BSM5 - Bayesian clustering

Lecturer: Riccardo Corradin

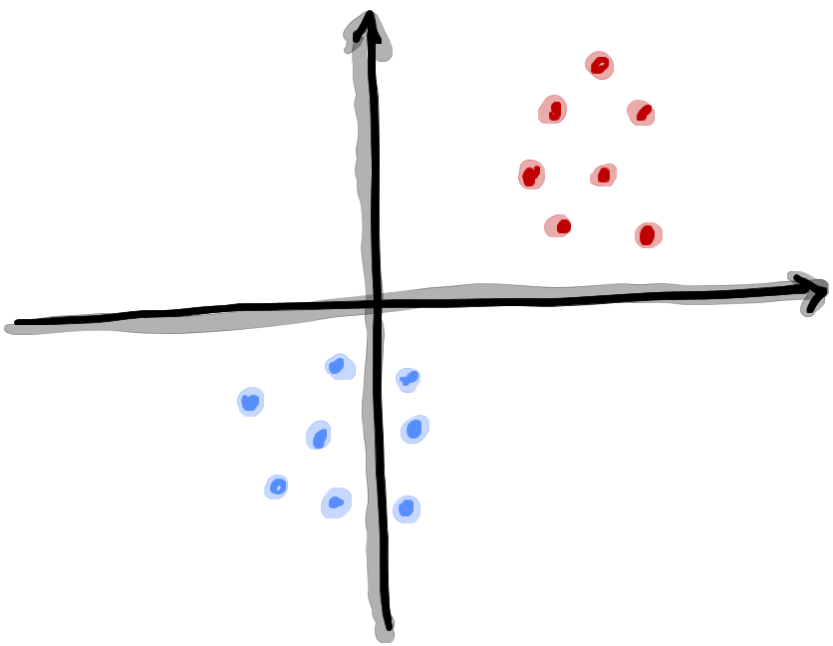
Introduction

Welcome back clustering! Clustering is one of the fundamental techniques in statistical analysis, and any statistician should know the fundamental approaches.

Suppose we have a sequence $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ of observations, where the generic $\boldsymbol{Y}_i \in \mathbb{Y}$. Ideally, in cluster analysis, we aim to identify subsets of observations which result to be **similar** in their observed values.

$\rightarrow$ Quite a general definition, what does it mean similar?
$\rightarrow$ Unsupervised learning.

Here we assume that $\mathbb{Y} \subseteq \mathbb{R}^p$, i.e. we aim to produce homogeneous clusters of real-valued quantities. However, many generalizations can be considered.



In simple scenarios, defining clusters can be quite trivial.

However, when the number of dimensions increases, or the domain is getting complex, we should be more careful in the specification of our model.

Introduction

We mainly distinguish between two different families of clustering approaches.

- **Model-based clustering**, where we assume the observations to be distributed according to cluster-specific distributions. Hence, the homogeneity among observations is driven by a probabilistic models.
  - → Quite flexible approach, the group-specific distribution can be any kind of model and it plays the role of likelihood.
  - → The group-specific distribution can be useful also to interpret the data behavior.
  - → In this case, we call the joint model **mixture model** (different from the mixed model of two slide blocks ago).

- **Distance-based clustering**, where we use a distance to measure dissimilarities among observations, and clusters are defined by observation closed to each others.
  - → There are several distances that we can use, depending on specific problems.
  - → There is no interpretation in terms of model structure.
  - → We do not have an usual likelihood function.

In the following, we will study how to perform cluster analysis with a Bayesian approach in the first case, where we assume a cluster-specific distribution.

Model-based clustering with Bayes

Model-based clustering with Bayes

The key objects to perform model-based clustering are the so-called **mixture models**.

A mixture model over $\mathbb{Y}$ is nothing but a weighted average of different density functions or probability mass functions, of the form

$$f(\boldsymbol{y}) = \sum_{j=1}^{k} w_j k(\boldsymbol{y}, \boldsymbol{\theta}_j^*),$$

