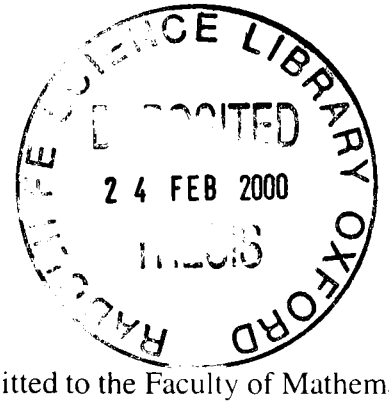
Predictive Classification using Mixtures of Normal Distributions

Rafael Perera Salazar

Wolfson College, Oxford

Michaelmas Term, 1998



A thesis submitted to the Faculty of Mathematical Sciences
for the degree of Doctor of Philosophy of the University of Oxford.

Abstract

Predictive Classification using Mixtures of Normal Distributions

Rafael Perera Salazar
Wolfson College, Oxford

D. Phil. Thesis
Michaelmas Term, 1998

Classification using mixture distributions to model each class has not received too much attention in the literature. The most important attempts use normal distributions as components in these mixtures. Recently developed methods have allowed the use of these kinds of models as a flexible approach for density estimation. Most of the methods developed so far use plug-in estimates for the parameters and assume that the number of components in the mixture is known.

We obtain a predictive classifier for the classes by using Markov Chain Monte Carlo techniques which allow us to obtain a sampling chain for the parameters. This fully Bayesian approach to classification has the advantage that the number of components for each class is taken as another variable parameter and integrated out of the classification. To achieve this we use a birth-and-death/Gibbs sampler algorithm developed by Stephens (1997).

We use five different datasets, two simulated ones to test the methods on a single class and three real datasets to test the methods for classification. We look at different models to define which gives better flexibility in the modelling and an overall better classification. We look at different types of priors for the means and dispersion matrices of the components. Joint conjugate priors and an independent conjugate priors for the means and dispersion matrices for the components are used. We use a model with a common dispersion matrix for all the components and another one with a reparametrisation of these dispersion matrices into size, shape and orientation (Banfield and Raftery (1993)). We allow the sizes to differ while keeping a common shape and orientation for the dispersion matrices of the components in a class.

We found that this type of modelling with independent conjugate priors for the means and dispersions while allowing the sizes of the dispersions to vary gave the best results for classification purposes as it allowed great flexibility and separation between the components of the classes.

Acknowledgements

I wish to thank Professor Brian Ripley and Dr. F. H. C. Marriott for their help and supervision of this thesis. Special thanks to Matthew Stephens who allowed me access to his code as well as several very useful conversations. I want to thank all my friends in Wolfson College and in the department of Statistics for making this period of my life unforgettable. Most of them have gone now to different parts of the world (Switzerland, Greece, Australia, Brussels, the States, etc.) so I guess they will have an excuse not to read this.

I would like to tank(!) all of Wolfson rowing and Football club who helped me go through these years keeping my sanity. I would like to thank specially Lars, Liz, Mark, Meghan, Phil, Sean, Derek, Anna, Emma, and Mike who had the patience to teach, cox, row and be coached by me. I specially would like to thank Matt Roberts for an incredible piece of writing about rowing which will aid my memory of "how it used to be".

When I am in Mexico playing in the grass-free football pitches I will look back at the times when being tackled or tackling was not a dangerous affair. I want to thank Franco, Matt, Martin, Duncan, Dave B. and Steve for allowing me to play in Oxford first division and always believing I was a much better player than I actually am. Also tennis in the parks will be one of the best memories that I take away from here (thanks Andy).

I will miss the Stats department for people like Renee, Hamid, Matthew and Erich, as well as Susan, Matthew E., Dave F., and Mario. I hope I can come back to visit them often (and pester them some more!).

A special thank to Nik and Rich for their special quirks and ways of viewing life, their amazing knowledge and their good heart. I miss you two already.

A huge thanks to my family. My mom, Gaby, Adrian, Jan, David, Mike, aunt Mela, uncle Bernardo, aunt Raquel, uncle Eliud, aunt Maria, Nano, Rosy, Ivonne, Victor, and Licha. They have always given me all the support I have needed.

I know I've left out so many people that have helped to make these years incredible I hope

they forgive me.

I also want to thank CONACyT for providing the funding that made possible this Doctorate.

Last here but first in my heart, thanks to Theresa. Her guidance, wisdom and patience has made this common goal possible. I take away from Oxford the best thing it has given me.

Contents

1 Introduction to Mixture Models **1**

1.1 Brief History and Motivation for the use of Mixtures 1

1.2 Finite Mixtures Setup and Notation 3

1.3 Definition of Statistical Problems to be addressed 4

2 Terminology and Assumptions **8**

2.1 Notation and Model 8

2.2 Bayesian Modelling: a Predictive Approach 9

3 Modelling a Single Class as a Mixture **20**

3.1 Description of a class as a mixture of multivariate normal distributions . . 20

3.2 Choice of priors for classes with common precision 21

3.3 Initial values and hyperparameters for classes with common precisions 25

3.4 Modelling for classes with unknown number of components 29

3.5 Variable against fixed number of components 49

3.6 Using precision matrices with different sizes 58

4 Predictive Classification using Mixture Models **71**

4.1 Crabs Data: *Leptograpsus variegatus* 73

4.2 Pima Indians Data 78

4.3 Forensic glass Data 90

5 Conclusions and Further Work **111**

5.1 Further Work 113

A Description of datasets used **115**

B Gibbs sampling μ_i and Λ **125**

Chapter 1

Introduction to Mixture Models

1.1 Brief History and Motivation for the use of Mixtures

The idea of using distribution mixtures for modelling is not a new one. Newcomb's (1886) application of normal mixtures as models for outliers and Pearson's (1894) decomposition of normal mixtures by the method of moments are two early works in the area. This type of modelling arises naturally in situations when the measurements of a random variable can be viewed as coming from two or more populations mixed in varying proportions. Mixtures are used in cases when measurements are available from experimental units which are known to belong to one set of classes, but whose individual class–memberships are unavailable; this is commonly called a *direct application* of mixture models. The term *indirect application* is used in the cases where mixtures are used as a mathematical device to obtain a flexible form of analysis.

Everitt and Hand (1981), Titterington *et al.* (1985), and McLachlan and Basford (1988), are good references on mixtures. In the case of McLachlan and Basford (1988) their main interest is in clustering (classification) so a physical interpretation is given to each component. Recent works in the area are Robert (1996), West (1997), Stephens (1997), and the papers by Diebolt and Robert (1994), Escobar and West (1995), and Richardson and Green (1997) to mention a few.

The main objective in the research is the use of finite mixture distributions as a tool to obtain an accurate classification of a dataset. This falls mainly into the *indirect application* for the use of mixtures. Specifically we are interested in the multivariate case. There are different possible sampling schemes to consider, but we will consider only the case when a sample of items all assigned to a specific class is available and all the classes are known. The information taken from this training sample is used to classify subsequent

observations. For this reason we are interested in using mixtures as a tool for density estimation of a fairly complex parametric space.

In most cases mixture models are used for a *direct application* as the intent is on finding the class-membership of datapoints to different components in the mixture. In our case we are interested in modelling *each class* as arising from a mixture with several components (k generally unknown) and for that reason in most cases a physical explanation of these components in a class might prove imposible. Up until recently modelling through mixtures has provided great flexibility, but its application has proved to be limited for two main reasons. Firstly, an explicit formula for the estimation of its parameters generally does not exist, so that numerical methods are required. Secondly, theoretical difficulties arise in certain aspects of the statistical analysis. Some of these problems are discussed in this chapter and will be addressed through the thesis. We believe that now it is possible to overcome most of these difficulties and an adequate model can be obtained using these techniques.

These types of models are not new. McLachlan (1992), Cheng and Titterington (1994), and Hastie and Tibshirani (1996) are a few of the authors who use mixtures to model classes to obtain a classification. Nevertheless in all these cases the models created do not take into consideration the "parameter uncertainty". All these methods use *plug-in* estimators in order to obtain an approximation to the probability density of the classes in order to classify a new datapoint. I believe this approach to be wrong as these point estimates can be misleading (if not totally wrong). As Stephens (1997) writes "*..[for mixtures] there may be several reasonable local maxima between which to choose, each of which may give quite different plug-in estimates for quantities of interest such as density or classification probabilities*". For this reason we intend on following a fully Bayesian approach in our modelling and classification, besides the fact that this type of modelling has indeed proven to be a more flexible tool than alternative methods.

In this chapter we present the use of finite mixture models to describe a class and discuss some of the statistical problems that arise from their use. Chapter 2 deals with the model we intend using for classification from a fully Bayesian approach. The general model and the algorithm used is also described in this chapter. We mention the problems that arise from the use of this type of methods as well as discussing some of the solutions that are proposed to solve them. In chapter 2 we also discuss the assumptions taken for this type of models for different setups.

In chapter 3 we look at single classes and assess our models (under different assumptions) and the success while using them to obtain a predictive density that describes the data. For example we look at different types of model for the dispersion matrix for each component, constraining all dispersion matrices in a class to be the same and allowing the size of each dispersion matrix to vary while maintaining the shape and orientation equal for all components in a class. We also look at the case when the number of components in a

class k_c is not known. We show why the use of Bayes Factors obtained using a harmonic mean estimator does not give consistent results. We also discuss the selection of values for some hyperparameters that can give an idea where the true number of components in a class can be and show that in general models with variable number of components k_c will produce a better classification as a result of better mixing of the sampler.

These types of methods will then be taken into chapter 4 and using these density estimates for each class obtain a predictive classification for a new datapoint (or a set of new datapoints). All the different setups are discussed and a model is recommended for general use.

All methods in this thesis use a Bayesian approach (predictive). Although most of the parameter estimation can be obtained with more traditional methods using plug-in estimates (EM algorithm), the use of structured priors (hierarchical models), variable number of components for each class (k_c), and the integration of the parameters out of the classification model (to account for parameter uncertainty) point towards the use of the Bayesian methods as the appropriate approach to the problem. Also the use of a variable k_c could help in the robust modelling for the cases in which outliers are present disturbing the parameter estimation. The datasets used in this thesis are discussed in detail in appendix A.

1.2 Finite Mixtures Setup and Notation

Multivariate observations on a set of n instances forming a set can be represented as $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, where each $\boldsymbol{x}_j$ $(j = 1, \ldots, n)$ is a vector of p dimensions, where p is equal to the number of attributes measured for a specific instance. Assume that a random vector $\boldsymbol{x}_j$ has a probability density function $g(\cdot;\cdot)$ (we will use the same notation for discrete cases). Based on some considerations we wish to model this p.d.f. as a mixture of k components. So the p.d.f. can be written as a finite mixture

$$g(\boldsymbol{x};\boldsymbol{\phi}) = \sum_{i=1}^{k} \pi_i f_i(\boldsymbol{x};\boldsymbol{\theta}_i),$$

where $\sum_{i=1}^{k} \pi_i = 1$, with $\pi_i > 0$, and $f_i(\cdot)$ is a probability density function with parameters represented by $\boldsymbol{\theta}_i$. This is equivalent to assuming that $g(\cdot;\cdot)$ is a p.d.f. of a superpopulation K consisting of k populations $K_1, ..., K_k$ in some proportions $\pi_1, ..., \pi_k$ respectively, also called *mixing weights*, and the i-th component has $f_i(\cdot)$ as its p.d.f..

Although the component densities can vary completely from group to group, in this thesis (as it is in most applications) these densities will be taken as if coming from the same

family of densities (e.g. normal, multinomial, etc.) with different parameter vectors $\boldsymbol{\theta}_i$. That is, $\boldsymbol{\theta}$ is the matrix defined by $\boldsymbol{\theta}^T = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k)$, of all unknown parameters associated with the component density functions and together with the mixing weights $\boldsymbol{\pi}$ ($\boldsymbol{\pi}^T = (\pi_1, \ldots, \pi_k)$) make the set of the total unknown parameters in the mixture denoted by $\boldsymbol{\phi} = (\boldsymbol{\pi}, \boldsymbol{\theta})$. We have omitted the number of components, k, from $\boldsymbol{\phi}$; this is because specific problems arise when trying to estimate $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ while assuming k is unknown. Later in the thesis (chapter 2) we included k as one of the variable parameters in the model.

One of the assumptions that will be made throughout this thesis is that the instances $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ are realised values of n independently and identically distributed (i.i.d.) random variables with common distribution $g(\boldsymbol{x}; \boldsymbol{\phi})$. This is appropriate in many situations in practice (but definitely not always) and simplifies the analysis.

1.3 Definition of Statistical Problems to be addressed

1.3.1 Unknown Parameters

Let us assume that the number of components in a mixture k is known. As we have seen in section 1.2, the probability density function of an observation $\boldsymbol{x}$ can be represented in the finite mixture form,

$$g(\boldsymbol{x}; \boldsymbol{\phi}) = \sum_{i=1}^{k} \pi_i f(\boldsymbol{x}; \boldsymbol{\theta}_i),$$

where all $f(\boldsymbol{x}; \boldsymbol{\theta}_i)$ $(i = 1, \ldots, k)$ belong to the same family of densities and $\boldsymbol{\theta}$ denotes the matrix of all unknown parameters associated with the parametric forms adopted for these k component densities. This matrix will have $k \times (\#$ *unknown parameters in distribution* $f)$ components. For example in the case of a univariate normal distribution $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ represent the vectors (of size k) of means and standard deviations for the components of the mixture.

There are some situations in which we are only interested in estimating a subset of $\boldsymbol{\phi}$. For the cases where the component densities $f_i(\cdot)$ are known, inference on the mixing weights $\pi_1, \ldots, \pi_k$ would be our only concern. In other applications the parameters of the components are unknown while the mixing weights are known. Nevertheless in most cases the whole vector $\boldsymbol{\phi}$ is unknown. Our discussion focuses on solutions to this case.

There are a large variety of estimation methods that have been applied to finite mixture problems. Recently the ones that have given the best results have been methods using the EM–algorithm and Monte Carlo Markov chain methods. With the EM-algorithm we

can obtain a maximum likelihood estimate for ϕ. Nevertheless using this method we cannot be sure that the estimate obtained is a global maximum or just a local maximum (this is particularly difficult for the case of mixture models). We will use instead MCMC methods (specifically Gibbs sampler) to obtain sampling chains for these parameters ϕ that will then be integrated out to obtain our classification.

1.3.2 Number of Components

So far most of the applications assume that the number of components is known. From the Bayesian point of view the number of components k is just another parameter which in some cases happens to be fixed. This approach will allow us to use an algorithm with variable k to obtain sampling chains for k and ϕ that will then be integrated for our classification (chapter 3).

In general the number of components k is unknown. In fact from the point of view of clustering this is actually the problem to solve. Nevertheless testing for the number of components in a mixture is a problem which has not been completely resolved. The approach of using the log–likelihood ratio test statistic λ to test for the smallest value of k compatible with the data is not straightforward in the case of mixture model. For this type of models the regularity conditions do not hold for $-2\log\lambda$ to have its usual asymptotic null distribution of chi-squared with degrees of freedom equal to the difference in the number of parameters in the two hypotheses. For example in the case of testing a mixture of two univariate normal distributions

$$g(\boldsymbol{x};\phi) = \pi_1\mathcal{N}(\mu_1,\sigma_1^2) + \pi_2\mathcal{N}(\mu_2,\sigma_2^2),$$

the null hypothesis of having only one distribution $H_0 : k = 1$ can be seen as testing if $\pi_1 = 1$, which is on the boundary of the parameter space where the standard regularity conditions break down. The same test can be expressed as testing whether $\mu_1 = \mu_2$ and $\sigma_1^2 = \sigma_2^2$. If no constraints on π, μ_i, or σ_i^2 are used, then H_0 can be stated as either $\pi_k = 0$ or $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$ so that the degrees of freedom for the log–likelihood ratio test are not well defined.

A number of authors have tried several modifications to the likelihood ratio statistic. One of the first approaches was made by Wolfe (1971). On the basis of a small–scale simulation on normal distributions he proposed that the null distribution of $2\log\lambda$ would be

$$-2c\log\lambda \sim \chi^2_d,$$

where d is equal to the degrees of freedom given by twice the difference in the number of parameters in the two hypotheses not including the mixing proportions, and the constant c is defined as $c = \frac{1}{n}(n - 1 - p - \frac{1}{2}k_2)$, where k_2 is the number of components of the

alternative hypothesis ($k_1 < k_2$). Hernandez-Avila (1979) agreed with Wolfe's proposal after more simulations. Everitt (1981) suggests that for this proposal to work, the ratio n/p needs to be at least ten (for $k_1 = 1$ and $k_2 = 2$) with $n > 50$. McLachlan (1987) tested Wolfe's criterion and decided that it is not suitable in the heteroscedastic case. Based on this McLachlan and Basford (1988) suggest that Wolfe's proposal should not be rigidly interpreted.

Other proposals have arisen using different methods like Akaike's information criterion (Bozdogan and Sclove (1984)) or the bootstrap (McLachlan (1987)). Akaike's information criterion relies essentially on the same regularity conditions needed for $-2\log\lambda$ to have its usual asymptotic distribution under the null hypothesis. Bootstrapping has the disadvantage that the amount of computation involved is considerably large even for simple problems; apart from that it is a real possibility for testing the number of components. Methods for special cases have been used as well. For example Titterington (1981) studied the mixture of two known densities in which the only unknown parameter is the mixing proportion π_1. He uses the likelihood ratio test of $H_0 : k = 1$ ($\pi_1 = 1$) versus $H_1 : k = 2$ ($\pi_1 < 1$) and concludes that "far from approaching a χ^2 random variable, $-2\log\lambda$ is, asymptotically, zero with probability $\frac{1}{2}$, under H_0" Other methods have been proposed using a Bayesian approach with degenerate priors for the means or restricting the values of the mixing proportions by taking truncated priors for them. The use of Bayes factors for the selection of the number of components k is discussed in section 3.4.1.

1.3.3 Identifiability (switching labels)

In the finite mixture

$$g(\boldsymbol{x};\boldsymbol{\phi}) = \sum_{i=1}^{k} \pi_i f_i(\boldsymbol{x};\boldsymbol{\theta}_i),$$

there exists a problem of identifiability of the components of the mixture. In this case if $f_i(\boldsymbol{x};\boldsymbol{\theta}_i)$ and $f_j(\boldsymbol{x};\boldsymbol{\theta}_j)$ belong to the same parametric family, then $g(\boldsymbol{x};\boldsymbol{\phi})$ will have the same value when the component labels i and j are interchanged in $\boldsymbol{\phi}$. In this sense, although the class of mixtures may be identifiable, the $\boldsymbol{\phi}$ is not. This is required if our aim is to obtain an estimate of the parameters $\boldsymbol{\phi}$ from $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$. In this case $g(\boldsymbol{x};\boldsymbol{\phi})$ is invariant under the $k!$ permutations of the component labels in $\boldsymbol{\phi}$, and if $\widehat{\boldsymbol{\phi}}$ corresponds to a local maximum, $L(\boldsymbol{\phi})$ will have exactly $k!$ local maxima of the same value.

This lack of identifiability can be overcome in practice by the imposition of an appropriate constraint on the parameters. Nevertheless constraining can cause bias in our estimations. Diebolt and Robert (1994) have proposed a selection of truncated priors for the mixing proportions while Mengersen and Robert (1995) and Robert (1996) propose a parameter-

isation of the form

$$\pi N(\mu, \sigma^2) + (1 - \pi)N(\mu + \sigma\theta, \sigma^2\tau^2),$$

with $\tau^2 < 1$, for the case of the mixture of two normal distributions. In general any restrictions on the parameters will reflect in higher autocorrelations in the sampling chain and possibly bad mixing. An alternative proposal has been made by Stephens (1997) in which having once obtained the sampling chain without any restrictions, a 'reordering' of the samples is made by means of a k–means type clustering algorithm. This has been applied to the univariate case so far with encouraging results.

The problem of multimodality of the likelihood due to the identifiability problem can be specially complicated for the multivariate case. Nevertheless we are not interested in obtaining estimates for the parameters (ϕ), instead these parameters will be integrated out from our model in order to take account of their uncertainty. This means that the lack of identifiability of components is for our specific case not a problem, and knowing the component labels is not necessary as we will discuss in section 2.2.

Chapter 2

Terminology and Assumptions

2.1 Notation and Model

The main objective is to classify observed multivariate data (p variables) into one of a number of classes $c = 1, \ldots, C$ (classes are known). This will be achieved with the use of a training set (already classified observations) and a test set (unclassified observations). This training set consists of n_T observations of p variables each. We will assume that the observations x_j are i.i.d. random variables. Each class has a total of n_c observations in the training set ($\sum_{c=1}^{C} n_c = n_T$). Each independent class will be described with a mixture model of the form

$$p_c(\boldsymbol{x}_c; \boldsymbol{\phi}_c) = \sum_{i=1}^{k_c} \pi_{ic} f_{ic}(\boldsymbol{x}_c; \boldsymbol{\theta}_{ic}),$$

where $p_c(\cdot;\cdot)$ is the true distribution for class c, $\boldsymbol{x}_c$ are the data belonging to class c, $\boldsymbol{\phi}_c$ are the parameters of distribution $p_c(\cdot;\cdot)$, k_c are the number of components in class c, π_{ic} is the proportion of component i in class c, $f_{ic}(\cdot;\cdot)$ is the distribution of component i in class c, and $\boldsymbol{\theta}_{ic}$ are the parameters for component i in class c. Under this notation the set of parameters used to describe class c will be $\boldsymbol{\phi}_c = k_c, \boldsymbol{\pi}_c, \boldsymbol{\theta}_c$. In some cases k_c is known and for these cases it will be treated as fixed but in general we will assume that k_c is unknown.

For this thesis the distribution used to represent each component $f_{ic}(\cdot;\cdot)$ will be the multivariate normal distribution. The explanation of the components within a class as real subclasses (direct application) is not necessarily the case; instead these can be seen as artificial ways of separating the classes for an easier description as a semi–parametric model. The use of multivariate normal mixtures is followed due to the flexibility of these models and the ease of its implementation (use of conjugate and non-conjugate priors).

This means that

$$f_{ic}(\cdot;\theta_{ic}) = \mathcal{N}_p(\cdot;\mu_{ic},\Sigma_{ic}),$$

with $\theta_{ic} = (\mu_{ic},\Sigma_{ic})$, where μ_{ic} = vector of size p and Σ_{ic} = dispersion matrix of size $p \times p$. Each class is modelled independently, nevertheless most of the methodology could be used to model the classes jointly.

We use two models, one that assumes all the dispersions for the components in *each class* to be equal $\Sigma_{ic} = \Lambda_c^{-1}$ (Λ = precision matrix) and another that assumes these dispersions to have the same shape and orientation but different sizes (following Banfield and Raftery (1993)) $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ A directed acyclic graph of these models (using joint-conjugate priors for the means and precisions) can be seen in figures 2.1 and 2.2. We discuss both models using different priors in detail in chapter 3 in which we model single classes.

2.2 Bayesian Modelling: a Predictive Approach

We will use a fully Bayesian approach for the classification of the observations by using the information contained in the training set to obtain a sampling chain from the posterior distribution of the parameters ϕ. Instead of estimating parameters and using them as 'plug-in's we integrate these parameters out by averaging to obtain estimates for the probabilities of a datapoint x_{new} being from class c. This way we are including the "uncertainty" of the parameters; from ignorance in the number of components in a class (k_c) to "uncertainty" in specific parameters.

For each of the classes, a Normal mixture model will be used. For part of the analysis a joint conjugate prior Normal–Wishart $(\mu_{ic},\Lambda_{ic} \mid \xi,\tau(\Lambda_{ic}^{-1}),\nu,\Omega)$ for the means and dispersion matrix of each component is used (Robert (1996)). We will also use independent conjugate priors for the means $\mathcal{N}_p(\mu_{ic} \mid \xi,\kappa)$ and precision matrix $\mathcal{W}_p(\Lambda_{ic} \mid \nu,\Omega)$ (West *et al.* (1994)). We use a Dirichlet prior for the mixing proportions π. As we mentioned in section 2.1 we either take all components in a class to have the same dispersion matrix or we allow the sizes of these dispersions to be different while keeping the orientation and the shape the same.

Taking a single class c defined by the following mixture of normal distributions

$$p(x;\phi_c) = \sum_{i=1}^{k_c} \pi_{ic}\mathcal{N}_p(x;\theta_{ic}),$$

where $\phi_c = (k_c,\pi_{.c},\theta_{.c})$, k_c is equal to the number of components of class c, π_{ic} is the proportion of component i in class c and $\mathcal{N}_p(x;\theta_{ic})$ is the multivariate normal distribution (with p variates) of component i, with $\theta_i = \mu_i$ (mean), Σ_i(dispersion matrix). The probability of a new datapoint belonging to class c given a training set T is given by

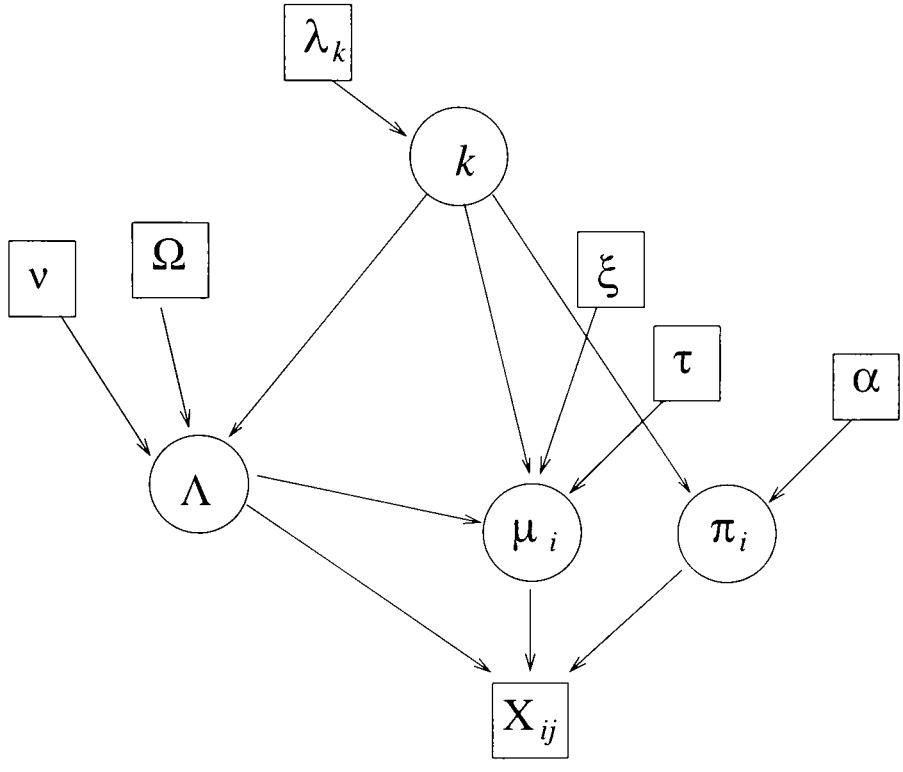


Figure 2.1: Directed acyclic graph of the parametric model used for mixtures of Normal distributions. This model takes into consideration only one class. The data X_{ij} is modelled as a mixture of k normal distributions with means μ_i, common covariance matrix $= \Lambda^{-1}$ for all i, and proportions π_i. The hyperparameters for this specific model are $\nu, \Omega, \xi, \tau, \alpha$, and λ_k. We discuss this model in more detail in chapter 3.

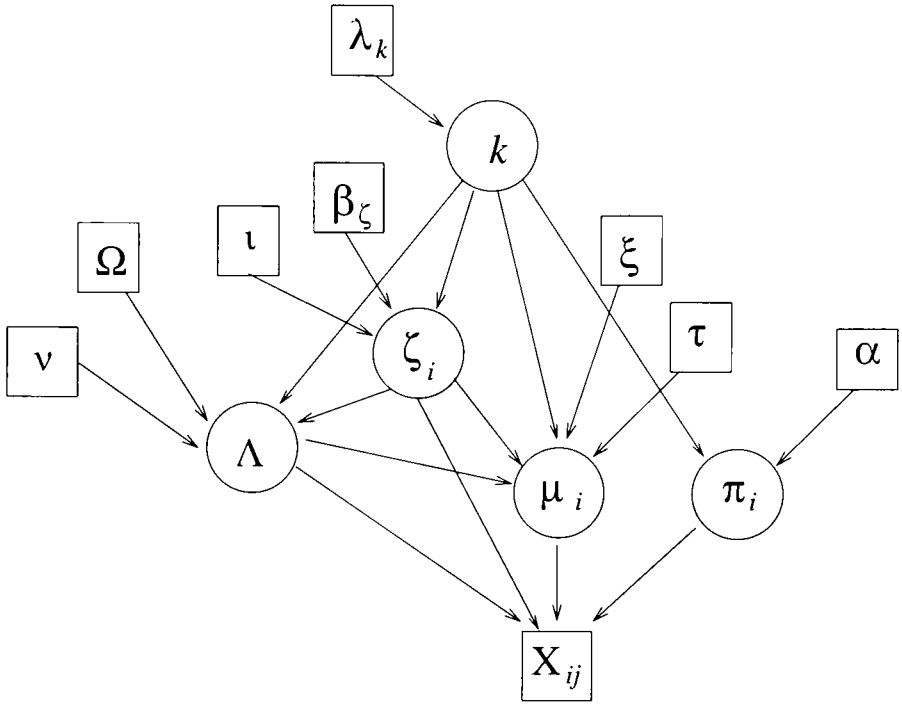


Figure 2.2: Directed acyclic graph of the parametric model used for mixtures of Normal distributions. This model takes into consideration only one class. The data X_{ij} is modelled as a mixture of k normal distributions with proportions π_i as in the model in figure 2.1 with means μ_i, but in this model the covariance matrix for component $i = (\zeta_i \Lambda)^{-1}$ instead. The hyperparameters for this specific model are $\nu, \Omega, \xi, \tau, \iota, \beta_\zeta, \alpha$, and λ_k. We discuss this model in more detail in chapter 3.

$$
\begin{aligned}
p(c \mid x_{new}, T) &\propto p(x_{new}, c \mid T), \\
&= p(c \mid T) p(x_{new} \mid c, T), \\
&= E(\lambda_c \mid T) p_c(x_{new} \mid T),
\end{aligned}
$$

where λ_c is the proportion of class c in the population under study. For reasons which we discuss in chapter 4, we will make the approximation $E(\lambda_c \mid T) \approx \frac{n_c}{n_T}$, the proportion of the training set in class c

For the models we consider in this thesis, conditional on knowing the class, z, of an observation x, x is independent of ϕ_c for $z \neq c$. (This assumption would be false for more general formulations, for example in a hierarchical structure with prior distributions on hyperparameters.) Thus

$$
p_c(x_{new} \mid T) = \int p(x_{new}; \phi_c) p(\phi_c \mid T) d\phi_c.
$$

where

$$
p(\phi_c \mid T) \propto p(\phi_c) \prod_{i=1}^{n_c} p(x_{j\{i\}}; \phi_c),
$$

with $x_{j\{1\}}, \ldots, x_{j\{n_c\}}$ denoting the observations in the training set from class c. The likelihood for class c will then be

$$
L(\phi_c, x) = \prod_{j=1}^{n_c} \sum_{i=1}^{k_c} \pi_{ic} f(x_j, \theta_{ic}). \tag{2.1}
$$

The sampling chains of the parameters involved in the model (π, μ, Σ) are obtained using Monte Carlo Markov Chain methods (in particular Gibbs sampler) described in section 2.2.2

We can divide the classification algorithm into two main stages. The first stage uses the training set (classified observations, no missing values) to obtain sampling chains for the parameters in the model using MCMC methods (Gibbs sampler). This first stage is the main object of our study as an accurate model (and classification) will depend on its effectiveness. Problems such as the choice of priors, the parameterisation used for the model, the duration of the 'burn-in' period (and convergence of the method to the posterior distribution) have to be dealt with in this stage. In this report we will address some of these problems like the priors used (section 3.3.1), the problem of convergence (section 2.2.3)

as well as describing an algorithm developed by Stephens (1997) that allows the use of finite mixtures when the number of components is unknown (section 3.4.1). The second stage is the use of these sampling chains of the parameters to 'integrate out' $\widetilde{p}(c \mid x)$ and obtain a predictive estimate of the classes which then is used for classification

$$\widehat{C}(x) = \begin{cases} l, & \text{if } p(l \mid x) = \max_c \widetilde{p}(c \mid x) \text{ and this exceeds } 1 - d, \\ D, & \text{if each } \widetilde{p}(c \mid x) \leq 1 - d. \end{cases}$$

where $\widetilde{p}(c \mid x)$ is the integrated probability of x belonging to component C, D would be the allocation of the datapoint to a new class, and $\widehat{C}(x)$ is our classifier, Ripley (1996). We will take $d = 1$ in all our discussion.

2.2.1 Problems with the Bayesian Approach

The Bayesian approach obtains posterior densities for the parameters from the product of the likelihood $L(\phi_c, x)$ and the prior distribution of the parameters $p(\phi_c)$. In this approach we need to specify a prior distribution for the parameters ϕ_c. For the case of mixture models the use of improper priors is not feasible as in the case when no elements from one of the components are sampled, the posterior will be improper (in this case the same as the prior). We will describe our choice of priors in section 3.3.1.

The implementation of the Bayesian approach has only recently been possible. To show the difficulty of using a fully Bayesian approach we will give an example for the univariate case (taken from Robert (1996)). Assume that the distributions of each component $f(x \mid \theta_i)$ are all from the same one–parameter exponential family

$$f(x \mid \theta_i) = h(x)e^{\theta_i x - \psi(\theta_i)}$$

with conjugate prior for θ_i

$$f(\theta_i \mid a_i, \lambda_i) \propto e^{\theta_i a_i - \lambda_i \psi(\theta_i)}$$

with a_i and λ_i real value constants.

A conjugate prior for the mixing proportions $\boldsymbol{\pi}^T = (\pi_1, \ldots, \pi_k)$ is a Dirichlet distribution $D(\alpha_1, \ldots, \alpha_k)$ with density

$$f^D(\pi_1, \ldots, \pi_k) \propto \pi_1^{\alpha_1 - 1} \ldots \pi_k^{\alpha_k - 1}$$

So the posterior distribution for the parameters $\boldsymbol{\phi} = (\boldsymbol{\pi}, \boldsymbol{\theta})$ given sample $(x_1, \ldots, x_n)$ is

$$f(\boldsymbol{\phi} \mid x_1, \ldots, x_n) \propto f^D(\pi_1, \ldots, \pi_k) \prod_{i=1}^{k} f(\theta_i \mid a_i, \lambda_i) \prod_{j=1}^{n} \left(\sum_{i=1}^{k} \pi_i f(x_j \mid \theta_i) \right). \qquad (2.2)$$

involving k^n terms of the form

$$\prod_{i=1}^{k} \pi_i^{\alpha_i + n_i - 1} f(\theta_i \mid a_i + n_i \bar{x}_i, \lambda_i + n_i)$$

where $n_1 + ... + n_k = n$ and $\bar{x}_i$ denotes the average of n_i of the n observations x_j.

Even in the case when it can be expressed in closed form, the computing time required to evaluate the k^n terms is too large to use brute force to implement a direct Bayesian approach. Nevertheless most of the k^n terms will be fairly improbable and there are some cases for which the number of terms can be reduced. For the direct application though, as the number of data increases, the calculation becomes more difficult. For example, a two component normal mixture cannot be estimated this way in a reasonable time for $n = 40$ observations (Robert (1995)). As we just mentioned most of these k^n terms will be fairly improbable, we use this fact in order to create algorithms that allow a reduction in the number of terms to be used. For example the work of Tanner and Wong (1987) and Gelfand and Smith (1990) lead to the application of MCMC techniques in mixture related problems which we describe in section 2.2.2.

2.2.2 MCMC methods: Gibbs Sampler Implementation

To obtain the posterior distribution of the parameters $\boldsymbol{\phi_c}$ we face an integration problem with high dimensionality (especially in the multivariate case). For this reason we use Monte Carlo Markov Chain (MCMC) methods. Besag *et al.* (1995) and Gilks *et al.* (1996) give a good description of these methods. The most commonly used are the Metropolis–Hastings algorithm (Metropolis *et al.* (1953), Hastings (1970)) and the Gibbs sampler named by Geman and Geman (1984). We will focus only on the latter algorithm. To implement the Gibbs sampler to solve the problem of estimating the parameters in a mixture, a variable z_j is defined which identifies to which of the components each of the observations belongs. The 'labels' are treated as missing data and sample values for them are obtained using the Gibbs sampler algorithm. Consider a sample $\boldsymbol{x}_1, ...\boldsymbol{x}_n$ from a finite mixture with k components

$$g(\boldsymbol{x}; \boldsymbol{\phi_c}) = \sum_{i=1}^{k_c} \pi_{ic} f(\boldsymbol{x}; \boldsymbol{\theta}_{ic}).$$

Under a hierarchical structure it is possible to rewrite the data vector $\boldsymbol{x}_j \sim g(\boldsymbol{x}; \boldsymbol{\phi_c})$ as $\boldsymbol{x}_j \sim f(\boldsymbol{x}; \boldsymbol{\theta}_{z_j c})$, where z_j is an integer identifying the component generating observation $\boldsymbol{x}$. This z_j will have a prior value i with probability π_i $(1 \leq i \leq k)$. In many cases the components may not be meaningful and the use of these labels may seem unnatural. For

the case of clustering the values of $z_1, ..., z_n$ are relevant and their sampled values can be used to estimate membership probabilities. In our case we are not interested in these values of z_j as we are using mixtures as a flexible approach for density estimation.

