PROM International Workshop 2022

A gentle introduction to **Mixture Models** - Part 1

Prof. Salvatore Ingrassia
Department of Economics and Business
University of Catania (Italy)

salvatore.ingrassia@unict.it
http://www.dei.unict.it/docenti/salvatore.ingrassia

Cracow University of Economics
October 6, 2022

1. Mixtures of Distributions and Model-Based Clustering

Essentially, all models are wrong,
but some are useful.

George E.P. Box.

Outline

Prerequisites

- Basics of statistics: descriptive statistics, statistical inference;
- Basics of probability: (multivariate) Gaussian distribution, Bayes' theorem;
- basics of cluster analysis.

1.1 Mixtures of distributions

Agenda

- Gaussian Mixture Models
- Direct and indirect applications of mixtures

Ouverture: clustering 1/3

What is Clustering

Clustering *refers to a very broad set of techniques for finding subgroups, or clusters, in a data set. More in detail, the aim of cluster analysis is to identify groups of similar objects (countries, enterprises, households) according to selected variables (unemployment rate of men and women in different countries, deprivation indicators of households, etc.).*

When we cluster the observations of a data set, we seek to partition them into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other.
Of course, to make this concrete, we must define what it means for two or more observations to be similar or different.

Ouverture: clustering 1/3

What is Clustering

Clustering *refers to a very broad set of techniques for finding subgroups, or clusters, in a data set. More in detail, the aim of cluster analysis is to identify groups of similar objects (countries, enterprises, households) according to selected variables (unemployment rate of men and women in different countries, deprivation indicators of households, etc.).*

When we cluster the observations of a data set, we seek to partition them into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other.
Of course, to make this concrete, we must define what it means for two or more observations to be similar or different.

Ouverture: clustering 2/3

For instance, suppose that we have a set of n observations, each with p features. The n observations could correspond to some countries, and the p features could correspond to economic indicators (e.g., economic activity) collected for each country.

We may have a reason to believe that there is some heterogeneity (that is there are groups of countries with similar characteristics) among the n countries based on the selected economic indicators and we want analyze this heterogeneity, i.e. we want to characterize the different group of countries according to these indicators

The scope of this lectures

The basic approaches are hierarchical clustering and k-means clustering. There are many types of these techniques: here we approach **model-based clustering**.

Ouverture: clustering 2/3

For instance, suppose that we have a set of n observations, each with p features. The n observations could correspond to some countries, and the p features could correspond to economic indicators (e.g., economic activity) collected for each country.

We may have a reason to believe that there is some heterogeneity (that is there are groups of countries with similar characteristics) among the n countries based on the selected economic indicators and we want analyze this heterogeneity, i.e. we want to characterize the different group of countries according to these indicators

The scope of this lectures

The basic approaches are hierarchical clustering and k-means clustering. There are many types of these techniques: here we approach **model-based clustering**.

Ouverture: clustering 3/3

Three key concepts:

Clustering

Clustering *refers to grouping objects into classes according to their "similarity", see e.g.* cluster analysis.

Discrimination

Discrimination *refers to the process of deriving classification rules from samples of already classified objects.*

Classification

Classification *refers to applying the rule to new objects of unknown class.*

Ouverture: clustering 3/3

Three key concepts:

Clustering

Clustering *refers to grouping objects into classes according to their "similarity", see e.g.* cluster analysis.

Discrimination

Discrimination *refers to the process of deriving classification rules from samples of already classified objects.*

Classification

Classification *refers to applying the rule to new objects of unknown class.*

Ouverture: clustering 3/3

Three key concepts:

Clustering

Clustering *refers to grouping objects into classes according to their "similarity", see e.g.* cluster analysis*.*

Discrimination

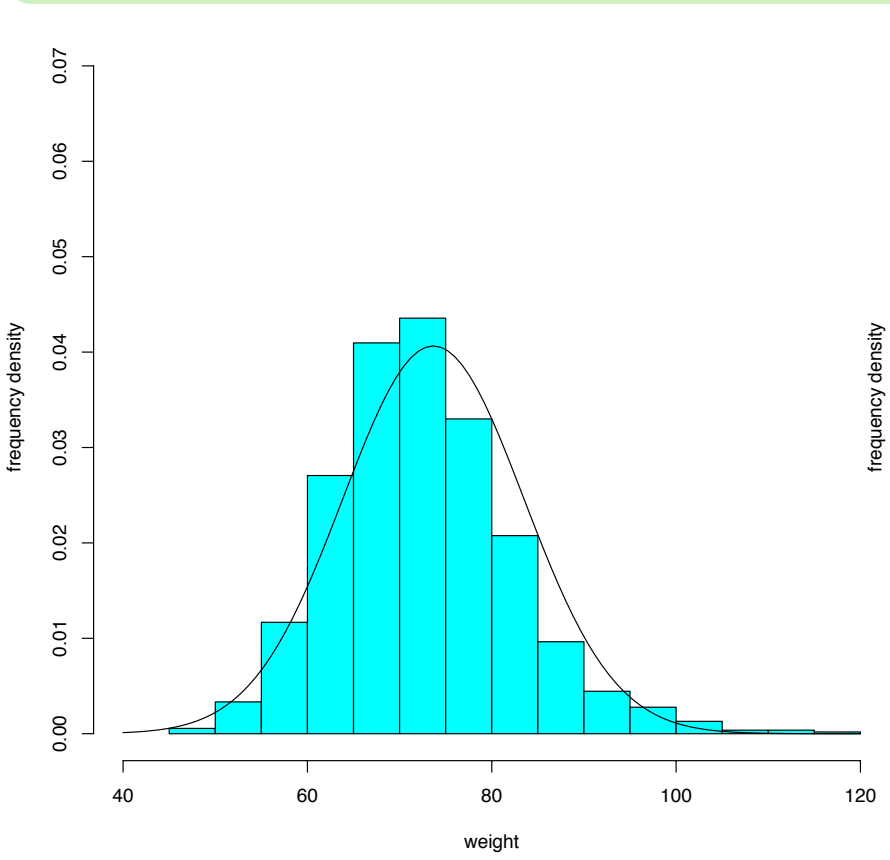
Discrimination *refers to the process of deriving classification rules from samples of already classified objects.*

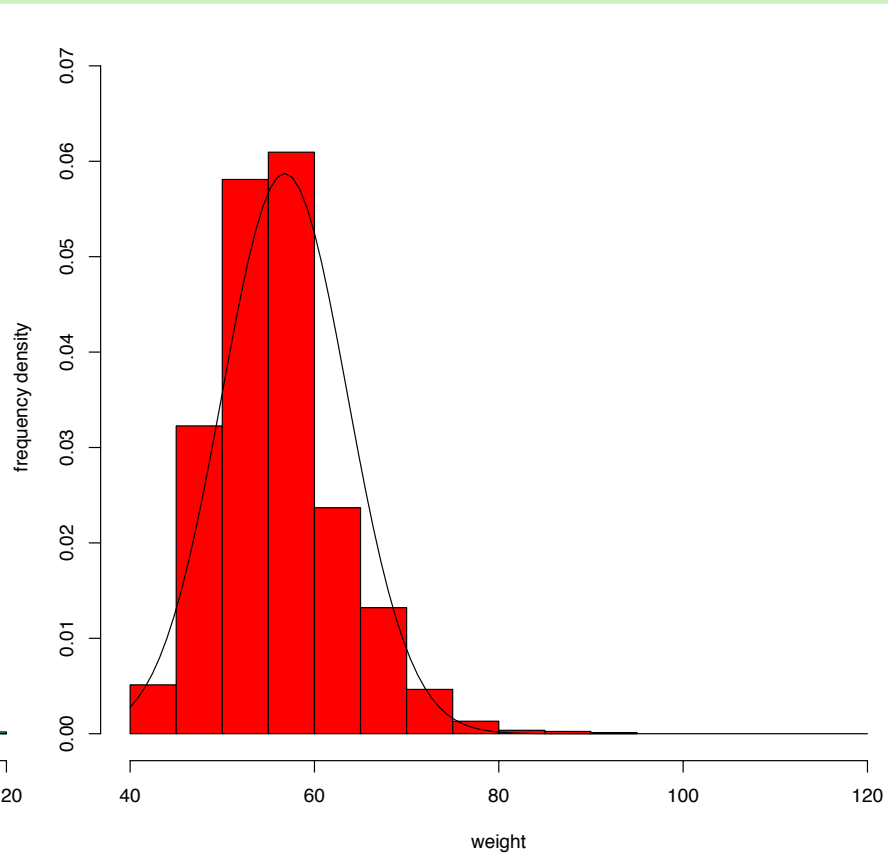
Classification

Classification *refers to applying the rule to new objects of unknown class.*

Introduction 1/4

The mixture model is essentially a tool for density estimation.

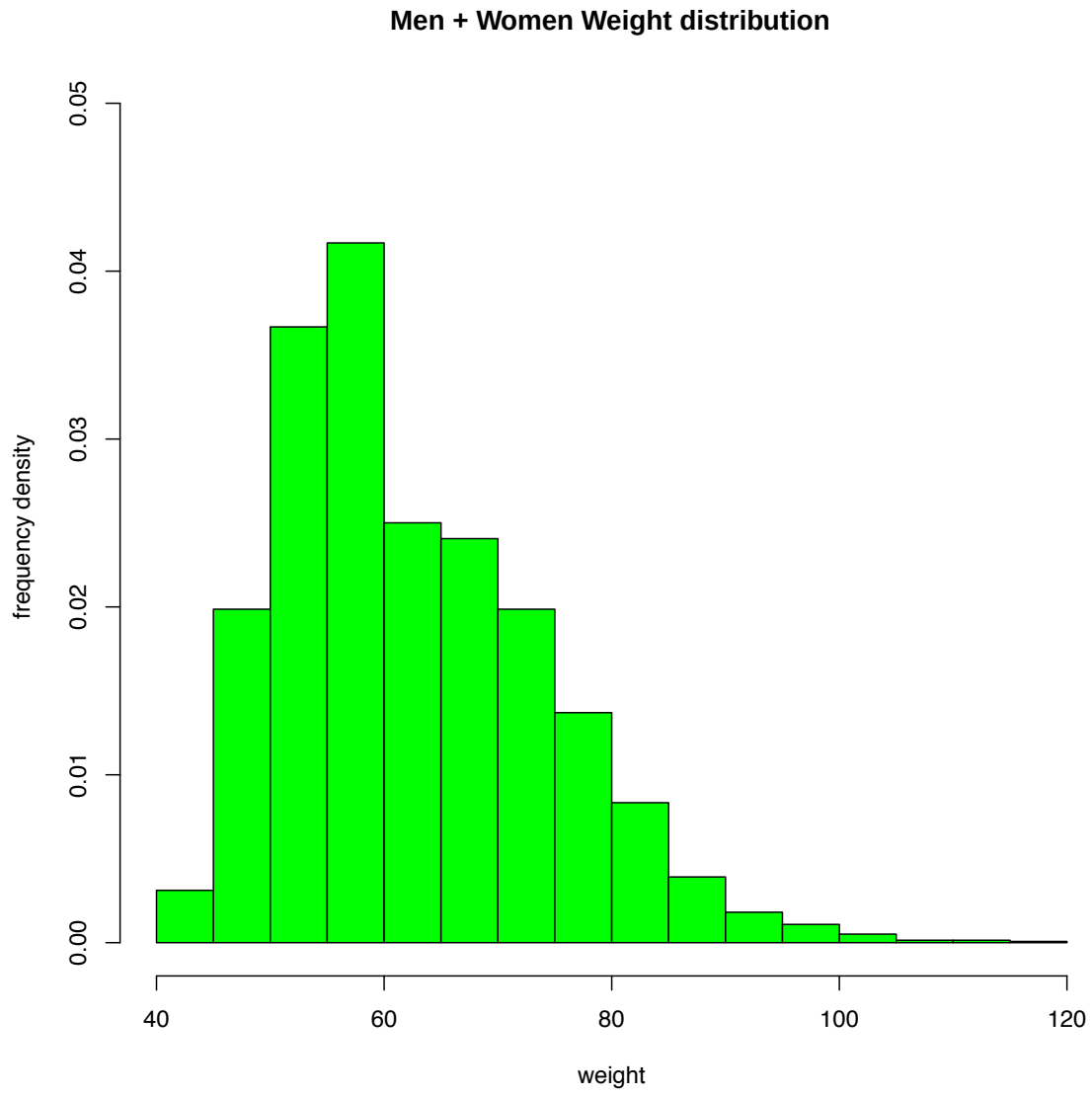




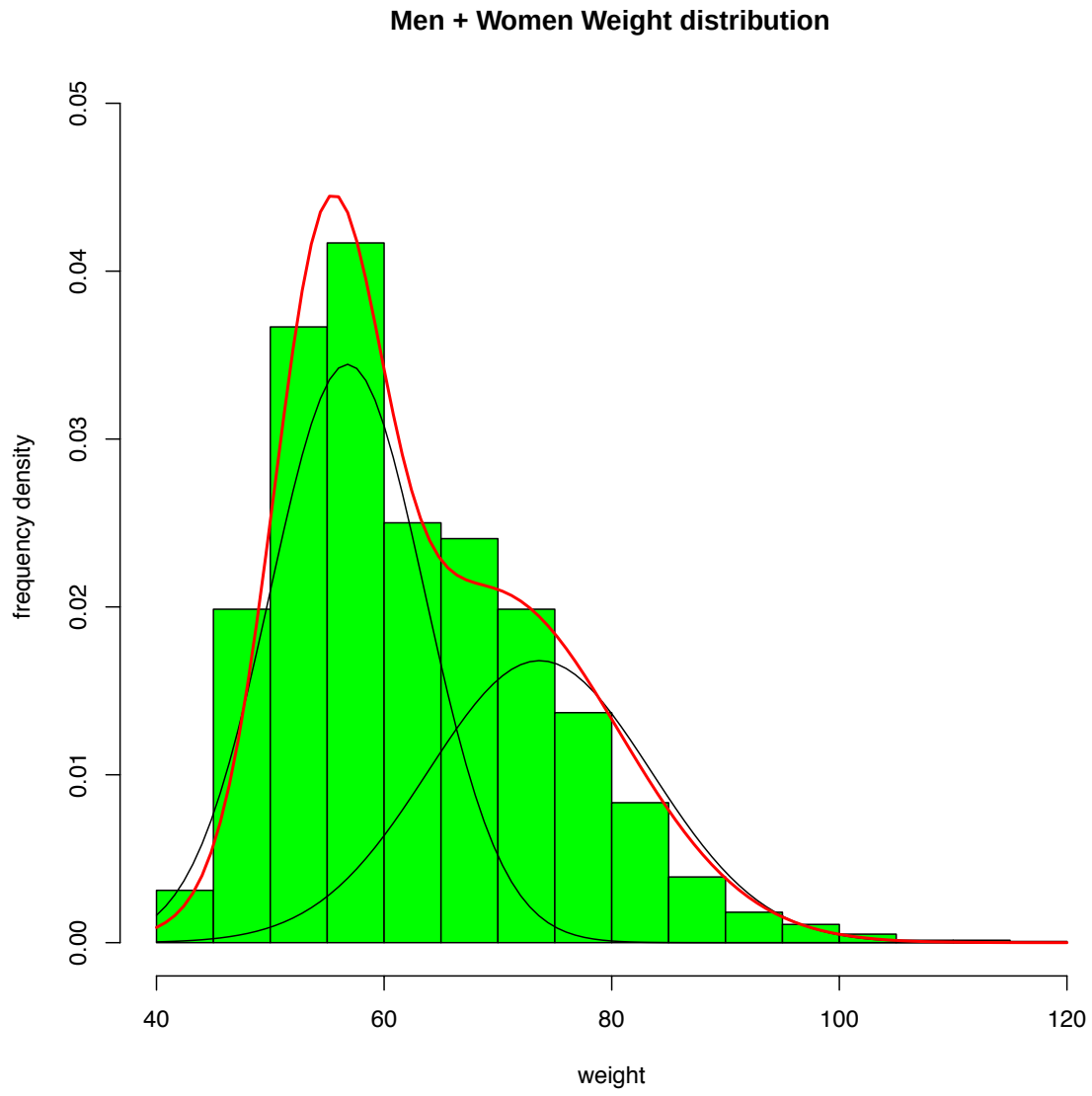
$\mu_m = 73.7$ Kg
$\sigma_m = 9.8$ Kg

$\mu_f = 56.8$ Kg
$\sigma_f = 6.8$ Kg

Introduction 2/4



Introduction 3/4



Introduction 4/4

The model:

$$p(x;\boldsymbol{\psi}) = \pi f(x;\mu_1,\sigma_1^2) + (1-\pi) f(x;\mu_2,\sigma_2^2)\,,$$

with

$$f(x;\mu_g,\sigma_g^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)\,.$$

The estimates:

$$\pi = 0.438 \quad \mu_1 = 55.05 \quad \sigma_1 = 4.87 \quad \mu_2 = 69.88 \quad \sigma_2 = 11.09\,.$$

In general:

$$p(x;\boldsymbol{\psi}) = \pi_1 f(x;\mu_1,\sigma_1^2) + \cdots + \pi_G f(x;\mu_G,\sigma_G^2)\,,$$

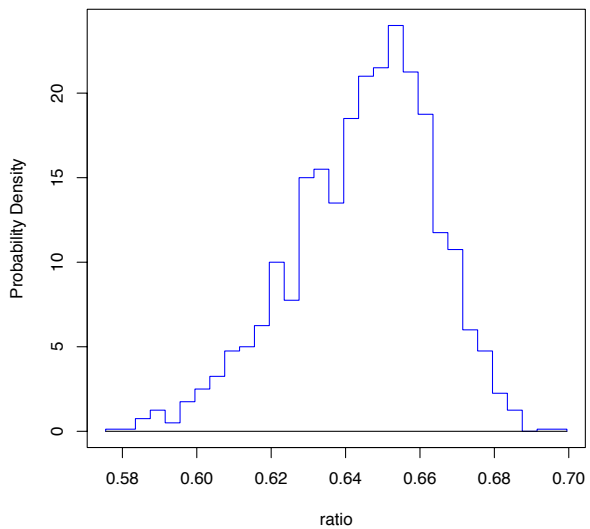
with $\pi_g > 0$ and $\pi_1 + \cdots + \pi_G = 1$.