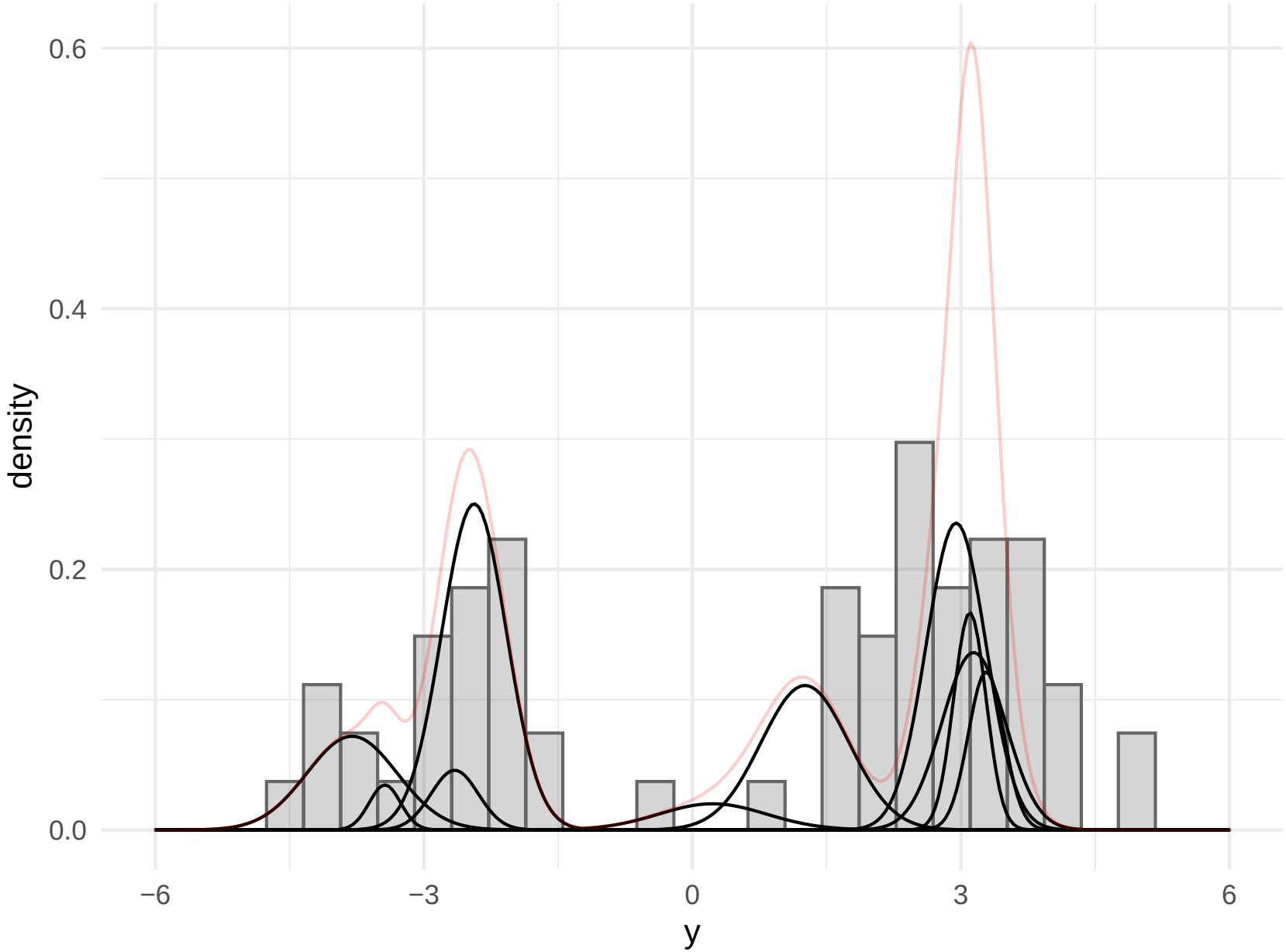
where $\boldsymbol{\theta}_j^* \in \Theta$, $j = 1, \ldots, k$, and $k \in \mathbb{N} \cup \{\infty\}$.

- The sequence $w_1, \ldots, w_k$ is a sequence of non-negative weights taking values in the $(k-1)$-dimensional simplex space, i.e.

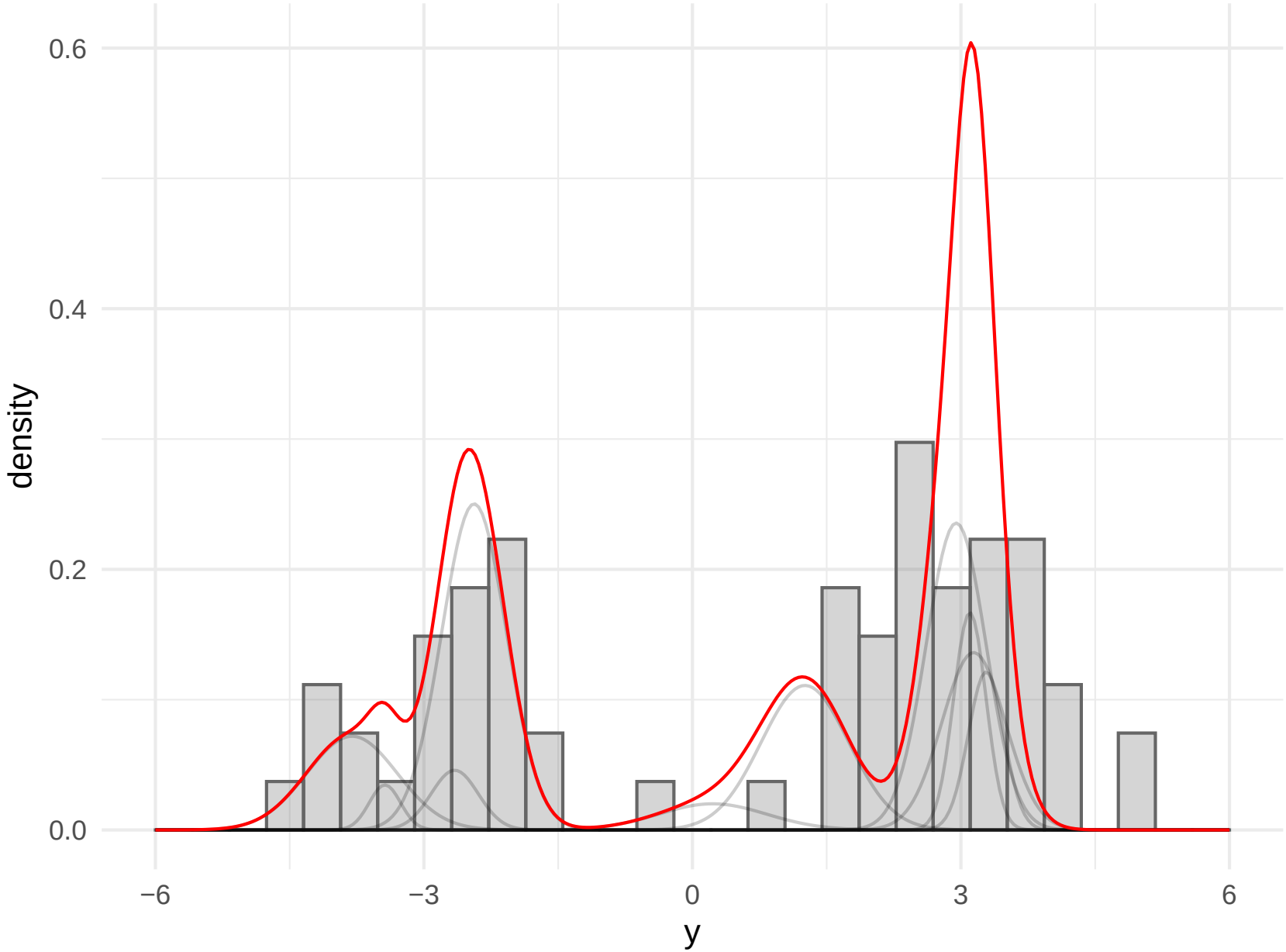
$$\triangle_{k-1}^1 = \left\{ w_1, \ldots, w_k : 0 \leq w_j \leq 1,\ j = 1 \ldots, k,\ \sum_{j=1}^{k} w_j = 1 \right\}.$$

