Contribution to the Analysis of Latent Structures

Ernest Fokoué

A Dissertation Submitted to
The University of Glasgow
For the Degree of

Doctor of Philosophy

UNIVERSITY
of
GLASGOW

DEPARTMENT OF STATISTICS

September 2001

Typeset with LaTeX

Contribution to the Analysis of Latent Structures

Ernest Fokoué
Doctor of Philosophy

Department of Statistics
University of Glasgow

Abstract

What is a *latent variable*? Simply defined, *a latent variable is a variable that cannot be directly measured or observed.* A latent variable model or *latent structure* model is a model whose structure contains one or many latent variables. The subject of this thesis is the study of various topics that arise during the analysis and/or use of latent structure models. Two classical models, namely the factor analysis (FA) model and the finite mixture (FM) model, are first considered and examined extensively, after which the mixture of factor analysers (MFA) model, constructed using ingredients from both FA and FM is introduced and studied at length. Several extensions of the MFA model are also presented, one of which consists of the incorporation of fixed observed covariates into the model. Common to all the models considered are such topics as: (a) **model selection** which consists of the determination or estimation of the dimensionality of the latent space; (b) **parameter estimation** which consists of estimating the parameters of the postulated model in order to interpret and characterise the mechanism that produced the observed data; (c) **prediction** which consists of estimating responses for future unseen observations. Other important topics such as **identifiability** (for unique solution, interpretability and parameter meaningfulness), **density estimation**, and to

a certain extent aspects of **unsupervised learning and exploration of group structure** (through clustering, data visualisation in 2D) are also covered. We approach such topics as parameter estimation and model selection from both the likelihood-based and Bayesian perspectives, with a concentration on Maximum Likelihood Estimation via the EM algorithm, and Bayesian Analysis via Stochastic Simulation (derivation of efficient Markov Chain Monte Carlo algorithms). The main emphasis of our work is on the derivation and construction of computationally efficient algorithms that perform well on both synthetic tasks and real-life problems, and that can be used as alternatives to other existing methods wherever appropriate.

This thesis is organised as follows: Chapter 1 presents a general introduction to latent variable models, together with a brief overview of the statistical and computational methods and tools used to study them. In chapter 2, we present a review of the factor analysis model. We propose a new approach to model selection based on stochastic simulation, and we suggest new ideas on a Bayesian sampling alternative to varimax factor rotation. Chapter 3 starts with a brief review of finite mixture models, along with a survey of some recent research in the field. However, the major part of this chapter introduces and extensively studies the mixture of factor analysers model. More specifically, we present a thorough analysis of the stochastic simulation treatment of mixtures of factor analysers, with applications to both real and synthetic data, and we offer a comparison between our approach and the existing results in the literature. Chapter 4 extends the mixtures of factor analysers model by incorporating fixed observed covariates into the model via the latent variables. An EM algorithm is then constructed for parameter estimation and prediction, and the resulting scheme is tested on artificial data. In Chapter 5, we use the assumption of conditional independence to allow our manifest vector to be made up of variables having different distributions, and we use a generalised linear models formulation to ease the analysis of the resulting model, and to construct the corresponding algorithms for parameter estimation. Chapter 6 presents our conclusion, a discussion and elements of our future research.

To The Inner Christ of God

whose Divine Light shines forever

to heal and illumine the mind of every earnest Truth Seeker,

I humbly dedicate and surrender this thesis and every subsequent research work.

Acknowledgements

This thesis is the outcome of my doctoral research work carried out from October 1998 to September 2001 within the department of Statistics of the University of Glasgow. During those three years, many people inspired me directly and/or indirectly.

First and foremost, I would like to express my heartfelt thanks and my deepest gratitude to my PhD supervisor **Professor D. Mike Titterington**. His patience, his humility, his vast knowledge and his unique experience have been a constant source of inspiration and encouragement throughout my research.

I am deeply grateful to **Professor Adrian W. Bowman** for awarding me the most needed PhD scholarship, and for offering me a Graduate Teaching Assistantship.

I wish to thank *Professor Ilya S. Molchanov* for inspiring me in various ways, and for opening my awareness to many interesting aspects of Mathematical Sciences.
I am indebted to *Dr Agostino Nobile* for availing himself to discuss various aspects of Bayesian statistics with me.
I would like to thank *Dr Jim Kay* for his contagious enthusiasm and his very inspiring conversations on various interesting statistical topics.

I wish to thank everyone in the Department of Statistics for making the whole department a conducive environment for knowledge exchange.
I owe a big debt of gratitude to my family and all my friends for their moral support throughout my research.
Last but not least, I would like to express my love to the people of Scotland.

Notation

Variables, Parameters and Sets

Notation	Description
$\mathcal{X}$	Generic name for the input (sample) space.
$\mathcal{H}$	Generic name for latent space.
Θ	Generic name for parameter space.
$\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$	Bold letters used for random variables.
$\boldsymbol{x}^\mathsf{T} = (x_1, \cdots, x_p)$	p-dimensional random variable (in column vector form).
$\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$	Bold small letters used for random variates.
$\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$	Bold capital letters used for sample, ie $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$.
$\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Z}$	Also used as data matrices, $\mathbf{X} = [\boldsymbol{x}_1 \vert \cdots \vert \boldsymbol{x}_n]^\mathsf{T}$ is an $n \times p$ matrix.
$\mathbf{X}^*$	Complete-data matrix (contains both observed and latent data).
$\mathbb{R}$	Set of real numbers
$\mathbf{I}_k$	k-dimensional identity matrix
$\mathsf{diag}(m_1, \cdots, m_k)$	k-dimensional diagonal matrix
$\mathbf{1}_k$	k-dimensional vector of ones.
θ	Generic name for one parameter.
$\boldsymbol{\theta}$	Complete collection of model parameters.
Λ^T	Transpose of matrix Λ.
$\Lambda_{\cdot c}$	c-th column of matrix Λ.
$\Lambda_{r\cdot}$	r-th row of matrix Λ.
λ_{rc}	rc-th entry of matrix Λ.
$\boldsymbol{\theta}^{(t)}$	Parameter set at the t-th iteration.

Functions, Densities and Distributions

Notation	Description
$\mathbf{Pr}(\boldsymbol{x} = a)$	Probability that $\boldsymbol{x}$ equals a.
$\boldsymbol{p}(\boldsymbol{x})$	Probability density of the variate $\boldsymbol{x}$.
$\mathbf{Pr}(\boldsymbol{x} = a \vert \boldsymbol{y} = b)$	Conditional probability that $\boldsymbol{x}$ equals a given $\boldsymbol{y} = b$.
$\boldsymbol{p}(\boldsymbol{x}\vert\boldsymbol{\theta})$	Conditional density of $\boldsymbol{x}$ given $\boldsymbol{\theta}$.
$\mathbb{E}[\boldsymbol{x}]$	Expectation of a Random Variable.
$\mathbb{V}[\boldsymbol{x}]$	Variance of a Random Variable $\boldsymbol{x}$
$\mathbb{I}_E(x)$	Indicator function value of x in the set E.
$\mathcal{N}_p(\mu, \Sigma)$	p-dimensional normal (Gaussian) distribution.
$\mathcal{N}(\mu, \sigma^2)$	Univariate normal (Gaussian) distribution.

$\mathsf{Ga}(\alpha, \beta)$	Gamma distribution.
$\mathsf{Di}(\alpha_1, \cdots, \alpha_k)$	Dirichlet distribution.
$\mathsf{Mn}(n; \pi_1, \cdots, \pi_k)$	Multinomial distribution.
$\mathsf{Po}(\eta)$	Poisson distribution.
$\mathsf{Bi}(n, p)$	Binomial distribution.
$\mathsf{Be}(\alpha, \beta)$	Beta distribution.
$\mathsf{Ber}(\pi)$	Bernoulli distribution.
$\mathsf{Exp}(\theta)$	Exponential distribution.
$\mathcal{W}_k(p, \Sigma)$	Wishart distribution.
$\boldsymbol{L}(\boldsymbol{\theta}; \mathbf{X})$	Likelihood function.
$\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X})$	Log-likelihood function.
$\mathsf{cov}(\boldsymbol{x}, \boldsymbol{z})$	Covariance of $\boldsymbol{x}$ and $\boldsymbol{z}$.
$[\boldsymbol{x} \vert \cdots] \sim \mathcal{D}(\boldsymbol{\theta})$	The full conditional distribution of $\boldsymbol{x}$ is $\mathcal{D}$ with parameters $\boldsymbol{\theta}$.
$\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{\theta})$	$\boldsymbol{x}$ follows distribution $\mathcal{D}$ with parameters $\boldsymbol{\theta}$.

Contents

List of Figures

Chapter 1

Introduction

Without a measureless and perpetual uncertainty,
the drama of human life would be destroyed.
Sir Winston Churchill

1.1 What is a latent variable model?

In recent years, the analysis of latent variable models has widened its scope, extending its ramifications from its original social sciences community to many other scientific communities such as mainstream statistics, neural networks and machine learning communities. This intensification of interest, partly encouraged and fuelled by the availability of powerful computational facilities and the development of a variety of sophisticated statistical methods, has allowed the use of latent variable models in many real-life applications in various different fields ranging from engineering to physical and biological sciences. The increasingly significant contribution brought in by the development of the Bayesian paradigm has also added to the established frequentist maximum likelihood estimation techniques, allowing the development of a great variety of methods and tools for latent structures analysis. In order to set the ground for an introduction to the building blocks of this vast topic, we begin this section by giving very general definitions of both a latent variable and a latent variable model.

Definition 1: A *latent variable* is a variable that cannot be directly measured or observed. The idea here is that such a variable has not yet *manifested* itself, and is therefore

qualified as *latent* as opposed to the *manifest* ones.

Definition 2: A *latent variable* model or *latent structure* model is a model whose structure contains a set of *latent* variables, a set of *manifest* variables and a mechanism linking the two sets of variables.

Note: It is worth pointing out here that the above definition of a latent variable model does not make any probabilistic assumption. This rather general definition is deliberate. In fact, it allows us to touch on deterministic models that have turned out to be latent variable models in their own right, because of the existence of non-directly observable variables in their structure. Very broadly speaking, we can essentially distinguish two different classes of latent variable models.

- Probabilistic latent variable models
  - Latent variable models for data reduction.
  - Latent variable models for density modelling.
- Deterministic latent variable models.

1.2 Probabilistic latent variable models

Probabilistic latent variable models are the ones generally referred to in the majority of texts on the topic. They all have in common the fact that they make probabilistic assumptions. We distinguish two subclasses of such models: (a) models for data reduction and (b) models for density modelling.

1.2.1 Latent variable models for data reduction

Latent variable models for data reduction are the subclass of models generally treated in the mainstream texts on the topic. Latent variable modelling was originally essentially a subclass of multivariate statistical analysis born from the need to condense many variables from large-scale statistical enquiries into a much smaller number of latent con-

structs with as little loss of information as possible.

Data reduction for interpretation: Historically, the need for latent variable modelling arose from the fields of social and behavioural sciences where investigators wanted to quantify information on non-directly measurable concepts such as *intelligence, social class, personality* and *ambition.* This social science perspective of latent variable modelling hypothesises a set of latent constructs (concepts) and then accordingly designs an experiment consisting of *manifest* variables which can be measured and which are related in some ways to the latent quantities of interest. From this perspective, the experimenter seeks to condense many observed variables into the fewer hypothesised latent constructs so as to provide an interpretation (meaning) of latent scores. In this case, he/she is generally also interested in the characterisation of the mechanism linking the latent and manifest variables. The observed quantities can therefore be thought of as the *effects*, while the latent scores are their *causes*, or vice-versa. According to this view, manifest quantities such as high grades in Mathematics, Physics, IQ tests and other intellectual disciplines would therefore be interpreted as the effects of high intelligence reflected by high scores on this latent variable named *intelligence.* It is important to note that, in this case, the starting point of the modelling exercise is the set of hypothetical latent constructs (with possibly some a priori meanings and labels attached to them), and the observed variables are just a means to this end. The source (cause) of what we see is in reality something latent that we do not see. This cause-effect interpretation generally raises a lot of controversies, and, for that reason, we do not address it here.

Pure data reduction: Another view of latent variable modelling common to social sciences, physical sciences and engineering uses latent scores as a convenient parsimonious description (representation) of complex high-dimensional observations. In fact, in applications where observed variables are high-dimensional and assumed to be highly correlated, it is tempting and often desirable to seek just a few uncorrelated latent variables that explain in some way all the associations existing among the initial manifest

variables. In pattern recognition for instance, a digit only occupies a small portion of the rectangular grid, and the essential information about a high-dimensional vector of a handwritten digit can therefore be represented in a much lower-dimensional subspace without much loss of information. From this perspective, the starting point of the modelling exercise is the manifest variable for which we simply seek an internal parsimonious representation. Here, no interpretation of latent scores is *a priori* sought[1].

Note: This first class of latent variable models explain the associations among the observed variables by making use of a concept known as the assumption or axiom of *conditional independence* that we will discuss later.

Examples: Generically, there are four main types of latent variable models for data reduction defined according to the types (continuous or categorical) of manifest and latent variables that they model. The general classification taken from Bartholomew (1987) is given in Table 1.1. Modern texts in multivariate statistical analysis provide a comprehensive coverage of the above models, and we refer the reader to such references as Lawley and Maxwell (1971), Everitt and Hand (1981), Bartholomew (1987), Anderson (1984), Press (1972), Johnson and Wichern (1998) and Krzanowski and Marriott (1995). There are many extensions of the above models, some of which consist of combinations of ingredients from the generic models. von Eye and Clogg (1994) provide a collection of articles on some relatively recent advances in the analysis of these probabilistic latent variable models.

Link to classical statistical techniques: As we shall see later, there exist close connections between the above data reduction latent variable models and some classical statistical techniques. Latent variable models are similar to measurement error models in the sense that manifest variables are modelled by a combination of latent variables plus an error term. In fact, as we shall see later, holding the latent variables fixed in the normal factor analysis model allows us to treat the resulting model as a multivariate

[1]From this perspective, no label or meaning is attached *a priori* to the latent constructs, but the experimenter can always carry out an *a posteriori* interpretation of the latent variables once the reduction is achieved.

		Manifest variables	
		Continuous	**Categorical**
Latent Variable	**Continuous**	Factor Analysis	Latent Trait Analysis
	Categorical	Latent Profile Analysis	Latent Class Analysis

Table 1.1: Classification of latent variable models

regression model. On the other hand, as established by Everitt and Hand (1981) and Titterington, Smith, and Makov (1985), there is a close connection between latent class models and finite mixture models. With an ever-increasing number of real-life problems giving rise to complex multivariate observations that can be adequately summarised by fewer latent variables, the analysis of latent structure models has now become an integral part of multivariate statistical analysis.

1.2.2 Latent variable models for density modelling

The concept of latent variable is also used as a convenient way to model the probability density function of random observations assumed to be related in some way to some other variables that are hidden (latent) and cannot therefore be directly observed. While latent variable models for data reduction are exclusively based on multivariate observations, latent variable models aimed at density modelling can be used for both univariate and multivariate random variables. In this case, the *axiom of conditional independence* is not needed, since there is no interest in data reduction[2].

Examples: Finite mixture models and Hidden Markov models fall into this category. Gaussian Process Classifiers, as studied in Fokoué (1998) and Csató, Fokoué, Opper, Schottky, and Winther (2000), are also latent variable models from this subclass.

[2]The Mixture of Factor Analysers model that we will be studying later is a combination of both data reduction and density modelling.

1.3 Deterministic latent variable models

For many purists, it might seem inadequate to call these models latent variable models, but, as we said earlier, they comply with our general definition. Deterministic models like feedforward neural networks are now presented in some texts as latent variable models. In fact, the hidden layer of a multilayer perceptron (MLP)[3] constitutes a latent space in its own right, and the differences with the traditional latent variable models lie in the fact that MLP's are essentially nonlinear models and they do not have any probabilistic assumptions attached to them.

Note: In this thesis, we restrict our focus to probabilistic latent variable models, and especially to various aspects of factor analysis and finite mixture distributions. From now on, all the latent variable models mentioned will be probabilistic latent variable models.

1.4 Notation and terminology

Both manifest and latent variables are represented mathematically by random variables, since they vary from one subject (entity) to another in a random manner. The relationships between them are therefore expressed in terms of probability distributions. For notational economy, we use $\boldsymbol{x}^{\mathsf{T}} = (x_1, \cdots, x_p)$ to denote both our p-dimensional random manifest variable in column vector form, and its corresponding random variate or sampled value. Similarly, $\boldsymbol{z}^{\mathsf{T}} = (z_1, \cdots, z_q)$ denotes both the q-dimensional continuous random latent variable and its corresponding random variate or sampled value. Our categorical latent variable is denoted by $\boldsymbol{y}$. With a slight abuse of notation, we also use $\boldsymbol{y}$ as a vector of indicators or categories. With k categories, we have $\boldsymbol{y}^{\mathsf{T}} = (y_1, \cdots, y_k)$, where $y_j = 1$ if $\boldsymbol{y} = j$ and $y_j = 0$ otherwise. We use $\mathcal{X}$ as a generic name for our sample space. We use $\boldsymbol{x}^*$ to denote a vector that contains the complete collection (all manifest and all latent variables) of the variables of the models. The corresponding sample is

[3]It is worth pointing out here that the Neural Networks literature now has a good number of articles treating probabilistic versions of multilayer perceptrons

denoted by $\mathbf{X}^*$. For the majority of this thesis, we consider a continuous sample space, namely $\mathcal{X} \subseteq I\!R^p$. Our latent space is either categorical or continuous, or a product of both, and is denoted by $\mathcal{H}$, while Θ denotes our parameter space. We use θ as a generic name for a parameter, and $\boldsymbol{\theta}$ denotes a collection of model parameters. Our samples are denoted by bold capital letters, and, according to that, $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ is a sample of n observations, while $\mathbf{Z} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n\}$ is the corresponding sample of continuous latent variables. We also use our samples as data matrices, which means that $\mathbf{X}^\intercal = [\boldsymbol{x}_1 | \cdots | \boldsymbol{x}_n]$ is an $n \times p$ matrix. For simplicity, we use the same $\boldsymbol{p}$ to denote the probability density function whatever the variable. Thus, $\boldsymbol{p}(\boldsymbol{x})$ is the density of the random variate $\boldsymbol{x}$, while $\boldsymbol{p}(\boldsymbol{z})$ is the density of the random variate $\boldsymbol{z}$. We adopt a similar simplification for $\mathbf{Pr}$, the probability distribution function.

1.5 Marginal density of manifest variables

By definition, a latent variable model contains both manifest and latent variables, and we can write an expression of the joint distribution of both variables as follows:

$$\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{p}(\boldsymbol{z})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z}) = \boldsymbol{p}(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}). \tag{1.1}$$

The joint density in equation (1.1) will also be referred to as the complete-data density for reasons that will become clear when we consider the statistical analysis of our models. Since we only observe $\boldsymbol{x}$, our inferences will be based on the marginal distribution of $\boldsymbol{x}$. For a continuous latent space $\mathcal{H} \subseteq I\!R^q$, the marginal density of $\boldsymbol{x}$ (also referred to as the observed-data density) is given by

$$\boldsymbol{p}(\boldsymbol{x}) = \int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{z})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z}. \tag{1.2}$$

For a categorical latent space $\mathcal{H} = \{1, \cdots, k\}$, this marginal density of $\boldsymbol{x}$ becomes

$$\boldsymbol{p}(\boldsymbol{x}) = \sum_{j=1}^{k} \mathbf{Pr}(\boldsymbol{y} = j)\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j). \tag{1.3}$$

For the rest of this chapter, and without loss of generality, we base our developments on models with a continuous latent space. In many cases, the results apply *mutatis mutandis* to cases where the latent space is discrete.

1.6 Axiom of conditional independence

Given a random observation $\boldsymbol{x}^\mathsf{T} = (x_1, \cdots, x_p)$ on p manifest variables assumed to be correlated, our aim in latent variable modelling is to determine a set of $q < p$ uncorrelated latent variables $\boldsymbol{z}^\mathsf{T} = (z_1, \cdots, z_q)$ that explain all[4] the associations (dependencies) among the manifest variables. This means that, once all the q values of the z_j's are known and held fixed, then the x_i's will be uncorrelated, since the correlations among the x_i's are induced by the z_j's. In probabilistic terms, this means that *the x_i's are conditionally independent given the values of the z_j's.* This statement is often referred to as the assumption (or axiom) of conditional (or local) independence[5]. It is a fundamental assumption of latent variable modelling when data reduction is the aim, and, as we shall see in the following chapters, it will appear in many of our models in various different ways. According to the conditional independence axiom, the number of latent variables q must therefore be chosen in such a way that the conditional density of $\boldsymbol{x}$ given $\boldsymbol{z}$ has the form

$$\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{p} \boldsymbol{p}_i(x_i|\boldsymbol{z}). \tag{1.4}$$

With that, the aim of data reduction can be expressed as follows:

> Given a sample $\mathbf{X}$ of multivariate observations, latent variable modelling essentially seeks the smallest q, the adequate $\boldsymbol{p}(\boldsymbol{z})$ and all the $\boldsymbol{p}_i(x_i|\boldsymbol{z})$ such that the marginal distribution $\boldsymbol{p}(\boldsymbol{x})$ of $\boldsymbol{x}$ has the structure
>
> $$\boldsymbol{p}(\boldsymbol{x}) = \int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{z}) \prod_{i=1}^{p} \boldsymbol{p}_i(x_i|\boldsymbol{z}) d\boldsymbol{z}. \tag{1.5}$$

Note: As stated by Bartholomew (1987), *it can be misleading from a pure statistical point of view to think of the axiom of conditional independence as an assumption of the*

[4]This is an important part of the assumption in the sense that we want the z_j's to be complete, meaning that no extra latent variable apart from the q chosen is needed to account for the assumed correlations among the x_i's. In other words, once the q latent variables are determined, any other latent variable should be redundant.

[5]As we said earlier this axiom is not needed when we simply want a convenient representation of the density of $\boldsymbol{x}$. All we need in such a case is either equation (1.2) or equation (1.3).

kind that could be tested empirically, because there is no way in which $\boldsymbol{z}$ can be fixed[6]*, and therefore no way in which the independence can be tested. It is better regarded as a definition of what we mean when we say that the set of latent variables $\boldsymbol{z}$ is complete*[7]*.*

1.7 Difficulties and Problems

1.7.1 Indeterminacy and non-identifiability

From a given sample of observations, all that we can truly model is the marginal density $\boldsymbol{p}(\boldsymbol{x})$ of the manifest variable, and it is obvious that, for a given value of q, there exist various choices $\boldsymbol{p}(\boldsymbol{z})$ and $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z})$ such that $\boldsymbol{p}(\boldsymbol{x})$ can be decomposed as in equation (1.2). In other words, such a decomposition of $\boldsymbol{p}(\boldsymbol{x})$ is not unique. This phenomenon, known as indeterminacy or non-identifiability, is one of the bottlenecks of latent variable modelling. We shall address this issue in each model considered subsequently.

1.7.2 Multimodality and computational difficulties

In high-dimensional spaces, the surface of the likelihood function is often likely to exhibit genuine multimodality, leading to the existence of many local maxima as its natural consequence. Besides this genuine multimodality, there is another type of potential multimodality that could arise from situations where the prior distribution $\boldsymbol{p}(\boldsymbol{z})$ is symmetric thereby causing $\boldsymbol{p}(\boldsymbol{x})$ to be invariant to permutations of the indices of $\boldsymbol{z}$. This is the case for instance with finite mixtures. This phenomenon constitutes a serious bottleneck for both maximum likelihood estimation and Bayesian inference. In practice, estimation and inference in such situations require more sophisticated algorithms. We shall return to this aspect in subsequent chapters.

[6] This statement is only partially true, since the use of the complete-data methods mentioned earlier allow the imputation of hypothetical values to the latent variables throughout the iterative estimation procedures.