How the story began

Mixture models have been largely considered in statistical modeling since the 19th century.

They have been proposed first in Newcomb (1886) concerning an application of normal mixtures as models for outliers.

Pearson(1894) fitted a mixture of two normal density functions on data consisted of measurements on the ratio of forehead to body length of crabs sampled from the Bay of Neaples.



Main references

- Titterington DM, Smith AFM and Makov UE (1985), *Statistical Analysis of Finite Mixture Distributions*, Wiley, Chichester.
- McLachlan GJ and Basford KE (1988), *Mixture Models: Inference and applications to clustering*, Dekker, New York.
- McLachlan GJ and Peel D.(2000), *Finite Mixture Models*, Wiley, New York.
- Frühwirth-Schnatter S (2006), *Finite Mixture and Markov Switching Models*, Springer, New York.
- McNicholas PD (2017). *Mixture Model-Based Classification*, CRC Press, Boca Raton.

Basic definitions 1/2

Finite Mixture Distribution

We say that $\mathbf{X}$ has a *finite mixture distribution* if its probability density function (pdf) has the form

$$p(\mathbf{x}) = \pi_1 f_1(\mathbf{x}) + \cdots + \pi_G f_G(\mathbf{x}) \qquad \mathbf{x} \in \mathcal{X}\,.$$

where

$$\pi_g > 0 \quad g = 1, \ldots, G, \quad \pi_1 + \cdots + \pi_G = 1$$

and

$$f_g(\cdot) \geq 0\,, \qquad \int_{\mathcal{X}} f_g(\mathbf{x}) d\mathbf{x} = 1\,, \qquad g = 1, \ldots, G\,.$$

The parameters $\pi_1, \ldots, \pi_G$ are the *mixing weights* and $f_1(\cdot), \ldots, f_G(\cdot)$ are the *component densities* of the mixture.

Basic definitions 2/2

In many situations, $f_1(\cdot), \ldots, f_G(\cdot)$ will have specified parametric forms and then we can have a more explicit representation of the pdf:

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 f_1(\mathbf{x}; \boldsymbol{\theta}_1) + \cdots + \pi_G f_G(\mathbf{x}; \boldsymbol{\theta}_G) \qquad \mathbf{x} \in \mathcal{X}$$

where $\boldsymbol{\theta}_g$ denotes the parameters of $f_g(\cdot)$ and $\boldsymbol{\psi}$ denotes the overall parameter of the mixture model. Moreover

- $\boldsymbol{\Theta}_g$ denotes the parameter space of $f_g(\cdot)$, thus $\boldsymbol{\theta}_g \in \boldsymbol{\Theta}_g$,
- $\boldsymbol{\Psi}$ denotes the parameter space of $\boldsymbol{\psi}$, thus $\boldsymbol{\psi} \in \boldsymbol{\Psi}$.

There is no requirement that the component densities should all belong to the same parametric family, but in most applications this is the case and then the finite mixture density function will have the form

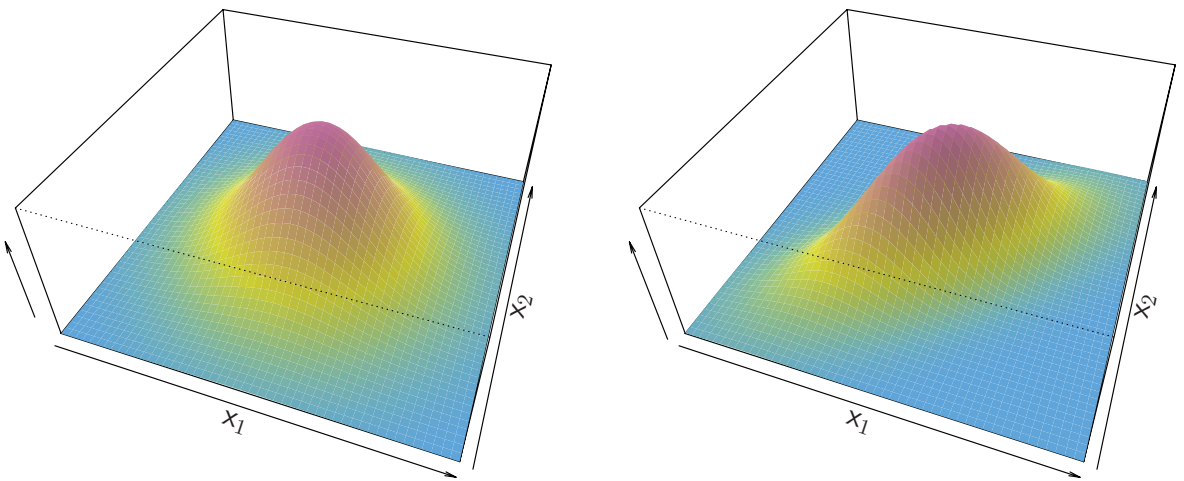
$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 f(\mathbf{x}; \boldsymbol{\theta}_1) + \cdots + \pi_G f(\mathbf{x}; \boldsymbol{\theta}_G) = \sum_{g=1}^{G} \pi_g f(\mathbf{x}; \boldsymbol{\theta}_g) \qquad \mathbf{x} \in \mathcal{X}$$

where $f(\cdot; \boldsymbol{\theta}_g)$ denotes a generic member of the parametric family and $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_G$ belong to the same parameter space $\boldsymbol{\Theta}$.

Preliminary note: Multivariate Gaussian distribution

Let $\mathbf{X} = (X_1, \ldots, X_p)$ a random vector with values in $\mathbb{R}^d$. We say that $\mathbf{X}$ has a multivariate Gaussian distribution with vector mean $\mathbb{E}(\mathbf{X}) = \boldsymbol{\mu}$ and covariance matrix $\text{Cov}(\mathbf{X}) = \boldsymbol{\Sigma}$, and write $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if it has density

$$\phi(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right).$$



Two multivariate Gaussian density functions are shown, with $p = 2$. Left: The two predictors are uncorrelated. Right: The two variables have a correlation of 0.7.

Gaussian Mixtures 1/7

Gaussian Mixture Models

An important class of mixture of distributions concerns mixtures of Gaussian distributions, which are usually referred to as **Gaussian Mixture Models (GMM)**

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 \phi(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \cdots + \pi_G \phi(\mathbf{x}; \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G) \qquad \mathbf{x} \in \mathbb{R}^d$$

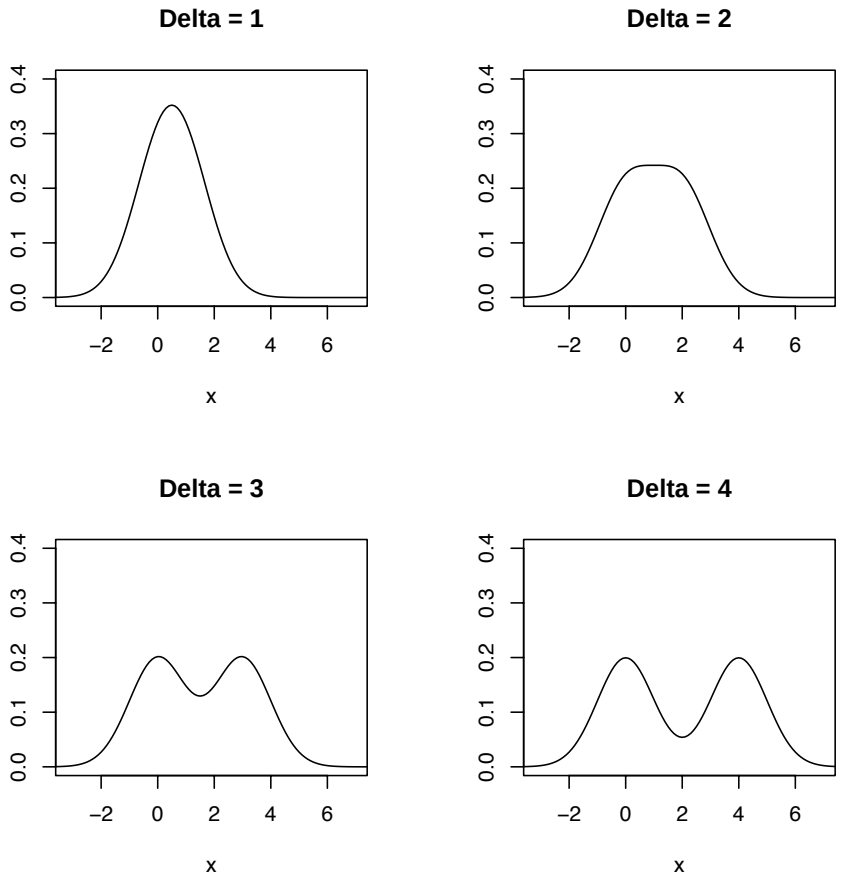
where the gth component density $\phi(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ has the form

$$\phi(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \frac{1}{|2\pi\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_g)' \boldsymbol{\Sigma}_g^{-1} (\mathbf{x} - \boldsymbol{\mu}_g)\right\} \qquad \mathbf{x} \in \mathbb{R}^d.$$

and the parameters $\pi_1, \ldots, \pi_G$ are the *mixing weights*.

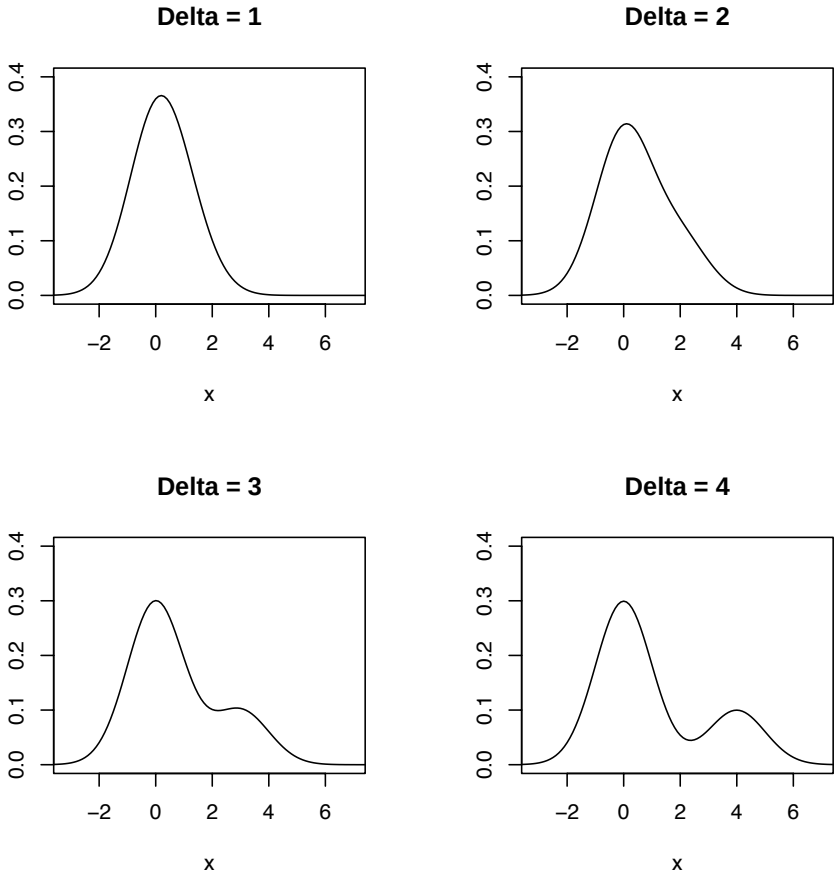
Gaussian mixtures 2/7

Gaussian mixtures can model a great variety of density functions, by selecting suitable parameters.



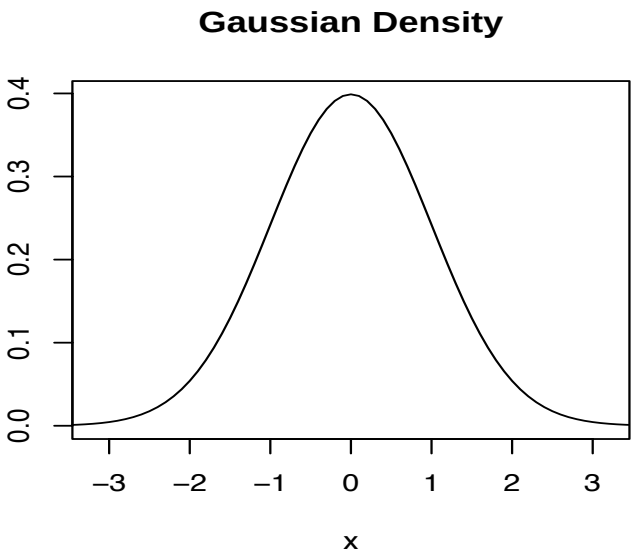
Shape of the univariate mixture $p(x;\pi,\Delta) = \pi\,\phi(x;0,1) + (1-\pi)\,\phi(x;\Delta,1)$ for $\Delta = 1, 2, 3, 4$ and weigth $\pi = 0.5$.

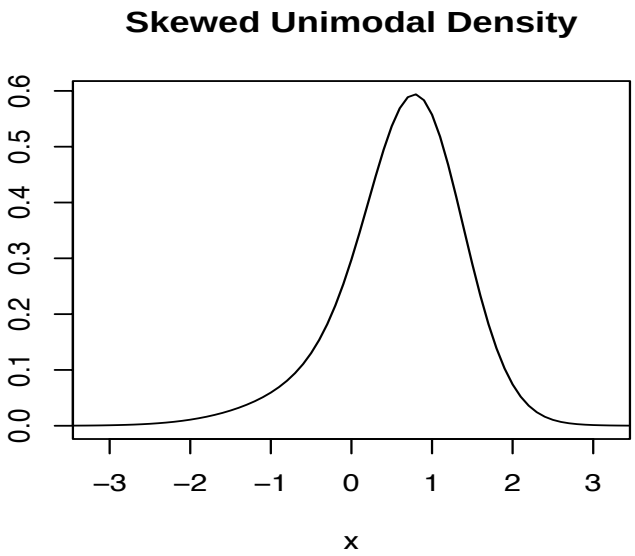
Gaussian mixtures 3/7

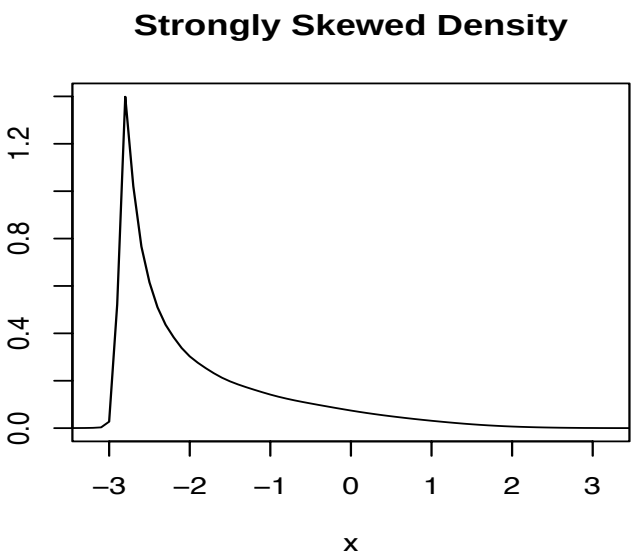


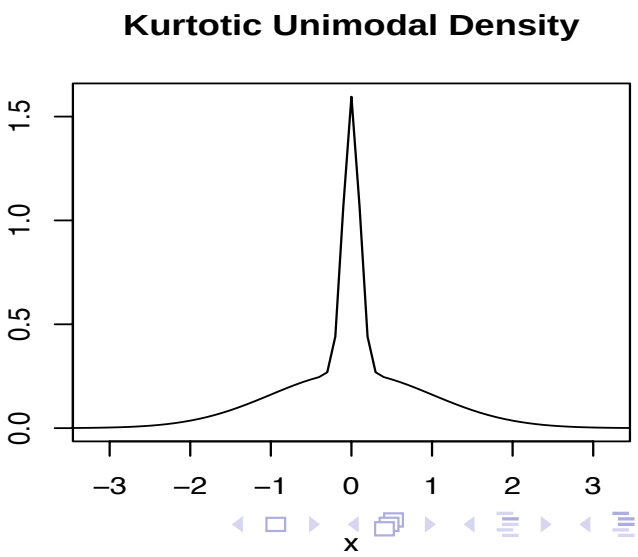
Shape of the univariate mixture
$p(x;\pi,\Delta) = \pi\,\phi(x;0,1) + (1-\pi)\,\phi(x;\Delta,1)$ for $\Delta = 1,2,3,4$ and weigth $\pi = 0.75$.