In case these missing data were available, it would be much easier to obtain estimates for the parameters in the mixture model (for a single class)

$$f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \mid z_1, \ldots, z_n) = \prod_{j:z_j=1} f(\boldsymbol{x}_j \mid \boldsymbol{\theta}_{1c}) \cdots \prod_{j:z_j=k} f(\boldsymbol{x}_j \mid \boldsymbol{\theta}_{kc}).$$

For the univariate case used in section 2.2.1 this would imply that:

$$f(\boldsymbol{\phi} \mid \boldsymbol{x}, \boldsymbol{z}) \propto \pi_1^{\alpha_1+n_1-1} \ldots \pi_k^{\alpha_k+n_k-1} f(\theta_1 \mid a_1 + n_1\bar{x}_1, \lambda_1 + n_1) \ldots$$
$$f(\theta_k \mid a_k + n_k\bar{x}_k, \lambda_k + n_k),$$

where now

$$n_i = \sum_j I(z_j = i), \qquad n_i\bar{x}_i = \sum_{j:z_j=i} x_j.$$

In this way, the decomposition permits the implementation of Gibbs sampling in two stages. Samples of the missing data $\boldsymbol{z} = (z_1, \ldots, z_n)$ and the parameters φ_c are alternated, producing a missing–data chain and a parameter chain. This special case of the Gibbs sampler was described by Tanner and Wong (1987) as the data augmentation algorithm. The data augmentation algorithm is run the following way: starting with initial value $\boldsymbol{\phi}^{(0)}$, at step m,

a) Generate $\boldsymbol{z}^{(m)} \sim f(z \mid \boldsymbol{x}, \boldsymbol{\phi}^{(m)})$,

b) Generate $\boldsymbol{\phi}^{(m+1)} \sim \pi(\boldsymbol{\phi} \mid \boldsymbol{x}, \boldsymbol{z}^{(m)})$.

In general, the Gibbs sampler algorithm allows any possible partition of $\boldsymbol{\eta} = (\boldsymbol{\phi}, z)$ into $(\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_t)$ and then iteratively generates $\boldsymbol{\eta}$ according to the conditional distributions

$$\boldsymbol{\eta}_1^{(m+1)} \sim f(\boldsymbol{\eta}_1 \mid \boldsymbol{x}, \boldsymbol{\eta}_2^{(m)}, \ldots, \boldsymbol{\eta}_t^{(m)}), \ldots, \boldsymbol{\eta}_t^{(m+1)} \sim f(\boldsymbol{\eta}_t \mid \boldsymbol{x}, \boldsymbol{\eta}_1^{(m+1)}, \ldots, \boldsymbol{\eta}_{t-1}^{(m+1)}).$$

Other implementations of Gibbs sampling generalise data augmentation by increasing the number of conditional simulations i.e. by breaking down $\boldsymbol{\phi}$ into subvectors. In mixture models, $\boldsymbol{\phi}$ will be separated into $\boldsymbol{\theta}$ (matrix of parameters in the distributions) and $\boldsymbol{\pi}$ (vector of proportions).

Algorithm 2.1. The Gibbs sampler algorithm used runs as follows: starting with initial value $\boldsymbol{\phi}^{(0)} = (\boldsymbol{\pi}^{(0)}, \boldsymbol{\theta}^{(0)})$, at step m,

a) Generate $\boldsymbol{z}^{(m)} \sim f(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\phi}^{(m)})$,

b) Generate $\boldsymbol{\pi}^{(m+1)} \sim f(\boldsymbol{\pi} \mid \boldsymbol{x}, \boldsymbol{z}^{(m)}, \boldsymbol{\theta}^{(m)})$.

c) Generate $\boldsymbol{\theta}_1^{(m+1)} \sim f(\boldsymbol{\theta}_1 \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\pi}^{(m+1)}, \boldsymbol{\theta}_2^{(m)}, \ldots, \boldsymbol{\theta}_k^{(m)})$,

$\boldsymbol{\theta}_2^{(m+1)} \sim f(\boldsymbol{\theta}_2 \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\pi}^{(m+1)}, \boldsymbol{\theta}_1^{(m+1)}, \boldsymbol{\theta}_3^{(m)}, \ldots, \boldsymbol{\theta}_k^{(m)})$,

$\boldsymbol{\theta}_k^{(m+1)} \sim f(\boldsymbol{\theta}_k \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\pi}^{(m+1)}, \boldsymbol{\theta}_1^{(m+1)}, \ldots, \boldsymbol{\theta}_{k-1}^{(m+1)})$,

For multivariate normal distributions, the estimation of vectors of means and covariance matrices for each of the components is required. For the cases when these parameters are not independent of each other, sampling is carried out from the conditional distributions. In our models we sample from the joint conditional for the means and the precision matrix or independently from the full conditional for the means and separately from the full conditional for the precisions. We also change the parameterisation for the precision matrix. As we mentioned in section 2.1 we use a common precision matrix for all components in a class or we allow the sizes of these precisions to vary keeping the shape and orientation equal for all the components in a class. These changes in models and parameterisations mean that step **c)** in our algorithm is modified in terms of a change in the full conditionals for our parameters. We discuss the full conditionals for step **c)** for the different models in chapter 3.

The conditionals for steps **a)** and **b)** are obtained as follows. Consider a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from the normal mixture model

$$\boldsymbol{x} \sim \sum_{i=1}^{k} \pi_{ic} \mathcal{N}_p(\boldsymbol{\mu}_{ic}, (\boldsymbol{\Lambda}_{ic})^{-1}),$$

where $\mathcal{N}_p(\cdot, \cdot)$ denotes a p dimensional multivariate normal distribution. We will use the precision matrix $\Lambda = \Sigma^{-1}$ for convenience in the programming and the notation. The corresponding conjugate prior distribution for π_c^k $(i = 1, \ldots, k)$:

$$(\pi_{1c}, \ldots, \pi_{kc}) \sim \mathbf{D}(\alpha_1, \ldots, \alpha_k).$$

Step **a)** derives the missing data $z_j (1 \leq j \leq n)$ from a simulated standard uniform random variable u_j, as follows: $z_j = i$ if $\varpi_{j1} + \cdots + \varpi_{j(i-1)} < u_j \leq \varpi_{j1} + \cdots + \varpi_{ji}$, where

$$\varpi_{ji} \propto \pi_{ic} \exp\{-(\boldsymbol{x}_j - \boldsymbol{\mu}_{ic})^T \boldsymbol{\Lambda}_{ic} (\boldsymbol{x}_j - \boldsymbol{\mu}_{ic})/2\} \mid \boldsymbol{\Lambda}_{ic} \mid^{\frac{1}{2}},$$

and

$$\sum_{i=1}^{k} \varpi_{ji} = 1,$$

using initial values for $\boldsymbol{\phi_c}$ in the first iteration.

Step **b)** of the Gibbs sampler algorithm simulates from the full conditional

$$(\pi_1, \ldots, \pi_k) \sim \mathrm{D}(\alpha_1 + n_1, \ldots, \alpha_k + n_k),$$

where n_i is the number of observations currently allocated to the i^{th} component.

2.2.3 Convergence of the Gibbs Sampler for Mixture Models

The Gibbs sampler has enabled us to reduce complex multidimensional problems to a sequence of much lower–dimensional ones. However, these methods suffer from a potentially serious drawback: it is difficult to decide when it is safe to terminate them and conclude their 'convergence' This is a more general notion of convergence than is usual for iterative procedures, because what is produced by the algorithm at convergence is not a single number or even a distribution, but rather a sample from a distribution. Worse still, the Markov nature of the algorithm means that the members of the sampled chain will generally be correlated with each other. This issue has been studied by Diebolt and Robert (1994) among others. They have reported that for 'almost' every starting point $\phi^{(0)}$, the general Gibbs sampling algorithm 'almost surely' converges.

The decision about the length of the 'burn-in' and the length of the chain (or chains) and how to sample form it (or them) are closely related to the problem of convergence. This issue has been studied in detail and several methods of testing the convergence of the chain have been developed. Cowles and Carlin (1996) give a survey of some of these methods and a comparison between them.

The most common approach in solving convergence has been to apply diagnostic tools to output algorithms. The problem with this approach is that the stationary distribution is always unknown in practice. For this reason, the diagnostic tool used often suggests convergence prematurely for slow mixing samplers, because it measures the distance separating the sampled distribution at two different iterations, either of the same chain or of two or more different chains, rather than the distance separating either distribution from the true stationary distribution. Nevertheless the reasoning of "a weak diagnostic is better than no diagnostic at all" seems to prevail.

Cowles & Carling recommend a combination of strategies: a) applying diagnostic procedures to a small number of parallel chains, b) monitoring autocorrelations and cross-correlations of long chains, and c) modifying parameterisation or sampling algorithms appropriately. These recommendations are not specific for mixture models but for the general case. As discussed in section 1.3.3 in mixture models the unconstrained posterior has $k!$ equal modes and the usual convergence diagnostics do not work if during the sampling a jump from one of these modes to another occur. So far work in this area has been

very limited for mixture models. Most methods for assessing convergence use the sampling chain and compare how this behaves using autocorrelations and cross–correlations for specific parameters. In the case of mixture models this type of methods fail if a jump between two components occurs or in the case when the number of components in a mixture (and the number of parameters) is not fixed. For good mixing to occur we need the sampler to jump between the different modes otherwise it is likely that we are not sampling from the posterior distribution.

Diebolt and Robert (1994) report a fast convergence (200 iterations) of mixture models (for a fixed number of components k). We have also seen this fast convergence for these cases with fixed components (given it is the right number!). As we mentioned in section 2.1 we are interested in obtaining a density estimate for each class and not in independent parameters. For this reason we use the following strategy to determine convergence:

1) Run a sampling chain one time and look at the likelihood chain to determine a burn-in period (when the likelihood appears to stabilise).
2) Set $I = 1000$.
3) Run 4 sampling chains with different starting points each. Each chain should be ran $itot = I$+burn-in times.
4) From these runs obtain the marginal predictive densities and compare them.
5) If very similar ($\|f_v^{itot} - f_b^{itot}\| < \epsilon$, where f_v^{itot}=marginal predictive density for starting point v and $itot$ iterations) then the process has converged otherwise $I = I + 500$ and go back to 3.
6) If convergence achieved then use any of the sampling chains for density estimation and classification.

We run 4 different sampling chains following Gelman (1996). As we mention in this section this convergence criteria can be over optimistic as we are comparing the distance of two sampling chains at a given number of iterations and not the target distribution. Nevertheless this can give a rough idea as to when our choice of initial values is no longer affecting the predictive density. The values used for ϵ were 0.02 for the forensic glass data and 0.01 for the rest of the data.

We use marginal predictive densities in our comparison instead of joint predictive densities as we can obtain a smoother estimate of the difference between marginal predictive densities than for joint densities (particularly if $p > 2$). We obtain the marginal predictive densities by choosing a certain number of points (m) in a region $[x_{min}, x_{max}]$ where to evaluate the density given the sampling chain. We estimate the density for each point by averaging over the sampling chain obtained (after burn-in). Say if we have a sample for the number of components $k_c^{(l)}$, the proportions $\pi_{1c}^{(l)}, \ldots, \pi_{k_c^{(l)}c}^{(l)}$, the means $\mu_{1c}^{(l)}, \ldots, \mu_{k_c^{(l)}c}^{(l)}$, and for the variances $\sigma_{1c}^{(l)}, \ldots, \sigma_{k_c^{(l)}c}^{(l)}$ for a given variable at iteration l with $l = 1, \ldots, m$ (as they are marginal predictive densities we are in the univariate case), then we obtain

the density for point x_d by

$$p(x_d \mid \phi_c^{(m)}) = \frac{1}{m} \sum_{l=1}^{m} \sum_{i=1}^{k_c^{(l)}} \pi_{ic}^{(l)} \mathcal{N}_1(x_d; \mu_{ic}^{(l)}, \sigma_{ic}^{(l)}),$$

where $x_{min} \leq x_d \leq x_{max}$.

As we will be using an algorithm that allows the number of components to vary (if the true number of components is unknown) and we assume that jumps between modes occur while we sample, this strategy seems to be optimal for assessing when the initial values are no longer affecting our density estimates and as a rough estimate to assess convergence.

Chapter 3

Modelling a Single Class as a Mixture

3.1 Description of a class as a mixture of multivariate normal distributions

As we mentioned in section 2.1 we will model each class independently (although they could be modelled jointly). For this reason in this chapter we discuss the algorithms that we will employ to obtain a density estimate for a particular class c. We will look at the full conditionals for the parameters θ_c to obtain step **c)** from the Gibbs sampler (algorithm 2.1). As we mentioned in chapter 2 each class will be described with a mixture model of the form

$$p_c(\boldsymbol{x}_c; \boldsymbol{\phi}_c) = \sum_{i=1}^{k_c} \pi_{ic} \mathcal{N}_p(\boldsymbol{x}_c; \boldsymbol{\theta}_{ic}),$$

where $p_c(\cdot \mid \cdot)$ is the true distribution for class c, $\boldsymbol{x}_c$ are the data belonging to class c, $\boldsymbol{\phi}_c$ are the parameters of distribution $p_c(\cdot \mid \cdot)$, k_c are the number of components in class c (not always known), π_{ic} is the proportion of component i in class c, and $\boldsymbol{\theta}_{ic} = (\mu_{ic}, \Sigma_{ic})$ are the parameters for component i in class c. The use of multivariate normal mixtures allows us to select between two parameterisations for the priors, a joint conjugate prior for the means and precisions in a component (μ_{ic}, Λ_{ic}, where $\Lambda_{ic}^{-1} = \Sigma_{ic}$) or independent conjugate priors for these parameters. We discuss both parameterisations in section 3.2.

The use of these types of models in the multivariate setting has been discussed by Gelman *et al.* (1995) and Robert (1996) among others. In most cases the number of components k_c is treated as known (or at least fixed). In section 3.4.2 we discuss an algorithm created by Stephens (1997) which allows the use of this kind of methods while varying the number of components in a class (k_c) during our sampling. For the first 3 sections in this chapter we will assume that the number of components in a class is known.

3.2 Choice of priors for classes with common precision

3.2.1 Joint conjugate priors for μ and Λ

In this section and until section 3.6 we will assume that the precision matrices for all the components in class c are equal, this means $\Sigma_{ic} = \Lambda_c^{-1}$. It is common to use this restriction for two reasons. The first one is practical in terms of not requiring to estimate k_c different $p \times p$ matrices at each iteration. The second one comes from the fact that a model that assumes the subpopulations within a class to have the same covariance matrix is very reasonable. We relax this restriction in section 3.6 where we reparametrize the precision matrix into size, shape and orientation and allow the sizes to vary for each component in class c.

Multivariate normals from a Bayesian perspective have the advantage that they have conjugate priors. When both the means and the variances are unknown we can use a joint conjugate prior Normal-Wishart distribution for the means and precision matrix (Gelman *et al.* (1995)). This is easily extended to the case of mixtures of multivariate normals. For the case when we restrict the precision matrices for all components in a class to be equal the new priors would be

$$\begin{aligned} \mu_{ic} &\sim \mathcal{N}_p(\xi_0, \tau_0(\Lambda_c)^{-1}) \\ \Lambda_c &\sim \mathcal{W}_p(\nu_0, \Omega_0), \end{aligned}$$

where $\mathcal{W}_p(\nu, \Omega)$ denotes a Wishart distribution with ν degrees of freedom and scale matrix Ω. The hyperparameters for μ_{ic} are ξ_0, the prior mean, and τ_0 which is the inverse of the number of prior measurements on the Λ_c^{-1} scale. We will follow the parameterisation given by DeGroot (1970) (page 59) for a Wishart distribution which is given $W \sim$ Wishart with parameters ν and B,

$$p(W) = const \times \mid B \mid^{\nu/2} \mid W \mid^{(\nu-k-1)/2} \exp(-\frac{1}{2}\mathrm{tr}(BW)),$$

with $E(W) = \nu B^{-1}$. This parameterisation for the Wishart has the advantage that now the conditional posterior for Λ can be obtained without the need of inverting two matrices at each iteration. Proof of the conditional posterior for Λ_c (equal precision matrix for all components in class c) can be found in appendix B. When simulating from a Wishart it was necessary to modify the algorithm by Ripley (1987) to deal with this change in the parameterisation. From Ripley (1987): We consider only the case of a central Wishart distribution $W \sim \mathcal{W}(\nu, B)$ with a DeGroot parameterisation with $E(W) = \nu B^{-1}$. For this case $W = (U^{-1})^T V (U^{-1})$ where $UU^T = B$ and $V = \sum Z_p Z_p^T$ (Z_p a p-variate standard normal). We use Bartlett's decomposition and let $\zeta_i \sim \chi^2_{\nu+1-i}$, $i = 1, \ldots p$, and $\varepsilon_{ij} \sim \mathcal{N}(0,1)$ for $1 \leq i \leq j \leq p$.

Then

$$V_{ii} = \zeta_i + \sum_{k<i} \varepsilon_{ki}^2,$$

$$V_{ij} = V_{ji} = \varepsilon_{ij}\zeta_i^{-1} + \sum_{k<i} \varepsilon_{ki}\varepsilon_{kj},$$

for $i < j$.

Using this parameterisation gives the following full conditional posteriors for the means and precision

$$f(\boldsymbol{\mu}_{ic} \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\Lambda_c}^{(m)}) = \mathcal{N}(\xi_{ic}^*, \tau_{ic}^*(\Lambda_c)^{-1}),$$
$$f(\boldsymbol{\Lambda_c} \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\mu}_{1c}^{(m)}, \ldots, \boldsymbol{\mu}_{k_c c}^{(m)}) = \mathcal{W}(\nu_c^*, \Omega_c^*),$$

where $\nu_c^* = \nu_0 + n_c$, and

$$\xi_{ic}^* = \frac{n_{ic}\bar{x}_{ic} + \xi_0\tau_0^{-1}}{n_{ic} + \tau_0^{-1}},$$
$$\tau_{ic}^* = \frac{\tau_0}{\tau_0 n_{ic} + 1},$$
$$\Omega_c^* = \Omega_0 + S_c + \frac{n_{ic}}{\tau_0 n_{ic} + 1}(\bar{x}_{ic} - \xi_0)(\bar{x}_{ic} - \xi_0)^T)$$

where $\bar{x}_{ic}$ is equal to the vector of means of the observations allocated to component i for class c, and

$$S_c = \sum_{i=1}^{k_c} \sum_{j:z_j=i} (\boldsymbol{x}_j - \bar{\boldsymbol{x}}_i)(\boldsymbol{x}_j - \bar{\boldsymbol{x}}_i)^T, \ (i = 1, \ldots, k).$$

A detailed account of these conditionals can be seen in appendix B.

Step **c)** of algorithm 2.1 for this case is obtained by sampling the means μ_{ic} and then the common precision matrix Λ_c whose conditional is independent from the means. A directed acyclic graph of this model can be seen in figure 2.1 (with variable k_c).

3.2.2 Independent conjugate priors for μ and Λ

Most applications to date have used joint-conjugate priors for μ_{ic} and Λ_{ic}. This type of parameterisation can present problems as recorded by West *et al.* (1994). They use hierarchical priors for mixture models and report that using a conjugate joint-prior, the conditional posterior for ξ (in our model) could be heavily influenced by a small group far away from the bulk of the data. This would make this type of parameterisation very

sensitive to outliers. Recent applications (Stephens (1997), West *et al.* (1994), Richardson and Green (1997)) use independent conjugate priors for the means μ_i and dispersion matrices Σ_i for each component in a mixture. This parameterisation has been described as a non-conjugate base prior (West *et al.* (1994)) or as semi-conjugate (Gelman *et al.* (1995) in the univariate case). In any case the idea is the use of independent conjugate priors for μ and Σ.

For the case when we restrict the precision matrices for all components in a class to be equal the new priors would be

$$\begin{aligned} \mu_{ic} &\sim \mathcal{N}_p(\xi_0, \kappa^{-1}) \\ \Lambda &\sim \mathcal{W}_p(\nu_0, \Omega_0), \end{aligned}$$

where now κ is a precision matrix. Values for the hyperparameters κ, ξ_0, ν_0, and Ω_0 are discussed in section 3.3.1.

This gives the following full conditional posteriors for the means and precision

$$\begin{aligned} f(\boldsymbol{\mu}_{ic} \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\Lambda_c}^{(m)}) &= \mathcal{N}(\xi^*_{ic}, (\kappa^*_{ic})^{-1}), \\ f(\boldsymbol{\Lambda_c} \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\mu}^{(m)}_{1c}, \ldots, \boldsymbol{\mu}^{(m)}_{k_c c}) &= \mathcal{W}(\Omega^*_c, \nu^*_c), \end{aligned}$$

where $\nu^*_c = \nu_0 + n_c$, and

$$\begin{aligned} \xi^*_{ic} &= (\kappa + n_{ic}\Lambda_c)^{-1}(\kappa\xi_0 + n_{ic}\Lambda_c\bar{x}_{ic}), \\ \kappa^*_{ic} &= \kappa + n_{ic}\Lambda_c, \\ \Omega^*_c &= \Omega_0 + \sum_{i=1}^{k_c} M_{ic}. \end{aligned}$$

where $\bar{x}_{ic}$ is equal to the vector of means of the observations allocated to component i, and

$$M_{ic} = \sum_{j:z_j=i} (x_j - \mu_{ic})(x_j - \mu_{ic})^T.$$

So step **c)** of algorithm 2.1 is obtained by sampling the means μ_{ic} from the conditionals and then using these means to sample the common precision matrix Λ_c. Figure 3.1 shows this model using independent conjugate priors for the means and precision in class c (with variable k_c).

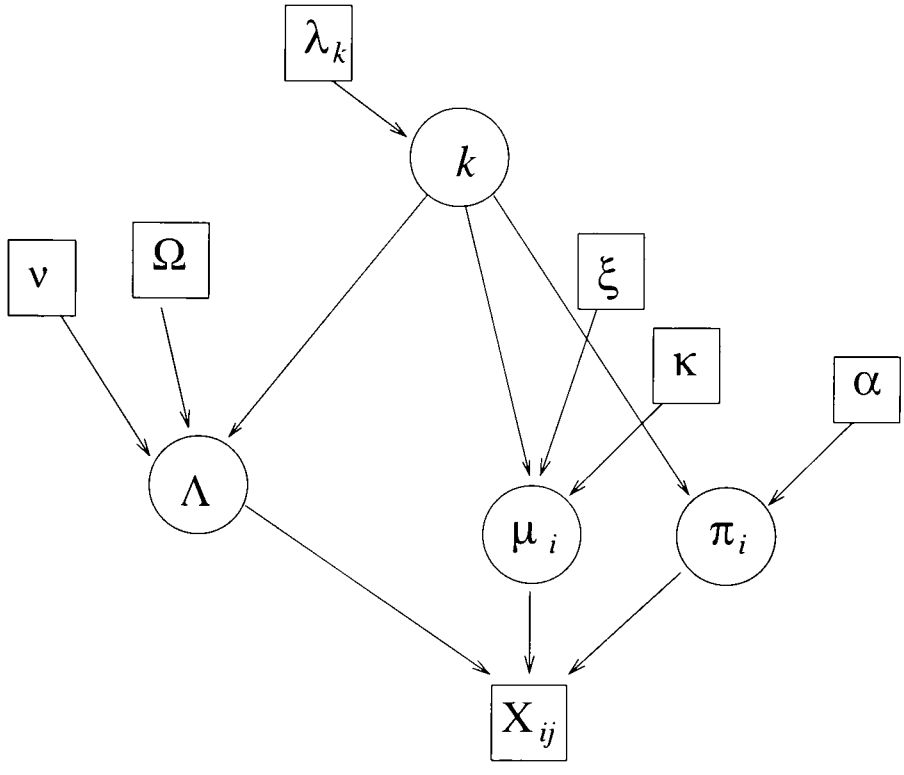


Figure 3.1: Directed acyclic graph of the parametric model used for mixtures of Normal distributions with independent conjugate priors for the means and precision. This model takes into consideration only one class. The data X_{ij} is modelled as a mixture of k normal distributions with means μ_i, common covariance matrix $= \Lambda^{-1}$ for all i, and proportions π_i. The hyperparameters for this specific model are $\nu, \Omega, \xi, \kappa, \alpha$ and λ_k. In this model the number of components is treated as variable with a Poisson prior with parameter λ_k.

3.3 Initial values and hyperparameters for classes with common precisions

3.3.1 Using joint conjugate priors

A program was developed using C (and later C++) to use the Gibbs sampler algorithm to create a sampling chain for the parameters. In parallel another three programs were created to obtain the marginal predictive density given the sampling chains of the parameters, the joint predictive density (only for two variables) given the sampling chain, and a classification obtained by integrating out the parameters. This method of classification is discussed and used in chapter 4. To test the performance of our program we compared the results obtained with those obtained using BUGS version 0.501 for SPARC (Spiegelhalter *et al.* (1994)).

When we use joint conjugate priors, the priors for the model consist of a Dirichlet prior for components proportions (π_i), and a Normal–Wishart joint prior for the means and precision matrix (μ_i, Λ). We assume the precision matrix to be equal for all the components in a specified class with a common global prior for all classes. This restriction is relaxed later in section 3.6.

The initial values used are $\pi_{ic} = \frac{1}{k_c}$, $\mu_{ic} \sim \mathcal{N}(\xi, \Sigma_{W^\cdot})$ where $\Sigma_{W^\cdot}$ =dispersion for the whole dataset, and $\Lambda = b\times$precision of the whole dataset, where $b \sim \Gamma(1,1)$. These initial values are obtained at random. As we mentioned in section 2.2.3 we run 4 chains with different initial values to determine when these values are no longer affecting the output of the chain.

The hyperparameters $(\alpha_i's)$ for the Dirichlet are chosen as a vector of $1's$, as this is a "non–informative", yet proper prior for a Dirichlet distribution. For the case of the Normal–Wishart we use a modified version of the Gelman *et al.* (1995) (page 80) with $(\xi_0, \tau_0(\Lambda)^{-1}; \nu_0, \Omega_0)$ where the first two are the parameters for the multivariate normal and the second pair are the parameters for the Wishart distribution (using DeGroot's parameterisation).

The values chosen for the hyperparameters ξ_0 and Ω_0 were the vector of means of the whole training data and the covariance matrix of the whole dataset, that is taking into consideration all the classes. We use these hyperparameters for all the classes. The use of the vector of means as ξ_0 was made with the intention of keeping ξ_0 close to the 'centre' of the data. The choice of Ω_0 as the dispersion matrix of the dataset gives a data dependent choice which should not be too informative as the dispersion matrix for a specific class should be less than Ω_0, and the dispersion matrix for component i in class c even smaller.

The hyperparameter τ_0 can be interpreted as a weight for the prior measurements on the

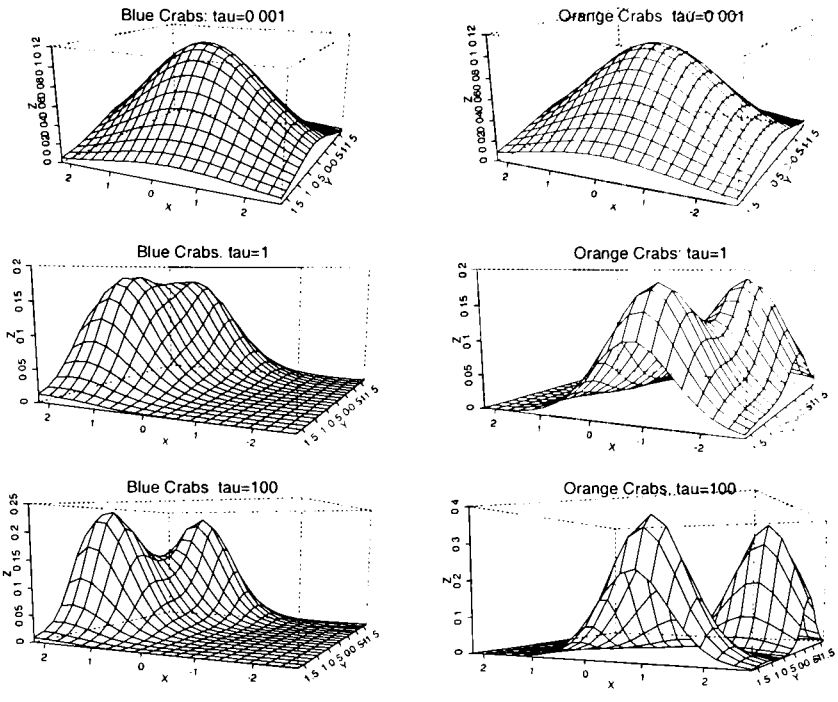


Figure 3.2: Joint predictive densities for the first two principal components of the Crabs data for different values of τ_0 and fixed number of components $k = 2$. The variables used are first two principal components. As τ_0 decreases the priors gain more influence in the predictive density. As we are using the same priors for both groups if the predictive density is heavily influenced by the priors the model is of no use for discrimination between the groups. As τ_0 increases we see a clearer separation between components in a class.

precision matrix scale. In practice our choice of τ_0 will reflect the importance attributed to the priors. As we can see in figure 3.2 for the Crab data (in appendix A), our choice of τ_0 will affect our model and the classification as well. For this section we will take $\tau_0 = 1$ and we defer more on this hyperparameter until section 3.5.1.

The choice of a value for the degrees of freedom (ν_0) of a Wishart distribution could present equal problems to our model. For a Wishart distribution to be proper it is required that $\nu_0 \geq p$ where p =number of variables. We study how different values for ν_0 affect our predictive densities and our model. The use of data dependent priors is important as the priors selected are relatively structured due to the nature of the problem (problems with mixtures). For this reason it is natural to use common priors for all the classes and take the initial parameters for these priors from the whole training dataset. We used the

Pima Indians data that consists of 68 diabetics and 132 non–diabetics as our training set, and 332 non-classified datapoints (appendix A). We simplified the problem to the use of the first linear discriminant (LDF) and a standardised plasma glucose concentration as variables to take into consideration for classification (to make the problem artificially bivariate).

The density for the LDF can be seen in appendix A figure A.6. The density of the LDF for the classified group of diabetics appears to be multi-modal (Ripley (1996), page 99). This multi-modality could indicate a mixture of (at least) two normal distributions with almost equal proportions. We decided to use the Gibbs sampler with $k = 2$ hoping to find a sampling chain that would reflect this separation of two components with equal proportions. Our choice of $\nu = 2$ will give a vague but proper prior to the Wishart distribution.

Nevertheless after running the process in the created code the two components were collapsing into one for most of the duration of the process. This made us doubt the performance of the code and because of that we used BUGS to obtain a sampling chain to compare with that obtained using the created program. We simplified the problem even more by using the univariate case with the LDF as the only variable. We used the equivalent priors of the multivariate case Dirichlet, Normal–Gamma with hyperparameters 1 for the Dirichlet, the mean (of the LDF) of the classified data, the variance (of the LDF) of the classified data (scale parameter), and with shape parameter equal to 1 (2 degrees of freedom). Even in this case the posterior distributions obtained for the parameters did not reflect the separation between components (again for two components) figure 3.3.

The precision in the sampling chain (for most of the time) is similar to the one obtained in the case of modelling with only one component. Our choice of prior for ν_0 and Ω_0 is fairly dispersed so we decided to make our priors more informative to create a higher precision artificially in order to obtain the separation between the two components we were looking for. We decided to increase the degrees of freedom. The effect of increasing ν is to give more weight to the prior as well as increasing the precision for the prior. This reflects in the univariate case as an increase in the shape parameter of the gamma distribution (prior for the precision) to 4. Again we used the package BUGS for the univariate case (LDF) and obtained a clear bimodal distribution for the posterior of the means of the components (this bimodal distribution is caused by the switching of the parameter during the sampling). This can be seen in figure 3.4. We can also appreciate that the kernel density for the precision has a peak around 1.2 and decreases very slowly (very heavy tail on the right hand side), this being caused by the strong prior used. We tried to modify the scale parameter instead of the shape parameter but the results did not give the clear cut separation of two components for this dataset. The same method was used for the bivariate case using both BUGS and the program created with similar results to the ones obtained for the univariate case. The predictive densities obtained using $\nu = p = 2$ and $\nu = 8$ can

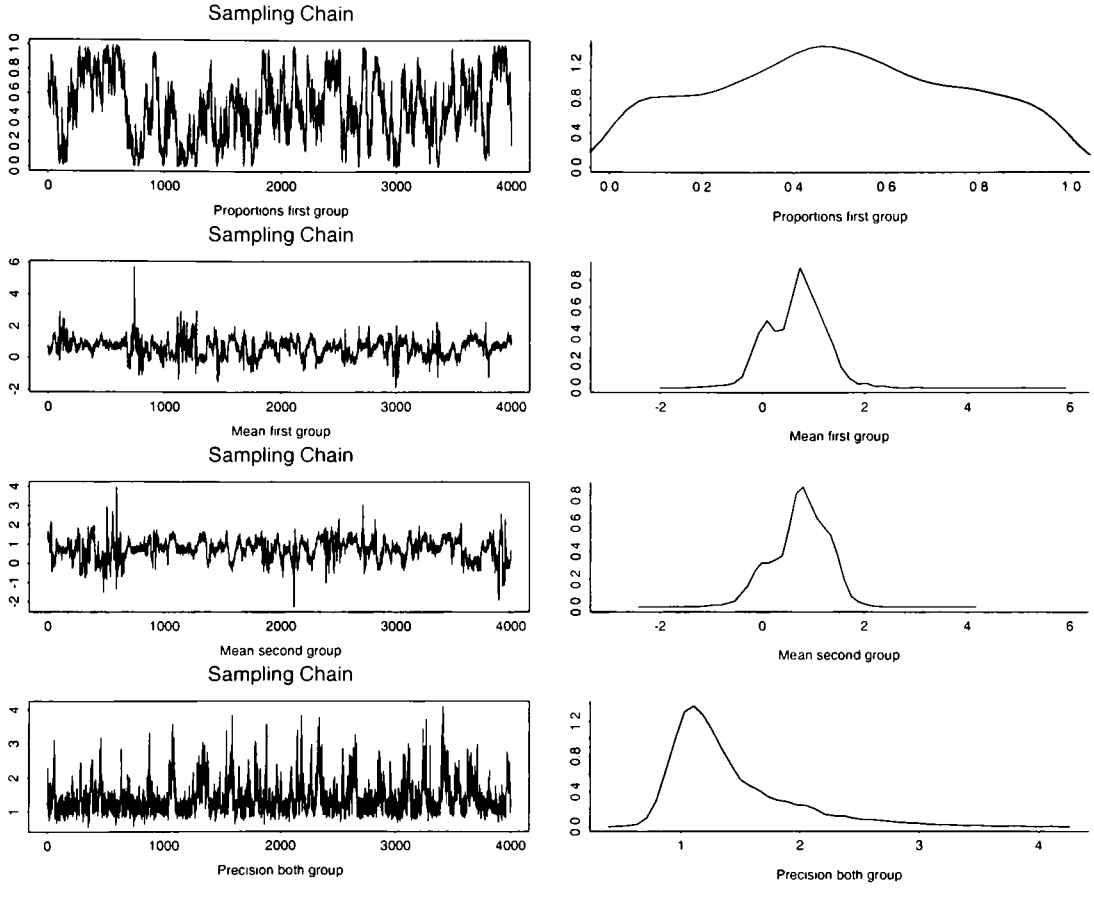


Figure 3.3: Sampling Chains and Densities for the posterior distribution of the parameters (2 components equal variance) obtained using BUGS for the LDF of the diabetics (classified) in the Pima Indians data (univariate). The parameters plotted are the proportions of the first component, the mean for the first and second component and the precision. The priors used were $\alpha = 1$ for the Dirichlet, mean of the whole dataset, precision of the whole dataset and 1 as the shape parameter for the Normal–Gamma prior. The burn-in period was 1000 iterations.

be seen in figure 3.5. These show that the predictive density for the diabetics could be a mixture of two components with roughly the same proportions while the non-diabetics do not show any evidence of multimodality.

3.3.2 Using independent conjugate priors

When we use independent conjugate priors discussed in section 3.2.2, we will use the same initial values for μ, π, and Λ as in section 3.3.1 and apart from the new hyperparameter κ the rest of the hyperparameters ξ, ν_0, Ω_0 have also been defined in section 3.3.1. For our choice of κ we will take the precision matrix of the whole dataset. The idea behind using this matrix is that we want to have a fairly uninformative prior for the means but at the same time be able to sample from the data.

3.4 Modelling for classes with unknown number of components

In chapter 1 we mentioned some of the problems arising in the context of Normal mixtures. The main problems discussed were that the parameters are unknown, that a unique test for the number of components in a mixture has not yet been obtained, and the lack of identifiability of the mixture components (or interchangeability of the parameters for different components). This last problem of identifiability is of no concern to us as in the classification algorithm the parameters will be integrated out of the model before the observations are classified and for that reason we are not interested if during the sampling algorithm any permutations occur between the components of a class.

The selection of a number of components in a class is a more difficult problem because if the usual MCMC methods are used and the number of classes used is wrong, the sampling chains for the parameters could be useless for the classification for example if the number of components selected for a class were less than the actual number. On the other hand it could be extremely inefficient if the number of components were extremely high in comparison with the actual number.