[7] This completeness of the set of q latent variables, by avoiding a more complex model with more latent variables, can be thought of as an application of Ockham's razor principle (Law of parsimony) which states that **unnecessarily complex models should not be preferred to simpler ones.**

1.7.3 Efficiency and interpretability

Since latent structure models may involve very high-dimensional data, the number of parameters that characterise these models can also be very large. First of all, it is clear that models with too many parameters are in general difficult to interpret, and are computationally very intensive. Unless the number of observations is large enough to contain sufficient items of information about the large number of parameters, the estimation of those parameters is often inefficient and prone to over-fitting. For many models considered in our work, this complexity is often dealt with by imposing some constraints on the parameters in order to have reduced models that are therefore easier to understand and interpret, computationally more realistic, and also more useful in prediction. As far as interpretability is concerned, I totally espouse Marriott (1974)'s view expressed in the following statement:

If the results disagree with informed opinion, do not admit a simple logical interpretation, and do not show up clearly in a graphical presentation, they are probably wrong. There is no magic about numerical methods, and many ways in which they can break down. They are a valuable aid to the interpretation of data, not sausage machines automatically transforming bodies of numbers into packets of scientific facts.

In other words, the analysis of models should produce meaningful results that can be easily interpreted.

1.8 Goals, Issues and Applications

Fundamental to any latent structure analysis is the crucial choice of the prior distribution of the latent variables and the conditional distribution of the manifest variables given the latent variables. The choice of these two distributions is essentially arbitrary and does not form part of the systematic analysis process. It consists of mere assumptions made on the basis of expertise or sometimes for convenience, but also on the basis of

the appropriateness of the distribution to the modelling task at hand. In practice, there are standard established distributions that have stood the test of time and that almost always produce satisfactory results.

If we assume that the above choice of distributions has been made, the analysis of latent structure models involves one or many of the following issues:

- **Data reduction.** Essentially, data reduction is achieved through the *estimation of latent scores.* Broadly speaking, we distinguish two main aspects here:
  - *Characterisation.* The interest in this case is in estimating the latent scores for the sample of observations used to analyse the model. It is often hoped that the set of estimated latent scores will provide a much simpler structure, and thereby allow an easier interpretation of the interdependence amongst the original variables.
  - *Prediction.* The aim in prediction is the estimation of latent scores for future unseen observations. This often presupposes that the model has been analysed and validated, and is being used as a device (tool) to provide intrinsic representation of new observations. This is particularly useful in pattern recognition where data reduction is used as a preprocessing tool.

  There are many applications of data reduction in real life, among which are the following:
  - *Exploration of group structure.* It is a common practice whenever that is reasonable, to project a high-dimensional dataset onto the plane. In general, a scatter plot of the resulting latent scores is a data visualisation device that can be used to explore the existence of a group structure in the population under study. It is however fair to point out that this is not always guaranteed to reveal the group structure, especially if a 2-dimensional latent space is not an adequate intrinsic representation of the original manifest variables.
  - *Data compression.* Data compression is used to reduce the amount of space required for storing huge amounts of data. It is particularly useful in *scientific*

imaging where image compression allows huge databases of images to be stored in minimum amounts of space. Obviously, the original data are later retrieved and recovered through a process known as *reconstruction.*

- *Feature extraction.* Data reduction is also extensively used in *pattern recognition* as a preprocessing tool for feature extraction. In fact, as explained by Bishop (1995), although a certain amount of information is always lost during the data reduction process, many classification and regression systems generally produce a better performance when the input is first projected onto its intrinsic lower-dimensional space before actually being processed. This is particularly true if the input variables are highly correlated, since the sparseness induced by strong correlation leads to an inefficient use of the input space.

- **Density estimation.** There are two main aspects of density estimation to be considered here: sample density estimation and predictive density estimation. By predictive density estimation in this context we have in mind the estimation of density for unseen observations. Sample density estimation on the other hand concerns itself with the estimation of density for the observations contained in the sample.

- **Parameter estimation.** Since latent structure models are essentially parametric, the estimation of model parameters constitutes one of the main issues of interest. In fact, some approaches to both data reduction and density estimation require the parameters of the model to have been estimated. There are two main ways in which parameters are used:

  - *Characterisation.* In many applications of latent structure modelling, the experimenter is interested in interpreting the way in which manifest variables affect latent variables or vice-versa, or the way in which groups of manifest variables combine to form latent constructs. Model parameters are often used as a way to achieve such characterisations. Such cases are frequent in social sciences, and the analysis of the model places an emphasis on the ability

to provide useful and meaningful parameter estimates that are as easy to interpret as possible.

- *Instrumental.* There are also many applications in which parameters are simply instrumental in that they are only needed as a way to compute either estimated latent scores or predictive density estimates. In such instances, the experimenter simply needs a valid set of model parameters and is not interested in interpreting them.

- **Model selection.** Essentially the aim here is to determine the dimension of minimal latent subspace. In other words model selection consists of the determination or estimation of the smallest number of latent variables that can be used to represent the original manifest variables without much loss of information and model the density of the manifest variables as adequately as possible.

Note: For simplicity, we have so far written the expressions of our probability density functions without explicitly showing their dependence on a set of parameters $\boldsymbol{\theta}$. However, since the models we are dealing with are essentially parametric and one of the main goals in the analysis of such models is the estimation of parameters, we now include parameters in our expressions whenever necessary.

1.9 Parameter estimation

Rigorously speaking, our complete collection of model parameters $\boldsymbol{\theta}$ can be divided into a subset of parameters for the observed part of the model, $\theta_{\boldsymbol{x}}$, say, and a subset for the missing part, denoted by $\theta_{\boldsymbol{z}}$. Thus, $\boldsymbol{\theta} = \{\theta_{\boldsymbol{x}}, \theta_{\boldsymbol{z}}\}$. For simplicity, we only insist on this difference if the need arises. The complete-data density is therefore given by

$$\boldsymbol{p}(\boldsymbol{x}^*|\boldsymbol{\theta}) = \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) = \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}), \tag{1.6}$$

and the corresponding observed-data density has the following form:

$$\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta}) = \int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{z} = \int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta}) d\boldsymbol{z}. \tag{1.7}$$

We approach all our parameter estimation tasks from both the likelihood-based and Bayesian perspectives.

1.9.1 Observed-data likelihood and posterior

Given a sample $\mathbf{X}$ of independent and identically distributed observations, the observed-data likelihood function can be written as

$$\boldsymbol{L}(\boldsymbol{\theta};\mathbf{X}) \propto \boldsymbol{p}(\mathbf{X}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{\theta}), \tag{1.8}$$

and, for a given prior density $\boldsymbol{p}(\boldsymbol{\theta})$, the observed-data posterior can be expressed as

$$\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X}) \propto \boldsymbol{L}(\boldsymbol{\theta};\mathbf{X})\boldsymbol{p}(\boldsymbol{\theta}). \tag{1.9}$$

However, in this particular setting where part of the model is latent, the marginal density over the latent variables generally leads to observed-data likelihood functions such as (1.8) that are generally not mathematically tractable. From a likelihood-based perspective for instance, such likelihood functions do not allow the derivation of closed form expressions for parameter estimates, and gradient methods like Newton-Raphson type iterative algorithms have been used for many decades to find maximum likelihood estimates. However, the main drawbacks of this class of algorithms is that they are generally very complicated and awkward, and their convergence is often not guaranteed. From a Bayesian perspective, posterior densities like (1.9) generally lead to intractable integrals in high-dimensional spaces which makes it impossible to obtain closed form expressions for the posterior averages of interest. In practice, asymptotic approximations are used to tackle such intractabilities, but they suffer from the crucial drawback of not allowing a systematic and objective assessment of how close approximating distributions get to the true posterior distribution of interest.

An alternative to the use of the marginal density of the manifest variable is based on the formulation of latent variable modelling as an incomplete-data problem where the latent variables are treated as missing data. In this thesis, all our algorithms, from both the likelihood-based and Bayesian perspectives, make use of this idea.

1.9.2 Latent variable models as missing data models

Since latent variables are not observed, they can be treated as missing variables[8]. Latent structure models can therefore be thought of as a subclass of *missing data models*, also referred to as *incomplete-data models*. As we said earlier, it turns out that the marginal density of $\boldsymbol{x}$ leads to likelihood functions that are computationally not easy to deal with. In practice, a vast body of algorithms have now been developed that avoid basing inferences and estimations on such complicated marginal likelihood functions, and that instead use the complete-data density of equation (1.1) and the corresponding complete-data likelihood functions. The EM algorithm Dempster, Laird, and Rubin (1977) and the Data Augmentation algorithm Tanner and Wong (1987) are respectively the likelihood-based and Bayesian applications of this complete-data approach. These two very popular algorithms perform parameter estimation through iterative processes, the former being *deterministic* while the latter is *stochastic*. The key ideas behind those two algorithms are essentially very simple and intuitive, and can be summarised in the following statement:

> *Solve a difficult incomplete-data problem by repeatedly solving a tractable complete-data version of it until some convergence criterion is satisfied.*

Despite the fact that these algorithms are generally slower than the ones based on marginal densities, they have two main advantages that certainly justify their ever increasing popularity: (a) They are simple and easy to write, especially for the very frequently encountered exponential family of distributions; (b) Proven theorems exist that establish their convergence.

1.9.3 Distribution of latent variables

In practice, once $\boldsymbol{x}$ is observed, one of our main interests is to gather information about $\boldsymbol{z}$ given $\boldsymbol{x}$. Since the distribution of $\boldsymbol{x}$ depends on $\boldsymbol{\theta}$, information on $\boldsymbol{z}$ is obtained by

[8]The missingness here is systematic, unlike in other cases where the missingness arises when data are simply not available for a given observation.

specifying the posterior conditional density $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})$ of $\boldsymbol{z}$ given $\boldsymbol{x}$ and $\boldsymbol{\theta}$ given by

$$\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}) \propto \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\theta}). \tag{1.10}$$

$\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})$ is also called the *predictive density of the missing data given* $\boldsymbol{\theta}$, and plays a central role in our complete-data algorithms since it captures one of the main ingredients of those algorithms, namely the interdependence between the parameters $\boldsymbol{\theta}$ and the missing data $\boldsymbol{z}$. Schafer (1997) explains the central role of $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})$ as follows: *when viewed as a probability distribution it summarises knowledge about* $\boldsymbol{z}$ *for any assumed value of* $\boldsymbol{\theta}$*, and when viewed as a function of* $\boldsymbol{\theta}$ *it conveys the evidence about* $\boldsymbol{\theta}$ *contained in* $\boldsymbol{z}$ *beyond that already provided by* $\boldsymbol{x}$.

Estimation of latent scores: Once $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})$ is completely specified, it can be used the compute the conditional expectation of the latent variable given $\boldsymbol{x}$ and $\boldsymbol{\theta}$ as follows:

$$\mathbb{E}\left[\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}\right] = \int_{\mathcal{H}} \boldsymbol{z}\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})d\boldsymbol{z}. \tag{1.11}$$

For many models like those with an assumption of normality, the integral calculation of equation (1.11) is not necessary, and the conditional expectation $\mathbb{E}\left[\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}\right]$ is easily obtained by direct application of the properties of expectations. In general, an expression for the conditional variance-covariance matrix of $\boldsymbol{z}$ given $\boldsymbol{x}$ is also easily derived. In the following sections, we give details of both the EM algorithm and the Data Augmentation algorithm, and we also touch on some of their beautiful properties that justify their appropriateness for our context. The complete-data likelihood that they both use is denoted by $\boldsymbol{L}(\boldsymbol{\theta};\mathbf{X}^*)$, and the corresponding complete-data log-likelihood is denoted by $\boldsymbol{\ell}(\boldsymbol{\theta};\mathbf{X}^*)$.

1.10 The EM Algorithm

As we anticipated earlier, the building blocks of the EM algorithm rest on the interdependence between the missing data $\boldsymbol{z}$ and the parameters $\boldsymbol{\theta}$ expressed by the central role of the predictive density $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta})$ of the missing data given $\boldsymbol{\theta}$. On the one hand, $\boldsymbol{z}$ contains information relevant to the estimation of $\boldsymbol{\theta}$, and, on the other hand, $\boldsymbol{\theta}$ contains

information that allows us to find likely values of $\boldsymbol{z}$. When only $\boldsymbol{x}$ is observed, this interdependence between $\boldsymbol{z}$ and $\boldsymbol{\theta}$ can be exploited to estimate $\boldsymbol{\theta}$ as follows: Choose an initial estimate for $\boldsymbol{\theta}$. (i) Fill in the missing $\boldsymbol{z}$ based on the current estimate of $\boldsymbol{\theta}$. (ii) Re-estimate $\boldsymbol{\theta}$ based on both $\boldsymbol{x}$ and the filled-in $\boldsymbol{z}$. Iterate the two-stage scheme defined by (i) and (ii) until the estimates converge. A pseudo-code form is as follows:

The EM Algorithm

- Choose a tolerance ϵ and initial ($t = 0$) values $\boldsymbol{\theta}^{(0)}$ for the parameters.
- **Repeat**
- $t = t + 1$
- – **E-step** - This Expectation step compensates for the missingness by averaging the complete-data log-likelihood of the parameters over the probability distribution $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ of the latent variables $\boldsymbol{z}$.
  $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}\left[\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X}^*)|\mathbf{X}, \boldsymbol{\theta}^{(t)}\right]$ with
  $\mathbb{E}\left[\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X}^*)|\mathbf{X}, \boldsymbol{\theta}^{(t)}\right] = \int_{\mathcal{H}} \boldsymbol{\ell}(\boldsymbol{\theta}, \mathbf{X}^*)\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})d\boldsymbol{z}$
  – **M-step** - This Maximisation step then performs the traditional Maximum Likelihood principle on the above expected log-likelihood $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$, which means determining $\boldsymbol{\theta}^{(t+1)}$ that maximises $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$.
  $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}}{\mathbf{argmax}} \quad Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$
- **Until** $\|\boldsymbol{\ell}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) - \boldsymbol{\ell}(\boldsymbol{\theta}^{(t)}; \mathbf{X})\| < \epsilon$ or $\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\| < \epsilon\|\boldsymbol{\theta}^{(t+1)}\|$

As we can see, the algorithm is so intuitively appealing that it is no surprise that applications of it (not known as the EM algorithm) seem to have appeared as far back as in 1926. The generic EM algorithm, as we know it today, was made popular by Dempster, Laird, and Rubin (1977). It is an iterative two-stage algorithm that uses its Expectation step (E-step) to average the log-likelihood function over the distribution of the latent variables, then uses its Maximisation step (M-step) to find current maximum likelihood estimates of the expected log-likelihood. The EM algorithm starts from some arbitrary

guess of parameter estimates, and then keeps repeating the E-step and the M-step until convergence is attained.

1.10.1 Aspects and properties of the EM algorithm

Convergence and stationary values: One of the most appealing and central results of the EM algorithm is that the sequence $\{\boldsymbol{\theta}^{(t)}, t = 0, 1, 2, \cdots\}$ converges, at least to a local maximum.

Theorem 1.1 (Convergence and stability) *Since $\boldsymbol{\theta}^{(t+1)}$ is chosen so as to maximise $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$, $\boldsymbol{\theta}^{(t+1)}$ is therefore a better estimate than $\boldsymbol{\theta}^{(t)}$ in the sense that its observed-data log-likelihood is at least as high as that of $\boldsymbol{\theta}^{(t)}$. Successive iterations of the EM algorithm are therefore guaranteed never to decrease $\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X})$. In other words, for $t = 0, 1, 2, \cdots$, we always have*

$$\boldsymbol{\ell}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) \geq \boldsymbol{\ell}(\boldsymbol{\theta}^{(t)}; \mathbf{X}) \tag{1.12}$$

Elements of Proof: More details on the convergence properties of the EM sequence can be found in such references as Dempster, Laird, and Rubin (1977) and Wu (1983). For now, we simply present very general ideas used in the more detailed proof. In fact, $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ can be expressed as

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X}) + H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + \text{constant}, \tag{1.13}$$

where

$$H(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \int \log \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}) \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) d\boldsymbol{z}. \tag{1.14}$$

The difference $\boldsymbol{\ell}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) - \boldsymbol{\ell}(\boldsymbol{\theta}^{(t)}; \mathbf{X})$ can therefore be expressed as

$$\begin{aligned} \boldsymbol{\ell}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) - \boldsymbol{\ell}(\boldsymbol{\theta}^{(t)}; \mathbf{X}) &= Q(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) \\ &+ H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}). \end{aligned} \tag{1.15}$$

In equation (1.15), the quantity $Q(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)})$ is non-negative because $\boldsymbol{\theta}^{(t+1)}$ is by construction chosen such that

$$Q(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) - Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) \geq 0 \quad \forall \boldsymbol{\theta} \in \Theta. \tag{1.16}$$

As for the remainder of (1.15), it can be written as

$$H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) = \int \log \left[\frac{\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})}{\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t+1)})} \right] \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) d\boldsymbol{z}, \qquad (1.17)$$

which turns out to be the Kullback-Leibler divergence of $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ from $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t+1)})$. It therefore follows that $H(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}^{(t)}) - H(\boldsymbol{\theta}^{(t+1)}|\boldsymbol{\theta}^{(t)}) \geq 0$, by virtue of the non-negativity of the Kullback-Leibler divergence, and as a result we have $\boldsymbol{\ell}(\boldsymbol{\theta}^{(t+1)}; \mathbf{X}) - \boldsymbol{\ell}(\boldsymbol{\theta}^{(t)}; \mathbf{X}) \geq 0$. □

The stability[9] of the EM algorithm is one of its most attractive features and constitutes its greatest advantage over gradient methods like Newton-Raphson for which stability is not guaranteed.

Characteristics of estimates: For well-behaved problems, especially in cases where the observed-data likelihood function $\boldsymbol{L}(\boldsymbol{\theta}; \mathbf{X})$ is smooth, bounded from above, unimodal and log-concave over the entire parameter space $\boldsymbol{\Theta}$, the stationary (fixed) point yielded by the algorithm is a global maximum, and the EM therefore produces the unique maximum-likelihood estimate of $\boldsymbol{\theta}$ which is the maximiser of $\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X})$. However there are many cases in practice, such as the analysis of finite mixtures, where the likelihood function is unbounded and has many local maxima. In such ill-behaved problems, the EM does not necessarily converge to a unique global maximum, and in fact easily gets trapped into local maxima. There have been many extensions and variants of the EM algorithm aimed at circumventing this crucial issue. Ueda, Nakano, Ghahramani, and Hinton (2000)'s Split-and-Merge EM (SMEM) algorithm provides an alternative to the generic EM in the context of the analysis of finite mixture models.

Rate of convergence: The EM algorithm is often criticised for its slow convergence. This is due to the fact that its rate of convergence is only linear. Many variants of the EM exist that use various schemes to accelerate the convergence of the sequence to at least super-linear or even quadratic in some special cases. It must however be said that in some cases this apparent slow convergence is caused by the shape of the likelihood surface. In fact, if the likelihood surface is very flat, then successive values of $\boldsymbol{\theta}^{(t)}$ will

[9] By stability in this setting, we have in mind the monotonic convergence of the EM algorithm.

not appreciably increase the observed likelihood, even if the values $\boldsymbol{\theta}^{(t)}$ are significantly different. It is therefore good practice to monitor both successive values of $\boldsymbol{\theta}^{(t)}$ and the corresponding values of $\boldsymbol{\ell}(\boldsymbol{\theta}^{(t)}; \mathbf{X})$ to detect this type of problem.

Starting points: One of the main drawbacks of the EM algorithm is that its limiting position is often sensitive to initial guesses. In practice, an empirical heuristic solution to this problem is the use of many different starting values and to keep on trying until something "reasonable" appears.

1.10.2 Further aspects of the EM algorithm

Restricted EM: As we mentioned earlier, many of our models involve a large number of parameters. For such models efficient and meaningful parameters estimates can only be obtained if we restrict the model by imposing some constraints on the parameters. There has recently been some research on the use of the EM algorithm under restrictions Dong and Taylor (1995) on the parameter space.

Maximum A Posteriori via the EM: While the EM algorithm is most often used as a tool for computing maximum likelihood estimates, it can also be used as a Maximum A Posteriori (MAP) technique for the computation of posterior modes. In other words, instead of using the EM algorithm to find values of $\boldsymbol{\theta}$ that maximise the observed-data log-likelihood $\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X})$, the EM algorithm can be used to find values of $\boldsymbol{\theta}$ for which the observed-data posterior $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ is the highest. This is easily done by replacing the complete-data log-likelihood $\boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X}^*)$ by the the complete-data posterior $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X}^*) = \boldsymbol{L}(\boldsymbol{\theta}; \mathbf{X}^*)\boldsymbol{p}(\boldsymbol{\theta})$ at the E-step. Since $\log[\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X}^*)] = \boldsymbol{\ell}(\boldsymbol{\theta}; \mathbf{X}^*) + \log(\boldsymbol{p}(\boldsymbol{\theta}))$, it can be shown easily that the objective function to maximise at the M-step now becomes

$$Q^{\mathsf{MAP}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) + \log(\boldsymbol{p}(\boldsymbol{\theta}^{(t)})) \tag{1.18}$$

Aspects of the steps of the algorithm: While the Expectation and Maximisation steps of the EM algorithm generally allow the derivation of closed form expressions for the well-behaved regular exponential families of distributions, there are many applications in practice where this is not possible. In many settings, for instance, the E-step involves

the computation of high-dimensional integrals which may be intractable. The Stochastic EM algorithm is one of the variants that provides an attractive solution to this problem. In some cases, Monte Carlo EM is a good alternative to the generic EM. McLachlan and Krishnan (1997) provide a comprehensive coverage of the EM algorithm and its extensions.

1.11 Inference by Stochastic Simulation

While the EM algorithm is aimed at finding maximum likelihood estimates of the parameters by deterministic iterations, Data Augmentation, which is its probabilistic analogue, is used in the Bayesian framework to draw samples from the posterior distribution of parameters by stochastic simulation. Before presenting Data Augmentation in greater details, we first briefly introduce some general elements of Bayesian inference via Markov Chain Monte Carlo.

1.11.1 Bayesian inference via MCMC

The main ingredient for Bayesian inference is the posterior distribution $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ of the parameters. Unlike its deterministic likelihood-based counterparts like MLE that produce a single point estimate of the parameter of interest, the Bayesian approach yields the posterior density $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$, and inference is made by computing summary statistics of the form

$$\mathbb{E}\left[g(\boldsymbol{\theta})\right] = \int_{\Theta} g(\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}, \tag{1.19}$$

for some function g having its domain in Θ, and integrable with respect to $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$.

Note: In the majority of cases, it turns out that there is no closed-form analytical expression for the integral of equation (1.19). Approximations are therefore needed. There are many ways in practice of tackling the intractability of such integrals encountered in the Bayesian analysis of complex models. One that has been applied for many years is the use of asymptotic approximations. However, because such approximations are not

guaranteed to provide an accurate representation of the posterior distribution of interest, an alternative is to construct algorithms that can simulate the posterior distribution. In this thesis, we shall more precisely resort to Monte Carlo methods and Markov Chain Monte Carlo algorithms to make our stochastic simulation based inferences.

1.11.2 Monte Carlo approximation

In some very few simple applications, while the integral of (1.19) remains intractable, it is possible to fully specify $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ in closed-form, and therefore to *directly* sample from it, which allows one to produce a Monte Carlo estimate of (1.19), namely

$$\widehat{\mathbb{E}[g(\boldsymbol{\theta})]} = \frac{1}{T}\sum_{t=1}^{T} g(\boldsymbol{\theta}^{(t)}), \tag{1.20}$$

where $\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(T)}$ are i.i.d samples drawn from the distribution with density $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$. The most interesting (and important) result in the theory of Monte Carlo simulation is that $\frac{1}{T}\sum_{t=1}^{T} g(\boldsymbol{\theta}^{(t)})$ is an unbiased estimate, and by the strong law of large numbers, it converges almost surely (with probability 1) to $\int_{\Theta} g(\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}$. In other words,

$$\Pr\left(\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} g(\boldsymbol{\theta}^{(t)}) = \int_{\Theta} g(\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}\right) = 1 \tag{1.21}$$

1.11.3 Markov Chain Monte Carlo (MCMC)

In the analysis of latent structures, it is often the case that a closed-form expression for $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ does not exist, so that it is not possible to simulate it directly as in the above simple Monte Carlo case. This further complication constitutes one of the bottlenecks of the analysis of complex latent structures. Markov Chain Monte Carlo (MCMC) methods offer a vast body of algorithms that approach posterior intractability by a stochastic simulation of the posterior. Essentially, there are two main classes of MCMC algorithms: the Metropolis-Hastings algorithms and the Gibbs sampler.