Gaussian mixtures 4/7

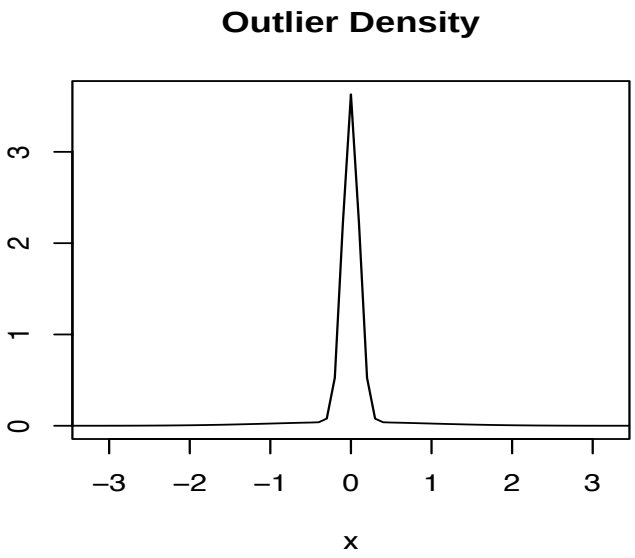


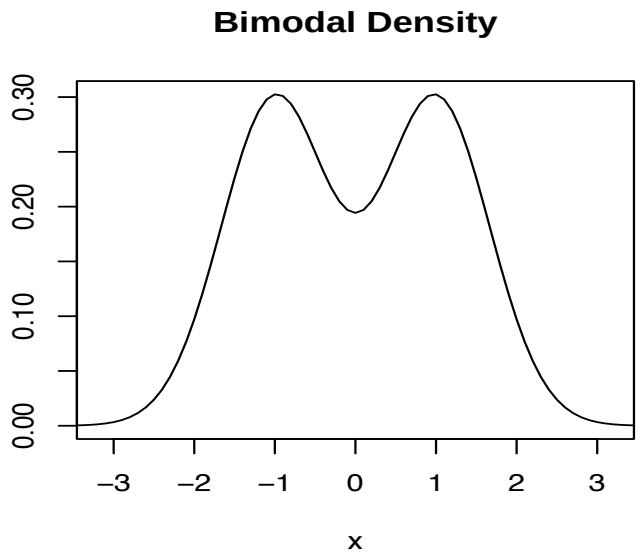


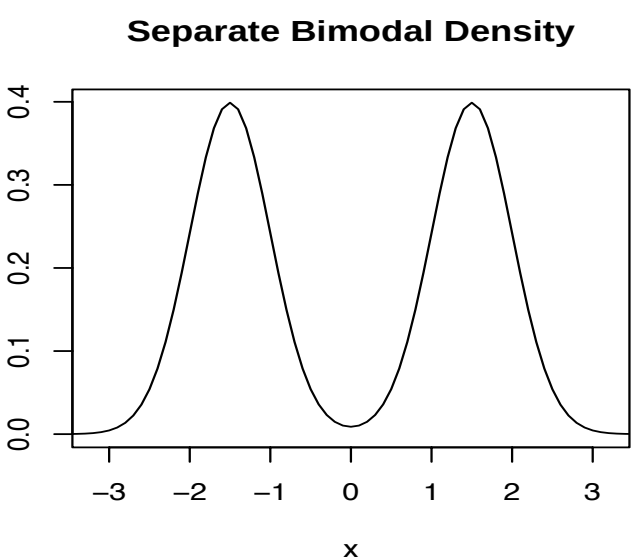


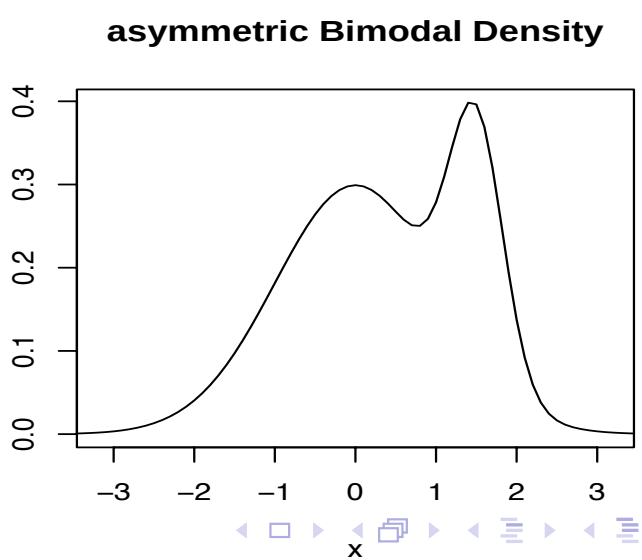


Gaussian mixtures 5/7

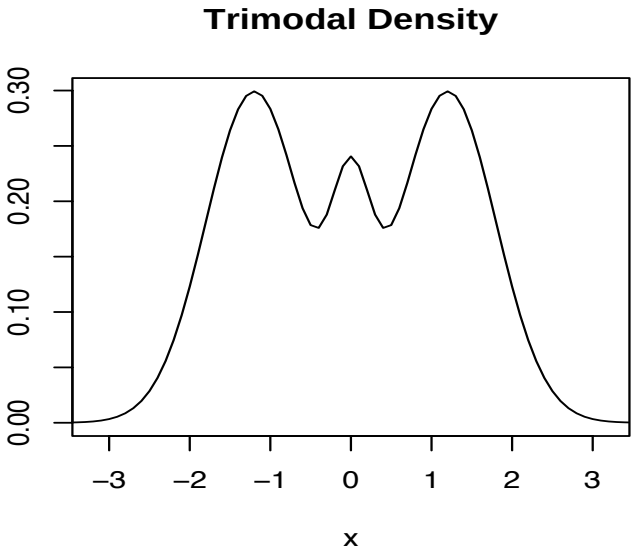


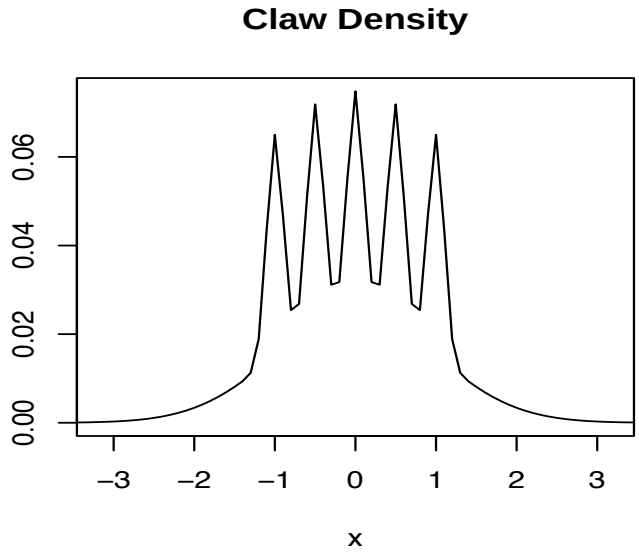


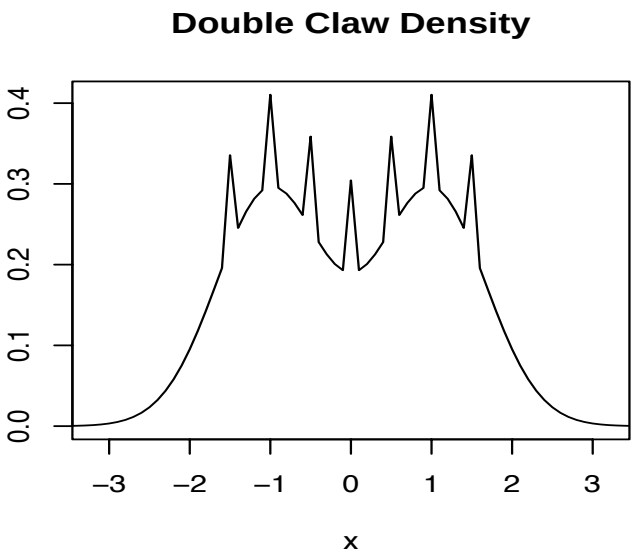


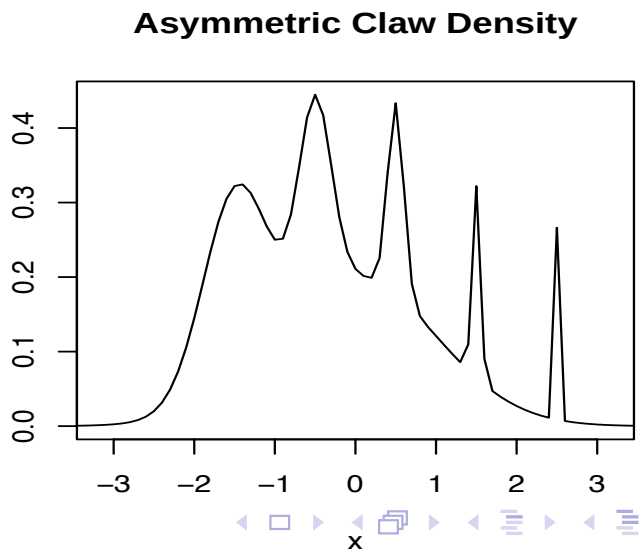


Gaussian mixtures 6/7

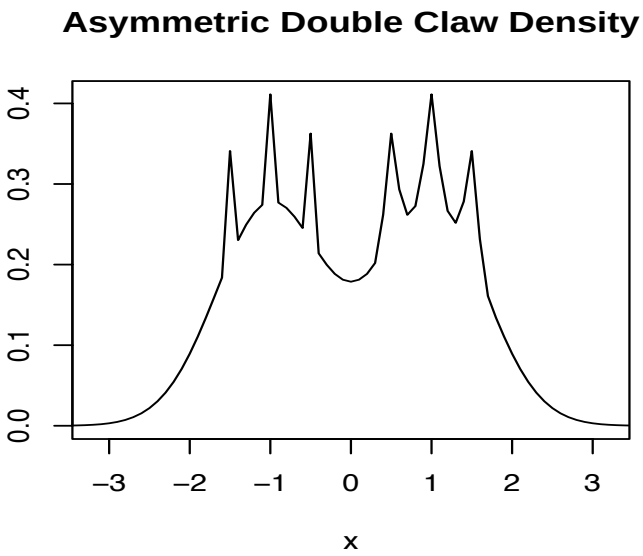


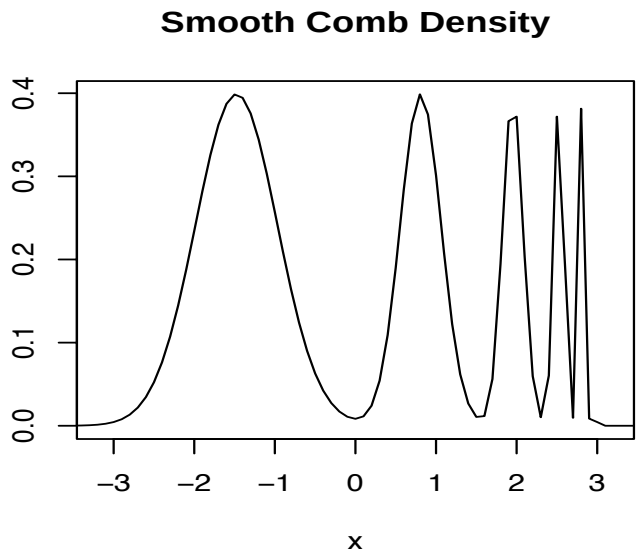


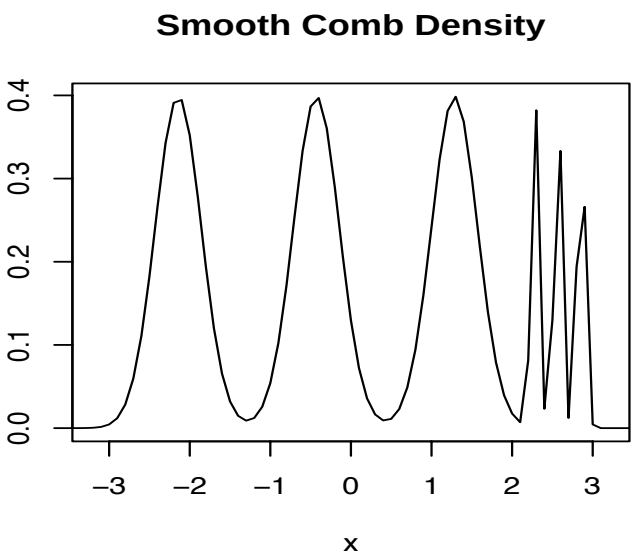




Gaussian mixtures 7/7







More in general...

Mixtures of elliptical distributions have density function of type

$$p(\mathbf{x};\boldsymbol{\psi}) = \pi_1 f(\mathbf{x};\boldsymbol{\theta}_1) + \cdots + \pi_G f(\mathbf{x};\boldsymbol{\theta}_G) \qquad \mathbf{x} \in \mathbb{R}^d$$

where $\pi_1, \ldots, \pi_G$ are the *mixing weights* and

$$f(\mathbf{x};\boldsymbol{\theta}_g) = \eta_g |\boldsymbol{\Sigma}_g|^{-1/2} h\{(\mathbf{x}-\boldsymbol{\mu}_g)' \boldsymbol{\Sigma}_g^{-1} (\mathbf{x}-\boldsymbol{\mu}_g)\}$$

- $\boldsymbol{\mu}_g \in \mathbb{R}^d$,
- $\boldsymbol{\Sigma}_g$ are positive definite matrices in $\mathbb{R}^{d^2}$ (denoted by $\boldsymbol{\Sigma}_g > 0$),
- h is a strictly positive, continuous function on $\mathbb{R}$, symmetrical about 0 and monotonicly decreasing on $[0, \infty)$,
- η_g is a positive constant depending on the dimension d of the Euclidean space.

A more general elliptical model than Gaussian mixtures concerns *mixtures of multivariate t distributions*, where the component densities depend also on a further parameter called “degrees of freedom".

Direct and indirect applications of mixtures

Titterington *et al.* (1985) distinguish two broad classes of usage of mixtures models as *direct* and *indirect applications*, although the dividing line is not always clear:

- by a *direct application*, we consider a situation where we believe in the existence of G underlying disjoint groups or subpopulations $\Omega_1, \ldots, \Omega_G$, such that $\Omega = \Omega_1 \cup \cdots \cup \Omega_G$. The observation $\mathbf{X}$ belongs to one of these subpopulations and we do not observe directly the source of $\mathbf{X}$.
- By an *indirect application*, we consider a situation where the mixture form is simply being used as a flexible and tractable mathematical tool for function approximation.

Direct applications

Assume that the population Ω can be considered as the union of G disjoint *groups* or *subpopulations* $\Omega_1, \ldots, \Omega_G$, such that $\Omega = \Omega_1 \cup \cdots \cup \Omega_G$. Each unit ω belongs to only one group. In this form of application, the model

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 f_1(\mathbf{x}; \boldsymbol{\theta}_1) + \cdots + \pi_G f_G(\mathbf{x}; \boldsymbol{\theta}_G) \qquad \mathbf{x} \in \mathbb{R}^d$$

summarizes the density function of $\mathbf{X}$ given that the observation actually comes from the subpopulation Ω_g and π_g denotes the probability that the unit ω comes from this subpopulation, that is $P(\omega \in \Omega_g) = \pi_g$.