3.4.1 Using different fixed values for k_c

The selection of the number of classes has not been solved yet, nevertheless, recent approaches that use MCMC methods that allow the number of classes to vary (as another variable parameter) have been developed by Escobar and West (1995), Richardson and

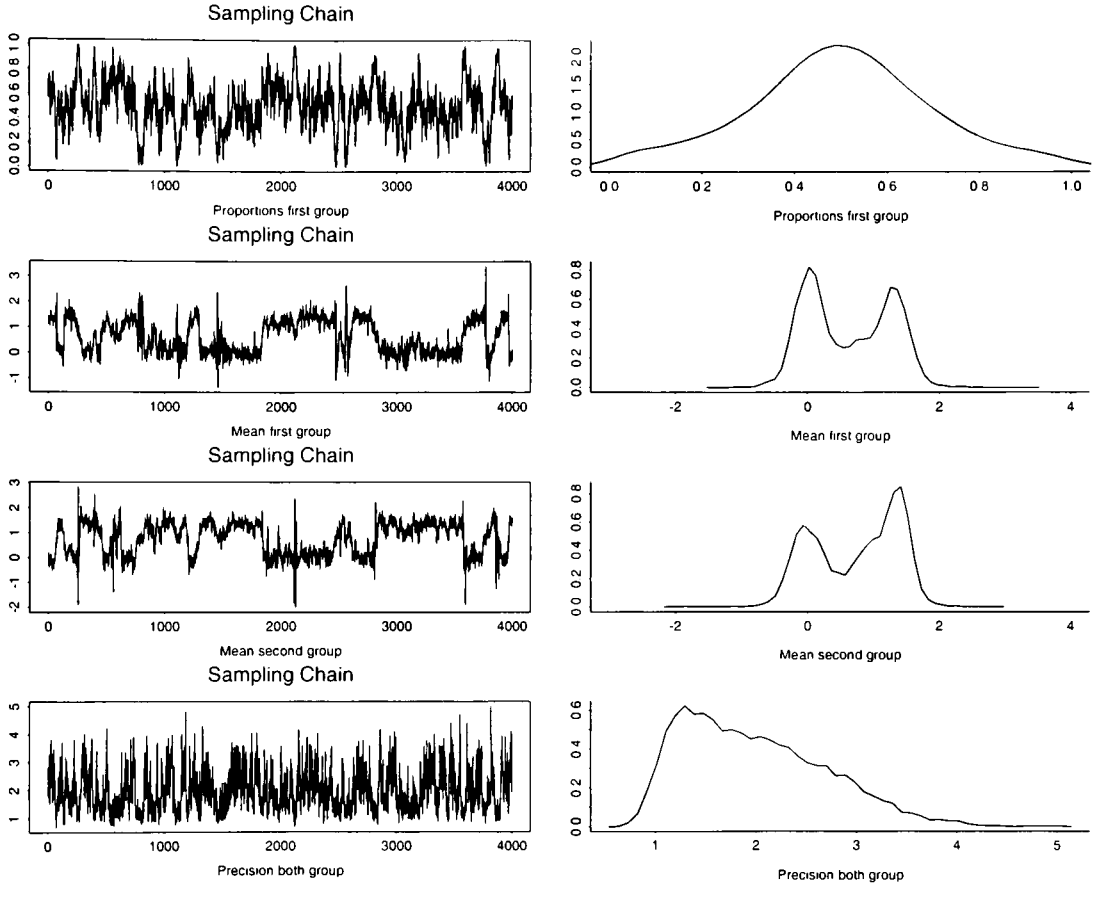


Figure 3.4: Sampling Chains and Densities for the posterior distribution of the parameters (2 components equal variance) obtained using BUGS for the ldf of the diabetics (classified) in the Pima Indians data (univariate). The parameters plotted are the proportions of the first component, the mean for the first and second component and the precision. The priors used were $\alpha = 1$ for the Dirichlet, mean of the whole dataset, precision of the whole dataset and 4 as the shape parameter for the Normal–Gamma prior. The burn-in period was 1000 iterations.

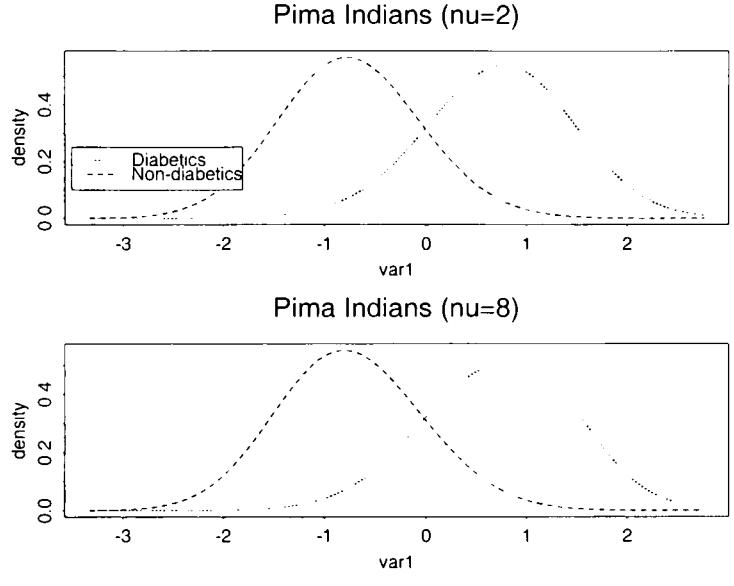


Figure 3.5: Predictive densities for the diabetics and non–diabetics in the Pima Indians data for different values of ν_0 and fixed number of components $k = 2$. The variable used was the LDF. We are using the same priors for both groups and we expected the predictive density to be heavily influenced by our choice of ν_0, nevertheless although the predictive densities were affected, the separation between both groups is still kept.

Green (1997), and Stephens (1997). These approaches take into consideration the option of modelling the number of components k as an unknown parameter. We studied first of all the use of Bayes Factors to determine the number of components in a class. This could be seen as the Bayesian alternative to a likelihood ratio test. Bayes factors are the ratio of the marginal likelihood under one model (M_1) to the marginal likelihood under a second model (M_0). The ratio of their posterior probabilities given data D would be

$$\frac{P(M_1 \mid D)}{P(M_0 \mid D)} = \frac{P(M_1)}{P(M_0)} \times B_{10},$$

where

$$B_{10} = \frac{P(D \mid M_1)}{P(D \mid M_0)} = \frac{\int p(\theta_{(1)} \mid M_1) p(D \mid \theta_{(1)}, M_1) d\theta_{(1)}}{\int p(\theta_{(0)} \mid M_0) p(D \mid \theta_{(0)}, M_0) d\theta_{(0)}},$$

and as such the key quantity necessary to obtain a Bayes factor (B_{10}) is the marginal likelihood $P(D \mid M_l)$ after integrating out the parameters $\theta_{(l)}$. We have a sampling chain for the parameters (we are sampling from the posterior) and as such we can obtain a chain for the likelihood of these parameters, and try to estimate this marginal likelihood with it.

To test the viability of the results of the program created to carry out the Gibbs sampler, a simulated dataset was used. The data are bivariate with the first variate being a mixture of 3 normals with means at $-4.0, 4$ and unit variance, and the second variate is a univariate normal with mean 0 and unit variance (the variables are independent). The size of the dataset is 300 (approximately 100 for each component in the first variable) and the data can be seen in figure A.1. We can see in figure 3.6 a histogram of the data for variable 1 and how the model for different k describe it in comparison with its empirical density. Figure 3.7 shows the log–likelihood chains obtained for the different number of components. Most of the time the log–likelihood chain remains close to the maximum likelihood. This maximum likelihood was obtained using S-plus options deriv and deriv3 to minimise $-L$ for a mixture of k univariate normal distributions as described in Venables and Ripley (1997). Once the parameters for the maximum likelihood were obtained for the first variate then these were used as initial parameters for the EM algorithm that obtained the maximum for the bivariate case. We can see in Table 3.1 that the maximum value obtained from the sampling chain is very near to the actual maximum log–likelihood.

To obtain an estimate of the marginal likelihood $P(D \mid M)$ we use importance sampling. Define

$$\| X \|_h = \frac{1}{T} \sum_{t=1}^{T} X(\theta^{(t)}),$$

where $X(\cdot)$ is a function and $\{\theta^{(t)}\}$ is a sample of size T. Assuming that we are dealing

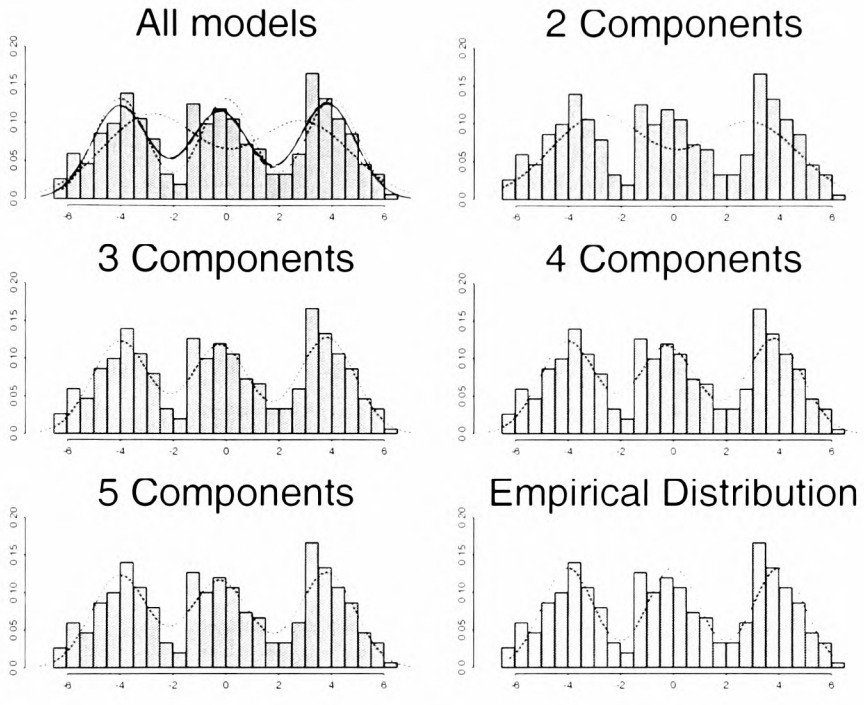


Figure 3.6: Histogram of a mixture of 3 normals with means at $-4, 0, 4$ and variance 1 the size of the dataset is 300. The top figure shows all the models with different number of components and the empirical distribution. On the right hand side we have the predictive densities for models with number of components $k = 2, 4$ and the empirical distribution. On the left hand side underneath the figure with all the components we have the predictive densities for $k = 3$ and 5. The Gibbs sampler was run for 20000 times with a burn-in period of 10000.

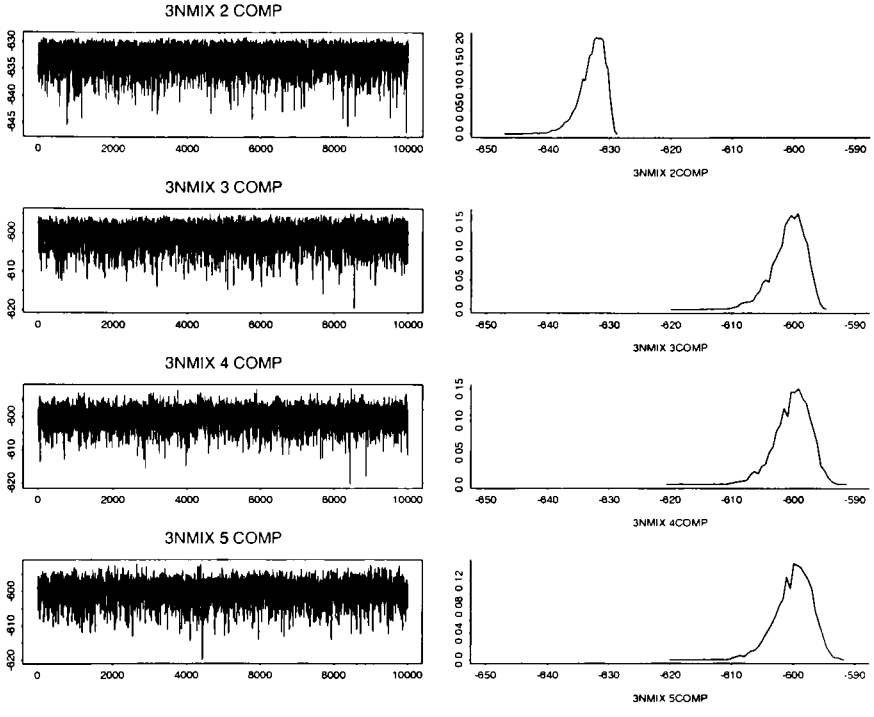


Figure 3.7: Log-likelihoods obtained from the sampling chains of parameters obtained using Gibbs sampler. The data used are a mixture of 3 normals with means at $-4, 0, 4$ and variance 1, and as second variate a univariate normal with mean 0 and unit variance. The size of the dataset was equal to 300 (100 for each component), the Gibbs sampler was run for 20000 times with a burn-in period of 10000.

Table 3.1: Maximum log-likelihood and log(maximum), log(harmonic estimate) and $\log(\widehat{p}_4)$ estimate obtained from the likelihood of the sampling chain. The data used are a mixture of 3 normals with means at $-4, 0, 4$ and variance 1, and as second variate a univariate normal with mean 0 and unit variance. The size of the dataset was equal to 300 (approximately 100 for each component), the Gibbs sampler was run for 20000 times with a burn-in period of 10000

$\nu = 8, \Sigma_0 = \Sigma$

# Components	Max(log-lik)	Max(s. chain)	log(Harm.Est.)	$\log(\widehat{p}_4)$
2	−628.8	−628.9	−638.6	−635.9
3	−594.2	−594.8	−610.6	−605.1
4	−589.8	−591.4	−611.3	−604.7
5	−589.2	−591.9	−610.8	−604.8

with estimating the marginal likelihood for a single model we can express this as

$$P(D) = \int L(\theta)\pi(\theta)d\theta,$$

where $L(\theta) = P(D \mid \theta)$ is the likelihood and $\pi(\theta) = P(\theta)$ is the prior. Using importance sampling to evaluate $P(D)$ we suppose we can sample from a density proportional to the positive function $g(\theta)$ so that

$$P(D) = \int L(\theta) \left[\frac{\pi(\theta)}{bg(\theta)}\right] bg(\theta)d\theta,$$

where $b^{-1} = \int g(\theta)d\theta$. Given a sample $\{\theta^{(t)} : t = 1, ..., T\}$ from the density $bg(\theta)$, then simulation-consistent estimators for $P(D) = \widehat{P}(D)$ and for $b = \widehat{b}$ where $\widehat{P}(D) = \| \frac{L\pi}{bg} \|_g$, and $\widehat{b} = \| \frac{\pi}{g} \|_g$. This yields the importance sampling estimator of $P(D)$

$$\widehat{P}_{IS} = \frac{\| \frac{L\pi}{g} \|_g}{\| \frac{\pi}{g} \|_g}$$

In our process (MCMC methods) we are obtaining samples from the posterior distribution $L\pi$ so it makes sense to use these samples and this distribution as our importance sampling distribution $g = L\pi$. This leads naturally to the use of the harmonic estimator in Equation 3.1 as $\widehat{P}(D) = 1/ \| \frac{1}{L} \|_{L\pi}$. A complete description of this can be found in Raftery (1996).

$$\widehat{p}_{harm}(x) = \frac{1}{\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\widehat{P}(x|\theta^{(i)})}} \tag{3.1}$$

As we can see in figure 3.7 the frequency of extreme values (low likelihoods) is higher than what would be expected for the usual asympthotic theory to work. Raftery (1996) describes these extreme low values and that the harmonic mean "converges almost surely" to the correct value but does not in general satisfy a Gaussian central limit theorem as the variance of the inverse of the harmonic estimate is normally infinite.

While dealing with the harmonic estimate for the likelihood obtained from modelling with mixtures of normal distributions we intend to calculate a standard error or variance for the estimate. The use of MCMC methods allows us to obtain a sampling chain from the parameters which enables the calculation of a harmonic estimate for the likelihood. Different priors were used to determine the sensitivity of the model and although the sampling chains seemed not to vary too much between each other (even in the cases when very structured priors are used) the harmonic estimates seemed to vary a large amount from one sampling chain to the other as we can see in Tables 3.1.

To determine the variance of this estimate we faced several problems. The delta method (Crowder *et al.* (1991)) was employed to obtain an estimate for this variance. Given the function for the harmonic estimate

$$h = \frac{n}{\sum^{n}\frac{1}{x_i}},$$

where in our case x_i = the likelihood of the parameters at iteration i. Using the delta method the variance of this estimate can be approximated by

$$Var(h) = \frac{n^2}{\left(\sum^{n}\frac{1}{x_i}\right)^4}\sum^{n}\frac{1}{x_i^4}Var(x).$$

From this equation we can see that as n increases the variance of the estimate should decrease by n. In a random sample from a uniform distribution between 0 and 1 this seemed to work well, nevertheless as the variance was calculated for the likelihood sample this proved wrong. As n increased the variance of the estimate was increasing as well and as the variance of the likelihood ($Var(x)$) remained roughly the same something was going wrong with the coefficient. The coefficient defined as the part of the variance of the

estimate which does not correspond to the variance of r

$$coef_n = \frac{n^2}{\left(\sum^n \frac{1}{x_i}\right)^1} \sum^n \frac{1}{x_i^1}$$

A closer look at the coefficients showed that these were indeed increasing with n and if compared with n^2 they were very similar. Figure 3.8 shows the inverse of the likelihoods obtained from the sampling chain at different lengths. We can see that in general one extreme (very low) value of the likelihood dominates the whole sampling chain making the harmonic estimate completely unstable (not consistent?) and variance impossible to calculate.

Newton and Raftery (1994) recommend the use of other estimates for the likelihood in order to avoid the difficulty of these extreme values. The main idea is to do importance sampling from a mixture of the prior and the posterior $g(\theta) = \delta(prior) + (1-\delta)(posterior)$ with $0 < \delta < 1$. This means sampling from the prior and posterior distribution respectively. Nevertheless this would involve sampling from both prior and posterior. A more appealing solution is sampling from the posterior and then using an iterative process to obtain an approximation to the estimate that uses both prior and posterior. The proposed iterative function is

$$\widehat{p}_4(x) = \frac{\frac{m\delta}{(1-\delta)} + \sum_{i=1}^m p(x \mid \theta^{(i)}) / \{\delta \widehat{p}_4(x) + (1-\delta) p(x \mid \theta^{(i)})\}}{\frac{m\delta}{(1-\delta)\widehat{p}_4(x)} + \sum_{i=1}^m \{\delta \widehat{p}_4(x) + (1-\delta) p(x \mid \theta^{(i)})\}^{-1}}, \tag{3.2}$$

Values of this estimate can be seen in table 3.1 and are a lot more stable than those obtained using the harmonic estimate. The value used for δ was .01 and the initial value for $\widehat{p}_4(x)$ was the harmonic estimate $\widehat{p}_{harm}(x)$.

3.4.2 Varying the number of components k_c while sampling

Up to this section in this chapter we have assumed that the number of components for each class k_c is known. This in general is not the case especially as sometimes a mixture model will be used as a mathematical tool to obtain an approximation to the class density. In this last case (*indirect approach*) the number of components in a class would not have a direct interpretation as a subclass in the class, and for that reason it would be impossible to know how many components exist in one class. In section 3.4.1 the harmonic estimate was shown to be of little value for calculating the Bayes Factors as a test for the number of components and the use of likelihood ratio tests (used by Wolfe (1971) and Hernandez-Avila (1979)) is not reliable either. Developments in the field in the last five years have incorporated the number of components as another variable parameter with encouraging

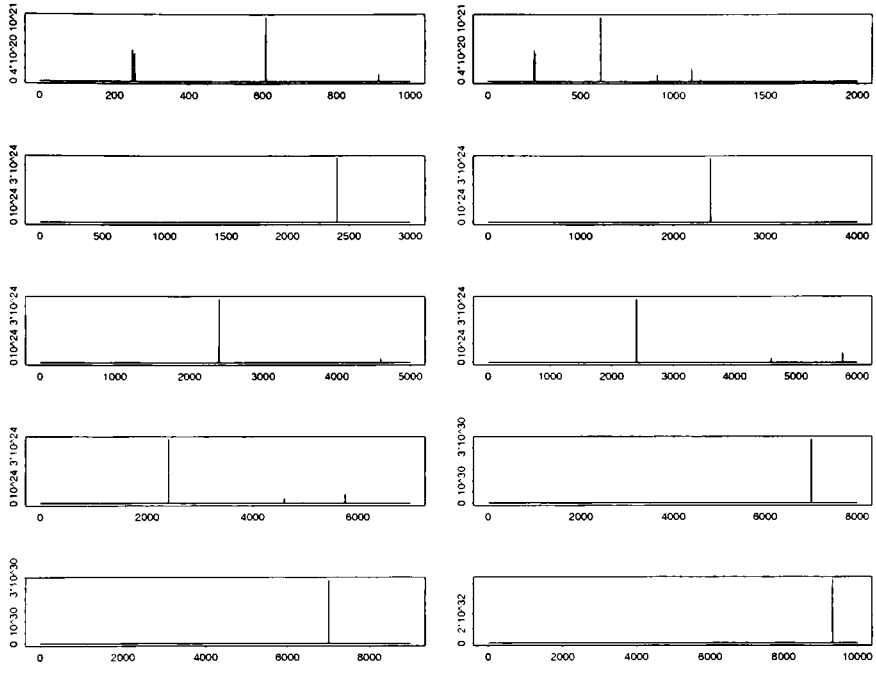


Figure 3.8: Inverse of likelihoods obtained from the sampling chains of parameters obtained using Gibbs sampler. The data used are a mixture of 3 normals with means at $-4.0, 4$ and variance 1, and as second variate a univariate normal with mean 0 and unit variance. The size of the dataset was equal to 300 (approximately 100 for each component). The Gibbs sampler was run for 20000 times with a burn-in period of 10000 and so the length of the sampling chain used is 10000. The plots show the inverse of the likelihood at different lengths $(1000, 2000, ..., 10000)$

results. Richardson and Green (1997), West *et al.* (1994), and Stephens (1997) model k_c as a parameter with prior and posterior distribution using algorithms that are variations from the MCMC methods. All these methods combine Gibbs sampling with some other algorithm that allows the number of components in a class k_c to vary at each iteration.

Richardson and Green (1997) use mixtures of univariate normal distributions with different means and variances for the components, with prior distributions $\mathcal{N}(\xi, \kappa)$ and $\Gamma(\alpha, \beta)$ for the mean and inverse variance respectively, with β distributed as $\Gamma(g, h)$. The algorithm has the following steps:

a) Update the mixing proportions (π_{ic}).
b) Update the parameters of the component distributions (μ, σ).
c) Update the allocation of the data.
d) Update the hyperparameter for the variance (β).
e) Splitting one mixture component into two, or combining into one.
f) The birth or death of an empty component.

The probability of a split or a combination depends on π_{ic}, μ, σ, β, ξ, and κ. This uses the theory developed by Green (1995) and allows the chain to change its component dimension and come back to it with a certain probability. Richardson and Green use a uniform prior for the number of components but accept that different priors such as a Poisson could be used instead. Nevertheless there seem to be definite problems in the generalisation to the multivariate case and the "reversible–jump" chains seem to increase in difficulty fast as the dimensionality increases.

On the other hand Escobar and West (1995) use mixtures of Dirichlet processes. Each datapoint x_i is a realization of a normal distribution with mean μ_i and variance Σ_i. The idea is that the parameters $\theta_i = (\mu_i, \Sigma_i)$ are realisations of a Dirichlet Process G with parameters α and G_0, where $E[G] = G_0$ is the base prior and $\alpha > 0$ the scalar precision parameter (see Antoniak (1974) for a description of Dirichlet Process). These processes are discrete and because of this, the probability of datapoints sharing θ is positive effectively reducing the number of components to something much smaller than the total size of the dataset.

The birth–and–death Algorithm

This algorithm was developed by Stephens (1997) for this kind of model, who also created the code to use it. We created our own code but his code (in C++) was of great help in the creation of ours. It is used to allow the number of components k_c in a mixture of normal distributions to vary and obtain sampling chains from its parameters and for the number of components as well (for this section we will refer to k_c as k). As we mention

in section 2.2 these sampling chains will then be used to integrate out the parameters to obtain a classification of a new datapoint. This approach of the use of a birth and death process as implemented by Stephens (1997) seems to be the most flexible of the three. The main idea is to model k as a point process with each point defined by θ (parameters for a component) and π on a bounded set A. This is possible because the prior, likelihood and posterior distribution of the parameters (π, θ) do not depend on the labelling of the components which mean that they can be permuted. So the set of parameters in the mixture $\{((\pi_1, \theta_1), \ldots, (\pi_k, \theta_k)\}$ can be seen as a set of k points in $[0, 1] \times \Theta$, where Θ is the state space for the parameter θ.

We define this birth and death process in the state space $\Omega = \bigcup_k \Omega_k$ (Preston (1976)) where $\Omega_1 = [0, 1] \times \Theta$, so $\phi_1 \in \Omega_1$ where $\phi_i = (\pi_i, \theta_i)$ and $\{\phi_1, \phi_2, \ldots, \phi_k\} \in \Omega_k$. The process is defined in continuous time and evolves by jumps which are of two types, births or deaths. A birth signifies an addition of another point (θ, π) to the ones already in the process. A death means that one of the points in the process is eliminated. When the process is at $y \in \Omega_k$ $(y = \{(\pi_1, \theta_1), \ldots, (\pi_k, \theta_k)\})$ the behaviour of the process is defined by the birth rate $\beta(y)$, the death rate $\delta(y)$, and the birth and death transition kernels $K_\beta^{(k)}(y; \cdot)$ and $K_\delta^{(k)}(y; \cdot)$ which are probability measures on Ω_{k+1} and Ω_{k-1} respectively.

Births and deaths occur as independent Poisson processes, with rates $\beta(y)$ and $\delta(y)$ respectively. The time between jumps is therefore exponentially distributed with mean $\frac{1}{\beta(y)+\delta(y)}$, and with probability of it being a birth of $\frac{\beta(y)}{\beta(y)+\delta(y)}$, and of being a death of $\frac{\delta(y)}{\beta(y)+\delta(y)}$. This method was developed by Stephens (1997) who extended the work by Ripley (1977) and Preston (1976) in simulation of point processes.

For the mixture model this point process is restricted to the fact that the sum of the proportions $\sum \pi_i = 1$. For this reason we specify that if at time t the process is at $y = \{(\pi_1, \theta_1), \ldots, (\pi_k, \theta_k)\} \in \Omega_k$ if the next jump is a:

Birth of component at $(\pi, \theta) \in \Omega_1$ the process jumps to

$$y \cup (\pi, \theta) = \{(\pi_1(1 - \pi), \theta_1), \ldots, (\pi_k(1 - \pi), \theta_k), (\pi, \theta)\} \in \Omega_{k+1}.$$

Death of component at $(\pi_i, \theta_i) \in y$ the process jumps to

$$y \setminus (\pi_i, \theta_i) = \{(\frac{\pi_1}{(1 - \pi_i)}, \theta_1), \ldots, (\frac{\pi_{i-1}}{(1 - \pi_i)}, \theta_{i-1}), (\frac{\pi_{i+1}}{(1 - \pi_i)}, \theta_{i+1}), \ldots, (\frac{\pi_k}{(1 - \pi_i)}, \theta_k)\}$$
$$\in \Omega_{k-1}.$$

So when we use $y \setminus (\pi_i, \theta_i)$ we refer to the set of points in y minus point (π_i, θ_i) which lies in y.

Assume that the prior distribution of (k, π, θ) is of the form $p(k, \pi, \theta) = p(k)p(\pi, \theta \mid k)$, and $p(\pi, \theta \mid k)$ is a distribution in which π and $\theta_1, ..., \theta_k$ are independent with prior for π

$$p(\pi \mid k) \sim D(1, ..., 1),$$

and $\theta_1, ...\theta_k$ independently distributed on a space Θ according to density $\tilde{p}(\cdot)$, which we can simulate from.

Then the general Markov birth and death process with birth rate $\beta(y)$, death rate $\delta(y)$, and transition kernels $K_\beta^{(k)}(y; \cdot)$ and $K_\delta^{(k)}(y; \cdot)$ which satisfy

$$\beta(y)K_\beta^{(k)}(y; F) = \int_{(\pi,\theta): y \cup (\pi,\theta) \in F} b(y; (\theta, \pi))k(1-\pi)^{k-1}\tilde{p}(\theta)d\theta d\pi,$$

and

$$\delta(y)K_\delta^{(k)}(y; F) = \sum_{(\pi,\theta) \in y: y \setminus (\pi,\theta) \in F} d(y \setminus (\pi, \theta); (\pi, \theta)),$$

for some $b : \Omega \times ([0.1] \times \Theta) \to R^+$ and $d : \Omega \times ([0, 1] \times \Theta) \to R^+$, has stationary distribution $p(k, \pi, \theta \mid x^n)$, where x^n represent the data, if

$$(k+1)d(y; (\pi, \theta))L(y \cup (\pi, \theta))\frac{p(k+1)}{p(k)} = b(y; (\pi, \theta))L(y),$$

where $L(\cdot)$ represents the likelihood and $p(k)$ represents the prior distribution for k. This is obtained by following the detailed balance conditions given by Preston (1976).

If we set the birth rate $\beta(y) = \lambda_b$ a constant, the process can be easily sampled with a birth occurring at constant rate λ_b and the birth occurring at point (π, θ) chosen according to density

$$q(\pi, \theta) = k(1-\pi)^{k-1}\tilde{p}(\theta).$$

A death of point (π_j, θ_j) occurs as an independent Poisson process with rate

$$d(y \setminus (\pi_j, \theta_j); (\pi_j, \theta_j)),$$

and so we have the death rate given by

$$\delta(y) = \sum_{(\pi,\theta) \in y} d(y \setminus (\pi_j, \theta_j); (\pi_j, \theta_j)),$$

and when a death occurs it occurs at (π_j, θ_j) with probability

$$\frac{d(y \setminus (\pi_j, \theta_j); (\pi_j, \theta_j))}{\delta(y)}$$

Algorithm 3.1. The Algorithm for the birth–and–death process would be as follows.

1) Set $\phi = (\pi, \theta)$ (initial values for parameters).

2) Define death $\delta(y)$ and birth rate $\beta(y)$ (constant).

3) Set time for next jump (if total time of jumps greater than set time, END).

4) Define characteristic for next jump.

5) Birth then obtain 'new' point (π, θ) from $\pi \sim Be(1, k)$ and $\theta \sim \tilde{p}(\cdot)$.
Death then kill parameter set (π_j, θ_j) with probability $\frac{d(y \setminus (\pi_j, \theta_j); (\pi_j, \theta_j))}{\delta(y)}$.

6) Go to 2

This algorithm is used in conjunction with the Gibbs sampler to obtain a sample from the posterior of the parameters in the model (k, π, θ). The way we will mix both of them is by observing the state of the birth and death algorithm at constant intervals of time and taking the values for the parameters (k, π, θ) and performing one iteration from algorithm 2.1 (with its appropriate full conditionals) and then going back to step **2)** in the birth and death algorithm. This algorithm requires the use of a prior distribution for the number of components. One option is the use of a discrete uniform on the integers $1, \ldots, kmax$. This $kmax$ is sometimes given by a proportion of the total number of observations in a class, the maximum being the number of observations themselves. Stephens (1997) uses a doubly truncated Poisson with 0 class and $kmax + 1$ removed and parameter λ_k. We have decided to use the same prior with $kmax = 20$ as this number seems to be sufficient for the priors tested so far. Stephens developed a computer program that uses mixtures of normal distributions. He studies only the univariate and bivariate case with different precision for each component. We extends this application to the general multivariate case. Another difference is the restriction we impose on the precision matrix for all the groups in a class to be the same ($\Sigma_i^{-1} = \Lambda$). Because of this restriction some modification to the original algorithm is required. In the original algorithm each time a birth occurs a new 'point' is created in the space (π, θ) where π represents the proportion of the component in the mixture and $\theta = (\mu, \Lambda)$, the mean and precision matrix. Nevertheless as the precision is the same for all the components in a class, the new point is created in the space (π, μ) and Λ is left to be sampled only from the Gibbs sampler part of the algorithm (after the birth–death process has taken place). This could change the behaviour of the birth–death process because the birth or death of a component at iteration j depends on the likelihood of the 'points' at that iteration and this likelihood is highly dependent on the covariance matrix used.

A code has been developed that obtains a sampling chain for all the classes that are represented in the test set. We also created a function in S-plus to obtain a file that serves as input for the code. This file contains the number of classes, the number observations

per class, the number of variables and the data for each class. The idea of sampling all classes at each iteration comes from the possible use of non-fixed hyperparameters in the future. At the moment the same function that creates the input file (file.in) creates the file with the initial parameters required to run the program (file.ini) dependent on our choice of priors (joint-conjugate or independent-conjugate) and the file with the values required to obtain a predictive density using the output (file.pdens.ini). The priors used are the conjugate priors Dirichlet, Normal–Wishart (joint conjugate) or Normal, Wishart (independent conjugate) and for the number of components a truncated Poisson with parameter λ_k (at the moment truncated at 20). The initial parameters for the program are: number of classes, number of variables, initial number of components, proportions for the components, vector of means for each component, and common precision matrix. For the priors we need to set values for the hyperparameters for the mean (ξ and τ for the joint conjugate case or ξ and κ for the independent conjugate case), hyperparameters for the precision matrix (Ω and ν), and hyperparameter for the number of components λ_k. The choice of the values for the hyperparameters has been discussed in section 3.3.1, these values are also included in the *file.ini*. The choice of λ_k and τ_0 is dealt with later in this chapter (section 3.5.1). We take $\tau_0 = 1$ for the present section as it seemed to give enough flexibility to our modelling.

Selecting λ_k

As we mentioned in section 3.4.2 we will use Stephens (1997) algorithm in order to let the number of components in a class k_c vary. The prior for k_c used by Stephens (1997) is a doubly truncated Poisson in 0 and $kmax$ with parameter λ_k. For Stephens $kmax = 100$ but in our case $kmax = 20$ has been good enough as with the values of λ_k used the posterior for the number of components stays below 15. Stephens found that the posterior for the number of components is highly dependent on the choice of λ_k. We found this to be the case too; nevertheless the posterior for the number of components is a bit more stable with our code compared to his and we believe this is due to the use of the same precision matrix for all the components in a class. We can see the posteriors for the number of components using different values of λ_k in figure 3.9. In this case the Pima Indians data is analysed using glucose and body mass index as only variables in the analysis. As we can see in figures 3.10 to 3.12 although the sampling chains for the number of components vary depending on λ_k the likelihood chains remain roughly the same. At the same time the predictive density for each one of the components is not affected by a different use of λ_k, as long as the other parameters remain constant. Figure 3.13 shows the different joint predictive densities obtained using different values for λ_k. The length of the chains were determined by running the same model with 4 different starting points and using the marginal predictive densities to determine when these were similar enough so that the influence of the starting values was irrelevant (as described in section 2.2.3).

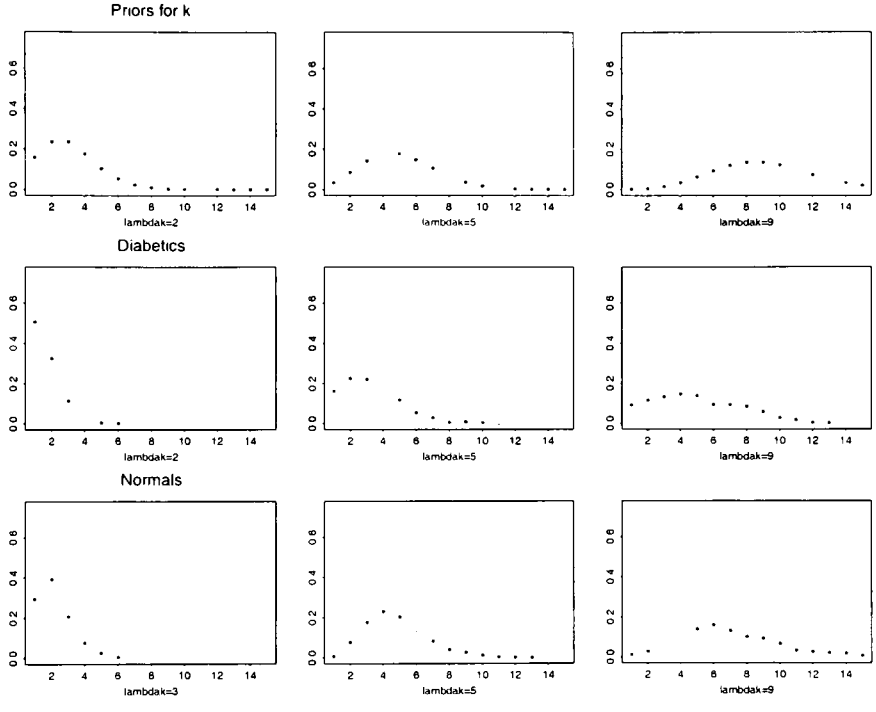


Figure 3.9: Prior and Posterior distribution for the number of components for both diabetics and non-diabetics in the Pima Indians data using different values of λ_k. The variables used are the only two continuous variables in the dataset glucose and body mass index. The training set consists of 68 diabetics and 132 non-diabetics. The number of total iterations was 2500 with 500 as burn–in period.

