Model based clustering

Mathieu Ribatet—Full Professor of Statistics

Grading

1. Select between one of the following options:
   - □ French Presidential election 2022 / 1st round (regional scale)
   - □ A data set from the Nantes Open Data platform (or other ones)
2. Perform a classification using Gaussian mixtures
3. Comment

References

[1] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

[2] Gilles Celeux and Gérard Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational Statistics & Data Analysis*, 14(3):315 – 332, 1992.

[3] Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781 – 793, 1995.

[4] Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[5] Geoffrey McLachlan and David Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

0. Introduction

What is cluster analysis?

- Group the observations in such a way that each group shares some common characteristics that are not shared with the other groups.
- Clustering is a statistical field related to the development of techniques to identify those groups.
- It is sometimes called unsupervised classification in contrast to (supervised) classification[1] where we still want to group observations but we do have (external) information about the group membership.
- In this lecture we will exclusively on clustering using Gaussian mixtures.
- You probably know already some well-known clustering techniques:
  - K–means
  - Hierarchical clustering

[1] supervised classification is often called discriminant analysis

Old faithful geyser

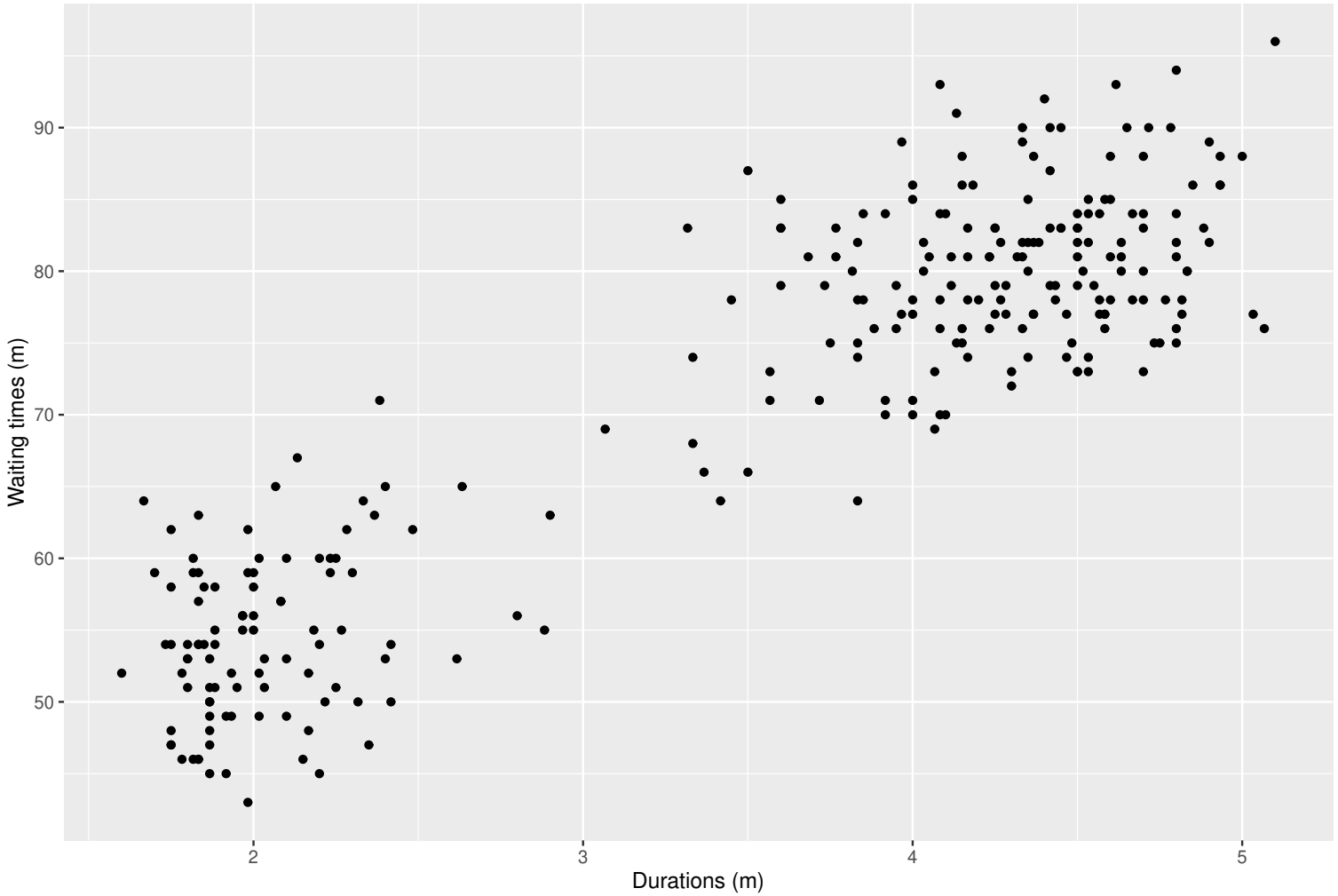


Figure 1: *Old faithful geyser data. Plot of 272 observed durations and times from one eruption to the next one.*

Diabete data

Figure 2: *The diabete data. Measurement made on 145 subjects:* `glucose`, `insuline` *are the area under a plasma glucose/insuline curve and* `sspg` *the steady–state plasma glucose response.*

- The above data sets have something in common:

- The above data sets have something in common: continuous variables.
- This is intended because this lecture focuses on Gaussian mixtures
- Of course we can handle discrete / categorical variables as well but it is beyond the scope of this course ;-)

- The above data sets have something in common: continuous variables.
- This is intended because this lecture focuses on Gaussian mixtures
- Of course we can handle discrete / categorical variables as well but it is beyond the scope of this course ;-)
- The reasoning for focusing exclusively on the Gaussian case is because:
  - it is widely applicable (we often deal with continuous variables and much less with discrete ones)
  - we have explicit expressions
  - numerically simple
  - extension to other distribution is rather easy

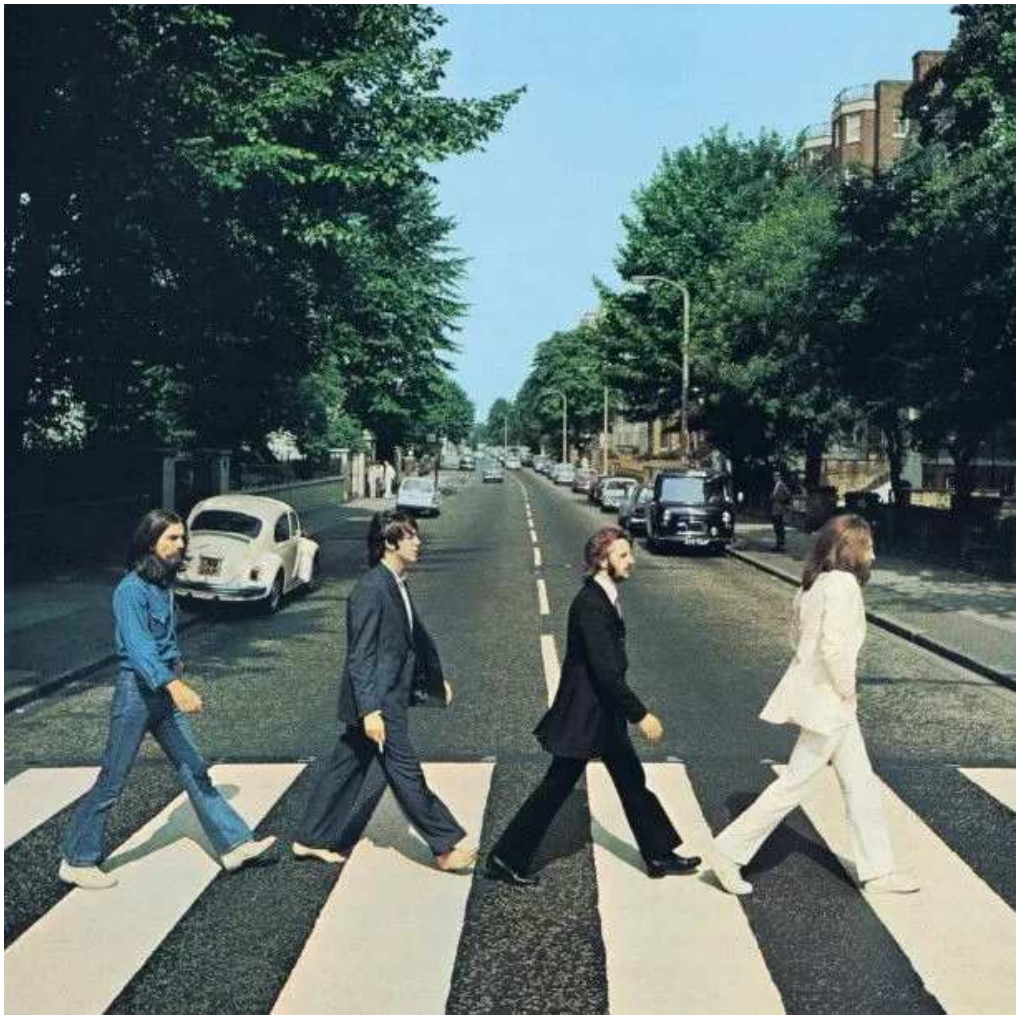


Figure 3: *Just for fun. We will try to segment this photograph.*

1. Mixture models

Finite mixture models

Definition 1. A finite mixture model is (typically) a parametric statistical model whose probability density function is of the form

$$p(y) = \sum_{g=1}^{G} \tau_g f_g(y; \theta_g),$$

where $\tau_g \geq 0$ for $g \in \{1, \ldots, G\}$ and $\sum_{g=1}^{G} \tau_g = 1$, while $f_g(\cdot; \theta_g)$ are probability density functions having (unknown) parameters θ_g.

Finite mixture models

Definition 1. A finite mixture model is (typically) a parametric statistical model whose probability density function is of the form

$$p(y) = \sum_{g=1}^{G} \tau_g f_g(y; \theta_g),$$

where $\tau_g \geq 0$ for $g \in \{1, \ldots, G\}$ and $\sum_{g=1}^{G} \tau_g = 1$, while $f_g(\cdot; \theta_g)$ are probability density functions having (unknown) parameters θ_g. To sum up, it is just a weighted average of G probability density functions.

- G is number of mixture component and is (usually) a hyper parameter, i.e., to be tuned.
- τ_g is the probability that an observation was generated from the g-th component, i.e., membership probability.
- Inference here conists in estimating $\{(\tau_g, \theta_g) : g = 1, \ldots, G\}$.
- Most often f_g, $g \in \{1, \ldots, G\}$, belongs to the "same family of distributions".

Simulation from mixture models

- Recall our mixture model $p(y) = \sum_{g=1}^{G} \tau_g f_g(y; \theta_g)$.
- Simulation from this density is a simple two stage procedure:
  1. Sample $Z \sim \text{Discrete}(\tau_1, \ldots, \tau_G)$;
  2. Conditionally on Z, draw $Y \sim f_Z(\cdot; \theta_Z)$.

Proof.

$\Pr(Y \leq y) =$

□

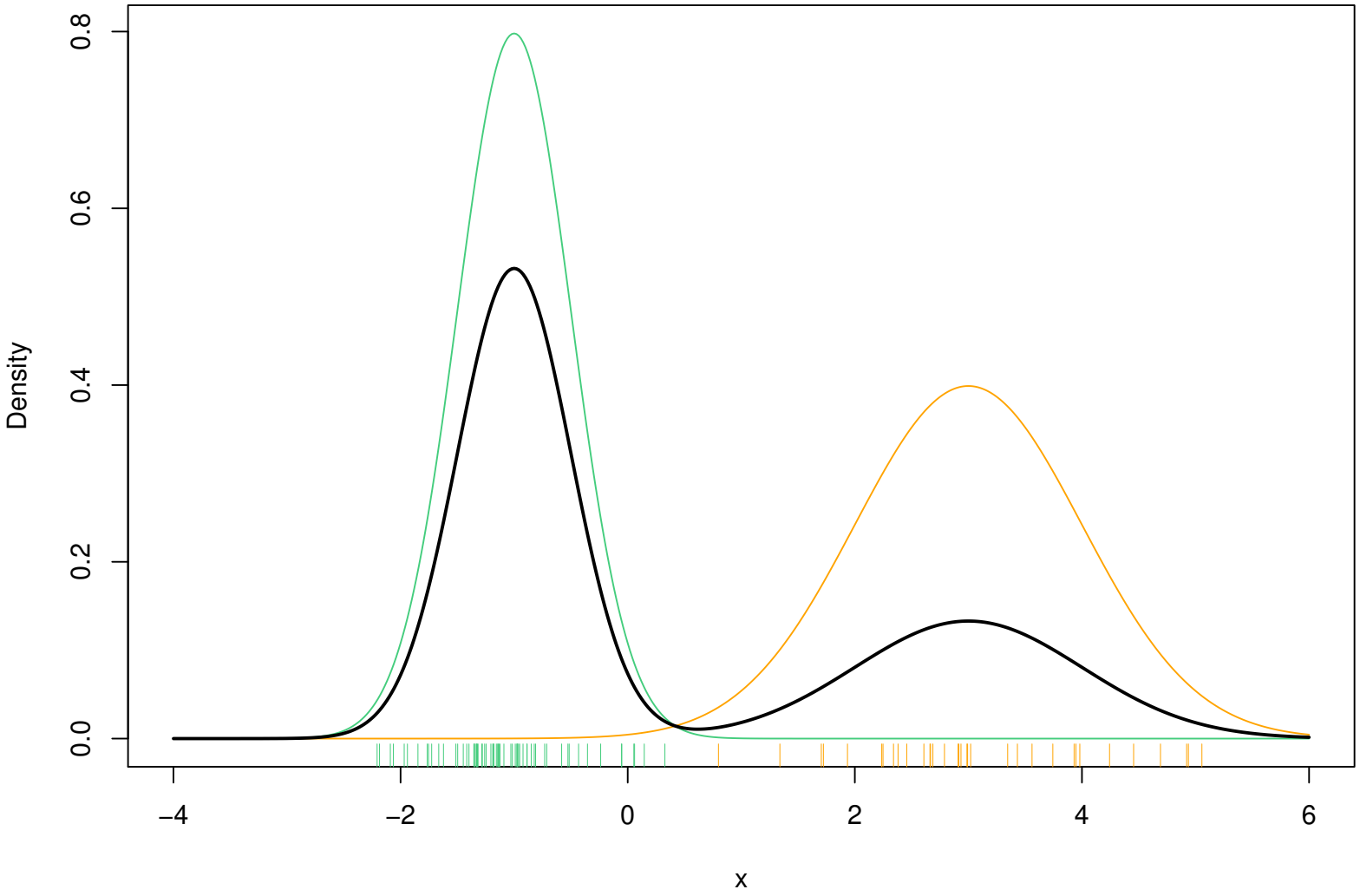


Figure 4: *One realization from a univariate Gaussian mixture with* $G = 2$, $f_1 = N(-1, 0.25)$, $f_2 = N(3, 1)$, $\tau_1 = 2/3$ *and* $\tau_2 = 1/3$.