Definition 1.1 *A Markov Chain Monte Carlo (MCMC) method for the simulation of a distribution $\boldsymbol{f}$ is any method producing an ergodic Markov chain $(\boldsymbol{\theta}^{(t)})$ whose stationary (equilibrium) distribution is $\boldsymbol{f}$.*

1.11.4 General properties of MCMC algorithms

For economy of notational space, let us assume that the distribution we want to sample from has density $\boldsymbol{p}(\boldsymbol{\theta})$. The key idea behind the MCMC body of algorithms can be described as follows: since we cannot sample directly from $\boldsymbol{p}(\boldsymbol{\theta})$, we iteratively construct a sequence of probability distributions having $\boldsymbol{p}(\boldsymbol{\theta})$ as its limit, so that draws from the converged sequence can be assumed to be draws from $\boldsymbol{p}(\boldsymbol{\theta})$. The Markovian property of the sequence is essential here, since we require the chain not to depend on its initial state, in such a way that the state of the chain at time $t+1$ only depends on the state of the chain at the previous time point t. The construction of such a stochastic sequence relies heavily on the specification of a transition kernel, $\boldsymbol{\mathcal{T}}$, say, that allows the chain to have the following two key properties:

- **Irreducibility:** The chain should be such that there is a positive probability of visiting all other states from any given state.
- **Aperiodicity:** The chain should be guaranteed not to get trapped in cycles.

A chain that is both *irreducible* and *aperiodic* is said to be *ergodic.* In practice, a sufficient, but not necessary condition that guarantees that $\boldsymbol{p}(\boldsymbol{\theta})$ is the desired invariant distribution is the so-called *detailed balance* or *reversibility* condition.

$$\boldsymbol{p}(\boldsymbol{\theta}')\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}|\boldsymbol{\theta}') = \boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}) \tag{1.22}$$

Intuitively, the reversibility (detailed balance) condition of (1.22) means that under the *target distribution* $\boldsymbol{p}(\boldsymbol{\theta})$*, the probability to go from state* $\boldsymbol{\theta}$ *to* $\boldsymbol{\theta}'$*, is exactly equal to the probability to go from state* $\boldsymbol{\theta}'$ *to* $\boldsymbol{\theta}$. All the MCMC samplers that we consider in our work throughout this thesis produce ergodic chains.

1.11.5 The Metropolis-Hastings algorithm

The Metropolis-Hastings (MH) algorithm is arguably the easiest to implement of all the MCMC algorithms. In fact, given an "appropriately" specified proposal distribution $\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta})$ and the target distribution with density $\boldsymbol{p}(\boldsymbol{\theta})$, the MH sampler moves the chain

from state $\boldsymbol{\theta}^{(t)}$ to state $\boldsymbol{\theta}'$ with acceptance probability $\boldsymbol{\alpha}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}') := \min\left(1, \frac{\boldsymbol{p}(\boldsymbol{\theta}')\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}')}{\boldsymbol{p}(\boldsymbol{\theta}^{(t)})\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})}\right)$, otherwise it remains in state $\boldsymbol{\theta}^{(t)}$. A very general description of the MH sampler is given below.

The Metropolis-Hastings Algorithm

Set $\boldsymbol{\theta}^{(0)} := \boldsymbol{\theta}_o$.
For $t = 0$ to $T - 1$
  Simulate $u \sim \mathcal{U}_{[0,1]}$
  Simulate $\boldsymbol{\theta}' \sim \boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})$
  Compute $\boldsymbol{\alpha}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}') := \min\left(1, \frac{\boldsymbol{p}(\boldsymbol{\theta}')\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}')}{\boldsymbol{p}(\boldsymbol{\theta}^{(t)})\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})}\right)$
  If $u < \boldsymbol{\alpha}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}')$ then
    $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}'$
  Else
    $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)}$
End.

While the MH sampler is easy and straightforward to implement, the choice of $\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta})$ can have a strong bearing on the performance of the sampler. For instance, if the proposal distribution $\boldsymbol{\mathcal{T}}(\boldsymbol{\theta}'|\boldsymbol{\theta})$ is too different from $\boldsymbol{p}(\boldsymbol{\theta})$, then the chain of interest might converge extremely slowly. More generally, it is important to find a proposal distribution such that transitions are not of very small size and occur relatively often.

1.11.6 The Gibbs sampler

The Gibbs sampler is the second most commonly used MCMC algorithm. It is particularly adapted to situations where it is possible to derive full conditional distributions $\boldsymbol{p}(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j})$, where $\boldsymbol{\theta}_{-j}$ is defined as $\boldsymbol{\theta}_{-j} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_j, \boldsymbol{\theta}_{j+1}, \cdots, \boldsymbol{\theta}_p)$. A very general description of the Gibbs sampler is given below.

There are well established and proven theorems Robert and Casella (2000) that show that the stationary distribution reached by the Gibbs sampler is indeed the target distribution $\boldsymbol{p}(\boldsymbol{\theta})$ of interest. In other words, if the chain has converged after T_o iterations, then $\boldsymbol{\theta}^{(t)} \sim \boldsymbol{p}(\boldsymbol{\theta}), \quad \forall t = T_o, \cdots, T$

The Gibbs sampler

Set $\boldsymbol{\theta}^{(0)} := (\theta_1^{(0)}, \cdots, \theta_p^{(0)})$.

For $t = 0$ to $T - 1$

Simulate $\theta_1^{(t+1)} \sim \boldsymbol{p}(\theta_1 | \theta_2^{(t)}, \theta_3^{(t)}, \cdots, \theta_p^{(t)})$

Simulate $\theta_2^{(t+1)} \sim \boldsymbol{p}(\theta_2 | \theta_1^{(t+1)}, \theta_3^{(t)}, \cdots, \theta_p^{(t)})$

$\vdots$

Simulate $\theta_j^{(t+1)} \sim \boldsymbol{p}(\theta_j | \theta_1^{(t+1)}, \cdots, \theta_{j-1}^{(t+1)}, \theta_{j+1}^{(t)}, \cdots, \theta_p^{(t)})$

$\vdots$

Simulate $\theta_p^{(t+1)} \sim \boldsymbol{p}(\theta_p | \theta_1^{(t+1)}, \theta_2^{(t+1)}, \cdots, \theta_{p-1}^{(t+1)})$

End.

1.11.7 Simulation by completion

Definition: Given a probability density f, a density g that satisfies

$$\int_{\mathcal{Z}} g(x, z) dz = f(x) \tag{1.23}$$

is called a *completion* of f. The density g is chosen so that its full conditionals are easy to sample from, and the Gibbs sampler is then applied on g instead of the original f.

1.11.8 The Data Augmentation algorithm

The Data Augmentation algorithm Tanner and Wong (1987) that we present in this section is also known as the *Two-stage Gibbs sampler.* It is essentially a special case of the Gibbs sampler. This idea of completion is the fundamental ingredient of the Data Augmentation algorithm. Since $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ is intractable and not easy to simulate, we instead use a completion of it that is more tractable in the sense that its conditionals are easy to simulate. We first remark that the observed-data posterior density $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ can be expressed as the marginal of $\boldsymbol{p}(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{X})$ as follows:

$$\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X}) = \int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{X}) dz \propto \int_{\mathcal{H}} \boldsymbol{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) \boldsymbol{p}(\boldsymbol{\theta}) d\mathbf{Z} \tag{1.24}$$

In other words, $\boldsymbol{p}(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{X})$ is a completion of $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$. The good news here is that the full conditional densities of $\boldsymbol{p}(\boldsymbol{\theta}, \mathbf{Z}|\mathbf{X})$, namely the predictive density of the latent

variables $p(\mathbf{Z}|\mathbf{X}, \theta)$ and the complete-data posterior $p(\theta|\mathbf{X}, \mathbf{Z})$, are easy to simulate for the majority of our latent structure models. Given the availability of these tractable conditionals, Data Augmentation is essentially the application of Gibbs sampling to $p(\theta, \mathbf{Z}|\mathbf{X})$. To put this in the more general context of missing data problems, let $V_{obs} \stackrel{\text{def}}{=} \boldsymbol{x}$ be our observed variable and let $V_{mis} \stackrel{\text{def}}{=} \boldsymbol{z}$ be our missing variable. Data Augmentation, also known as the *Imputation-Posterior* algorithm, is a two-step iterative process with each iteration alternating between (i) **Imputation:** drawing samples from $p(V_{mis}|V_{obs}, \theta)$ to "augment" (fill-in or complete) the data, and then (ii) **Posterior:** drawing new samples of parameters values from $p(\theta|V_{obs}, V_{mis})$ given the observed-data and filled-in values of the missing data. Below is a pseudo-code for the algorithm.

The Data Augmentation Algorithm

- Choose a value for $\theta^{(0)}$, then choose a number of iterations T and a burn-in number T_o.
- For $t = 1, \cdots, T$
  - **I-step** - Draw a value of the missing data from the conditional predictive distribution of V_{mis}. (Augmentation)

$$V_{mis}^{(t+1)} \sim p(V_{mis}|V_{obs}, \theta^{(t)}) \tag{1.25}$$

  - **P-step** - Then, conditioning on $V_{mis}^{(t+1)}$, draw a new set of parameters θ from its complete-data posterior.

$$\theta^{(t+1)} \sim p(\theta|V_{obs}, V_{mis}^{(t+1)}) \tag{1.26}$$

- Extract $\theta^{(t)}, t = T_o + 1, \cdots, T$ from the final Markov chain.

As we can see from the above algorithm, the basic idea behind the Data Augmentation algorithm is just as intuitively appealing as the one central to the EM algorithm. In fact, the I-step of equation (1.25) corresponds to *imputing* a value of the missing data V_{mis}, hence the term **Imputation**, and the P-step of equation (1.26) corresponds to drawing

a value of $\boldsymbol{\theta}$ from a complete-data posterior, and hence the term **Posterior.**

In fact, repeating steps (1.25)-(1.26) iteratively from a starting value $\boldsymbol{\theta}^{(0)}$ yields a stochastic sequence $\{(\boldsymbol{\theta}^{(t)}, V_{mis}^{(t)}) : t = 1, 2, \cdots\}$ which is a *Markov chain* that, under mild regularity conditions, has a stationary distribution with density $\boldsymbol{p}(\boldsymbol{\theta}, V_{mis}|V_{obs})$. Such an iterative algorithm clearly describes the steps of a typical **Gibbs sampler** where the joint distribution $\boldsymbol{p}(\boldsymbol{\theta}, V_{mis}|V_{obs})$ of the pair $(\boldsymbol{\theta}, V_{mis})$ is 'simulated' iteratively through its conditionals $\boldsymbol{p}(V_{mis}|\boldsymbol{\theta}, V_{obs})$ and $\boldsymbol{p}(\boldsymbol{\theta}|V_{mis}, V_{obs})$ until the successive draws converge to draws from $\boldsymbol{p}(\boldsymbol{\theta}, V_{mis}|V_{obs})$. The very good news is that the sequence $\{\boldsymbol{\theta}^{(t)} : t = 1, 2, \cdots\}$ has $\boldsymbol{p}(\boldsymbol{\theta}|V_{obs})$ as its stationary distribution, which is exactly what we want. By the same token, the sequence $\{V_{mis}^{(t)} : t = 1, 2, \cdots\}$ has $\boldsymbol{p}(V_{mis}|V_{obs})$ as its stationary distribution.

It is pretty straightforward to realise that the IP algorithm bears a striking resemblance to the EM algorithm, and it would be right to say that *The Data Augmentation algorithm is to the Bayesian what the EM algorithm is to the frequentist when it comes to the study of incomplete-data problems.*

1.11.9 Aspects and properties of Data Augmentation

A convergence theorem: One of the stimulating features of this approach is the fact that it is proven theoretically Robert and Casella (2000) that, if we assume the chain $\{(\boldsymbol{\theta}^{(t)}, V_{mis}^{(t)}) : t = 1, 2, \cdots\}$ produced by the Data Augmentation algorithm (Two-stage Gibbs sampler) to be ergodic, then $\{\boldsymbol{\theta}^{(t)} : t = 1, 2, \cdots\}$ has $\boldsymbol{p}(\boldsymbol{\theta}|V_{obs})$ as its stationary distribution.

General advantages of stochastic simulations: As a stochastic simulation method, Data Augmentation, unlike methods based on asymptotic approximations, offers many advantages: (a) for the types of complex model that we consider, it is often conceptually and computationally easier to implement than other methods; (b) instead of exploring only an approximation, in theory it explores the entire posterior distribution, and it converges stochastically to the true (exact) posterior distribution of interest, regardless of the sample size and the complexity of the problem; (c) samples drawn from the posterior distribution are available and can be used for various inferential tasks.

Convergence: This is one of the most difficult aspects of many, if not all, MCMC algorithms. Unlike deterministic schemes where an objective function to be optimised offers a way to assess the convergence, the assessment of the convergence of MCMC sequences still remains an open and difficult problem. There are many ad hoc ways in practice to tackle this crucial issue, but many still remain mostly empirical. Besides the difficulty to monitor and assess the convergence of MCMC sequences, there is the crucial issue of rate of convergence. In fact, one of the main drawbacks of MCMC schemes is that to date they are in most cases slower than their competitors, and Data Augmentation, as a special case of Gibbs sampling, is even worse than Metropolis-Hastings type algorithms in this context. Throughout this thesis, we have used various ways to address these two aspects of convergence.

Mixing: While the majority of our sampling schemes are theoretically irreducible and aperiodic and therefore ergodic, it is very common in practice to notice very poor mixing of the chains, especially with variants of the Gibbs sampler like the Data Augmentation algorithm. Whereas this poor mixing can be harmless in some cases, it is a serious issue when the interest is in density estimation, since one would like in such cases to explore the posterior surface as exhaustively as possible. In some of our analyses, we have addressed this important issue using methods available in the literature, and adapting them accordingly.

Storage of sample paths: Whereas the availability of sample paths offers the advantage of allowing a variety of inferential tasks to be performed without extra computation, it is fair to point out that this also constitutes one of the serious problems of MCMC methods. In fact, as the size of the problem grows, having to store these sample paths can quickly become a serious bottleneck. In many of our high-dimensional latent variable models, we had to resort to a variety of methods to deal with this. In some cases, we had to resort to on-line methods consisting of computing all our desired summaries each time a new sample is drawn and therefore avoiding storage. In some cases, we simply had to store only sample paths of the parameters and use the latent variable draws just as instruments. In such cases, our random draws of latent variables are simply used to

complete the data so as to draw new values of parameters. Only parameters are stored for future inferential tasks, and we use closed-form expressions such as (1.11) to estimate latent scores once parameter estimates are obtained.

1.12 Variational Approximation

In the Bayesian paradigm, the intractability of $p(\boldsymbol{\theta}|\mathbf{X})$ is also tackled through the use of variational approximations. Although we do not use such an approach in our work, it is useful to give a brief description of what it is.

The key idea behind variational inference can be expressed as follows: to obtain the true posterior density $p(\boldsymbol{\theta}|\mathbf{X})$, we need to normalise $p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})$. When the computation of the normalising constant is intractable, one can construct $q(\mathbf{X}|\boldsymbol{\theta}, \Upsilon)$ such that the bound

$$p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \geq q(\mathbf{X}|\boldsymbol{\theta}, \Upsilon)p(\boldsymbol{\theta}), \tag{1.27}$$

is as tight as possible, and then normalise the variational approximation $q(\mathbf{X}|\boldsymbol{\theta}, \Upsilon)p(\boldsymbol{\theta})$ to form a proper posterior density function $q(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)$ that makes the computation of averages tractable. With a Gaussian prior $p(\boldsymbol{\theta})$ and the choice of a Gaussian variational form for $q(\mathbf{X}|\boldsymbol{\theta}, \Upsilon)$, the normalised variational distribution is also Gaussian.

Another important aspect of this method is that the normalised variational posterior approximation $q(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)$ depends on the variational parameter Υ. For the method to be complete, one therefore has to specify or determine Υ. This is generally done via an optimisation procedure that finds a value of Υ that yields that tightest lower bound in equation (1.27). Amongst the different approaches to finding the best variational parameter Υ, there is the use of the Kullback-Leibler divergence as the objective function for the optimisation procedure. The objective is to find a tractable approximation to $p(\boldsymbol{\theta}|\mathbf{X})$, say $q(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)$, such that the divergence (1.28) of $q(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)$ from $p(\boldsymbol{\theta}|\mathbf{X})$ is as small as possible.

$$KL(q||p) = \int \log\left[\frac{q(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)}{p(\boldsymbol{\theta}|\mathbf{X})}\right] q(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)d\boldsymbol{\theta} \tag{1.28}$$

As we can see, the idea is very intuitively appealing, and has been successfully applied to many problems in Statistical Physics, Neural Networks Fokoué (1998),Ghahramani and Beal (2000), Machine Learning and Information Theory, just to name a few.
Once the approximating distribution is fully specified, approximate estimates of the summary statistics of interest are then easily obtained using $\boldsymbol{q}(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)$ in place of $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$. More specifically, the integral of (1.19) is approximated by

$$\widehat{\mathbb{E}[g(\boldsymbol{\theta})]} = \int_{\Theta} g(\boldsymbol{\theta}) \boldsymbol{q}(\boldsymbol{\theta}|\mathbf{X}, \Upsilon^{(opt)}) d\boldsymbol{\theta}, \tag{1.29}$$

where the form of $\boldsymbol{q}(\boldsymbol{\theta}|\mathbf{X}, \Upsilon)$ is chosen so as to allow an analytical expression for (1.29). As we said earlier, approximations often produce results faster than stochastic simulations, but they suffer from three main drawbacks: (a) finding a suitable approximating distribution often requires a lot of mathematical sophistication, and many commonly used approximating schemes like mean field approximations make assumptions that can be mathematically very convenient, but that could well be unrealistic and unreasonable; (b) unlike MCMC methods that explore the true posterior, these methods rely on the exploration of some approximation of it, with no guarantee of covering the posterior of interest itself; (c) while many objective functions exist to assess the divergence of the approximating distribution from the true distribution, only lower bounds are usually computable, and it is therefore hard to measure how close the approximation gets to the truth.

1.13 Model selection

Determining the dimension of the minimal latent space is one of the most fundamental issues in latent structures analysis. In fact, it is the very starting point of the analysis, and nothing else can be done until this dimension is either estimated or obtained from experience or exploratory methods. It must however be stressed that model selection in latent structures analysis is an extremely difficult problem, made even more difficult by the weak identifiability of the models, the arbitrary nature of the probability densities used to represent the marginal density of the manifest variables, and the subjective

definition of what a latent dimension should be. In data reduction, for instance, it is not always easy to find a clear-cut and objective way of deciding on what makes a latent factor distinct. The same type of problem arises in finite mixture modelling where deciding on what is a distinct component can become a very subjective decision. However, despite these considerations that can sometimes turn rather philosophical, there are many approaches to model selection that have yielded satisfactory results. Again, we distinguish the frequentist approach from the Bayesian treatment. In this thesis, our approach to model selection is essentially Bayesian, and more specifically based on stochastic simulation.

The Reversible Jump MCMC (RJMCMC) algorithm Green (1995), which is a generalisation of the Metropolis-Hastings algorithm to parameter spaces of varying dimensions, is a model selection algorithm based on posterior simulation. RJMCMC has been extensively applied to some of the models of interest to us. It turns out that, for the models considered in our work, the dimensions of the latent spaces can be treated as points in a point process, thereby allowing us to approach our posterior simulation as the simulation of a point process. That is why many of the ideas that we will use in this context are borrowed from stochastic geometry and spatial statistics Barndorff-Nielsen, Kendall, and van Lieshout (1999), Stoyan, Kendall, and Mecke (1995), where they have been applied successfully to a wide range of problems. To be more specific, we will adopt an approach based on the simulation of a continuous-time birth-and-death process with the distribution of all the parameters (including the dimension) as its limiting distribution, in the spirit of Stephens (2000) who constructed a Birth-and-Death MCMC algorithm for model selection for finite mixtures.

Chapter 2

Elements of Factor Analysis

Science is the attempt to make the chaotic diversity of our sense-experience correspond to a logically uniform system of thought
Albert Einstein

The factor analysis model is arguably the oldest of all the latent variable models that we will be considering throughout this thesis. To the best of our knowledge, the first development of factor analysis was due to Charles Spearman who, while studying the correlations between test scores, noted that many observed correlations could be accounted for by a simple model Spearman (1904). In this chapter, we study the factor analysis model from both the likelihood-based and Bayesian perspectives. We review the EM algorithm for factor analysis, and we explore the possibility of a restricted EM extension for interpretability and efficiency of parameter estimation. We also present a detailed Bayesian treatment of the model from a stochastic simulation perspective. A new stochastic simulation method for model selection is derived and applied to both synthetic and real-life data. The chapter also touches on ideas for a new Bayesian sampling alternative to varimax factor rotation for interpretability and derivation of simple structures.

2.1 Introduction

The main goal of factor analysis (FA) is to describe the covariance relationships among many variables in terms of fewer underlying latent (unobservable) constructs represented

by random quantities known as *factors*. Factor analysis is therefore a *data reduction or dimensionality reduction* technique, since the number of factors is always assumed to be far less than the number of originally observed variables. FA is one of the most popular techniques for dimensionality reduction, and can be considered as an extension of principal component analysis[1] (PCA) Jolliffe (1986). While FA and PCA both attempt to approximate the structured covariance matrix, the approximation based on FA is clearly more elaborate. Besides data reduction, FA is also used as a way of providing an *interpretation* of the covariation among the observed variables, although it must be said that such interpretations can be very subjective as they depend on each experimenter. FA was originally developed by psychometricians whose aim was to quantify and possibly explain unobservable (not directly measurable) concepts like intelligence and physical fitness by modelling their relationship with such observable (and therefore measurable) quantities as test scores in various disciplines. While data reduction still remains important in such a context, the ability to produce a simple and easy to interpret structure is obviously far more important here. In general, social scientists and psychologists need to come up with an interpretation of their factor analysis results.

FA has been extensively used in the Neural Networks, Machine Learning and Artificial Intelligence communities as a probabilistic model for unsupervised learning in the context of statistical pattern recognition. In such a context, FA is a feature extraction technique essentially used to preprocess the data in order to obtain features or special characteristics of observed quantities. If we consider the handwritten digits recognition task for instance, we can easily conceive that a 64-dimensional (digitised 8×8 picture) vector representing one single observed character can be reduced to a far lower-dimensional internal vector since the character itself occupies only a tiny portion of the space allocated for writing. In contexts like pattern recognition, FA is just a means to an end, not the end itself, since its results (estimated factor scores) are then used for other tasks such as classification, clustering, density estimation or even regression, to name just a few. It is worth mentioning that FA is not used for interpretation in pattern recognition, since no

[1]This method is sometimes preferred over FA because of its simplicity and the fact that it does not assume any model.

particular meaning needs to be attached to the estimated factor scores in such a context. We see from the above that FA is motivated by problems of great interest to both social scientists and physical scientists. The former carefully and purposefully design the study and therefore select a set of variables that have a particular meaning to them. For such scientists, it would be natural to also think of assigning meaning to the common factors. However, for a modeller in pattern recognition solely interested in the reduced dimension of an image, for instance, the high-dimensional observed vector of image characteristics does not have any particular meaning, and therefore such an experimenter will not seek any interpretation whatsoever of the estimated factor scores or factor loadings.