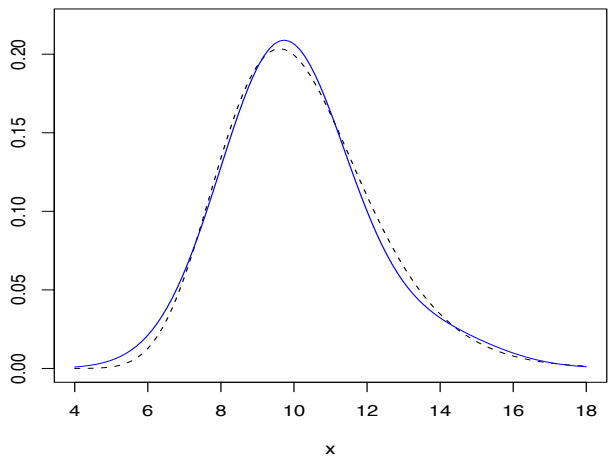
Indirect applications

Indirect applications concern cases where the underlying categories do not have necessarily a direct physical interpretation.

An important example concerns kernel-based density estimation. For example, Gaussian mixtures can approximate the lognormal distribution with density

$$\frac{1}{\sqrt{2\pi}x\sigma}\exp\left(-\frac{1}{2}(\log x-\mu)^2/\sigma^2\right),$$

with parameters $\mu = \log(10)$ e $\sigma^2 = 0.04$ and compare with a two component Gaussian mixture $p(x) = 0.9\,\phi(x; 9.7, 3) + 0.1\,\phi(x; 13.7, 3)$.



The closeness between the lognormal (dotted line) and the two-component normal mixture distributions (solid line) means that it is very difficult to discriminate between them in practice.

1.2. Maximum Likelihood Approach to Parameter Estimation and Numerical Issues

Agenda

- MLE of Gaussian mixtures
- Numerical issues
- The EM algorithm

Maximum likelihood estimate 1/2

Assume G as given (afterwards, we'll relax this assumption) .

Let $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a sample of N i.i.d. observation taken from a parametric mixture density

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 f(\mathbf{x}; \boldsymbol{\theta}_1) + \cdots + \pi_G f(\mathbf{x}; \boldsymbol{\theta}_G) \qquad \mathbf{x} \in \mathbb{R}^d.$$

The corresponding likelihood function is

$$L(\boldsymbol{\psi}) = \prod_{n=1}^{N} p(\mathbf{x}_n; \boldsymbol{\psi}) = \prod_{n=1}^{N} \left[\sum_{g=1}^{G} \pi_g f(\mathbf{x}_n; \boldsymbol{\theta}_g) \right] .$$

Maximization of $L(\boldsymbol{\psi})$ with respect to $\boldsymbol{\psi}$, for given data $\underline{\mathbf{x}}$, yields the *maximum likelihood (ML) estimate* $\widehat{\boldsymbol{\psi}}$ of $\boldsymbol{\psi}$ i.e.

$$\widehat{\boldsymbol{\psi}} = \arg \max_{\boldsymbol{\psi} \in \boldsymbol{\Psi}} L(\boldsymbol{\psi}) .$$

Maximum likelihood estimate 2/2

In practice, it is usually convenient to deal with the log-likelihood function

$$\mathcal{L}(\boldsymbol{\psi}) = \log L(\boldsymbol{\psi}) = \sum_{n=1}^{N} \log \left[\sum_{g=1}^{G} \pi_g f(\mathbf{x}_n; \boldsymbol{\theta}_g) \right]$$

which requires solving the likelihood equations

$$\frac{\partial \mathcal{L}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} = \mathbf{0}\,,$$

e.g. by means of some Newton-like method, where

$$\frac{\partial \mathcal{L}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}}$$

is the gradient vector of the log-likelihood function with respect to $\boldsymbol{\psi}$.
Thus, we get

$$\widehat{\boldsymbol{\psi}} = \arg\max_{\psi \in \boldsymbol{\Psi}} \mathcal{L}(\boldsymbol{\psi})\,.$$

MLE of Gaussian mixtures

An important case concerns mixture of Gaussian distributions

$$p(\mathbf{x};\boldsymbol{\psi}) = \pi_1\phi(\mathbf{x};\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) + \cdots + \pi_G\phi(\mathbf{x};\boldsymbol{\mu}_G,\boldsymbol{\Sigma}_G) \qquad \mathbf{x} \in \mathbb{R}^d \,.$$

and the log-likelihood function specializes as

$$\mathcal{L}(\boldsymbol{\psi}) = \sum_{n=1}^{N} \log \left[\sum_{g=1}^{G} \pi_g \frac{1}{|2\pi\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_g) \right\} \right]$$

where $\boldsymbol{\psi} = \{(\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g),\ g = 1,\ldots,G\} \in \boldsymbol{\Psi}$ and $\boldsymbol{\Psi}$ being the parameter space:

$$\boldsymbol{\Psi} = \{(\pi_1,\ldots,\pi_G,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_G,\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_G) \in \mathbb{R}^{G[1+d+(d^2+d)/2]} \ :$$
$$\pi_1 + \cdots + \pi_G = 1, \pi_g > 0,\ \boldsymbol{\Sigma}_g > 0 \quad \text{for } g = 1,...,G\} \,,$$

with $\boldsymbol{\Sigma}_g > 0$ denoting positive definite.

Remark

In this framework, the problem of maximizing the log-likelihood function $\mathcal{L}(\boldsymbol{\psi})$ is not well posed, because it may present

1. *singularities (degenerate solutions)*
2. *local maxima.*

MLE of Gaussian mixtures

An important case concerns mixture of Gaussian distributions

$$p(\mathbf{x};\boldsymbol{\psi}) = \pi_1\phi(\mathbf{x};\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) + \cdots + \pi_G\phi(\mathbf{x};\boldsymbol{\mu}_G,\boldsymbol{\Sigma}_G) \qquad \mathbf{x} \in \mathbb{R}^d \,.$$

and the log-likelihood function specializes as

$$\mathcal{L}(\boldsymbol{\psi}) = \sum_{n=1}^{N} \log \left[\sum_{g=1}^{G} \pi_g \frac{1}{|2\pi\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_g) \right\} \right]$$

where $\boldsymbol{\psi} = \{(\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g),\ g = 1, \ldots, G\} \in \boldsymbol{\Psi}$ and $\boldsymbol{\Psi}$ being the parameter space:

$$\boldsymbol{\Psi} = \{(\pi_1, \ldots, \pi_G, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_G) \in \mathbb{R}^{G[1+d+(d^2+d)/2]} \ :$$
$$\pi_1 + \cdots + \pi_G = 1, \pi_g > 0,\ \boldsymbol{\Sigma}_g > 0 \quad \text{for } g = 1, ..., G\} \,,$$

with $\boldsymbol{\Sigma}_g > 0$ denoting positive definite.

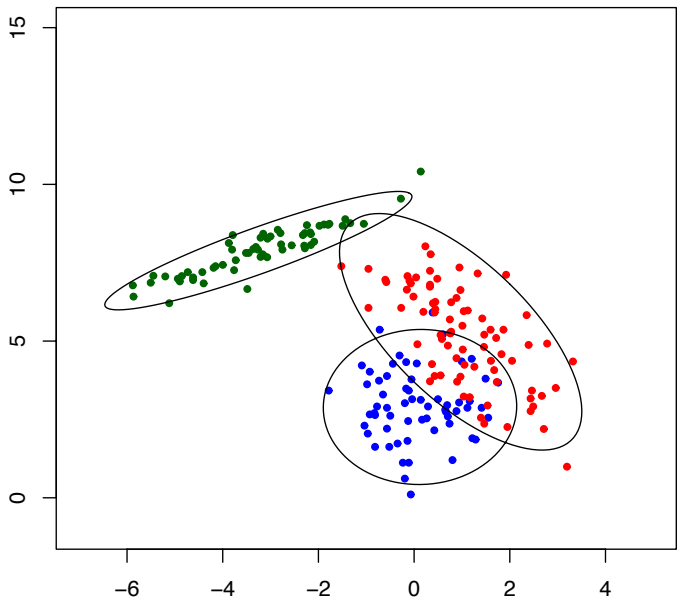
Remark

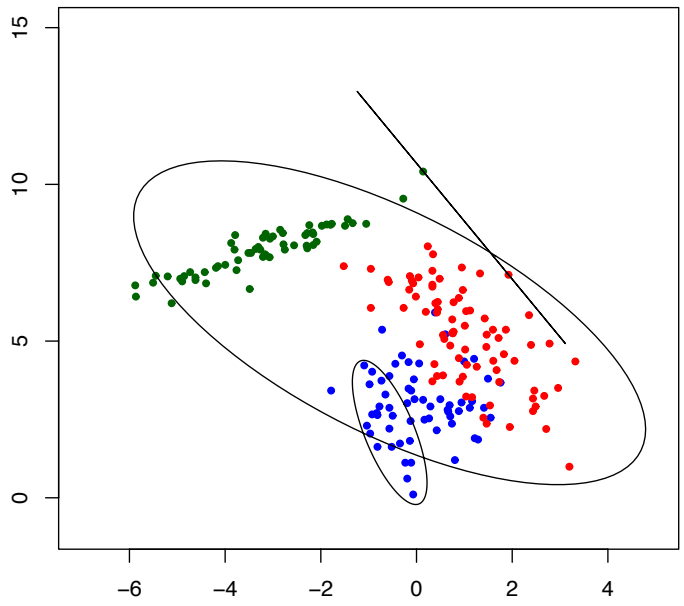
In this framework, the problem of maximizing the log-likelihood function $\mathcal{L}(\boldsymbol{\psi})$ is not well posed, because it may present

1. *singularities (degenerate solutions)*
2. *local maxima.*

Numerical issue 1: singularities (degenerate solutions)

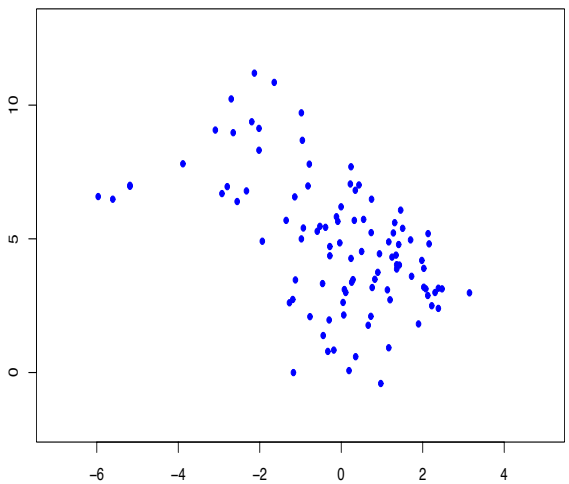
Singularities of the likelihood function are caused by clusters containing few data points lying in a lower-dimensional subspace (in the case of multivariate data).These components have been referred to as *degenerate*.



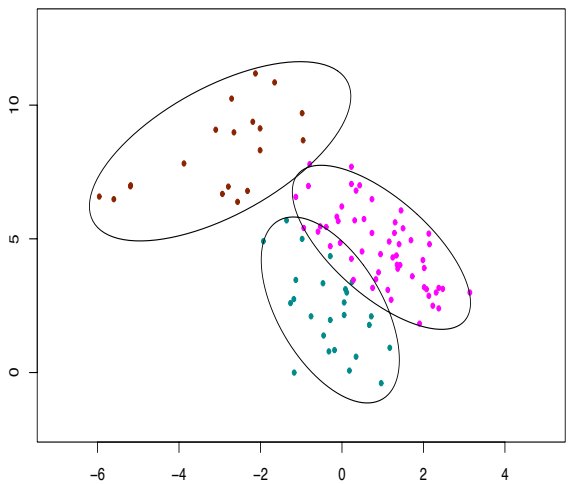


Numerical issue 2: Local maxima and spurious maximizers

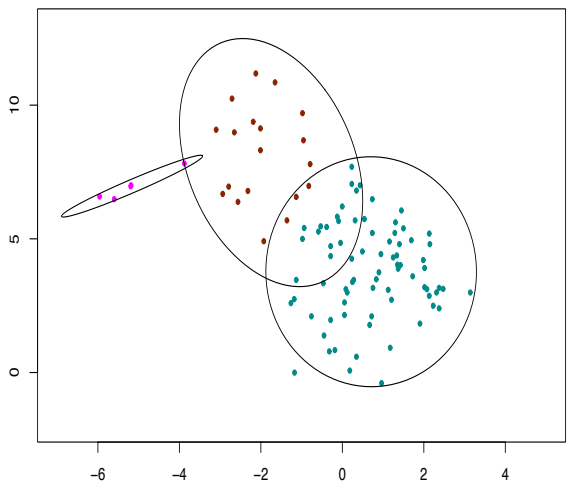
A particular type of local maxima occurs as a consequence of a fitted component having a very small (generalized) variance compared to the others (few data points either relatively close together or almost lying in a lower-dimensional subspace) see Day (1969).



Data



$\mathcal{L}(\hat{\psi}) = -600.15$

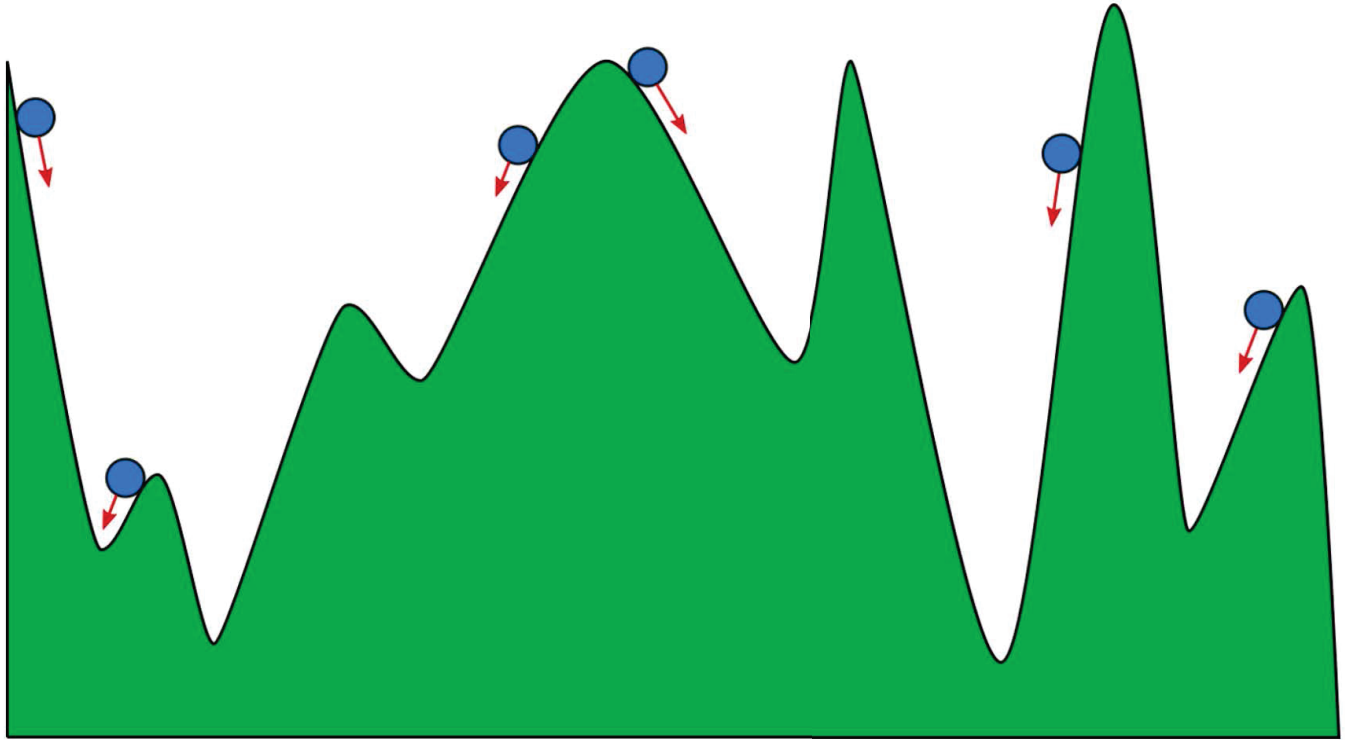


$\mathcal{L}(\psi^*) = -609.46$

- Spurious maximizers are investigated in detail in Section 3 of McLachlan and Peel (2000).