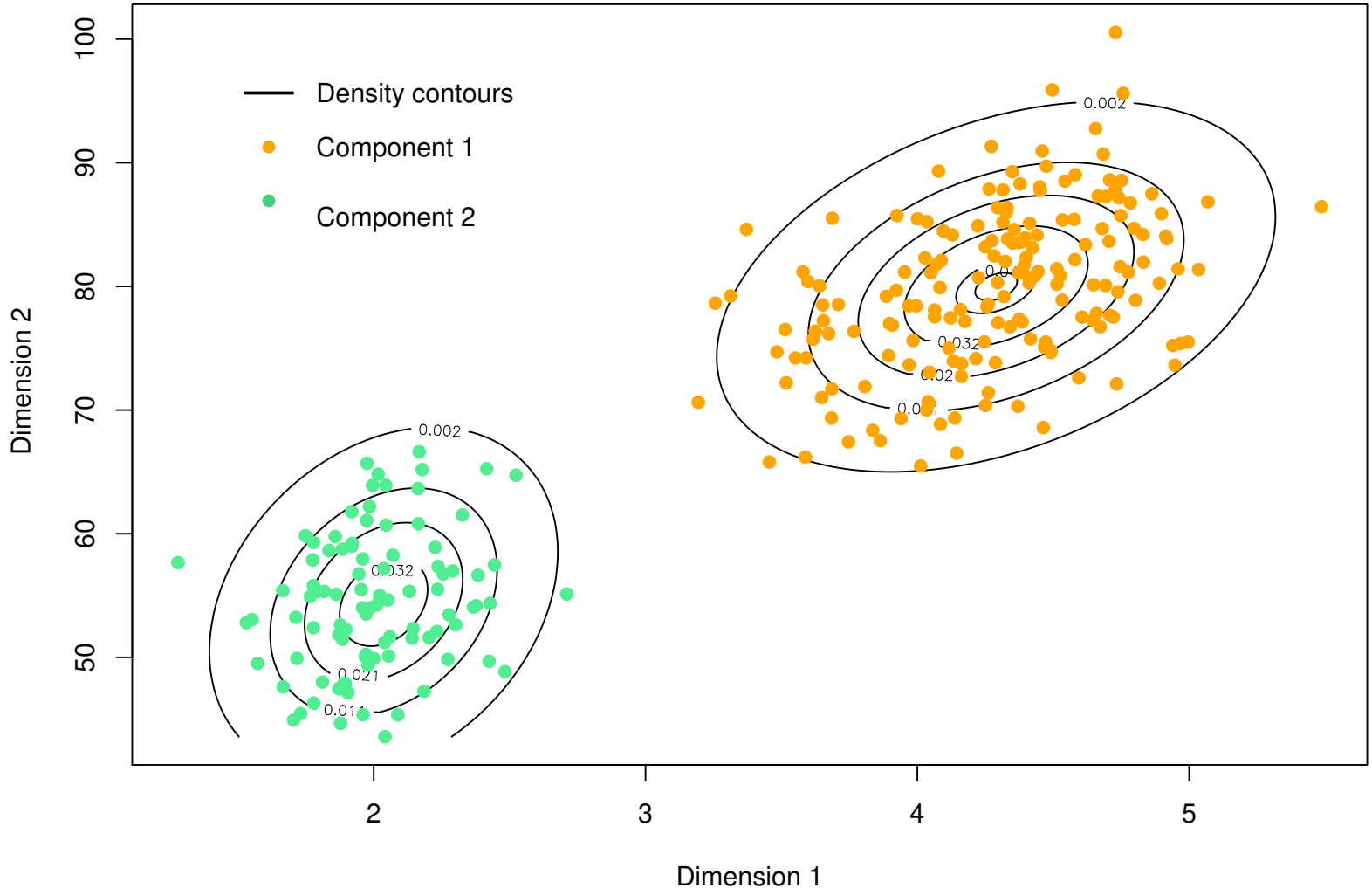


Figure 4: *One realization from a bivariate Gaussian mixture with* $G = 2$

Gaussian mixture models

- Gaussian mixture model is a specific case of mixture model where the distribution within each cluster is assumed to be Gaussian but with different parameters.
- It is by far the most used mixture model because:
  - analytically tractable (as we will see later on);
  - Gaussian mixtures are a kind of universal approximator of p.d.f.[2]

[2] but actually this is not specific to the Gaussian case...

The multivariate Gaussian distribution

Definition 2. The density of a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix Σ[3] is

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

The multivariate Gaussian distribution

Definition 2. The density of a multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix Σ[3] is

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

Figure 5: *Click on me to watch a movie!*

[3] Σ is a $p \times p$ posivite–(semi)definite matrix.

2. Inference

- Recall our mixture model $p(y) = \sum_{g=1}^{G} \tau_g f_g(y; \theta_g)$.
- Having observed iid realizations $\mathbf{y} = (y_1, \ldots, y_n)^\top$, the log-likelihood is

$$\ell(\theta; \mathbf{y}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \tau_g f_g(y_i; \theta_g) \right\}.$$

- Recall our mixture model $p(y) = \sum_{g=1}^{G} \tau_g f_g(y; \theta_g)$.
- Having observed iid realizations $\mathbf{y} = (y_1, \ldots, y_n)^\top$, the log-likelihood is

$$\ell(\theta; \mathbf{y}) = \sum_{i=1}^{n} \log \left\{ \sum_{g=1}^{G} \tau_g f_g(y_i; \theta_g) \right\}.$$

- This likelihood is nasty:
  - complexity in $O(nG)$ rather than the usual $O(n)$;
  - in general not convex hence local maxima;
  - for gaussian mixture, the likelihood is even not upper bounded![4]

☞ Sensible solutions are local maxima internal to the parameter space Θ.

[4] Set $\mu_1 = y_i$ for some $i \in \{1, \ldots, n\}$ and let $|\Sigma_1| \downarrow 0 \ldots$

- At first sight deriving the maximum likelihood estimator from such mixture models is hopeless.
- The main difficulty comes from the fact that we work under an unsupervised framework.
- But hidden we have the so-called latent variables $\mathbf{z} = (z_1, \ldots, z_n)^\top$ that gives the class label to each observation y_i.
- They are two equivalent representations for those z_i's:
  - $Z \sim$ Discrete(G), i.e., $\Pr(Z = g) = \tau_g$, $g = 1, \ldots, G$; with G modalities;
  - $Z \sim$ Multinomial$(1, G)$, i.e., $\Pr(Z = e_g) = \tau_g$ where e_g is the g-th vector of the canonical basis of $\mathbb{R}^G$.

Wouldn't it be nice

- Suppose for a while that we have access to the class label, $\mathbf{z} = (z_1, \ldots, z_n)^\top$, i.e., we are supervised.
- In this setting the log-likelihood is much more friendly since

$$\ell_c(\theta; \mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n} \{\log \tau_{z_i} + \log f_{z_i}(y_i; \theta_{z_i})\}$$
$$= \sum_{g=1}^{G} \left\{ n_g \log \tau_g + \sum_{i \in C_g} \log f_g(y_i; \theta_g) \right\},$$

where $C_g = \{i : z_i = g\}$ and $n_g = |C_g|$.

Wouldn't it be nice

- Suppose for a while that we have access to the class label, $\mathbf{z} = (z_1, \ldots, z_n)^\top$, i.e., we are supervised.
- In this setting the log-likelihood is much more friendly since

$$\ell_c(\theta; \mathbf{y}, \mathbf{z}) = \sum_{i=1}^{n} \{\log \tau_{z_i} + \log f_{z_i}(y_i; \theta_{z_i})\}$$
$$= \sum_{g=1}^{G} \left\{ n_g \log \tau_g + \sum_{i \in C_g} \log f_g(y_i; \theta_g) \right\},$$

where $C_g = \{i : z_i = g\}$ and $n_g = |C_g|$.

☞ In the statistical literature, $\ell_c(\theta; \mathbf{y}, \mathbf{z})$ is known as the **completed likelihood** since we assume having access to the (unknown) **latent variables** z_i identifying the class label.

- The likelihood, sometimes called observed likelihood, is clearly retrieved from this completed likelihood.
- More precisely, we have to marginalize w.r.t. the latent variable $\mathbf{z}$, i.e., since the z_i are discrete r.v.:

$$
\begin{aligned}
L(\theta; \mathbf{y}) &= \prod_{i=1}^{n} \sum_{g=1}^{G} f(y_i, g; \theta) \\
&= \prod_{i=1}^{n} \sum_{g=1}^{G} f(y_i \mid g; \theta) \Pr(Z = g) \\
&= \prod_{i=1}^{n} \sum_{g=1}^{G} f_g(y_i; \theta_g) \tau_g, \qquad \text{as expected.}
\end{aligned}
$$

The EM algorithm

- The Expectation Maximization algorithm is an iterative procedure that outputs a local maxima of the likelihood.
- As its name suggests, it is a 2 steps procedure:

  E Step Computation of the function

$$Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{\theta^{(t)}} \left\{ \ell_c(\theta; \mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y} = \mathbf{y} \right\},$$

  where $\theta^{(t)}$ is the value of the model parameter θ at the t–th iteration of the algorithm and $\mathbb{E}_\theta(\cdot)$ denotes expectation where the parameter of the model are set to θ.

  M Step Maximization of the above function w.r.t. θ

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} Q(\theta \mid \theta^{(t)}).$$

- These 2 steps are repeated until convergence, e.g., until the increase is less than 10^{-5}.

Why does it work? (very loose notations)

$$
\begin{aligned}
\ell(\theta; \mathbf{y}) &= \log \sum_z f(\mathbf{y}, \mathbf{z}; \theta) \mathrm{d}\mathbf{z} \\
&= \log \sum_z \frac{f(\mathbf{y}, \mathbf{z}; \theta)}{f(\mathbf{z} \mid \mathbf{y}; \theta^{(t)})} f(\mathbf{z} \mid \mathbf{y}; \theta^{(t)}) \mathrm{d}\mathbf{z} \\
&= \log \mathbb{E}_{\theta^{(t)}} \left\{ \frac{f(\mathbf{Y}, \mathbf{Z}; \theta)}{f(\mathbf{Z} \mid \mathbf{Y}; \theta^{(t)})} \mid \mathbf{Y} = \mathbf{y} \right\} \\
&\geq \mathbb{E}_{\theta^{(t)}} \left\{ \log \frac{f(\mathbf{Y}, \mathbf{Z}; \theta)}{f(\mathbf{Z} \mid \mathbf{Y}; \theta^{(t)})} \mid \mathbf{Y} = \mathbf{y} \right\}, \qquad \text{Jensen's inequality} \\
&= \mathbb{E}_{\theta^{(t)}} \left\{ \log f(\mathbf{Y}, \mathbf{Z}; \theta) \mid \mathbf{Y} = \mathbf{y} \right\} - \mathbb{E}_{\theta^{(t)}} \left\{ \log f(\mathbf{Z} \mid \mathbf{Y}; \theta^{(t)}) \mid \mathbf{Y} = \mathbf{y} \right\} \\
&= Q(\theta \mid \theta^{(t)}) + \mathsf{Shannon}(\mathbf{Z} \mid \mathbf{Y} = \mathbf{y}).
\end{aligned}
$$

Since we have equality when $\theta = \theta^{(t)}$ (why?), we have

$$\ell(\theta; \mathbf{y}) - \ell(\theta^{(t)}; \mathbf{y}) \geq Q(\theta \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}).$$

☞ Improving Q improves the log-likelihood at least as much!

As an aside: EM Variants

- In the EM algorithm, the M Step $\theta^{(t+1)} = \underset{\theta \in \Theta}{\arg\max}\, Q(\theta \mid \theta^{(t)})$ clearly improves Q and hence the log-likelihood.

As an aside: EM Variants

- In the EM algorithm, the M Step $\theta^{(t+1)} = \arg\max_{\theta \in \Theta} Q(\theta \mid \theta^{(t)})$ clearly improves Q and hence the log-likelihood.
- However, from our proof, it is sufficient to set $\theta^{(t+1)}$ such that

$$Q(\theta^{(t+1)} \mid \theta^{(t)}) > Q(\theta^{(t)} \mid \theta^{(t)}),$$

this is the **GEM** algorithm.

- We can skip the expectation and work with the completed likelihood, i.e., imputing the latent variable z_i:

SEM Sample from $\mathbf{Z} \mid \mathbf{Y} = \mathbf{y}$ with parameter $\theta^{(t)}$ (outputs a Markov chain).
CEM Set $\mathbf{z}^{(t+1)} = \arg\max_{\mathbf{z}} L(\theta^{(t)}; \mathbf{y}, \mathbf{z})$,

and set

$$\theta^{(t+1)} = \arg\max_{\theta \in \Theta} L(\theta; \mathbf{y}, \mathbf{z}^{(t+1)}).$$

The E Step

- We need to compute $Q(\theta \mid \theta^{(t)}) = \mathbb{E}_{\theta^{(t)}}\{\ell_c(\theta; \mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y} = \mathbf{y}\}$.
- Remember that Z is a discrete r.v. with support $\{1, \ldots, G\}$, hence

$$Q(\theta \mid \theta^{(t)}) = \sum_{g=1}^{G}\sum_{i=1}^{n} \log\{\tau_g f(y_i, g; \theta)\} \Pr_{\theta^{(t)}}(Z = g \mid Y = y_i)$$
$$= \sum_{g=1}^{G}\sum_{i=1}^{n} \log\{\tau_g f_g(y_i; \theta_g)\} t_g^{(t)}(y_i),$$

where $t_g^{(t)}(y_i) = \Pr_{\theta^{(t)}}(Z = g \mid Y = y_i)$ is the conditional probability of membership to class g.