Whatever the objective, assuming that there are fewer factors than there are observed variables implies that the fundamental structure of the data is no more complicated than that of the observed variables, and the primary question in factor analysis is whether the data are consistent with the postulated structure.

Many issues of interest in FA modelling have been extensively studied over the years from a purely frequentist perspective. There has been a reasonable amount of research from a Bayesian perspective recently. It must however be said that, apart from Lopes and West (1999) and a couple of other authors, Bayesian analysis of the FA model from a stochastic simulation perspective has not benefited from enough attention. In this chapter, we aim at providing a coverage of that alternative perspective of Bayesian FA. Section 2 of the current chapter provides an introduction to the orthogonal factor model, together with the main issues of interest in factor analysis, the inferential difficulties inherent to the structure of the FA model. Elements of solution to the most important difficulties are also presented. The section concludes with some ingredients for parameter estimation. Section 3 re-examines the EM algorithm for FA, while section 4 mainly deals with Bayesian parameter estimation from a stochastic simulation perspective. The appropriateness of Bayesian sampling is clearly presented, and efficient Markov Chain Monte Carlo (MCMC) algorithms are derived for both the identified and the non-identified versions of the model, with the use of suitably chosen loss functions as a way to deal with rotation invariance. Elements of model goodness-of-fit are pre-

sented in the Bayesian framework. Section 5 is dedicated to model selection. A brief summary of the most popular methods used to determine (estimate) the number of common factors is given, but the main concentration is on the construction of an ergodic Markov chain used to simulate a continuous time birth-and-death point process having the posterior distribution of the number of common factors and the other parameters as its equilibrium distribution. Section 6 presents examples and applications.

2.2 The Orthogonal Factor model

The factor analysis (FA) model assumes that a p-dimensional manifest random vector $\boldsymbol{x} \in \mathbb{R}^p$ is made up of highly correlated variables that can be grouped by their correlations, with variables within a particular group being highly correlated among themselves, but having relatively small correlations with variables belonging to a different group. With such an assumption, each group of variables can be thought of as the representation of a single underlying construct also known as a *factor* or more precisely a *common factor*, that is responsible for the observed correlations. The factor analysis model postulates that $\boldsymbol{x}$ is a linear combination of $q < p$ latent random variables $z_1, z_2, \cdots, z_q$, called *common factors*, plus p additional sources of variation $e_1, e_2, \cdots, e_p$ referred to as *errors* or *disturbances* or even *noise*. These additional sources of variation are sometimes called *specific factors* as opposed to the above *common factors*, since each e_i is specifically associated only with its corresponding observed variable x_i. We recognise here all the ingredients of a typical latent variable model for data reduction as defined in the previous chapter.

2.2.1 Probabilistic construction of the FA model

In a typical setting, all we have is a sample of i.i.d. observations, and the starting point of the statistical analysis is to assume a probability distribution for each of the observations. The FA model traditionally assumes that our p-dimensional manifest random vector $\boldsymbol{x}$ has a Normal distribution with mean μ and covariance matrix $\boldsymbol{\Omega}$. Our interest being in

data reduction, the next fundamental step in our analysis is to find q, $\boldsymbol{p}(\boldsymbol{z})$ and $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z})$ such that the distribution of $\boldsymbol{x}$ can admit a representation of the form (1.5). One such representation is derived using results from distribution theory. In fact, if we assume

$$\boldsymbol{z} \sim \mathcal{N}_q(0, \mathbf{I}_q) \quad \text{and} \quad [\boldsymbol{x}|\boldsymbol{z}] \sim \mathcal{N}_p(\boldsymbol{x}; \mu + \Lambda\boldsymbol{z}, \Sigma), \tag{2.1}$$

then it is straightforward to show that the marginal density $\boldsymbol{p}(\boldsymbol{x})$ of $\boldsymbol{x}$ is Gaussian with mean μ and covariance matrix $\boldsymbol{\Omega} = \Lambda\Lambda^{\mathsf{T}} + \Sigma$, which is what we required. From now on, we use $\boldsymbol{p}(\boldsymbol{x}) = \mathcal{N}_p(\boldsymbol{x}; \mu, \Lambda\Lambda^{\mathsf{T}} + \Sigma)$ and $\boldsymbol{x} \sim \mathcal{N}_p(\mu, \Lambda\Lambda^{\mathsf{T}} + \Sigma)$ interchangeably. In (2.1), Σ is assumed to be a diagonal matrix. This assumption is a very crucial and fundamental one, since it satisfies the axiom of conditional independence stated in the previous chapter. The conditional mean $\mathbb{E}[\boldsymbol{x}|\boldsymbol{z}] = \mu + \Lambda\boldsymbol{z}$ of $\boldsymbol{x}$ given $\boldsymbol{z}$ conveys the fact that the manifest variable $\boldsymbol{x}$ is a linear function of the latent variable $\boldsymbol{z}$. With all the above developments, the generative equation of the model can therefore be expressed as:

$$\boldsymbol{x} - \mu = \Lambda\boldsymbol{z} + \boldsymbol{e} \quad \text{or} \quad \boldsymbol{x} = \Lambda\boldsymbol{z} + \mu + \boldsymbol{e}, \tag{2.2}$$

which can be written in a more detailed form as

$$\left[\begin{array}{rcccccccc} x_1 - \mu_1 & = & \lambda_{11}z_1 & + & \lambda_{12}z_2 & + & \cdots & + & \lambda_{1q}z_q & + & e_1 \\ x_2 - \mu_2 & = & \lambda_{21}z_1 & + & \lambda_{22}z_2 & + & \cdots & + & \lambda_{2q}z_q & + & e_2 \\ \vdots & & & & & & \vdots & & & & \\ x_p - \mu_p & = & \lambda_{p1}z_1 & + & \lambda_{p2}z_2 & + & \cdots & + & \lambda_{pq}z_q & + & e_p \end{array}\right. \tag{2.3}$$

In matrix-vector form, equation (2.3) becomes

$$\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \\ \vdots \\ \vdots \\ x_p - \mu_p \end{pmatrix} = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \cdots & \lambda_{1q} \\ \lambda_{21} & \lambda_{22} & \cdots & \lambda_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{p1} & \lambda_{p2} & \cdots & \lambda_{pq} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_q \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ \vdots \\ e_p \end{pmatrix} \tag{2.4}$$

The coefficient λ_{ij} is called the *loading* of the ith variable on the jth factor, and the matrix $\Lambda \in \mathbb{R}^{p\times q}$ is referred to as the *matrix of factor loadings*. $\mu \in \mathbb{R}^p$ is the marginal mean of $\boldsymbol{x}$, and $\boldsymbol{e} \in \mathbb{R}^p$ is the independent disturbance vector. The FA model further assumes

that $\boldsymbol{e}$ and $\boldsymbol{z}$ are independent, so that with $\boldsymbol{e} \sim \mathcal{N}_p(0, \Sigma)$, where $\Sigma = \mathsf{diag}(\sigma_1^2, \cdots, \sigma_p^2)$, it is easy to see that $\mathsf{cov}(\boldsymbol{x}, \boldsymbol{z}) = \Lambda$, and that $\mathsf{cov}(\boldsymbol{e}, \boldsymbol{z}) = \mathbb{E}\left[\boldsymbol{e}\boldsymbol{z}^\mathsf{T}\right] = 0$. With $\boldsymbol{x} \sim \mathcal{N}_p(\mu, \Lambda\Lambda^\mathsf{T} + \Sigma)$, the orthogonal factor model implies that $\boldsymbol{x}$ can be simply seen as multivariate Gaussian random vector with a structured covariance matrix. This structure of the covariance matrix of $\boldsymbol{x}$ allows us to write,

$$\mathbb{V}[x_i] = \lambda_{i1}^2 + \lambda_{i2}^2 + \cdots + \lambda_{iq}^2 + \sigma_i^2 = h_i^2 + \sigma_i^2 \tag{2.5}$$

where $h_i^2 = \lambda_{i1}^2 + \lambda_{i2}^2 + \cdots + \lambda_{iq}^2$ represents the portion of the variance of x_i contributed by the q common factors, while σ_i^2 is contributed by the specific factor. h_i^2 is called the ith *communality*, and σ_i^2 is known as the *uniqueness* or *specific variance.*

Note: In some real life applications of factor analysis, especially in such fields as sociology, psychology, marketing research, psychometrics and education, many problems give rise to common factors that are *correlated.* In such cases, $\mathsf{cov}(\boldsymbol{z})$ is no longer diagonal, and the corresponding model is known as an *oblique factor model.* Press (1972) gives the following example to illustrate the idea: *suppose the observable vectors represent the socioeconomic characteristics of buyers of a certain type of automobile. The latent factors, though different from one another, probably all depend in some complicated way upon the utility function of the buyer. Therefore, it is quite likely that the factor structure is composed of mutually correlated factors.* One could rightly argue that the complexity of the oblique factor model fails to achieve the aim of factor analysis which is to derive an elemental or simplest structure, and the argument could go as far as imagining a further factor analytic step on the results of oblique FA aimed at reaching the simplest possible structure. Throughout this thesis, we shall assume an orthogonal factor model. The reader is referred to Manly (1986) and Johnson and Wichern (1998) for a simple introduction to the orthogonal factor model. Press (1972), Everitt (1984), Anderson (1984) and Bartholomew (1987) present a more general coverage of the topic.

2.2.2 Issues of interest in factor analysis

Essentially, there are three mains goals in FA. **Model selection** consists of the determination or estimation of the adequate number of factors that can be used to represent the original manifest variables with as little loss of information as possible; mathematically speaking, this means finding the intrinsic dimensionality q of the data or the dimension of the minimal latent subspace, such that the distribution of $\boldsymbol{x}$ can have the form (1.5). **Parameter estimation** consists of estimating the parameters (especially the factor loadings) of the postulated model in order to interpret and characterise the covariance (association) structure of the manifest variables. **Prediction** has to do with *estimating factor scores* for future unseen observations for such purposes as data-reduction. Estimated factor scores can be used in image compression to store high-dimensional images that are later reconstructed. Estimated factor scores can also be used for data visualisation in the plane (2-factor model) to explore group structures in the observed population of interest. Psychometricians, sociologists and educationalists are generally interested in factor loadings as a way to explain or at least interpret the associations (correlations) amongst some designed variables (tests grades, monthly expenses, etc.) and their relationships to some hypothesised latent (not directly measurable) concepts like intelligence, social class or aptitude.

However, the estimation of factor scores requires a set of model parameters, which in turn requires some knowledge (estimate) of the number of factors. Estimating the number of factors is therefore central, and we address it later. For now, we assume the number of factors to be known and fixed, and we focus our attention on the other issues.

2.2.3 Identifiability, Unique Solution

Parameter estimation presupposes the existence of a unique set of parameters that characterise the proposed model, so that the objective of the estimation task is to determine that unique set of parameters [2]. Unfortunately, as we shall see, our model as specified

[2] In some applications, the meanings of parameters are not relevant. In such cases, FA is just a means to an end, so that all that is needed is any set of factor scores providing a valid reduced representation

by equation (2.2) is indeterminate (unidentifiable), and does not provide a unique set of parameters, but a multiplicity of parameter sets, each related to the other by an orthogonal transformation.

In fact, Λ has pq free parameters (factor loadings). The diagonal matrix Σ has p free parameters (specific variances). We therefore have $p(q+1)$ variance-covariance parameters to be estimated. Now, given a sample of observations $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, our objective is to use the $\frac{1}{2}p(p+1)$ items of information provided by the sample covariance matrix to estimate our $p(q+1)$ unknown free parameters. As we see, in most cases, we will have $p(q+1) > \frac{1}{2}p(p+1)$, and the sample will therefore not provide enough information to allow the estimation of a unique set of Λ and Σ. In fact, the FA model is inherently a non-identified model: for a given set of data, there exists an infinity of orthogonal transformations of the matrix of factor loadings that would produce the same covariance structure. To see this more clearly, let us assume $q > 1$, and let Γ be any $q \times q$ orthogonal matrix, so that $\Gamma\Gamma^\mathsf{T} = \Gamma^\mathsf{T}\Gamma = \mathbf{I}_q$. Then the expression in equation (2.2) can be written

$$\boldsymbol{x} - \mu = \Lambda\boldsymbol{z} + \boldsymbol{e} = \Lambda\Gamma\Gamma^\mathsf{T}\boldsymbol{z} + \boldsymbol{e} = \Lambda^*\boldsymbol{z}^* + \boldsymbol{e}, \tag{2.6}$$

where $\Lambda^* = \Lambda\Gamma$ and $\boldsymbol{z}^* = \Gamma^\mathsf{T}\boldsymbol{z}$. It is easy to see that $\mathbb{E}[\boldsymbol{z}^*] = \Gamma^\mathsf{T}\mathbb{E}[\boldsymbol{z}] = 0$, and that $\mathsf{cov}[\boldsymbol{z}^*] = \Gamma^\mathsf{T}\mathsf{cov}[\boldsymbol{z}]\Gamma = \Gamma^\mathsf{T}\Gamma = \mathbf{I}_q$. In other words, the factors $\boldsymbol{z}$ and $\boldsymbol{z}^* = \Gamma^\mathsf{T}\boldsymbol{z}$ have the same statistical properties. Looking at equations (2.2) and (2.6), it is therefore impossible, on the basis of observations on $\boldsymbol{x}$, to distinguish the matrices of factor loadings Λ and Λ^*. Moreover, $\boldsymbol{\Omega} = \Lambda\Lambda^\mathsf{T} + \Sigma = \Lambda\Gamma\Gamma^\mathsf{T}\Lambda^\mathsf{T} + \Sigma = (\Lambda^*)(\Lambda^*)^\mathsf{T} + \Sigma$, which means, that although different in general, Λ and Λ^* both generate the same covariance matrix $\boldsymbol{\Omega}$, and therefore the same representation of the data.

Geometrically speaking, the columns of Λ can be viewed as defining the axes of the lower-dimensional latent space (coordinate system) of factors. Since a rotation is a non-singular orthogonal transformation, and a permutation of columns is particular type of rotation, we say that a factor solution is invariant to permutations of axes. This feature will be useful later when we address the estimation of the number of factors. In practice, a unique solution is guaranteed by imposing some constraints on Λ so that the only valid

of the manifest vector. In such situations, one need not worry about identifiability.

solution is the one that satisfies the constraints.

In order to achieve efficient estimation of parameters, constraints are imposed in such a way that the number of parameters to be estimated is at most equal to the number of items of information provided by the sample. Traditionally, there are two types of constraint that are equivalent:

1. Constrain Λ to be such that $\Lambda^T\Sigma^{-1}\Lambda$ is diagonal. Since $\Lambda^T\Sigma^{-1}\Lambda \in I\!R^{q\times q}$ is symmetric and diagonal, $\frac{1}{2}q(q-1)$ of its elements are all zeros. This means that $\frac{1}{2}q(q-1)$ elements do not need to be estimated by the parameter estimation procedure. This approach is used when estimation is done via a deterministic optimisation algorithm.

2. A second approach equivalent to the above consists of preassigning values to some entries of Λ as in equation (2.7). This particular lower diagonal form[3] of Λ reduces the number of parameters to be estimated by $\frac{1}{2}q(q-1)$ as above. This is the form of constraints that we use in the Bayesian sampling framework, since its application is straightforward.

$$\Lambda = \begin{pmatrix} \lambda_{11} & 0 & 0 & \cdots & 0 & 0 \\ \lambda_{21} & \lambda_{22} & 0 & \cdots & 0 & 0 \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{q-1,1} & \lambda_{q-1,2} & \lambda_{q-1,3} & \cdots & \lambda_{q-1,q-1} & 0 \\ \lambda_{q,1} & \lambda_{q,2} & \lambda_{q,3} & \cdots & \lambda_{q,q-1} & \lambda_{q,q} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \lambda_{p,1} & \lambda_{p,2} & \lambda_{p,3} & \cdots & \lambda_{p,q-1} & \lambda_{p,q} \end{pmatrix} \tag{2.7}$$

In fact, to guarantee a unique solution under our constraints, all we need is to determine q such that $p(q+1) - \frac{1}{2}q(q-1) \leq \frac{1}{2}p(p+1)$, which means

$$(p+q) \leq (p-q)^2. \tag{2.8}$$

[3]We assume Λ to be full rank, so we constrain its "diagonal" elements to be nonzero.

From equation (2.8) an upper bound on the number of factors that can be included in a model is given by

$$q \leq \frac{1}{2}(2p + 1 - \sqrt{8p+1}). \tag{2.9}$$

Note: It must be said that there are situations where solutions satisfying constraint (2.8) might not provide an adequate fit for the data. In fact, given a data set, a fundamental question (without an obvious answer) is whether there exists a matrix of factor loadings Λ such that the model in equation (2.2) adequately fits the data. An exploration of this issue and many other related topics of FA can be found in such references as Bartholomew (1987), Everitt (1984) and Press (1972) amongst others.

2.2.4 Rotation and Interpretability

We saw earlier that applying any orthogonal transformation to the matrix of factor loadings would yield a new matrix that would provide exactly the same representation of the data as the non-transformed matrix. When interpretation is one of the main aims of FA, deriving a simple structure therefore becomes very important. In such cases, it is desirable, although not always possible, to derive a structure that would form the grouping of observed variables into factors in a way that is as straightforward as possible. This can be achieved for instance if the entries of the estimated matrix of factor loadings differ appreciably (Kaiser's varimax) amongst themselves, thereby making it easy and straightforward to allocate a given variable to the particular group (factor) for which its loading is very high. In this thesis, we do not use rotation to find simpler structure.

2.3 Elements of parameter estimation

If we assume that identifiability is dealt with and that the intrinsic dimensionality q is known and fixed, then parameter estimation can be carried out. For the FA model, there are many ways to approach the topic of parameter estimation, but we will focus on Maximum Likelihood via the EM algorithm and Bayesian Estimation by stochastic

simulation. The complete collection $\boldsymbol{\theta}$ of parameters for the FA model is $\boldsymbol{\theta} = \{\mu, \Lambda, \Sigma\}$. In reality, the parameter that is of great importance here is Λ, the matrix of factor loadings, since the main interest is the characterisation of the covariance structure.

2.3.1 Effect of scale in estimation

It is a well known fact that, in the analysis of the relationship between two variables, correlation is often easier and more straightforward to interpret than covariance, since correlation is scale-invariant, and easily conveys the strength of the relationship as a proportion. This fact explains in part the widespread use of the sample correlation matrix instead of the sample covariance matrix as the main ingredient for fitting the FA model. Before we embark on estimation, we first explore some aspects of this scale-invariant estimation issue in this section. The need for scale-invariant estimation is often justified by the fact that the units of measurement of the manifest variables can be arbitrary. Two approaches are generally used to tackle this problem.

Preprocessing approach. The first approach consists of basing the analysis on the correlation matrix instead of the covariance matrix. This is equivalent to standardising the manifest variables to ensure that changes in scale have no effect on the analysis. In practice, this means that the data actually used for the analysis are $\tilde{\boldsymbol{x}} = C\boldsymbol{x}$, where $C = [\mathsf{diag}(S)]^{-\frac{1}{2}}$, and $\mathsf{diag}(S)$ is a diagonal matrix made up of the diagonal elements of S, the sample covariance matrix. Theoretically, such a preprocessing of the data, while intuitively appealing, leads to inconsistencies, especially when methods based on the likelihood are used; in fact, the distribution of the correlation matrix is not the same as that of the covariance matrix, which means that the results obtained after the transformation might well not reflect the truth about the data. Fortunately, Krane and McDonald (1978) have shown that estimates obtained this way are the maximum likelihood estimates of the scale-invariant parameters.

Post-processing. A second approach with a more convincing theoretical foundation consists of first estimating scale-dependent parameters $\hat{\Lambda}$ and $\hat{\Sigma}$ using the sample covari-

ance matrix, then transforming them into scale-invariant estimates $\hat{\Lambda}^*$ and $\hat{\Sigma}^*$ as

$$\hat{\Lambda}^* = [\mathsf{diag}(\hat{\Omega})]^{-\frac{1}{2}}\hat{\Lambda} \quad \text{and} \quad \hat{\Sigma}^* = [\mathsf{diag}(\hat{\Omega})]^{-\frac{1}{2}}\hat{\Sigma}[\mathsf{diag}(\hat{\Omega})]^{-\frac{1}{2}}, \tag{2.10}$$

where $\mathsf{diag}(\hat{\Omega})$ is a diagonal matrix made up of the diagonal elements of $\hat{\Omega} = \hat{\Lambda}\hat{\Lambda}^\mathsf{T} + \hat{\Sigma}$. The obvious advantage here is that one is guaranteed not to be dealing with distributions that are inconsistent with the original assumptions.

2.3.2 A principal component analysis approach to FA

This is one of the oldest methods for obtaining rough estimates of both the number of factors q and the corresponding matrix of factor loadings Λ.

Estimating factor loadings. As far as Λ is concerned, the method basically exploits the properties of the covariance matrix Ω to find an expression for Λ such that the representation (decomposition) $\Omega = \Lambda\Lambda^\mathsf{T} + \Sigma$ holds. In fact, since Ω is a real symmetric positive definite matrix, it is diagonalisable by virtue of a well known theorem of linear algebra. This means that Ω can be decomposed and expressed as $\Omega = \mathbf{PDP}^\mathsf{T}$, where $\mathbf{D}$ is a diagonal matrix whose elements are the eigenvalues of Ω, and $\mathbf{P}$ is an orthogonal matrix whose columns are the corresponding eigenvectors of Ω. Consequently, if we assume that $\Sigma = 0$ and choose $\Lambda = \mathbf{PD}^{\frac{1}{2}}$, we achieve the desired structure (representation) of Ω. The link with principal component analysis (PCA) comes from the fact, that by setting $\Sigma = 0$, we do not assume a noise model as is the case in PCA, and, with $\Lambda = \mathbf{PD}^{\frac{1}{2}}$, it is easy to see from equation (2.2) that $\tilde{\boldsymbol{z}} = \mathbf{P}^\mathsf{T}(\boldsymbol{x} - \mu) = \mathbf{D}^{\frac{1}{2}}\boldsymbol{z}$ defines principal components.

Estimating the number of factors. The above choice of Λ still contains all the p original dimensions. Since our aim is to reduce the dimensionality of the data, we will only retain the $q < p$ eigenvectors corresponding to the q dominant eigenvalues (that is, those eigenvalues that are "appreciably" larger in magnitude). We first remark that, if the above diagonalisation is done using the sample correlation matrix R, then, if $\tilde{\boldsymbol{x}}^\mathsf{T} = (\tilde{x}_1, \cdots, \tilde{x}_p)^\mathsf{T}$ is the standardised manifest variable and $\mathbf{D} = \mathsf{diag}(\mathsf{d}_1, \cdots, \mathsf{d}_p)$, a well known theorem of linear algebra allows us to write

$$p = \sum_{j=1}^{p} \mathbf{V}(\tilde{x}_j) = \mathbf{tr}(R) = \mathbf{tr}(\mathbf{PDP}^\mathsf{T}) = \mathbf{tr}(\mathbf{D}) = \sum_{j=1}^{p} \mathsf{d}_j. \tag{2.11}$$

(a) According to Kaiser's criterion, we retain only factors with eigenvalues greater than 1. This criterion uses the identity of equation (2.11) and the fact that $\mathbb{V}(\tilde{x}_j) = 1$, and essentially means that, unless a factor extracts at least as much variability as the equivalent of one original variable, we ignore it. (b) Another criterion, which is less clear-cut, consists of retaining only the q-factor model that explains a sufficiently high *percentage of variability.* If we use the above eigenvalues, then the proportion of variability can be measured by

$$\frac{\mathsf{d}_1 + \cdots + \mathsf{d}_q}{\mathsf{d}_1 + \mathsf{d}_2 + \cdots + \mathsf{d}_p}. \tag{2.12}$$

The above proportion of variability explained by the q-factor model can also be assessed using the communalities. In fact, if x_i is standardised, then the communality h_i^2 now represents a percentage, since $\mathbb{V}[\tilde{x}_i] = h_i^2 + \sigma_i^2 = 1$. Computing

$$\sum_{i=1}^{p}\sum_{j=1}^{q}\lambda_{ij}^2 \tag{2.13}$$

therefore provides a measure of the total proportion of variability explained by the q-factor model. By this criterion, if adding a new factor does not "substantially" increase the proportion of variability explained, then that factor is deemed "unimportant". (c) Finally, a graphical assessment can be carried out on the screeplot of the eigenvalues. The number of factors is then chosen as the point where the "elbow" occurs on the screeplot.