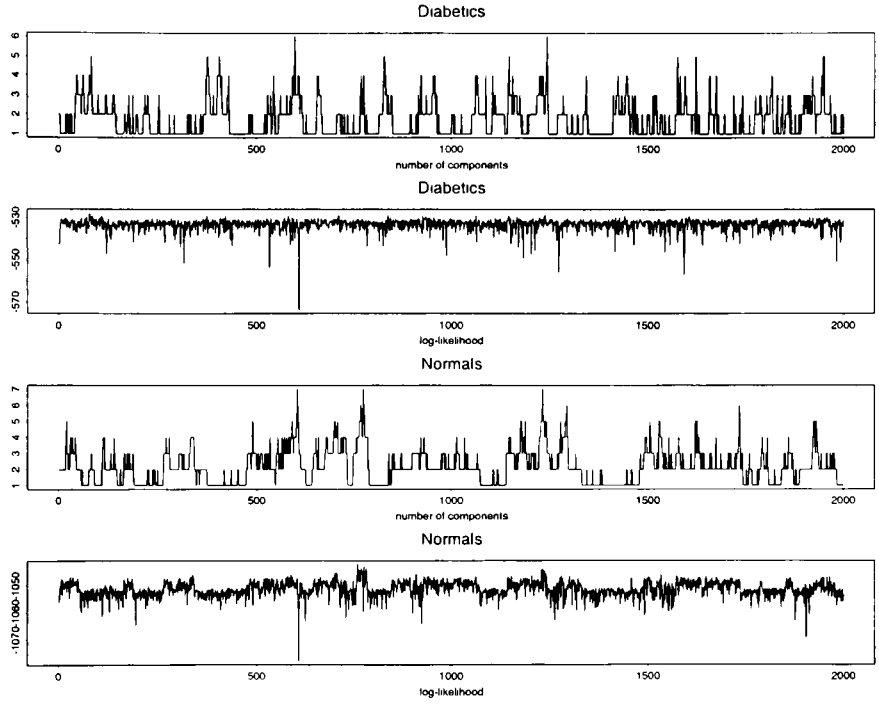


Figure 3.10: Sampling chain for number of components and log-likelihood for both diabetics and non-diabetics in the Pima Indians data with $\lambda_k = 2$. The variables used are the only two continuous variables in the dataset glucose and body mass index. The training set consists of 68 diabetics and 132 non-diabetics. The number of total iterations was 2500 with 500 as burn–in period.

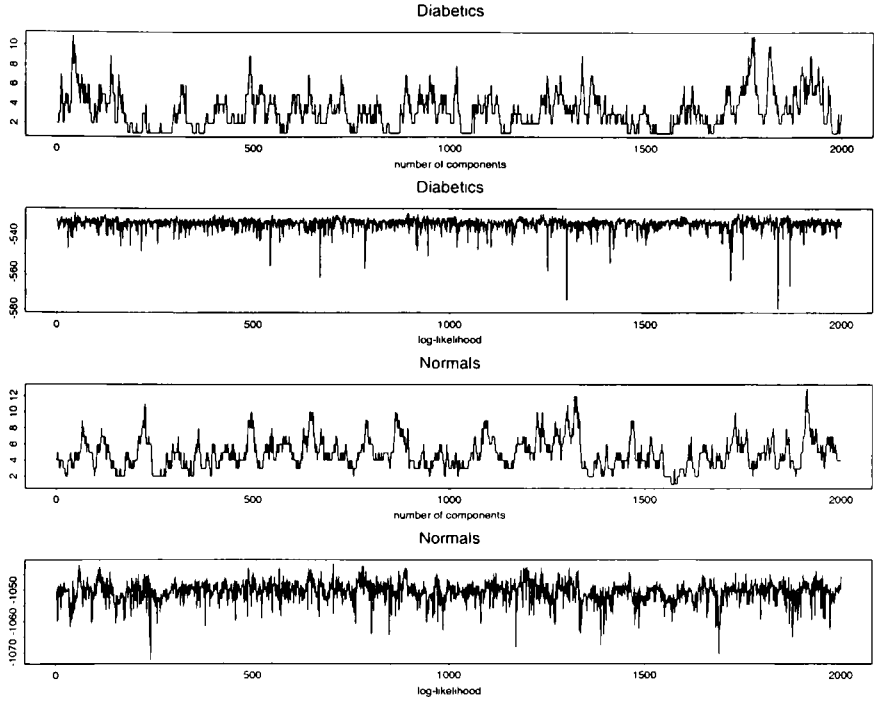


Figure 3.11: Sampling chain for number of components and log-likelihood for both diabetics and non-diabetics in the Pima Indians data with $\lambda_k = 5$. The variables used are the only two continuous variables in the dataset glucose and body mass index. The training set consists of 68 diabetics and 132 non-diabetics. The number of total iterations was 2500 with 500 as burn–in period.

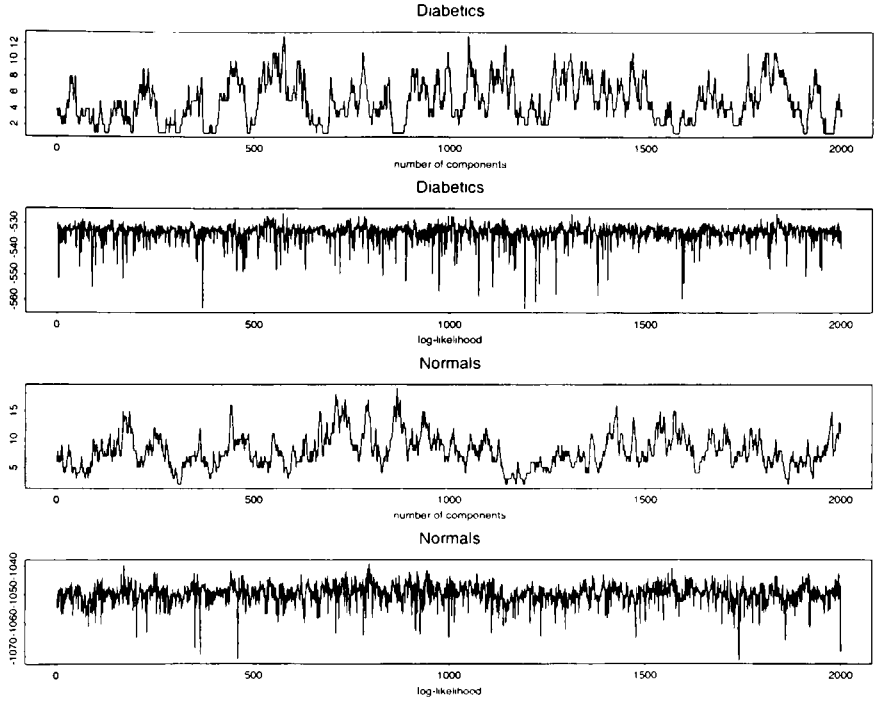


Figure 3.12: Sampling chain for number of components and log-likelihood for both diabetics and non-diabetics in the Pima Indians data with $\lambda_k = 9$. The variables used are the only two continuous variables in the dataset glucose and body mass index. The training set consists of 68 diabetics and 132 non-diabetics. The number of total iterations was 2500 with 500 as burn–in period.

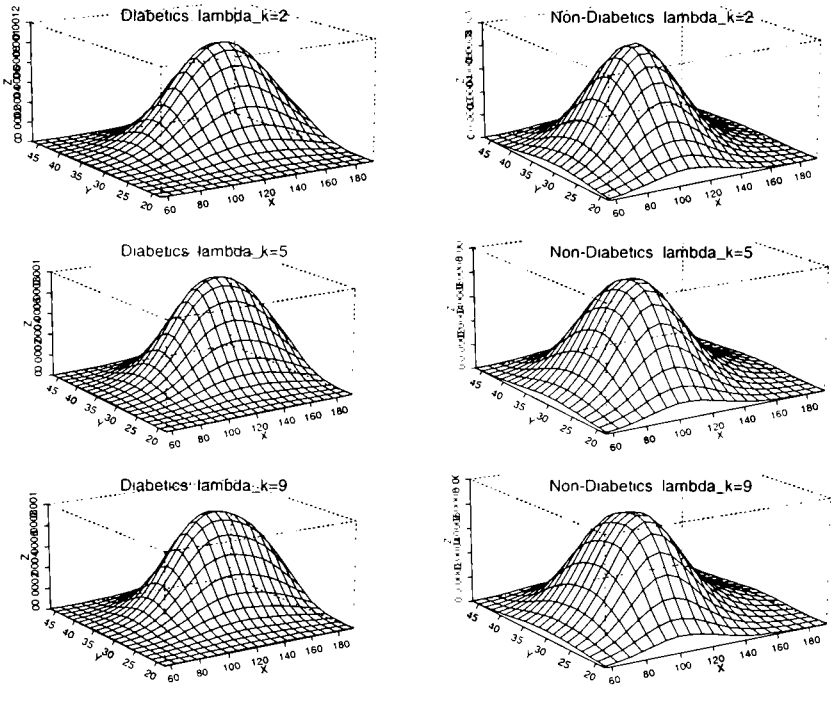


Figure 3.13: Joint predictive densities of the glucose and body mass index variables for the diabetics and non-diabetics in the Pima Indians data. The joint predictive densities were obtained using different values for λ_k and although the sampling chain for the number of components is different, the joint predictive densities are all effectively the same. The training set consists of 68 diabetics and 132 non-diabetics. The number of total iterations was 2500 with 500 as burn–in period.

Stephens (1997) reports a high dependence of the posterior of the number of components on the prior used. In his work the value of λ_k heavily determines the shape of the posterior for the number of components. While modelling with the same covariance matrix for all the components, the birth–death algorithm seemed a lot more stable than using different covariances for each component. For example in the case of the diabetics for the Pima Indians data (68 in total) when using $\lambda_k = 9$ for the case of equal covariances the posterior for the number of components has mode at 4, while using Stephens code the mode for the posterior is 8.

3.5 Variable against fixed number of components

The flexibility gained with the use of a random variable to determine the number of components can be observed by plotting the joint predictive densities obtained. In the case of the simulated data we know the number of components (3), nevertheless if we did not know the true number of components in the data we could model it using the number of components as a variable. This would produce a predictive density very similiar to that obtained with the right number of component as can be seen in figure 3.14. If the number of components used to model the data is greater or equal to the 'true' number of components, the predictive density does not alter greatly. Nevertheless if the number of components used to model is less than the true number, then the predictive density can give a deceptive idea of the distribution of the data. Stephens (1997) describes the same behaviour in all the examples used. It could be argued that a use of a high number of components would be the adequate solution, but this would complicate the computer process.

In the case of the crab data the joint predictive densities for the first and second principal components obtained using two components and a variable number of components show some variation. The variable number of components seem to give an average between two and one component as we see in figure 3.15. For this example we know that there are at least two different components for each class (male and female) nevertheless they are not as clearly separated in the blue crabs as they are in the orange crabs. As we see in figure 3.15 for the case of variable k the second principal component presents two modes for the orange crabs but only one for the blue crabs.

For the case of the forensic glass data we compared the marginal predictive densities obtained using variable k and a fixed $k = 3$. Ripley (1996) (page 211) fits mixtures of normal densities with common covariance matrices (for all components in all classes) and reports that for this case three components per class appears to be a reasonable compromise (he uses plug-in estimates). The predictive densities obtained for the first 6 principal components can be seen in figure 3.16 for variable k, and figure 3.17 using a mixture of

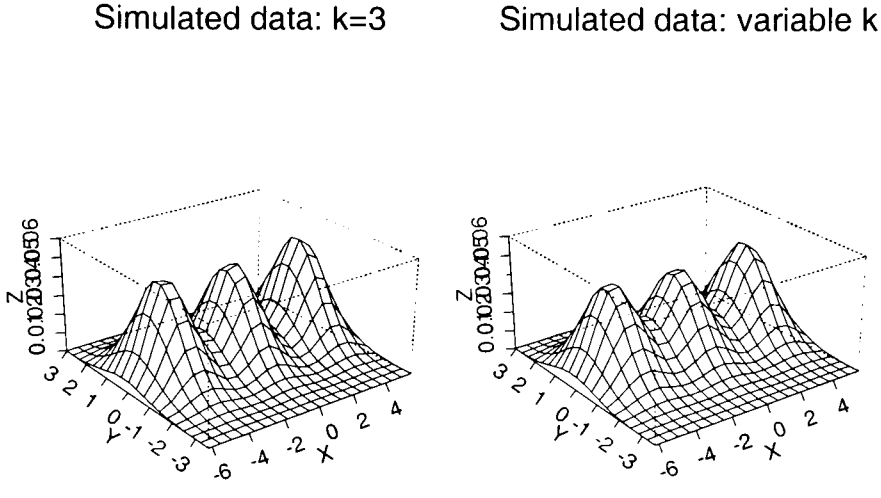


Figure 3.14: Joint predictive densities for the simulated data. Variable 1 is a mixture of three normal distributions with means $(-4, 0, 4)$ and unit variance, while variable 2 is a standard normal distribution. The top figures show the predictive density for these two variables given a fix number of components = 2. The bottom figures show the predictive density of the same variables obtained using a variable number of components. The data consist of 300 observations. The number of total iterations was 500 with 300 as burn–in period.

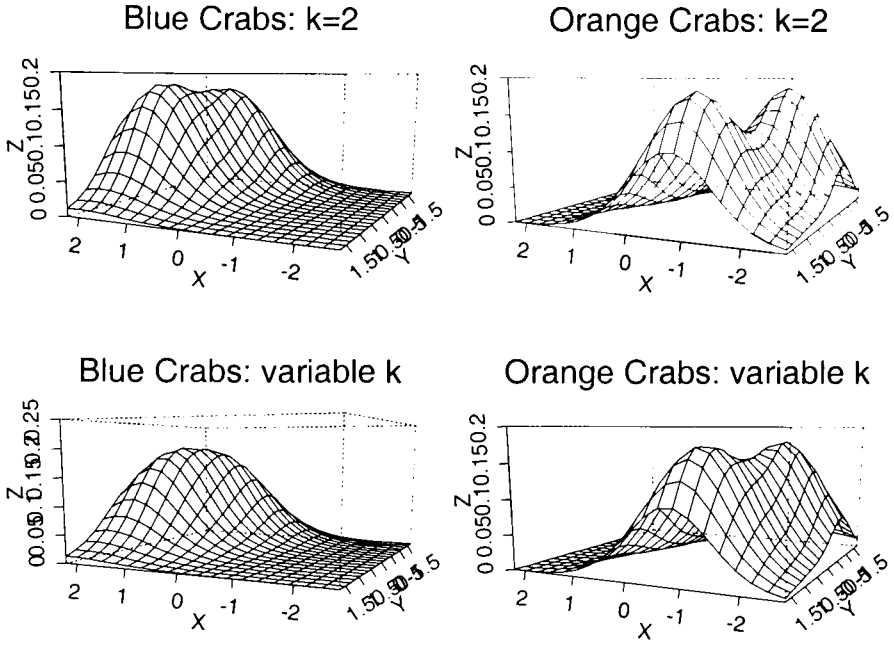


Figure 3.15: Joint predictive densities for the first two principal components in the crab data. The plots show the predictive densities for each class and how they are separated by the first principal component. The top figures show the predictive density for these two variables given a fix number of components = 2. The bottom figures show the predictive density of the same variables obtained using a variable number of components. The training set consists of 40 observations (20 for each class: 10 male, 10 female). The number of total iterations was 2500 with 500 as burn–in period.

3 normals for each class. As we can see in these figures, the predictive densities obtained for variable k and $k = 3$ are very different which would suggest that the proposed value of 3 mixtures for each class inadequate for this dataset at least with a common precision matrix Λ for all components in a class. We can see in the first principal component that the predictive densities for each class are very different in form. Nevertheless they are very close together and this will affect the classification (chapter 4).

3.5.1 The choice of τ_0 and λ_k

We have described that the choice of λ_k has an important effect on the posterior for the number of components k. At the same time the choice of τ_0 defines our certainty on the priors used. The values for the priors used so far for Ω and ξ are obtained from the whole dataset (regardless of class). The use of a low value ($\ll 1$) for τ_0 means that the posterior is heavily influenced by these priors (section 3.3.1). This would also reflect on the predictive distributions for the classes which would be similar to each other resulting in poor discrimination between classes.

A choice of $\tau_0 \geq 1$ reflects the lack of certainty in the priors used and the posterior density obtained will reflect the data a lot more closely and the classification will be better (assuming the training set has enough information). We can see different predictive densities for the Crab data using different values for τ_0 in figure 3.2. Nevertheless the choice of τ_0 is restricted by the birth and death part of the algorithm. A high value of τ_0 ($\gg 1$) means that we can be sampling from an area in which the data represent only a small portion. If this is the case, a new point for μ will have a high probability of not lying inside the data and the importance of this new value in the overall likelihood is small and its chances of dying in the next iteration are very high.

This means that models with $\tau_0 \gg 1$ will favour a low number of components but at the same time they describe the data better as they do not give added importance to the priors. In the parameterisation we are using with $\Sigma_i = \Lambda^{-1}$ for all Σ_i's in a class and with $\mu \sim \mathcal{N}(\xi, \tau_0(\Lambda)^{-1})$ we want $\tau_0(\Lambda)^{-1}$ to be roughly equal to the dispersion of the whole data as the new values obtained for μ will have a high probability of describing the data better and not dying in the next iteration.

For this reason we choose τ_0 to be equal to the total number of components, that is the sum of the number of components in each class. When the number of components are unknown we take λ_k to be a rough approximation to the true number of components in each class and we take $\tau_0 = \#classes \times \lambda_k$.

This choice of τ_0 gave good results as it also allowed us to obtain a stabler posterior for the number of components with a smaller dependence on our choice of λ_k. We can see the priors and posteriors for the number of components for a simulated example with two

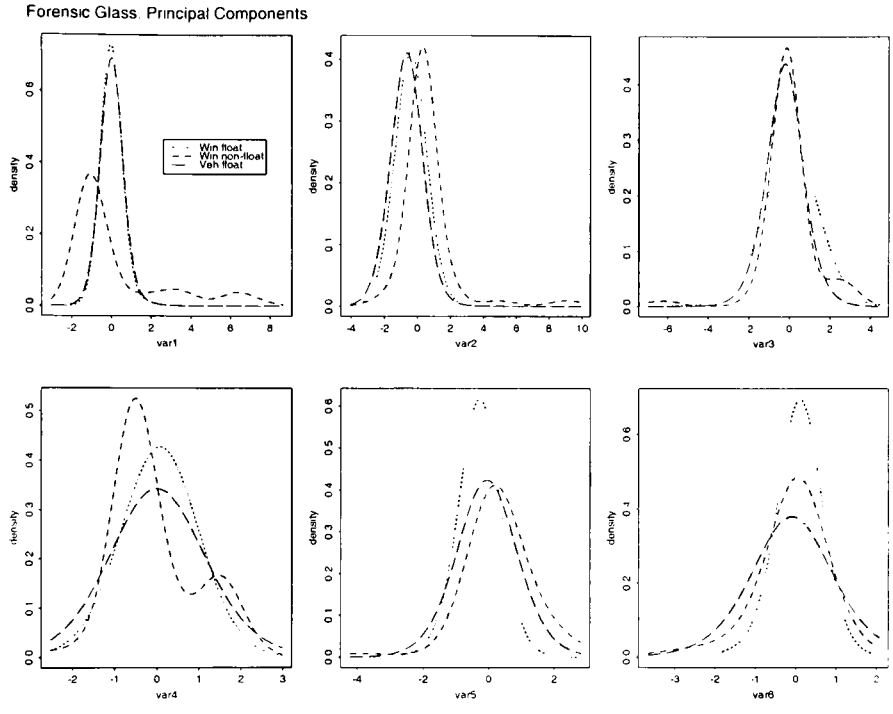


Figure 3.16: Predictive densities for forensic glass data. The variables used are the first 6 principal components. The plots show the predictive densities for each class. The first principal component shows a very different predictive density for each class. Nevertheless these densities are close together and this will reflect in the classification. These predictive densities were obtained using variable number of components. All the data from the windows groups were used in the modelling and consist of 70 building float, 76 building non-float, and 17 Vehicle float. The number of total iterations was 2500 with 500 as burn–in period.

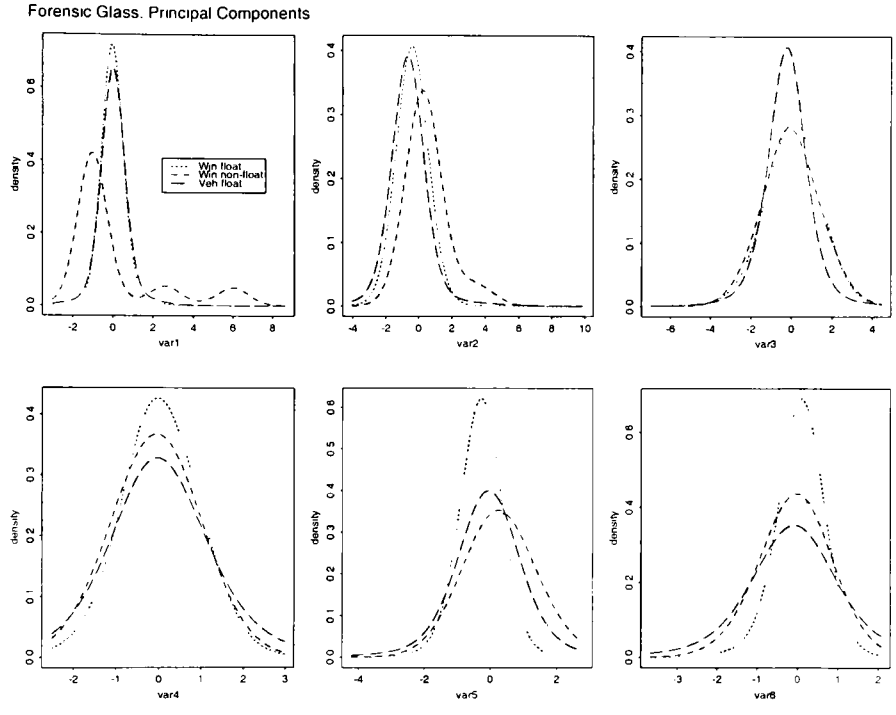


Figure 3.17: Predictive densities for forensic glass data. The variables used are the first 6 principal components. The plots show the predictive densities for each class. The predictive densities obtained for a fixed number of components= 3 do not differentiate the classes as well as with variable k. These predictive densities were obtained using a fix number of components = 3. All the data from the windows groups were used in the modelling and consist of 70 building float, 76 building non-float, and 17 Vehicle float. The number of total iterations was 2500 with 500 as burn–in period.

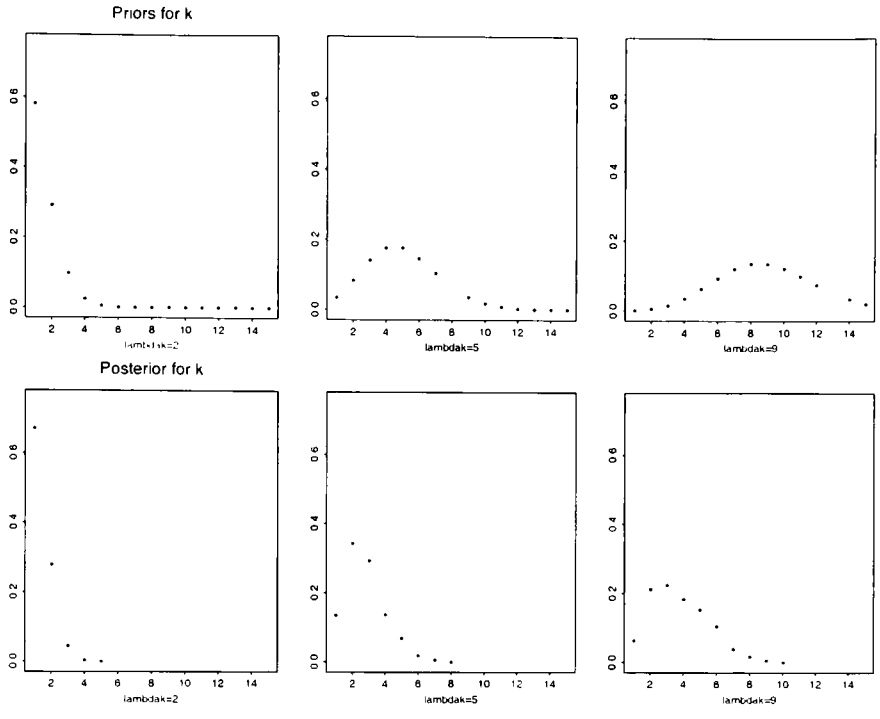


Figure 3.18: Prior and Posterior distributions for the number of components for the simulated example of two independent variables using different values of λ_k and $\tau_0 = \lambda_k$. The first variable is a mixture of three univariate normal distributions with means at $-4, 0, 4$ and unit variance and the second variate is a univariate standard normal distribution. The dataset consists of 300 datapoints with roughly the same proportion for the three groups. The number of total iterations was 3000 with 500 as burn–in period.

independent variables the first one a mixture of three normals with means at $-4, 0, 4$ and unit variance and the second a standard normal distribution in figure 3.18, for the Pima Indians obtained when using glucose and body mass index as variables in figure 3.19, and for the Crab data in figure 3.20.

Using these values for τ_0 and λ_k we can obtain an idea of the value of k_c when $\Sigma_{ic} = \Lambda_i^{-1}$. We notice that for the Simulated data and the Crabs data the true number of components (3 and 2) are clearly the modes for the posterior distributions for the number of components when we take a relatively large value for λ_k (9). We can also see a big difference between figure 3.9 and figure 3.19 which uses different value of λ_k with $\tau_0 = 1$ for the Pima Indians data (glucose and body mass index as variables). Stephens (1997) reported that a higher number of classes than the actual true number benefits the mixing process making

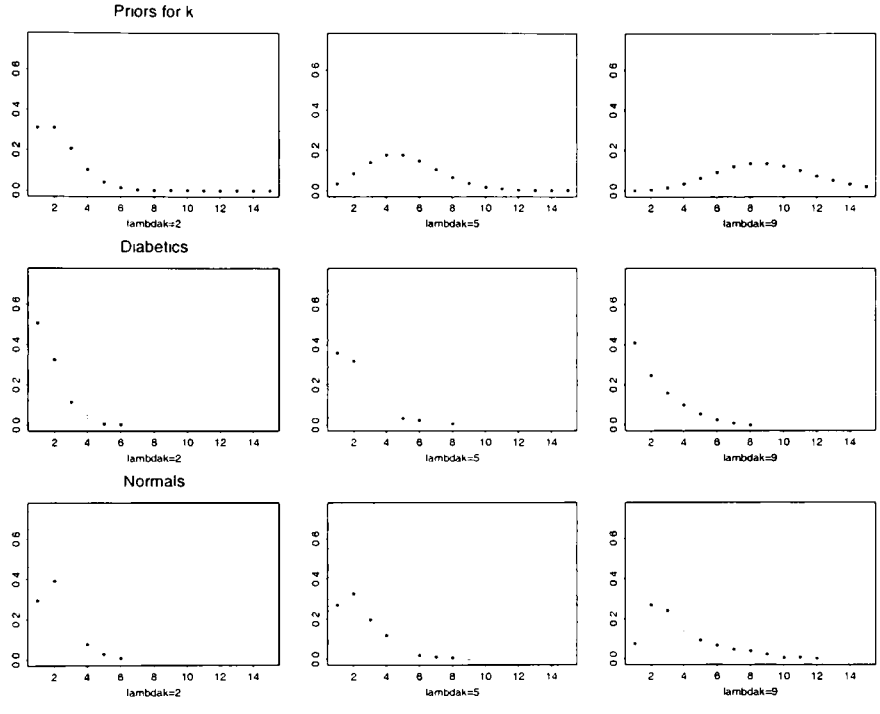


Figure 3.19: Prior and Posterior distributions for the number of components for both diabetics and non-diabetics in the Pima Indians data using different values of λ_k and $\tau_0 = \lambda_k \times 2$. The variables used are the only two continuous variables in the dataset, glucose and body mass index. The training set consists of 68 diabetics and 132 non–diabetics. The number of total iterations was 1000 with 500 as burn–in period.

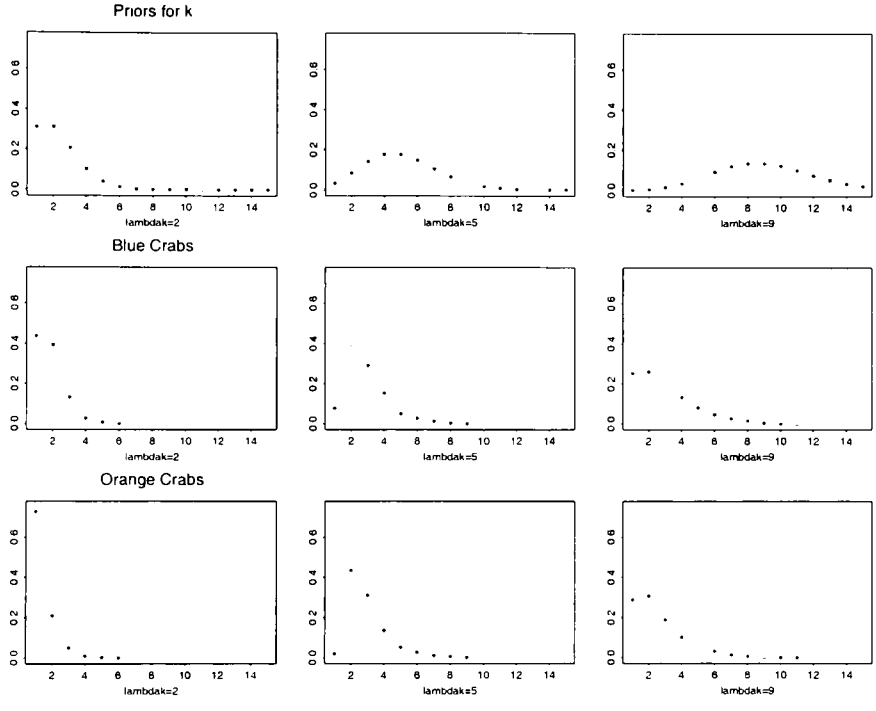


Figure 3.20: Prior and Posterior distributions for the number of components for the Bllue and Orange crabs in the crabs data using different values of λ_k and $\tau_0 = \lambda_k \times 2$. The variables used are the first 3 principal components. The training dataset consists of 20 Blue crabs (10 male and 10 female) and 20 Orange crabs (10 male and 10 female). The number of total iterations was 2500 with 500 as burn–in period.

our sampler more efficient, and we have observed the same behaviour in all our examples.

3.6 Using precision matrices with different sizes

So far we have restricted the covariance matrices of all the components in the mixture to be the same ($\Sigma_i = \Sigma$). This has the problem that there are cases when the number of components in a class is known but these components do not have the same covariance matrix and as such a poor sampling chain of the parameters could be obtained. For example suppose we know that a class consists of two components (multivariate normals), but one of them has most of its data close together while the other has its data a lot more dispersed. In this case the predictive density obtained with fixed number of components would give a model that would try to 'average' between the two different covariances and the fit of the model would be rather poor. Again thinking about classification in a multivariate case we can assume that the relationship of the variables within a class is roughly the same but some parts of a population have a larger spread in the variable space.

At the present two options are available, the use of the same covariance matrix for all the components or the use of a different covariance matrix for each component. Specifically in the multivariate case, the use of a different covariance matrix for each component might prove intractable because of the number of parameters we need to sample from. In our case where the number of components is not known and it is treated as a variable, the problem is even more complicated as we might be trying to model with more parameters than variables!

Another option was introduced by Banfield and Raftery (1993) which uses the decomposition of the covariance matrix into size, shape and orientation. A reparameterisation of the covariance matrix (Σ_i) for component i can relax the constraint of using the same covariance matrix while allowing the use of a similar structure for the covariances. Following the eigenvalue decomposition of the covariance matrix

$$\Sigma_i = \boldsymbol{D}_i \boldsymbol{\Psi}_i \boldsymbol{D}_i^T$$

where $\boldsymbol{D}_i$ is the matrix of eigenvectors and $\boldsymbol{\Psi}_i$ is a diagonal matrix with the eigenvalues of Σ_i. This diagonal matrix can be rewritten as $\boldsymbol{\Psi}_i = \zeta_i^{-1} \boldsymbol{A}_i$ where ζ_i^{-1} is the largest eigenvalue and $\boldsymbol{A}_i = \text{diag}(\alpha_{1i}, \ldots, \alpha_{pi})$, with $1 = \alpha_{1i} \geq \alpha_{2i} \geq \cdots \geq \alpha_{pi} > 0$ (we use ζ_i^{-1} for ease in our notation). Thus ζ_i^{-1}, $\boldsymbol{A}_i$, and $\boldsymbol{D}_i$ represent respectively the size, shape and orientation of the covariance matrix Σ_i. This parameterisation allows to constrain only one of the elements of the decomposition (e.g. shape $\boldsymbol{A}_i = \boldsymbol{A}$) while the other two vary depending on the component.

West *et al.* (1994) use a common factor σ_c ('size') for all the dispersion matrices in a mixture while allowing the covariances of the components Σ_{ic} to be different using as prior

for $\sigma_c = \mathcal{IG}(u/2, uU/2)$ with $u = 0$. We believe that the use of a different size for each component while keeping the same structure for the rest of the covariance will be more adequate for the type of modelling we are interested in. We are assuming that a population is composed of subpopulations and that each of these has roughly the same orientation and shape but it can be more or less dispersed in relation to another subpopulation in the class. We will allow ζ_i to vary where $(\Sigma_i)^{-1} = \zeta_i \Lambda$. In this new parameterisation $\theta = \mu, \zeta, \Lambda$, with Λ representing now the 'common' part of the precision matrix for all the components in one class. Liu and Rubin (1998) apply a similar decomposition of $\Sigma_i = (\zeta_i \Lambda)^{-1}$ to the general location model (not discussed here) allowing ζ_i to vary and keeping Λ the same for all their variables. As we are still looking at modelling only one class at a time, for this section we will use π_i for π_{ic}, μ_i for μ_{ic} and Λ for Λ_c; and when we write ζ_i we are refering to ζ_{ic}.

3.6.1 Full Conditionals allowing the sizes (ζ_i) to vary: joint conjugate priors

Banfield and Raftery (1993) use the largest eigenvalue as the size factor for the covariance matrix. One problem with using eigenvalues is that there is no closed form for these values and in our specific case we require to model these sizes with a known density in order to be able to sample from it. Nevertheless if we reparametrise the covariance matrix Σ from section 2.2 and instead use $\Sigma_i = (\zeta_i \Lambda)^{-1}$ we can obtain a closed form for the posterior density for ζ while the posteriors of μ and Λ remain roughly the same.

Using the unknown components setup (labels z as unknown data) we can separate the components of each class. Let us take the case of a single class m. We will look first at the posterior of $\mu_1, \mu_2, \ldots, \mu_k, \Lambda$, where Λ is the part of the covariance matrix common to all the components in a class. With the new parameterisation, the prior for μ, Λ (joint conjugate prior) changes to

$$p(\mu_1, \ldots, \mu_k, \Lambda) = p(\mu_1 \mid \Lambda, \zeta_1) p(\mu_2 \mid \Lambda, \zeta_2) \ldots p(\mu_k \mid \Lambda, \zeta_k) p(\Lambda \mid \zeta_1, \zeta_2, \ldots, \zeta_k)$$

$$\propto |\Lambda|^{(\nu-p-1)/2} \exp\left(-\frac{1}{2} \mathrm{tr}\,(\Omega_0 \Lambda) - \frac{1}{2\tau_0} \sum_{i=1}^{k} (\mu_i - \xi_0)^T \zeta_i \Lambda (\mu_i - \xi_0) \right)$$

Using the likelihood described in equation 2.1 the likelihood (for class m) is

$$p(x \mid \theta) \propto \left(\prod_{i=1}^{k} \zeta_i^{(pn_i)/2} \right) |\Lambda|^{n/2} \exp\left(-\frac{1}{2} \mathrm{tr}\left(\Lambda \left(\sum_{i=1}^{k} \zeta_i M_i \right) \right) \right), \tag{3.3}$$

with

$$M_i = \sum_{j:z_j=i}^{n_i} (x_j - \mu_i)(x_j - \mu_i)^T,$$

where n_i equal to the number of datapoints allocated to component i.