- Now for all $y \in E$ and $g \in \{1, \ldots, G\}$, we have

$$t_g^{(t)}(y) = \frac{\Pr_{\theta^{(t)}}(Y = y \mid Z = g)\Pr(Z = g)}{p(y)} = \frac{f_g(y; \theta_g^{(t)})\tau_g^{(t)}}{C(y)},$$

with normalizing constant $C(y) = \sum_{g=1}^{G} \tau_g^{(t)} f_g(y; \theta_g^{(t)})$.

The M Step

- Recall that the M Step consists in

$$\{(\tau_g^{(t+1)}, \theta_g^{(t+1)})\}_{g=1,\dots,G} = \arg\max_{(\tau_g,\theta_g)} \sum_{i=1}^{n}\sum_{g=1}^{G} \log\{\tau_g f_g(y_i;\theta_g)\} t_g^{(t)}(y_i).$$

- To update τ_g, it amounts to solve for

$$\begin{cases} \sum_{i=1}^n \frac{t_g^{(t)}(y_i)}{\tau_g} + \lambda = 0 \\ \sum_{g=1}^G \tau_g = 1 \end{cases} \iff \begin{cases} \tau_g = -\frac{\sum_{i=1}^n t_g^{(t)}(y_i)}{\lambda} \\ \lambda = -\sum_{g=1}^G \sum_{i=1}^n t_g^{(t)}(y_i) = -n \end{cases} \quad \text{(why?)}$$

$$\boxed{\implies \tau_g^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} t_g^{(t)}(y_i).}$$

The M Step

- Recall that the M Step consists in

$$\{(\tau_g^{(t+1)}, \theta_g^{(t+1)})\}_{g=1,\dots,G} = \underset{(\tau_g,\theta_g)}{\arg\max} \sum_{i=1}^{n}\sum_{g=1}^{G} \log\{\tau_g f_g(y_i;\theta_g)\} t_g^{(t)}(y_i).$$

- To update τ_g, it amounts to solve for

$$\begin{cases} \sum_{i=1}^n \frac{t_g^{(t)}(y_i)}{\tau_g} + \lambda = 0 \\ \sum_{g=1}^G \tau_g = 1 \end{cases} \iff \begin{cases} \tau_g = -\frac{\sum_{i=1}^n t_g^{(t)}(y_i)}{\lambda} \\ \lambda = -\sum_{g=1}^G \sum_{i=1}^n t_g^{(t)}(y_i) = -n \end{cases} \text{ (why?)}$$

$$\boxed{\implies \tau_g^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} t_g^{(t)}(y_i).}$$

☞ Updating the class probabilities is indenpendent of the parametric choices made on the p.d.f. f_g.

Gaussian mixtures

- Updating each θ_g, $g = 1, \ldots, G$, depends on the choices made on f_g.
- Although other choices than the multivariate Gaussian distribution are possible (and useful!), we will restrict our attention to this specific situation.
- It turns out that explicit formulas are available for Gaussian mixtures, i.e., when

$$\begin{aligned} f_g(y; \theta_g) &= \varphi(y; \mu_g, \Sigma_g) \\ &= (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left\{ -\frac{(y - \mu_g)^\top \Sigma^{-1} (y - \mu_g)}{2} \right\}, \; g = 1, \ldots, G. \end{aligned}$$

- The next slides derive the updating schemes for μ_g and Σ_g.

M Step: Updating μ_g

- It is not difficult to show that

$$\nabla_{\mu_g} \varphi(y; \mu_g, \Sigma_g) = -\Sigma_g^{-1}(y - \mu_g)\varphi(y; \mu_g, \Sigma_g),$$

- To update μ_g, we thus have to solve for

$$\sum_{i=1}^{n} t_g^{(t)}(y_i) \frac{-\Sigma_g^{-1}(y_i - \mu_g)\varphi(y_i; \mu_g, \Sigma_g)}{\varphi(y_i; \mu_g, \Sigma_g)} = 0$$

$$\Longrightarrow \mu_g^{(t+1)} = \frac{\sum_{i=1}^{n} t_g^{(t)}(y_i) y_i}{\sum_{i=1}^{n} t_g^{(t)}(y_i)}.$$

M Step: Updating μ_g

- It is not difficult to show that

$$\nabla_{\mu_g} \varphi(y; \mu_g, \Sigma_g) = -\Sigma_g^{-1}(y - \mu_g)\varphi(y; \mu_g, \Sigma_g),$$

- To update μ_g, we thus have to solve for

$$\sum_{i=1}^{n} t_g^{(t)}(y_i) \frac{-\Sigma_g^{-1}(y_i - \mu_g)\varphi(y_i; \mu_g, \Sigma_g)}{\varphi(y_i; \mu_g, \Sigma_g)} = 0$$

$$\Longrightarrow \mu_g^{(t+1)} = \frac{\sum_{i=1}^{n} t_g^{(t)}(y_i) y_i}{\sum_{i=1}^{n} t_g^{(t)}(y_i)}.$$

☞ It is a weighted mean of the observations y_i with weights proportional to $t_g(y_i) = \Pr_{\theta^{(t)}}(Z = g \mid Y = y_i)$.

M Step: Updating Σ_g

- Recall that $\frac{\partial}{\partial A} x^\top A^{-1} x = -A^{-\top} x x^\top A^{-\top}$ and $\frac{\partial}{\partial A}|A| = |A| A^{-\top}$, so

$$\frac{\partial}{\partial \Sigma} \varphi(y; \mu, \Sigma) = \frac{1}{2} \left\{ -\Sigma^{-1} + \Sigma^{-1}(y-\mu)(y-\mu)^\top \Sigma^{-1} \right\} \varphi(y; \mu, \Sigma)$$

- We thus have to solve for

$$\sum_{i=1}^{n} t_g^{(t)}(y_i) \frac{1}{2} \left\{ -\Sigma_g^{-1} + \Sigma_g^{-1}(y_i - \mu_g^{(t+1)})(y_i - \mu_g^{(t+1)})^\top \Sigma_g^{-1} \right\} = 0$$

$$\iff \sum_{i=1}^{n} t_g^{(t)}(y_i) \left\{ -\mathsf{Id} + (y_i - \mu_g^{(t+1)})(y_i - \mu_g^{(t+1)})^\top \Sigma_g^{-1} \right\} = 0$$

$$\implies \Sigma_g^{(t+1)} = \frac{\sum_{i=1}^{n} t_g^{(t)}(y_i)(y_i - \mu_g^{(t+1)})(y_i - \mu_g^{(t+1)})^\top}{\sum_{i=1}^{n} t_g^{(t)}(y_i)}.$$

M Step: Updating Σ_g

- Recall that $\frac{\partial}{\partial A} x^\top A^{-1} x = -A^{-\top} x x^\top A^{-\top}$ and $\frac{\partial}{\partial A}|A| = |A| A^{-\top}$, so

$$\frac{\partial}{\partial \Sigma} \varphi(y; \mu, \Sigma) = \frac{1}{2} \left\{ -\Sigma^{-1} + \Sigma^{-1}(y-\mu)(y-\mu)^\top \Sigma^{-1} \right\} \varphi(y; \mu, \Sigma)$$

- We thus have to solve for

$$\sum_{i=1}^{n} t_g^{(t)}(y_i) \frac{1}{2} \left\{ -\Sigma_g^{-1} + \Sigma_g^{-1}(y_i - \mu_g^{(t+1)})(y_i - \mu_g^{(t+1)})^\top \Sigma_g^{-1} \right\} = 0$$

$$\iff \sum_{i=1}^{n} t_g^{(t)}(y_i) \left\{ -\mathsf{Id} + (y_i - \mu_g^{(t+1)})(y_i - \mu_g^{(t+1)})^\top \Sigma_g^{-1} \right\} = 0$$

$$\boxed{\implies \Sigma_g^{(t+1)} = \frac{\sum_{i=1}^{n} t_g^{(t)}(y_i)(y_i - \mu_g^{(t+1)})(y_i - \mu_g^{(t+1)})^\top}{\sum_{i=1}^{n} t_g^{(t)}(y_i)}.}$$

☞ This is a weighted empirical covariance matrix with known mean $\mu_g^{(t+1)}$.

EM for Gaussian mixtures: Sum up

Algorithm 1: EM algorithm for (unconstrained) Gaussian mixtures.

while *not converged* **do**

`/* E Step: */`

Compute the conditional probability of membership:

$$t_g(y_i) \propto \varphi_g(y_i; \mu_g, \Sigma_g)\tau_g, \qquad i = 1, \ldots, n, \quad g = 1, \ldots, G$$

`/* M Step: */`

Update the probability of membership:

$$\tau_g \leftarrow \frac{1}{n}\sum_{i=1}^{n} t_g(y_i), \qquad g = 1, \ldots, G$$

Update the Gaussian means:

$$\mu_g \leftarrow \frac{\sum_{i=1}^{n} t_g(y_i) y_i}{\sum_{i=1}^{n} t_g(y_i)}, \qquad g = 1, \ldots, G$$

Update the Gaussian covariances:

$$\Sigma_g \leftarrow \frac{\sum_{i=1}^{n} t_g(y_i)(y_i - \mu_g)(y_i - \mu_g)^\top}{\sum_{i=1}^{n} t_g(y_i)}, \qquad g = 1, \ldots, G$$

Check for convergence;

Return τ_g, μ_g and Σ_g, $g = 1, \ldots, G$;

Convergence of the EM

Figure 6: *Click on me to watch a movie!*

Initialization

- Remember that the log-likelihood for mixture model is nasty
- As a side effect, estimates based on the EM algorithm are strongly impacted on how the latter was initialized.

- While you were eating  , I initialized the EM algorithm for the `faithful` dataset using a "fair" random partition, i.e, $Z_i \overset{\text{iid}}{\sim} \text{Ber}(1/2)$.
- In general, such an initialization will fail![5]

Initialization

- Remember that the log-likelihood for mixture model is nasty
- As a side effect, estimates based on the EM algorithm are strongly impacted on how the latter was initialized.

- While you were eating  , I initialized the EM algorithm for the `faithful` dataset using a "fair" random partition, i.e, $Z_i \overset{\text{iid}}{\sim} \text{Ber}(1/2)$.
- In general, such an initialization will fail![5]
- Two popular strategies to mitigate this issue are:
  - the small EM approach;
  - the use of the classification likelihood.

[5] If you really insist on initializing using random state, you would rather sample the initial parameter ;-)

The small EM initialization

Algorithm 2: Small EM initialization.

input : Number of small EM trials M and tolerance value ε (typically $M = 5$ and $\varepsilon = 0.01$)

output: An initial state θ_{init}

bestLogLik $\leftarrow -\infty$
for $i = 1, \ldots, M$ **do**
    Simulate a random set of parameter values $\theta^{(1)}$;
    Run a "short" EM algorithm with intial state $\theta^{(1)}$ until

$$\frac{\ell(\theta^{(t)}; \mathbf{y}) - \ell(\theta^{(t-1)}; \mathbf{y})}{\ell(\theta^{(t)}; \mathbf{y}) - \ell(\theta^{(1)}; \mathbf{y})} < \varepsilon$$

    /* Below T denotes the last iteration of the EM algo */
    if $\ell(\theta^{(T)}; \mathbf{y}) >$ *bestLogLik* **then**
        bestLogLik $\leftarrow \ell(\theta^{(T)}; \mathbf{y})$;
        $\theta_{\text{init}} \leftarrow \theta^{(T)}$;
Return θ_{init};

Initialization using the classification likelihood

- The idea here consists in maximizing the so–called classification likelihood

$$L_{\text{Classif}}(\theta, z; y) = \prod_{i=1}^{n} f_{z_i}(y_i; \theta_{z_i}).$$

- This likelihood differs from both the (mixture) likelihood and the completed likelihood

$$L(\theta; y) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g f(y_i; \theta_g), \qquad L_C(\theta; y, z) = \prod_{i=1}^{n} \tau_{z_i} f(y_i; \theta_{z_i}).$$

Initialization using the classification likelihood

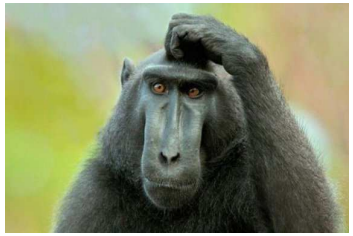
- The idea here consists in maximizing the so–called classification likelihood

$$L_{\text{Classif}}(\theta, z; y) = \prod_{i=1}^{n} f_{z_i}(y_i; \theta_{z_i}).$$

- This likelihood differs from both the (mixture) likelihood and the completed likelihood

$$L(\theta; y) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g f(y_i; \theta_g), \qquad L_C(\theta; y, z) = \prod_{i=1}^{n} \tau_{z_i} f(y_i; \theta_{z_i}).$$

- Now you may scratch your head and wonder why not using this classification likelihood directly?



Initialization using the classification likelihood

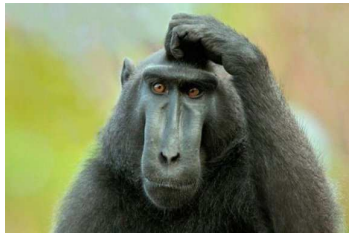
- The idea here consists in maximizing the so–called classification likelihood

$$L_{\text{Classif}}(\theta, z; y) = \prod_{i=1}^{n} f_{z_i}(y_i; \theta_{z_i}).$$

- This likelihood differs from both the (mixture) likelihood and the completed likelihood

$$L(\theta; y) = \prod_{i=1}^{n} \sum_{g=1}^{G} \tau_g f(y_i; \theta_g), \qquad L_C(\theta; y, z) = \prod_{i=1}^{n} \tau_{z_i} f(y_i; \theta_{z_i}).$$

- Now you may scratch your head and wonder why not using this classification likelihood directly?
- One reason is that it yields inconsistent estimator.
- But since it is fast to compute, it makes sense to use it as initialization.



Predicting the cluster membership

- Suppose you have a fitted mixture model[6] with parameter estimate $\hat{\theta} = \{(\hat{\tau}_g, \hat{\theta}_g) : g = 1, \ldots, G\}$.
- Let $y \in E$, the conditional membership probabilities

$$t_g(y) = \Pr_{\hat{\theta}}(Z = g \mid Y = y) \propto \hat{\tau}_g \varphi_g(y; \hat{\mu}_g, \hat{\Sigma}_g), \qquad g = 1, \ldots, G,$$

  can be used to identify the mixture component y belong to.
- More precisely, from the optimal Bayes rule, we set

$$z_*(y) = \arg\max_{g \in \{1,\ldots,G\}} t_g(y),$$

  such rule is known as maximum a posteriori.