Example 1: We illustrate this method of **parameter estimation** using a sample of $n = 200$ artificial observations generated from a factor model with $p = 9$, $q = 2$,

$$\begin{aligned}
\Lambda^{\mathsf{T}} &= \begin{pmatrix} 0.99 & 0.00 & 0.00 & 0.99 & 0.99 & 0.00 & 0.00 & 0.90 & 0.90 \\ 0.00 & 0.95 & 0.90 & 0.00 & 0.00 & 0.95 & 0.95 & 0.00 & 0.00 \end{pmatrix} \\
\Sigma &= \mathsf{diag}(0.02, 0.19, 0.36, 0.02, 0.02, 0.19, 0.19, 0.36, 0.36) \\
\mu^{\mathsf{T}} &= (4.5, 6.5, 7.5, 9.5, 11.5, -11.5, -9.5, -7.5, -6.5)
\end{aligned}$$

For our example, $\mathbf{D} = \mathsf{diag}(4.62, 3.25, 0.32, 0.25, 0.21, 0.17, 0.14, 0.02, 0.02)$, where eigenvalues are ranked in decreasing order of magnitude. Figure (2.1) provides the corresponding screeplot. For our toy problem, it is therefore obvious from the values of the

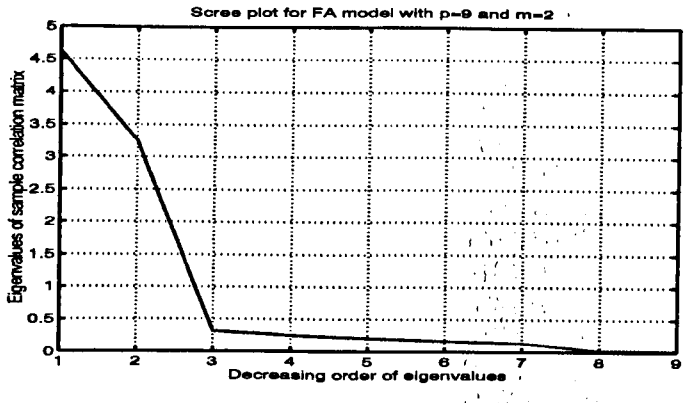


Figure 2.1: Scree plot for an artificial FA model with $p = 9$ and $q = 2$

eigenvalues and the shape of the screeplot, that $q = 2$, which is teh correct value. We therefore simply retain the corresponding first 2 eigenvectors to form our $\hat{\Lambda}_{PC}$. Setting to zero all those entries of the estimate of the matrix of factor loadings that are less than 0.2 for ease of interpretation, we obtain

$$\hat{\Lambda}_{PC}^{T} = \begin{pmatrix} 0.978 & 0.000 & 0.000 & 0.976 & 0.975 & 0.000 & 0.000 & 0.881 & 0.885 \\ 0.000 & -0.921 & -0.866 & 0.000 & 0.000 & -0.929 & -0.911 & 0.000 & 0.000 \end{pmatrix}$$

If we ignore the signs, the above rough estimate of Λ is indeed a very good one, and conveys the structure of the matrix of factor loadings accurately. Although this is a toy problem for which everything was known in advance, it is fair to say that this ad hoc method provides rough estimates that in some cases can be satisfactorily accurate.

Our aim in mentioning the principal component approach to the determination of the number of factors was simply to indicate the existence of methods other than the one we will be mainly focusing on in this chapter.

In the new edition of his book, Jolliffe (1986) dedicates the entirety of the first section of the sixth chapter to an extensive review of methods used to determine the number of principal components. From subsections 6.1.1 to 6.1.3, the book essentially presents the above (a), (b) and (c) criteria in more details. The remainder of the section introduces and describes the choice of the number of principal components using a computationally intensive cross-validation method along the lines of Krzanowski and Marriott (1994), and concludes with a method based on partial correlation. These last two methods are particularly of interest to us as they are reported to work better for the determination of

the number of factors than the estimation of the number of principal components. For that reason, we provide a brief description of these two methods.

Cross-validatory methods: Before giving a brief description of the key ideas behind these cross-validatory methods, it is worth mentioning that they are computationally very intensive, like any other method based on cross-validation. Unlike all the methods described earlier that are based of the eigenvalue decomposition of the sample correlation matrix, these cross-validatory methods are based of the data matrix $\mathbf{X}$ and used procedures similar to the Singular Value Decomposition (SVD) of the data matrix $\mathbf{X}$. The key idea is to predict each element $\boldsymbol{x}_{ij}$ of $\mathbf{X}$ from an equation like the SVD, but based on a submatrix of $\mathbf{X}$ that does not include $\boldsymbol{x}_{ij}$. Wold (1978) and Eastment and Krzanowski (1982) developed two of these cross-validatory approaches to the determination of the number of PCs. The number of terms in the estimate of $\mathbf{X}$ corresponding to the number of PCs is successively taken as $1, 2, 3, \cdots$, and so on, until the overall prediction of the $\boldsymbol{x}_{ij}$'s is no longer significantly improved by the addition of extra PCs. Both Wold (1978) and Eastment and Krzanowski (1982) suggest that the number of PCs to be retained, q, can be taken to be the minimum number necessary for adequate prediction. In this regard, they both use the PREdiction Sum of Squares (PRESS), which is the sum of squared differences between predicted and observed $\boldsymbol{x}_{ij}$, namely,

$$\text{PRESS}(q) = \sum_{i=1}^{n} \sum_{j=1}^{p} (\hat{\boldsymbol{x}}_{ij}^{(q)} - \boldsymbol{x}_{ij})^2. \tag{2.14}$$

Partial correlation: The method of partial correlation that we briefly present here was first proposed by Velicer (1976). The criterion suggested is the average of the squared partial correlations,

$$V = \sum_{i=1, i \neq j}^{p} \sum_{j=1}^{p} \frac{(r_{ij}^*)^2}{p(p-1)}, \tag{2.15}$$

where r_{ij}^* is the partial correlation between the ith and jth variables, given the first q principal components(PCs). The statistic r_{ij}^* is defined as the correlation between the residuals from the linear regression of the ith variable on the first q PCs. It therefore measures the strength of the linear relationship between the ith and jth variables, after

removing the common effect of the first q PCs. Velicer (1976) suggests that the optimal value of q corresponds to the minimum value of the criterion. What makes this method even more relevant to our context is the fact that it is reported to perform reasonably well on deciding the number of factors in factor analysis. Another method based on partial correlation is proposed by Beltrando (1990) who, instead of choosing q such that V is minimised, rather selects q for which the number of statistically significant elements in the matrix of partial correlations is minimised.

Readers inclined to know more about the above mentioned methods are referred to the corresponding references for a more detailed coverage of the topic.

2.3.3 Expression of the likelihood function

In both the frequentist and the Bayesian frameworks, we will need the likelihood function in order to perform our parameter estimation. Depending on the method of estimation that we intend to use, there are two ways of dealing with the likelihood function:

- **Observed-data likelihood:** if we choose to integrate the latent variables out, then we can deal with the marginal distribution of the manifest variable $\boldsymbol{x} \sim \mathcal{N}(\mu, \Lambda\Lambda^{\mathsf{T}} + \Sigma)$, and form the likelihood from the corresponding marginal density. Given the i.i.d. sample $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, the observed-data likelihood in this case is therefore given by

$$\boldsymbol{L}(\boldsymbol{\theta}; \mathbf{X}) \propto |\Lambda\Lambda^{\mathsf{T}} + \Sigma|^{-\frac{n}{2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \mu)^{\mathsf{T}}(\Lambda\Lambda^{\mathsf{T}} + \Sigma)^{-1}(\boldsymbol{x}_i - \mu)\right]. \quad (2.16)$$

  As we said earlier, the observed-data likelihood is often not easy to deal with mathematically, and, for that reason, we will not use it in our estimation procedures.

- **Complete-data likelihood**: As we anticipated in the previous chapter, most of our methods are based on the incomplete-data formulation of latent variable modelling, and the complete-data is therefore our main ingredient. Given the complete-data sample $\mathbf{X}^*$, the complete-data likelihood is given by

$$\boldsymbol{L}(\boldsymbol{\theta}; \mathbf{X}^*) \propto |\Sigma|^{-\frac{n}{2}} \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \Lambda\boldsymbol{z}_i - \mu)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}_i - \Lambda\boldsymbol{z}_i - \mu)\right], \quad (2.17)$$

and the corresponding complete-data log-likelihood $\ell(\theta; \mathbf{X}^*)$ is given by

$$\begin{aligned}\ell(\theta; \mathbf{X}^*) &= -\frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Sigma^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^\mathsf{T}\right] + \sum_{i=1}^{n}\left[\boldsymbol{x}_i^\mathsf{T}\Sigma^{-1}\Lambda\boldsymbol{z}_i\right] + \sum_{i=1}^{n}\left[\boldsymbol{x}_i^\mathsf{T}\Sigma^{-1}\mu\right] \\ &\quad - \sum_{i=1}^{n}\left[\mu^\mathsf{T}\Sigma^{-1}\Lambda\boldsymbol{z}_i\right] - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Lambda^\mathsf{T}\Sigma^{-1}\Lambda\boldsymbol{z}_i\boldsymbol{z}_i^\mathsf{T}\right] - \frac{1}{2}\sum_{i=1}^{n}\mu^\mathsf{T}\Sigma^{-1}\mu. \quad (2.18)\end{aligned}$$

Note: It turns out that the use of the complete-data likelihood allows the derivation (construction) of efficient and easy-to-implement estimating procedures in both the frequentist and Bayesian frameworks.

2.3.4 Multivariate Linear Regression Formulation

The complete-data formulation of the FA model, by providing the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{z}$, allows its analysis to be tackled as a multivariate linear regression problem. In fact, if we define $\ddot{\boldsymbol{x}} \equiv \boldsymbol{x} - \mu = \Lambda\boldsymbol{z} + \boldsymbol{e}$, then for a given complete-data sample $\mathbf{X}^*$, equation (2.2) is equivalent to

$$\ddot{\mathbf{X}} = \mathbf{Z}\Lambda^\mathsf{T} + \mathcal{E}, \quad (2.19)$$

where $\mathcal{E} = (\boldsymbol{e}_1, \cdots, \boldsymbol{e}_n)^\mathsf{T}$ is the $n \times p$ data matrix of errors. Since we assume that $\boldsymbol{e}_i \sim \mathcal{N}_p(0, \Sigma)$ for $i = 1, \cdots, n$, we now have $\mathbf{cov}\left[\mathbf{vec}(\mathcal{E})\right] = \Sigma \otimes \mathbf{I}_n$, where $\otimes$ denotes the Kronecker or direct matrix multiplication operator. In the above formulation 2.19, the transpose Λ^T of the matrix of factor loadings plays the role of regression parameters. It is therefore easy to show that $\hat{\Lambda}^\mathsf{T}$, the least squares estimate of Λ^T, is given by

$$\hat{\Lambda}^\mathsf{T} = \left[\mathbf{Z}^\mathsf{T}\mathbf{Z}\right]^{-1}\mathbf{Z}^\mathsf{T}\ddot{\mathbf{X}}. \quad (2.20)$$

Note: Equation (2.20) is actually saying that each time we provide a set of imputed ("filled-in") values for the latent (missing) variables, we can easily compute the corresponding least squares estimate of Λ as

$$\hat{\Lambda} = \ddot{\mathbf{X}}^\mathsf{T}\mathbf{Z}\left[\mathbf{Z}^\mathsf{T}\mathbf{Z}\right]^{-1}. \quad (2.21)$$

2.4 The EM Algorithm for Factor Analysis

2.4.1 Construction of the generic algorithm

In their seminal paper, Dempster, Laird, and Rubin (1977) suggested the use of the EM algorithm for FA, but the first paper to derive the algorithm explicitly was written by Rubin and Thayer (1982), who later added some extra developments in Rubin and Thayer (1983). The EM for FA is essentially very simple, and we will only touch on the major aspects. The expression of the complete-data log-likelihood in equation (2.18) suggests that, for the E-step, expressions for $\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}]$ and $\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}|\boldsymbol{x}\right]$ must be derived. It turns out that this is easily done using some linear algebra results, and a closed form expression is then obtained for $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. In this setting, we have

$$\boldsymbol{x}^* = \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{x} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} \mathbf{I}_q & \Lambda^{\mathsf{T}} \\ \Lambda & \Lambda\Lambda^{\mathsf{T}} + \Sigma \end{bmatrix}\right). \tag{2.22}$$

From properties of the Gaussian distribution Graybill (1969), Press (1972), the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{x}$ is also Gaussian, with

$$\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}] = \Lambda^{\mathsf{T}}\left[\Lambda\Lambda^{\mathsf{T}} + \Sigma\right]^{-1}[\boldsymbol{x} - \mu] \quad \text{and} \quad \mathbb{V}[\boldsymbol{z}|\boldsymbol{x}] = \mathbf{I}_q - \Lambda^{\mathsf{T}}\left[\Lambda\Lambda^{\mathsf{T}} + \Sigma\right]^{-1}\Lambda. \tag{2.23}$$

Another approach is to use the identity $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{z})\boldsymbol{p}(\boldsymbol{z}) = \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x})$, and it is easy to show that the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{x}$ is also Gaussian with

$$\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}] = (\mathbf{I}_q + \Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda)^{-1}\Lambda^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x} - \mu) \quad \text{and} \quad \mathbb{V}[\boldsymbol{z}|\boldsymbol{x}] = (\mathbf{I}_q + \Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda)^{-1}. \tag{2.24}$$

Note: Computationally, equation (2.23) involves the inversion of a $p \times p$ matrix for which the computational complexity is $O(p^3)$, while equation (2.24) only inverts a $q \times q$ matrix for which the computational complexity is $O(q^3)$ and a diagonal matrix which is computationally easier (only $O(p)$). Since we assume that $q < p$, it is therefore computationally more efficient to use equation (2.24), and from now on we use

$$[\boldsymbol{z}|\boldsymbol{x}] \sim \mathcal{N}_q((\mathbf{I}_q + \Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda)^{-1}\Lambda^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x} - \mu), (\mathbf{I}_q + \Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda)^{-1}). \tag{2.25}$$

With $\boldsymbol{\theta}^{(t)} = \{\mu^{(t)}, \Lambda^{(t)}, \Sigma^{(t)}\}$ and $\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}|\boldsymbol{x}\right] = \mathbb{V}[\boldsymbol{z}|\boldsymbol{x}] + \mathbb{E}[\boldsymbol{z}|\boldsymbol{x}]\left[\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}]\right]^{\mathsf{T}}$, we now write:

Algorithm 1: The EM Algorithm for Factor Analysis

- **E-step** - Compute $\mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i]$ and $\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^\mathsf{T}|\boldsymbol{x}_i\right]$ for $\mathbf{X} = \{\boldsymbol{x}_i : i = 1, \cdots, n\}$.
- **M-step** -

$$\begin{aligned}
\mu^{(t+1)} &= \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}_i - \Lambda^{(t)}\mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i]\right) \qquad (2.26)\\
\Lambda^{(t+1)} &= \left[\sum_{i=1}^{n}\left(\boldsymbol{x}_i - \mu^{(t+1)}\right)\left(\mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i]\right)^\mathsf{T}\right]\left[\sum_{i'=1}^{n}\mathbb{E}\left[\boldsymbol{z}_{i'}\boldsymbol{z}_{i'}^\mathsf{T}|\boldsymbol{x}_{i'}\right]\right]^{-1}\\
\Sigma^{(t+1)} &= \frac{1}{n}\mathrm{diag}\left[\sum_{i=1}^{n}\left(\boldsymbol{x}_i - \mu^{(t+1)} - \Lambda^{(t+1)}\mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i]\right)\left(\boldsymbol{x}_i - \mu^{(t+1)}\right)^\mathsf{T}\right].
\end{aligned}$$

Proposition 2.1 *The new estimate* $\Lambda^{(t+1)}$ *of* Λ *at each iteration of the EM algorithm* (2.26) *has exactly the same form as the least squares estimate of* (2.21).

Proof: In fact, since $\sum_{i=1}^{n}\boldsymbol{z}_i\boldsymbol{z}_i^\mathsf{T} = \mathbf{Z}^\mathsf{T}\mathbf{Z}$ and $\sum_{i=1}^{n}\ddot{\boldsymbol{x}}_i\boldsymbol{z}_i^\mathsf{T} = \ddot{\mathbf{X}}^\mathsf{T}\mathbf{Z}$, equation (2.26) can be rewritten as $\Lambda^{(t+1)} = \ddot{\mathbf{X}}^\mathsf{T}\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}}[\mathbf{Z}]\left[\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}}\left[\mathbf{Z}^\mathsf{T}\mathbf{Z}\right]\right]^{-1}$. If we drop expectations, then $\Lambda^{(t+1)}$ and $\hat{\Lambda}$ will be identical. □

Note: While $\Lambda^{(t+1)}$ is used in the EM algorithm, its least squares look-alike $\hat{\Lambda}$ will be frequently used when we consider the construction of sampling schemes for the FA model.

2.4.2 Numerical results

As we said earlier, principal component estimates can at least be used as initial values for more sophisticated estimating procedures like the EM algorithm. For the specific variances σ_i^2, we use initial estimates suggested by Jöreskog (1975), namely

$$\hat{\sigma}_i^2 = \left(1 - \frac{1}{2}\cdot\frac{q}{p}\right)\left(\frac{1}{s_{ii}^{-1}}\right) \qquad (2.27)$$

where s_{ii}^{-1} is the i-th diagonal element of the sample covariance matrix S^{-1}. We use $\epsilon = 10^{-5}$ as our tolerance.

Example 1 revisited. We reconsider the example used for principal component factor analysis where we had $p = 9$ and $q = 2$. For this example, $T = 2000$ iterations take

approximately 3 minutes and produce the following estimates.

$$\hat{\Lambda}_{EM}^{T} = \begin{pmatrix} 0.989 & 0.000 & 0.000 & 0.985 & 0.986 & 0.000 & 0.000 & 0.832 & 0.829 \\ 0.000 & -0.897 & -0.807 & 0.000 & 0.000 & -0.922 & -0.889 & 0.000 & 0.000 \end{pmatrix}$$

$$\hat{\Sigma}_{EM} = \text{diag}(0.015, 0.189, 0.342, 0.022, 0.021, 0.144, 0.204, 0.302, 0.305)$$

Below is a plot of the log-likelihood throughout the iterations of the EM algorithm.

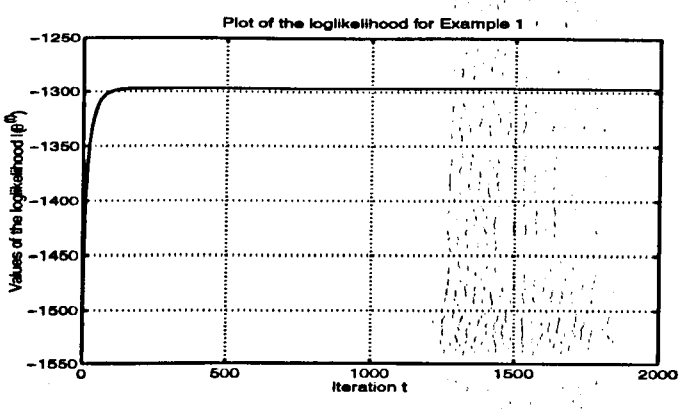


Figure 2.2: Plot of the log-likelihood for Example 1

Figure (2.2) shows that the maximum is reached after just $t = 100$ iterations. This corresponds to less than 9 seconds, meaning that the EM is reasonable fast. Moreover, with principal components initial values, the final EM estimates are satisfactorily accurate.

Example 2: Exploration of Group Structure. For this example, we generate data from a population with three groups, each group being adequately modelled by its own factor model, but all groups having the same matrix of specific variances $\Sigma = \text{diag}(0.02, 0.19, 0.36, 0.02, 0.02, 0.19, 0.19, 0.36, 0.36)$. Below are the 3 matrices of factor loadings for the model.

$$\Lambda_1^T = \begin{pmatrix} 0.80 & 0.65 & 0.00 & 0.50 & 0.00 & 0.00 & 0.95 & 0.00 & 0.00 \\ 0.00 & 0.35 & 0.90 & 0.00 & 0.50 & 0.90 & 0.00 & 0.90 & 0.90 \end{pmatrix}$$

$$\Lambda_2^T = \begin{pmatrix} 0.69 & 0.15 & 0.00 & 0.19 & 0.99 & 0.80 & 0.00 & 0.00 & 0.99 \\ 0.00 & 0.90 & 0.90 & 0.00 & 0.00 & 0.00 & 0.60 & 0.90 & 0.10 \end{pmatrix}$$

$$\Lambda_3^T = \begin{pmatrix} 0.59 & 0.95 & 0.00 & 0.19 & 0.29 & 0.00 & 0.00 & 0.00 & 0.99 \\ 0.00 & 0.35 & 0.60 & 0.90 & 0.90 & 0.75 & 0.80 & 0.50 & 0.00 \end{pmatrix}$$

Using the same initialisation procedure as before, $T = 2000$ iterations of the EM algorithm allow us to get a 2-D scatter plot (Fig. 2.3) of the estimated factor scores for the sample of $n = 300$ observations. The parameter estimates obtained in this case are not

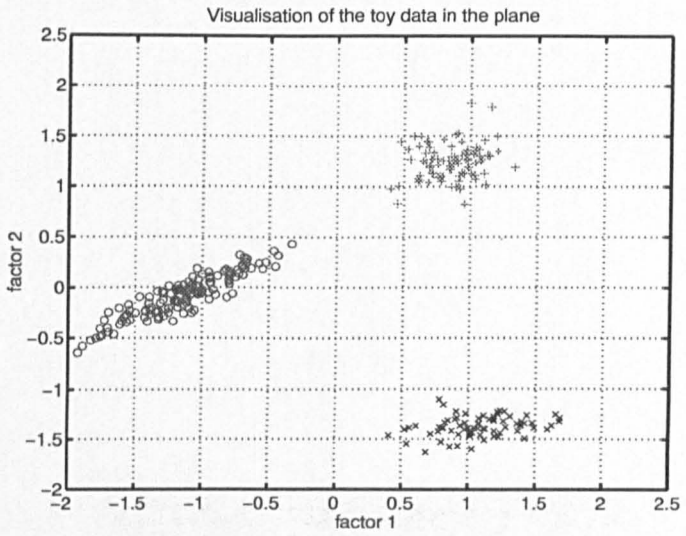


Figure 2.3: Visualisation of Example 2 in the plane

the true parameters of the model, since we are using a single factor model to tackle a problem requiring 3 distinct sub-models. However, this single factor model is still able to provide estimated factor scores that allow us to discover the existence of three different subgroups in the population of interest. Here, the single factor model has served as a useful exploratory tool, and has revealed the grouping structure very well.

2.4.3 Some aspects of the EM algorithm

Multiplicity of solutions: As reported by McLachlan and Krishnan (1997) and Rubin and Thayer (1982) and confirmed by our simulations, one of the main weaknesses of this generic version of the EM for FA is the fact that it produces a multiplicity of solutions that are not rotated images of each other. In the above **Example 2** for instance, using different random initial values systematically leads to different solutions. Mathematically, this multiplicity is easily explained as a consequence of the fact that the unrestricted factor model is indeterminate (unidentifiable). Restricting the factor model to guarantee a unique solution is therefore a natural candidate solution to this

problem, and this essentially boils down to modifying the M-Step of the EM algorithm in such a way that the maximisation with respect to Λ becomes a constrained optimisation problem, with constraints being the ones presented in Section 2.2.3. Liu and Rubin (1994) developed a variant of the EM algorithm known as the ECME that deals with this problem. Dong and Taylor (1995)'s restricted EM is a much more general setting that should provide a good way of tackling this application of the EM under restrictions. Because these restricted versions of the EM algorithm require extra computational effort through the implementation of constrained optimisation algorithms at the M-Step, we do not use them in our work. Finally, it is worth pointing out that the Stochastic EM algorithm could be another alternative to this multiplicity of solutions, since it is theoretically insensitive to initial values and does converge to the same fixed point.