- $k(y, \theta)$ is a kernel function satisfying the followings:
  - $k(, \theta)$ is a density function or a probability mass function for any value of $\theta \in \Theta$;
  - $k(y,)$ is measurable for any value of $\boldsymbol{y} \in \mathbb{Y}$.

Model-based clustering with Bayes



Model-based clustering with Bayes



Model-based clustering with Bayes

Pros

- Mixture models are flexible and can capture many complex distributional behavior, such as multimodality and skewness.
- They are naturally tailored to perform model based clustering, as each component can represent a single cluster.
- Despite their simple specification, they can accommodate many different type of data by suitably specifying the kernel function.
  - $\rightarrow$ $k(\boldsymbol{y}, \boldsymbol{\theta}) \stackrel{d}{=} N(\mu, \sigma)$ to cluster univariate data defined on $\mathbb{R}$.
  - $\rightarrow$ $k(\boldsymbol{y}, \boldsymbol{\theta}) \stackrel{d}{=} N(\boldsymbol{\mu}, \Sigma)$ to cluster univariate data defined on $\mathbb{R}^d$.
  - $\rightarrow$ $k(\boldsymbol{y}, \boldsymbol{\theta}) \stackrel{d}{=} GP(\mu(t), \mathrm{R}(t, t'))$ to cluster functional data taking values on $C_{\mathbb{X}}$, space of continuous function with support $\mathbb{X}$.
  - $\rightarrow$ $k(\boldsymbol{y}, \boldsymbol{\theta}) \stackrel{d}{=} ERGM(\boldsymbol{\theta})$ to cluster graph data.
  - $\rightarrow$ ...

Cons

- The estimation is usually computationally intensive.
- They might be prohibitive with large sample sizes.
- Finite mixture models can be sensible to their specification.

Model-based clustering with Bayes

To complete the model specification, we need to assume a specific **weights distribution** $\boldsymbol{w} \sim \pi(\boldsymbol{w})$, the kernel function $k(\boldsymbol{y}, \boldsymbol{\theta})$, and a distribution for the cluster-specific parameters θ_j^*.

- One common choice for the weights distribution is the **Dirichlet distribution**, i.e. $\boldsymbol{w} \sim \textit{Dirichlet}(\boldsymbol{\alpha})$ with

$$\pi(\boldsymbol{w}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_k)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_k)} \prod_{j=1}^{k} w_j^{\alpha_j - 1},$$

  having support $\triangle_{k-1}^1$, the $k-1$-dimensional simplex with total mass 1.

  → Recall that $\mathrm{E}[W_j] = \frac{\alpha_j}{\alpha^+}$ and $\mathrm{var}(W_j) = \frac{\alpha_j(\alpha^+ - \alpha_j)}{\alpha^+(\alpha^+ + 1)}$, where $\alpha^+ = \sum_{j=1}^{k} \alpha_j$.
  → Different choices of $\alpha_1, \ldots, \alpha_k$ lead to different specification.
  → The case $\alpha_1 = \cdots = \alpha_k = \alpha$ is called symmetric.
  → In the symmetric case, common settings are $\alpha = 1$ or $\alpha = \frac{1}{k}$.

- Specific kernel function choices **depend on the type of data** we are analyzing.
  → If the data are discrete, continuous, functions, etc.
  → If we want a particular feature from the kernel function, such as skewness, heavy tails, etc.

- The distribution of $\boldsymbol{\theta}_j^*$, denoted by $\pi(\boldsymbol{\theta}_j^*)$, **depends on the kernel assumption**.