Numerical issue 3: Local maxima

Starting from some initial guess $\mathbf{x}^{(0)}$ (usually selected randomly), a descent algorithm generates a sequence $\{\mathbf{x}^{(r)}\}_r$ of points that converges to some local minimum $\mathbf{x}^*$ (no guarantee convergence towards global optimum).



Therefore, strategies based on multiple initial starts could be adopted.

The EM algorithm

The *Expectation-Maximization (EM)* algorithm is a broadly applicable approach to iterative computation of maximum likelihood estimates, useful in a variety of incomplete-data problems where algorithms such as the Newthon-Raphson method may turn out to be more complicated.

On each iteration ot the EM algorithm, there are two steps - called

- the *Expectation step* (*E-step*) and
- the *Maximization step* (*M-step*).

EM algorithm for univariate Gaussian mixtures 1/3

Let $\mathbf{x} = (x_1, \ldots, x_N)'$ be a sample from a mixture of G Gaussian components

$$p(x; \boldsymbol{\psi}) = \pi_1 \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right\} + \cdots + \pi_G \frac{1}{\sqrt{2\pi\sigma_G^2}} \exp\left\{-\frac{(x-\mu_G)^2}{2\sigma_G^2}\right\}$$

where $\boldsymbol{\psi}$ is the overall parameter, i.e. $\boldsymbol{\psi} = \{\pi_g, \mu_g, \sigma_g^2,\ g = 1, \ldots, G\}$.

At iteration r, based on the sample $\mathbf{x} = \{x_1, \ldots, x_N\}$, let $\boldsymbol{\psi}^{(r)}$ be the current estimate of the overall parameter $\boldsymbol{\psi}$. Consider the quantity

$$\tau_g(x_n; \boldsymbol{\psi}^{(r)}) = \frac{\pi_g^{(r)} \phi\left(x_n; \mu_g^{(r)}, \sigma_g^{2(r)}\right)}{p(x_n; \boldsymbol{\psi}^{(r)})} \quad n = 1, \ldots, N,\ g = 1, \ldots, G.$$

Remark

In the classification setting, according to the Bayes' rule, $\tau_g(x_n; \boldsymbol{\psi}^{(r)})$ is the posterior probability that the unit x_n comes from the group g.

EM algorithm for univariate Gaussian mixtures 1/3

Let $\mathbf{x} = (x_1, \ldots, x_N)'$ be a sample from a mixture of G Gaussian components

$$p(x; \boldsymbol{\psi}) = \pi_1 \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right\} + \cdots + \pi_G \frac{1}{\sqrt{2\pi\sigma_G^2}} \exp\left\{-\frac{(x-\mu_G)^2}{2\sigma_G^2}\right\}$$

where $\boldsymbol{\psi}$ is the overall parameter, i.e. $\boldsymbol{\psi} = \{\pi_g, \mu_g, \sigma_g^2,\ g = 1, \ldots, G\}$.

At iteration r, based on the sample $\mathbf{x} = \{x_1, \ldots, x_N\}$, let $\boldsymbol{\psi}^{(r)}$ be the current estimate of the overall parameter $\boldsymbol{\psi}$. Consider the quantity

$$\tau_g(x_n; \boldsymbol{\psi}^{(r)}) = \frac{\pi_g^{(r)} \phi\left(x_n; \mu_g^{(r)}, \sigma_g^{2(r)}\right)}{p(x_n; \boldsymbol{\psi}^{(r)})} \quad n = 1, \ldots, N,\ g = 1, \ldots, G.$$

Remark

In the classification setting, according to the Bayes' rule, $\tau_g(x_n; \boldsymbol{\psi}^{(r)})$ is the posterior probability that the unit x_n comes from the group g.

EM algorithm for univariate Gaussian mixtures 1/3

Let $\mathbf{x} = (x_1, \ldots, x_N)'$ be a sample from a mixture of G Gaussian components

$$p(x; \boldsymbol{\psi}) = \pi_1 \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right\} + \cdots + \pi_G \frac{1}{\sqrt{2\pi\sigma_G^2}} \exp\left\{-\frac{(x-\mu_G)^2}{2\sigma_G^2}\right\}$$

where $\boldsymbol{\psi}$ is the overall parameter, i.e. $\boldsymbol{\psi} = \{\pi_g, \mu_g, \sigma_g^2,\ g = 1, \ldots, G\}$.

At iteration r, based on the sample $\mathbf{x} = \{x_1, \ldots, x_N\}$, let $\boldsymbol{\psi}^{(r)}$ be the current estimate of the overall parameter $\boldsymbol{\psi}$. Consider the quantity

$$\tau_g(x_n; \boldsymbol{\psi}^{(r)}) = \frac{\pi_g^{(r)} \phi\left(x_n; \mu_g^{(r)}, \sigma_g^{2(r)}\right)}{p(x_n; \boldsymbol{\psi}^{(r)})} \quad n = 1, \ldots, N,\ g = 1, \ldots, G.$$

Remark

In the classification setting, according to the Bayes' rule, $\tau_g(x_n; \boldsymbol{\psi}^{(r)})$ is the posterior probability that the unit x_n comes from the group g.

EM algorithm for univariate Gaussian mixtures 2/3

After some algebra, we get the estimate of the weight π_g and of the mean μ_g on the $(r+1)$th iteration

$$\pi_g^{(r+1)} = \frac{1}{N} \sum_{n=1}^{N} \tau_{ng}^{(r+1)}$$

$$\mu_g^{(r+1)} = \frac{\sum_{n=1}^{N} \tau_{ng}^{(r+1)} x_n}{\sum_{n=1}^{N} \tau_{ng}^{(r+1)}}$$

for $g = 1, \ldots, G$.

EM algorithm for univariate Gaussian mixtures 3/3

As for the estimation of the variances σ_g^2, we distinguish two cases

- The heteroscedastic case (no constraints on σ_g^2, for $g = 1, \ldots, G$)

$$\left(\sigma_g^2\right)^{(r+1)} = \frac{\sum_{n=1}^{N} \tau_{ng}^{(r+1)} \left(x_n - \mu_g^{(r+1)}\right)^2}{\sum_{n=1}^{N} \tau_{ng}^{(r+1)}}$$

- The homoscedastic case ($\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_G^2 = \sigma^2$)

$$\left(\sigma^2\right)^{(r+1)} = \frac{1}{N} \sum_{g=1}^{G} \sum_{n=1}^{N} \tau_{ng}^{(r+1)} \left(x_n - \mu_g^{(r+1)}\right)^2 .$$

EM algorithm for univariate Gaussian mixtures 3/3

As for the estimation of the variances σ_g^2, we distinguish two cases

- The heteroscedastic case (no constraints on σ_g^2, for $g = 1, \ldots, G$)

$$\left(\sigma_g^2\right)^{(r+1)} = \frac{\sum_{n=1}^{N} \tau_{ng}^{(r+1)} \left(x_n - \mu_g^{(r+1)}\right)^2}{\sum_{n=1}^{N} \tau_{ng}^{(r+1)}}$$

- The homoscedastic case ($\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_G^2 = \sigma^2$)

$$\left(\sigma^2\right)^{(r+1)} = \frac{1}{N} \sum_{g=1}^{G} \sum_{n=1}^{N} \tau_{ng}^{(r+1)} \left(x_n - \mu_g^{(r+1)}\right)^2 .$$

EM algorithm for multivariate Gaussian mixtures 1/2

Let $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a sample taken from a mixture of G multivariate normal distribution

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 \phi(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \cdots + \pi_g \phi(\mathbf{x}; \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G)$$

where

$$\phi(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \frac{1}{|2\pi\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_g)' \boldsymbol{\Sigma}_g^{-1} (\mathbf{x} - \boldsymbol{\mu}_g)\right\} \quad g = 1, \ldots, G.$$

and $\boldsymbol{\psi} = \{\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g^2,\ g = 1, \ldots, G\}$.

At iteration r, based on the sample $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, let $\boldsymbol{\psi}^{(r)}$ be the current estimate of the overall parameter $\boldsymbol{\psi}$. Consider the quantity

$$\tau_g(\mathbf{x}_n; \boldsymbol{\psi}^{(r)}) = \frac{\pi_g^{(r)} \phi\left(\mathbf{x}_n; \mu_g^{(r)}, \sigma_g^{2(r)}\right)}{p(\mathbf{x}_n; \boldsymbol{\psi}^{(r)})} \quad n = 1, \ldots, N,\ g = 1, \ldots, G.$$

Remark

In the classification setting, according to the Bayes' rule, $\tau_g(\mathbf{x}_n; \boldsymbol{\psi}^{(r)})$ is the posterior probability that the unit $\mathbf{x}_n$ comes from the group g.

EM algorithm for multivariate Gaussian mixtures 1/2

Let $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a sample taken from a mixture of G multivariate normal distribution

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 \phi(\mathbf{x}; \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + \cdots + \pi_g \phi(\mathbf{x}; \boldsymbol{\mu}_G, \boldsymbol{\Sigma}_G)$$

where

$$\phi(\mathbf{x}; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g) = \frac{1}{|2\pi\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_g)' \boldsymbol{\Sigma}_g^{-1} (\mathbf{x} - \boldsymbol{\mu}_g)\right\} \quad g = 1, \ldots, G.$$

and $\boldsymbol{\psi} = \{\pi_g, \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g^2,\ g = 1, \ldots, G\}$.

At iteration r, based on the sample $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, let $\boldsymbol{\psi}^{(r)}$ be the current estimate of the overall parameter $\boldsymbol{\psi}$. Consider the quantity

$$\tau_g(\mathbf{x}_n; \boldsymbol{\psi}^{(r)}) = \frac{\pi_g^{(r)} \phi\left(\mathbf{x}_n; \mu_g^{(r)}, \sigma_g^{2(r)}\right)}{p(\mathbf{x}_n; \boldsymbol{\psi}^{(r)})} \quad n = 1, \ldots, N,\ g = 1, \ldots, G.$$

Remark

In the classification setting, according to the Bayes' rule, $\tau_g(\mathbf{x}_n; \boldsymbol{\psi}^{(r)})$ is the posterior probability that the unit $\mathbf{x}_n$ comes from the group g.

EM algorithm for multivariate Gaussian mixtures 2/2

Some algebras yield the following iterative procedure:

$$\pi_g^{(r+1)} = \frac{1}{n}\sum_{g=1}^{g}\tau_{ng}^{(r+1)}$$

$$\boldsymbol{\mu}_g^{(r+1)} = \frac{\sum_{n=1}^{N}\tau_{ng}^{(r+1)}\mathbf{x}_n}{\sum_{n=1}^{N}\tau_{ng}^{(r+1)}}$$

$$\boldsymbol{\Sigma}_g^{(r+1)} = \frac{\sum_{n=1}^{N}\tau_{ng}^{(r+1)}(\mathbf{x}_n-\boldsymbol{\mu}_g^{(r+1)})(\mathbf{x}_n-\boldsymbol{\mu}_g^{(r+1)})'}{\sum_{n=1}^{N}\tau_{ng}^{(r+1)}},$$

for $g = 1, \ldots, G$.

Computational issues: Starting values 1/3

The EM algorithm is an iterative procedure choice of the starting values constitutes an important issue.

- The standard initialization consists in selecting a value for $\boldsymbol{\psi}^{(0)}$. An alternative approach consists in performing the first E-step by specifying, the values of $z_{ng}^{(0)}$ for $n = 1, \ldots, N$ and $g = 1, \ldots, G$.
- Other initialization strategies concern a random initialization which is repeated t times from different random positions and the solution maximizing the observed-data log-likelihood among these t runs is selected.
- In each run, the n vectors $z_{n1}^{(0)}, \ldots, z_{nG}^{(0)}$ can be randomly generated in either a "soft" way, that is by generating G positive values summing to one, or alternatively in a "hard" way through a multinomial distribution with probabilities $(1/G, \ldots, 1/G)'$.

Computational issues: Starting values 1/3

The EM algorithm is an iterative procedure choice of the starting values constitutes an important issue.

- The standard initialization consists in selecting a value for $\boldsymbol{\psi}^{(0)}$. An alternative approach consists in performing the first E-step by specifying, the values of $z_{ng}^{(0)}$ for $n = 1, \ldots, N$ and $g = 1, \ldots, G$.
- Other initialization strategies concern a random initialization which is repeated t times from different random positions and the solution maximizing the observed-data log-likelihood among these t runs is selected.
- In each run, the n vectors $z_{n1}^{(0)}, \ldots, z_{nG}^{(0)}$ can be randomly generated in either a "soft" way, that is by generating G positive values summing to one, or alternatively in a "hard" way through a multinomial distribution with probabilities $(1/G, \ldots, 1/G)'$.

Computational issues: Starting values 1/3

The EM algorithm is an iterative procedure choice of the starting values constitutes an important issue.

- The standard initialization consists in selecting a value for $\boldsymbol{\psi}^{(0)}$. An alternative approach consists in performing the first E-step by specifying, the values of $z_{ng}^{(0)}$ for $n = 1, \ldots, N$ and $g = 1, \ldots, G$.
- Other initialization strategies concern a random initialization which is repeated t times from different random positions and the solution maximizing the observed-data log-likelihood among these t runs is selected.
- In each run, the n vectors $z_{n1}^{(0)}, \ldots, z_{nG}^{(0)}$ can be randomly generated in either a "soft" way, that is by generating G positive values summing to one, or alternatively in a "hard" way through a multinomial distribution with probabilities $(1/G, \ldots, 1/G)'$.

Computational issues: Stopping rule 2/3

The stopping criterion usually adopted is in terms of either the size of the relative change in the parameter estimates of the log-likelihood. This is a measure of lack of progress but not of actual convergence.

- The Aitken acceleration procedure provides a useful estimate of the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm. The **Aitken acceleration** at iteration $r+1$, $r = 0, 1, \ldots$, is given by

$$a^{(r+1)} = \frac{\mathcal{L}(\boldsymbol{\psi}^{(r+2)}) - \mathcal{L}(\boldsymbol{\psi}^{(r+1)})}{\mathcal{L}(\boldsymbol{\psi}^{(r+1)}) - \mathcal{L}(\boldsymbol{\psi}^{(r)})},$$

  where $\mathcal{L}(\boldsymbol{\psi}^{(r+2)})$, $\mathcal{L}(\boldsymbol{\psi}^{(r+1)})$, and $\mathcal{L}(\boldsymbol{\psi}^{(r)})$ are the log-likelihood values computed at iterations $r+2$, $r+1$, and r, respectively.

- Then, the asymptotic estimate of the log-likelihood at iteration $r+2$ is given by

$$\mathcal{L}_\infty^{(r+2)} = \mathcal{L}(\boldsymbol{\psi}^{(r+1)}) + \frac{1}{1 - a^{(r+1)}} \left(\mathcal{L}(\boldsymbol{\psi}^{(r+2)}) - \mathcal{L}(\boldsymbol{\psi}^{(r+1)}) \right).$$

- Thus the EM algorithm can be stopped when $\mathcal{L}_\infty^{(r+2)} - \mathcal{L}_\infty^{(r+1)}$ results less than some desired tolerance ϵ.

Computational issues: Stopping rule 2/3

The stopping criterion usually adopted is in terms of either the size of the relative change in the parameter estimates of the log-likelihood. This is a measure of lack of progress but not of actual convergence.

- The Aitken acceleration procedure provides a useful estimate of the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm. The **Aitken acceleration** at iteration $r+1$, $r=0,1,\ldots$, is given by

$$a^{(r+1)} = \frac{\mathcal{L}(\boldsymbol{\psi}^{(r+2)}) - \mathcal{L}(\boldsymbol{\psi}^{(r+1)})}{\mathcal{L}(\boldsymbol{\psi}^{(r+1)}) - \mathcal{L}(\boldsymbol{\psi}^{(r)})},$$

  where $\mathcal{L}(\boldsymbol{\psi}^{(r+2)})$, $\mathcal{L}(\boldsymbol{\psi}^{(r+1)})$, and $\mathcal{L}(\boldsymbol{\psi}^{(r)})$ are the log-likelihood values computed at iterations $r+2$, $r+1$, and r, respectively.
- Then, the asymptotic estimate of the log-likelihood at iteration $r+2$ is given by

$$\mathcal{L}_{\infty}^{(r+2)} = \mathcal{L}(\boldsymbol{\psi}^{(r+1)}) + \frac{1}{1-a^{(r+1)}}\left(\mathcal{L}(\boldsymbol{\psi}^{(r+2)}) - \mathcal{L}(\boldsymbol{\psi}^{(r+1)})\right).$$

- Thus the EM algorithm can be stopped when $\mathcal{L}_{\infty}^{(r+2)} - \mathcal{L}_{\infty}^{(r+1)}$ results less than some desired tolerance ϵ.