Relative convergence rate: The relatively slow convergence of the EM algorithm can be attributed to the amount of missing data, which is directly and systematically proportional to the amount of observed data in this context of latent variable models. We noticed in our simulations that as q grows, convergence gets even slower, and, for the same q, a larger sample would require more iterations to convergence, since more missing data in such situations need to be "filled-in" or accounted for.

Post Analysis of Estimates: The EM algorithm only provides a point estimate of $\boldsymbol{\theta}$. This is one of the major drawbacks of the algorithm, since it is desirable to also provide estimated standard errors, in order to give an indication of confidence intervals or confidence regions for the population parameters of interest. It turns out that such error estimates can be obtained thanks to some properties of maximum likelihood estimators, but at the expense of extra computational efforts. In fact, from the properties of the algorithm, the estimate obtained is a maximum likelihood estimate and is therefore asymptotically unbiased. Since $\hat{\boldsymbol{\theta}}_{\mathsf{EM}}$ is a maximum likelihood estimate of $\boldsymbol{\theta}$, it is asymptotically normally distributed. More specifically

$$\hat{\boldsymbol{\theta}}_{\mathsf{EM}} \sim \mathcal{N}(\boldsymbol{\theta}, J^{-1}(\boldsymbol{\theta})) \quad \text{with} \quad j_{ik}(\boldsymbol{\theta}) = -\mathbb{E}\left[\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial\theta_i \partial\theta_k}\right], \tag{2.28}$$

where $J(\boldsymbol{\theta}) = -\mathbb{E}[\mathbf{H}(\boldsymbol{\theta})]$ is *minus the expected Hessian matrix* (the matrix of second derivatives). In practice however, what is generally used is the *observed Fisher Informa-*

tion matrix $I_F(\hat{\boldsymbol{\theta}}) = -H(\hat{\boldsymbol{\theta}})$. For a more detailed explanation see Morgan (2000). Since the EM does not deliver I_F, extra computational effort is needed to find it.

2.4.4 Goodness-of-fit test for Factor Analysis

With the normality assumption for the manifest variable $\boldsymbol{x}$, we performed maximum likelihood via the EM algorithm. On the other hand, the goodness-of-fit of the resulting q-factor model can be judged using a classical likelihood ratio test, with the null hypothesis stating the covariance matrix of $\boldsymbol{x}$ has the structure $\boldsymbol{\Omega} = \Lambda\Lambda^{\mathsf{T}} + \Sigma$, and the alternative saying the covariance matrix is unconstrained. Under the normal assumption, it is easy to see that the test statistic for the test is

$$\omega = n(\mathbf{tr}(\hat{\boldsymbol{\Omega}}^{-1}S) - \log|\hat{\boldsymbol{\Omega}}^{-1}S| - p), \tag{2.29}$$

where $\hat{\boldsymbol{\Omega}} = \hat{\Lambda}\hat{\Lambda}^{\mathsf{T}} + \hat{\Sigma}$ is the estimate of $\boldsymbol{\Omega}$, and S is the sample covariance matrix defined and encountered earlier. A standard result in the literature shows that if $\Sigma > 0$, then ω is asymptotically χ^2 distributed with $\nu = \frac{1}{2}[(p-q)^2 - (p+q)]$ degrees of freedom under the null hypothesis. An alternative setting proposed by Bartlett (1954) suggests to replace n in (2.29) by $n - 1 - \frac{1}{6}(2p+5) - \frac{2}{3}q$. It must be said that the value of ν used above presupposes that we have efficiently fitted the model, and therefore that instead of the $p(q+1)$ parameters of the unrestricted FA model, only $p(q+1) - \frac{1}{2}q(q-1)$ parameters have to be estimated.

2.5 Data Augmentation for Factor Analysis

In the Bayesian framework, we would ideally like to use the posterior density $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ of $\boldsymbol{\theta}$ to make various inferences. Unfortunately, as we said earlier, closed-form expressions of the desired moments and marginals are not obtainable. Data Augmentation provides an elegant way to tackle this problem. Our main ingredient for the derivation of the Data Augmentation algorithm for FA is the completion of $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ given by the following

equation:

$$p(\theta, \mathbf{Z}|\mathbf{X}) \propto L(\theta; \mathbf{X}, \mathbf{Z})p(\theta) \tag{2.30}$$

The **I-step** in this case is straightforward, and simply consists of drawing samples from the conditional predictive distribution of z given x and the current set of parameter values $\theta^{(t)}$ as given by equation (2.25). For the **P-step**, we combine the prior density $p(\theta)$ with the expression for the complete-data likelihood $L(\theta; \mathbf{X}, \mathbf{Z})$ to derive the corresponding full conditional posteriors $p(\theta|\mathbf{X}, \mathbf{Z})$.

2.5.1 Aspects of prior specification

We shall use only natural conjugate priors in this context. In fact, the complete-data likelihood (2.17) belongs to the regular exponential family of distributions, and therefore allows a straightforward derivation of conjugate priors. While this choice is made for mathematical convenience, it also turns out to be the only computationally viable choice in this context. Martin and McDonald (1981) and Ihara and Kano (1995) have shown that the use of standard improper reference priors leads to the Bayesian analogue of what is known in factor analysis as *Heywood cases*[4]. Treated as a function of the variance parameter, the negative likelihood of the FA model is bounded below away from zero as σ_i^2 tends to zero. Throughout our work, we exclusively use conjugate priors.

2.5.2 From likelihood to natural conjugate priors

We now express the complete-data likelihood as a function of each parameter in turn, and we derive the corresponding natural conjugate prior. A more comprehensive coverage of prior specification for this normal model can be found in Box and Tiao (1973), Bernardo and Smith (1994), Zellner (1971) or Gelman, Carlin, Stern, and Rubin (1995), Press (1972).

[4]In the classical maximum likelihood estimation of the FA model, it is often convenient to minimise the negative log-likelihood or some extensions of it. It often happens that the objective function used has a relative minimum corresponding to negative values for some variances. Such solutions are clearly inadmissible and are referred to as improper solutions or Heywood cases.

Prior distribution for Σ: Treating equation (2.17) as a function of Σ, we can write

$$\boldsymbol{L}(\Sigma^{-1}) \propto |\Sigma^{-1}|^{n/2} \exp\left[-\frac{1}{2}\mathbf{tr}(\Sigma^{-1}\mathbf{S})\right], \tag{2.31}$$

where $\mathbf{S} = \sum_{i=1}^{n}(\boldsymbol{x}_i - \Lambda\boldsymbol{z}_i - \mu)(\boldsymbol{x}_i - \Lambda\boldsymbol{z}_i - \mu)^{\mathsf{T}} = (\ddot{\mathbf{X}} - \mathbf{Z}\Lambda^{\mathsf{T}})^{\mathsf{T}}(\ddot{\mathbf{X}} - \mathbf{Z}\Lambda^{\mathsf{T}})$. The form of (2.31) suggests that a natural conjugate prior for Σ^{-1} would be a Wishart distribution. However, since Σ^{-1} is diagonal, (2.31) can be rewritten as

$$\boldsymbol{L}(\Sigma^{-1}) \propto \prod_{i=1}^{p}\left[\sigma_i^{-2}\right]^{n/2} \exp\left[-\frac{1}{2}\mathbf{S}_{ii}\sigma_i^{-2}\right], \tag{2.32}$$

which has the form of a product of Gamma densities, suggesting that we use a product of Gamma prior densities $\boldsymbol{p}(\sigma_i^{-2})$ for each σ_i^{-2}.

Prior distribution for Λ: To write the likelihood as a function of Λ, we remark that

$$(\ddot{\mathbf{X}} - \mathbf{Z}\Lambda^{\mathsf{T}})^{\mathsf{T}}(\ddot{\mathbf{X}} - \mathbf{Z}\Lambda^{\mathsf{T}}) = (\ddot{\mathbf{X}} - \mathbf{Z}\hat{\Lambda}^{\mathsf{T}})^{\mathsf{T}}(\ddot{\mathbf{X}} - \mathbf{Z}\hat{\Lambda}^{\mathsf{T}}) + (\Lambda^{\mathsf{T}} - \hat{\Lambda}^{\mathsf{T}})^{\mathsf{T}}\mathbf{Z}^{\mathsf{T}}\mathbf{Z}(\Lambda^{\mathsf{T}} - \hat{\Lambda}^{\mathsf{T}}).$$

Since $(\ddot{\mathbf{X}} - \mathbf{Z}\hat{\Lambda}^{\mathsf{T}})^{\mathsf{T}}(\ddot{\mathbf{X}} - \mathbf{Z}\hat{\Lambda}^{\mathsf{T}})$ does not depend on Λ, we can then write

$$\boldsymbol{L}(\Lambda) \propto \exp\left[-\frac{1}{2}\mathbf{tr}\Sigma^{-1}(\Lambda^{\mathsf{T}} - \hat{\Lambda}^{\mathsf{T}})^{\mathsf{T}}(\mathbf{Z}^{\mathsf{T}}\mathbf{Z})(\Lambda^{\mathsf{T}} - \hat{\Lambda}^{\mathsf{T}})\right]. \tag{2.33}$$

Let $\vartheta = [\vartheta_1^{\mathsf{T}}, \cdots, \vartheta_p^{\mathsf{T}}]^{\mathsf{T}} = \mathsf{vec}(\Lambda^{\mathsf{T}}) = [\Lambda_{1.}^{\mathsf{T}}, \cdots, \Lambda_{p.}^{\mathsf{T}}]^{\mathsf{T}}$. Each ϑ_j is a $q \times 1$ column vector made up of the elements of the j-th row $\Lambda_{j.}$ of Λ. Clearly, ϑ is a $qp \times 1$ column vector. Since $\mathbf{tr}\{\Sigma^{-1}(\Lambda^{\mathsf{T}} - \hat{\Lambda}^{\mathsf{T}})^{\mathsf{T}}(\mathbf{Z}^{\mathsf{T}}\mathbf{Z})(\Lambda^{\mathsf{T}} - \hat{\Lambda}^{\mathsf{T}})\} = (\vartheta - \hat{\vartheta})^{\mathsf{T}}[\Sigma^{-1} \otimes (\mathbf{Z}^{\mathsf{T}}\mathbf{Z})](\vartheta - \hat{\vartheta})$ and $[\Sigma^{-1} \otimes (\mathbf{Z}^{\mathsf{T}}\mathbf{Z})]^{-1} = \Sigma \otimes (\mathbf{Z}^{\mathsf{T}}\mathbf{Z})^{-1}$, the likelihood as a function of ϑ then has the form

$$\boldsymbol{L}(\Lambda) = \boldsymbol{L}(\vartheta) \propto \exp\left[-\frac{1}{2}(\vartheta - \hat{\vartheta})^{\mathsf{T}}[\Sigma \otimes (\mathbf{Z}^{\mathsf{T}}\mathbf{Z})^{-1}]^{-1}(\vartheta - \hat{\vartheta})\right], \tag{2.34}$$

which suggests that a Gaussian distribution would be a natural conjugate prior for ϑ.

Prior distribution for μ: As a function of μ, (2.17) has the form

$$\boldsymbol{L}(\mu) \propto \exp\left[-\frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \Lambda\boldsymbol{z}_i - \mu)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}_i - \Lambda\boldsymbol{z}_i - \mu)\right], \tag{2.35}$$

which suggests that a Gaussian distribution would be a natural conjugate prior for μ.

2.5.3 Derivation of full conditional distributions

Using all the elements of prior specification from the previous section, we now derive the full conditional posteriors to be used in our sampling scheme. We assume that our parameters are *a priori* independent, and this allows us to write the joint prior density as $\boldsymbol{p}(\boldsymbol{\theta}) = \boldsymbol{p}(\Lambda, \mu, \Sigma) = \boldsymbol{p}(\mu|\xi, \kappa)\boldsymbol{p}(\Sigma|\alpha, \tau)\boldsymbol{p}(\Lambda|\eta, \Omega)$, where $\xi, \kappa, \alpha, \tau, \eta, \Omega$ are hyperparameters.

Full conditional distribution of Σ. As we said earlier, the Wishart distribution for Σ^{-1} reduces to a product of Gamma distributions because of the diagonality of Σ. In other words, if we assume $\sigma_i^{-2} \sim \mathsf{Ga}(\alpha/2, \tau/2)$, then the prior density for Σ^{-1} becomes

$$\boldsymbol{p}(\Sigma^{-1}|\alpha, \tau) = \prod_{i=1}^{p} \boldsymbol{p}(\sigma_i^{-2}|\alpha, \tau) \propto \prod_{i=1}^{p} [\sigma_i^{-2}]^{\frac{1}{2}\alpha - 1} \exp\left[-\frac{1}{2}\tau\sigma_i^{-2}\right]. \tag{2.36}$$

If we combine equations (2.36) and (2.32), we easily derive a Gamma full conditional distribution of each σ_i^{-2}, that is, $[\sigma_i^{-2}|\cdots] \sim \mathsf{Ga}((n+\alpha)/2, (S_{ii}+\tau)/2)$, for $i = 1, \cdots, p$.

Full conditional distribution of μ. Let $\mu \sim \mathcal{N}(\xi, \kappa)$ where ξ is a $p \times 1$ column vector and κ is a $p \times p$ symmetric positive definite matrix, be the natural conjugate prior for μ. If we combine this prior with the complete-data likelihood as expressed in equation (2.35), standard results from Gaussian theory allow us to derive a Gaussian full conditional distribution of μ, that is $[\mu|\cdots] \sim \mathcal{N}_p(m_\mu, C_\mu)$, where $\xi_{\boldsymbol{x}} = \sum_{i=1}^{n} (\boldsymbol{x}_i - \Lambda \boldsymbol{z}_i)$,

$$C_\mu^{-1} = \kappa^{-1} + n\Sigma^{-1} \quad \text{and} \quad m_\mu = C_\mu(\kappa^{-1}\xi + \Sigma^{-1}\xi_{\boldsymbol{x}}). \tag{2.37}$$

Full conditional distribution of unrestricted Λ. We assume that the rows of Λ are independent, and therefore $\boldsymbol{p}(\Lambda|\eta, \Omega) = \prod_{i=1}^{p} \boldsymbol{p}(\Lambda_{i\cdot}|\eta, \Omega)$. The natural conjugate prior for $\vartheta = \mathsf{vec}(\Lambda^\mathsf{T})$ being Gaussian as we derived earlier, we also have Gaussian conjugate priors for the rows of Λ. More specifically, we take $\Lambda_{i\cdot} \sim \mathcal{N}_q(\eta, \Omega)$, where $\Omega \in \mathbb{R}^{q \times q}$, and $\eta \in \mathbb{R}^q$. To derive the full conditional posterior for Λ, we write the prior for ϑ as

$$\boldsymbol{p}(\vartheta) \propto \exp\left[-\frac{1}{2}(\vartheta - \vartheta_0)^\mathsf{T} B^{-1}(\vartheta - \vartheta_0)\right], \tag{2.38}$$

where $\vartheta_o = [\eta^\mathsf{T}, \cdots, \eta^\mathsf{T}]^\mathsf{T}$ and $B = \mathbf{I}_q \otimes \Omega$. Note that $\vartheta_o \in \mathbb{R}^{pq \times 1}$ and $B \in \mathbb{R}^{pq \times pq}$. If we combine equations (2.34) and (2.38), properties of the Gaussian distribution allow us to

derive the full conditional posterior $[\vartheta|\cdots] \sim \mathcal{N}_{pq}(m_\vartheta, C_\vartheta)$, where

$$C_\vartheta^{-1} = B^{-1} + [\Sigma^{-1} \otimes (\mathbf{Z}^\top \mathbf{Z})] \quad \text{and} \quad m_\vartheta = C_\vartheta[B^{-1}\vartheta_0 + [\Sigma^{-1} \otimes (\mathbf{Z}^\top \mathbf{Z})]^{-1}\hat{\vartheta}]. \tag{2.39}$$

Recall that $[\Sigma^{-1} \otimes (\mathbf{Z}^\top \mathbf{Z})]^{-1} = \Sigma \otimes (\mathbf{Z}^\top \mathbf{Z})^{-1}$. By definition of the Kronecker product, and by virtue of the diagonality of Σ, $\Sigma \otimes (\mathbf{Z}^\top \mathbf{Z})^{-1}$ is block-diagonal, and we can write

$$\Sigma \otimes (\mathbf{Z}^\top \mathbf{Z})^{-1} = \begin{bmatrix} \sigma_1^2(\mathbf{Z}^\top \mathbf{Z})^{-1} & & 0 & & 0 \\ & \ddots & & & \\ 0 & & \sigma_r^2(\mathbf{Z}^\top \mathbf{Z})^{-1} & & 0 \\ & & & \ddots & \\ 0 & & 0 & & \sigma_p^2(\mathbf{Z}^\top \mathbf{Z})^{-1} \end{bmatrix} \tag{2.40}$$

Using (2.21), we write $\hat{\vartheta} = \text{vec}(\hat{\Lambda})$, and it is easy to verify that the full conditional for each row of Λ is given by $[\Lambda_{i.}|\cdots] \sim \mathcal{N}_q(m_{\Lambda_{i.}}, C_{\Lambda_{i.}})$, where

$$C_{\Lambda_{i.}}^{-1} = \Omega^{-1} + \sigma_i^{-2}(\mathbf{Z}^\top \mathbf{Z}), \qquad m_{\Lambda_{i.}} = C_{\Lambda_{i.}}(\Omega^{-1}\eta + \sigma_i^2 \mathbf{Z}^\top \ddot{\mathbf{X}}_{.i}), \tag{2.41}$$

and $\ddot{\mathbf{X}}_{.i}$ is the i-th column of the data matrix $\ddot{\mathbf{X}}$.

Full conditional distribution of restricted Λ. For interpretability and estimation efficiency, Λ needs to be restricted as in equation (2.7). Such a restriction leads to a slight modification in the specification of the full conditional distribution of Λ. For simplicity, we now assume a univariate Gaussian[5] prior for each of the non-preassigned λ_{ij}, that is, $\lambda_{ij} \sim \mathcal{N}(m_o, C_o)$. We define $\mathbf{Z}_{(i)} \in I\!R^{n \times i}$ to be the $n \times i$ matrix containing the first i columns of $\mathbf{Z}$. The mean vector and covariance matrix of the full conditional distribution of $\Lambda_{i.}$ for the first q rows $(i = 1, \cdots, q)$ are determined as follows:

$$C_{\Lambda_{i.}}^{-1} = C_o^{-1}\mathbf{I}_i + \sigma_i^{-2}(\mathbf{Z}_{(i)}^\top \mathbf{Z}_{(i)}) \quad \text{and} \quad m_{\Lambda_{i.}} = C_{\Lambda_{i.}}(C_o^{-1} m_o \mathbf{1}_i + \sigma_i^2 \mathbf{Z}_{(i)}^\top \ddot{\mathbf{X}}_{.i}). \tag{2.42}$$

For $i = (q+1), \cdots, p$, the hyperparameters become

$$C_{\Lambda_{i.}}^{-1} = C_o^{-1}\mathbf{I}_q + \sigma_i^{-2}(\mathbf{Z}^\top \mathbf{Z}) \quad \text{and} \quad m_{\Lambda_{i.}} = C_{\Lambda_{i.}}(C_o^{-1} m_o \mathbf{1}_q + \sigma_i^2 \mathbf{Z}^\top \ddot{\mathbf{X}}_{.i}). \tag{2.43}$$

[5]When factor loadings are viewed as correlations instead of covariances, Press (1972) suggests an alternative assumption which consists of using a transformed Dirichlet prior whose elements are transformed beta variates defined on the range $(-1, 1)$. We do not use this approach in our work.

With all the posterior distributions specified above, the two steps of the Data Augmentation for Factor Analysis have the following form:

Algorithm 2: Data Augmentation for Factor Analysis

- **I-step -**

$$[\boldsymbol{z}|\boldsymbol{x}] \sim \mathcal{N}_q((\mathbf{I}_q + \Lambda^\mathsf{T}\boldsymbol{\Sigma}^{-1}\Lambda)^{-1}\Lambda^\mathsf{T}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\mu), (\mathbf{I}_q + \Lambda^\mathsf{T}\boldsymbol{\Sigma}^{-1}\Lambda)^{-1})$$

- **P-step -**

$$\begin{aligned}
[\sigma_i^{-2}|\cdots] &\sim \mathsf{Ga}((n+\alpha)/2, (S_{ii}+\tau)/2) \quad i = 1, \cdots, p \\
[\mu|\cdots] &\sim \mathcal{N}_p(m_\mu, C_\mu) \\
[\Lambda_{i.}|\cdots] &\sim \mathcal{N}_q(m_{\Lambda_{i.}}, C_{\Lambda_{i.}}), \quad i = 1, \cdots, p
\end{aligned}$$

2.5.4 Some advantages of Bayesian sampling

Computational efficiency. One of the most striking features of the sampling scheme we have just derived is that all the full conditional posterior distributions are familiar distributions that are easy to simulate. Hence, Data Augmentation is efficient here.

Initial conditions. Unlike the EM for which different initial values tend to lead to different limiting values, Data Augmentation overcomes the multiplicity of solutions as the algorithm always converges to the same limiting distribution regardless of the initialisation of the chain.

Adaptability. On the other hand, the scheme is also elegant and adaptable as we can see from the ease in deriving the conditional posteriors for the restricted FA model.

Post analysis. One of the main strengths of the Bayesian sampling approach is its ability to allow a post-analysis of the derived model. In fact, the sample generated from the Markov Chain Monte Carlo iterations can be used to make further inferences about the model. For instance, the MCMC sample path can be readily used to assess the fitness of the proposed model, and also derive standard error estimates of the parameters for the construction of confidence intervals or confidence regions. One of the key advantages

here is that these additional inferential tasks are all by-products of the same process and do not require any further heavy computations, unlike with the EM algorithm where such inferences could mean resorting to computationally intensive methods.

Note: While the Bayesian paradigm offers all the above advantages, it is fair to point out that it requires the specification of the prior distribution. In any real application, care should be taken to assess how sensitive the results are to prior specification.

2.5.5 Elements of MCMC convergence

The crucial issues of convergence of MCMC iterations are twofold. First of all, MCMC algorithms are relatively slower than their deterministic counterparts. As we shall see in our numerical section, the Data Augmentation algorithm requires considerably more computation time than the EM algorithm on the same tasks. Secondly, as we said earlier, one of the most difficult areas in MCMC methods is the assessment of convergence. While deterministic methods like the EM have objective functions that provide a straightforward way to assess the convergence of the iterative process, MCMC methods, especially in the multivariate setting are still far from offering clear-cut and computationally realistic and efficient tools to monitor the convergence of chains. In a multivariate setting, like the one of interest to us, there are methods that consist of assessing the convergence of each scalar quantity separately. One of the easiest ways to assess the convergence of scalar quantities is the use of *time-series plots* over iterations. This is obviously not a practical solution if one has a model with many parameters. Gelman and Rubin (1992)'s method based on the analysis of variance of multiple chains provides satisfactory results, but again it focuses on individual scalar parameters, and the need to have many chains run simultaneously makes the implementation of the method both complicated and computationally intensive. In general, all such methods based on scalar quantities can quickly become impractical when the number of parameters is large, since some parameters would converge while others have not yet reached their equilibrium regime. There is a crucial need for methods of convergence assessment that can be both easy to implement and easy to interpret. The topic is currently very active, and both

theoretical and empirical methods are being devised to address this important issue. The reader is referred to such references as Cowles and Carlin (1996), Gelman and Rubin (1992), Gelman (1995), Geyer (1992) for a more detailed account and review of convergence monitoring and assessment for MCMC. As far as our work is concerned, whenever possible, we use some of the tested methods to monitor the convergence of our chains, and, in some cases, we rely on large numbers of iterations to achieve convergence if our theoretically ergodic chains tends to be slow to converge.