Model-based clustering with Bayes

Ideally, we are in a situation where **our observed data** are distributed according to a **mixture model**

$$\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k} \overset{\text{iid}}{\sim} f(y) = \sum_{j=1}^{k} w_j k(\boldsymbol{y}, \boldsymbol{\theta}^*_j)$$

$$\boldsymbol{w} \sim Dir(\boldsymbol{\alpha}),$$

$$\boldsymbol{\theta}^*_1, \dots, \boldsymbol{\theta}^*_k \overset{\text{iid}}{\sim} \pi(\boldsymbol{\theta}),$$

where $\pi(\boldsymbol{\theta})$ is a suitable prior for the component-specific parameter, matching its support.

A first augmentation

Equivalently, we can consider a **sequence of latent parameters** $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n$, each associated to a **specific observations**, having

$$\boldsymbol{Y}_i \mid \boldsymbol{\theta}_i \sim k(\boldsymbol{y}_i, \boldsymbol{\theta}_i), \qquad i = 1, \dots, n$$

$$\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k} \overset{\text{iid}}{\sim} \sum_{j=1}^{k} w_j \delta_{\boldsymbol{\theta}^*_j}(\boldsymbol{\theta}),$$

$$\boldsymbol{w} \sim Dir(\boldsymbol{\alpha}),$$

$$\boldsymbol{\theta}^*_1, \dots, \boldsymbol{\theta}^*_k \overset{\text{iid}}{\sim} \pi(\boldsymbol{\theta}).$$

Model-based clustering with Bayes

We can easily see from the previous that marginalizing with respect to a generic $\boldsymbol{\theta}_i$ the joint distribution of $\boldsymbol{Y}_i, \boldsymbol{\theta}_i \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k}$, we recover the distribution of $\boldsymbol{Y}_i \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k}$.

Model-based clustering with Bayes

In practice, we are matching each observation with a latent parameter, with the distributional assumption

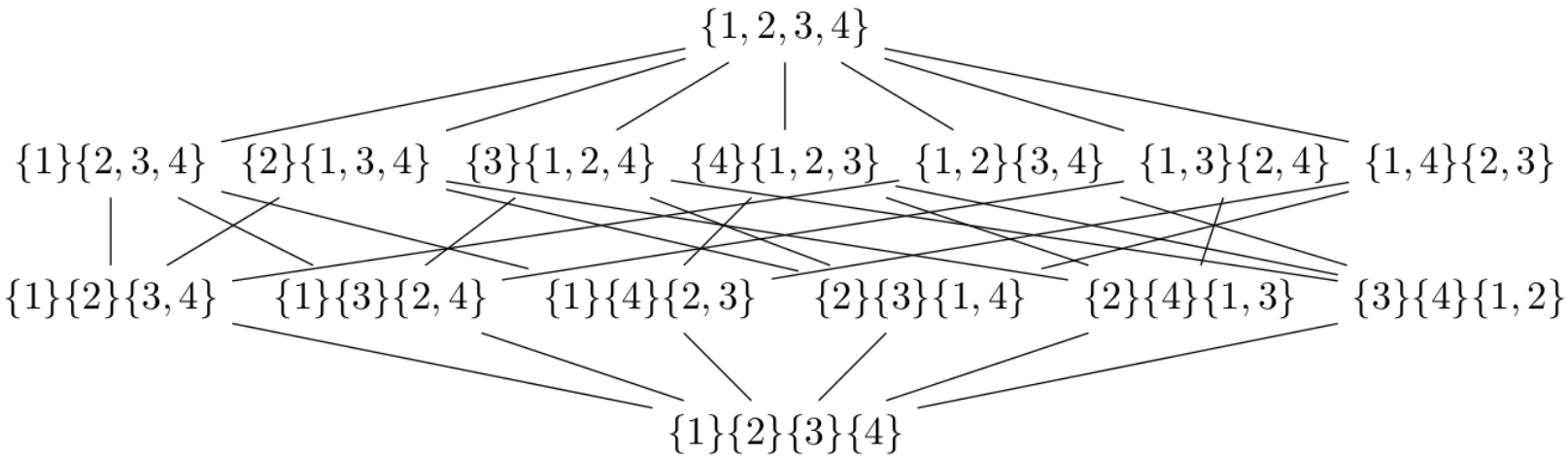
$$\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k} \overset{\text{iid}}{\sim} \sum_{j=1}^{k} w_j \delta_{\boldsymbol{\theta}^*_j}(\boldsymbol{\theta}).$$

- The previous is a case of **random probability measure**, since
  - $\rightarrow$ it is a probability measure;
  - $\rightarrow$ both weights $\boldsymbol{w}$ and atoms $\boldsymbol{\theta}^*_{1:k}$ are random quantities.

- Further, the previous is a discrete random probability measure.
  - $\rightarrow$ The sequence of latent parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ can then have ties, meaning that
  $$P(\boldsymbol{\theta}_i = \boldsymbol{\theta}_j) > 0, \qquad i \neq j.$$
  - $\rightarrow$ Ties implies that the observations can be grouped together when the corresponding latent parameters take the same unique values $\boldsymbol{\theta}^*_j$, hence they are generated from the same kernel function.
  - $\rightarrow$ This is naturally inducing a clustering within the data, where clusters mean that observations share the same distribution in a model-based case.

Model-based clustering with Bayes

Ideally, we have a latent partition induced by **ties among latent parameters**, defining equivalence classes. Unfortunately, the space spanning possible partitions is complex and **grows rapidly** as far the sample size increases.