[6] and that we are happy with it!

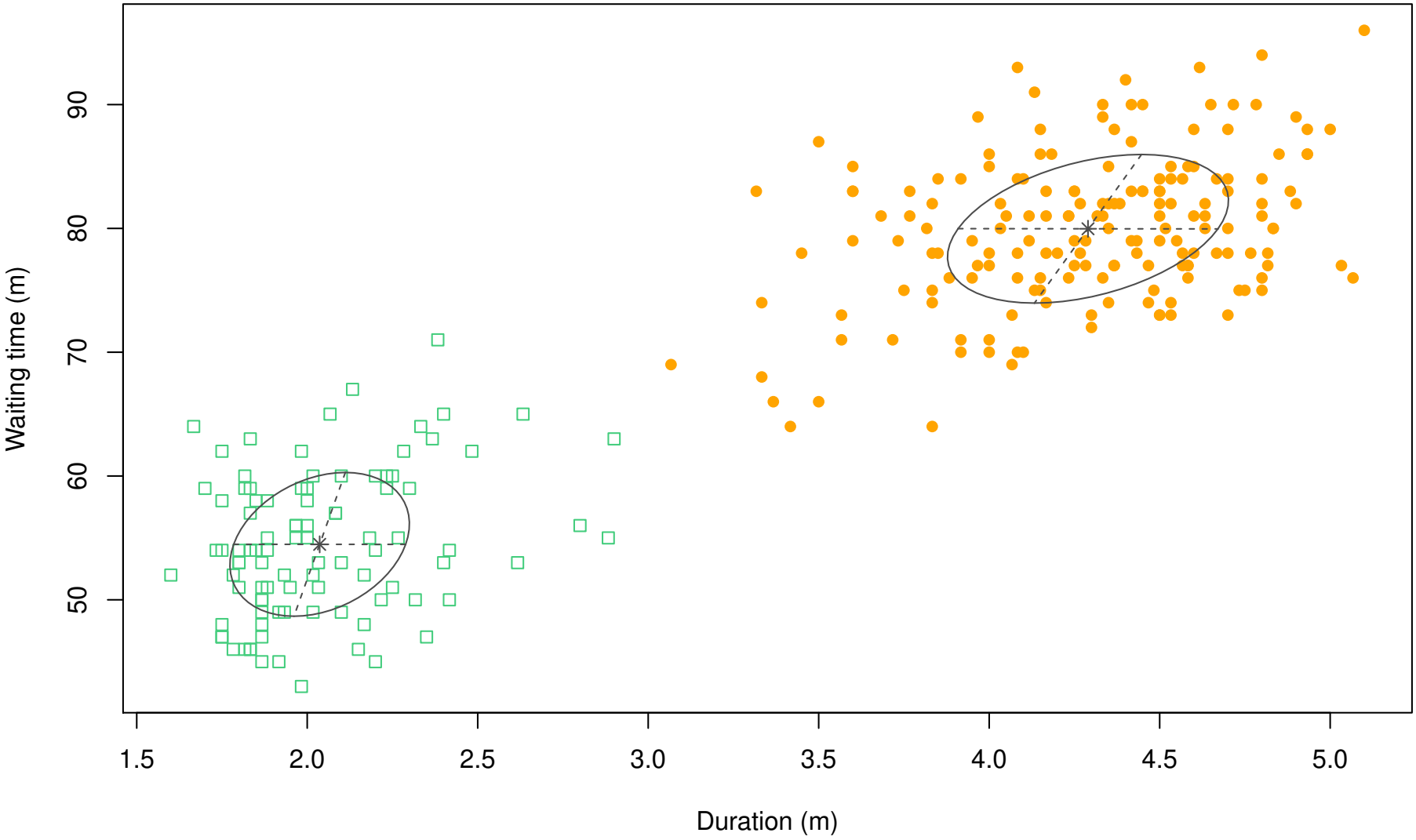


Figure 7: *Mixture component attribution for the model "VVV" with $G = 2$ fitted on the old faithful geyser dataset.*

Uncertainties in classification

- One main asset in model based clustering is that, due to the use of a statistical model, we often have direct access to uncertainties related to the class attribution.
- Again from our fitted mixture model, we define the uncertainty about the class attribution for $y \in E$ as

$$\text{Uncertainty}(y) = 1 - \max_{g=1,\ldots,G} t_g(y) \in \left[0, \frac{1}{G}\right],$$

ranging from 100% sure attribution up to "coin tossing".

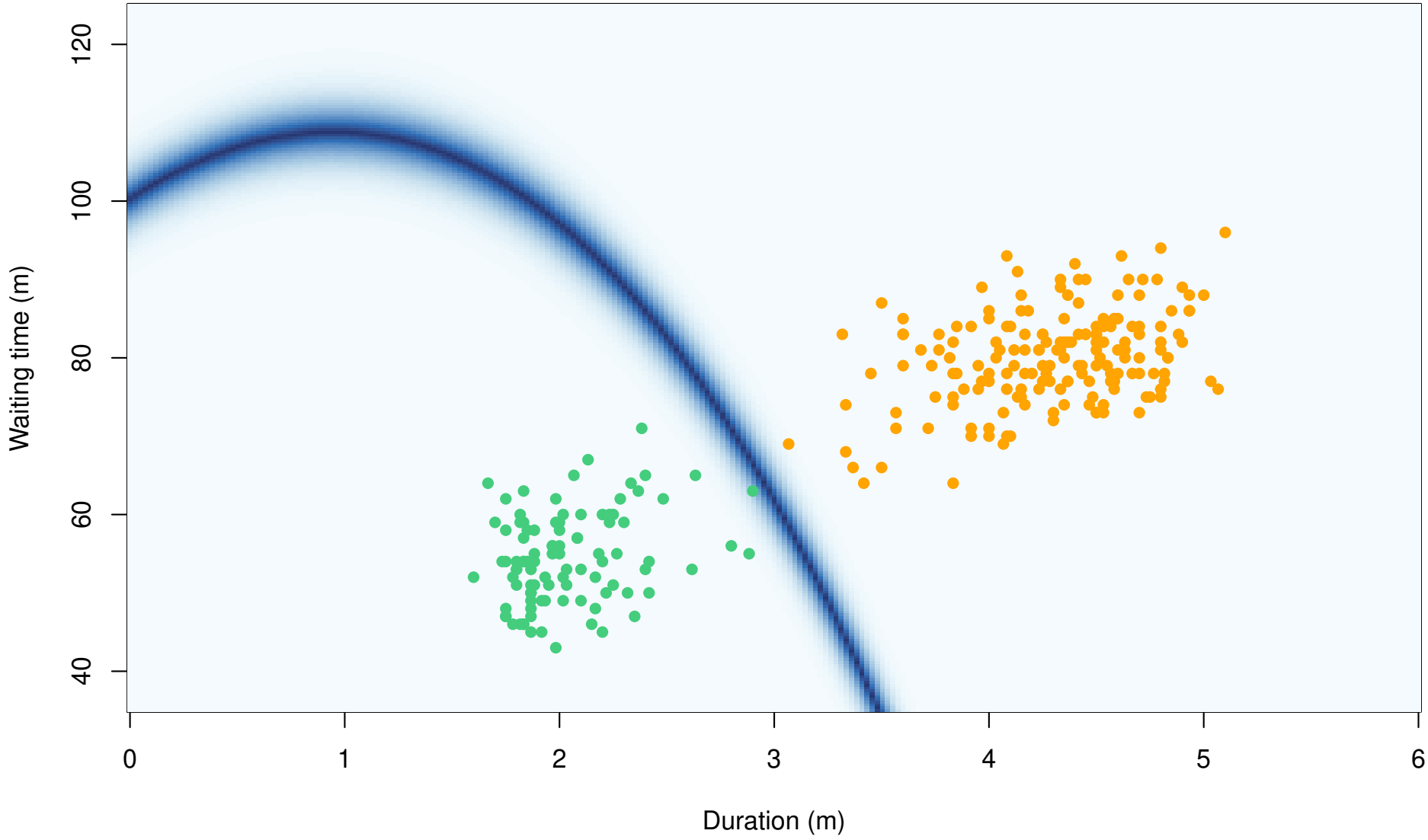


Figure 8: *Uncertainties values for the old faithful geyser dataset (with $G = 2$).*

3. Parsimonious models

The Gaussian mixture model

$$p(y) = \sum_{g=1}^{G} \tau_g \varphi(y_i; \theta_g), \qquad \theta_g = \{\mu_g, \Sigma_g\}.$$

- Number of parameters: $(G-1) + G \times d + G \times d(d+1)/2$
- High-dimensional data lead to estimation problem $\Rightarrow$ parsimonious models

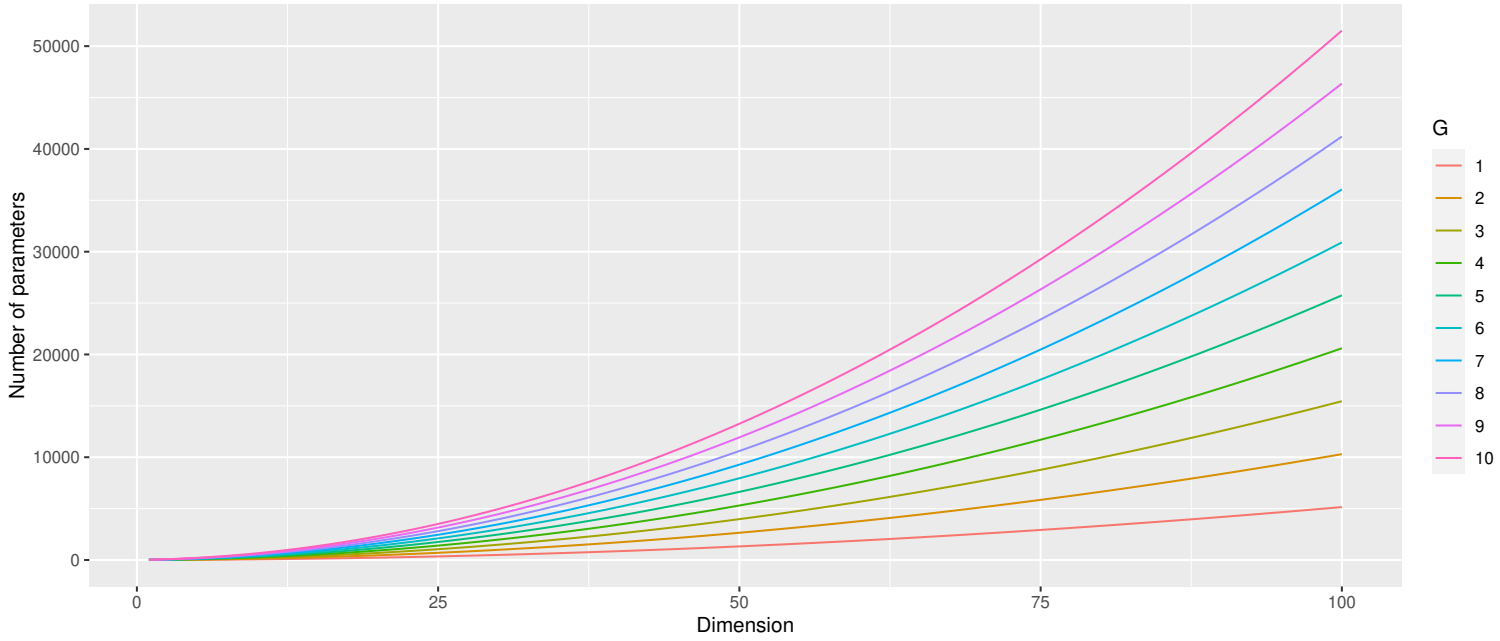


Figure 9: *Evolution of the number of parameters for the Gaussian mixture model as the number of clusters G and the dimension d increase.*

Volume–Shape–Orientation decomposition

- Since Σ_g is positive definite we decompose it as

$$\Sigma_g = \lambda_g Q_g A_g Q_g^\top,$$

where

- Q_g is the matrix of eigenvectors of Σ_g;
- A_g is a diagonal matrix proportional that composed of the eigenvalues of Σ_g;
- λ_g is the associated constant of proportionality.

Volume–Shape–Orientation decomposition

- Since Σ_g is positive definite we decompose it as

$$\Sigma_g = \lambda_g Q_g A_g Q_g^\top,$$

where

  - Q_g is the matrix of eigenvectors of Σ_g;
  - A_g is a diagonal matrix proportional that composed of the eigenvalues of Σ_g;
  - λ_g is the associated constant of proportionality.

- The matrix of eigenvectors Q_g controls the orientation in $\mathbb{R}^d$;
- The diagonal matrix A_g determines its shape;
- The constant of proportionality λ_g drives its spread / volume.

Remark. It is common practice to constrain A_g such that $|A_g| = 1$.

Zoology of Gaussian mixture models

- Based on the above Volume–Shape–Orientation decomposition, a wide range of structure can be obtained.
- More precisely it starts with the least flexible model for which

$$\Sigma_g = \lambda \mathsf{Id}_d, \qquad g \in \{1, \ldots, d\},$$

up to the most flexible one for which

$$\lambda_g \neq \lambda_\ell, \qquad Q_g \neq Q_\ell, \qquad A_g = A_\ell, \qquad g \neq \ell.$$

Zoology of Gaussian mixture models (2)

Table 1: *Zoology and model identifier in* `mclust`.