The posterior distribution for μ and Λ is then

$$p(\mu, \Lambda)p(x \mid \theta) \propto |\Lambda|^{(n+\nu-d-1)/2)} \exp\left(-\frac{1}{2}\mathrm{tr}\,(\Lambda V)\right),$$

where

$$V = \Omega_0 + \sum_{i=1}^{k} \zeta_i M_i + \frac{1}{\tau_0} \sum_{i=1}^{k} \zeta_i (\mu_i - \xi_0)(\mu_i - \xi_0)^T$$

Taking the last two components of the sum in the exponent

$$\begin{aligned}
A &= \sum_{i=1}^{k} \zeta_i M_i + \frac{1}{\tau_0} \sum_{i=1}^{k} \zeta_i (\mu_i - \xi_0)(\mu_i - \xi_0)^T \\
&= \sum_{i=1}^{k} \sum_{j:z_j=i}^{n_i} \zeta_i (x_j - \mu_i)(x_j - \mu_i)^T + \frac{1}{\tau_0} \sum_{i=1}^{k} \zeta_i (\mu_i - \xi_0)(\mu_i - \xi_0)^T \\
&= \sum_{i=1}^{k} \sum_{j:z_j=i}^{n_i} \zeta_i (x_j - \bar{x}_i)(x_j - \bar{x}_i)^T + \sum_{i=1}^{k} \sum_{j:z_j=i}^{n_i} \zeta_i (\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T \\
&+ \frac{1}{\tau_0} \sum_{i=1}^{k} \zeta_i (\mu_i - \xi_0)(\mu_i - \xi_0)^T
\end{aligned}$$

Taking the last two components of this sum

$$
\begin{aligned}
A2 &= \sum_{i=1}^{k} \sum_{j:z_j=i} \zeta_i (\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T + \frac{1}{\tau_0} \sum_{i=1}^{k} \zeta_i (\mu_i - \xi_0)(\mu_i - \xi_0)^T \\
&= \sum_{i=1}^{k} \zeta_i n_i (\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T + \frac{1}{\tau_0} \sum_{i=1}^{k} \zeta_i (\mu_i - \xi_0)(\mu_i - \xi_0)^T \\
&= \sum_{i=1}^{k} \zeta_i \left(n_i (\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T + \frac{1}{\tau_0} (\mu_i - \xi_0)(\mu_i - \xi_0)^T \right) \\
&= \sum_{i=1}^{k} \zeta_i \left[(n_i + \frac{1}{\tau_0}) \left(\mu_i - \frac{\frac{1}{\tau_0}}{\frac{1}{\tau_0} + n_i} \xi_0 - \frac{n_i}{\frac{1}{\tau_0} + n_i} \bar{x}_i \right) \left(\mu_i - \frac{\frac{1}{\tau_0}}{\frac{1}{\tau_0} + n_i} \xi_0 - \frac{n_i}{\frac{1}{\tau_0} + n_i} \bar{x}_i \right)^T \right] \\
&+ \sum_{i=1}^{k} \zeta_i \left[\frac{n_i \frac{1}{\tau_0}}{\frac{1}{\tau_0} + n_i} (\bar{x}_i - \xi_0)(\bar{x}_i - \xi_0)^T \right]
\end{aligned}
$$

So the posterior distribution of μ, Λ can be written as

$$
p(\mu, \Lambda \mid x, \zeta) \propto |\Lambda|^{(\nu^* - p)/2 - 1} \exp \left(-\frac{1}{2} \mathrm{tr}\,(\Omega^* \Lambda) - \frac{1}{2} \sum_{i=1}^{k} \frac{1}{\tau_i^*} (\mu_i - \xi_i^*)^t \zeta_i \Lambda (\mu_i - \xi_i^*) \right),
$$

with

$$
\nu^* = \nu + n, \tag{3.4}
$$

$$
1/\tau_i^* = n_i + 1/\tau_0, \tag{3.5}
$$

$$
\xi_i^* = \frac{\frac{1}{\tau_0}}{\frac{1}{\tau_0} + n_i} \xi_0 + \frac{n_i}{\frac{1}{\tau_0} + n_i} \bar{x}_i, \tag{3.6}
$$

$$
\Omega^* = \Omega_0 + \sum_{i=1}^{k} \zeta_i \sum_{j:z_j=i}^{n_i} (x_j - \bar{x}_i)(x_j - \bar{x}_i)^t + \sum_{i=1}^{k} \frac{\zeta_i n_i \frac{1}{\tau_0}}{\frac{1}{\tau_0} + n_i} (\bar{x}_i - \xi_0)(\bar{x}_i - \xi_0)^T \tag{3.7}
$$

Now to sample from the sizes ζ_i we assume a prior $p(\zeta_i)$ and look at the likelihood to obtain the form of the distribution and choose a conjugate prior for it. From equation 3.3 we can see that ζ_i is distributed as a Gamma in the likelihood and as such we choose a Gamma distribution with parameters ι and β_ζ as its prior. So the posterior for ζ_i looks like

$$p(\zeta_i \mid r, \mu_i, \Lambda) \propto \zeta_i^{\iota-1}\zeta_i^{(pn_i)/2} \exp\left(-\zeta_i\left(\beta_{\zeta 0} + \frac{1}{2}\mathrm{tr}(\Lambda M_i)\right)\right)$$

which gives a Gamma distribution with parameters $\iota_i^*, \beta_{\zeta i}^*$, with

$$\iota_i^* = \iota_0 + (pn_i)/2, \tag{3.8}$$

$$\beta_{\zeta i}^* = \beta_{\zeta 0} + \frac{1}{2}\mathrm{tr}(\Lambda M_i) \tag{3.9}$$

where

$$M_i = \sum_{j:z_j=i}^{n_i} (x_j - \mu_i)(x_j - \mu_i)^T,$$

n_i equal to the number of datapoints allocated to component i. An iteration for the Gibbs sampler algorithm 2.1 would be: sample the labels, sample the proportions given the labels, sample the $\mu_i's$ and Λ given the labels and the $\zeta_i's$, and sample the $\zeta_i's$ given the labels, the $\mu_i's$, and Λ.

The initial values for ζ_i are obtained from the prior which is a $\Gamma(\rho, \beta_\zeta)$ where $\beta_\zeta = (eigen(\Sigma_W))$, where $eigen(\Sigma_W)$ represents the largest eigenvalue of the dispersion of the whole dataset (using Banfield and Raftery (1993)). The values of ζ_i seem to be highly sensitive to our choice of β_ζ. This is mainly caused by the fact that we are not restricting Λ or $\zeta_i's$ and the same model can be obtained if we divide Λ and multiply the $\zeta_i's$ by the same constant. We did not normalize Λ_i at any point but this did not affect the sampling chain.

3.6.2 Full Conditionals allowing the sizes (ζ_i) to vary: independent conjugate priors

For the case when we restrict the precision matrices for all components in a class to have a common shape and orientation Λ the priors for μ_i and Λ are taken from section 3.2.2 and the prior for ζ_i is the same as in section 3.6.1. This gives the following full conditional posteriors for the means, the precision, and the sizes

$$f(\boldsymbol{\mu}_i \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\Lambda}^{(m)}) = \mathcal{N}(\xi_i^*, (\kappa_i^*)^{-1}),$$

$$f(\boldsymbol{\Lambda} \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\mu}_1^{(m)}, \ldots, \boldsymbol{\mu}_k^{(m)}) = \mathcal{W}(\Omega^*, \nu^*),$$

$$f(\boldsymbol{\zeta_i} \mid x, \boldsymbol{z}^{(m)}, \boldsymbol{\mu}_1^{(m)}, \ldots, \boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Lambda}^{(m)})) = \Gamma(\iota_i^*, \beta_{\zeta i}^*),$$

where $\nu^* = \nu_0 + n$, and

$$\begin{aligned}
\xi_i^* &= (\kappa + n_i\zeta_i\Lambda)^{-1}(\kappa\xi_0 + n_i\zeta_i\Lambda\bar{x}_i), \\
\kappa_i^* &= \kappa + n_i\zeta_i\Lambda, \\
\Omega^* &= \Omega_0 + \sum_{i=1}^{k} \zeta_i M_i, \\
\iota_i^* &= (pn_i)/2 + \iota, \\
\beta_{\zeta_i}^* &= \beta_\zeta + \frac{1}{2}\mathrm{tr}(\Lambda M_i),
\end{aligned}$$

where $\bar{x}_i$ is equal to the vector of means of the observations allocated to component i, and

$$M_i = \sum_{j:z_j=i} (x_j - \mu_i)(x_j - \mu_i)^T.$$

So step **c)** of algorithm 2.1 is obtained by sampling the means μ_i from the conditionals, using these means to sample the common precision matrix Λ_c and then using the means and common precision to sample the $\zeta_i's$. Figure 3.21 shows this model using independent conjugate priors for the means and precision in class c and with variable sizes for the precisions (with variable k_c).

3.6.3 Values for the Hyperparameters while using different Sizes.

The initial values for the parameters have to change to take into consideration the use of components with covariance matrices of different sizes. The initial parameters for $\alpha_1, \ldots, \alpha_k$ ($\alpha_i = 1$) and ξ =vector of means for the whole dataset (all classes included) remain the same as in section 3.3.1. Nevertheless for Ω_0 we need to take into consideration that Λ is now the common component of the precision. We take the highest eigenvalue for the dispersion matrix of the whole dataset Σ_W and this will be our value for parameter β_ζ. We then obtain the value of Ω_0 as the ratio between the dispersion matrix of the whole dataset and β_ζ. The initial degrees of freedom ν_0 and τ_0 are still the same, the number of variables p and the total number of components in the dataset; while the new parameter ι_0 is equal to the number of variables p. For the case when we are using independent conjugate priors the hyperparameter κ is obtained as in section 3.3.2.

3.6.4 Example of the use of Different sizes ζ_i

To show the effectiveness of this new parameterisation we introduce a new simulated example. This time we will use a mixture of two multivariate normal distributions, the first

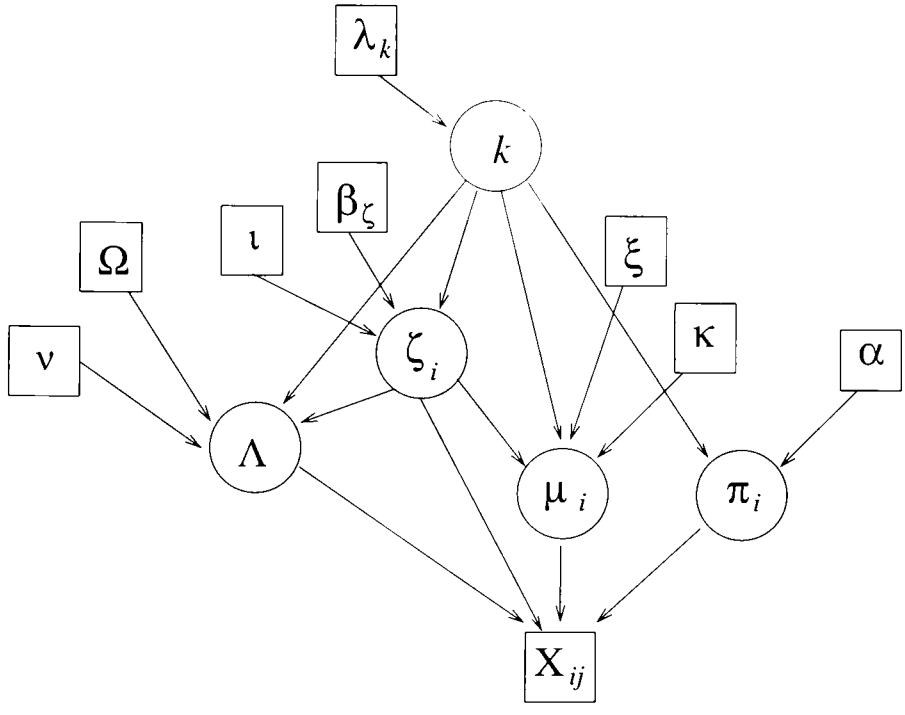


Figure 3.21: Directed acyclic graph of the parametric model used for mixtures of Normal distributions. This model takes into consideration only one class. The precision matrices for the components are allowed to have different sizes ζ_i but a common shape and orientation (given by Λ). So the covariance matrix for component i is now $\Sigma_i = (\zeta_i \Lambda)^{-1}$. In this model the number of components is treated as variable with a Poisson prior with parameter λ_k.

one with mean at $(0, 0)$ and dispersion matrix $\Sigma = [1, .2, .2, .5]$; the second a multivariate normal with mean at $(10, 10)$ and dispersion matrix 9Σ. We discuss these data in appendix A.

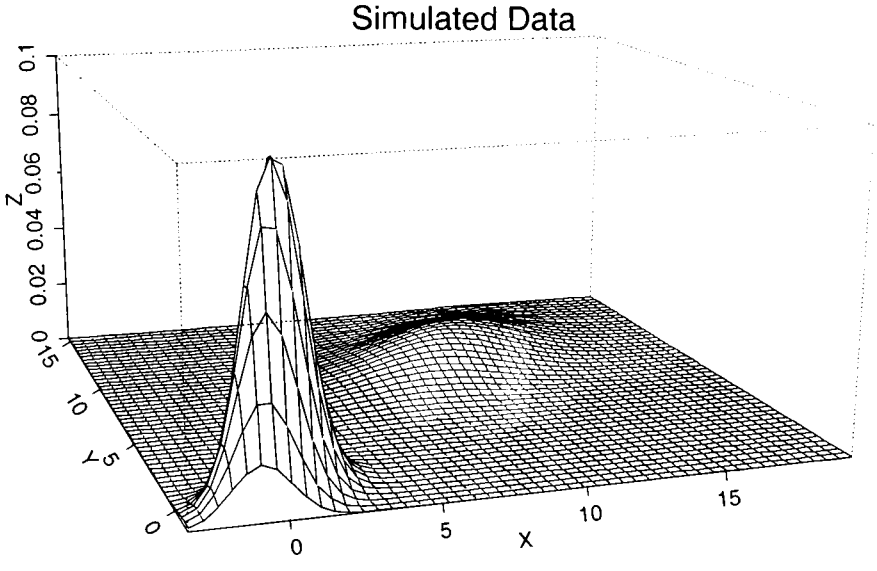
These data fit perfectly the model in which we allow the sizes of the dispersion matrices ζ_i to be different. The empirical distribution for these data can be seen in figure A.4 in the appendix A. We used a joint conjugate prior with variable sizes ζ_i to model these data. The initial values and hyperparameters were obtained as in section 3.6.3. The sampling chain ran for 1500 iterations with 500 burn–in period. Later runs with 10000 iterations and 9000 burn–in gave identical results. The joint predictive density for this dataset obtained using this parameterisation can be seen in figure 3.22. In figures 3.23 and 3.24 we can see the different joint predictive densities obtained with $\Sigma_i = \Lambda^{-1}$ for all components using a fixed number of components $k = 2$ and using variable number of components with birth–and–death/Gibbs algorithm. All figures have the same viewing point. In the case with variable k we used $\lambda_k = 2$; the number of components used to model the data varied from 3 to 6 staying mostly at 4 in this case.

As we can see from figures 3.23 and 3.24, the cases when the $\Sigma_i's$ are restricted to $\Sigma_i = \Lambda^{-1}$ gave a poor predictive density when compared to the empirical distribution. We use the same viewing point for all the figures to make the difference between them more easily interpreted. In the case with fixed number of components $k = 2$ the method is obtaining an 'average' of the dispersion matrices for the groups. When a variable k is used the group with the higher dispersion is sometimes modelled with more than one component but the predictive densities are still far away from the true density. The new parameterisation $\Sigma_i = \zeta_i\Lambda$ with fixed number of components $k = 2$ gives a good approximation to the true density. This is hardly suprising as the data were created on purpose for this kind of model.

We used another simulated dataset of two independent variables (one a mixture of three univariate normals and the other a standard univariate normal distribution). We obtained a predictive density which is very similar to the one obatined by fixing the precisions of all components in a class to be equal. The joint predictive density can be seen in figure 3.25 and can be compared to that obtained for the model with same precision matrices for all components 3.14. This means that this type of model will give more flexibility and accuracy for the cases when the precisions for the components in a class are different; while giving a good description of the data when the dispersions are the same.

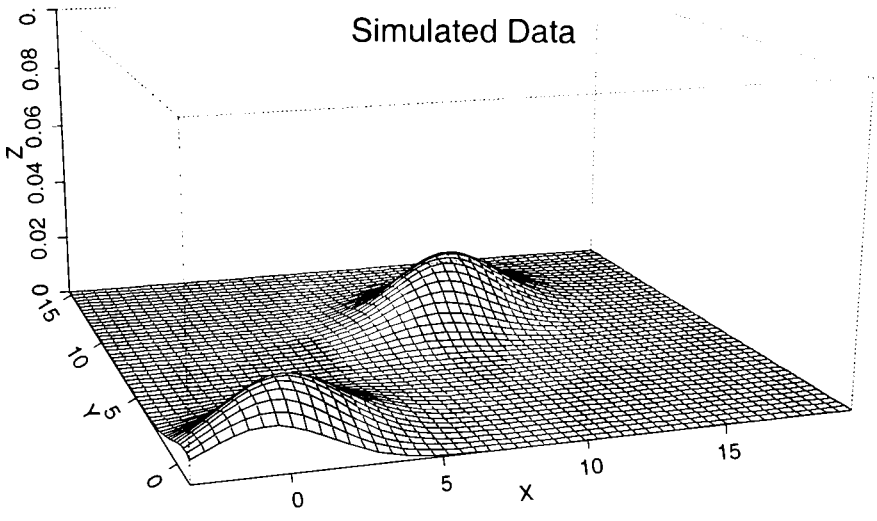
3.6.5 Hierarchical Modelling for the Sizes ζ_i

As we are modelling the different sizes ζ_i as a Gamma distribution, $\Gamma(\iota, \beta_\zeta)$, there is a probability that the sizes could increase making the common precision Λ decrease and



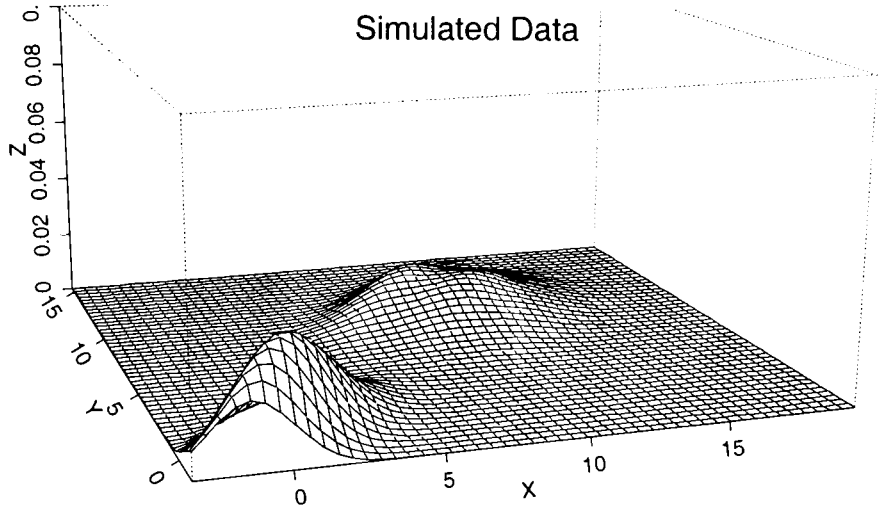
Variable sizes, k=2

Figure 3.22: Joint Predictive density for the simulated data of two multivariate normals with dispersion matrices of different sizes. A fixed number of components $k = 2$ and the parameterisation $\Sigma_i = \zeta_i \Lambda$ were used. The total number of iterations was 1500 and the burn–in period was 500. View point at $(-50, -104, .2)$



Common precision, k=2

Figure 3.23: Joint Predictive density for the simulated data of two multivariate normals with dispersion matrices of different sizes. The parameterisation used is $\Sigma_i = \Lambda$ (all dispersion matrices are the same). We use a fixed number of components $k = 2$. The total number of iterations was 1500 with a burn–in period of 500. View point at $(-50, -104, .2)$



Common precision, variable k

Figure 3.24: Joint Predictive density for the simulated data of two multivariate normals with dispersion matrices of different sizes. The parameterisation used is $\Sigma_i = \Lambda$ (all dispersion matrices are the same). A variable number of components was used with a Poisson prior with $\lambda_k = 2$. The total number of iterations was 1500 with a burn–in period of 500. View point at $(-50, -104, .2)$

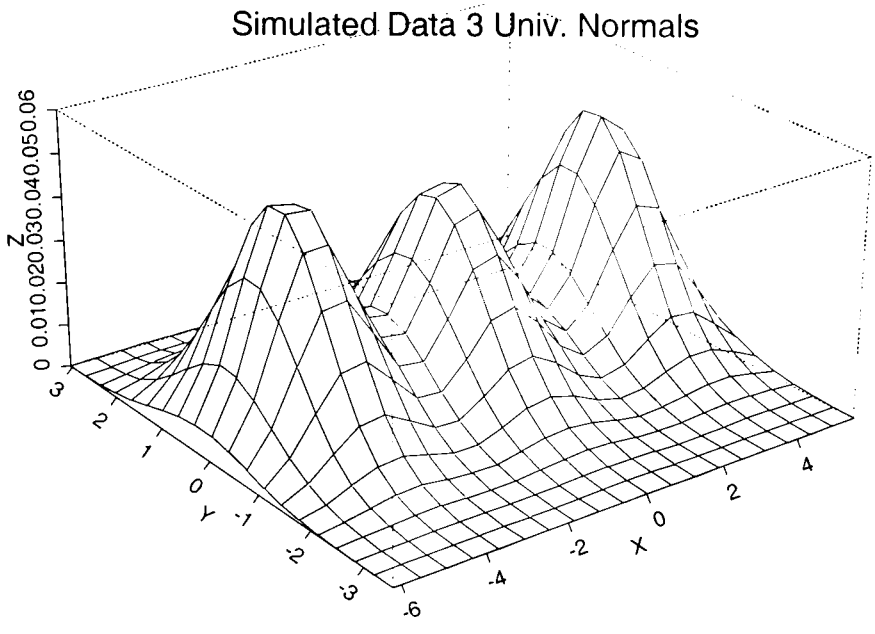


Figure 3.25: Predictive densities for the simulated data using different sizes ζ_i. Variable 1 is a mixture of three normal distributions with means (-4,0,4) and unit variance, while variable 2 is a standard. The total number of data-points is 300 (approximately 100 points for each component). A fixed number of components $k = 2$ was used but as we can see in the figure one of them has taken most of the data and the other one models only a few observations in the middle of the data.

get very close to zero as the sizes ζ_i take most of the weight. This does not seem to happen when we used a fixed value for β_ζ and although the sizes can increase from the initial values these sizes come to a certain stability quickly during the Gibbs sampling.

The fact that these values for ζ_i can be far away from the prior could mean that when we incorporate the birth–and–death part into the algorithm the values obtained for a new point at $(\pi, \mu, \zeta, \Lambda)$ will have in general low precision compared with other 'older' points. This could prove a problem when using a birth–death/Gibbs algorithm. For this reason we tried to obtain a value for β_ζ that would reflect the changes in the sizes ζ_i. To do this we used hierarchical modelling increasing our model by one level and letting β_ζ vary. We used a $\Gamma(1, 0.001)$ as prior as the posterior would look like a Gamma distribution as well. Nevertheless due to the sensitivity of the sizes ζ_i to β_ζ, when we let β_ζ vary the sizes increase rapidly, making Λ close to zero which reflects almost invariably into an error when the program is ran. For this reason we believe that in case the prior needs changing this should be done by modifying ι to which the posterior for the $\zeta_i's$ is less sensitive.

Chapter 4

Predictive Classification using Mixture Models

We have described in chapter 3 the algorithms and the models used to obtain a predictive density for a class. In this chapter we use these methods to obtain a classification for three real datasets, the Crabs data, the Pima Indians data, and the forensic glass data. The first one was chosen because of its simplicity and the fact that an excellent classification can be obtained with almost any method. The Pima Indians data and the forensic glass data present different problems. For example in the case of the Pima Indians most of the measured variables are not continuous and a transformation is required (to be able to model the data using mixtures of normals) and for the forensic glass we have 6 different classes some of them which are difficult to separate using most classification methods. All three datasets are described more accurately in appendix A.

As we mention in section 2.2 we will obtain a *predictive classification* of the data. A **predictive classifier** is *a classifier constructed by averaging over the uncertainty in the estimated parameter values* (Ripley 1996, page 352). The idea is to obtain a classifier for datapoint x_{new}, $\widetilde{p}(c \mid x_{new})$. In section 2.2 we describe this predictive classifier as

$$\widetilde{p}(c \mid x_{new}) = \frac{\lambda_c \widetilde{p_c}(x_{new})}{\sum_{l=1}^{C} \lambda_l \widetilde{p_l}(x_{new})}, \tag{4.1}$$

with

$$\widetilde{p_c}(x_{new}) = \int p(x_{new}; \phi_c) p(\phi_c \mid T) d\varphi, \tag{4.2}$$

where λ_c is the parameter for the proportion of class c in the population, and C equal to the total number of classes. If λ_c is unknown (like in our case) the natural choice of prior

is a Dirichlet with parameters (α_c) which gives a Dirichlet $(\alpha_c + n_c)$ as posterior. This makes the posteriors and priors for λ_c and ϕ_c independent. The difference between the MLE= $\frac{n_c}{n_T}$ for λ_c and its Bayesian counterpart is negligible unless the class sizes are very small or the priors (α_c) extremely strong (Ripley (1996)). For this reason we use the MLE as value for λ_c.

Using the methods developed in chapter 3 we can obtain a sampling chain for the parameters $\phi_c = (k_c, \pi_c, \theta_c)$. For notation purposes we will define the values for the parameters ϕ in the sampling chain as $\phi^{(m)}$, $m = 1, \ldots, Itot$, where $Itot$ = total length of the sampling chain minus the burn-in period. The burn-in period was obtained (as mentioned in section 2.2.3) by looking at the log-likelihood chains obtained from the MCMC process and determining when these log-likelihood chains appeared to have stabilised. We will discuss these burn-in periods in more detail for each database we studied in sections 4.1, 4.2, and 4.3.

We are modelling $p(x_{new}; \phi_c)$ as a finite mixture of multivariate normal distributions. With the sampling chain for the parameters of all classes obtained via the Gibbs sampler or birth-and-death/Gibbs sampler algorithms we can obtain an approximation to integral 4.2

$$\begin{aligned}
\widetilde{p}_c(x_{new}) &= \int p(x_{new}; \phi_c) p(\phi_c \mid T) d\phi_c \\
&= \int \left(\sum_{i=1}^{k_c} \pi_{ic} \mathcal{N}_p(x_{new}; \mu_{ic}, \Sigma_{ic}) \right) p(k_c, \pi_{\cdot c}, \mu_{\cdot c}, \Sigma_{\cdot c}) dk_c d\pi_{\cdot c} d\mu_{\cdot c} d\Sigma_{\cdot c}, \\
&\propto \frac{1}{Itot} \sum_{m=1}^{Itot} \sum_{i=1}^{k_c^{(m)}} \pi_{ic}^{(m)} \mathcal{N}_p(x_{new}; \mu_{ic}^{(m)}, \Sigma_{ic}^{(m)}),
\end{aligned}$$

where $\mathcal{N}_p(\cdot)$ represent a p-variate normal density, $\pi_{\cdot c} = \pi_{1c}, \ldots \pi_{k_c c}$, $\mu_{\cdot c} = \mu_{1c}, \ldots, \mu_{k_c c}$, and $\Sigma_{\cdot c} = \Sigma_{1c}, \ldots, \Sigma_{k_c c}$. So we can obtain an estimate for the predictive classifier by

$$\widetilde{p}(c \mid x) \propto \frac{\lambda_c \frac{1}{Itot} \sum_{m=1}^{Itot} \sum_{i=1}^{k_i^{(m)}} \pi_{ic}^{(m)} \mathcal{N}_p(x_{new}; \mu_{ic}^{(m)}, \Sigma_{ic}^{(m)})}{\sum_{l=1}^{C} \lambda_l \frac{1}{Itot} \sum_{m=1}^{Itot} \sum_{i=1}^{k_l^{(m)}} \pi_{il}^{(m)} \mathcal{N}_p(x_{new}; \mu_{il}^{(m)}, \Sigma_{il}^{(m)})}. \tag{4.3}$$

We use the same priors for all the classes $p(\phi_c) = p(\phi)$. If the chosen priors are too informative the posteriors for the classes will tend to be similar making discrimination between them difficult. At the same time the priors have to be proper otherwise the posteriors could be improper. For these reasons we try to create priors that are proper and at the same time give as little information as possible. We will see an example of how more informative priors affect classification in section 4.2. In our estimation of $\widetilde{p}(c \mid x)$ we will use the same $Itot$ for all classes although this could be modified very easily under

this setup. In fact we noticed that some of the classes 'converge' fairly quickly while some others take much longer (we describe this behaviour in more detail in section 4.3). As we can see with the estimator in equation 4.3 we allow the number of components in a class k_c to vary at each iteration. So far estimates that use plug-in methods do not allow k_c to vary and take the number of components for each class as known.

4.1 Crabs Data: *Leptograpsus variegatus*

These data contain 200 observations with 100 observations from each class of crabs (Blue and Orange) 50 of each sex (male and female). For the classification we decided to split the data into a training set of 40 observations (10 males and 10 females from each class) and a test set of the 160 observations remaining. The data were transformed to take into account the size by dividing by the square root of the width by the length of the carapace. Then the variables were standardised to unit variance. The first principal component separates quite well between orange and blue crabs and the classification obtained for most methods is very good. A description of this dataset can be seen in appendix A.

We obtained a first sampling chain for different models in order to obtain an idea of what to use as burn-in period. As we can see in figure 4.1 for both classes the log–likelihoods seem to stabilise quickly in the process so we decided to take a burn-in period of 500 although a smaller burn-in period could have been equally acceptable. The initial values and the values for the hyperparameters where obtained as described in sections 3.3.1, 3.3.2, and 3.6.3. The value of τ_0 used was 4 and the number of components $k_c = 2$ as we know that each class is composed by a male and a female group. As we discussed in section 2.2.3 after we selected the burn-in period each model was run with 4 different starting values in order to assess when the predictive densities obtained for each of the models is no longer being affected by our choice of initial values. We can see the marginal predictive densities for the three principal components in figure 4.2. These were obtained using a model that allows different precision matrices $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ and independent conjugate priors for μ_{ic} and Λ_c. For this specific case the ϵ used (from step 5 in the convergence strategy section 2.2.3) was 0.01.

The classifications obtained for these data were good, as expected. There does not seem to be much variation in our classifications regardless of the model used. In all but one model the number of misclassified are only 2 out of 160 datapoints in the test set. The only model for which the number of misclassifications is 3 the the one with variable sizes and joint conjugate priors for μ_{ic} and Λ_c. Although the classifications do not show any real difference between the models, we can see in figure 4.3 how the different models do affect the joint predictive densities obtained. In this figure the joint predictive densities for the first two principal components is shown for the different models. The first two

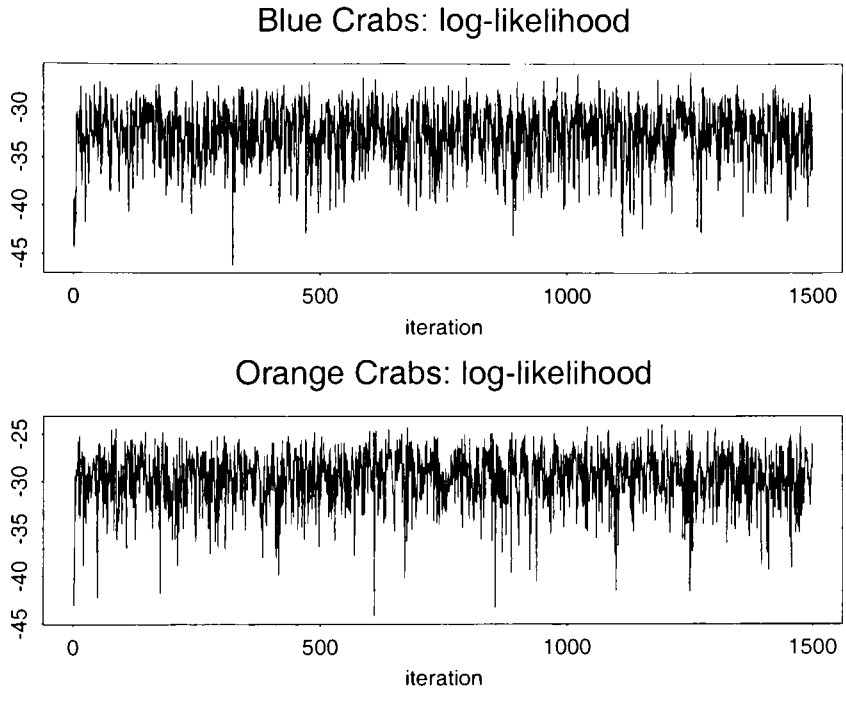


Figure 4.1: Log-likelihood chains obtained for the Crabs data (blue and orange species). The training dataset has 40 datapoints and consists of 20 crabs from each class (10 males and 10 females). The variables used in the analysis were the first three principal components. These log-likelihood chains were obtained using a fixed number of component for each class $k = 2$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_i)^{-1}$. These chains show the process from its starting point in order to determine a burn-in period to use. These chains allow us to see that the chain quickly stabilizes and the burn-in period required is not big.

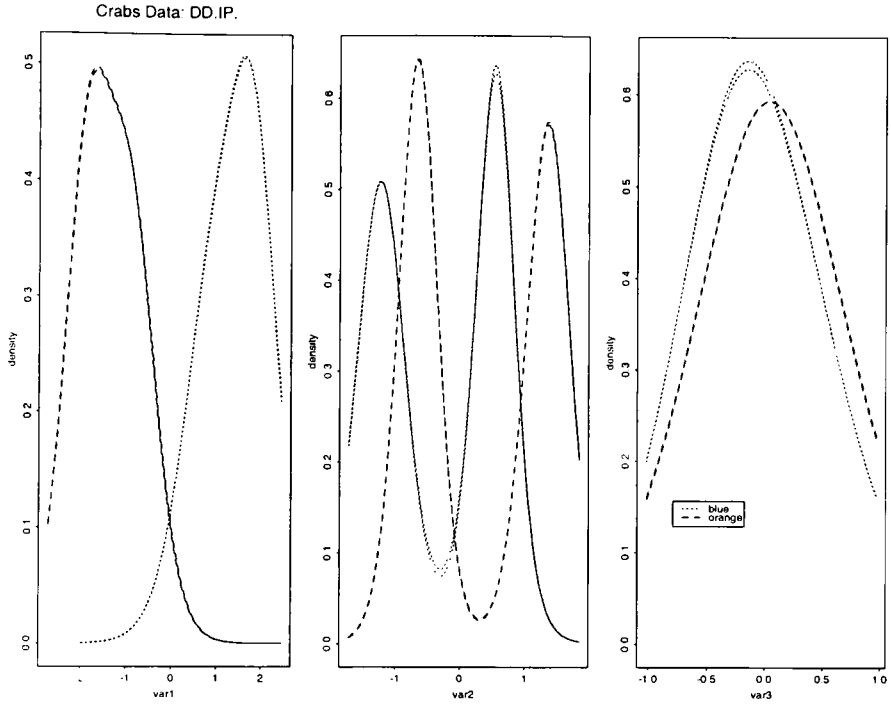


Figure 4.2: Marginal Predictive densities obtained for the Crabs data (blue and orange species) using 4 different starting values. The training dataset has 40 datapoints and consists of 20 crabs from each class (10 males and 10 females). The variables used in the analysis were the first three principal components. These marginal predictive densities were obtained using a fixed number of component for each class $k = 2$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ and uses independent conjugate priors for μ_{ic} and Λ_c. These predictive densities are very similar and have been obtained using a burn-in of 500 and 2000 iterations (after burn-in).

principal components have been chosen as the third principal component does not give any added information for classification. We can see in this figure (4.3) that the models that use independent conjugate priors for μ_{ic} and Λ_c tend to model the component in each class with more separation. This does not reflect in the classification as we are trying to discriminate between species but if we were required to discriminate between sexes then the models with independent conjugate priors would obtain better results than when using joint conjugate priors.

In this dataset we know the number of components for each class $k_c = 2$ but we decided to put to the test our use of a variable k_c. For this we used the birth-and-death/Gibbs algorithm. The plot of the joint predictive density obtained using variable k_c with a Poisson prior and $\lambda_k = 2$ can be seen in figure 3.15. For this model again the total misclassification was only 2 with 1 from each class being classified incorrectly.

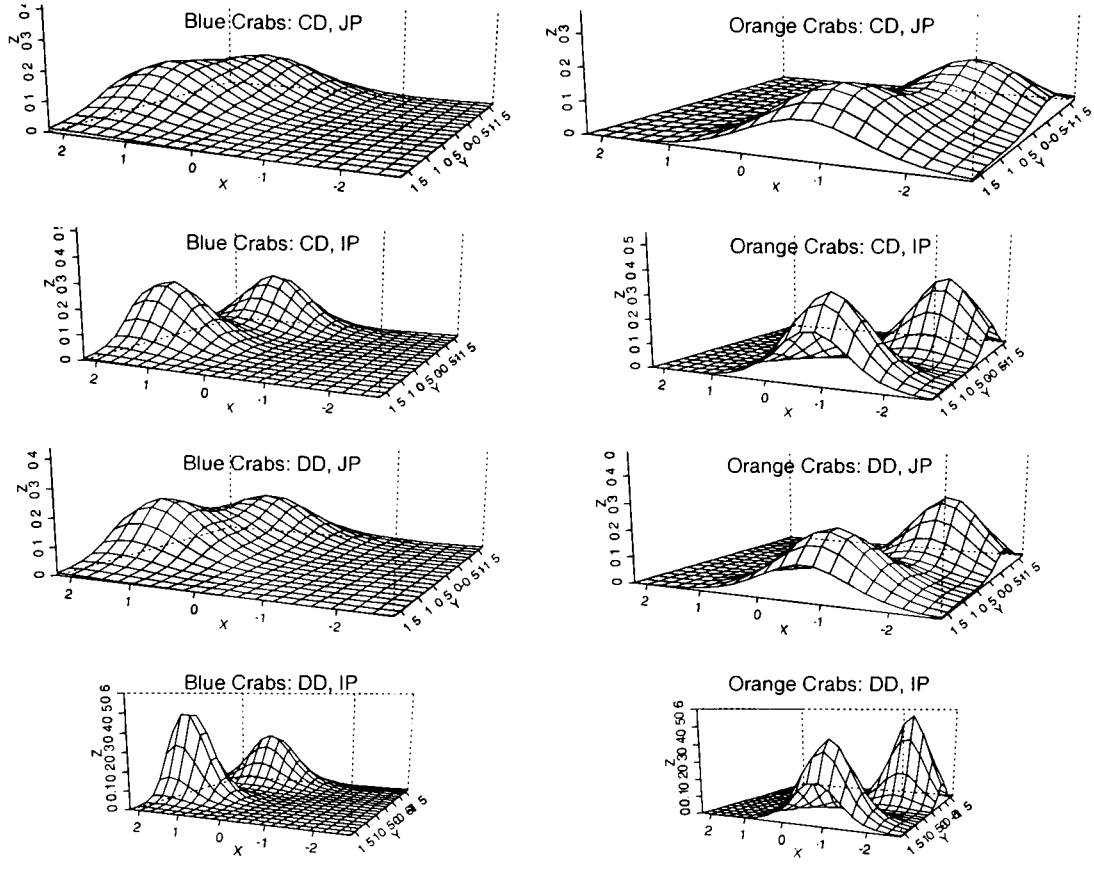


Figure 4.3: Joint Predictive densities obtained for the Crab data (blue and orange species) for all models. The training dataset has 40 datapoints and consists of 20 crabs from each class (10 males and 10 females). The variables used in the analysis were the first three principal components. The left column show the models obtained for the Blue species while the right column shows the ones obtained for the Orange species. From the top, the models used are 1) common dispersion with joint priors (CD JP), 2) common dispersion with independent priors (CD IP), 3) different dispersions with joint priors (DD JP), and 4) different dispersions with independent priors (DD IP). We can see how different models show a clearer (or less clear) separation between the components for each class.