2.5.6 Point estimates and standard errors

It is a well known fact in statistical estimation and inference that uncertainty about the point estimate of a parameter should be summarised by computing estimated standard errors and constructing the corresponding confidence intervals or regions. However, while the construction of such confidence intervals or regions often requires extra computational burden in likelihood-based methods in our context, it is easily done in the Bayesian sampling framework as follows. Suppose that we have discarded the T_o draws of the *burn-in* step of the process once convergence is judged to have been reached. In the event of very slow convergence, successive draws from the equilibrium distribution tend to be correlated, and, as explained by Schafer (1997), dependent samples can make the inferential task extremely difficult. In practice, subsampling the chain is a solution that works. Hence, we use *subsampling* to overcome the dependency of our MCMC samples: instead of summarising our posterior by $\boldsymbol{\theta}^{(T_o+1)}, \boldsymbol{\theta}^{(T_o+2)}, \cdots, \boldsymbol{\theta}^{(T_o+M)}$, we rather use $\boldsymbol{\theta}^{(T_o+c)}, \boldsymbol{\theta}^{(T_o+2c)}, \cdots, \boldsymbol{\theta}^{(T_o+Mc)}$, where c is chosen large enough to make the sample values approximately independent. Let $\{\boldsymbol{\theta}^{(t)} : t = 1, \cdots, M\}$ be our resulting chain of parameters. As we said in Chapter 1, this can be regarded as a genuine sample from the observed-data posterior $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$. Our point estimates for both the parameters and the factor scores are

$$\hat{\boldsymbol{\theta}} = \frac{1}{M}\sum_{t=1}^{M}\boldsymbol{\theta}^{(t)} \quad \text{and} \quad \hat{\boldsymbol{z}}_i = \frac{1}{M}\sum_{t=1}^{M}\boldsymbol{z}_i^{(t)}, i = 1, \cdots, n, \tag{2.44}$$

and the corresponding estimates of variances are given by

$$\widehat{\mathbb{V}[\boldsymbol{\theta}|\mathbf{X}]} = \frac{1}{M-1}\sum_{t=1}^{M}(\boldsymbol{\theta}^{(t)}-\hat{\boldsymbol{\theta}})(\boldsymbol{\theta}^{(t)}-\hat{\boldsymbol{\theta}})^{\mathsf{T}}$$

$$\widehat{\mathbb{V}[\boldsymbol{z}_i|\mathbf{X}]} = \frac{1}{M-1}\sum_{t=1}^{M}(\boldsymbol{z}_i^{(t)}-\hat{\boldsymbol{z}}_i)(\boldsymbol{z}_i^{(t)}-\hat{\boldsymbol{z}}_i)^{\mathsf{T}}, i=1,\cdots,n. \tag{2.45}$$

Remark: Even without subsampling, the above estimate of $\boldsymbol{\theta}$ can still be appropriate[6]. In fact, a law of large numbers for MCMC Tierney (1994) states that under quite general conditions, if $Z^{(1)}, Z^{(2)}, \cdots, Z^{(M)}$ is a realisation of an MCMC run with target distribution $\boldsymbol{f}$, then $M^{-1}\sum_{t=1}^{M}\boldsymbol{g}(Z^{(t)})$ converges to $\mathbb{E}_{\boldsymbol{f}}[\boldsymbol{g}(Z)]$ almost surely, for any real-valued function $\boldsymbol{g}(Z)$ as $M \longrightarrow \infty$, provided that $\mathbb{E}_{\boldsymbol{f}}[\boldsymbol{g}(Z)]$ exists.

2.6 Bayesian assessment of model fitness

Fitting a model to a given set of data is one thing, but whether or not the fitted model is the most plausible mechanism that generated the data is another issue altogether, an issue of paramount importance in statistical modelling. Every serious statistical analysis should therefore include at least a check to see if the posited model should be excluded, based on whether or not it does provide a reasonable summary of the data at hand. A standard classical approach for this kind of model-checking is to perform a goodness-of-fit test, which consists of calculating a tail-area probability under the posited model to quantify the extremeness of the observed value of some selected discrepancy (used as test statistic), one of the natural candidates used to measure discrepancy being some measure of the difference between the observations and the predictions. Essentially, in the classical approach the tail-area probability, or p-value, is used as a computationally convenient way to locate the observed value of the discrepancy in the reference distribution under the proposed model. The bad news with this classical approach is that, for many complex problems like the ones we are interested in, it is not always possible (or at least it is not easy) to specify a reference distribution for the test statistic, and even the test statistic itself is not always easy to define and specify. The good news is that, in the Bayesian

[6]It is important to note that this does not hold true for the variance estimates.

paradigm, this is both feasible and requires almost no extra computational burden in the framework of Bayesian sampling.

2.6.1 Posterior predictive assessment of model fitness

The method we will now describe and use provides a satisfactory Bayesian alternative to the classical approach. It was introduced in the Bayesian framework by Gelman, Meng, and Stern (1996) to perform posterior predictive assessment of model fitness, following earlier work by Rubin and Thayer (1983). While the classical approach to model-checking overemphasises the need to *check and test the correctness or trueness* of the proposed model, this new Bayesian counterpart focuses on *assessing the discrepancies between a model and the data.* The emphasis in this new method is therefore placed on the usefulness rather than correctness. The starting point is obviously the statement of the null hypothesis, which in this case is simply H_o : *The proposed model is correct.*

2.6.2 Details of the method

The method provides a new way of assessing the plausibility of the proposed model via the use of replicates and realised discrepancies. For our FA model with $\boldsymbol{x} \sim \mathcal{N}_p(\mu, \Lambda\Lambda^\mathsf{T} + \Sigma)$, we use a χ^2 type discrepancy measure, namely the sum of squares of standardised residuals

$$D(\mathbf{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} (\boldsymbol{x}_i - \mu)^\mathsf{T} \left[\Lambda\Lambda^\mathsf{T} + \Sigma\right]^{-1} (\boldsymbol{x}_i - \mu). \tag{2.46}$$

For the above discrepancy, $D(\mathbf{X}, \boldsymbol{\theta})$, we derive a reference distribution for the joint posterior distribution of $\boldsymbol{x}^{rep}$ and $\boldsymbol{\theta}$ given the proposed model H and the data $\mathbf{X}$.

$$\Pr(\boldsymbol{x}^{rep}, \boldsymbol{\theta} | \mathsf{H}, \mathbf{X}) = \Pr(\boldsymbol{x}^{rep} | \mathsf{H}, \boldsymbol{\theta}) \boldsymbol{p}(\boldsymbol{\theta} | \mathsf{H}, \mathbf{X}) \tag{2.47}$$

where $\Pr(\boldsymbol{x}^{rep} | \mathsf{H}, \boldsymbol{\theta})$ is the posterior predictive distribution of $\boldsymbol{x}^{rep}$ under the proposed model, while $\boldsymbol{p}(\boldsymbol{\theta} | \mathsf{H}, \mathbf{X})$ is the posterior density of the corresponding model parameters. The tail-area probability of $D(\mathbf{X}, \boldsymbol{\theta})$ under the derived posterior reference distribution

is then defined as follows:

$$\text{p-value} = \Pr(D(\mathbf{X}^{rep}, \boldsymbol{\theta}) > D(\mathbf{X}, \boldsymbol{\theta})|\mathsf{H}, \mathbf{X}). \tag{2.48}$$

For each of the iterates in the set $\{\boldsymbol{\theta}^{(t)} : t = 1, \cdots, M\}$, it is easy to simulate a replicate data set $\mathbf{X}^{rep(t)} = \{\boldsymbol{x}_1^{rep(t)}, \cdots, \boldsymbol{x}_n^{rep(t)}\}$ by drawing samples $\boldsymbol{x}_i^{rep(t)}$ from the posterior predictive distribution $\Pr(\boldsymbol{x}^{rep}|\boldsymbol{\theta}^{(t)}, \mathsf{H})$ of $\boldsymbol{x}^{rep}$, given the proposed model H and its corresponding parameters, after which realised discrepancies $D(\mathbf{X}, \boldsymbol{\theta}^{(t)})$ and $D(\mathbf{X}^{rep(t)}, \boldsymbol{\theta}^{(t)})$ are computed for both the original sample and the replicate respectively. Given the sample path $\{\boldsymbol{\theta}^{(t)} : t = 1, \cdots, M\}$, an empirical version of the above p-value is given by

$$\text{p-value} = \frac{1}{M}\sum_{t=1}^{M} \mathrm{I}(t : D(\mathbf{X}^{rep(t)}, \boldsymbol{\theta}^{(t)}) > D(\mathbf{X}, \boldsymbol{\theta}^{(t)})) \tag{2.49}$$

$$= \frac{1}{M}\#\{t : D(\mathbf{X}^{rep(t)}, \boldsymbol{\theta}^{(t)}) > D(\mathbf{X}, \boldsymbol{\theta}^{(t)})\} \tag{2.50}$$

which intuitively is the proportion of subsequent experiments that support the null hypothesis, and therefore serves to numerically assess the fitness of the proposed model. A graphical assessment is also obtained through a scatterplot of $D(\mathbf{X}^{rep(t)}, \boldsymbol{\theta}^{(t)})$ against $D(\mathbf{X}, \boldsymbol{\theta}^{(t)})$, and the p-value is in fact the proportion of points above the 45^o line.

Remark: It is worth stressing the point that the above test does not indicate whether a model is correct or not, but rather aims at *finding out whether there is evidence of lack of fitness.*

2.6.3 What makes Bayesian sampling appropriate?

It is worth pointing out that this assessment of model fitness is made easier because of the availability of MCMC samples: in fact, at each iteration of the MCMC process, each draw of the complete collection of model parameters actually defines a possible model. This can therefore be thought of as a realisation of a subsequent experiment that can generate other datasets (replicates in this case) from the posterior predictive distribution and then compute the corresponding test statistic (measure of discrepancy). On the other hand, while the theoretical specification of the reference distribution of the

test statistic can be very complicated if not impossible, the availability of MCMC sample paths makes it possible to specify an empirical approximation to this distribution as we shall see later. In other words, with the availability of MCMC sample paths, one can readily implement the idea (concept) of comparing subsequent experiments to the one that generated the data at hand, and therefore (at least empirically) decide whether the data at hand support the null hypothesis more than any other subsequently generated data. It is therefore fair to say that Bayesian sampling offers a complete platform to perform this posterior predictive assessment of model fitness via realised discrepancies.

2.6.4 Numerical results

Example 1 revisited: For this first example, we run $T_o = 6500$ burn-in iterations, after which we use subsampling with $c = 5$, and we finally retain a sample path of $M = 300$ draws. As expected, the restricted version of the algorithm produces satisfactorily accurate estimates, and does so regardless of the initial values[7]. However, the algorithm takes considerably longer than the EM algorithm, but has the following additional advantages: (a) it allows the easy computation of estimates of precision in the form of estimates of posterior standard deviations, and (b) provides a satisfactory assessment of the fitness of the proposed model. For economy of space, we only provide estimates of uniquenesses and their standard errors. In this case, the estimate is $\hat{\Sigma}_{\mathsf{DA}} = \mathsf{diag}(0.017, 0.184, 0.340, 0.022, 0.022, 0.166, 0.199, 0.401, 0.378)$, and its standard error estimates are $\mathsf{Error}(\hat{\Sigma}_{\mathsf{DA}}) = \mathsf{diag}(0.002, 0.028, 0.039, 0.003, 0.003, 0.023, 0.028, 0.030, 0.032)$. Figure (2.4) shows a reasonably large number of points above the 45^o line, and the p-value is 0.485. With a cut-off at 0.05 in the spirit of the significance level used in classical tests, $0.485 > 0.05$ is very supportive of the null hypothesis, suggesting that there is no evidence of lack of fitness. As expected, we can therefore conclude that the proposed model is a plausible fit of the data.

[7]Despite the fact that algorithm is insensitive to starting values, we make it a point to always use the same initial values as in the previous sections

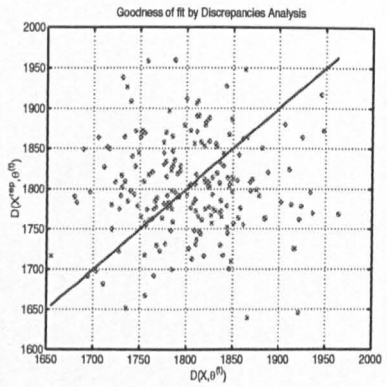


Figure 2.4: Scatterplot of realised discrepancies for Example 1.

Example 2 revisited: For this example, we used exactly the same T_o, c and M as before, and our aim in this case was simply to assess the fitness of the proposed model. Figure (2.5) shows a rather very small number of points above the 45^o line, and the corresponding p-value in this case is 0.03. If we use the same cut-off of 0.05 as earlier, the small p-value of $0.03 < 0.05$ can serve as a evidence to reject the null hypothesis, therefore suggesting lack of fitness of the proposed model. As expected, the single factor model is NOT a plausible candidate model to handle our data generated from a three-component population.

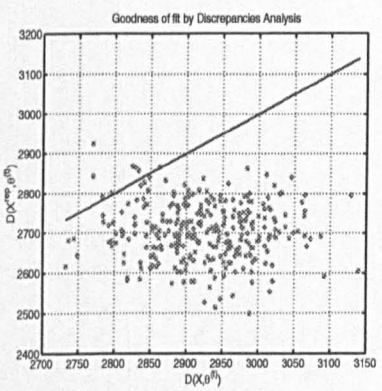


Figure 2.5: Scatterplot of realised discrepancies for Example 2.

2.7 Stochastic model selection for FA

While there are many cases in practice where the number of factors q is known and/or fixed, as we assumed earlier, it must be said that this value is very often unknown in real-life applications, and the study of the FA model therefore needs to address its uncertainty. At the root of model determination in Factor Analysis lies the difficult issue of finding and/or defining principled methods to decide what makes a particular factor important. In fact, for FA, this difficult problem of model determination has been one of the burning issues over the years, captivating the interests of researchers from both the likelihood-based and Bayesian perspectives. The reader is referred to such references as Krzanowski and Marriott (1994), Krzanowski and Marriott (1995) and Press (1972) for detailed coverage of these approaches.

2.7.1 A review of a classical empirical approach

The most widely used method is entirely based on the eigenvalues of the sample correlation matrix. While this very often produces satisfactory results as we saw in the previous sections, the fact of focusing only on the eigenvalues could lead to the neglect of vital information: almost all the criteria used to decide on the number of factors to retain are essentially ad hoc (eigenvalues less than 1) and often subjective (elbow of the screeplot) criteria that in some special cases would either overestimate or underestimate the adequate number of factors. For instance, if one variable is virtually independent of all the rest, it will appear as a separate component with variance slightly less than 1, but there is no reason to suppose that such a variable is uninformative. Thus, while this method may provide rough estimates of the number of factors, there is a clear need for more principled and objective model selection methods in this context. In the new edition of his book, Jolliffe (1986) offers a comprehensive coverage of this classical approach, along with recent developments of more principled methods.

2.7.2 Likelihood-based approach

From a classical likelihood-based standpoint, model selection in factor analysis simply consists of sequentially applying a series of likelihood ratio tests as described in section (2.4.4). In practice, one starts with $q = 1$ (single factor model), then fits successive values and tests the goodness-of-fit as in section (2.4.4), until the test produces a non-significant result indicating in a sense that the fit of the model is adequate. However, while this method appears as an objective procedure for estimating q, it is not strictly valid as a hypothesis test as argued by Krzanowski and Marriott (1995), since *no adjustment is made to the significance level to allow for its sequential nature.* On the other hand, the fact of having a non-significant p-value cannot be taken to indicate that the optimum value of q has been found, since large values of q correspond to more parameters and therefore better fits, obviously at the expense of more complex models and risks of overfitting. For the "best" model to be determined, there needs to be a trade-off between the number of parameters and the goodness-of-fit. In this likelihood-based framework, one way to determine the "best" model is to use Akaike's Information Criterion, which consists of selecting the model that minimises AIC as defined in (2.51).

$$\mathsf{AIC} = -2\log(\text{maximised likelihood}) + 2(\text{number of parameters fitted}). \qquad (2.51)$$

In the factor analysis context, the above criterion (2.51) is equivalent to choosing q that minimises $\omega - 2\nu$, as suggested by Akaike (1987), where ω and ν are respectively the test statistic and the number of degrees of freedom as defined in section (2.4.4). It has been noticed in practice that AIC tends to overfit models. In the analysis of mixtures for instance, AIC tends to overestimate the correct number of components. The Bayesian Information Criterion (BIC)of equation (2.52) is often used as an alternative to AIC.

$$\mathsf{BIC} = -2\log(\text{maximised likelihood}) + \log n(\text{number of parameters fitted}) \qquad (2.52)$$

The reason why BIC performs better than AIC can be explained simply as follows: the penalty term of BIC penalises complex models more heavily than AIC, whose penalty term does not depend on the sample size. BIC therefore reduces the tendency of the AIC criterion to overfit models.

2.7.3 Elements of stochastic model selection for FA

In this thesis, we adopt a stochastic simulation approach to model selection. This approach is based on the construction of an ergodic Markov chain having the posterior distribution of the complete collection of all the unknowns (parameters and q) as its equilibrium distribution.

When the dimension of the parameter space is known and fixed as we assumed earlier, traditional MCMC algorithms like the Gibbs sampler or the Metropolis-Hastings and their hybrid versions are used to construct the ergodic Markov chains of interest. However, if this dimension is allowed to vary throughout the MCMC iterative procedure, the classical algorithms mentioned earlier are no longer valid, and they have to be replaced by birth-and-death type algorithms capable of jumping between spaces of different dimensions.

In the Bayesian framework, Green (1995)'s Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm is one such algorithm. These algorithms make transitions based on extended versions of the classical MCMC detailed balanced requirement that take into account the varying dimensionality of the support of the parameters. Richardson and Green (1997) offer a detailed and comprehensive presentation of the application of RJMCMC to the Bayesian analysis of mixtures of univariate distributions with an unknown number of components. Lopes and West (1999) applied an adaptation of RJMCMC to the factor analysis model with an unknown number of common factors, and obtained good results on both synthetic and real-life problems. More recently, Stephens (2000), using ideas from stochastic geometry and spatial statistics, developed an alternative to RJMCMC, based on the simulation of a continuous-time birth-and-death Markov marked point process. He applied the derived Birth-and-Death MCMC (BDMCMC) method to mixtures of univariate and bivariate Gaussians with unknown numbers of components, and obtained promising results. Despite the fact that RJMCMC is based on a discrete-time Markov process while BDMCMC is based on a continuous time Markov process, the two methods are essentially equivalent in that they both successfully construct ergodic Markov chains in spaces of varying dimensions. In fact, BDMCMC can be thought of

as a limit of RJMCMC. However, for practical reasons and to a certain extent for computational convenience, we adopt an approach closer to BDMCMC. To the best of our knowledge, no one before us has treated model selection in factor analysis using an adaptation of Stephens (2000)'s BDMCMC as we do in this thesis. Our contribution in this context therefore offers an alternative to Lopes and West (1999)'s treatment and other existing methods of model selection in factor analysis. The first reason is the modularity and portability of the BDMCMC scheme: we note that, unlike with RJMCMC, ideas developed in BDMCMC and applied to mixtures can be readily adapted to model selection in FA. The fact that RJMCMC makes use of the latent variables is not appealing in our context in that it would be very complicated to apply it to a scheme with a mix of both continuous and categorical latent variables as we shall see in the subsequent chapters. From a computational point of view, we find death rates calculated on the basis of the "importance" (as measured by functions of the likelihood) of the component easier to interpret than RJMCMC's birth-and-death moves occurring uniformly. Finally, while RJMCMC has been extensively used in the analysis of mixtures of univariate distributions, its extension to mixtures of multivariate distributions still poses many difficulties, such as the complexity of the Jacobian calculations, and this prevents it from being a good candidate method for an essentially multivariate model like ours where we intend to analyse mixtures of multivariate distributions. Since the BDMCMC scheme treats parameters as points (no ordering) in a point process, it does not make use of such identifiability constraints as ordering, and its extension to multivariate distributions such as the ones of interest to us is therefore straightforward. Moreover, in contrast to the RJMCMC, the method requires very little mathematical sophistication and is easy to implement and interpret.

2.7.4 A point process view of Bayesian sampling

The central idea behind this approach is to view and treat each parameter that directly affects the dimensionality of the model as a point in the parameter space, and adapt the methodology of point process simulation to help construct a Markov chain with the

posterior distribution of the parameters as its equilibrium distribution. The method developed is therefore general and applicable to every context where parameters can be treated as point processes. Ideas used in the BDMCMC scheme are similar to those developed by Grenander and Miller (1994) and Phillips and Smith (1995) who approached this same problem of Bayesian model comparison via *jump diffusions.* However, it is fair to point out that the implementation of the schemes developed by Grenander and Miller (1994) and Phillips and Smith (1995) is more complicated than Stephens (2000)'s BDMCMC.

Definition: The mathematical definition of a point process [8] on $\mathbb{R}^d$ is as a random variable[9] taking values in a measurable space of families of all sequences $\varphi = \{v_1, v_2, \cdots, v_d\}$ of points in $\mathbb{R}^d$ satisfying two regularity conditions:

1. the sequence φ is locally finite (each bounded subset of $\mathbb{R}^d$ must contain only a finite number of points of φ),

2. the sequence is simple (with elements such that $v_i \neq v_j$ if $i \neq j$).

As we discussed earlier, FA has a posterior distribution that is invariant to permutations of the order of their parameters. From a stochastic simulation perspective, the collection of parameters can therefore be viewed as a random configuration or point process. This complete collection of our model parameters is now given by $\boldsymbol{\theta} = \{q, \mu, \Lambda, \Sigma\}$. If we assume that q is unknown a priori, our aim in parameter estimation from a stochastic simulation perspective now extends to the construction of an ergodic Markov chain with the joint posterior distribution $\boldsymbol{p}(q, \mu, \Lambda, \Sigma|\mathbf{X})$ as its equilibrium distribution. In the previous section, we constructed a Markov chain with $\boldsymbol{p}(\mu, \Lambda, \Sigma|q, \mathbf{X})$ as its equilibrium distribution using Data Augmentation. Now, we must accommodate the new *counting* random variable q. Intuitively, our overall sampling scheme takes a Gibbs sampler-like form, with each iteration consisting of the following two steps:

---

[8] See Stoyan, Kendall, and Mecke (1995) for a detailed version.

[9] Grenander and Miller (1994) used the term *random configurations* to refer to point processes.

Step 1: $q^{(t+1)} \sim \boldsymbol{p}(q|\mu^{(t)}, \Lambda^{(t)}, \Sigma^{(t)}, \mathbf{X})$.

Step 2: $(\mu^{(t+1)}, \Lambda^{(t+1)}, \Sigma^{(t+1)}) \sim \boldsymbol{p}(\mu, \Lambda, \Sigma|q^{(t+1)}, \mathbf{X})$,

---

In the above scheme, **Step 1** allows us to draw a new value of $q = q^{(t+1)}$ by simulating a birth-and-death Markov point process, the main difference with a classical algorithm of this type being that the dimension of the parameter vector is allowed to vary at each iteration. **Step 2** draws a new set of model parameters via Data Augmentation (as described in Algorithm 2), using the value of q obtained from the run of the birth-and-death process.

The simulation of the type of birth-and-death process that we use in our work has been extensively studied and applied in recent years, and the reader is referred to references like Stoyan, Kendall, and Mecke (1995) and Barndorff-Nielsen, Kendall, and van Lieshout (1999) for comprehensive coverage of applications of such sampling schemes in stochastic geometry and spatial statistics. Baddeley (1994) and van Lieshout (1994) also provide very useful insights into other aspects of such sampling schemes. Stephens (2000) provides a detailed account of his application of BDMCMC to mixtures. Here we focus on our adaptation of BDMCMC to factor analysis.