Identifier	Model	Distribution	Volume	Shape	Orientation	# covariance parameters
E		Univariate	Equal			1
V		Univariate	Variable			G
EII	λId	Spherical	Equal	Equal	NA	1
VII	λ_gId	Spherical	Variable	Equal	NA	G
EEI	λA	Diagonal	Equal	Equal	Axis–aligned	d
VEI	$\lambda_g A$	Diagonal	Variable	Equal	Axis–aligned	$G + (d-1)$
EVI	λA_g	Diagonal	Equal	Variable	Axis–aligned	$1 + G(d-1)$
VVI	$\lambda_g A_g$	Diagonal	Variable	Variable	Axis–aligned	Gd
EEE	Σ	Ellipsoidal	Equal	Equal	Equal	$d(d+1)/2$
VEE	$\lambda_g DAD^\top$	Ellipsoidal	Variable	Equal	Equal	$G + (d+2)(d-1)/2$
EVE	$\lambda DA_g D^\top$	Ellipsoidal	Equal	Variable	Equal	$1 + (d+2G)(d-1)/2$
EEV	$\lambda D_g A D_g^\top$	Ellipsoidal	Equal	Equal	Variable	$1 + (d-1) + G\{d(d-1)/2\}$
VVE	$\lambda_g D A_g D^\top$	Ellipsoidal	Variable	Variable	Equal	$G + (d+2G)(d-1)/2$
EVV	$\lambda D_g A_g D_g^\top$	Ellipsoidal	Equal	Variable	Variable	$1 + G(d+2)(d-1)/2$
VEV	$\lambda_g D_g A D_g^\top$	Ellipsoidal	Variable	Equal	Variable	$G + (d-1) + G(\{d(d-1)/2\}$
VVV	Σ_g	Ellipsoidal	Variable	Variable	Variable	$G\{d(d+1)/2\}$

Zoology of Gaussian mixture models (2)

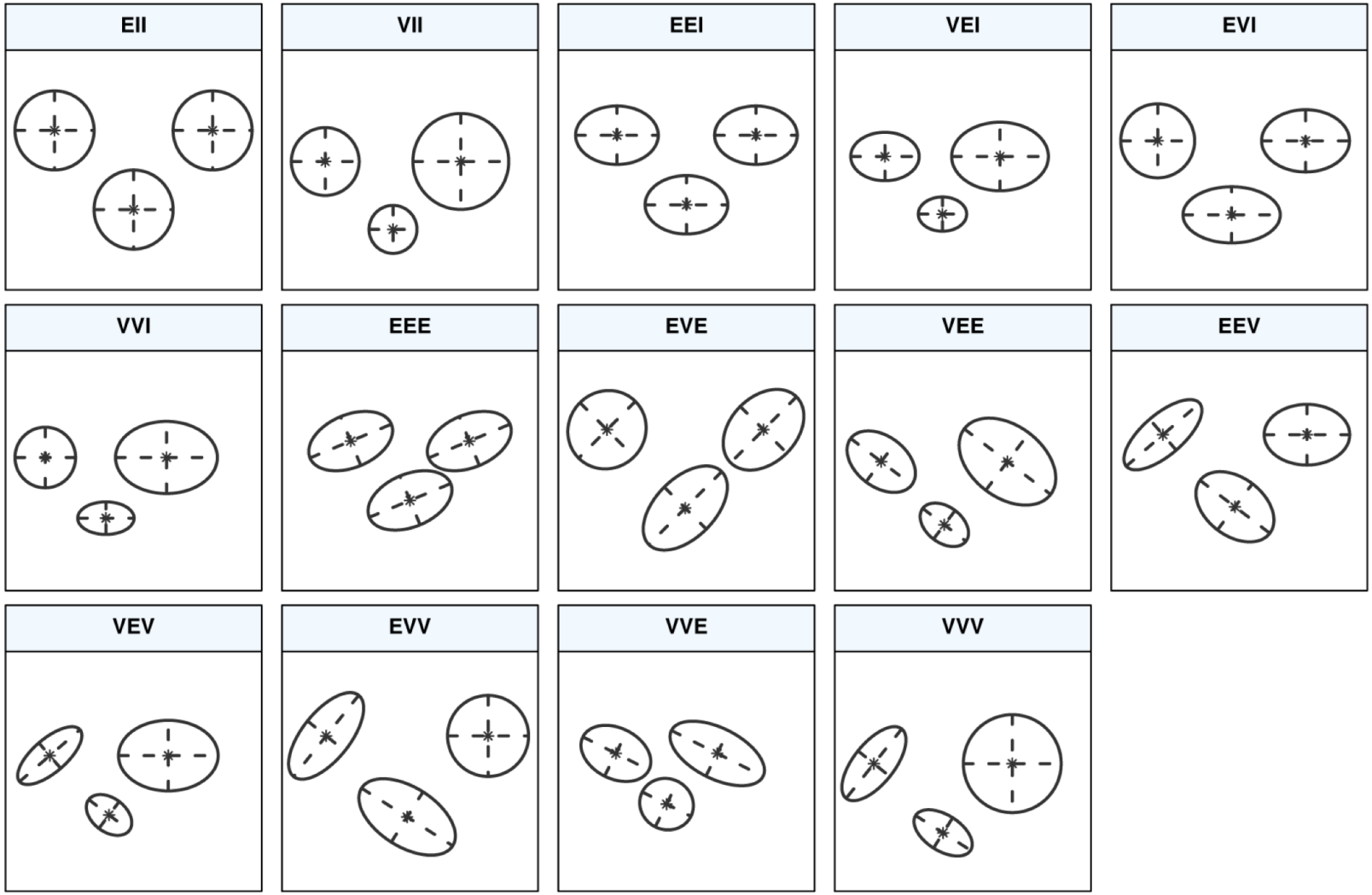


Figure 10: *Zoology of the different VSO assumption for Gaussian mixture models. [Taken from Strucca et al., 2016]*

- Recall the decomposition

$$\Sigma_g = \lambda_g Q_g A_g Q_g^\top,$$

  where $\lambda_g \in \mathbb{R}_+$, Q_g is a $d \times d$ matrix and A_g is a diagonal matrix.
- We see that there is
  - more parameter to specify the shape than the volume
  - and (much!) more parameters to specify the orientation

☞ Huge gain in parsimony when the orientation is constant across mixture components

4. Model selection

What is model selection?

- For model based clustering it is a stage where you want to find the appropriate number of mixture components G and/or the strucutre of the model;
- Since within an unsupervised setting, we cannot evaluate the misclassification error.
- Hopefully there is a long history for statistical model selection (much (much!) older than cross-validation ;-))
- Essentially we will use penalized likelihood criterion

Model selection in clustering assumes that you use the same variables / features, i.e., don't do feature selection then clustering then model selection: that's unfair !

Model posterior probability

- Suppose you have several competitive (fitted) models $M_1, \ldots, M_K$.
- Each model is assigned a prior probability $\pi(M_k)$, often $\pi(M_k) = 1/K$.
- The posterior probability for model M_k is thus

$$\begin{aligned}\pi(M_k \mid \mathbf{y}) &\propto \pi(\mathbf{y}; M_k)\pi(M_k) \\ &\propto \underbrace{\int L(\mathbf{y}; \theta_{M_k}, M_k)\pi(\theta_{M_k} \mid M_k)\mathrm{d}\theta_{M_k}}_{\text{known as the integrated likelihood}} \pi(M_k),\end{aligned}$$

  where $\pi(\theta_{M_k} \mid M_k)$ is the prior distribution for θ_{M_k}.
- It makes senses to opt for the model

$$M_* = \underset{M \in \{M_1, \ldots, M_k\}}{\arg\max} \pi(M \mid \mathbf{y}),$$

  i.e., the most likely a posteriori.

Model posterior probability

- Suppose you have several competitive (fitted) models $M_1, \ldots, M_K$.
- Each model is assigned a prior probability $\pi(M_k)$, often $\pi(M_k) = 1/K$.
- The posterior probability for model M_k is thus

$$\begin{aligned}\pi(M_k \mid \mathbf{y}) &\propto \pi(\mathbf{y}; M_k)\pi(M_k) \\ &\propto \underbrace{\int L(\mathbf{y}; \theta_{M_k}, M_k)\pi(\theta_{M_k} \mid M_k)\mathrm{d}\theta_{M_k}}_{\text{known as the integrated likelihood}} \pi(M_k),\end{aligned}$$

  where $\pi(\theta_{M_k} \mid M_k)$ is the prior distribution for θ_{M_k}.
- It makes senses to opt for the model

$$M_* = \underset{M \in \{M_1, \ldots, M_k\}}{\arg\max} \pi(M \mid \mathbf{y}),$$

  i.e., the most likely a posteriori.

☞ If $\pi(M_k) \propto 1$, it simplifies to maximizing the integrated likelihood.

Looking for BIC

- Typically the integrated likelihood

$$\int L(\mathbf{y}; \theta_{M_k}, M_k) \pi(\theta_{M_k} \mid M_k) \mathrm{d}\theta_{M_k}$$

  has no closed form and we need to approximate it numerically.
- A Laplace approximation on $L(\mathbf{y}; \theta_{M_k}, M_k)$[7] gives:

$$\pi(\mathbf{y} \mid M_k) \approx L(\mathbf{y} \mid \hat{\theta}_{M_k}, M_k)(2\pi)^{k/2} |nI(\hat{\theta}_{M_k})|^{-1/2} \pi(\hat{\theta}_{M_k} \mid M_k).$$

- We get the so-called Bayesian Information Criterion (BIC)

$$\pi(M_k \mid \mathbf{y}) \propto \pi(\mathbf{y} \mid M_k)\pi(M_k) \approx \exp\left\{-\frac{\mathrm{BIC}(M_k)}{2}\right\} \pi(M_k),$$

  where $\mathrm{BIC}(M_k) = -2 \log L(\mathbf{y} \mid \hat{\theta}_{M_k}, M_k) + k \log n$ and $k = |M_k|$.

Looking for BIC

- Typically the integrated likelihood

$$\int L(\mathbf{y}; \theta_{M_k}, M_k)\pi(\theta_{M_k} \mid M_k)\mathrm{d}\theta_{M_k}$$

  has no closed form and we need to approximate it numerically.
- A Laplace approximation on $L(\mathbf{y}; \theta_{M_k}, M_k)$[7] gives:

$$\pi(\mathbf{y} \mid M_k) \approx L(\mathbf{y} \mid \hat{\theta}_{M_k}, M_k)(2\pi)^{k/2}|nI(\hat{\theta}_{M_k})|^{-1/2}\pi(\hat{\theta}_{M_k} \mid M_k).$$

- We get the so-called Bayesian Information Criterion (BIC)

$$\pi(M_k \mid \mathbf{y}) \propto \pi(\mathbf{y} \mid M_k)\pi(M_k) \approx \exp\left\{-\frac{\mathrm{BIC}(M_k)}{2}\right\}\pi(M_k),$$

  where $\mathrm{BIC}(M_k) = -2\log L(\mathbf{y} \mid \hat{\theta}_{M_k}, M_k) + k\log n$ and $k = |M_k|$.

☞ We would opt the model that minimizes the BIC!

[7] and using the crude approximation $\pi(\theta_k \mid M_k) \approx \pi(\hat{\theta}_{M_k})$

clusters vs. mixture components

- The BIC can be used to select the number of mixture components G but, in general, G is different from the number of clusters.

clusters vs. mixture components

- The BIC can be used to select the number of mixture components G but, in general, G is different from the number of clusters.

clusters vs. mixture components

- The BIC can be used to select the number of mixture components G but, in general, G is different from the number of clusters.
- Think about a non Gaussian cluster: you may use several Gaussian components as an approximation!
- If identifying the right number of clusters rather than the best fitted mixture model is your goal, then use the integrated completed likelihood

$$\pi(M_k \mid \mathbf{y}, \mathbf{z}) \propto \int L_C(\mathbf{y}, \mathbf{z} \mid \theta_{M_k}, M_k)\pi(\theta_{M_k} \mid M_k)\mathrm{d}\theta_{M_k}\pi(M_k)$$

- Doing the same steps as for the derivation of the BIC, we should get

$$\text{ICL}(M_k) = -2 \log L_C(\mathbf{y}, \mathbf{z}_*; \hat{\theta}_{M_k}, M_k) + k \log n, \qquad k = |M_k|,$$

where $\mathbf{z}_*$ is the maximum a posteriori, i.e., $z_{*,i} = \underset{g \in \{1,\ldots,G\}}{\arg\max}\, t_g(y_i)$.

BIC or ICL?

- There is a strong connection between these two quantities
- More precisely it is not difficult to show that

$$ICL(M_k) = BIC(M_k) + \text{Shannon}(\mathbf{Z} \mid \mathbf{Y} = \mathbf{y}; \hat{\theta}_{M_k}, M_k).$$

- Recall that the Shannon's entropy is:
  - 0 when classification is 100% sure;
  - $n \log K$ when classification is a "coin tossing" rule.

BIC or ICL?

- There is a strong connection between these two quantities
- More precisely it is not difficult to show that

$$ICL(M_k) = BIC(M_k) + \text{Shannon}(\mathbf{Z} \mid \mathbf{Y} = \mathbf{y}; \hat{\theta}_{M_k}, M_k).$$

- Recall that the Shannon's entropy is:
  - 0 when classification is 100% sure;
  - $n \log K$ when classification is a "coin tossing" rule.

☞ Compared to BIC, ICL tends to favor models that produce more clearly separated clusters and points to the same or a smaller number of clusters.