4.2 Pima Indians Data

This dataset consists of 532 cases of Pima Indians that have been classified as being diabetic or not (all of them are women). We have divided into a training set (200) and a test set (332), with the training set consisting of 68 diabetics and 132 non-diabetics. A full description of the data can be seen in appendix A. We used different transformations to the variables; we use the Linear discriminant function, the glucose and body mass index, and the first 6 principal components in our analysis.

We first use the Linear Discriminant function (LDF) as our only variable (univariate case). In section 3.4.1 we use this variable and obtained a classification for the Pima Indians using different values for the hyperparameters ν_0 and Ω_0. The model used was a joint prior for μ_{ic} and Λ_c and a common precision matrix for all the components $\Sigma_{ic} = \Lambda_c^{-1}$ The initial values and values for the hyperparameters were obtained as described in section 3.3.1, with the value of $\tau_0 = 4$. We can see the classification obtained for them in table 4.1. For these classifications we used $k = 2$ for both classes (diabetics and non-diabetics). As we can see, in this specific dataset the best classification was obtained with the most vague priors ($\nu_0 = p, \Omega_0 = \Sigma_W$, Σ_W = dispersion matrix for the whole dataset). This could be due to the fact that the dispersion matrix for both classes is different and by making the priors more informative the posteriors for the classes become more similar which could then bias the classification towards the larger component (the non-diabetics). We believe that this will be the case regardless of the dataset and that the priors should be kept as vague as possible.

Still using the LDF as our only variable we obtained a log-likelihood chain and a chain for the number of components this time with variable k and with $\lambda_k = 2$ as we do not know the true number of components for either class. In figure 4.4 we can see how the chains appear to be stable from the beginning of the run and so we decided to take a burn-in period of 500 (although even a burn-in period of 0 could have been chosen). The different models were run with 4 different starting values and the criteria for convergence (see section 2.2.3) was achieved after just 1000 iterations after burn-in. The predictive densities obtained for these models show no clear difference between them and can be seen in figure 4.5. The classification obtained for the different models can be seen in table 4.2. Surprisingly we obtained the best classification using a common dispersion matrix for all components in a class ($\Sigma_{ic} = \Lambda_c$) with a joint conjugate priors for μ_{ic} and Λ_c. The number of misclassified for this setup was of 67 comparable to that reported by Ripley (1996) using LDA (66) and using mixtures with the LDF as variable (64).

We now use the glucose and the body mass index with no transformations. We use these two variables as they are the only ones in these data that are continuous. The initial values for the data and the values for the hyperparameters were obtained as described in sections

Table 4.1: Classification Tables for the test set of the Pima Indians data. From 332 total cases 223 are non-diabetics and 109 diabetics. The variable used for the classification was the Linear Discriminant function. A fixed number of components of $k = 2$ was taken for both classes. The model used for this classification is a common precision matrix for all the components in a class and a joint independent prior for μ_{ic} and Λ_c.

$\Omega_0 = \Sigma_W, \nu_0 = 2$

	Classif		
true	non-diab	diab	
non-diab	197	26	
diab	43	66	
		total	332

$\Omega_0 = \Sigma_W, \nu_0 = 8$

	Classif		
true	non-diab	diab	
non-diab	199	24	
diab	51	58	
		total	332

$\Omega_0 = \Sigma_W/2, \nu_0 = 2$

	Classif		
true	non-diab	diab	
non-diab	202	21	
diab	49	60	
		total	332

$\Omega_0 = \Sigma_W/2, \nu_0 = 8$

	Classif		
true	non-diab	diab	
non-diab	208	15	
diab	58	51	
		total	332

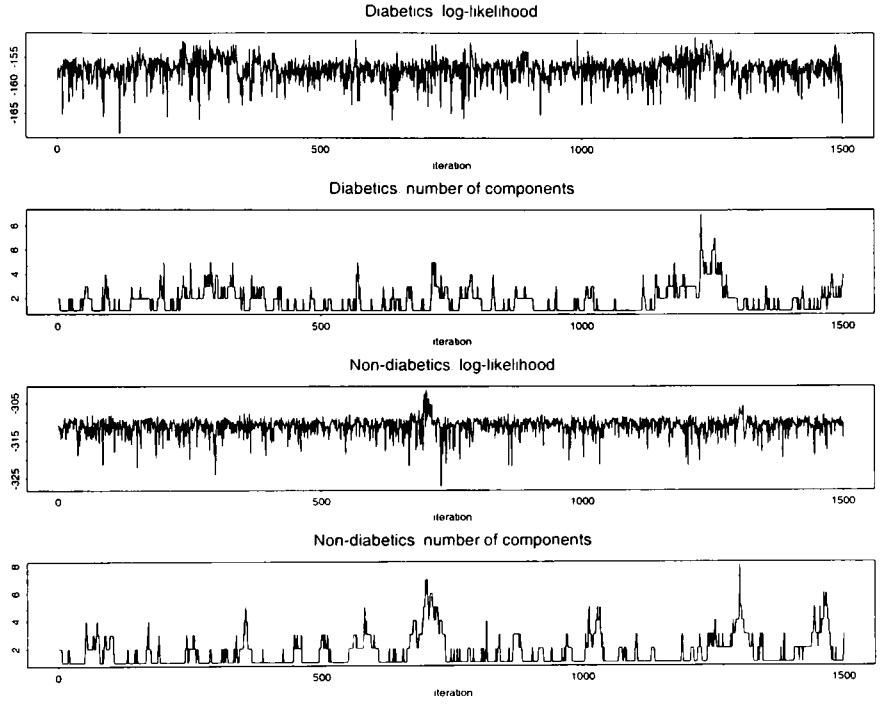


Figure 4.4: Log-likelihood chains obtained for the Pima Indians data (diabetics and non-diabetics). The training dataset has 200 datapoints and consists of 68 diabetics and 132 non-diabetics. The variable used in the analysis was the Linear Discriminant Function. These log-likelihood chains were obtained using a variable number of components for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 2$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$. These chains show the process from its starting point in order to determine a burn-in period to use. For both classes the log-likelihood chains start off already fairly stable so the burn-in period required was small.

Table 4.2: Classification Tables for the test set of the Pima Indians data. From 332 total cases 223 are non-diabetics and 109 diabetics. The variable used for the classification was the Linear Discriminant function. All the classifications have been done using variable k with $\lambda_k = 2$. The letters J.P. and I.P. stand for joint conjugate priors and independent conjugate priors for μ_{ic} and Λ_c.

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

	Classif		
true	non-diab	diab	
non-diab	198	25	
diab	42	67	
		total	332

$\Sigma_{ic} = \Lambda_c^{-1}$ I.P.

	Classif		
true	non-diab	diab	
non-diab	192	31	
diab	42	67	
		total	332

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

	Classif		
true	non-diab	diab	
non-diab	194	29	
diab	43	66	
		total	332

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ I.P.

	Classif		
true	non-diab	diab	
non-diab	195	28	
diab	43	66	
		total	332

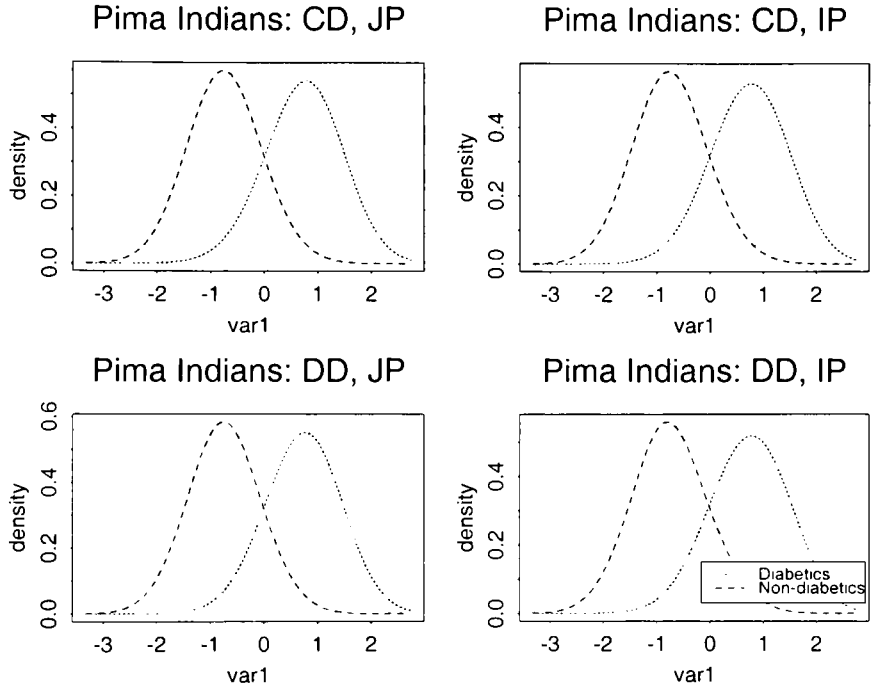


Figure 4.5: Predictive densities obtained for the Pima Indians data (diabetics and non-diabetics). The training dataset has 200 datapoints and consists of 68 diabetics and 132 non-diabetics. The variable used in the analysis was the Linear Discriminant function. From the top, the models used are 1) common dispersion with joint priors (CD JP), 2) common dispersion with independent priors (CD IP), 3) different dispersions with joint priors (DD JP), and 4) different dispersions with independent priors (DD IP). We really can see no difference between these densities, nevertheless there was some difference in the models for the classification (table 4.2).

3.3.1, 3.3.2, and 3.6.3. We used $\tau_0 = 4$ and $\nu_0 = 2$. A initial sampling chain was obtained to determine the burn-in period. These chains can be seen in figure 4.6 and again as in the case when we are using the Linear Discriminant function, the chains appear to be stable since the beginning of the run. We chose a burn-in of 500 although again 0 could have been equally effective. The total number of iterations (after burn-in) was of 2000. The joint predictive densities for the different models can be seen in figure 4.7 which show that these joint predictive densities for the different models appear to be extremely similar. This is also reflected on the classification as in this case three out of the four models gave the same classification with only the model with common dispersion matrices for all components in a class and independent conjugate priors for μ_{ic} and Λ_c giving a slightly better classification with 78 misclassified.

Table 4.3: Classification Tables for the test set of the Pima Indians data. From 332 total cases 223 are non-diabetics and 109 diabetics. The variables used for the classification were the glucose and the body mass index. All the classifications have been done using variable k with $\lambda_k = 2$. The letters J.P. and I.P. stand for joint conjugate priors and independent conjugate priors for μ_{ic} and Λ_c.

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

	Classif		
true	non-diab	diab	
non-diab	205	18	
diab	62	47	
		total	332

$\Sigma_{ic} = \Lambda_c^{-1}$ I.P.

	Classif		
true	non-diab	diab	
non-diab	205	18	
diab	60	49	
		total	332

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

	Classif		
true	non-diab	diab	
non-diab	205	18	
diab	62	47	
		total	332

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ I.P.

	Classif		
true	non-diab	diab	
non-diab	205	18	
diab	62	47	
		total	332

As our last approach to these data we used the first 6 principal components which account for 95% of the variability in the data. Again we obtained an initial run and the log-likelihood chains and the chains for the variable number of components can be seen in figure 4.8. This figure show how in this case the log-likelihood takes a lot longer to

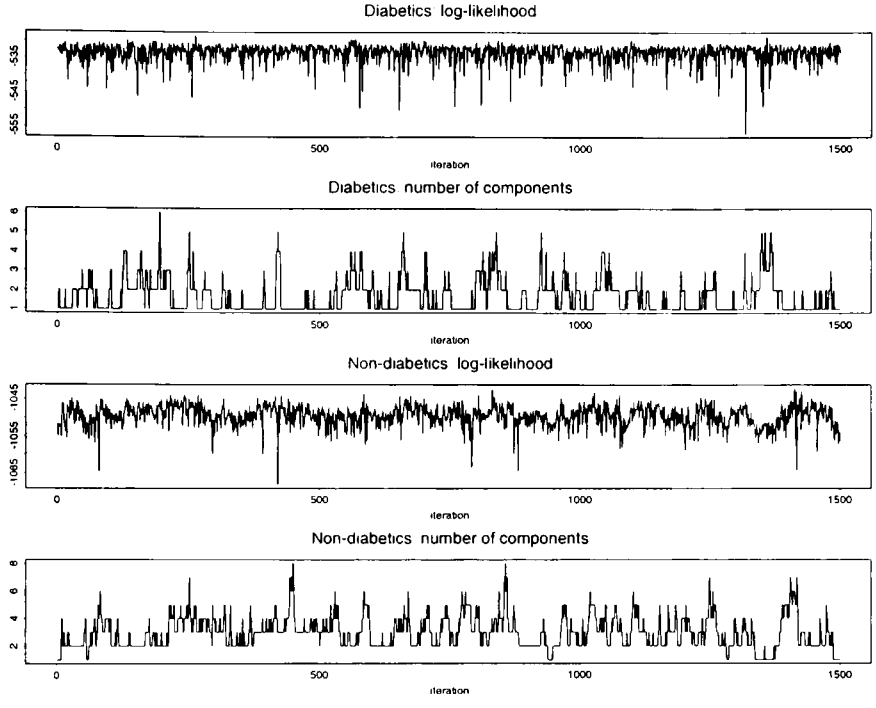


Figure 4.6: Log-likelihood chains obtained for the Pima Indians data (diabetics and non-diabetics). The training dataset has 200 datapoints and consists of 68 diabetics and 132 non-diabetics. The variables used in the analysis were the glucose and body mass index as they are the only two continuous variables in the dataset. These log-likelihood chains were obtained using a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 2$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$. As in the case when we use the LDF as our variable for classification, the process appears to be fairly stable. This could be mainly because both classes are not that well separated from each other and the initial values will tend to model the data fairly well for both classes.

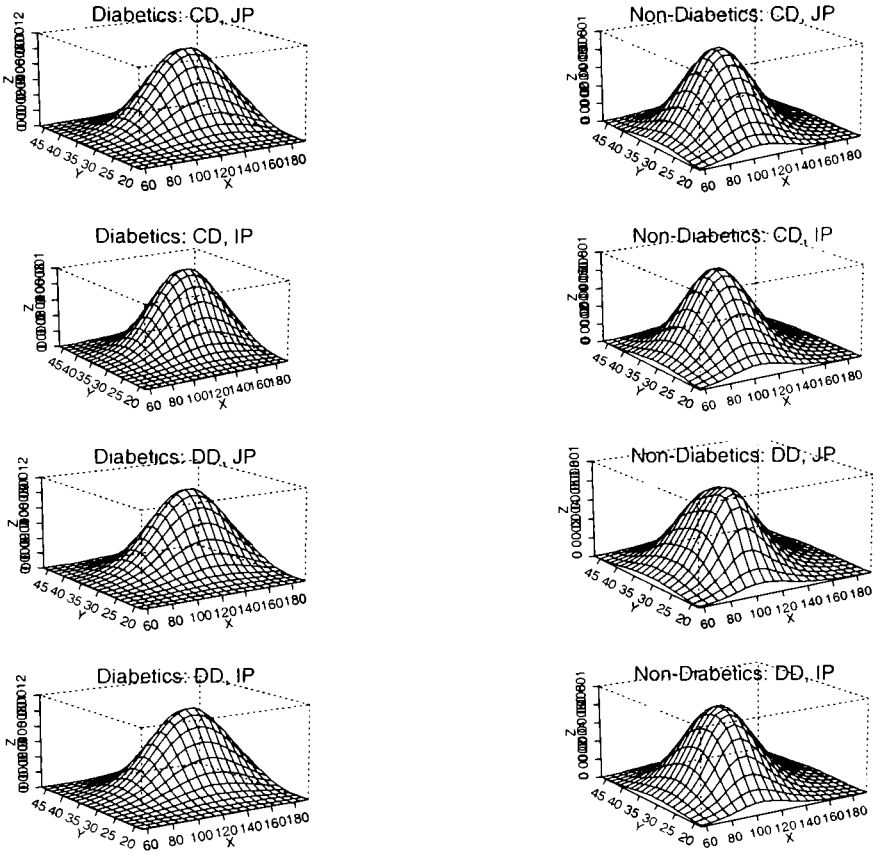


Figure 4.7: Joint Predictive densities obtained for the Pima Indians data (diabetics and non-diabetics). The training dataset has 200 datapoints and consists of 68 diabetics and 132 non-diabetics. The variables used in the analysis were the glucose and body mass index as they are the only two continuous variables in the dataset. The left column show the models obtained for the diabetics while the right column shows the ones obtained for the non-diabetics. From the top, the models used are 1) common dispersion with joint priors (CD JP), 2) common dispersion with independent priors (CD IP), 3) different dispersions with joint priors (DD JP), and 4) different dispersions with independent priors (DD IP). For these variables different models appear to give very similar joint predictive densities. This is reflected in the classification in table 4.3 where we can see all models giving almost the same classification.

stabilise than in the other two cases (using LDF or glucose and body mass index). This is hardly surprising as we have increased the number of dimensions and at least for the non-diabetics group the number of components required seems to be at least 6 or higher. It appears in the case of the diabetics the log-likelihood stabilises a lot sooner. We decided to take 500 as burn-in period as it appears that the log-likelihood chain for the non-diabetics has stabilised after 400 iterations. The different models we ran with different initial values and again convergence was achieved fairly quickly. The total number of iterations ran (after burn-in) was 2000.

We obtained the joint predictive densities for each class for the first two principal components as these two give the best discrimination between the two classes. As we can see in figure 4.9 the different models do give fairly different predictive densities. In particular it appears that the models that use different sizes for the precision matrices of the components ($\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$) give sharper peaks specially in the non-diabetics class. In the classification obtained for these different models the ones obtained for different precision matrices do marginally better than the ones with common precision matrices for all components in a class. These classifications can be seen in table 4.4.

As expected the biggest factor in the classification is the selection of the variables in the model. For this dataset we used the glucose and body mass index, the Linear Discriminant function, and the first 6 principal components. The classifications varied between the three and the best classification was obtained by using the linear discriminant function as reported by Ripley (1996, page 99). For the case when we are using this variable the number of misclassified is around 70 while for the other two cases the number of misclassification is around 80.

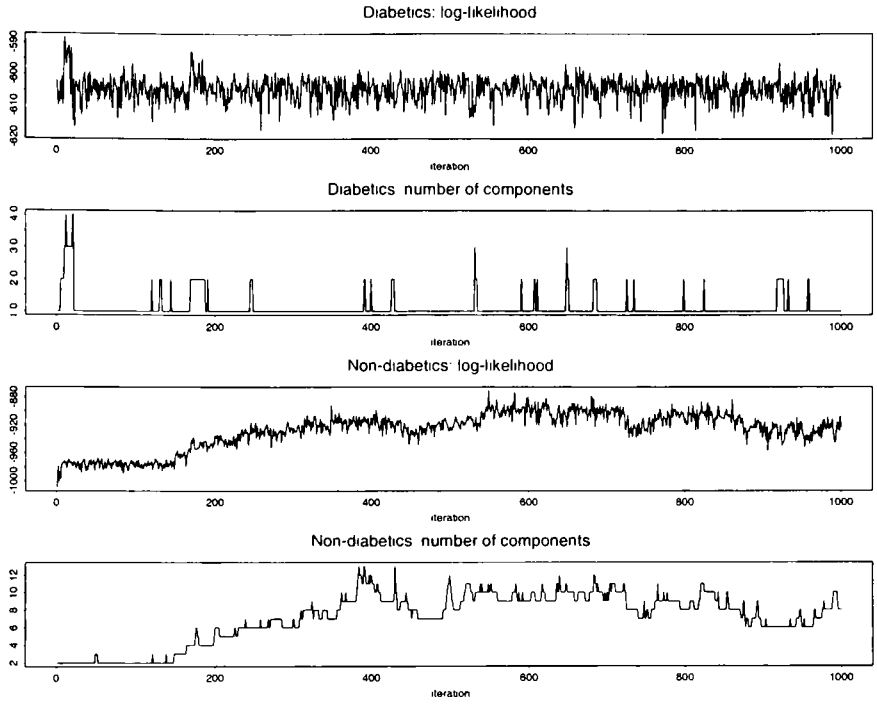


Figure 4.8: Log-likelihood chains obtained for the Pima Indians data (diabetics and non-diabetics). The training dataset has 200 datapoints and consists of 68 diabetics and 132 non-diabetics. The variables used in the analysis were the first 6 principal components as they account for 95% of the variability in the data. These log-likelihood chains were obtained using a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 2$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}.\Lambda_c)^{-1}$. As expected in this case where the dataspace is more complicated than the univariate or bivariate case, the chains took longer to stabilise. In the case of the non-diabetics it seems clear that the number of components required for this class is relatively high (≥ 6) compared to the diabetics class where one component seems to be enough. It appears that by iteration 500 the log-likelihoods for both classes have stabilised and we took this value as our burn-in period.

Table 4.4: Classification Tables for the test set of the Pima Indians data. From 332 total cases 223 are non-diabetics and 109 diabetics. The variables used for the classification were the first 6 Principal Components. All the classifications have been done using variable k with $\lambda_k = 2$. The letters J.P. and I.P. stand for joint conjugate priors and independent conjugate priors for μ_{ic} and Λ_c.

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

	Classif		
true	non-diab	diab	
non-diab	190	33	
diab	47	62	
		total	332

$\Sigma_{ic} = \Lambda_c^{-1}$ I.P.

	Classif		
true	non-diab	diab	
non-diab	190	33	
diab	48	61	
		total	332

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

	Classif		
true	non-diab	diab	
non-diab	189	34	
diab	44	65	
		total	332

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ I.P.

	Classif		
true	non-diab	diab	
non-diab	194	29	
diab	49	60	
		total	332

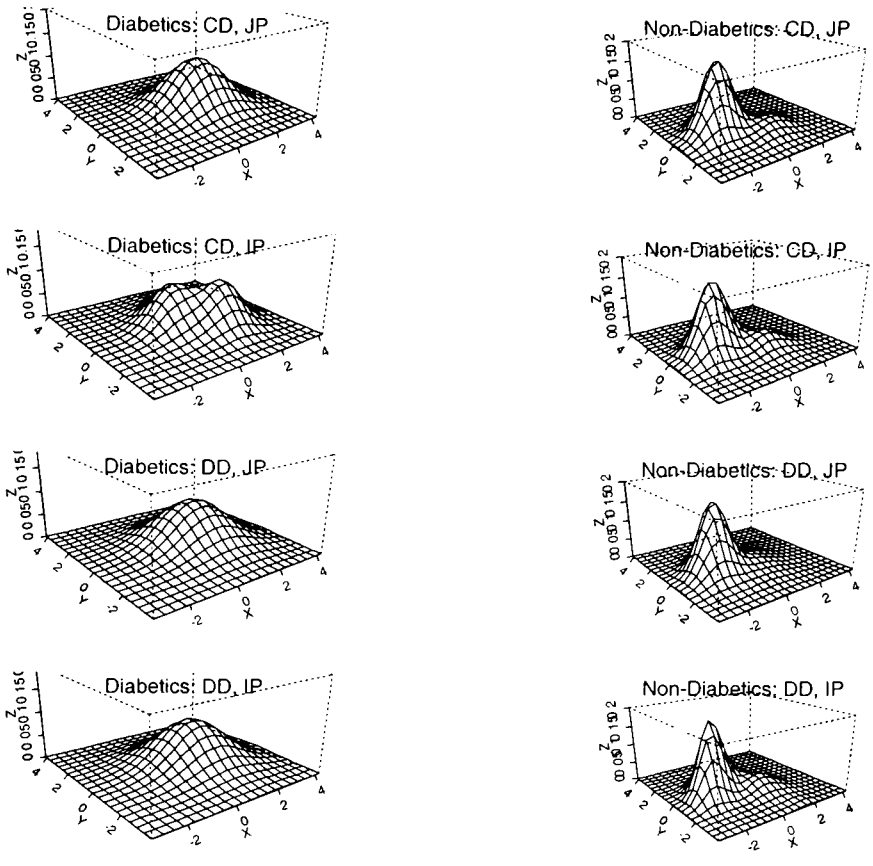


Figure 4.9: Joint Predictive densities obtained for the Pima Indians data (diabetics and non-diabetics). The training dataset has 200 datapoints and consists of 68 diabetics and 132 non-diabetics. The variables used in the analysis were the first 6 principal component. These joint predictive densities where obtained just for the first and second principal component as most of the information for discrimination between the two classes lie in these two variables. The left column show the models obtained for the diabetics while the right column shows the ones obtained for the non-diabetics. From the top, the models used are 1) common dispersion with joint priors (CD JP), 2) common dispersion with independent priors (CD IP), 3) different dispersions with joint priors (DD JP), and 4) different dispersions with independent priors (DD IP). The models that allow the sizes of the dispersions to be different give densities with sharper peaks this helped (marginally) the classification as we can see in table 4.4.

4.3 Forensic glass Data

This dataset consists of 214 datapoints with 6 different classes. The classes are: building windows float (70), building windows non-float (76), vehicle windows float (17), containers (13), tableware (9), and headlamps (29). There are 9 different measurements taken for these data, in order to reduce the dimension of the problem we will take the 5 canonical variates in one approach and then we will look more closely only at the windows classes (building windows float, building windows non-float, vehicle windows) as they seem to be the ones that are more difficult to obtain a good discrimination between. For this last approach we will use the first 6 principal components which account for 97% of the variability in the windows subgroup of the data. A full description of the data is given in appendix A

In order to be able to compare our classifications for these data to the ones obtained by Murphy and Aha (1995) and Ripley (1996) we decided to use the whole dataset as training set and test set (or all the cases of windows for the case when we are using the 6 principal components). Ripley (1996) uses most of the data for the modelling and the classifying by using cross–validation, on the other hand Murphy and Aha (1995) report the misclassification obtained using a package called BEAGLE, nearest–neighbours, and LDA. These methods involve the use of all the datapoints. In our setup the use of cross–validation is not practical although our classifications will be biased we assume this approach will allow us to obtain a comparison between our modelling and theirs.

First we look at all the classes and taking the 5 canonical variates as our variables to model the data. We can see the original classification for the data using the first 2 canonical variates in figure 4.10. In figure 4.11 we can see the log-likelihood chains and the chains for the number of components obtained using $\lambda_k = 3$. We use this value from Ripley (1996, page 211) where he suggest the use of a mixture of 3 normals with equal covariances to model the data. For the modelling of this dataset we will take this value for λ_k and when we model with a fixed number of components we will take $k_c = 3$ for all classes (table 4.6). As we can see in figure 4.11 the classes building windows non-float and headlamps take longer than the other classes to stabilise. It appears that all of them have stabilised after around 300 iterations so we took that initially as our burn-in period for all our models. For the models with common precision matrix for all components in a class we ran the process with 4 different starting values 2500 times (after burn-in) to achieve convergence, but for the cases when we were using variable sizes for the precision matrices of the components it was required to increase the burn-in time to 1000 and then run the process for an extra 2500 times to obtain predictive densities that were not affected by the starting values.

In figures 4.12 to 4.15 we can see the different marginal predictive densities obtained using the different models. We can see how the headlamps group is clearly distinguishable

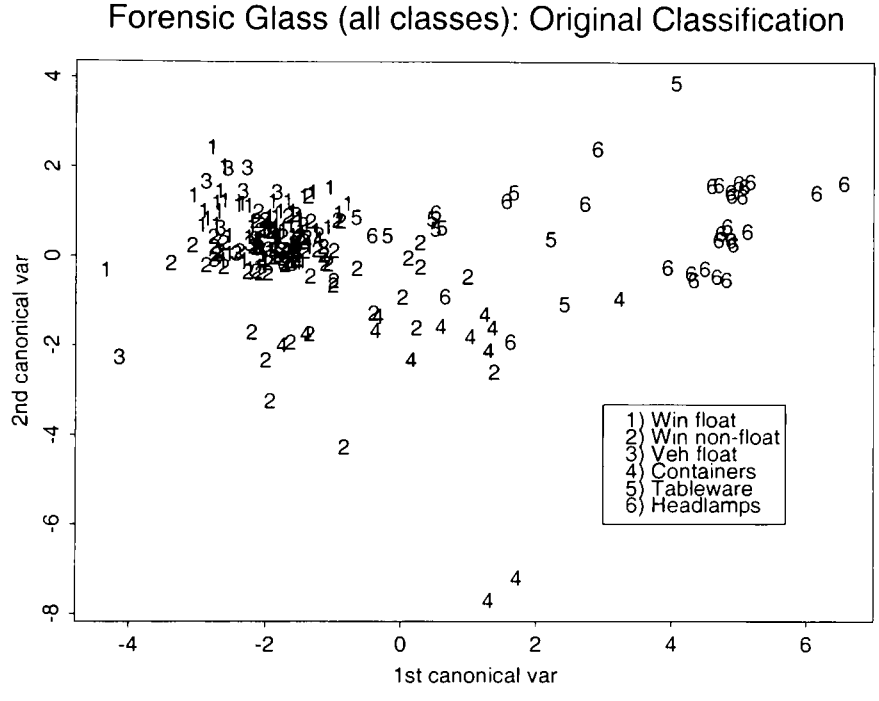


Figure 4.10: Original Classification of the forensic glass dataset. The classes are 1)building windows float (70), 2)building windows non-float (76), 3)vehicle windows float (17), 4) containers (13), 5) tableware (9), and 6) headlamps (29). The variables used for the plot are the first and second canonical variates. These show how the windows classes (1, 2, and 3) are fairly close together while the other classes are further away from these groups.

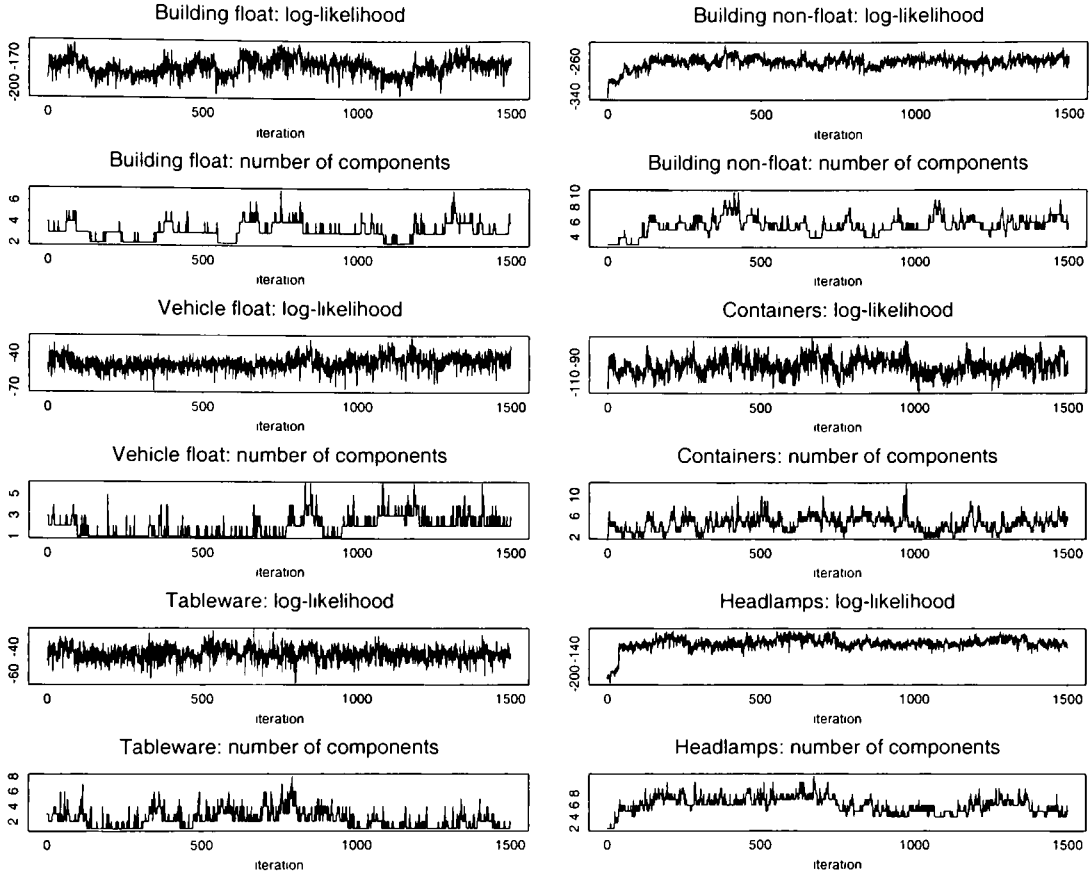


Figure 4.11: Log-likelihood chains obtained for the forensic glass data (all classes). The whole dataset was used as training set and test set and has 214 datapoints and consists of 70 building windows float, 76 building windows non-float, 17 vehicle windows float, 13 containers, 9 tableware, and 29 headlamps. The variables used in the analysis were the canonical variates (obtained using S-plus command lda). These log-likelihood chains were obtained using a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$. In this dataset some classes take longer to stabilize than others. In this particular case it appears that classes building windows non-float and headlamps take around 200 iterations to stabilise, while other classes like vehicle float appear to be fairly stable from the start.

Table 4.5: Confusion matrices for the forensic glass data. The whole dataset was used as training and test set (214 datapoints). The variables used for the classification were the 5 canonical variates. The classifications have been done using variable k with $\lambda_k = 3$. The letters J.P. and I.P. stand for joint conjugate priors and independent conjugate priors for μ_{ic} and Λ_c.

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

true	B. float	B. n-float	V. float	Contain	Tablew	Headlmp
B. float	55	15	0	0	0	0
B. n-float	20	53	1	1	1	0
V. float	9	3	5	0	0	0
Contain	0	0	0	13	0	0
Tablew	0	2	0	0	6	1
Headlmp	1	0	0	0	1	27

$\Sigma_{ic} = \Lambda_c^{-1}$ I.P.

true	B. float	B. n-float	V. float	Contain	Tablew	Headlmp
B. float	52	16	2	0	0	0
B. n-float	17	58	1	0	0	0
V. float	7	4	6	0	0	0
Contain	0	2	0	10	1	0
Tablew	0	3	0	0	6	0
Headlmp	0	1	0	0	1	27

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

true	B. float	B. n-float	V. float	Contain	Tablew	Headlmp
B. float	54	13	3	0	0	0
B. n-float	12	59	3	1	1	0
V. float	5	2	10	0	0	0
Contain	0	0	0	13	0	0
Tablew	0	0	0	0	9	0
Headlmp	0	0	0	0	1	28

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ I.P.

true	B. float	B. n-float	V. float	Contain	Tablew	Headlmp
B. float	55	11	4	0	0	0
B. n-float	13	61	1	0	1	0
V. float	2	1	14	0	0	0
Contain	0	0	0	13	0	0
Tablew	0	0	0	0	9	0
Headlmp	0	0	0	0	0	29

using the first canonical variate while with the same variate all the windows classes are fairly close together and containers and tableware are close to each other as well. The containers class is very different to the other classes if we look at the second canonical variate. As we can see from these figures the biggest confusion lies within the windows classes (building windows float, building windows non-float, and vehicle windows float).

The classifications obtained for the different models can be seen in table 4.5. It is important to notice that for the case when we are modelling the classes allowing different sizes for the precision matrices ($\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$) and independent conjugate priors for μ_{ic} and Λ_c, for the tableware class (9 datapoints) twice during the sampling chain (total length 2500) we used 8 components. Nevertheless most of the time for this particular model we use only 3 components. As we mentioned before these classifications are certainly biased as we are using the same dataset as training and test set; nevertheless we can compare the different models to determine which one fits the data better.

In table 4.5 we can see that the use of independent conjugate priors instead of joint conjugate priors for μ_{ic} and Λ_c (regardless of our restrictions on the precisions) give a better classification in the windows subgroup. When we use joint conjugate priors there is a bias towards classifying more data into class 1 (building windows float) which disappears when we use independent conjugate priors.

