^{(q)}) \\
&= \frac{f_{|\tilde{z}_{ik}=1}(\boldsymbol{x}_i, \boldsymbol{\theta}^{(q)}) P(\tilde{z}_{ik} = 1|\boldsymbol{\theta}^{(q)})}{f(\boldsymbol{x}_i, \boldsymbol{\theta}^{(q)})} \\
&= \frac{f_k(\boldsymbol{x}_i, \boldsymbol{\theta}^{(q)}) p_k^{(q)}}{f_{\boldsymbol{X}}(\boldsymbol{x}_i, \boldsymbol{\theta}^{(q)})} \\
&= t_k^{(q)}(\boldsymbol{x}_i)
\end{aligned}$$

EM algorithm - M step

Maximisation of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$ according to $\boldsymbol{\theta}$:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) = \sum_{k=1}^{K} \sum_{i=1}^{n} t_k^{(q)}(\boldsymbol{x}_i) \left(\ln p_k - \frac{p}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (\boldsymbol{x}_i - \mu_k)^t \Sigma_k^{-1} (\boldsymbol{x}_i - \mu_k) \right)$$

is equivalent to the log-likelihood maximization in the classification context, but by **ponderating** each observation by $t_k^{(q)}(\boldsymbol{x}_i)$

- $\hat{p}_k = \frac{n_k^{(q)}}{n}$ with $n_k = \sum_{i=1}^{n} t_k^{(q)}(\boldsymbol{x}_i)$
- $\hat{\mu}_k = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} t_k^{(q)}(\boldsymbol{x}_i) \boldsymbol{x}_i$
- $\hat{\Sigma}_k = \frac{1}{n_k^{(q)}} \sum_{i=1}^{n} t_k^{(q)}(\boldsymbol{x}_i)(\boldsymbol{x}_i - \mu_k)^t (\boldsymbol{x}_i - \mu_k)$

EM algorithm - properties

- the EM increases the likelihood at each step: $\Rightarrow$ it converges to a local maximum of the likelihood
- convergence to the global maximum is expected to be achieved with multiple initializations of the algorithm
- in practice, the most efficient initialization strategy is:
  - run several small EM (with 10 iterations)
  - run a long EM starting from the small EM solution with highest log-likelihood

EM algorithm - proof of convergence

Since

$$p(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}) p(\boldsymbol{x}; \boldsymbol{\theta})$$

taking the logarithm we have:

$$\ell_c(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) = \ln p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}) + \ell(\boldsymbol{x}; \boldsymbol{\theta})$$

and then

$$\ell(\boldsymbol{x}; \boldsymbol{\theta}) = \ell_c(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta}) - \ln p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta})$$

Let's compute $E_{\theta^{(q)}}[\cdot|\boldsymbol{x}]$ of these terms:

$$\ell(\boldsymbol{x}; \boldsymbol{\theta}) = \underbrace{E_{\theta^{(q)}}[\ell_c(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{\theta})|\boldsymbol{x}]}_{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})} - \underbrace{E_{\theta^{(q)}}[\ln p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta})]}_{H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})} \tag{1}$$

EM algorithm - proof of convergence

Let's look at $H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$:

$$\begin{aligned} H(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)}) - H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) &= E_{\theta^{(q)}}[\ln p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}^{(q)}) - \ln p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta})] \\ &= E_{\theta^{(q)}}[\ln \frac{p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}^{(q)})}{p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta})}] \\ &= \int \ln \frac{p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}^{(q)})}{p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta})} p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}^{(q)}) dz \\ &= KL(p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}^{(q)}), p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta})) \\ &\geq 0 \end{aligned}$$

Consequently, for all $\boldsymbol{\theta}$

$$H(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)}) \leq H(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)})$$

EM algorithm - proof of convergence

Since at each M step, we look for

$$\boldsymbol{\theta}^{(q+1)} = argmax_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$$

we have:

$$Q(\boldsymbol{\theta}^{(q+1)}, \boldsymbol{\theta}^{(q)}) \geq Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(q)})$$

for all $\boldsymbol{\theta}$ and in particular for $\boldsymbol{\theta} = \boldsymbol{\theta}^{(q)}$.

Consequently:

$$Q(\boldsymbol{\theta}^{(q+1)}, \boldsymbol{\theta}^{(q)}) \geq Q(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)})$$

EM algorithm - proof of convergence

Using (1) with $\boldsymbol{\theta} = \boldsymbol{\theta}^{(q+1)}$ we have

$$\ell(\boldsymbol{x}; \boldsymbol{\theta}^{(q+1)}) = \underbrace{Q(\boldsymbol{\theta}^{(q+1)}, \boldsymbol{\theta}^{(q)})}_{\geq Q(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)})} - \underbrace{H(\boldsymbol{\theta}^{(q+1)}, \boldsymbol{\theta}^{(q)})}_{\leq H(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)})}$$

and thus

$$\begin{aligned} \ell(\boldsymbol{x}; \boldsymbol{\theta}^{(q+1)}) &\geq Q(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)}) - H(\boldsymbol{\theta}^{(q)}, \boldsymbol{\theta}^{(q)}) \\ &\geq \ell(\boldsymbol{x}; \boldsymbol{\theta}^{(q)}) \end{aligned}$$

$\Rightarrow$ **After each Mstep of the EM algorithm, the likelihood increases**

Exercice 3

Implement your own EM algorithm for Gaussian mixture model estimation.

Test it on simulated data.

Compare multiple random initialization with kmeans initialization.

Some variants of the EM algorithm

The **Classification EM** algorithm is a variant of the EM algorithm, obtained by *rounding* the $t_k^{(q)}(\boldsymbol{x}_i)$:

- $t_k^{(q)}(\boldsymbol{x}_i) = 1$ for the group k s.t. $t_k^{(q)}(\boldsymbol{x}_i)$ is maximum
- $t_k^{(q)}(\boldsymbol{x}_i) = 0$ for the other groups

CEM properties:

- CEM performs **hard** classification whereas EM performs **soft** classification
- the convergence of CEM is faster than EM, but leads to a biased estimator
- nevertheless, for large samples and well separated groups, the CEM is very efficient
- CEM can be a good way to initialize an EM

Some variants of the EM algorithm

The **Stochastic EM** algorithm is a variant of the EM algorithm, obtained by *generating* the z_i according to the probabilities $t_k^{(q)}(\boldsymbol{x}_i)$.

- after a burn-in period, SEM generate a sample of $\boldsymbol{\theta}^{(q)}$ whose distribution is *around* the maximum likelihood
- final estimation can be obtained by the mean/median of this generated distribution
- SEM could be an alternative to EM for more complex model in which the $t_k^{(q)}(\boldsymbol{x}_i)$ are intractable

A particular Gaussian mixture model

Mixture models sometimes generalized well known clustering algorithm

- assume (for parcimony) that $\Sigma_k = \alpha I_p$ for every clusters
- assume equal proportions: $\pi_1 = \ldots = \pi_K$
- estimate the model with the CEM algorithm

A particular Gaussian mixture model

Mixture models sometimes generalized well known clustering algorithm

- assume (for parcimony) that $\Sigma_k = \alpha I_p$ for every clusters
- assume equal proportions: $\pi_1 = \ldots = \pi_K$
- estimate the model with the CEM algorithm

$\Rightarrow$ your are running **the k-means algorithm**

Exercice 4

Implement your own kmeans algorithm.

Test it on simulated data.