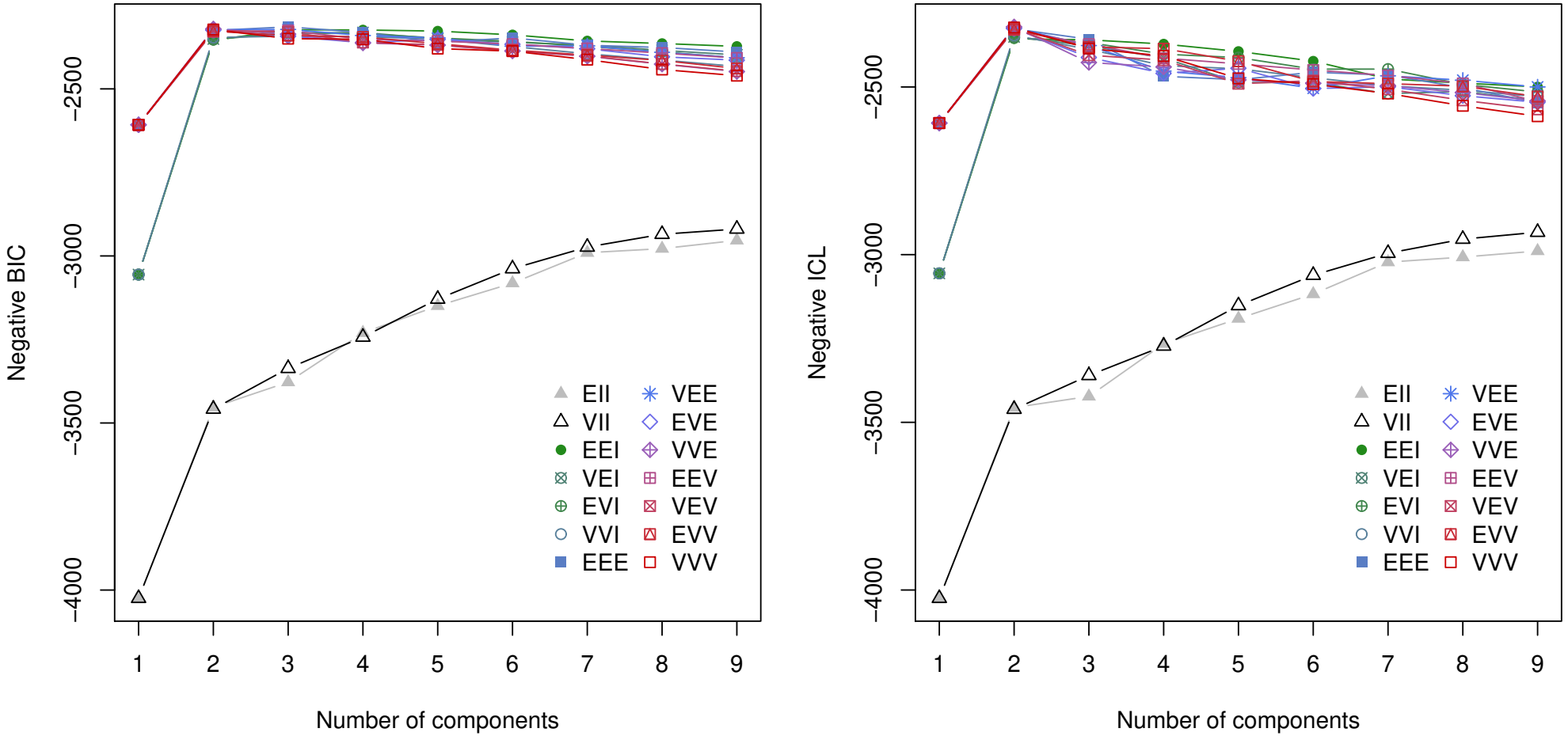


Figure 11: *Trace of the negative BIC and ICL on the old faithful dataset.*

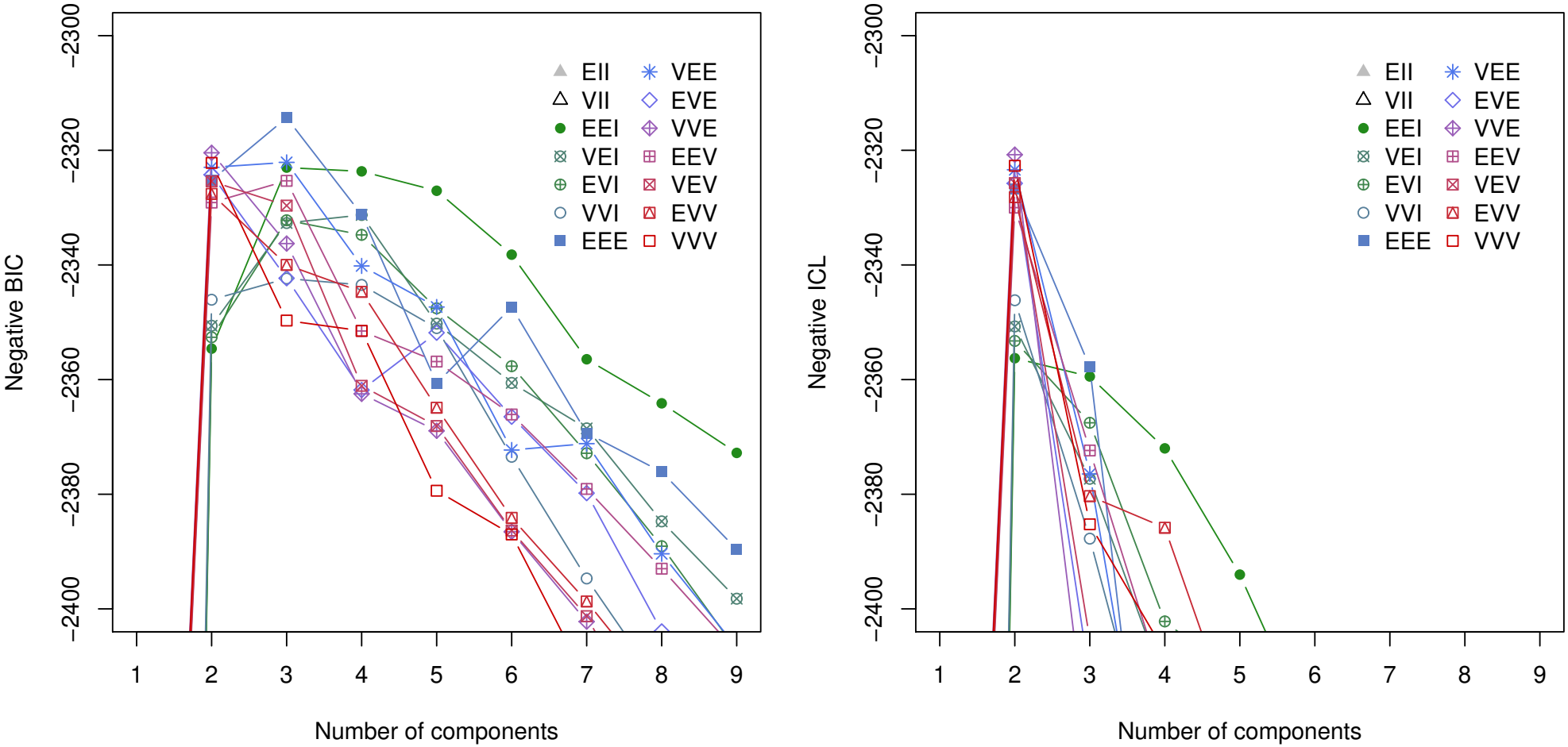


Figure 11: *Trace of the negative BIC and ICL on the old faithful dataset.*

- BIC selects the `EEE` model, i.e., $\Sigma_g = \Sigma$, with $G = 3$ clusters.
- ICL selects the `VVE` model, i.e., equal orientation, with $G = 2$ clusters.

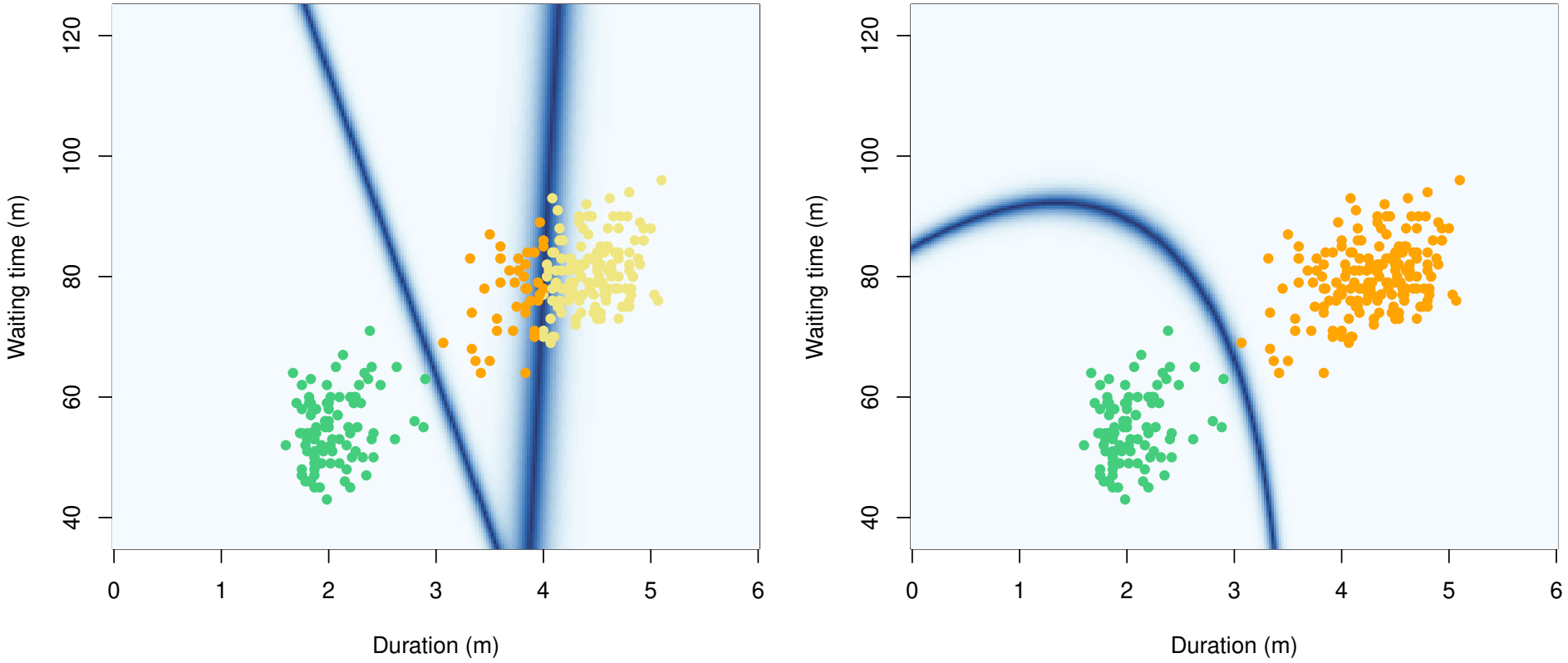


Figure 12: *Clustering from the best selected models using BIC (left) and ICL (right) for the old faithful dataset.*

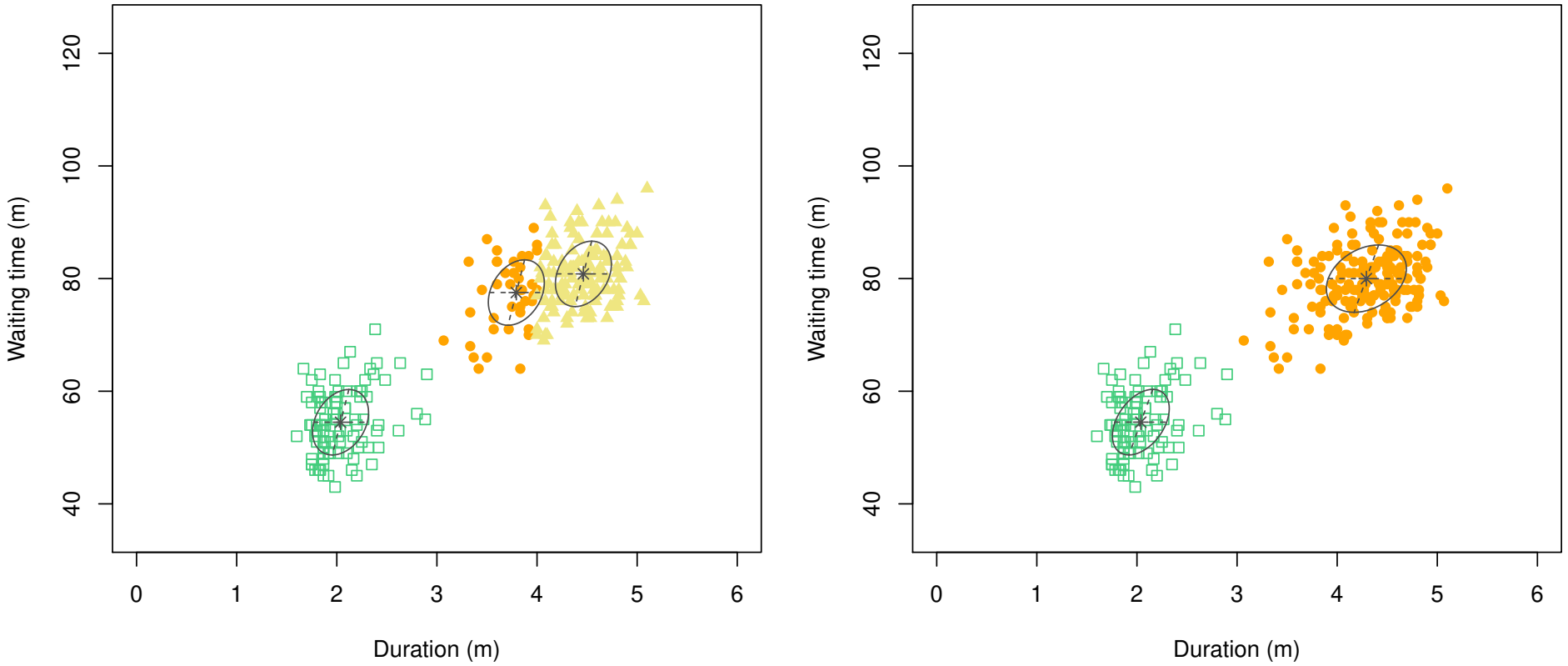


Figure 12: *Clustering from the best selected models using BIC (left) and ICL (right) for the old faithful dataset.*

☞ There are likely two clusters but one is non Gaussian but can be modelled using a mixture of two gaussian distributions.

A. High dimensional data

- Curse of dimensionality is often heard... But what is it?
- It is not just a computational issue, it is something that most often surprises[8] as high dimensional spaces have unexpected behavior.
- We will start with two famous examples:
  - the volume of the sphere
  - probability of lying within a thin shell

[8] and may well be a bless actually...