Also we can see that allowing the sizes of the precision matrices of the components in each class to vary gives a much better classification for the smaller classes (tableware, containers and headlamps), regardless of our choice of priors. Again this could be caused by the fact that for this type of modelling, the parameters will describe the training set a lot closer than when we use a common precision matrix for all the components in a class. The classifications obtained using the different models can be seen in figure 4.16.

We obtained a classification using a fixed number of components $k = 3$ using a joint conjugate prior and for the case when we keep all precision matrices for the components in a class the same and when we allow them to vary. For this setup we obtain a worse classification for the windows classes with most of the data from the smaller classes being correctly classified. In the case when we restrict the precisions for the components in a class to be the same ($\Sigma_{ic} = \Lambda_c^{-1}$) the building windows float class gets allocated high number of datapoints from the other two windows classes. This could be happening because class 1 is highly concentrated compared with the other two (windows) classes. That is why most of the points in the area where the building windows float data appears are classified into class 1 as in the model this class has a higher density around that area. As we described in section 4.2 for the Pima Indians data (6 principal components) the use of different sizes for the precision matrices of the components allow the peaks of the predictive densities to be a lot sharper. In this particular dataset this allows a much better discrimination between the different windows classes.

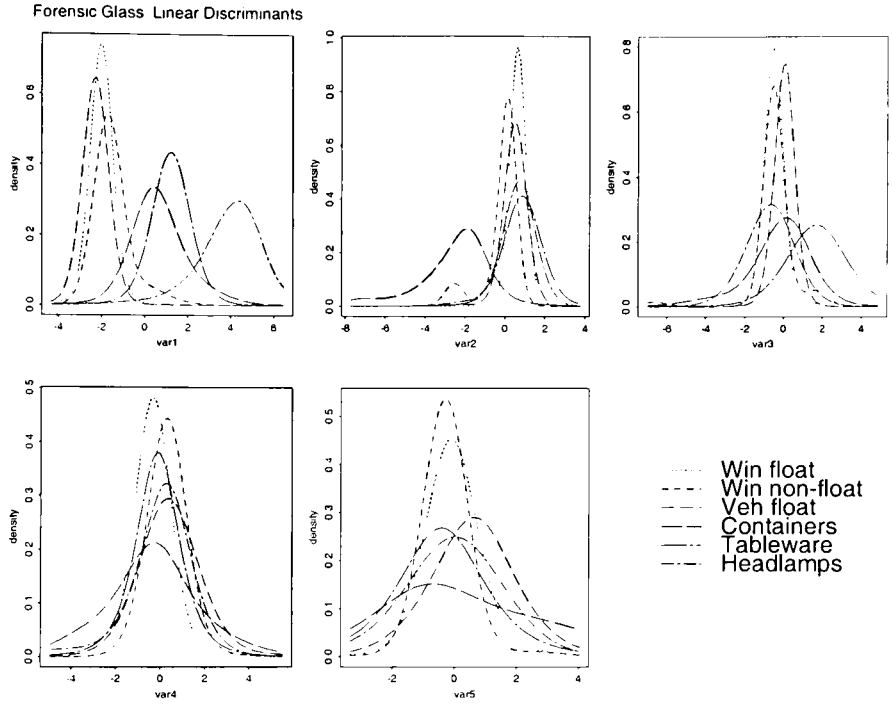


Figure 4.12: Model: common dispersion matrix (for the components in the class) and joint conjugate priors. Marginal Predictive densities for the 5 canonical variates of the forensic glass data (all classes). The whole dataset was used as training set and test set and has 214 datapoints and consists of 70 building windows float, 76 building windows non-float, 17 vehicle windows float, 13 containers, 9 tableware, and 29 headlamps. These marginal predictive densities were obtained for the model with common dispersion matrix ($\Sigma_{ic} = \Lambda_c^{-1}$) and *joint* priors for μ_{ic} and Λ_c. We used a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$.

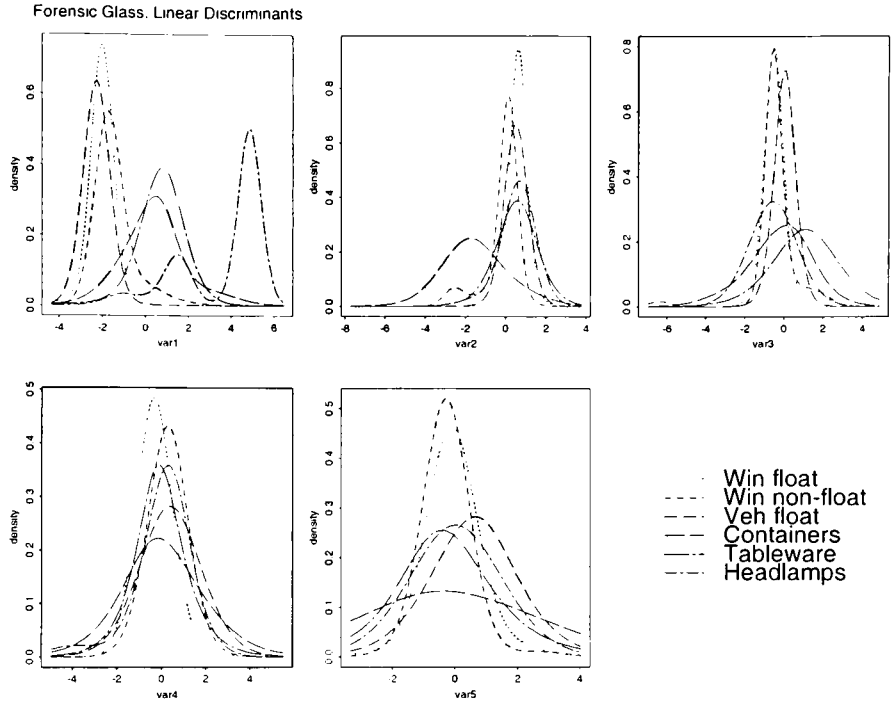


Figure 4.13: Model: common dispersion matrix (for the components in the class) and independent conjugate priors. Marginal Predictive densities for the 5 canonical variates of the forensic glass data (all classes). The whole dataset was used as training set and test set and has 214 datapoints and consists of 70 building windows float, 76 building windows non-float, 17 vehicle windows float, 13 containers, 9 tableware, and 29 headlamps. These marginal predictive densities were obtained for the model with common dispersion matrix ($\Sigma_{ic} = \Lambda_c^{-1}$) and *independent* priors for μ_{ic} and Λ_c. We used a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. Compared to figure 4.12 we can see that for the headlamps class there is clearly two groups in the first canonical variate. This second group appears in figures 4.14 and 4.15 as well.

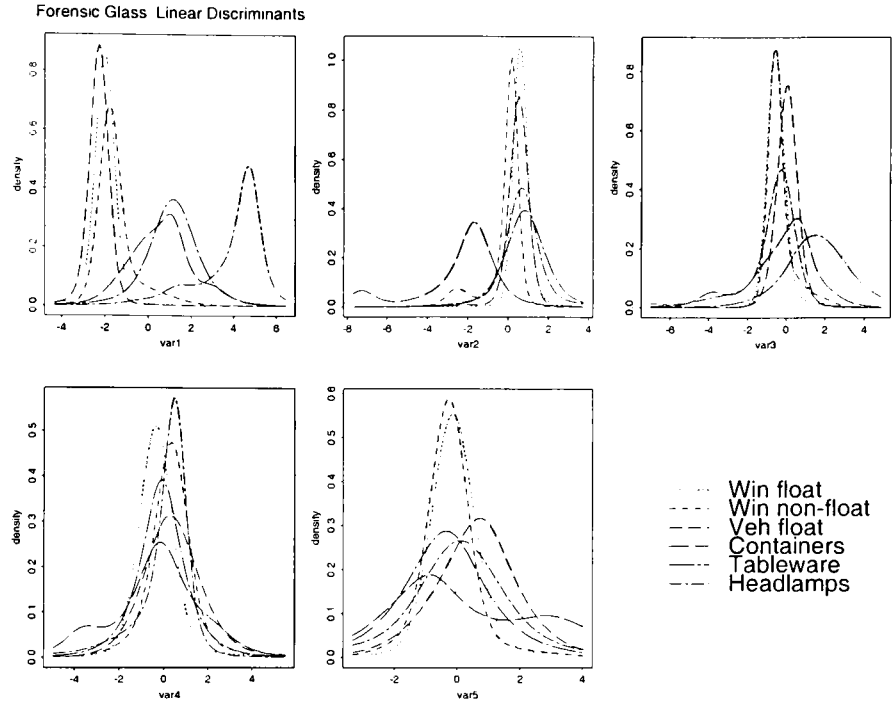


Figure 4.14: Model: dispersions with different sizes (for the components in the class) and joint conjugate priors. Marginal Predictive densities for the 5 canonical variates of the forensic glass data (all classes). The whole dataset was used as training set and test set and has 214 datapoints and consists of 70 building windows float, 76 building windows non-float, 17 vehicle windows float, 13 containers, 9 tableware, and 29 headlamps. These marginal predictive densities were obtained for the model with different dispersion matrices ($\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$) and *joint* priors for μ_{ic} and Λ_c. We used a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. As with figure 4.13 we can see two separate components for the headlamps class in the first canonical variate but because we are using dispersions with different sizes (for the components) the peaks for the distributions appear to be a lot sharper.

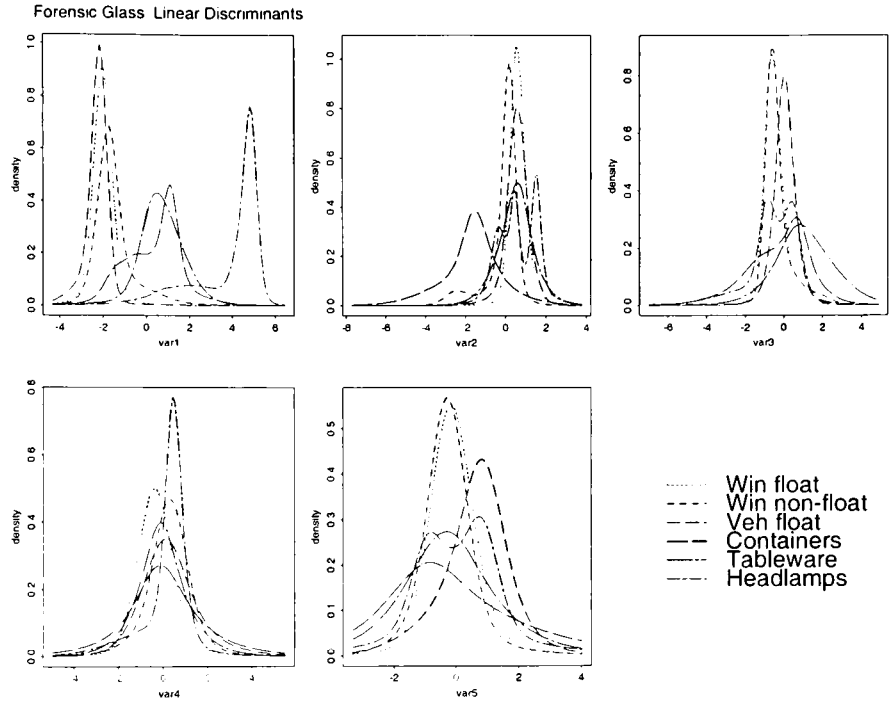


Figure 4.15: Model: dispersions with different sizes (for the components in the class) and independent conjugate priors. Marginal Predictive densities for the 5 canonical variates of the forensic glass data (all classes). The whole dataset was used as training set and test set and has 214 datapoints and consists of 70 building windows float, 76 building windows non-float, 17 vehicle windows float, 13 containers, 9 tableware, and 29 headlamps. These marginal predictive densities were obtained for the model with different dispersion matrices ($\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$) and *independent* priors for μ_{ic} and Λ_c. We used a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. In this figure we can see that the predictive densities obtained have the sharpest and narrowest peaks if compared with figures 4.12, 4.13, and 4.14.

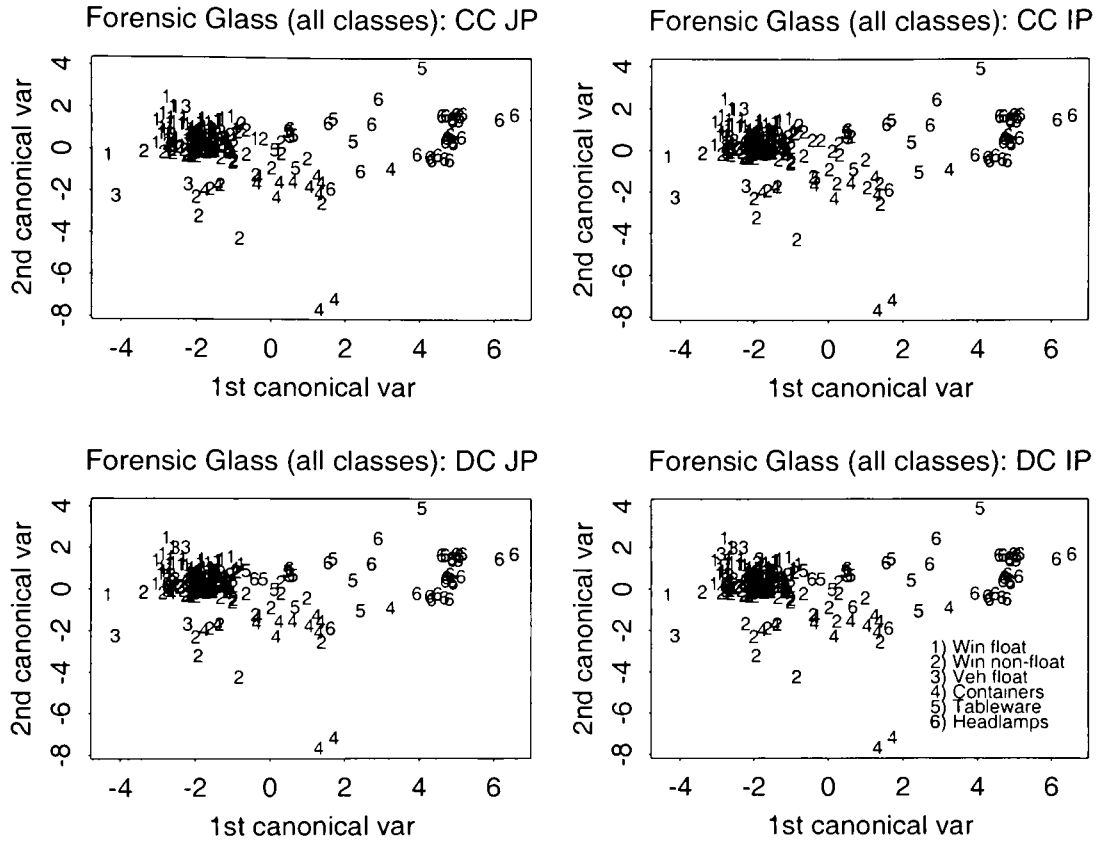


Figure 4.16: Classifications obtained for the forensic glass dataset using all the different models. The classes are 1)building windows float (70), 2)building windows non-float (76), 3)vehicle windows float (17), 4) containers (13), 5) tableware (9), and 6) headlamps (29). The variables used in the analysis are the 5 canonical variates and for the plot we use only the first and second canonical variates. The models used are (from top left) common dispersion with joint priors (CD JP), common dispersion with independent priors (CD IP), different dispersions with joint priors (DD JP), and different dispersions with independent priors (DD IP). The whole dataset was used both as training set and test set.

Table 4.6: Confusion matrices for the forensic glass data. The whole dataset was used as training and test set. The variables used for the classification were the 5 canonical variates. The classifications have been done using fixed $k = 3$. These classifications were obtained using joint conjugate priors for μ_{ic} and Λ_c.

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

	Classification					
true	B. float	B. n-float	V. float	Contain	Tablew	Headlmp
B. float	58	11	1	0	0	0
B. n-float	26	46	1	0	3	0
V. float	7	3	7	0	0	0
V. float	0	1	0	12	0	0
V. float	0	2	0	0	6	1
V. float	1	0	0	0	1	27

214

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

	Classification					
true	B. float	B. n-float	V. float	Contain	Tablew	Headlmp
B. float	51	15	4	0	0	0
B. n-float	15	55	2	1	3	0
V. float	3	2	12	0	0	0
V. float	0	1	0	12	0	0
V. float	0	0	0	0	9	0
V. float	1	1	0	0	1	26

214

For these data the most difficult problem is to discriminate between the windows classes. For this reason we decided to use only these three classes, building windows float, building windows non-float, and vehicle windows float in this second analysis. We use the first 6 principal components which account for 97% of the variance. The different windows classes appear to be mingled very close together in the region between $[-2, 0]$ for the first principal component and $[-2, 2]$ for the second principal component. Figure 4.22 shows the classification for the data in this region.

As when we were modelling for the whole class, we used a birth-and-death/Gibbs algorithm as the number of components in each class is unknown. We used the value of $\lambda_k = 3$ as when we were modelling all classes in the dataset. A initial run can be seen in figure 4.17. As we can see it appears to take a long time for the log-likelihood of the building windows float class to stabilise. We took a burn-in of 600 for the cases when we were modelling with a joint conjugate prior with a total number of iterations (after burn-in) of 2500, nevertheless this was not enough for the cases when we were modelling with independent conjugate priors. For these cases we required a burn-in period of 2000 with a further 2500 iterations used for our predictive densities and our predictive classifications.

In all datasets the use of different sizes to model the data made the model fit the data a lot better. Figures 4.18 and 4.19 are a clear example of this behaviour. Both figures use joint conjugate priors for μ_{ic} and Λ_c; figure 4.18 shows the log-likelihood chains and chains for the number of components for the different classes when a common precision matrix is used for the all the components in a class, while figure figure 4.19 shows the log-likelihood chains and chains for the number of components for the different classes when different sizes for the precision matrices of the components are used. The log-likelihood chains for the classes are much larger for the cases when the sizes are allowed to vary than when a common precision matrix is used.

We obtained the joint predictive densities of the first two principal components for each class. These can be seen in figure 4.20. As we can see for the models with independent conjugate priors tend to separate the components more clearly than when using joint conjugate priors. In the case of class 2 (building windows non-float) the two subgroups that are obtained far away from the main bulk of the class with the model that uses common precision for the classes with joint priors are modelled as just one very flat subgroup as can be seen in figure 4.21. What this allows is a much higher peak for the other subgroup in this class which means that a better discrimination is obtained between class 1 and class 2.

The classification for the building float, building non–float, and vehicle float can be seen in table 4.7. For the model with common precision matrix for the components in a class and joint conjugate prior for μ_{ic} and Λ_i most of the data were classified into the building float class (class 1) regardless that the proportions for class 1 and class 2 are roughly the same. This is due to the fact that the density obtained for class 2 (building non-float)

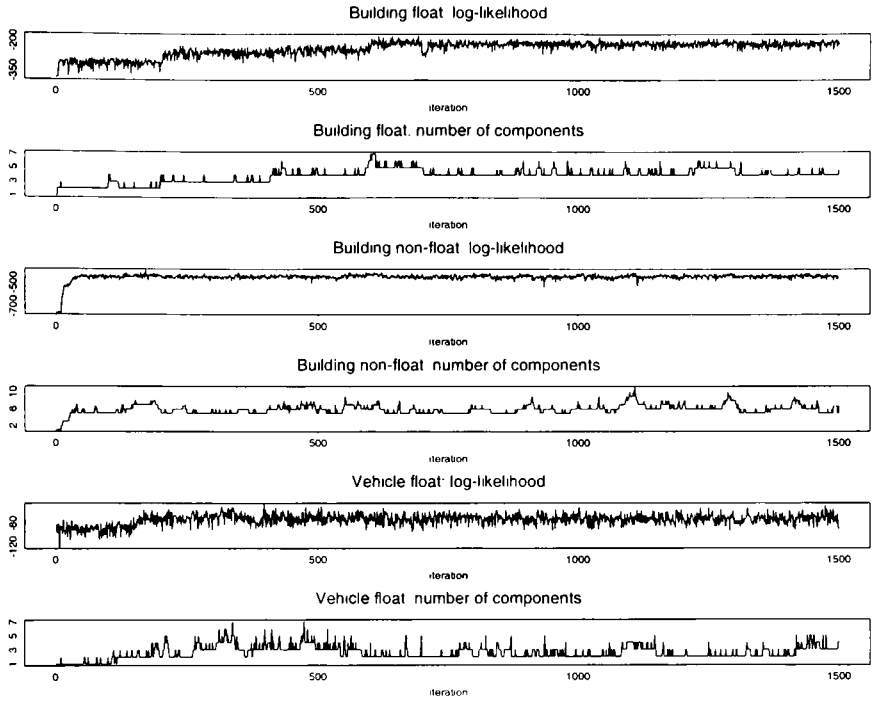


Figure 4.17: Log-likelihood chains and number of components (k_c) obtained for the forensic glass data (windows). The windows classes (building windows float 70, building windows non-float 76, and vehicle windows float 17) were used both as training set and test set (total of 163 datapoints). The variables used in the analysis were the first 6 principal components (obtained using S-plus command princomp). These log-likelihood chains were obtained using a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$. For these data class 1) buildings windows float take longer to stabilize than the other two.

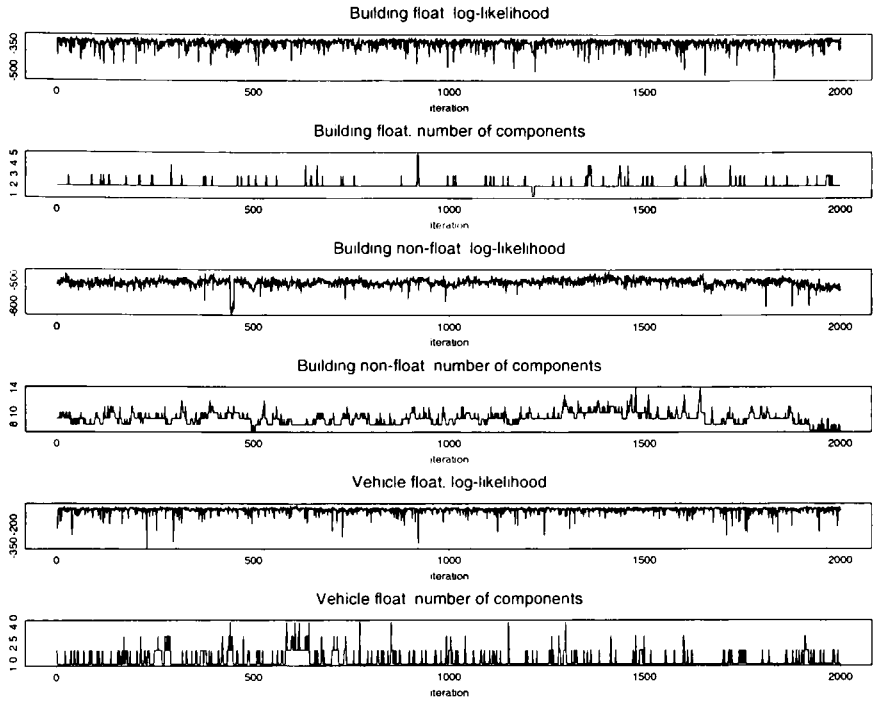


Figure 4.18: Log-likelihood chains and number of component (k_c) chains obtained for the forensic glass data (windows). The windows classes (building windows float 70, building windows non-float 76, and vehicle windows float 17) were used both as training set and test set (total of 163 datapoints). The variables used in the analysis were the first 6 principal components (obtained using S-plus command princomp). These log-likelihood chains were obtained using a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. The model used in this case use a common precision matrix for all components in a class to $\Sigma_{ic} = \Lambda_c^{-1}$ and use a joint prior for μ_{ic} and Λ_c. These chains were obtained after a burn-in period of 600 so the total number of runs was 2600.

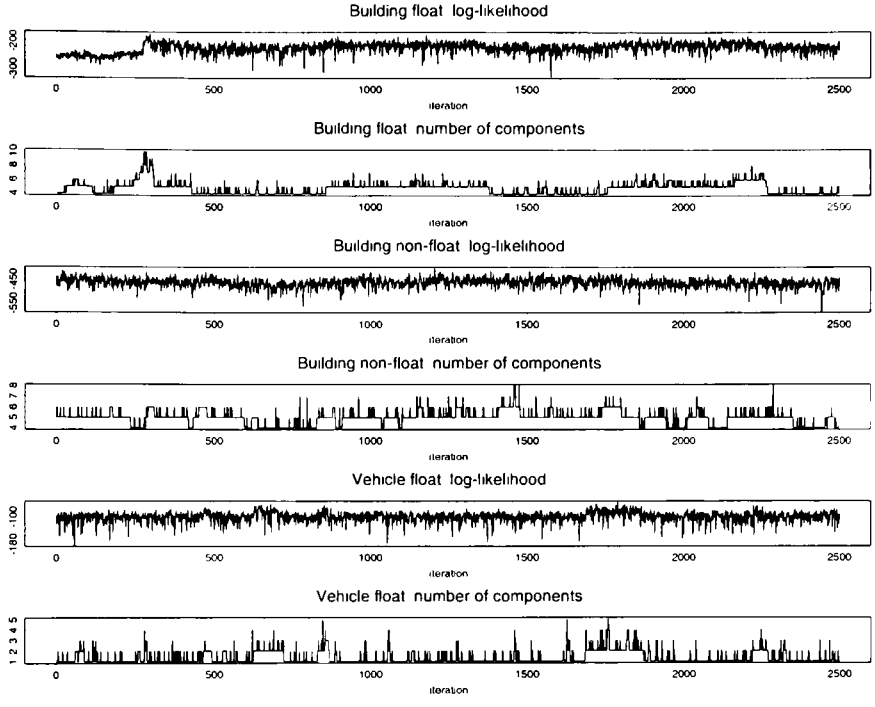


Figure 4.19: Log-likelihood chains and number of component (k_c) chains obtained for the forensic glass data (windows). The windows classes (building windows float 70, building windows non-float 76, and vehicle windows float 17) were used both as training set and test set (total of 163 datapoints). The variables used in the analysis were the first 6 principal components (obtained using S-plus command princomp). These log-likelihood chains were obtained using a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. The model used in this case allowed the sizes of the precision matrices for the components in a class to vary $\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ and use a joint prior for μ_{ic} and Λ_c. These chains were obtained after a burn-in period of 600 so the total number of runs was 3100.

for this type of model is a lot more dispersed than for class 1. As this happens, all the datapoints from class 2 that are in the same region as the data from class 1 are classified as building float while most of the data outside this region will be classified as buildings non–float. Nevertheless the model with variable sizes and independent conjugate priors give an excellent classification to the data (although it is biased). This is mainly due to the fact that this type of model allows some subclasses to be very concentrated while others fairly dispersed within the same class.

All models produce a good classification outside the region $[-2.0]$ for the first principal component and $[-2, 2]$ for the second principal component, even for the model with common precision for the components in the class and joint conjugate priors for μ_{ic} and Λ_c. In figure 4.22 we can see the original classification and the classifications obtained using the different models inside the mentioned region. These plots show how the different models are picking up certain regions in the data more sharply than others.

Just as we saw when we used the canonical variates and we were modelling all the classes, when we fix the number of components to $k_c = 3$ the classifications we obtained are in general much worse than using a variable k_c. Table 4.8 show these classifications using joint conjugate priors for μ_{ic} and Λ_c for the model with common precision for all components in a class and for the model with precisions of different sizes. In this case the classification obtained with a common precision is markedly worse than any other obtained previously. For this model, class 1 is allocated most of the data in the region $[-2, 0]$ for the first principal component and $[-2, 2]$ for the second. A much better classification is obtained using precisions with different sizes as we can see in table 4.8.

Table 4.7: Confusion matrices for the forensic glass data (windows) using different models. The windows classes of the data were used as training and test set. From a total of 163 cases 70 are buildings float, 76 buildings non–float, and 17 vehicle float. The variables used for the classification were the first 6 principal components which account for 97% of the variation. The classifications have been done using variable k with $\lambda_k = 3$. The letters J.P. and I.P. stand for joint conjugate priors and independent conjugate priors for μ_{ic} and Λ_c.

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

	Classification		
true	Buildings float	Buildings non-float	Vehicle float
Buildings float	62	7	1
Buildings non-float	34	42	0
Vehicle float	9	3	5

163

$\Sigma_{ic} = \Lambda_c^{-1}$ I.P.

	Classification		
true	Buildings float	Buildings non-float	Vehicle float
Buildings float	67	3	0
Buildings non-float	23	53	0
Vehicle float	11	3	3

163

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

	Classification		
true	Buildings float	Buildings non-float	Vehicle float
Buildings float	64	5	1
Buildings non-float	8	68	0
Vehicle float	8	4	5

163

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ I.P.

	Classification		
true	Buildings float	Buildings non-float	Vehicle float
Buildings float	63	6	1
Buildings non-float	7	68	1
Vehicle float	6	3	8

163

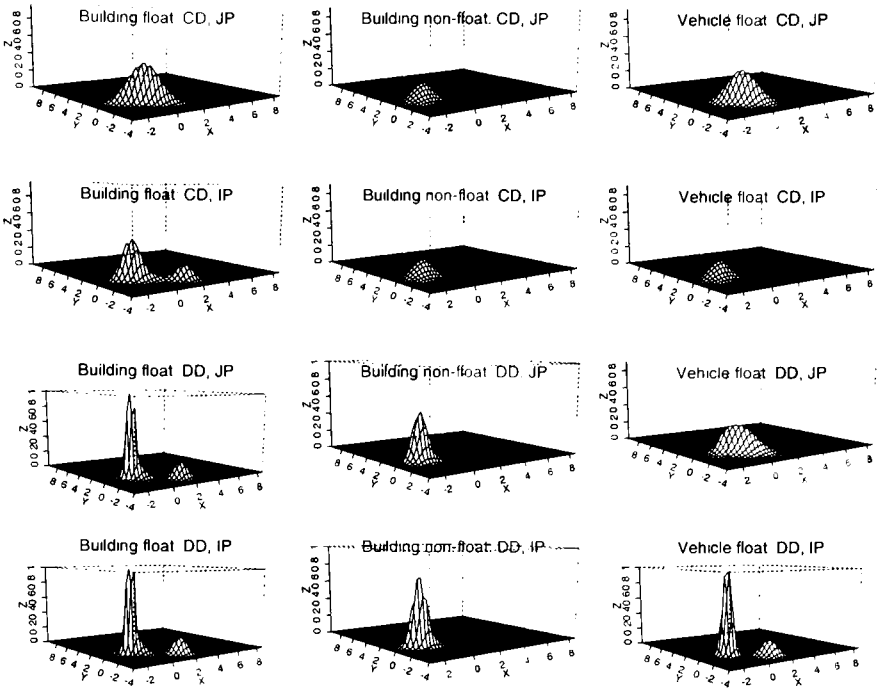


Figure 4.20: Joint Predictive densities obtained for the forensic glass data (windows). The windows classes (building windows float 70, building windows non-float 76, and vehicle windows float 17) were used both as training set and test set (total of 163 datapoints). The variables used in the analysis were the first 6 principal components. These joint predictive densities where obtained just for the first and second principal component as most of the information for discrimination between the three classes lie in these variables. The far left column show the models obtained for the building windows float, the column in the centre show the models obtained for the building windows non-float class, and the far right column show the ones obtained for the vehicle windows float class. From the top, the models used are common dispersion with joint priors (CD JP), common dispersion with independent priors (CD IP), different dispersions with joint priors (DD JP), and different dispersions with independent priors (DD IP). All plots use the same point of view to allow a better comparison between them. As in the crabs data we can see that some models show a clearer separation between components in a class than others. For this dataset a better separation between components in a specific class will allow a better classification.

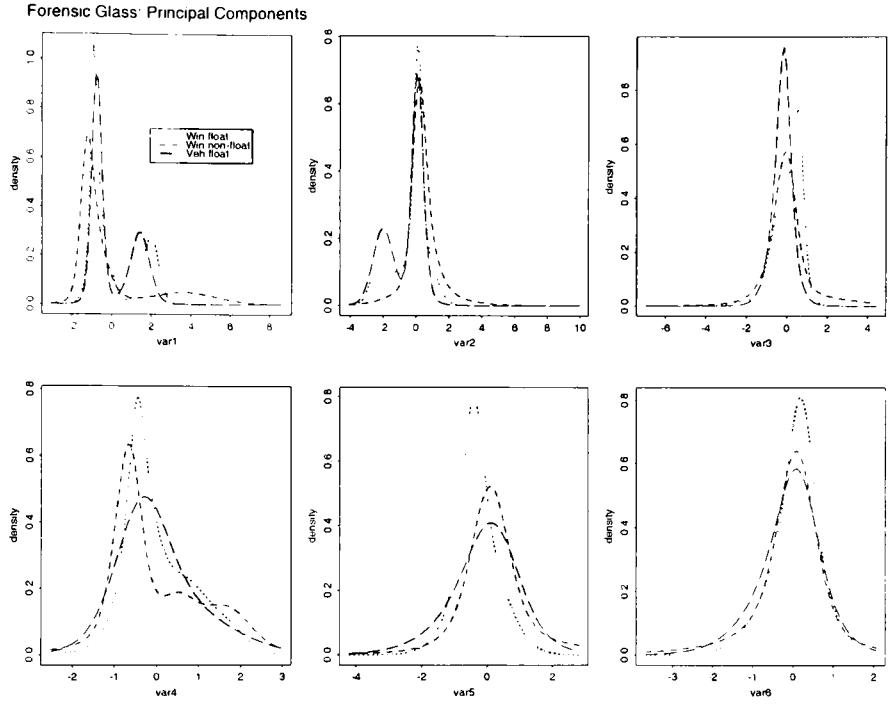


Figure 4.21: Marginal Predictive densities for the 6 principal components of the forensic glass data (windows only). The windows classes (building windows float 70, building windows non-float 76, and vehicle windows float 17) were used both as training set and test set (total of 163 datapoints). These marginal predictive densities were obtained for the model with different dispersion matrices ($\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$) and *independent* priors for μ_{ic} and Λ_c. We used a variable number of component for each class k_c modelled with a Poisson prior with parameter $\lambda_k = 3$. As we can see in the top right figure, the marginal predictive density for the first principal component in the building windows float class shows a sharp peak just at the left of the other two classes while a second mode lies to the far right and is very flat.

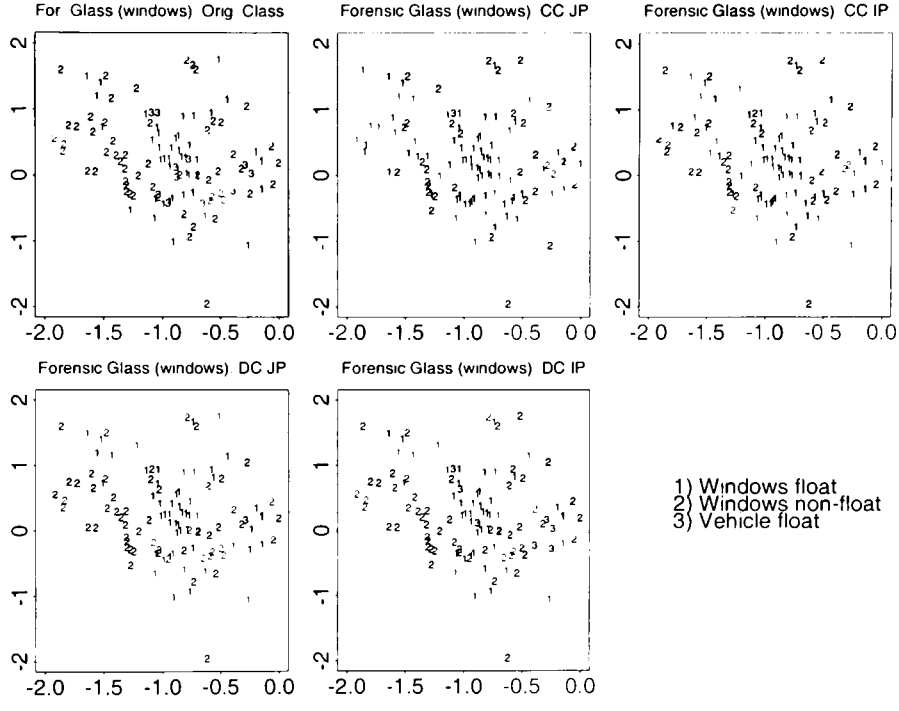


Figure 4.22: Classification obtained for the windows in the forensic glass dataset using all the different models. The classes used are buildings windows float (class 1), buildings windows non–float (class 2), and vehicle windows float (class 3). Only the datapoints within $[-2, 0]$ for the first principal component and within $[-2, 2]$ for the second principal component are plotted. This means a total of 46 datapoints for class 1, 56 for class 2, and 11 for class 3. The variables used in the analysis are the first 6 principal components and the figures show the first two. The same set was used as training set and test set.

Table 4.8: Confusion matrices for the forensic glass data using different models. The whole dataset was used as training and test set. From a total of 163 cases 70 are buildings float, 76 buildings non–float, and 17 vehicle float. The variables used for the classification were the first 6 principal components which account for 97% of the variation. The classifications have been done using variable k with $\lambda_k = 3$. These classifications were obtained using joint conjugate priors for μ_{ic} and Λ_i (J.P.).

$\Sigma_{ic} = \Lambda_c^{-1}$ J.P.

	Classification		
true	Buildings float	Buildings non-float	Vehicle float
Buildings float	69	0	1
Buildings non-float	47	29	0
Vehicle float	9	1	7

163

$\Sigma_{ic} = (\zeta_{ic}\Lambda_c)^{-1}$ J.P.