Computational issues: Stopping rule 2/3

The stopping criterion usually adopted is in terms of either the size of the relative change in the parameter estimates of the log-likelihood. This is a measure of lack of progress but not of actual convergence.

- The Aitken acceleration procedure provides a useful estimate of the asymptotic maximum of the log-likelihood at each iteration of the EM algorithm. The **Aitken acceleration** at iteration $r+1$, $r=0,1,\ldots$, is given by

$$a^{(r+1)} = \frac{\mathcal{L}(\boldsymbol{\psi}^{(r+2)}) - \mathcal{L}(\boldsymbol{\psi}^{(r+1)})}{\mathcal{L}(\boldsymbol{\psi}^{(r+1)}) - \mathcal{L}(\boldsymbol{\psi}^{(r)})},$$

  where $\mathcal{L}(\boldsymbol{\psi}^{(r+2)})$, $\mathcal{L}(\boldsymbol{\psi}^{(r+1)})$, and $\mathcal{L}(\boldsymbol{\psi}^{(r)})$ are the log-likelihood values computed at iterations $r+2$, $r+1$, and r, respectively.
- Then, the asymptotic estimate of the log-likelihood at iteration $r+2$ is given by

$$\mathcal{L}_{\infty}^{(r+2)} = \mathcal{L}(\boldsymbol{\psi}^{(r+1)}) + \frac{1}{1-a^{(r+1)}}\left(\mathcal{L}(\boldsymbol{\psi}^{(r+2)}) - \mathcal{L}(\boldsymbol{\psi}^{(r+1)})\right).$$

- Thus the EM algorithm can be stopped when $\mathcal{L}_{\infty}^{(r+2)} - \mathcal{L}_{\infty}^{(r+1)}$ results less than some desired tolerance ϵ.

Computational issues: Rate of convergence 3/3

Remarks

- The EM algorithm is quite slow, compared with other numerical optimization procedures. It can be proved that in a neighborhood of the MLE $\hat{\psi}$, the EM algorithm is essentially a linear iteration.
- Moreover, it can be proved that there exists a domain of attraction of the EM algorithm around a degenerated solution and that convergence to these particular solutions is extremely fast.

Computational issues: Rate of convergence 3/3

Remarks

- The EM algorithm is quite slow, compared with other numerical optimization procedures. It can be proved that in a neighborhood of the MLE $\hat{\psi}$, the EM algorithm is essentially a linear iteration.
- Moreover, it can be proved that there exists a domain of attraction of the EM algorithm around a degenerated solution and that convergence to these particular solutions is extremely fast.

The EM algorithm - Properties 1/3 - Advantages

The EM algorithm has several appealing properties relative to other iterative algorithms such as Newton-Raphson and Fisher's scoring method for finding MLEs. Some of its advantages compared to its competitors are as follows.

1. The EM algorithm is numerically stable, with each EM iteration not decreasing the likelihood (except at the fixed point of the algorithm).
2. Under fairly general conditions, the EM algorithm has reliable global convergence. That is, starting from an arbitrary point $\psi^{(0)} \in \Psi$, convergence is nearly always to a local maximizer, barring very bad luck in the choice of the initial guess $\psi^{(0)}$ or some pathology in the log-likelihood function.
3. The EM algorithm is tipically easily implemented, because it relies on complete-data computations: *i*) the E-step of each iteration only involves taking expectations over complete-data conditional distributions and *ii*) the M-step of each iteration only requires complete-data ML estimation, which is often in simple closed forms.

The EM algorithm - Properties 1/3 - Advantages

The EM algorithm has several appealing properties relative to other iterative algorithms such as Newton-Raphson and Fisher's scoring method for finding MLEs. Some of its advantages compared to its competitors are as follows.

1. The EM algorithm is numerically stable, with each EM iteration not decreasing the likelihood (except at the fixed point of the algorithm).
2. Under fairly general conditions, the EM algorithm has reliable global convergence. That is, starting from an arbitrary point $\boldsymbol{\psi}^{(0)} \in \boldsymbol{\Psi}$, convergence is nearly always to a local maximizer, barring very bad luck in the choice of the initial guess $\boldsymbol{\psi}^{(0)}$ or some pathology in the log-likelihood function.
3. The EM algorithm is tipically easily implemented, because it relies on complete-data computations: *i*) the E-step of each iteration only involves taking expectations over complete-data conditional distributions and *ii*) the M-step of each iteration only requires complete-data ML estimation, which is often in simple closed forms.

The EM algorithm - Properties 1/3 - Advantages

The EM algorithm has several appealing properties relative to other iterative algorithms such as Newton-Raphson and Fisher's scoring method for finding MLEs. Some of its advantages compared to its competitors are as follows.

1. The EM algorithm is numerically stable, with each EM iteration not decreasing the likelihood (except at the fixed point of the algorithm).
2. Under fairly general conditions, the EM algorithm has reliable global convergence. That is, starting from an arbitrary point $\boldsymbol{\psi}^{(0)} \in \boldsymbol{\Psi}$, convergence is nearly always to a local maximizer, barring very bad luck in the choice of the initial guess $\boldsymbol{\psi}^{(0)}$ or some pathology in the log-likelihood function.
3. The EM algorithm is tipically easily implemented, because it relies on complete-data computations: *i*) the E-step of each iteration only involves taking expectations over complete-data conditional distributions and *ii*) the M-step of each iteration only requires complete-data ML estimation, which is often in simple closed forms.

The EM algorithm - Properties 2/3 - Advantages

4. The EM algorithm is generally easy to program, since no evaluation of the likelihood nor its derivatives is involved.
5. The EM algorithm requires small storage space; for instance, it does not have to store the information matrix nor its inverse at any iteration.
6. The cost per iteration is generally low, which can offset the larger number of iterations needed for the EM algorithm compared to other competiting procedures.
7. By watching the monotone increase in likelihood over iterations, it is easy to monitor convergence and programming errors.
8. The EM algorithm can be used to provide estimated values of the missing data.

The EM algorithm - Properties 2/3 - Advantages

4. The EM algorithm is generally easy to program, since no evaluation of the likelihood nor its derivatives is involved.
5. The EM algorithm requires small storage space; for instance, it does not have to store the information matrix nor its inverse at any iteration.
6. The cost per iteration is generally low, which can offset the larger number of iterations needed for the EM algorithm compared to other competiting procedures.
7. By watching the monotone increase in likelihood over iterations, it is easy to monitor convergence and programming errors.
8. The EM algorithm can be used to provide estimated values of the missing data.

The EM algorithm - Properties 2/3 - Advantages

4. The EM algorithm is generally easy to program, since no evaluation of the likelihood nor its derivatives is involved.
5. The EM algorithm requires small storage space; for instance, it does not have to store the information matrix nor its inverse at any iteration.
6. The cost per iteration is generally low, which can offset the larger number of iterations needed for the EM algorithm compared to other competiting procedures.
7. By watching the monotone increase in likelihood over iterations, it is easy to monitor convergence and programming errors.
8. The EM algorithm can be used to provide estimated values of the missing data.

The EM algorithm - Properties 2/3 - Advantages

4. The EM algorithm is generally easy to program, since no evaluation of the likelihood nor its derivatives is involved.
5. The EM algorithm requires small storage space; for instance, it does not have to store the information matrix nor its inverse at any iteration.
6. The cost per iteration is generally low, which can offset the larger number of iterations needed for the EM algorithm compared to other competiting procedures.
7. By watching the monotone increase in likelihood over iterations, it is easy to monitor convergence and programming errors.
8. The EM algorithm can be used to provide estimated values of the missing data.

The EM algorithm - Properties 2/3 - Advantages

4. The EM algorithm is generally easy to program, since no evaluation of the likelihood nor its derivatives is involved.
5. The EM algorithm requires small storage space; for instance, it does not have to store the information matrix nor its inverse at any iteration.
6. The cost per iteration is generally low, which can offset the larger number of iterations needed for the EM algorithm compared to other competiting procedures.
7. By watching the monotone increase in likelihood over iterations, it is easy to monitor convergence and programming errors.
8. The EM algorithm can be used to provide estimated values of the missing data.

The EM algorithm - Properties 3/3 - Criticisms

1. The EM algorithm may converge slowly even in some seemingly innocuous problems and in problems where there is too much 'incomplete information'.
2. The EM algorithm is a deterministic optimization procedure and then it does not guarantee convergence towards the global maximum when there are multiple maxima. Further, in this case, the estimate obtained depends upon the initial value $\psi^{(0)}$.

Labo activity with R

Lab_activity_MM_1.R (lines 1-145).

1.3. Model-Based Clustering

Agenda

- Probabilistic clustering
- Clustering of i.i.d. data via mixture models

Clustering

Clustering methods provide a powerful tool for analyzing data sets with applications in quite different domains such as, e.g., marketing, web commerce, biology, pattern recognition and image processing, document retrieval, and linguistics.

Let $\Omega = \{\omega_1, \ldots, \omega_N\}$ be a set of N objects and consider a set of p numerical features associated to the set Ω, describing the objects of Ω, according to some fixed criterion. For a given $\omega \in \Omega$, we denote by $\mathbf{x} = \mathbf{x}(\omega)$ the p-dimensional feature vector associated to the object ω, i.e. $\mathbf{x} \in \mathbb{R}^d$.

The term *clustering* means subdividing Ω into a partition of G hopefully well separated classes $\Omega_1, \ldots, \Omega_G \subset \Omega$ which will be called textcolorred1*clusters*, thereby using the data set $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of observed data.

It is expected that objects from the same cluster Ω_g comprise mainly similar or neighbouring data points, whereas data points from different clusters will be dissimilar and not too close to each other, as a rule.

Clustering

Clustering methods provide a powerful tool for analyzing data sets with applications in quite different domains such as, e.g., marketing, web commerce, biology, pattern recognition and image processing, document retrieval, and linguistics.

Let $\Omega = \{\omega_1, \ldots, \omega_N\}$ be a set of N objects and consider a set of p numerical features associated to the set Ω, describing the objects of Ω, according to some fixed criterion. For a given $\omega \in \Omega$, we denote by $\mathbf{x} = \mathbf{x}(\omega)$ the p-dimensional feature vector associated to the object ω, i.e. $\mathbf{x} \in \mathbb{R}^d$.

The term *clustering* means subdividing Ω into a partition of G hopefully well separated classes $\Omega_1, \ldots, \Omega_G \subset \Omega$ which will be called textcolorred1*clusters*, thereby using the data set $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of observed data.

It is expected that objects from the same cluster Ω_g comprise mainly similar or neighbouring data points, whereas data points from different clusters will be dissimilar and not too close to each other, as a rule.

Clustering

Clustering methods provide a powerful tool for analyzing data sets with applications in quite different domains such as, e.g., marketing, web commerce, biology, pattern recognition and image processing, document retrieval, and linguistics.

Let $\Omega = \{\omega_1, \ldots, \omega_N\}$ be a set of N objects and consider a set of p numerical features associated to the set Ω, describing the objects of Ω, according to some fixed criterion. For a given $\omega \in \Omega$, we denote by $\mathbf{x} = \mathbf{x}(\omega)$ the p-dimensional feature vector associated to the object ω, i.e. $\mathbf{x} \in \mathbb{R}^d$.

The term *clustering* means subdividing Ω into a partition of G hopefully well separated classes $\Omega_1, \ldots, \Omega_G \subset \Omega$ which will be called textcolorred1*clusters*, thereby using the data set $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of observed data.

It is expected that objects from the same cluster Ω_g comprise mainly similar or neighbouring data points, whereas data points from different clusters will be dissimilar and not too close to each other, as a rule.

Clustering

Clustering methods provide a powerful tool for analyzing data sets with applications in quite different domains such as, e.g., marketing, web commerce, biology, pattern recognition and image processing, document retrieval, and linguistics.

Let $\Omega = \{\omega_1, \ldots, \omega_N\}$ be a set of N objects and consider a set of p numerical features associated to the set Ω, describing the objects of Ω, according to some fixed criterion. For a given $\omega \in \Omega$, we denote by $\mathbf{x} = \mathbf{x}(\omega)$ the p-dimensional feature vector associated to the object ω, i.e. $\mathbf{x} \in \mathbb{R}^d$.

The term *clustering* means subdividing Ω into a partition of G hopefully well separated classes $\Omega_1, \ldots, \Omega_G \subset \Omega$ which will be called textcolorred1*clusters*, thereby using the data set $\underline{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of observed data.

It is expected that objects from the same cluster Ω_g comprise mainly similar or neighbouring data points, whereas data points from different clusters will be dissimilar and not too close to each other, as a rule.

Classification through the Bayes's theorem 1/2

Let

- $\pi_g = P(\Omega_g)$ be the overall or prior probability that a randomly chosen observation comes from the prior gth class Ω_g.
- Assume that each class is provided with a probability density f_g (for example, a Gaussian density) and let

$$f_g(\mathbf{x}) = p(\mathbf{x}|\Omega_g)$$

be the probability density of the unit $\mathbf{x}$ in the gth class, i.e. $\mathbf{X} \sim f_g|\Omega_g$.

In other words, from a practical point of view, in the discrete case, $f_g(\mathbf{x})$ assumes a relatively large value if there is a high probability that $\mathbf{x}$ comes from the kth class, and $f_g(\mathbf{x})$ is small if it is very unlikely if it is unlikely that $\mathbf{x}$ belongs to the gth class.

Classification through the Bayes's theorem 2/2

The **Bayes's theorem** states that:

$$P(\Omega_g|\mathbf{x}) = \frac{\pi_g f_g(\mathbf{x})}{\sum_{h=1}^{G} \pi_h f_h(\mathbf{x})}$$

and we'll denote by

$$\tau_g(\mathbf{x}) = P(\Omega_g|\mathbf{x})$$

the posterior probability that an observation $\mathbf{x}$ belongs to the gth class.

That is, it is the probability that the observation belongs to the gth class, given the predictor value $\mathbf{x}$ for that observation.

Clustering of i.i.d. data via mixture models 1/3

Cluster analysis has been reformulated as a problem of estimating the parameters of a mixture of multivariate distributions. In this case $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a sample of N i.i.d. random vectors having density

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 \phi_1(\mathbf{x}; \boldsymbol{\theta}_1) + \cdots + \pi_G \phi_G(\mathbf{x}; \boldsymbol{\theta}_G) \qquad \mathbf{x} \in \mathbb{R}^d .$$

Even if the mixture model provides no clustering approach in the strong sense, once the mixture model has been fitted, a ***fuzzy classification*** of data into G clusters can be obtained in terms of the fitted posterior probabilities that the point $\mathbf{x}_n$ belongs to the first, ..., Gth components, respectively, of the mixture ($n = 1, \ldots, N$).

Clustering of i.i.d. data via mixture models 1/3

Cluster analysis has been reformulated as a problem of estimating the parameters of a mixture of multivariate distributions. In this case $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a sample of N i.i.d. random vectors having density

$$p(\mathbf{x}; \boldsymbol{\psi}) = \pi_1 \phi_1(\mathbf{x}; \boldsymbol{\theta}_1) + \cdots + \pi_G \phi_G(\mathbf{x}; \boldsymbol{\theta}_G) \qquad \mathbf{x} \in \mathbb{R}^d .$$

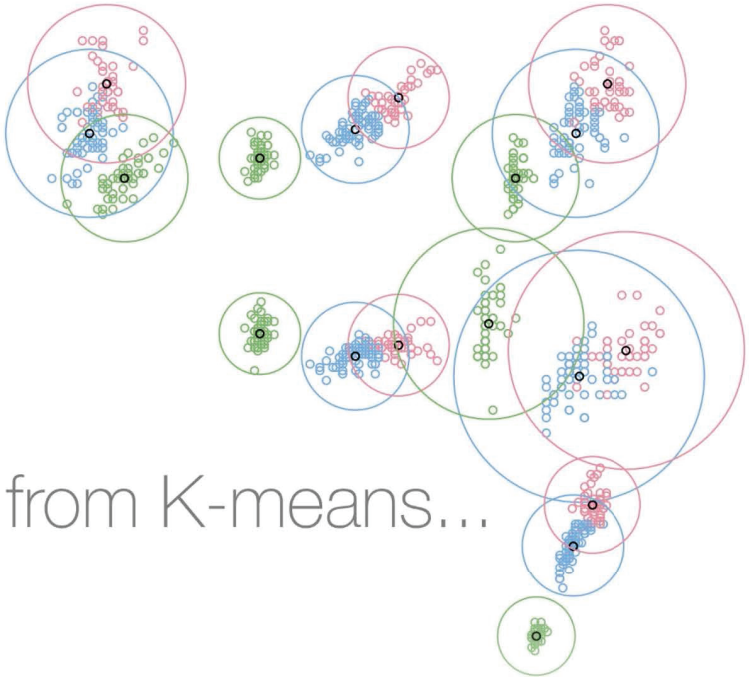
Even if the mixture model provides no clustering approach in the strong sense, once the mixture model has been fitted, a ***fuzzy classification*** of data into G clusters can be obtained in terms of the fitted posterior probabilities that the point $\mathbf{x}_n$ belongs to the first, ..., Gth components, respectively, of the mixture ($n = 1, \ldots, N$).