Volume of the unit hyper–sphere

Recall that the volume of the unit sphere of $\mathbb{R}^d$, $d \geq 1$ is

$$V(d) = \frac{\pi^{d/2}}{\Gamma(1 + d/2)}.$$

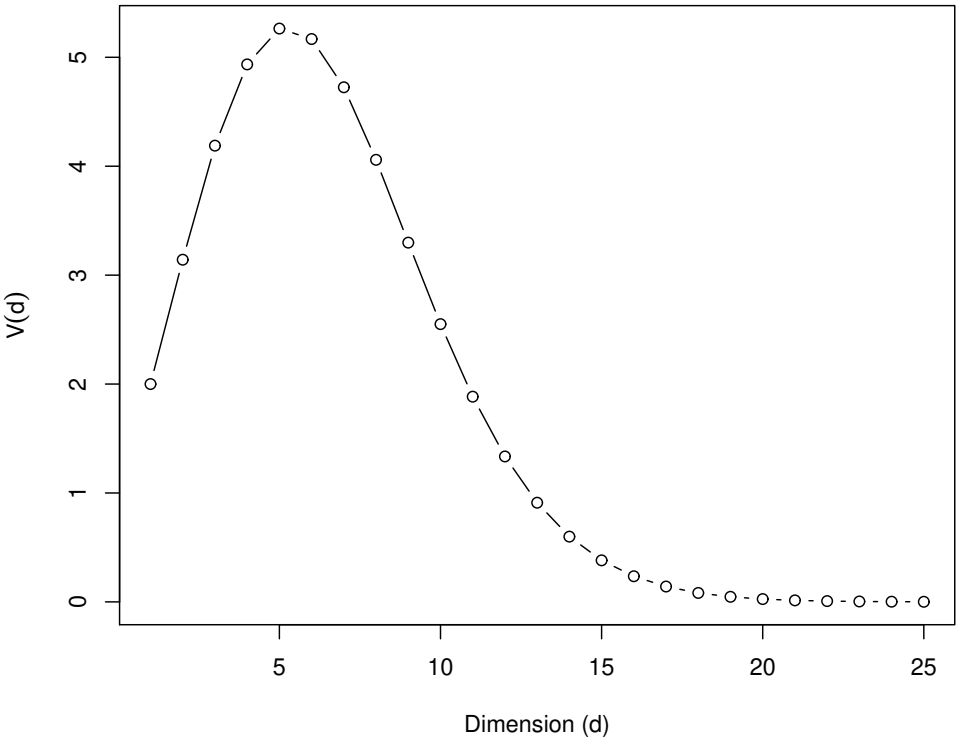


Figure 13: *Evolution of the volume of the unit hyper–sphere as the dimension increases.*

Volume of the unit hyper–sphere

Recall that the volume of the unit sphere of $\mathbb{R}^d$, $d \geq 1$ is

$$V(d) = \frac{\pi^{d/2}}{\Gamma(1 + d/2)}.$$

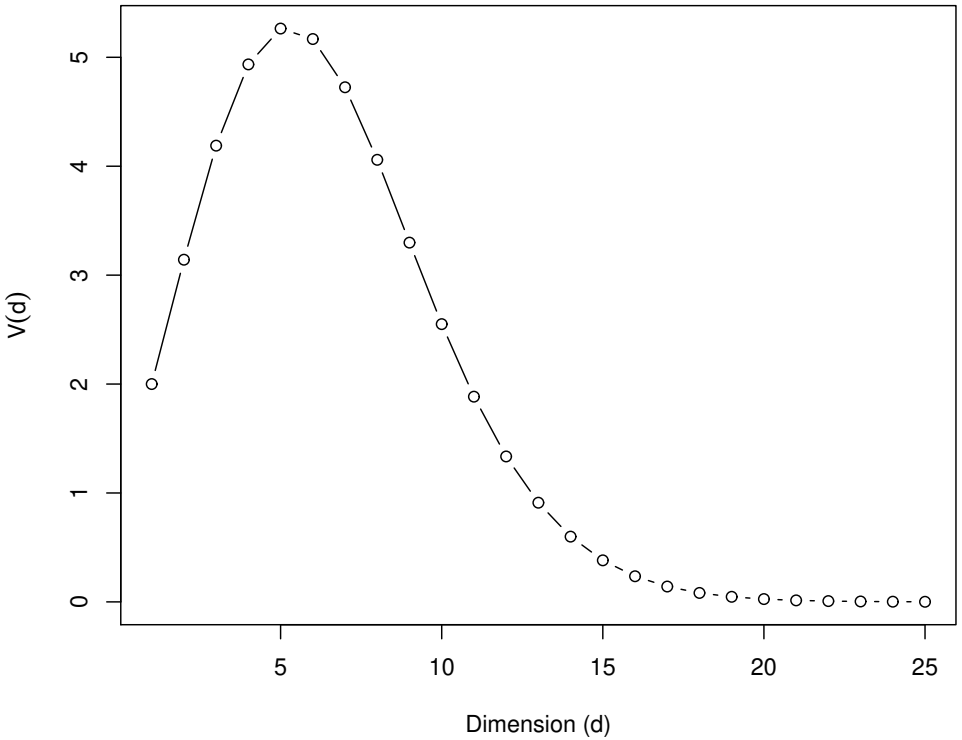


Figure 13: *Evolution of the volume of the unit hyper–sphere as the dimension increases.*

☞ The volume drops after $d = 5$!!!

Lying in shell

Consider the shell

$$\text{Shell}_d(r) = S_d(1) \setminus S_d(r),$$

where $S_d(r) = \{x \in \mathbb{R}^d : \|x\| \leq r\}$.
Let $X \sim U(S_d(1))$. We thus have

$$\begin{aligned}\Pr\{X \in \text{Shell}_d(r)\} &= 1 - \Pr(\|X\| < r) \\ &= 1 - \frac{r^d V(d)}{V(d)} \\ &= 1 - r^d\end{aligned}$$

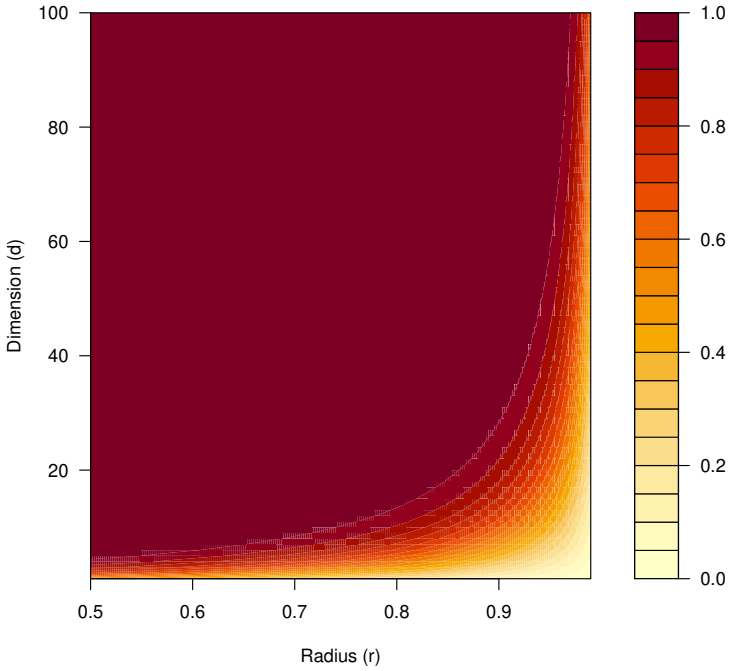


Figure 14: *Evolution of the probability to fall within* $\text{Shell}_d(r)$ *as the dimension* d *and the radius* r *of the nested hyper–sphere vary.*

Lying in shell

Consider the shell

$$\text{Shell}_d(r) = S_d(1) \setminus S_d(r),$$

where $S_d(r) = \{x \in \mathbb{R}^d : \|x\| \leq r\}$.
Let $X \sim U(S_d(1))$. We thus have

$$\begin{aligned}\Pr\{X \in \text{Shell}_d(r)\} &= 1 - \Pr(\|X\| < r) \\ &= 1 - \frac{r^d V(d)}{V(d)} \\ &= 1 - r^d\end{aligned}$$

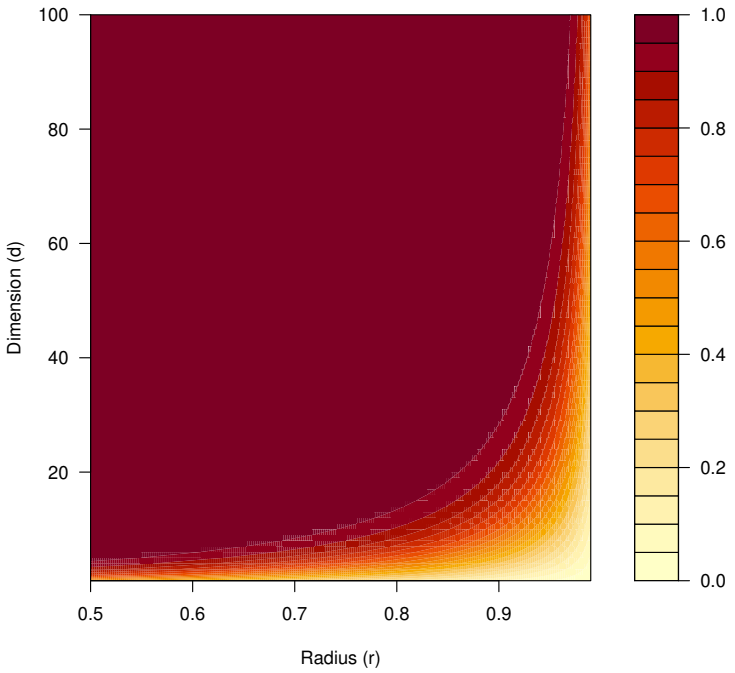


Figure 14: *Evolution of the probability to fall within* $\text{Shell}_d(r)$ *as the dimension* d *and the radius* r *of the nested hyper–sphere vary.*

☞ When r is fixed, the probability gets larger as d increases, i.e., most of the space is empty!

Dimension reduction

- A common way to circumvent this curse of dimensionality is to use dimension reduction techniques such as principal component analysis
- To be more specific, you apply your mixture model on a subset of the principal components.
- However it has been shown that this may fail has:
  - there is no reason to ensure that the first k principal components, say, give the best results in terms of separation of clusters;
  - dimension reduction may not be a sensible approach at all: remember that high dimensional spaces are empty so clusters are likely to live into more "reasonable dimensional spaces"

Dimension reduction

- A common way to circumvent this curse of dimensionality is to use dimension reduction techniques such as principal component analysis
- To be more specific, you apply your mixture model on a subset of the principal components.
- However it has been shown that this may fail has:
  - there is no reason to ensure that the first k principal components, say, give the best results in terms of separation of clusters;
  - dimension reduction may not be a sensible approach at all: remember that high dimensional spaces are empty so clusters are likely to live into more "reasonable dimensional spaces"

☞ It is therefore tempting to define Gaussian distribution suited to such "emptiness structure" of space!

- Recall our Volume–Shape–Orientation decomposition

$$\Sigma_g = \lambda_g Q_g A_g Q_g^\top.$$

- One possibility is to consider the situation where the shape component is structured as

$$A_g = \begin{bmatrix} \tilde{A}_{d_g} & \mathbf{0} \\ \mathbf{0} & c\mathsf{Id}_{d-d_g} \end{bmatrix}, \qquad 1 \leq d_g < d,$$

  where A_{d_g} is a diagonal matrix and $c = |\tilde{A}_{d_g}|^{-1/(d-d_g)}$ to ensure that $|A_g| = 1$.

- Such structure imposes that the g–th cluster actually lives within a d_g dimensional space while the remaining $d - d_g$ coordinates can be thought as white noise.

B. Non Gaussian mixtures

Dealing with Categorical variable

- So far the Gaussian distribution plays a central role to our modelling strategy
- Other continuous multivariate distribution are possible: Student, Skew-normal, ...

Dealing with Categorical variable

- So far the Gaussian distribution plays a central role to our modelling strategy
- Other continuous multivariate distribution are possible: Student, Skew-normal, ... But what about categorical variables?

Dealing with Categorical variable

- So far the Gaussian distribution plays a central role to our modelling strategy
- Other continuous multivariate distribution are possible: Student, Skew-normal, ... But what about categorical variables?

"Well just switch your multivariate Gaussian distribution for that of a categorical one!"

Dealing with Categorical variable

- So far the Gaussian distribution plays a central role to our modelling strategy
- Other continuous multivariate distribution are possible: Student, Skew-normal, ... But what about categorical variables?

"Well just switch your multivariate Gaussian distribution for that of a categorical one!"

"Thanks Dude, but do you know any (flexible) multivariate categorical distribution?"

Dealing with Categorical variable

- So far the Gaussian distribution plays a central role to our modelling strategy
- Other continuous multivariate distribution are possible: Student, Skew-normal, ... But what about categorical variables?

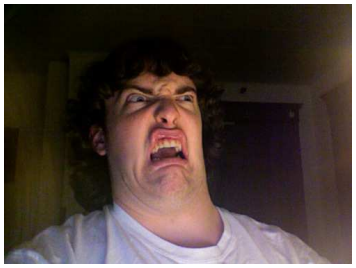
"Well just switch your multivariate Gaussian distribution for that of a categorical one!"

"Thanks Dude, but do you know any (flexible) multivariate categorical distribution?"

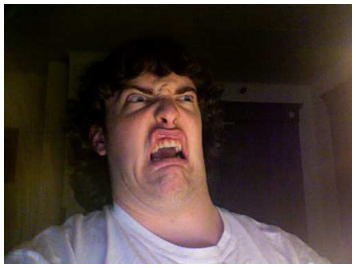
☞ One way to bypass this hurdle is to use a conditional independence assumption.

Notation nightmare

- Watch out we are entering in a notation nightmare zone!



Notation nightmare



- Watch out we are entering in a notation nightmare zone!

- We have a $Y = (Y_1, \ldots, Y_d)$ multivariate random vector whose components are discrete random variable with $m_1, \ldots, m_d$ levels.
- We will use the nasty indices notation $\cdot_{g,j,\ell}$ where
  - g indices the cluster label, i.e., $Z = g$;
  - j indices the variable, i.e., Y_j;
  - ℓ indices the level, i.e., $Y_j = \ell$.

Conditional independence

- The latent class model assumes that, conditionally on z, those categorical variables are independent so that we will deal with univariate categorical distribution only!
- The density of an observation $y = (y_1, \ldots, y_d)$, $d \geq 1$, is now

$$
\begin{aligned}
p(y) &= \sum_{g=1}^{G} \tau_g \Pr_{\theta_g}(Y = y \mid Z = g) \\
&= \sum_{g=1}^{G} \tau_g \prod_{j=1}^{d} \Pr_{\theta_g}(Y_j = y_j \mid Z = g) \\
&= \sum_{g=1}^{G} \tau_g \prod_{j=1}^{d} p_{g,j,y_j},
\end{aligned}
$$

where $p_{g,j} = (p_{g,j,1}, \ldots, p_{g,j,m_j})$ is the vector of mass probabilities for the categorical r.v. Y_j that has m_j levels.