	Classification		
true	Buildings float	Buildings non-float	Vehicle float
Buildings float	51	16	3
Buildings non-float	6	66	4
Vehicle float	5	4	8

163

Chapter 5

Conclusions and Further Work

Modelling classes with mixtures of distributions for classification purposes allows great flexibility. One of the main problems is that this type of models by their nature have a very complicated likelihood surface. This means that commonly used techniques that obtain estimates for the parameters and use these estimates to obtain a classification could give seriously misleading results. The problem with these methods is that they do not take account of the "uncertainty" of these parameters and in some cases the estimates obtained could be local maxima where the EM-algorithm could get trapped or in the case of using MCMC methods it would be necessary to prevent the sampler from jumping from one mode to another or study the sampling chain and determine each time these jumps occur and take them into consideration to create our estimates.

It is possible that because of these reasons mixtures of distributions have not been used more often for classification or discrimination purposes. By obtaining a predictive classifier we not only take account of the "uncertainty" of the parameters but we also eliminate the problem of identifiability (label switching, section 1.3.3) as at no point an estimate for a specific parameter is required. Another advantage is that using a fully Bayesian approach the number of components used to describe a class is not requiered to be known in advance. Most previous work in classification that uses mixtures to model a class assumes that the right number of components in a class is previously available (for example Hastie and Tibshirani (1996)). The use of a predictive classifier again allowed us the use of a variable number of components in a class k_c as the selection of a fixed k_c is a problem which still has not been solved adequately.

In the case when inference on the number of subgroups in a class is our main interest we give a possible parameterisation that may allow some inference to be made (section 3.5.1). The use of a common dispersion matrix for the number of components with joint conjugate priors for μ_{ic} and Λ_c allow us to obtain some information of the total number

of components required. Neverthless inference on the number of components in a class is heavily influenced by our choice of prior; if the number of components used is less than the true number of components classification would not be adequate, and in the case when the number of components is too large then the procedure would be highly inefficient. Our choice of a Poisson prior with parameter λ_k gave good results, the birth-and-death/Gibbs algorithm used appeared to be flexible enough to obtain an adequate number of components and create a sampling chain from them. Nevertheless more work in the area is required to compare different priors for the number of components.

For this thesis the components in a mixture in general do not have a physical interpretation and the mixture setup is used just as a flexible tool to model the data. We have shown that this type of modelling using variable number of components in a class k_c produces a good classification. We have compared this looking at the density estimation obtained for two simulated samples and at the crabs data for which the number of components in a class k_c was known. We also used a variable k_c for the Pima Indians data and the forensic glass data and in all cases these models gave at least as good a classification as using fixed k_c while in most cases these classifications were better.

In this thesis we have used two types of models for the dispersion matrices of the components. First we restricted the covariance matrices of all the components in a class to be the same ($\Sigma_{ic} = \Lambda_c$) and then reparametrised the covariance matrix into size (ζ_{ic}), orientation and shape (Λ_c) and allowed the sizes to vary while keeping a common orientation and shape for all the components in a class. This gave further flexibility to our modelling as was seen in section 4.3. Not only that but also using this parameterisation jointly with a variable number of components the classifications obtained were in most of the cases much better than for the rest of our models.

In the case of the forensic glass data although we use all the data as training set and test set, we believe that modelling with variable sizes ζ_{ic} for the component precisions with independent conjugate priors for μ_{ic} and Λ_c allows the components in a mixture to have the right location as well as the right shape, size and orientation to give a better classification than for the rest of the models. For example in the forensic glass data for the building windows non-float class most of the cases far away from the bulk of the data are modelled using one component with large dispersion as we can see in figure 4.21 while another component with much smaller dispersion is used to model the datapoints for this class that are close to the bulk of the data. This in return gave a much better classification as was seen in table 4.7.

For most of the data the classifications obtained by using independent conjugate priors for μ_{ic} and Λ_c are better than in the case of using a joint conjugate prior. This better classification can be explained by the fact that the components that are born (in the birth-and-death part of the algorithm) are allowed to model the data without restrictions on the precision matrix currently being used. This allows the components in a class to be better

separated from each other which in turn gives the class a much better definition of where it starts and where it finishes inside the dataspace.

5.1 Further Work

In this report we have only studied the problem when part of the data is fully classified and the classes are modelled separately. Future work could involve the study of a predictive classification when the classes are not modelled separately. So far we have discussed priors on the basis of using the same hyperpriors for the whole data (regardless of class). This is done with the idea that the parameters in the model roughly correspond to the same family as the classes (and subclasses) in the data will have several things in common. For example in the case of the Pima Indians data, they all will come from the same population that will have some differences but if compared to another population probably will have more things in common than not. So far we have not used priors for these hyperparameters. If a hierarchical model was used these hyperparameters could then be updated every now and again (not necessarily at each iteration). We believe that this would help the mixing and also possibly the birth and death process.

We could use other types of hierarchical modelling on the data. For example we could assume that there is a common structure on the precisions on all the subclasses in the data (regardless to which class they belong to) and either use a common shape and orientation for hierarchical modelling on the hyperparameter Ω_0 which can be updated once after all the precisions for all the classes have been updated.

The use of a birth-and-death means that we look at this continuous process at certain points in time. At the moment this is done at a constant rate, but it should be possible to decide to vary the intervals to allow the process to stabilise (longer intervals) and once we have defined it has look at the birth-and-death process more often (shorter intervals) thus making the sampling process more efficient. This would go hand in hand with the selection of a burn-in period for the chain.

Other sampling schemes could be employed. For example when the classes are not fully known or when not all the classes are known. This could lead to the use of sequential classification: obtaining the classification for a new datapoint and then using this datapoint to obtain information for the class it was classified to. This way if a class is not represented in the training set, we can still obtain datapoints classified to this class if there is strong evidence they do not belong to any of the other previously used classes. For this our predictive estimate would be obtained as in section 2.2) with

$$\widehat{C}(x) = \begin{cases} l, & \text{if } p(l \mid x) = \max_c \widetilde{p}(c \mid x) \text{ and this exceeds } 1-d, \\ C_{new}, & \text{if each } \widetilde{p}(c \mid x) \leq 1-d, \end{cases}$$

where C_{new} would be a new class, in this case $d < 1$.

We have assesed convergence of our sampling chains by comparing the marginal predictive densities obtained from 4 different starting points. We are only comparing different runs and this does not mean that we are actually any closer from the target distribution but just that the sampling chains obtained are not far apart from each other. Further analysis of other convergence criteria specifically under mixture models is required as they present typical problems that affect convergence and convergence diagnostics.

Another area of study could involve the use of multivariate Student-t-distribution instead of the multivariate normals. These models can prove to be more robust specially in the cases where outliers are present.

Appendix A

Description of datasets used

Simulated mixture of three normals with common covariance

In this thesis we use two simulated datasets and three real datasets. The two simulated datasets were created to analyse and test the methods described in chapter 3 for modelling a single class as a mixture of multivariate normal distributions. The real datasets were used to apply these methods to classification. Our first simulated dataset was obtained from a bivariate distribution with independent variables, one a normal mixture (3 components, means $-4, 0, 4$ and unit variance) and the other a standard normal. These data were used to compare how well the model with a common covariance worked while using different priors as well as assesing the sampling chain obtained from the birth-and-death algorithm. We can see a plot of the data in figure A.1 and the empirical distribution in figure A.2.

Simulated mixture of two normals with different covariances

The second simulated data are a mixture of two bivariate normal distributions, the first one with mean at $(0, 0)$ and dispersion matrix $\Sigma = [1, .2, .2, .5]$; the second a multivariate normal with mean at $(10, 10)$ and dispersion matrix 9Σ. This second dataset was obtained to assess our model when we allow the sizes of the dispersion matrices for the components to be different while keeping a common shape and orientation for all components in a class. The plot of these data can be seen in figure A.3 and the empirical distribution can be seen in figure A.4.

Crab data

The three real datasets that we use are the *Leptograpsus* crabs (we refer to it as the Crab data), the Pima Indians data, and the forensic glass data. The data in the *Leptograpsus* crabs were grouped as one species *L. variegatus* and now has been split into two depending on colour, blue or orange. These data were first reported by Campbell and Mahon

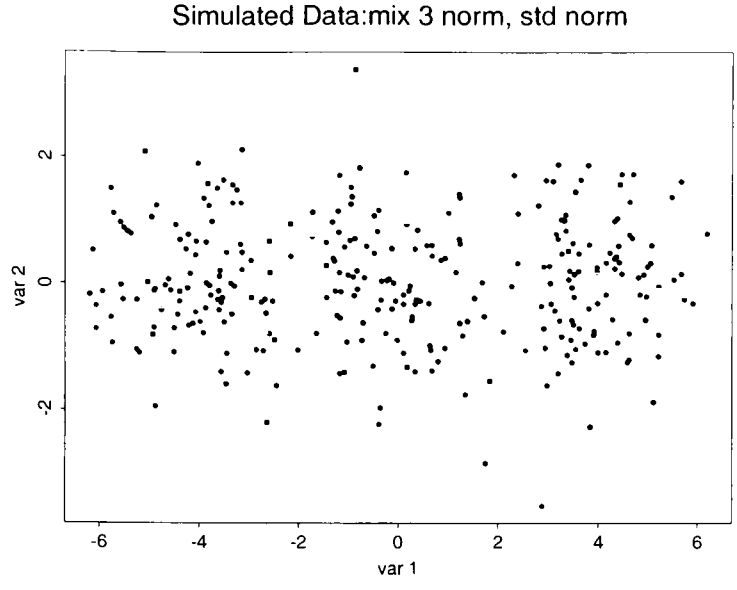


Figure A.1: Simulated data, variable 1 is a mixture of three normal distributions with means $(-4.0, 4)$ and unit variance, while variable 2 is a standard normal distribution. The data consist of 300 observations.

(1974) and consist of 200 crabs (100 of each species, half female and half male) and the variables measured are the carapace length (CL) and width (CW), the size of the frontal lobe (FL) and rear width (RW), and the body depth (BD). The data were divided into a training set of only 40 crabs, 20 from each class (10 from each sex) and a test set of 160 datapoints. We transformed the data to remove the dominant effect of size by dividing all measurements by the square root of the area. The carapace area was calculated as the product of the length times the width of the carapace. We also substracted the mean for each sex and then used the first three principal components in our analysis which account for 99% of the variablity in the data. The two classes (Blue and Orange) are well separated and a good classification will be expected from using this dataset. Figure A.5 shows the training set for the two principal components.

Pima Indians data

The second real dataset that we use is the Pima Indians data which consist of 532 complete records of indian women (all at least 21 years old) who were tested for diabetes according to criteria by the World Health Organization and approximately one third of them are diabetics. These data are available from Murphy and Aha (1995) and have 8 reported variables: number of pregnancies, plasma glucose concentration in an oral glucose tolerance test (we refer to it only as glucose), diastolic blood pressure (mm Hg), triceps skin fold

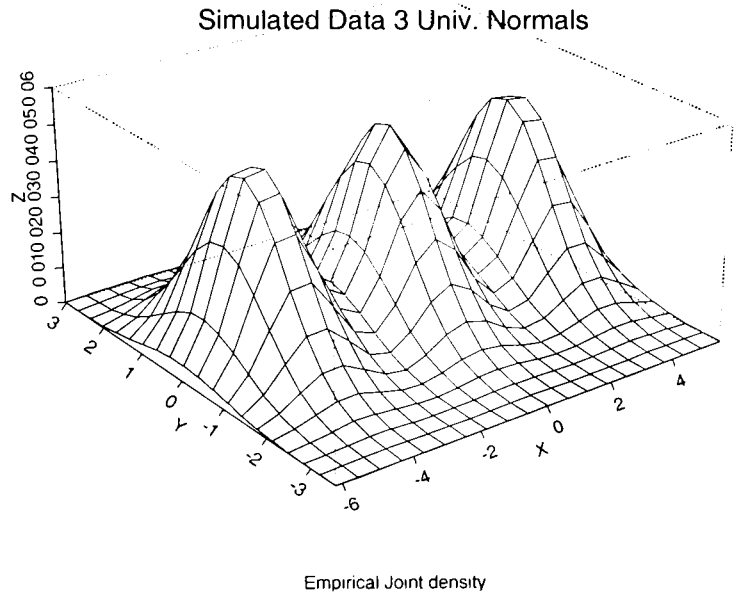


Figure A.2: Empirical density for the simulated data of 3 normal distributions with equal proportions. The variables are independent with variable 1 a mixture of three normal distributions with means $(-4, 0, 4)$ and unit variance, and variable 2 a standard normal distribution.

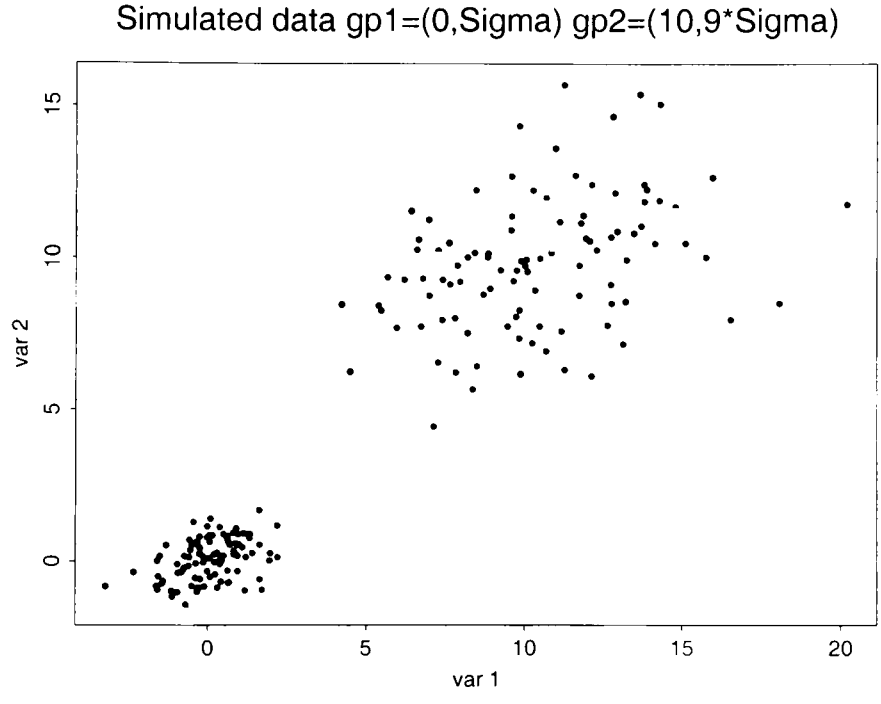


Figure A.3: Plot of simulated data of two multivariate normals with dispersion matrices of different sizes. The first component is a multivariate normal with mean at $(0, 0)$ and dispersion matrix $[1, 0.2, 0.2, 0.5]$. The second component is another multivariate normal with mean $(10, 10)$ and dispersion matrix $[9, 1.8, 1.8, 4.5]$.

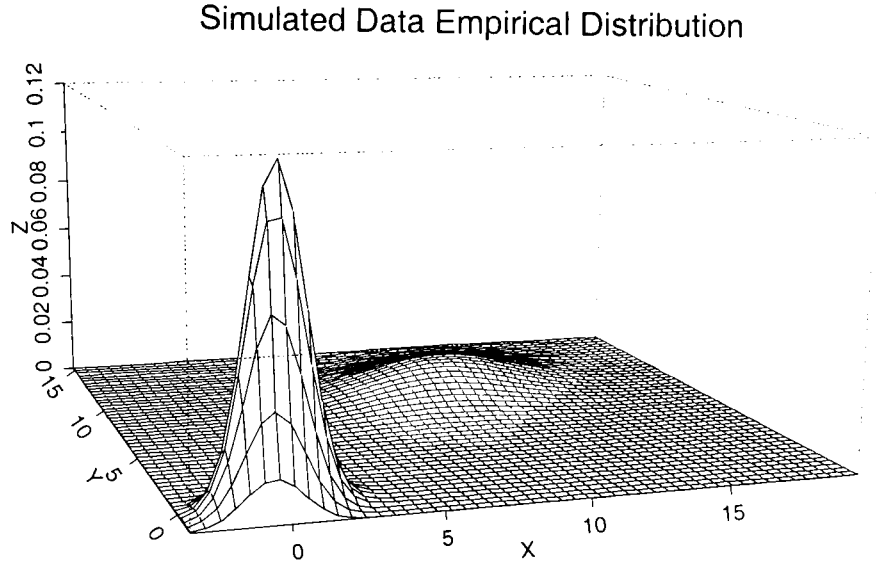


Figure A.4: Empirical distribution for the simulated data of two multivariate normals with dispersion matrices of different sizes. The first component is a multivariate normal with mean at $(0, 0)$ and dispersion matrix $[1, 0.2, 0.2, 0.5]$. The second component is another multivariate normal with mean $(10, 10)$ and dispersion matrix $[9, 1.8, 1.8. 4.5]$.

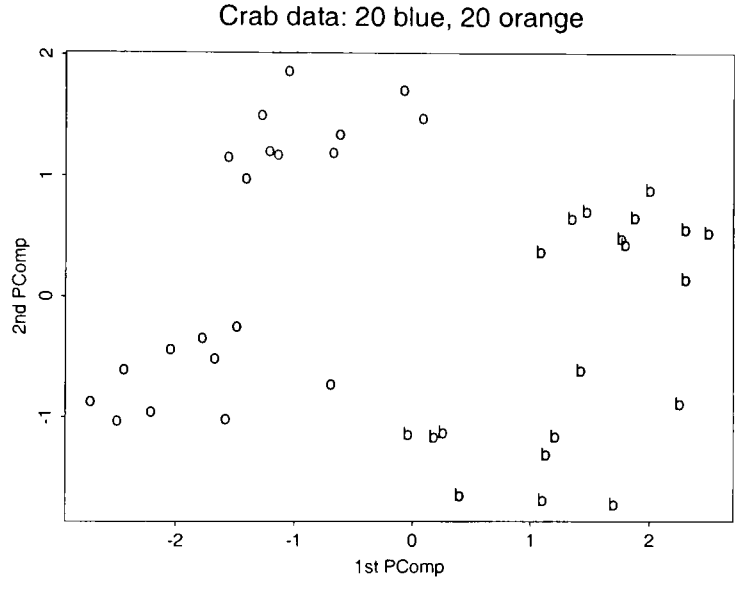


Figure A.5: Crab data (training set), first two principal components. The training set consist only of 20 datapoints from each class (10 male and 10 female). The 40 datapoints are labelled 'b' for Blue species and 'o' for the Orange one.

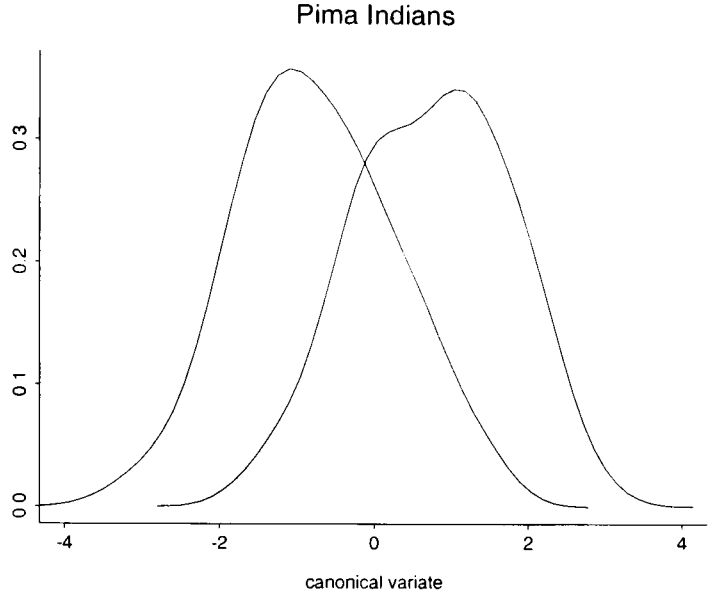


Figure A.6: Density estimates of the non-diabetics (left) and diabetics (right) on the canonical variate for the Pima Indians data. The data consist of 68 diabetics and 132 non-diabetics. Kernel methods with automatically chosen bandwidths were used to obtain these density estimates.

thickness (mm), serum insulin (μU/ml), body mass index (weight in kg/(height in m)^2), diabetics pedigree function, and age in years. The variable serum insulin presented a lot of missing data so it was not used in the analysis leaving only the other 7 variables. Ripley (1996) describes this dataset and uses 200 (68 diabetics, 132 non-diabetics) as training set and the 332 remaining as a test set. The lowest number of misclassification obtained by Ripley used LDA (66 misclassified). In figure A.6 we can see the LDF for the diabetics and non-diabetics groups. The density estimates in figure A.6 were obtained using Kernel methods whith automatically chosen bandwidths (using function bandwidth.nrd in Venables and Ripley (1997)). A similar figure can be found in Ripley (1996) (page 99). For this dataset we use the linear discriminant function as one approach, the two continuous variables glucose and body mass index for a second approach and the first 6 principal components (95% of the variance) for our third approach.

Forensic glass data

The last dataset studied is the forensic glass (also found in Murphy and Aha (1995)). These data consist of 214 cases of 6 different groups in the data (7 originally but one reports no data). The classes are: building windows float (70), building windows non-float (76), vehicle windows float (17), containers (13), tableware (9), and headlamps (29).

There are 9 different variables: refractive index, sodium, magnesium, aluminium, silica, potassium, calcium, barium, and iron. In Ripley (1996) because the number of datapoints is very small cross-validation is used in all analyses. Due to the nature of the MCMC method used this is highly impractical so we decided to use the whole dataset as training set and test set to have a general idea of how well this method works (the classification obtained will be biased). A distance method to represent the data was used to create figure A.7 (Sammon mapping (Sammon 1969)). As we can see the windows classes are very close together compared to the other three classes. In particular the headlamps class is well separated from the rest of the classes. The classes in these data appear to have subclasses as we can see in figure A.8 because of this, the use of mixtures to model each class might prove adequate. In this thesis we use the 5 canonical variates to obtain a classification for the dataset. As our second approach to the data we analyse only the first three groups (windows) which as we mention are very similar. In this second approach we use the first 6 principal components which account for 97% of the variance.

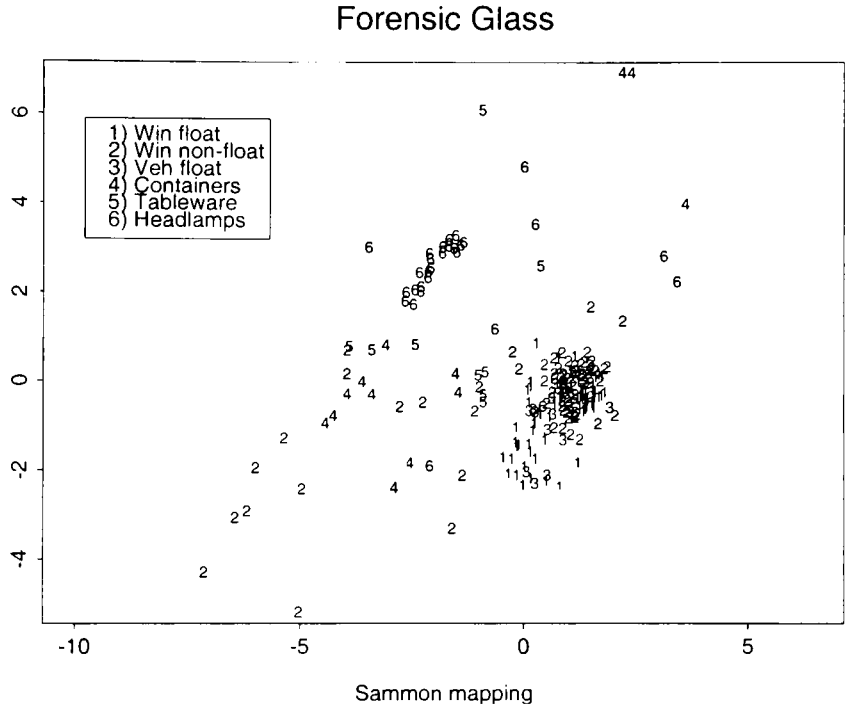


Figure A.7: Sammon mapping for the forensic glass dataset. The classes used are buildings float (class 1), buildings non–float (class 2), vehicle float (class 3), containers (13), tableware (9), and headlamps (29). The Splus commands dist and sammon were used to obtain the distance matrix and the scaling in two dimensions respectively. The data plotted consist of 213 points as case 40 was dropped because it is a duplicate of case 39 (both cases are from the buildings float class).

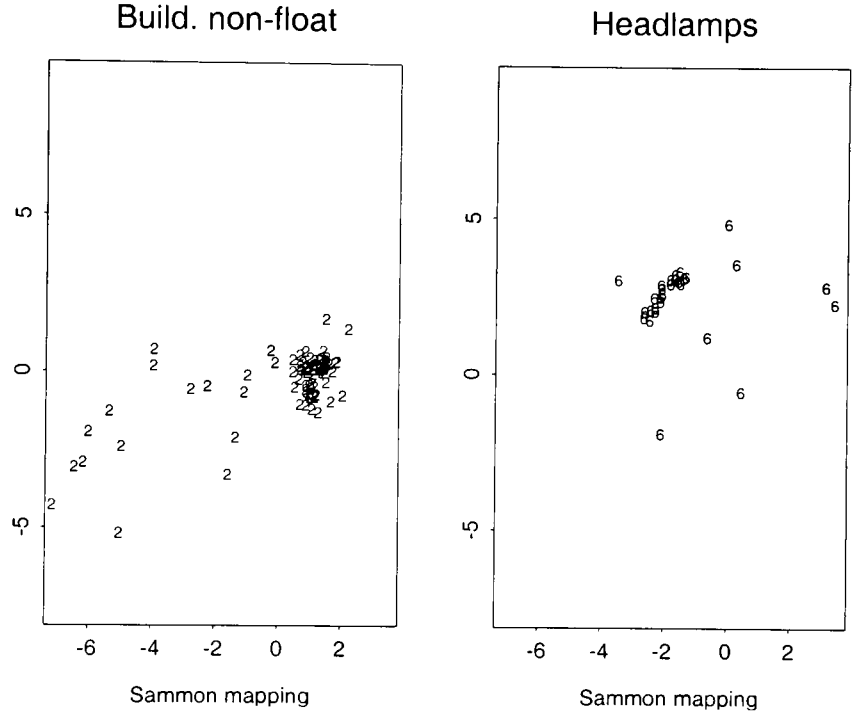


Figure A.8: Sammon mapping for classes 2 (buildings windows non-float) and 6 (headlamps) in the forensic glass dataset. The Splus commands dist and sammon were used to obtain the distance matrix and the scaling in two dimensions respectively. The buildings windows non-float class contains 76 cases while the headlamps class has 29.

Appendix B

Gibbs sampling μ_i and Λ

Focus only on a single class c. The joint prior for $\theta = (\mu_1, \ldots, \mu_k, \Lambda)$ given hyperparameters ξ, τ, and Ω

$$\begin{aligned} p(\theta) &= p(\mu_1 \mid \Lambda)p(\mu_2 \mid \Lambda)\ldots p(\mu_k \mid \Lambda)p(\Lambda) \\ &\propto |\Lambda|^{(\nu-p)/2-1} \exp\left(-\frac{1}{2}\mathrm{tr}\,(\Omega\Lambda) - \frac{1}{2\tau}\sum_{i=1}^{k}(\mu_i - \xi)^T\Lambda(\mu_i - \xi)\right) \end{aligned}$$

Using the likelihood described in section 2.2.2 the likelihood (for class c) is

$$p(x \mid \theta) \propto |\Lambda|^{n/2} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Lambda\sum_{i=1}^{k} M_i\right)\right),$$

with

$$M_i = \sum_{j:z_j=i}^{n_i}(x_j - \mu_i)(x_j - \mu_i)^T,$$

where n_i equal to the number of datapoints allocated to component i.

The posterior distribution is then

$$p(\theta)p(x \mid \theta) \propto |\Lambda|^{(n+\nu-p)/2-1} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\Lambda\left(\Omega + \sum_{i=1}^{k} M_i + \frac{1}{\tau}\sum_{i=1}^{k}(\mu_i - \xi)(\mu_i - \xi)^T\right)\right)\right)$$

Taking the last two components of the sum in the exponent

$$\begin{aligned}
A &= \sum_{i=1}^{k} M_i + \frac{1}{\tau}\sum_{i=1}^{k}(\mu_i - \xi)(\mu_i - \xi)^T \\
&= \sum_{i=1}^{k}\sum_{j:z_j=i}^{n_i}(x_j - \mu_i)(x_j - \mu_i)^T + \frac{1}{\tau}\sum_{i=1}^{k}(\mu_i - \xi)(\mu_i - \xi)^T \\
&= \sum_{i=1}^{k}\sum_{j:z_j=i}^{n_i}(x_j - \bar{x}_i)(x_j - \bar{x}_i)^T + \sum_{i=1}^{k}\sum_{j:z_j=i}^{n_i}(\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T \\
&+ \frac{1}{\tau}\sum_{i=1}^{k}(\mu_i - \xi)(\mu_i - \xi)^T.
\end{aligned}$$

Taking the last two components of this sum

$$\begin{aligned}
A2 &= \sum_{i=1}^{k}\sum_{j:z_j=i}^{n_i}(\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T + \frac{1}{\tau}\sum_{i=1}^{k}(\mu_i - \xi)(\mu_i - \xi)^T \\
&= \sum_{i=1}^{k} n_i(\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T + \frac{1}{\tau}\sum_{i=1}^{k}(\mu_i - \xi)(\mu_i - \xi)^T \\
&= \sum_{i=1}^{k} n_i(\bar{x}_i - \mu_i)(\bar{x}_i - \mu_i)^T + \frac{1}{\tau}(\mu_i - \xi)(\mu_i - \xi)^T \\
&= \sum_{i=1}^{k}(n_i + \frac{1}{\tau})\left(\mu_i - \frac{\frac{1}{\tau}}{\frac{1}{\tau} + n_i}\xi - \frac{n_i}{\frac{1}{\tau} + n_i}\bar{x}_i\right)\left(\mu_i - \frac{\frac{1}{\tau}}{\frac{1}{\tau} + n_i}\xi - \frac{n_i}{\frac{1}{\tau} + n_i}\bar{x}_i\right)^T \\
&+ \sum_{i=1}^{k}\frac{n_i\frac{1}{\tau}}{\frac{1}{\tau} + n_i}(\bar{x}_i - \xi)(\bar{x}_i - \xi)^T
\end{aligned}$$

So the posterior distribution of θ can be written as

$$p(\theta \mid x) \propto |\Lambda|^{(\nu^* - p)/2 - 1}\exp\left(-\frac{1}{2}\mathrm{tr}\,(\Omega^*\Lambda) - \frac{1}{2}\sum_{i=1}^{k}\frac{1}{\tau_i^*}(\mu_i - \xi^*)^T\Lambda(\mu_i - \xi^*)\right).$$

with

$$\nu^* = \nu + n, \tag{B.1}$$

$$1/\tau_i^* = n_i + 1/\tau, \tag{B.2}$$

$$\mu^* = \frac{\frac{1}{\tau}}{\frac{1}{\tau} + n_i}\xi + \frac{n_i}{\frac{1}{\tau} + n_i}\bar{x}_i, \tag{B.3}$$

$$\Omega^* = \Omega + \sum_{i=1}^{k} \sum_{j:z_j=i}^{n_i} (x_j - \bar{x}_i)(x_j - \bar{x}_i)^T + \sum_{i=1}^{k} \frac{n_i \frac{1}{\tau}}{\frac{1}{\tau} + n_i}(\bar{x}_i - \xi)(\bar{x}_i - \xi)^T \tag{B.4}$$

Bibliography

Antoniak, C. E. (1974) Mixtures of Dirichlet processes with applications to nonparametric problems. *The Annals of Statistics*, **2**, 1152–1174.

Banfield, J. D. and Raftery, A. E. (1993) Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**, 803–821.

Besag, J., Green, P., Higdon, D. and Mengersen, K. (1995) Bayesian computation and stochastic systems. *Statistical Science*, **10**, 3–66.

Bozdogan, H. and Sclove, S. L. (1984) Multi–sample cluster analysis using Akaike's information criterion. *Annals of the Institute of Statistical Mathematics*, **36**, 163–180.

Campbell, N. A. and Mahon, R. J. (1974) A multivariate study of variation in two species of rock crab of genus *Leptograpsus*. *Australian Journal of Zoology*, **22**, 417–425.

Cheng, B. and Titterington, D. M. (1994) Neural networks: A review from a statistical perspective. *Statistical Science*, **9**, 2–54.

Cowles, M. K. and Carlin, B. P. (1996) Markov Chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, **91**, 883–904.

Crowder, M. J., Kimber, A. C., Smith, R. L. and Sweeting, T. J. (1991) *Statistical Analysis of Reliability Data*. London: Chapman and Hall.

DeGroot, M. H. (1970) *Optimal Statistical Decisions*. New York: McGraw-Hill.

Diebolt, J. and Robert, C. P. (1994) Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society, series B*, **56**, 363–375.

Escobar, M. D. and West, M. (1995) Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, **90**, 577–588.

Everitt, B. S. (1981) A Monte Carlo investigation of the likelihood ratio test for the number of components in a mixture of normal distributions. *Multivar. Behav. Res.*, **16**, 171–180.

Everitt, B. S. and Hand, D. J. (1981) *Finite Mixture Distributions*. London: Chapman and Hall.

Gelfand, A. E. and Smith, A. F. M. (1990) Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, **85**, 398–409.

Gelman, A. (1996) Inference and monitoring convergence. In *Markov Chain Monte Carlo in Practice* (Eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 131–143. London: Chapman and Hall.

Gelman, A. G., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995) *Bayesian Data Analysis*. London: Chapman and Hall.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, **6**, 721–741.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (eds) (1996) *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall.

Green, P. J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711–732.

Hastie, T. and Tibshirani, R. (1996) Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society, series B*, **58**, 158–176.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Hernandez-Avila, A. (1979) *Problems in Cluster Analysis*. Ph.D. thesis, Oxford.

Liu, C. and Rubin, D. B. (1998) Ellipsoidally symmetric extensions of the general location model for mixed categorical and continuous data. *Biometrika*, **85**, 673–688.

McLachlan, G. J. (1987) On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics*, **36**, 318–324.

McLachlan, G. J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. New York: John Wiley and Son.

McLachlan, G. J. and Basford, K. E. (1988) *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker.

Mengersen, K. L. and Robert, C. P. (1995) Testing for mixture via entropy distance and Gibbs sampling. In *Bayesian Statistics 5* (Eds J. O. Berger, J. M. Bernardo, A. P. Dawid, D. V. Lindley and A. F. M. Smith), pp. 255–276. Oxford: Oxford University Press.

Metropolis, N., Rosenbluth, A. W., Rosenblut, M. N., Teller, A. H. and Teller, E. (1953) Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1091.

Murphy, P. M. and Aha, D. (1995) Repository of machine learning databases. Irvine, CA: University of California, Dept of Information and Computer Science. Available by anonymous ftp from ics.uci.edu in directory pub\machine-learning-databases.

Newton, M. A. and Raftery, A. E. (1994) Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society*, **56**, 3–48.

Preston, C. J. (1976) Spatial birth-and-death processes. *Bulletin of the Institute of International Statistics*, **46**, 371–391.

Raftery, A. E. (1996) Hypothesis testing and model selection. In *Markov Chain Monte Carlo in Practice* (Eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 163–201. London: Chapman and Hall.

Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, series B*, **59**, 731–792.

Ripley, B. D. (1977) Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society, series B*, **39**, 172–212.

Ripley, B. D. (1987) *Stochastic Simulation*. New York: Wiley.

Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*. Cambridge University Press.

Robert, C. P. (1995) Convergence control methods for Markov Chain Monte Carlo algorithms. *Statistical Science*, **10**(3), 231–253.

Robert, C. P. (1996) Mixtures of distributions: Inference and estimation. In *Markov Chain Monte Carlo in Practice* (Eds W. R. Gilks, S. Richardson and D. J. Spiegelhalter), pp. 441–464. London: Chapman and Hall.

Spiegelhalter, D. J., Thomas, A., Best, N. and Gilks, W. R. (1994) *BUGS: Bayesian inference Using Gibbs Sampling, Version 0.30*. Medical Research Council Biostatistics Unit, Cambridge.

Stephens, M. (1997) *Bayesian Methods for Mixtures of Normal Distributions*. Ph.D. thesis, University of Oxford.

Tanner, M. and Wong, W. (1987) The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, **82**, 528–550.

Titterington, D. M. (1981) Contribution to the discussion of paper by M. Aitkin, D. Anderson and J. Hinde. *Journal of the Royal Statistical Society, series A*, **144**, 459.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985) *Statistical Analysis of Finite Mixture Distributions*. Wiley.

Venables, W. N. and Ripley, B. D. (1997) *Modern Applied Statistics with S-Plus*. Springer-Verlag, second edition.

West, M. (1997) Contribution to the discussion of paper by Richardson and Green (1997). *Journal of the Royal Statistical Society, series B*, **59**, 783–784.

West, M., Müller, P. and Escobar, M. D. (1994) Hierarchical priors and mixture models, with application in regression and density estimation. In *Aspects of Uncertainty* (Eds P. R. Freeman and A. F. M. Smith), pp. 363–386. Chichester: John Wiley and Sons.

Wolfe, J. H. (1971) A Monte Carlo study of the sampling distribution of the likelihood ratio for mixtures of multinormal distributions. Technical Report Tech. Bull. STB 72-2, Nav. Pers. and Trans. Res. Lab., San Diego.