Clustering of i.i.d. data via mixture models 2/3

According to the Bayes' rule, we get the **fitted posterior probabilities** that the point $\mathbf{x}_n$ belongs to the first, ..., Gth components, respectively, of the mixture ($n = 1, \ldots, N$).

$$\tau_g(\mathbf{x}_n; \widehat{\boldsymbol{\psi}}) = \frac{\pi_g \phi(\mathbf{x}_n; \widehat{\boldsymbol{\theta}}_g)}{\sum_{h=1}^{G} \pi_h \phi(\mathbf{x}_n; \widehat{\boldsymbol{\theta}}_h)} = \frac{\pi_g \phi(\mathbf{x}_n; \widehat{\boldsymbol{\theta}}_g)}{p(\mathbf{x}_n; \widehat{\boldsymbol{\psi}})} \quad g = 1, \ldots, G \, .$$

Clustering of i.i.d. data via mixture models 3/3



The Maximum A Posteriori Probability (MAP) criterion

Finally, an outright assignment of the data into G clusters is achieved by assigning each data point to the component to which it has the highest estimated posterior probabilities of belonging.

T

Maximum A Posteriori Probability (MAP)

We estimate the component-label vector $\mathbf{z}_n$ by $\widehat{\mathbf{z}}_n$ where $\widehat{z}_{ng} = (\widehat{\mathbf{z}}_n)_g$, where

$$\widehat{z}_{ng} = \begin{cases} 1 & \text{if } g = \arg\max_h \tau_h(\mathbf{x}_n; \widehat{\boldsymbol{\psi}}) \\ 0 & \text{otherwise.} \end{cases}$$

In other words, a crisp clustering is obtained from a fuzzy clustering based on the *maximum a posteriori probability* criterion.

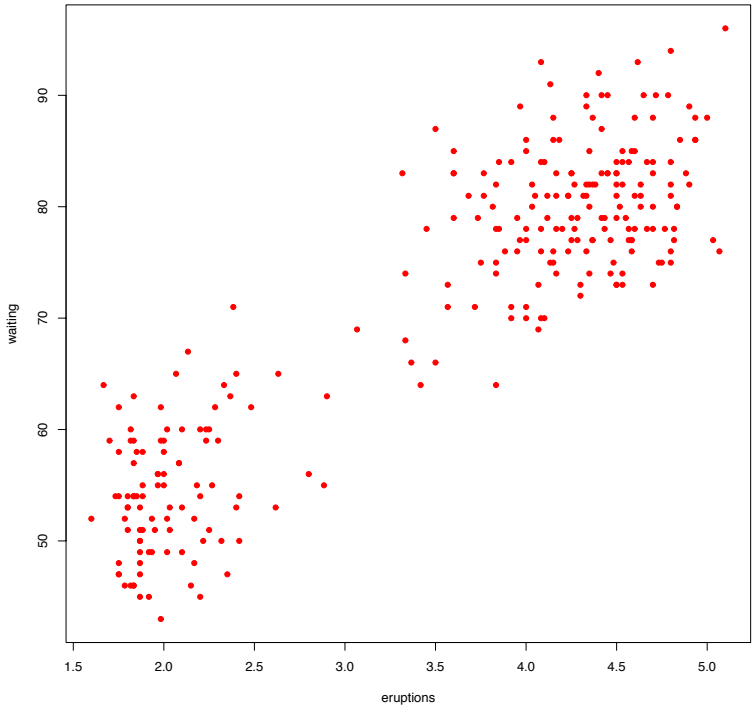
Example: Faithful data 1/4

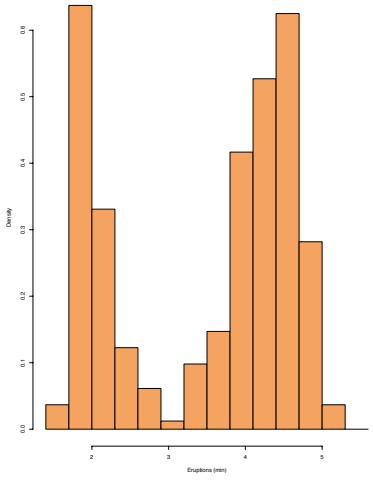
Faithful data concern the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.

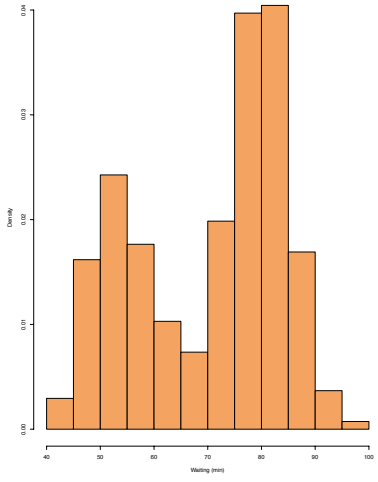
Data consist of $N = 272$ observations on two numeric variables:

eruptions:	Eruption time (in mins)
waiting:	Waiting time to next eruption (in mins)

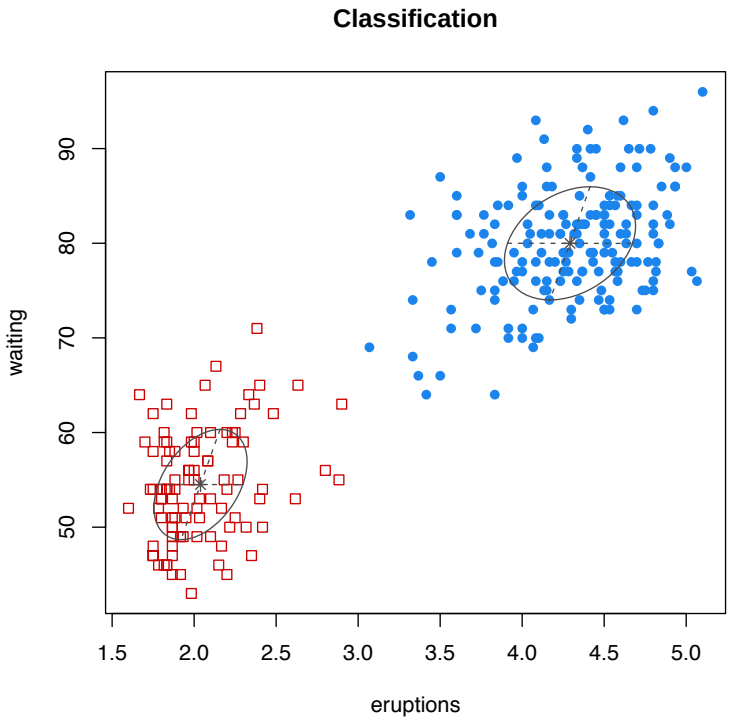
Example: Faithful data 2/4





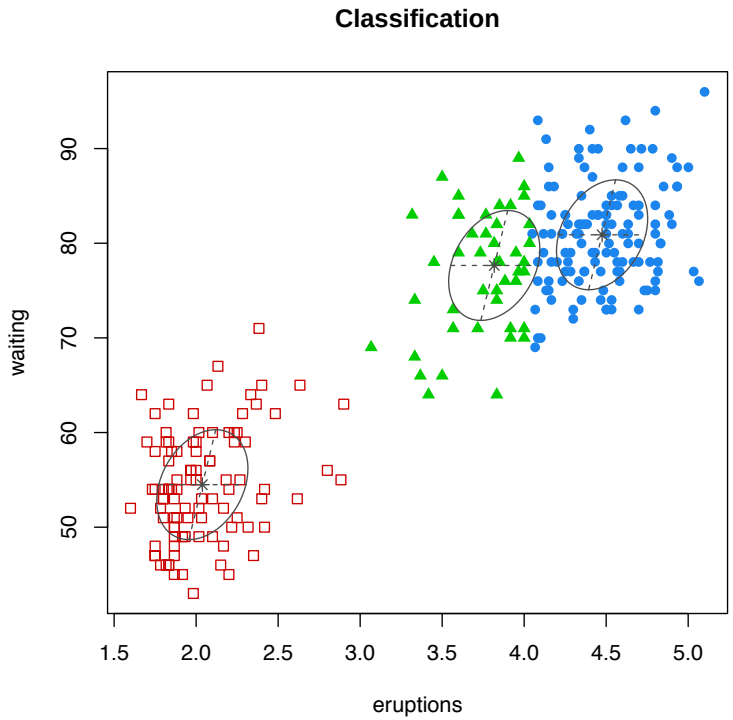


Example: Faithful data 3/4



$$n_1 = 175,\ n_2 = 97$$

$$\pi_1 = 0.6432,\ \pi_2 = 0.3568$$



$$n_1 = 130,\ n_2 = 97,\ n_3 = 45$$

$$\pi_1 = 0.4633,\ \pi_2 = 0.3564,\ \pi_3 = 0.1803$$

Example: Faithful data 4/4

ω_n	*eruptions*	*waiting*	$P(\omega_n \in \Omega_1)$	$P(\omega_n \in \Omega_2)$	$P(\omega_n \in \Omega_3)$	class
1	3.600	79	0.021817	0.000000	0.978183	3
2	1.800	54	0.000000	1.000000	0.000000	2
3	3.333	74	0.002522	0.000021	0.997458	3
4	2.283	62	0.000000	1.000000	0.000000	2
5	4.533	85	0.983897	0.000000	0.016103	1
34	4.033	80	0.479591	0.000000	0.520409	3
74	4.000	71	0.468440	0.000000	0.531560	3
105	4.050	81	0.509674	0.000000	0.490326	1
156	4.000	70	0.475111	0.000000	0.524889	3
158	4.083	93	0.500779	0.000000	0.499221	1

Example: Faithful data 4/4

ω_n	*eruptions*	*waiting*	$P(\omega_n \in \Omega_1)$	$P(\omega_n \in \Omega_2)$	$P(\omega_n \in \Omega_3)$	class
1	3.600	79	0.021817	0.000000	0.978183	3
2	1.800	54	0.000000	1.000000	0.000000	2
3	3.333	74	0.002522	0.000021	0.997458	3
4	2.283	62	0.000000	1.000000	0.000000	2
5	4.533	85	0.983897	0.000000	0.016103	1
34	4.033	80	0.479591	0.000000	0.520409	3
74	4.000	71	0.468440	0.000000	0.531560	3
105	4.050	81	0.509674	0.000000	0.490326	1
156	4.000	70	0.475111	0.000000	0.524889	3
158	4.083	93	0.500779	0.000000	0.499221	1

1.4 Measures of Class Agreement

Agenda

- Confusion matrix
- The Adjusted Rand Index (ARI)

Confusion matrix

When the true classifications are known for the data, we can compute the *confusion matrix* C which is a contigency table containing the information about actual and predicted classifications.

In particular the entry c_{hg} denotes the relative frequency of units coming from Ω_j and classified in Ω_g. The trace of C, namely tr(C), gives the percentage of right classified units and then 1-tr(C) is the *misclassification error* δ.

Remark: pay attention to label switching

	predicted		
True	A	B	C
1	320	34	51
2	48	238	31
3	61	31	185

	predicted		
True	A	B	C
1	48	31	238
2	320	51	34
3	61	185	31

Class agreement

A more elaborate approach is based on measures of class agreement. Given a set of N objects $\mathcal{O} = \{\omega_1, \ldots, \omega_N\}$, suppose $U = \{u_l, \ldots, u_r\}$ and $V = \{v_1, \ldots, v_c\}$ represent two different partitions of $\mathcal{O}$, i.e.,i.e., the entries in U and V are subsets of Ω such that $\cup_{i=1}^{r} u_i = \mathcal{O} = \cup_{j=1}^{c} v_j$, $u_i \cap u_{i'} = \emptyset$ for $1 \leq i \neq i' \leq r$ and $v_j \cap v_{j'} = \emptyset$ for $1 \leq j \neq j' \leq c$.

		Partition V					
	Class	v_1	v_2	$\cdots$	v_j	$\cdots$	v_c	
	u_1	n_{11}	n_{12}	$\cdots$	n_{1j}	$\cdots$	n_{1c}	$n_{1.}$
	u_2	n_{21}	n_{22}	$\cdots$	n_{2j}	$\cdots$	n_{2c}	$n_{2.}$
	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\ldots$	$\ldots$	$\vdots$	$\vdots$
Partition U	u_i	n_{i1}	n_{i2}	$\cdots$	n_{ij}	$\cdots$	n_{ic}	$n_{i.}$
	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\ldots$	$\ldots$	$\vdots$	$\vdots$
	u_r	n_{r1}	n_{r2}	$\cdots$	n_{rj}	$\cdots$	n_{rc}	$n_{r.}$
		$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.j}$	$\cdots$	$n_{.c}$	N

The Rand Index 1/2

The original *Rand index* (RI; Rand, 1971) is based on a measure of correspondence between U and V on how object pairs are classified in the R x C contingency table. Specifically, there are four different types among the $\binom{N}{2}$ distinct pairs that could be found:

1. objects in the pair are placed in the same class in U and in the same class in V;
2. objects in the pair are placed in different classes in U and in different classes in V;
3. objects in the pair are placed in different classes in U and in the same class in V;
4. objects in the pair are placed in the same class in U and in classes in V.

Types 1. and 2 are typically interpreted as agreements in the classification of the objects from a pair; types 3. and 4. represent disagreements.

The Rand Index 1/2

The Rand Index (RI)

The **Rand index (RI)** is defined by

$$\text{RI} = \frac{\text{number of agreements}}{\text{number of agreements + number of disagreements}}.$$

The RI assumes values between 0 and 1, where 0 indicates no pairwise agreement between the MAP classification and true group membership and 1 indicates perfect agreement.

The Adjusted Rand Index

One criticism of the RI is that its expected value is greater than 0, making smaller values difficult to interpret.

Hubert and Arabie (1985) have proposed a correction to the Rand index called *Adjusted Rand Index* (ARI).

The Rand Index (RI)

The **Adjusted Rand index (ARI** corrects the RI for chance by allowing for the possibility that classification performed randomly will correctly classify some observations

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}(\text{RI})}{\max(\text{RI}) - \mathbb{E}(\text{RI})},$$

where the expected value $\mathbb{E}(\text{RI})$ of the Rand Index is computed considering all pairs of distinct partitions picked at random, subject to having the original number of classes and objects in each.

Thus, the ARI has an expected value of 0 and perfect classification would result in a value equal to 1.

The Adjusted Rand Index

One criticism of the RI is that its expected value is greater than 0, making smaller values difficult to interpret.

Hubert and Arabie (1985) have proposed a correction to the Rand index called *Adjusted Rand Index* (ARI).

The Rand Index (RI)

The **Adjusted Rand index (ARI** corrects the RI for chance by allowing for the possibility that classification performed randomly will correctly classify some observations

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}(\text{RI})}{\max(\text{RI}) - \mathbb{E}(\text{RI})},$$

where the expected value $\mathbb{E}(\text{RI})$ of the Rand Index is computed considering all pairs of distinct partitions picked at random, subject to having the original number of classes and objects in each.

Thus, the ARI has an expected value of 0 and perfect classification would result in a value equal to 1.

Labo activity with R

Lab_activity_MM_1.R (lines 150-224).

Model selection

The problem

How to estimate the number G of groups?

- For model selection, usually we conside BIC and ICL,

Criterion	*Definition*
BIC	$= -2\mathcal{L}(\boldsymbol{\psi}) + K \log N$
ICL	$\approx$ BIC + $\sum_n 1_{\hat{z}_{ig}} \ln \hat{z}_{ig}$

  are usually adopted, where K is the number of parameter in the model.

Remark

There may be no particular reasons for choosing a single best model over the other ones. On the contrary, it makes more sense to "deselect" models that are obviously poor, maintaining a subset for further considerations.

Model selection

The problem

How to estimate the number G of groups?

- For model selection, usually we conside BIC and ICL,

Criterion	*Definition*
BIC	$= -2\mathcal{L}(\boldsymbol{\psi}) + K \log N$
ICL	$\approx$ BIC + $\sum_n 1_{\hat{z}_{ig}} \ln \hat{z}_{ig}$

  are usually adopted, where K is the number of parameter in the model.

Remark

There may be no particular reasons for choosing a single best model over the other ones. On the contrary, it makes more sense to "deselect" models that are obviously poor, maintaining a subset for further considerations.