- A partition, here denoted by ρ_n, is then a set of blocks $\{A_1, \ldots, A_k\}$, where the generic jth element $A_j = \{i \in \{1, \ldots, n\} \text{ s.t. } S_i = j\}$, $j = 1, \ldots, k$.
  - The sets are disjoint, i.e. $A_i \cap A_j = \emptyset$, for $i \neq j$. This means that an element can belong only to a single cluster.
  - The union of all the blocks recover the set of observed indices, i.e. $A_1 \cup A_2 \cup \cdots \cup A_k = \{1, \ldots, n\}$.
- We denote by n_j the size of each block, i.e. $n_j = |A_j| = \sum_{i=1}^{n} \mathbf{1}_{[S_i = j]}$.

Model-based clustering with Bayes

We have

$$A_{n,k} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^j \binom{k}{j} (k-j)^n$$

(Stirling number of the second kind) ways to partition n elements in k groups, and

$$B_n = \sum_{k=1}^{n} A_{n,k}$$

(Bell number) ways to partition n elements, which explodes as far n grows.

Scan the entire partitions space is unfeasible in a reasonable time, as far as $n \nearrow$.

n	1	2	3	4	5	6	7	8	9	10
B_n	1	2	5	15	52	203	877	4140	21147	115975

In practice, we produce a sample of the posterior distribution obtaining a set of partitions which are representative of the latent cluster of the data.

Model-based clustering with Bayes

While the previous augmentation has a nice interpretation, in practice is more convenient to work with another one.

A second augmentation

We can consider a **sequence of latent indicators** $S_1, \dots, S_n$, one for each **observations** and describing which component is the observation associated to, having

$$\begin{aligned} \boldsymbol{Y}_i \mid S_i, \boldsymbol{\theta}^*_{1:k} &\sim k(\boldsymbol{y}_i, \boldsymbol{\theta}^*_{S_i}), \qquad i = 1, \dots, n \\ S_1, \dots, S_n \mid \boldsymbol{w} &\overset{\text{iid}}{\sim} Cat(w_1, \dots, w_k), \\ \boldsymbol{w} &\sim Dir(\boldsymbol{\alpha}), \\ \boldsymbol{\theta}^*_1, \dots, \boldsymbol{\theta}^*_k &\overset{\text{iid}}{\sim} \pi(\boldsymbol{\theta}). \end{aligned}$$

- Two observations i and j, with $i \neq j$, are in the same cluster if they are associated to the same components, i.e. if $S_i = S_j$.
- This different augmentation, even if is not easily interpretable in terms of latent parameter, is quite useful to perform posterior inference with these models.

Model-based clustering with Bayes

Also for the second augmentation, we can easily see that marginalizing with respect to a generic S_i the joint distribution of $\boldsymbol{Y}_i, S_i \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k}$, we recover the distribution of $\boldsymbol{Y}_i \mid \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k}$.

Model-based clustering with Bayes

With the previous model setting, we are able to derive the full conditional distributions of **the weights $\boldsymbol{w}$**, the **group-specific parameters $\boldsymbol{\theta}^*_{1:k}$**, and the augmented variables.

Let us assume a mixture model with a generic kernel function $k(\boldsymbol{y}, \boldsymbol{\theta})$, a Dirichlet prior for the weights $\boldsymbol{w} \sim Dir(\boldsymbol{\alpha})$, and a suitable prior for the group-specific parameters $\pi(\boldsymbol{\theta})$. Then we have

$$
\begin{aligned}
S_i \mid \boldsymbol{y}_i, \boldsymbol{w}, \boldsymbol{\theta}^*_{1:k} &\overset{\text{ind}}{\sim} Cat(w_1 k(\boldsymbol{y}_i, \boldsymbol{\theta}^*_1), \ldots, w_k k(\boldsymbol{y}_i, \boldsymbol{\theta}^*_k)), \qquad i = 1, \ldots, n, \\
\boldsymbol{w} \mid S_{1:n} &\sim Dir(\alpha_1 + n_1, \ldots, \alpha_k + n_k), \\
\boldsymbol{\theta}^*_j \mid S_{1:n}, \boldsymbol{y}_{1:n} &\overset{\text{iid}}{\sim} \pi(\boldsymbol{\theta}^*_j \mid S_{1:n}, \boldsymbol{y}_{1:n}) \propto \pi(\boldsymbol{\theta}^*_j) \prod_{i:S_i=j} k(\boldsymbol{y}_i, \boldsymbol{\theta}^*_j), \qquad j = 1, \ldots, k,
\end{aligned}
$$

where $n_j = \sum_{i=1}^{n} \mathbf{1}_{[S_i=j]}$ is the number of observations assigned to the jth component of the mixture.

The previous can guide us in the specification of the prior distributions parameters, in particular

- → The Dirichlet distribution should be symmetric, in force of the label switching, whereas each α_j can be interpreted as the **prior sample sizes** of the generic jth cluster.
- → The prior distribution $\pi(\boldsymbol{\theta}^*_j)$ depends on the specific kernel function, but usually is set to be vague or noninformative.

Model-based clustering with Bayes

Model-based clustering with Bayes

We can exploit the previous full conditionals to derive a sampling strategies to perform posterior inference with those models, by iteratively sampling the quantities we need.