- Inference will be done as before using our best–friend the EM algorithm.
- In this setting, from n ind. obs $y_i = (y_{i,1}, \ldots, y_{i,d})$, we thus have

$$
\begin{aligned}
Q(\theta \mid \theta^{(t)}) &= \sum_{g=1}^{G} \sum_{i=1}^{n} \log \left\{ \tau_g \prod_{j=1}^{d} p_{g,j,y_{i,j}} \right\} t_g^{(t)}(y_i) \\
&= \sum_{g=1}^{G} \sum_{i=1}^{n} t_g^{(t)}(y_i) \log \tau_g + \sum_{j=1}^{d} \sum_{g=1}^{G} \sum_{i=1}^{n} t_g^{(t)}(y_i) \log p_{g,j,y_{i,j}}.
\end{aligned}
$$

- Inference will be done as before using our best–friend the EM algorithm.
- In this setting, from n ind. obs $y_i = (y_{i,1}, \ldots, y_{i,d})$, we thus have

$$Q(\theta \mid \theta^{(t)}) = \sum_{g=1}^{G} \sum_{i=1}^{n} \log \left\{ \tau_g \prod_{j=1}^{d} p_{g,j,y_{i,j}} \right\} t_g^{(t)}(y_i)$$

$$= \sum_{g=1}^{G} \sum_{i=1}^{n} t_g^{(t)}(y_i) \log \tau_g + \sum_{j=1}^{d} \sum_{g=1}^{G} \sum_{i=1}^{n} t_g^{(t)}(y_i) \log p_{g,j,y_{i,j}}.$$

☞ The M Step consists in $d+1$ independent optimization problems.

M Step for categorical distribution

- As we know updating τ_g will be identical, i.e., $\tau_g^{(t+1)} = \sum_{i=1}^n t_g^{(t)}(y_i)/n$;
- It is straightforward to see that updating the probability masses is

$$p_{g,j,\ell}^{(t+1)} = \frac{\sum_{i=1}^n t_g^{(t)}(y_i) 1_{\{y_{i,j}=\ell\}}}{\sum_{i=1}^n t_g^{(t)}(y_i)},$$

  for $g = 1, \ldots, G$, $j = 1, \ldots, d$ and $\ell = 1, \ldots, m_j$.
- However we may experience division by 0 issues, so we may consider the regularized updating formula

$$p_{g,j,\ell}^{(t+1)} = \frac{c + \sum_{i=1}^n t_g^{(t)}(y_i) 1_{\{y_{i,j}=\ell\}}}{m_j c + \sum_{i=1}^n t_g^{(t)}(y_i)},$$

  for some fixed $c > 0$.[9]

[9] Results are sensitive to the choice of c though...

Parsimonious categorical mixture model

- The above model has $(G-1)+G\sum_{j=1}^{d}(m_j-1)$ parameters, and, as for the Gaussian case, there is a pressing need for parsimonious models.
- We first parametrize the Discrete(p) distribution with $p=(p_1,\ldots,p_m)$, $m\geq 1$, using the most probable level and scattering parameters

$$\ell_* = \underset{\ell\in\{1,\ldots,m\}}{\arg\max}\ p_\ell, \qquad \varepsilon_\ell = \begin{cases} 1-p_\ell, & \text{if } \ell=\ell_* \\ p_\ell, & \text{otherwise.} \end{cases}.$$

- Parsimony is obtained by adding constraints on the ε:

$\varepsilon_{g,j,\ell}=\varepsilon$	constant scattering;
$\varepsilon_{g,j,\ell}=\varepsilon_j$	dependence on variables, but not on clusters and levels;
$\varepsilon_{g,j,\ell}=\varepsilon_g$	dependence on clusters, but not upon variables;
$\varepsilon_{g,j,\ell}=\varepsilon_{g,j}$	dependence on clusters and variables, but not levels;
$\varepsilon_{g,j,\ell}=\varepsilon_{g,j,\ell}$	dependence on clusters, variables and levels.

Parsimonious categorical mixture model

- The above model has $(G-1)+G\sum_{j=1}^{d}(m_j-1)$ parameters, and, as for the Gaussian case, there is a pressing need for parsimonious models.
- We first parametrize the Discrete(p) distribution with $p=(p_1,\ldots,p_m)$, $m\geq 1$, using the most probable level and scattering parameters

$$\ell_* = \underset{\ell\in\{1,\ldots,m\}}{\arg\max}\ p_\ell, \qquad \varepsilon_\ell = \begin{cases} 1-p_\ell, & \text{if } \ell=\ell_* \\ p_\ell, & \text{otherwise.} \end{cases}.$$

- Parsimony is obtained by adding constraints on the ε:

$\varepsilon_{g,j,\ell}=\varepsilon$	constant scattering;
$\varepsilon_{g,j,\ell}=\varepsilon_j$	dependence on variables, but not on clusters and levels;
$\varepsilon_{g,j,\ell}=\varepsilon_g$	dependence on clusters, but not upon variables;
$\varepsilon_{g,j,\ell}=\varepsilon_{g,j}$	dependence on clusters and variables, but not levels;
$\varepsilon_{g,j,\ell}=\varepsilon_{g,j,\ell}$	dependence on clusters, variables and levels.

Focus on the $\varepsilon_{g,j,\ell} = \varepsilon_{g,j}$ model

- This model corresponds to the case where

$$\varepsilon_{g,j,\ell} = \begin{cases} 1 - p_{g,j,\ell}, & \text{if } \ell = \ell_{*,j} \\ \frac{1-p_{g,j,\ell_{*,j}}}{m_j - 1}, & \text{otherwise} \end{cases},$$

  where $\ell_{*,j}$ is the most probable level for the j-th variable.
- We now have $(G-1) + Gd$ parameters, to be compared to the completely free model $(G-1) + G\sum_{j=1}^{d}(m_j - 1)$.
- This model is generally a good trade-off between parsimony and flexibility.
- It has the main advantage that the EM algorithm still have explicit forms.

Focus on the $\varepsilon_{g,j,\ell} = \varepsilon_{g,j}$ model

- This model corresponds to the case where

$$\varepsilon_{g,j,\ell} = \begin{cases} 1 - p_{g,j,\ell}, & \text{if } \ell = \ell_{*,j} \\ \frac{1-p_{g,j,\ell_{*,j}}}{m_j - 1}, & \text{otherwise} \end{cases},$$

  where $\ell_{*,j}$ is the most probable level for the j-th variable.
- We now have $(G-1) + Gd$ parameters, to be compared to the completely free model $(G-1) + G\sum_{j=1}^{d}(m_j - 1)$.
- This model is generally a good trade-off between parsimony and flexibility.
- It has the main advantage that the EM algorithm still have explicit forms.

☞ When all the variable have two levels, this model matches the "completely free" model.

Variable of mixed type

- Suppose that the first d_1 variables are continuous and the last $d - d_1$ are categorical.
- To handle such data we still rely on our conditional independence assumption.
- More precisely, conditionaly on Z, the continuous and categorical variables are independent (and the categorical variables are mutually independent as before).
- Because of this conditional independence, devising an EM algorithm is straightforward.

C. Revisiting the k–means

- Recall the the CEM algorithm is a modification of the original EM algorithm where you "insert" an imputation step for the latent variables $\mathbf{z}$ between the E and the M steps.
- This imputation stage is done using maximum a posteriori allocation, i.e.,

$$\mathbf{z}^{(t+1)} = \arg\max_{\mathbf{z}} L(\theta^{(t)}; \mathbf{y}, \mathbf{z}).$$

- Then the parameter θ is updated by maximizing the completed likelihood with those imputed values.

- Consider the Gaussian mixture model where $\Sigma_g = \lambda \mathsf{Id}$ and $\tau_g = 1/G$, $g = 1, \ldots, G$.
- Then given the current parameter $\theta^{(t)}$, we have

$$\begin{aligned}
\log f(y_i, z_i; \theta^{(t)}) &= \log \tau_{z_i}^{(t)} + \log \varphi(y_i; \mu_{z_i}^{(t)}; \lambda^{(t)} \mathsf{Id}) \\
&= -\log G - \frac{d}{2} \log(2\pi\lambda^{(t)}) - \frac{(y_i - \mu_{z_i}^{(t)})^\top (y_i - \mu_{z_i}^{(t)})}{2\lambda^{(t)}} \\
&= \mathsf{cst} - \frac{\|y_i - \mu_{z_i}^{(t)}\|_2^2}{2\lambda^{(t)}}.
\end{aligned}$$

- Hence it is clear that

$$z_{*,i} = \underset{g \in \{1,\ldots,G\}}{\arg\max} \ \log f(y_i, z_i; \theta^{(t)}) = \underset{g \in \{1,\ldots,G\}}{\arg\min} \ \|y_i - \mu_g^{(t)}\|_2.$$

- Now using such imputed values $\mathbf{z}_*$, we have

$$\log L(\theta; \mathbf{y}, \mathbf{z}_*) = \text{cst} - \frac{n}{2}\log(2\pi\lambda) - \sum_{g=1}^{G} \sum_{i:\ z_{*,i}=g} \frac{\|y_i - \mu_g\|_2^2}{2\lambda}.$$

- For any $g \in \{1, \ldots, G\}$, this completed log-likelihood is maximized at

$$\mu_g = \frac{1}{n_g} \sum_{i:\ z_i=g} y_i, \qquad \lambda = \frac{1}{n} \sum_{g=1}^{n} \sum_{i:\ z_i=g} \|y_i - \mu_g\|_2^2.$$

Algorithm 3: CEM algorithm for spherical and "tied" Gaussian mixtures.

input : Suitable initial values for μ_g and λ
output: Estimates for μ_g, λ and class labels $\mathbf{z}$

while *not converged* **do**

`/* C Step: */`

$$z_{*,i} = \underset{g \in \{1,\ldots,G\}}{\arg\min} \|y_i - \mu_g^{(t)}\|_2$$

`/* M Step: */`

for $g = 1, \ldots, G$ **do**

$$\mu_g = \frac{1}{n_g} \sum_{i:\ z_i = g} y_i, \qquad \lambda = \frac{1}{n} \sum_{g=1}^{n} \sum_{i:\ z_i = g} \|y_i - \mu_g\|_2^2$$

Return μ_g, λ and $\mathbf{z}_*$;

Algorithm 3: CEM algorithm for spherical and "tied" Gaussian mixtures.

input : Suitable initial values for μ_g and λ
output: Estimates for μ_g, λ and class labels $\mathbf{z}$

while *not converged* **do**
  `/* C Step:` `*/`

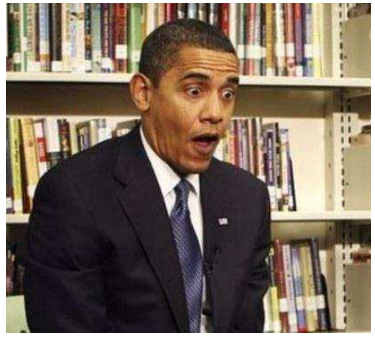
$$z_{*,i} = \underset{g \in \{1,\ldots,G\}}{\arg\min} \|y_i - \mu_g^{(t)}\|_2$$

  `/* M Step:` `*/`
  for $g = 1, \ldots, G$ **do**

$$\mu_g = \frac{1}{n_g} \sum_{i:\ z_i = g} y_i, \qquad \lambda = \frac{1}{n} \sum_{g=1}^{n} \sum_{i:\ z_i = g} \|y_i - \mu_g\|_2^2$$

Return μ_g, λ and $\mathbf{z}_*$;

☞ Hey but that's the k–means algorithm!



THAT'S IT! I HOPE YOU ENJOYED THIS LECTURE!

Wait! One more slide...

Wait! One more slide. . .

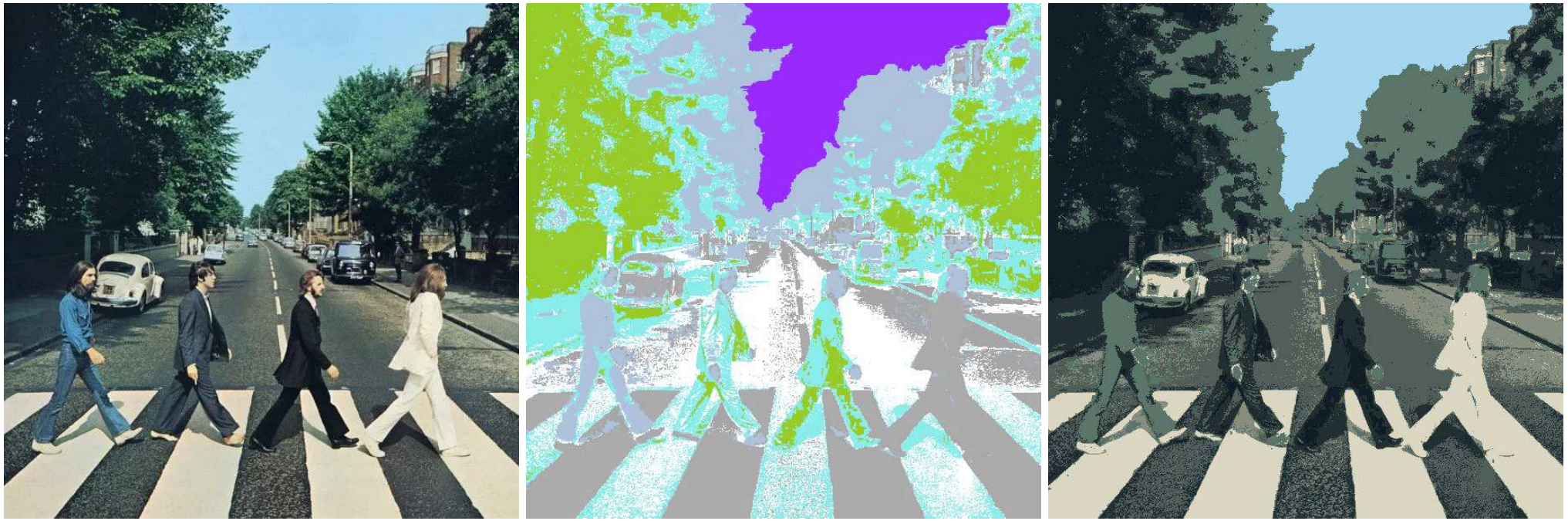


Figure 15: *We had fun right!*