Parameter estimation and classification

Assume we are provided with a set of N independent observation pairs $\{\mathbf{x}_1, \ldots, \mathbf{x}_N)\}$ drawn from a mixture model. Then:

1. for fixed G, estimate model parameters, usually according to the maximum likelihood approach, using the EM algorithm (see later on);
2. if G is unknown, then
    1. repeat step 1 for different number G of groups;
    2. select G according to model selection criteria like BIC, ICL (see later on);
3. based on the estimate $\widehat{\psi}$, compute the posterior probability $\tau_g(\mathbf{x}_n; \widehat{\psi})$ that the nth unit $\mathbf{x}_n$ belongs to the gth group Ω_g and classify units into groups according to the maximum a posteriori probability (MAP) criterion.

Parameter estimation and classification

Assume we are provided with a set of N independent observation pairs $\{\mathbf{x}_1, \ldots, \mathbf{x}_N)\}$ drawn from a mixture model. Then:

1. for fixed G, estimate model parameters, usually according to the maximum likelihood approach, using the EM algorithm (see later on);
2. if G is unknown, then
    1. repeat step 1 for different number G of groups;
    2. select G according to model selection criteria like BIC, ICL (see later on);
3. based on the estimate $\widehat{\psi}$, compute the posterior probability $\tau_g(\mathbf{x}_n; \widehat{\psi})$ that the nth unit $\mathbf{x}_n$ belongs to the gth group Ω_g and classify units into groups according to the maximum a posteriori probability (MAP) criterion.

Parameter estimation and classification

Assume we are provided with a set of N independent observation pairs $\{\mathbf{x}_1, \ldots, \mathbf{x}_N)\}$ drawn from a mixture model. Then:

1. for fixed G, estimate model parameters, usually according to the maximum likelihood approach, using the EM algorithm (see later on);
2. if G is unknown, then
    1. repeat step 1 for different number G of groups;
    2. select G according to model selection criteria like BIC, ICL (see later on);
3. based on the estimate $\widehat{\psi}$, compute the posterior probability $\tau_g(\mathbf{x}_n; \widehat{\psi})$ that the nth unit $\mathbf{x}_n$ belongs to the gth group Ω_g and classify units into groups according to the maximum a posteriori probability (MAP) criterion.

1.5. Parsimonious Model-Based Clustering

Agenda

- Covariance structure of Gaussian multivariate mixtures
- Parsimonious Models

Covariance structure of Gaussian multivariate mixtures

Let $p(\mathbf{x};\boldsymbol{\psi})$ be the density of a mixture of G multivariate normal distributions:

$$p(\mathbf{x};\boldsymbol{\psi}) = \pi_1 f(\mathbf{x};\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) + \cdots + \pi_G f(\mathbf{x};\boldsymbol{\mu}_G,\boldsymbol{\Sigma}_G)$$

where

$$f(\mathbf{x};\boldsymbol{\mu}_g,\boldsymbol{\Sigma}_g) = \frac{1}{|2\pi\boldsymbol{\Sigma}_g|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_g)\right\} \quad g=1,\ldots,G.$$

In clustering context, four commonly used assumptions on the component variance matrices are considered:

1. $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_G = \sigma^2\mathbf{I}$ with σ^2 unknown;
2. $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_G = \text{diag}(\sigma_1^2,\ldots,\sigma_G^2)$, with $(\sigma_1^2,\ldots,\sigma_G^2)$ unknown and $\text{diag}(a_1,\ldots,a_G)$ denoting the diagonal matrix with entries $(a_1,\ldots,a_G)$;
3. $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_G = \boldsymbol{\Sigma}$ where $\boldsymbol{\Sigma}$ is an unknown symmetric matrix (namely the homoscedastic case);
4. no restriction is placed on the covariance matrices $\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_G$.

Parsimonious model-based clustering 1/5

Banfield and Raftery (1993), Celeux and Govaert (1995) proposed a general framework for geometric cross-cluster constraints in multivariate normal mixtures by parameterizing covariance matrices through eigenvalue decomposition in the form

$$\boldsymbol{\Sigma}_g = \lambda_g \mathbf{D}_g \mathbf{A}_g \mathbf{D}_g' \qquad g = 1, \ldots, G$$

where

- $\mathbf{D}_g$ is the orthogonal matrix of the eigenvectors of $\boldsymbol{\Sigma}_g$,
- $\mathbf{A}_g$ is a diagonal matrix diagonal matrix whose elements are proportional to the eigenvalues of $\boldsymbol{\Sigma}_g$ on the diagonal in a decreasing order,
- λ_g is an associated constant of proportionality.

Parsimonious model-based clustering 2/5

The idea was to treat $\lambda_g, \mathbf{D}_g$ and $\mathbf{A}_g$ as independent sets of parameters and either constrain them to be the same for each cluster or allow them to vary among clusters. When parameters are fixed, clusters will share certain geometric properties

- $\mathbf{D}_g$ governs the *orientation* of the gth component of the mixture,
- $\mathbf{A}_g$ its *shape*, and
- λ_g its *volume*, which is proportional to $\lambda_g^d \det(\mathbf{A}_g)$.

For example, if the largest eigenvalue of Σ_g is much larger than the other eigenvalues, then the gth cluster will be concentrated close to a line in d-space, which will be the first principal component of the distribution of the gth group.

Similarly, if the two largest eigenvalues are of the same magnitude and dominate the other eigenvalues, then the gth cluster will be concentrated close to a plane in d-space.

The gth cluster will be roughly spherical if the largest and smallest eigenvalues of Σ_g are of the same magnitude.

Parsimonious model-based clustering 2/5

The idea was to treat $\lambda_g, \mathbf{D}_g$ and $\mathbf{A}_g$ as independent sets of parameters and either constrain them to be the same for each cluster or allow them to vary among clusters. When parameters are fixed, clusters will share certain geometric properties

- $\mathbf{D}_g$ governs the *orientation* of the gth component of the mixture,
- $\mathbf{A}_g$ its *shape*, and
- λ_g its *volume*, which is proportional to $\lambda_g^d \det(\mathbf{A}_g)$.

For example, if the largest eigenvalue of $\mathbf{\Sigma}_g$ is much larger than the other eigenvalues, then the gth cluster will be concentrated close to a line in d-space, which will be the first principal component of the distribution of the gth group.

Similarly, if the two largest eigenvalues are of the same magnitude and dominate the other eigenvalues, then the gth cluster will be concentrated close to a plane in d-space.

The gth cluster will be roughly spherical if the largest and smallest eigenvalues of Σ_g are of the same magnitude.

Parsimonious model-based clustering 2/5

The idea was to treat $\lambda_g, \mathbf{D}_g$ and $\mathbf{A}_g$ as independent sets of parameters and either constrain them to be the same for each cluster or allow them to vary among clusters. When parameters are fixed, clusters will share certain geometric properties

- $\mathbf{D}_g$ governs the *orientation* of the gth component of the mixture,
- $\mathbf{A}_g$ its *shape*, and
- λ_g its *volume*, which is proportional to $\lambda_g^d \det(\mathbf{A}_g)$.

For example, if the largest eigenvalue of $\mathbf{\Sigma}_g$ is much larger than the other eigenvalues, then the gth cluster will be concentrated close to a line in d-space, which will be the first principal component of the distribution of the gth group.

Similarly, if the two largest eigenvalues are of the same magnitude and dominate the other eigenvalues, then the gth cluster will be concentrated close to a plane in d-space.

The gth cluster will be roughly spherical if the largest and smallest eigenvalues of $\mathbf{\Sigma}_g$ are of the same magnitude.

Parsimonious model-based clustering 2/5

The idea was to treat $\lambda_g, \mathbf{D}_g$ and $\mathbf{A}_g$ as independent sets of parameters and either constrain them to be the same for each cluster or allow them to vary among clusters. When parameters are fixed, clusters will share certain geometric properties

- $\mathbf{D}_g$ governs the *orientation* of the gth component of the mixture,
- $\mathbf{A}_g$ its *shape*, and
- λ_g its *volume*, which is proportional to $\lambda_g^d \det(\mathbf{A}_g)$.

For example, if the largest eigenvalue of $\mathbf{\Sigma}_g$ is much larger than the other eigenvalues, then the gth cluster will be concentrated close to a line in d-space, which will be the first principal component of the distribution of the gth group.

Similarly, if the two largest eigenvalues are of the same magnitude and dominate the other eigenvalues, then the gth cluster will be concentrated close to a plane in d-space.

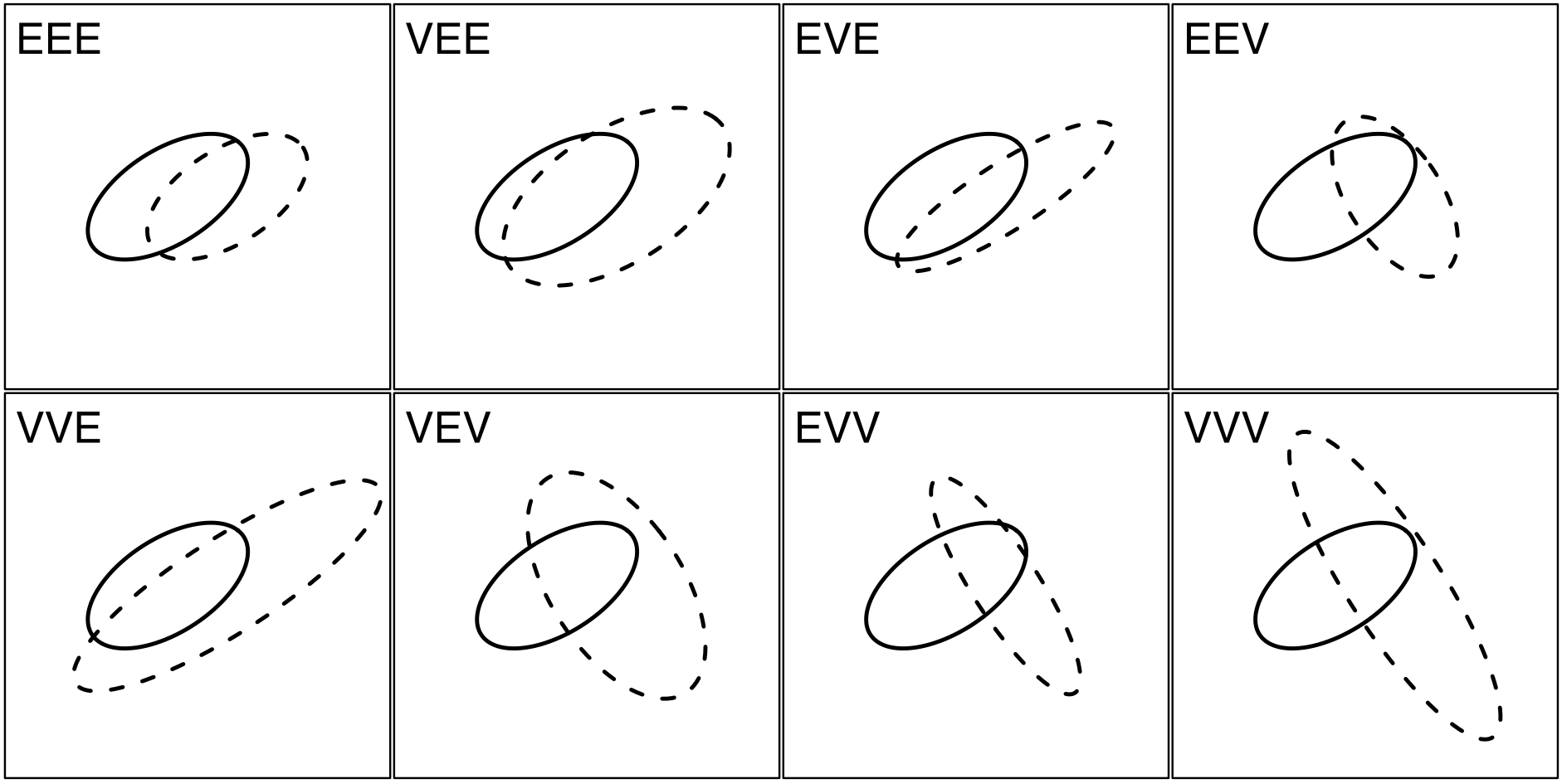
The gth cluster will be roughly spherical if the largest and smallest eigenvalues of $\mathbf{\Sigma}_g$ are of the same magnitude.

Parsimonious model-based clustering 3/5

identifier	*Model*	*Distribution*	*Volume*	*Shape*	*Orientation*	*# parameters*
E		univariate	equal			1
V		univariate	variable			G
EII	$[\lambda \mathbf{I}]$	spherical	equal	equal	NA	$\alpha + 1$
VII	$[\lambda_g \mathbf{I}]$	spherical	variable	equal	NA	$\alpha + d$
EII	$[\lambda \mathbf{A}]$	diagonal	equal	equal	coordinate axes	$\alpha + d$
VEI	$[\lambda_g \mathbf{A}]$	diagonal	variable	equal	coordinate axes	$\alpha + d + G - 1$
EVI	$[\lambda \mathbf{A}_g]$	diagonal	equal	variable	coordinate axes	$\alpha + dG - G + 1$
VVI	$[\lambda_g \mathbf{A}_g]$	diagonal	variable	variable	coordinate axes	$\alpha + dG$
EEE	$[\lambda \mathbf{D}\mathbf{A}\mathbf{D}']$	ellipsoidal	equal	equal	equal	$\alpha + \beta$
VEE	$[\lambda_g \mathbf{D}\mathbf{A}\mathbf{D}']$	ellipsoidal	variable	equal	equal	$\alpha + \beta + G - 1$
EVE	$[\lambda \mathbf{D}\mathbf{A}_g\mathbf{D}']$	ellipsoidal	equal	variable	equal	$\alpha + \beta + (G-1)(d-1)$
VVE	$[\lambda_g \mathbf{D}\mathbf{A}_g\mathbf{D}']$	ellipsoidal	variable	variable	equal	$\alpha + \beta + (G-1)d$
EEV	$[\lambda \mathbf{D}_g\mathbf{A}\mathbf{D}_g']$	ellipsoidal	equal	equal	variable	$\alpha + G\beta - (G-1)d$
VEV	$[\lambda_g \mathbf{D}_g\mathbf{A}\mathbf{D}_g']$	ellipsoidal	variable	equal	variable	$\alpha + G\beta - (G-1)(d-1)$
EVV	$[\lambda \mathbf{D}_g\mathbf{A}_g\mathbf{D}_g']$	ellipsoidal	equal	variable	variable	$\alpha + G\beta - (G-1)$
VVV	$[\lambda_g \mathbf{D}_g\mathbf{A}_g\mathbf{D}_g']$	ellipsoidal	variable	variable	variable	$\alpha + G\beta$

Parameterizations of the covariance matrix $\boldsymbol{\Sigma}_g$. We have $\alpha = Gd$ in the restricted case (equal weights, $\pi_g = 1/G$) and $\alpha = Gd + G - 1$ in the unrestricted case. β denotes the number of parameters of a covariance matrix, i.e. $\beta = d(d+1)/2$.

Parsimonious model-based clustering 4/5



Parsimonious model-based clustering 5/5

Counting parameters:

matrix	*# free parameters*
λ_g	1
$\mathbf{D}_g$	$d(d-1)/2$
$\mathbf{A}_g$	$d-1$
$\boldsymbol{\Sigma}_g$	$d(d+1)/2$

model VEI: $\underbrace{\lambda_1 \mathbf{A}}_{d} \quad \underbrace{\lambda_2 \mathbf{A} \cdots \lambda_G \mathbf{A}}_{G-1}$

model EVI: $\underbrace{\lambda \mathbf{A}_1}_{d} \quad \underbrace{\lambda \mathbf{A}_2}_{d-1} \cdots \quad \underbrace{\lambda_G \mathbf{A}}_{d-1}$

model EVE: $\underbrace{\lambda \mathbf{D} \mathbf{A}_1 \mathbf{D}}_{d(d+1)/2} \quad \underbrace{\lambda \mathbf{D} \mathbf{A}_2 \mathbf{D}}_{d-1} \cdots \quad \underbrace{\lambda \mathbf{D} \mathbf{A}_G \mathbf{D}}_{d-1}$

model VEV: $\underbrace{\lambda_1 \mathbf{D}_1 \mathbf{A} \mathbf{D}_1}_{d(d+1)/2} \quad \underbrace{\lambda_2 \mathbf{D}_2 \mathbf{A} \mathbf{D}_2}_{d(d-1)/2} \cdots \quad \underbrace{\lambda_G \mathbf{D}_G \mathbf{A} \mathbf{D}_G}_{d(d-1)/2}.$

Labo activity with R

Lab_activity_MM_1.R (lines 226-340).