Conditional algorithm to perform model-based clustering

Input: $\alpha_1, \ldots, \alpha_k$, parameter of $\pi(\theta_j^*)$. Initial values for $\boldsymbol{\theta}_{1:k}^*$.

for $r = 1$ *to* R **do**

  for $i = 1$ *to* n **do**

    Sample the **cluster allocation** of the ith observation, with

$$P(S_i = j \mid \boldsymbol{y}_{1:n}, \boldsymbol{w}, \boldsymbol{\theta}_{1:k}^*) \propto w_j k(\boldsymbol{y}_i, \boldsymbol{\theta}_j^*), \qquad j = 1, \ldots, k.$$

  end

  for $j = 1$ *to* k **do**

    Update the **cluster-specific parameters**, with

$$\pi(\boldsymbol{\theta}_j^* \mid y_{1:n}, S_{1:n}) \propto \pi(\boldsymbol{\theta}_j^*) \prod_{i:S_i=j} k(y_i, \boldsymbol{\theta}_j^*), \qquad j = 1, \ldots, k.$$

  end

  Update the **weights** with $\boldsymbol{w} \sim Dir(\alpha_1 + n_1, \ldots, \alpha_k + n_k)$, where n_j is the size of the generic jth cluster.

end

Model-based clustering with Bayes

Example

Let us consider the following data

```
set.seed(42)
y <- c(rnorm(50, -3, 1), rnorm(25, 3, 1))
```

We want to cluster those observations, assuming a mixture of univariate Gaussian, using a $N(\mu, \sigma^2)$ as kernel function. Further, we set as prior for the group-specific parameter

$$\mu \mid \sigma^2 \sim N(\mu, \sigma^2/k_0),$$
$$\sigma^2 \sim IG(a_0, b_0).$$

- Show that the previous prior assumption is conjugate for the Gaussian likelihood, with $\mu, \sigma^2 \mid y_{1:n} \sim NIG(m_n, k_n, a_n, b_n)$, where

$$k_n = k_0 + n, \qquad m_n = (k_0 m_0 + n\bar{y})/(k_0 + n)$$
$$a_n = a_0 + n/2, \qquad b_n = b_0 + \frac{1}{2}\left(\sum_{i=1}^{n}(y_i - \bar{y})^2 + \frac{k_0 n}{k_n}(\bar{y} - m_0)^2\right).$$

- implement a Gibbs sampler to sample from the posterior distribution of the latent partition in the data. Check the convergence of the algorithm by looking at the partition entropy at each iteration.

Summarizing our posterior inference

Model-based clustering with Bayes

We have a new problem, the so-called **label switching**.

Once we produce a sample from the posterior distribution of interest, i.e. a sample of latent partitions $\rho_1, \ldots, \rho_R$, we want to produce summaries of these sampled values, such as **a single point estimate**.

Data are assumed to be exchangeable, the model is invariant with respect to permutation of the data or the components.

Two observations can be both assigned to a specific component at a generic iteration r, and both to a different component at iteration $r + \ell$.

- From a clustering perspective, they belong to the same cluster, but from a label perspective they have different values.
- In our inferential procedure, we should take in account such a problem.

Summarizing our posterior inference

We can approach the problem from a decision theory point of view. Let $L(\cdot, \cdot)$ denotes a loss function, having two partitions of n elements as arguments. The **optimal partition** $\rho_n^* \in \mathbb{B}$ (or equivalently $(S_1^*, \ldots, S_n^*) \in \mathbb{N}^n$) is the solution of

$$\begin{aligned} \rho_n^* &= \arg\min_{\hat{\rho}_n \in \mathbb{B}} \{\mathrm{E}[L(\rho_n, \hat{\rho}_n) \mid \boldsymbol{y}_{1:n}]\} \\ &= \arg\min_{\hat{\rho}_n \in \mathbb{B}} \left\{ \sum_{\rho_n \in \mathbb{B}} L(\rho_n, \hat{\rho}_n) \mathrm{P}(\rho_n \mid \boldsymbol{y}_{1:n}) \right\} \end{aligned}$$

where ρ_n^* denotes the optimal partition, $\hat{\rho}_n$ the partition we are considering, ρ_n the partitions wrt we take the expectation, and $\mathrm{P}(\rho_n \mid \boldsymbol{Y})$ is the posterior probability of ρ_n, given a set of observations $\boldsymbol{y}_{1:n}$.

- The space $\mathbb{B}$ is too large
- We can choose $L(\cdot, \cdot)$ in several ways

Summarizing our posterior inference

A first loss function that we can consider is the $0-1$ loss function, i.e.

$$L_{0-1}(\rho_n, \hat{\rho}_n) = \mathbf{1}_{[\rho_n \neq \hat{\rho}_n]}$$

- Such loss function is taking value 1 when all the blocks of ρ_n and $\hat{\rho}_n$ coincide, and 0 otherwise.
- Is not accommodating possible similarities in the partitions, all the partitions different from $\hat{\rho}_n$ are penalized in the same way
- the point estimate coincides with

$$\arg\min_{\hat{\rho}_n \in \mathbb{B}} \left\{ \sum_{\rho_n \in \mathbb{B}} \mathbf{1}_{[\rho_n \neq \hat{\rho}_n]} \mathcal{L}(\rho_n = \rho_n \mid \boldsymbol{Y}) \right\} = \arg\max_{\hat{\rho}_n \in \mathbb{B}} \{\mathcal{L}(\rho_n = \hat{\rho}_n \mid \boldsymbol{Y})\}$$

  the maximum a posteriori of the distribution of ρ_n.

Summarizing our posterior inference

A more relaxed loss function: the Binder loss function.

$$L_B(\rho_n, \hat{\rho}_n) = \sum_{j<i} \left[C_1 \mathbf{1}_{[S_i = S_j]} \mathbf{1}_{[\hat{S}_i \neq \hat{S}_j]} + C_2 \mathbf{1}_{[S_i \neq S_j]} \mathbf{1}_{[\hat{S}_i = \hat{S}_j]} \right]$$

and by setting $C_1 = C_2$ it can be written as

$$L_B(\rho_n, \hat{\rho}_n) = \frac{1}{2} \left(\sum_{i=1}^{k_n} n_{i\bullet}^2 + \sum_{j=1}^{\hat{k}_n} n_{\bullet j}^2 - 2 \sum_{j=1}^{\hat{k}_n} \sum_{i=1}^{k_n} n_{ij}^2 \right)$$

It is based on the cluster frequencies, where

- k_n is the number of blocks in ρ_n
- $\hat{k}_n$ is the number of blocks in $\hat{\rho}_n$
- $n_{ij} = \sum_{\ell=1}^{n} \mathbf{1}_{[S_\ell = i]} \mathbf{1}_{[\hat{S}_\ell = j]}$ is the number of observations in the i-th block of ρ_n and in the j-th block of $\hat{\rho}_n$
- $n_{i\bullet} = \sum_{j=1}^{\hat{k}_n} n_{ij} = \sum_{\ell=1}^{n} \mathbf{1}_{[S_\ell = i]}$ is the number of observations in the i-th block of ρ_n
- $n_{\bullet j} = \sum_{i=1}^{k_n} n_{ij} = \sum_{\ell=1}^{n} \mathbf{1}_{[\hat{S}_\ell = j]}$ is the number of observations in the j-th block of $\hat{\rho}_n$

Summarizing our posterior inference

You can show that the solution of

$$\rho_n^* = \underset{\hat{\rho}_n \in \mathbb{B}}{\arg\min} \left\{ \sum_{\rho_n \in \mathbb{B}} L_B(\rho_n, \hat{\rho}_n) \mathrm{P}(\rho_n \mid \boldsymbol{y}_{1:n}) \right\}$$

is given by the partition which, in binary representation, minimize the distance from the posterior similarity matrix $\mathcal{M}$, where

$$\mathcal{M}_{ij} = P(S_i = S_j \mid \boldsymbol{Y})$$

and

$$\boldsymbol{S}^* = \underset{\hat{\boldsymbol{S}} \in \mathbb{N}^n}{\arg\min} \left\{ \sum_{i<j} |\mathbf{1}_{[\hat{S}_i = \hat{S}_j]} - \mathcal{M}_{ij}| \right\}$$

The Binder loss function, differently from the $0-1$ loss function, is not penalizing in the same way the partitions different from the optimal one.

Summarizing our posterior inference

Let's check the previous!

Summarizing our posterior inference

A practical example: consider $n = 3$, and all the possible partitions are

$$\mathbb{B} = \{\{1,2,3\},\{12,3\},\{1,23\},\{13,2\},\{123\}\}$$

Equivalently in a binary representation

$$W_1 = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & 0 & 1 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1 & & \\ 1 & 1 & \\ 0 & 0 & 1 \end{bmatrix}, \quad W_3 = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & 1 & 1 \end{bmatrix},$$

$$W_4 = \begin{bmatrix} 1 & & \\ 0 & 1 & \\ 1 & 0 & 1 \end{bmatrix}, \quad W_5 = \begin{bmatrix} 1 & & \\ 1 & 1 & \\ 1 & 1 & 1 \end{bmatrix}$$

and $\mathcal{M} = \begin{bmatrix} 1 & & \\ 0.8 & 1 & \\ 0.2 & 0.2 & 1 \end{bmatrix}$, then we have

$$d(W_1,\mathcal{M}) = 0.8^2 + 0.2^2 + 0.2^2 = 0.72; \quad d(W_2,\mathcal{M}) = 0.08; \quad d(W_3,\mathcal{M}) = 0.68;$$
$$d(W_4,\mathcal{M}) = 0.68; \quad d(W_5,\mathcal{M}) = 1.32,$$

and the point estimate is given by R_2.

Model-based clustering with Bayes

Upon the previous, we can build an algorithm that produce a point estimate of the latent partition, under the Binder loss function, starting from a MCMC sample from the posterior distribution of interest.

Point estimate under Binder loss function

Input: A sample of partitions from the posterior distribution $\{\rho^{(1)}, \ldots, \rho^{(R)}\}$.
Produce an estimate of the posterior similarity matrix $\mathcal{M}$, where

$$[\mathcal{M}]_{i,j} = \frac{1}{R} \sum_{r=1}^{R} \mathbf{1}_{[S_i^{(r)} = S_j^{(r)}]}$$

for $r = 1$ *to* R **do**

Reconstruct the binary matrix for the rth partition, with

$$[W_r]_{ij} = \begin{cases} 1 & : \ S_i^{(r)} = S_j^{(r)}, \\ 0 & : \ \text{otherwise} \end{cases}$$

Compute the squared distance d_r between $\mathcal{M}$ and W_r.

end

Return the partitions R_r that minimize the distance.

Summarizing our posterior inference

Example

Let us consider the previous data

```
set.seed(42)
y <- c(rnorm(50, -3, 1), rnorm(25, 3, 1))
```

with the same assumption of the previous example, i.e.using a $N(\mu, \sigma^2)$ as kernel function. Further, we set as prior for the group-specific parameter

$$\mu \mid \sigma^2 \sim N(\mu, \sigma^2/k_0),$$
$$\sigma^2 \sim IG(a_0, b_0),$$

that a posteriori leads to $\mu, \sigma^2 \mid y_{1:n} \sim NIG(m_n, k_n, a_n, b_n)$, where

$$k_n = k_0 + n, \qquad m_n = (k_0 m_0 + n\bar{y})/(k_0 + n)$$
$$a_n = a_0 + n/2, \qquad b_n = b_0 + \frac{1}{2}\left(\sum_{i=1}^{n}(y_i - \bar{y})^2 + \frac{k_0 n}{k_n}(\bar{y} - m_0)^2\right).$$

- Implement an R function to obtain a point estimate with the Binder loss function, starting from an MCMC output.
- Use the function to find the point estimate with the partitions sampled in the previous example.

The multivariate Gaussian case

The multivariate Gaussian case

A common case while dealing with multivariate real-valued data is to consider a **mixture of multivariate Gaussian distribution**. Suppose we observe a sample $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$. Here, we want to identify clusters of observations where the generic $\boldsymbol{y}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$.

In this framework, each term describing a cluster of data is given by a multivariate Gaussian component, with its own location and covariance matrix.

The kernel function is then given by the density function of a multivariate Gaussian distribution, which is

$$f(\boldsymbol{y}_i \mid \boldsymbol{\mu}, \Sigma) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\boldsymbol{y}_i - \boldsymbol{\mu})^\intercal \Sigma^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}) \right\}.$$

- Observations close to the centroid $\boldsymbol{\mu}$ belong to the cluster with high probability.
- Implicitly, we are looking at the Mahalanobis distance among observations belonging to the same cluster.
- The covariance matrix Σ drives the cluster dispersion and shape.

The multivariate Gaussian case

To implement the algorithm we saw earlier in this slide block, we need first to set a distribution on the cluster-specific parameters.

- **The location parameter $\boldsymbol{\mu}$** is a p-dimensional real-valued vector, hence we can set a multivariate Gaussian distribution.
- **The covariance matrix** is a $p \times p$ positive definite real-valued matrix. A possible distributional assumption for this parameter is given by the inverse-Wishart distribution.

Let X be a $p \times p$ positive definite real-valued matrix. We say that X is distributed as an inverse-Wishart distribution, with ν_0 degrees of freedom and scale parameter Λ_0, if its density function corresponds to

$$f(\Sigma \mid \nu_0, \Lambda_0) = \frac{|\Lambda_0|^{\nu_0/2}}{2^{\nu_0 p/2}\Gamma_p(\nu_0/2)}|\Sigma|^{-(\nu_0+p+1)/2}\mathrm{e}^{-\frac{1}{2}\mathrm{tr}(\Lambda_0\Sigma^{-1})},$$

where $\Gamma_p(\nu_0/2) = \pi^{p(p-1)/4}\prod_{j=1}^{p}\Gamma(\nu_0/2+(1-j)/2)$ is the mutivariate gamma function.

Note that, with the previous distributional assumption,

$$\mathrm{E}[\Sigma] = \Lambda_0/(\nu_0 - p - 1),$$

defined for $\nu_0 > p + 1$. This might help us to specify the parameters ν_0 and Λ_0.

The multivariate Gaussian case

Specifically, as prior assumption for the main parameter of the model, we consider an hierarchical specification of the form

$$\begin{aligned} \boldsymbol{\mu} \mid \Sigma &\sim N(\boldsymbol{m}_0, \Sigma/k_0), \\ \Sigma &\sim IW(\nu_0, \Lambda_0). \end{aligned}$$

which is called **normal-inverse-Wishart distribution**.

Let us assume the data Gaussian distributed. Under the previous prior assumption, we have

$$\begin{aligned} \boldsymbol{\mu} \mid \Sigma &\sim N(\boldsymbol{m}_n, \Sigma/k_n), \\ \Sigma &\sim IW(\nu_n, \Lambda_n), \end{aligned}$$

with

$$k_n = k_0 + n, \qquad \boldsymbol{m}_n = (k_0 \boldsymbol{m}_0 + n\bar{\boldsymbol{y}})/k_n$$

$$\nu_n = \nu_0 + n, \qquad \Lambda_n = \Lambda_0 + \sum_{i=1}^{n} (\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^{\mathsf{T}} + \frac{k_0 n}{k_n} (\bar{\boldsymbol{y}} - \boldsymbol{m}_0)(\bar{\boldsymbol{y}} - \boldsymbol{m}_0)^{\mathsf{T}}.$$

Hence, the normal-inverse-Wishart prior assumption is conjugate to the multivariate Gaussian likelihood.

The multivariate Gaussian case

The multivariate Gaussian case

The multivariate Gaussian case

The multivariate Gaussian case