2.7.5 Birth-and-death point process for Factor Analysis

From our previous arguments, the number of common factors is nothing but the number of columns of Λ. We showed earlier that $\Lambda\Lambda^{\mathsf{T}} + \Sigma$ is invariant to permutations of axes in Λ. From a probabilistic perspective, the invariance to permutations allows us to treat Λ as a "random configuration" or point process as defined earlier. For simplicity, we adopt a vector notation for Λ by defining the configuration variable $\mathbf{c} = \{\Lambda_{.1}, \Lambda_{.2}, \cdots\}$ [10]. Our aim being to construct a Markov chain with $\boldsymbol{p}(q, \Lambda, \mu, \Sigma|\mathbf{X})$ as its stationary distribution, we also simplify further and use $\boldsymbol{h}(\mathbf{c})$ in place of the posterior $\boldsymbol{p}(\cdot|\mathbf{X})$, since our emphasis in this context is on the configuration $\mathbf{c}$. Assuming fixed hyperparameters

[10] By configuration here, we have in mind the complete collection of distinct parameters (μ, Λ, Σ) for a given value of q. However, since only Λ plays a role in the determination of the complexity of the factor model, we explicitly only show Λ for simplicity.

ϱ and ι for the densities of q and $\mathbf{c}$ respectively, we can write

$$\boldsymbol{h}(\mathbf{c}) \propto \boldsymbol{L}(\mathbf{c})\boldsymbol{p}(q|\varrho)\boldsymbol{p}(\mathbf{c}|\iota) \tag{2.53}$$

It turns out that we can efficiently construct our ergodic Markov chain by simulating a sampling scheme comprising a birth-and-death point process step and a Data Augmentation step, both jointly converging to $\boldsymbol{p}(q, \Lambda, \mu, \Sigma|\mathbf{X})$ as the stationary distribution. The key idea behind the simulation of the birth-and-death process is that each birth increases the number of points in the configuration by one, while each death decreases this number by one. Furthermore, both the birth and the death processes are constructed in such a way that they are inverse operations to each other in the equilibrium state of the chain. One way to construct such a process is to define births and deaths[11] as follows:

Births: We define a birth density $b(\mathbf{c}; \upsilon)$ according to which new points are added to the current configuration of the point process. For simplicity, we restrict ourselves to cases where births are assumed to be occurring at an overall constant rate $\beta(\mathbf{c}) = \beta$. However, as we shall see in our simulations, such a simplification has the disadvantage that many different birth rates have to be tried empirically before the "appropriate" one is found.

Deaths: When the current configuration of the chain is $\mathbf{c} = \{\Lambda_{.1}, \Lambda_{.2}, \cdots\}$, each point $\Lambda_{.i}$ dies independently of the others as a Poisson process with rate $\delta_i(\mathbf{c}) = d(\mathbf{c}; \Lambda_{.i})$, where $d(\mathbf{c}; \upsilon)$ is the death density function, so that the overall death rate is given by $\delta(\mathbf{c}) = \sum \delta_i(\mathbf{c})$.

Remark: In their simulations of similar birth-and-death point processes, Ripley (1977) and van Lieshout (1994) have found it more convenient to define births and deaths in an alternative way, namely: use an overall constant death rate and instead compute the birth rate using information from the data. In her application of the similar scheme to image analysis, van Lieshout (1994) provides a more general definition where both the birth rate and the death rate are computed from the data.

[11] The general practice consists of imposing suitable constraints on the birth and death functions b and d to ensure that the process does not jump to an area with zero density.

With $\beta(\mathbf{c})$ and $\delta(\mathbf{c})$ defined, we use the following general theorem (stated without proof) on Poisson processes to obtain the distribution of the time to the next event in the simulation of the birth-and-death process.

Theorem 2.1 *The birth and the death being independent Poisson processes, the time to the next event (birth or death) is exponentially distributed with mean* $1/(\beta(\mathbf{c}) + \delta(\mathbf{c}))$.

Property 2.1 *Since the overall rate of the birth-and-death process is equal to* $\beta(\mathbf{c})+\delta(\mathbf{c})$, *the next event will be a birth with probability* $\beta(\mathbf{c})/(\beta(\mathbf{c}) + \delta(\mathbf{c}))$, *while the death of* $\Lambda_{\cdot i}$ *will occur with probability* $\delta_i(\mathbf{c})/(\beta(\mathbf{c}) + \delta(\mathbf{c}))$.

We are therefore in the presence of a continuous-time process since the time to the next event is a continuous random variable, and, by virtue of the *memorylessness* property of the exponential distribution, we have a continuous time Markov process. In order to simulate such a continuous time process, we define a fixed unit of time, ρ, say, and we construct a discrete-time Markov chain $\{\mathbf{c}^{(\rho)}, \mathbf{c}^{(2\rho)}, \mathbf{c}^{(3\rho)}, \cdots\}$ that we use as an approximation of the continuous-time chain $\{\mathbf{c}^{(\rho+s)} : s > 0\}$. This simply means that, at each discrete iteration $(t = 1, \cdots, T)$, we run the birth-and-death process for a duration of ρ. Preston (1976) stated sufficient conditions that the above densities b and d must satisfy for the above birth-and-death process to define an ergodic Markov chain with the desired equilibrium distribution. Preston (1976)'s work was later extended and applied by Ripley (1977), and recently adapted to the analysis of finite mixtures by Stephens (2000). The following theorem, which states the sufficient conditions that b and d must satisfy, is from Preston (1976) and Ripley (1977). A proof of its extended version as applied to finite mixtures can be found in Stephens (2000).

Theorem 2.2 *If the birth density* b *and the death density* d *satisfy*

$$(q + 1)d(\mathbf{c} \cup \{v\}; v)\boldsymbol{h}(\mathbf{c} \cup \{v\}) = \beta(\mathbf{c})b(\mathbf{c}; v)\boldsymbol{h}(\mathbf{c}) \tag{2.54}$$

for all configurations $\mathbf{c}$ *and all points* v, *then the birth-and-death process defined above has* $\boldsymbol{p}(q, \Lambda, \mu, \Sigma|\mathbf{X})$ *as its stationary distribution.*

Remark: In the above theorem, $h(\mathbf{c} \cup \{v\})$ represents the posterior density of a configuration with $q+1$ points. Intuitively, equation (2.54) means that, under the equilibrium distribution $p(\cdot|\mathbf{X})$, transitions from $\mathbf{c}$ into $\mathbf{c} \cup \{v\}$ are exactly matched by transitions from $\mathbf{c} \cup \{v\}$ into $\mathbf{c}$. From equation (2.54), it is easy to see that

$$d(\mathbf{c}; v) = b(\mathbf{c}; v) \left[\frac{\beta(\mathbf{c})}{q}\right] \left[\frac{h(\mathbf{c}\backslash\{v\})}{h(\mathbf{c})}\right] \tag{2.55}$$

where $\mathbf{c}\backslash\{v\}$ represents the current configuration $\mathbf{c}$ less the element v. If we choose our birth density to be the prior density of a candidate element v to be added to the current configuration, then we can write $b(\mathbf{c}; v) = p(v|\iota)$. If we use (2.53) and (2.55), it is easy to see that the appropriate death rate for element $\Lambda_{\cdot i}$ $(i = 1, \cdots, q)$ is given by

$$\delta_i(\mathbf{c}) = \left[\frac{\beta}{q}\right] \left[\frac{L(\mathbf{c}\backslash\Lambda_{\cdot i})}{L(\mathbf{c})}\right] \left[\frac{p(q-1|\varrho)}{p(q|\varrho)}\right] \tag{2.56}$$

As far as the prior distribution of q is concerned, a good candidate is a Poisson distribution truncated at the right end by a predetermined value q_{max}. Thus, we have

$$p(q|\varrho) \propto \frac{\varrho^q}{q!} e^{-\varrho} \quad \text{for} \quad q = 1, \cdots, q_{max} \tag{2.57}$$

Based on all the above ingredients, a pseudocode of the birth-and-death process is.

Algorithm 3: Birth-and-death MCMC for FA.

Use $\beta(\mathbf{c}) := \beta$, set $t_{fa} = 0$ and $q := q^{(t-1)}$ from $\mathbf{c}^{(t-1)}$
Repeat
  Compute $\delta_j(\mathbf{c}) := \dfrac{L(\mathbf{c}\backslash\Lambda_{\cdot j})}{L(\mathbf{c})} \dfrac{\beta}{\varrho}$ for $j = 1, \cdots, q$
  Compute $\delta(\mathbf{c}) := \sum_{j=1}^{q} \delta_j(\mathbf{c})$
  Simulate $s \sim \text{Exp}(1/((\beta(\mathbf{c}) + \delta(\mathbf{c})))$ and Set $t_{fa} := t_{fa} + s$
  If $(\text{Ber}(\beta(\mathbf{c})/(\beta(\mathbf{c}) + \delta(\mathbf{c})) = 1)$ /* It is a birth */
    Set $q := q + 1$
    Simulate $\Lambda_{\cdot q} \sim p(v|\iota)$
    Set $\mathbf{c} := \mathbf{c} \cup \{\Lambda_{\cdot q}\}$
  Else /* It is a death */
    Simulate $j' = \text{Mn}(\delta_1(\mathbf{c})/\delta(\mathbf{c}), \cdots, \delta_q(\mathbf{c})/\delta(\mathbf{c}))$
    Set $\mathbf{c} := \mathbf{c}\backslash\{\Lambda_{\cdot j'}\}$
    Set $q := q - 1$
Until $(t_{fa} \geq \rho)$
Return $\mathbf{c}$ and q[12].

---

[12]In reality, the knowledge of $\mathbf{c}$ implies the knowledge of q, but we indicate both for the sake of clarity

Ber, Exp and Mn represent the Bernoulli, the exponential and the multinomial distributions respectively. The full stochastic simulation scheme for FA is therefore as follows:

Algorithm 4: Stochastic simulation for FA.

Initialise q and $\mathbf{c}$, and choose $\beta(\mathbf{c}) = \beta$.
For $t = 1, \cdots, T$
  Run Algorithm 3 using $q^{(t)}$ and $\mathbf{c}^{(t)}$, and return $q^{(t+1)}$ and $\mathbf{c}^{(t+1)}$
  Run Algorithm 2 to update the complete collection of model parameters.
End

2.7.6 Bayesian inference for q

Once the Markov chain $\{(q^{(t)}, \mathbf{c}^{(t)}) : t = 1, \cdots, T\}$ has converged to the desired equilibrium distribution, the sequence $\{q^{(t)} : t = 1, \cdots, T\}$ is essentially a sequence of draws from the marginal distribution $\boldsymbol{p}(q|\mathbf{X})$. Inference for q can be based on an estimate of this marginal posterior distribution obtained from the MCMC sample path as follows:

$$\Pr[q = i|\mathbf{X}] = \lim_{M\to\infty} \frac{1}{M}\#\{t: \quad q^{(t)} = i\} \approx \frac{1}{M}\#\{t: \quad q^{(t)} = i\} \tag{2.58}$$

Intuitively, (2.58) simply means that the appropriate estimate for q is obtained by choosing the value of q having the highest frequency in the sample path of the Markov chain.

2.8 Implementation and Results

In this section, we present two simulations, one based on the real-life and moderately high-dimensional ($p = 13$) wine dataset[13], and the other based on a synthetic dataset that we generated to illustrate our methods. All our simulations are written in Matlab 6.0 for Unix. By personal preference, we use the golden section [14] or its multiples wherever we need an arbitrary real constant. From time to time, we also use multiples of the inverse of the golden section as our "arbitrary" constants.

[13]This data set is available at the Machine Learning repository of the University of California, Irvine Blake and Merz (1998)

[14]$(\sqrt{5}-1)/2 = 0.61803....$ The golden section is also known as the golden ratio, the golden mean and sometimes the divine proportion. It is closely connected with the Fibonacci series. The inverse of the golden section is $2/(\sqrt{5}-1)$, and has the property $2/(\sqrt{5}-1) = 1+(\sqrt{5}-1)/2 = (\sqrt{5}+1)/2 = 1.61803....$

2.8.1 Example 3: Analysis of the wine data set

For the wine dataset, it is believed that there are $k = 3$ distinct varieties of wine. Our first aim here is therefore to use a single factor model to explore the group structure in the data, so as to (subjectively at least) find out whether there are actually 3 groups in the provided dataset. Our second aim is to use our BDMCMC for FA to estimate the intrinsic dimensionality of the data, assuming that q is the same in all the groups. As far

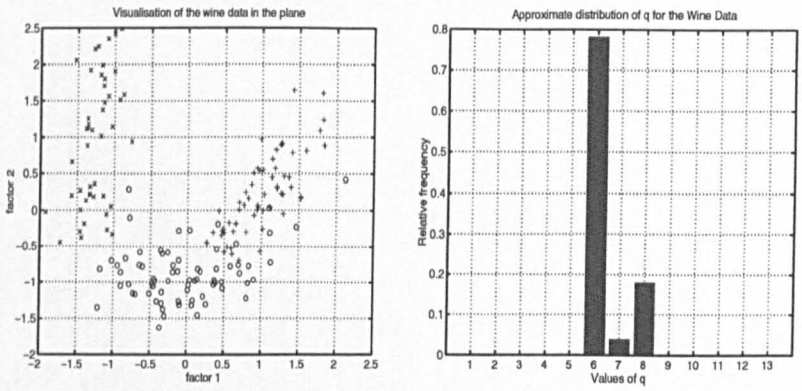


Figure 2.6: 2D Visualisation and histogram for the wine data.

as group structure is concerned, we run Algorithm 2 assuming $q = 2$ to project the data on to the plane and therefore explore its group structure. We plot the estimated factor scores as shown in (Figure 2.6-left). At least in the plane, the figure seems to agree with the hypothesis that there could be 3 classes of wines. We shall be reconsidering this example in the next chapter when we study mixtures of factor analysers. To estimate q, we run Algorithm 4, using $T_o = 9500$ burn-in iterations, $\beta = 0.618$ as our overall constant birth-rate, and $M = 2500$ useful final MCMC samples. (Figure 2.6-right) shows the histogram of the relative frequencies of values of q as produced by the sample path of the Markov chain obtained from the BDMCMC. This strongly suggests that $q = 6$ would be the intrinsic dimensionality of the wine data. The good news is that the result obtained here by stochastic simulation is consistent with the one obtained by McLachlan and Peel (2000) through the use of sequential likelihood ratio tests.

2.8.2 Example 2 revisited: Analysis of Simulated data

This second example is purely illustrative, and is based on simulated data. We generate a synthetic data set from a Mixture of Factor Analysers with $k = 3$, $p = 9$ and $q = 2$. Our sole aim in this example was to test Algorithm 4 on a toy problem. With $T_o = 12000$

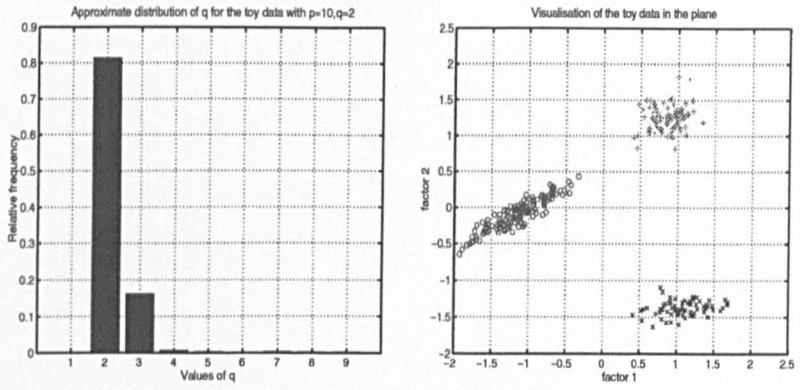


Figure 2.7: Histogram and scatterplot for the data with $k = 3$, $p = 9$ and $q = 2$.

burn-in iterations and $M = 2000$ useful MCMC final samples, the algorithm easily infers $q = 2$ as shown in (Figure 2.7-left). The visualisation of estimated factor scores in (Figure 2.7-right) clearly reveals that there are indeed 3 groups.

2.8.3 Simulation remarks

Our simulations reveal that the BDMCMC algorithm is insensitive to initial conditions and always infers the right number of common factors regardless of whether we start the chain with one factor ($q = 1$) or with many factors ($q = q_{max}$) [15]. However, since we are in a setting where we wish to determine the smallest number of factors consistent with the data, we prefer to start the chain with $q = 1$ and let it create more as becomes necessary.

[15] Although convergence to the right number of factors is still achieved from this starting value, it must be said that it results in more iterations and therefore a computationally less efficient procedure.

2.9 Discussion and future work

2.9.1 General comments

The performance of the stochastic simulation scheme for model selection is satisfactory when applied to the single factor analysis model. However, it is fair to point out that the choice of the overall constant birth rate is crucial, and does have a strong bearing on the mixing properties of the Markov chain. In practice, some birth rates do allow the chain to mix very well, making it possible to visit as many potential models as possible. Unfortunately, some other choices of birth rates cause the chain to remain only in some areas of the entire parameter space. One of the solutions to this problem consists of adding a small uniform perturbation to the birth rate at every iteration. We have implemented this empirical solution, and it generally leads to remarkable improvement. On the other hand, in the stochastic geometry literature, there are many variants of the basic algorithm that we have used here. We consider exploring some of those variants in our future research. Our results suggest that the stochastic simulation method we have proposed for model selection is a good alternative to existing methods. We believe that a careful study of the limitations noticed so far would lead to remarkable improvements. Besides, the Bayesian sampling approach to factor analysis, despite being slower that its counterparts, offers many advantages that would justify resorting to it as a valid competitor to existing approaches. Lopes and West (1999) amongst others have used Bayesian sampling to tackle factor analytic model uncertainty, with applications to such fields as **analysis of exchange rates** and **portfolio management**, and we believe that our overall scheme has the potential to address such real applications.

We have explored aspects of the Factor Analysis model from both the frequentist and Bayesian perspectives. Both approaches have their strengths and weaknesses[16], and the decision to adopt either one heavily depends on the context, and can in many cases become a simple matter of taste. However, it is fair to say that our Bayesian approach

[16]It is my opinion from practical experimentation that the alleged slowness of the EM algorithm is very often overstated. In fact, in many practical settings, the EM algorithm can be very quick at providing satisfactory parameter estimates.

based on stochastic simulation offers a flexible alternative to existing methods, unifying parameter estimation, assessment of model fitness and model selection in a single efficient scheme. The good performance of the scheme on both artificial and real data strongly encourages an exploration of improvements and extensions of the scheme.

2.9.2 Beyond the single linear factor model

As we saw with **Example 2**, a single factor model is inherently linear, and therefore fails to capture the generative mechanism underlying the data when there is structural nonlinearity in it. It is therefore more reasonable and more realistic to use models capable of handling nonlinearity. In the Neural Computation community, the particular task of dimensionality reduction that FA performs is sometimes done by MultiLayer Perceptrons, which are essentially nonlinear tools. In the next chapter, we introduce and explore an alternative solution based on the Mixture of Factor Analysers model which is an extension of the basic FA model constructed in such a way that it can handle nonlinearity.

Chapter 3

Mixtures of Factor Analysers

The most important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.
Sir William Bragg

A shortened version of the content of this chapter has been accepted for publication Fokoué and Titterington (2001) in the special issue of *Machine Learning* on MCMC.

In the previous chapter, we studied various aspects of the classical Factor Analysis (FA) model, and we noted that besides the mainstream statistics and the psychometrics communities, FA has over the years been recognised by the Machine Learning and Neural Computation communities as a well established probabilistic approach to unsupervised learning for complex systems involving correlated variables in high-dimensional spaces. As we saw earlier, FA aims principally to reduce the dimensionality of the data by projecting high-dimensional vectors on to lower-dimensional spaces. However, because of its inherent linearity, the generic FA model is unable to capture data complexity when the input space is nonhomogeneous. One way to overcome the limitation due to the inherent linearity of the FA model would be to resort to a nonlinear version of it by constructing an extended model in which the manifest variable would be a nonlinear combination of factors. Another way would be to construct an extended model that would allow the data in each cluster of the nonhomogeneous input space to be modelled by a local FA model, thus creating a finite Mixture of Factor Analysers (MFA). A finite Mixture of Factor Analysers (MFA) is a globally nonlinear and therefore more flexible extension of the ba-

sic FA model that overcomes the above limitation by combining the local factor analysers of each cluster of the heterogeneous input space. The structure of the MFA model offers the potential to model the density of high-dimensional observations adequately while also allowing both clustering and local dimensionality reduction. Many aspects of the MFA model have recently come under close scrutiny, from both the likelihood-based and the Bayesian perspectives. In this chapter, we will touch on elements of recent developments concerning the MFA model, but we shall mainly focus our attention on the Bayesian approach, and more specifically on a treatment that bases estimation and inference on the stochastic simulation of the posterior distributions of interest. We first treat the case where the number of mixture components and the number of common factors are known and fixed, and we derive an efficient Markov Chain Monte Carlo (MCMC) algorithm based on Data Augmentation to perform inference and estimation. We also consider the more general setting where there is uncertainty about the dimensionalities of the latent spaces (number of mixture components and number of common factors *unknown*), and we estimate the complexity of the model by using the sample paths of an ergodic Markov chain obtained through the simulation of a continuous-time stochastic birth-and-death point process. As we noted in the previous chapter, the main strengths of our algorithms are that they are both efficient (our algorithms are all based on familiar and standard distributions that are easy to sample from, and many characteristics of interest are by-products of the same process) and easy to interpret. Moreover, they are straightforward to implement and offer the possibility of assessing the fitness of the model via the use of realised discrepancies. Experimental results on both artificial and real data reveal that our approach performs well, and can therefore be envisaged as an alternative to the other approaches used for this model. Before studying Mixtures of Factor Analysers, it makes sense to present a brief review of finite mixture models.

3.1 Introduction to finite mixtures of distributions

Finite mixture models provide another rich class of latent variable models that are heavily used in statistical modelling, and that have been extensively studied in recent years by many scientific communities for a variety of practical applications. The use of finite mixture models is particularly relevant to applications where the input space is assumed to be nonhomogeneous (heteregeneous), so that it would be unrealistic to use a single density function to model the distribution of the data. Mixture models allow the representation of the density function as a weighted sum of component densities, thereby making it possible to take into account the heterogeneity of the input space. Mixture models can therefore be used for density estimation as an alternative to traditional nonparametric kernel density estimators. The analysis of finite mixtures is a vast topic. In this section, we only review some of the key issues related to mixtures, and the reader is referred to such references as Titterington, Smith, and Makov (1985), Everitt and Hand (1981) and McLachlan and Peel (2000) for more detailed presentations. Chapter 9 of Robert and Casella (2000) provides a recent account of the analysis of finite mixtures by MCMC methods.

3.1.1 Definitions, concepts and notations

Our basic definition of finite mixtures is taken from Titterington, Smith, and Makov (1985). Suppose that a random variable or random vector, $\boldsymbol{x}$, takes values in a sample space $\mathcal{X}$, and that its distribution can be represented by a probability density function of the form

$$\boldsymbol{p}(\boldsymbol{x}) = \pi_1 \boldsymbol{f}_1(\boldsymbol{x}) + \cdots + \pi_k \boldsymbol{f}_k(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j \boldsymbol{f}_j(\boldsymbol{x}) \qquad (\boldsymbol{x} \in \mathcal{X}) \tag{3.1}$$

where

$$\pi_j > 0, \qquad j = 1, \cdots, k; \quad \pi_1 + \cdots + \pi_k = \sum_{j=1}^{k} \pi_j = 1,$$

and

$$\boldsymbol{f}_j(\cdot) \geq 0, \qquad \int_{\mathcal{X}} \boldsymbol{f}_j(\boldsymbol{x})d\boldsymbol{x} = 1, \qquad j = 1, \cdots, k.$$

In such a case, we shall say that $\boldsymbol{x}$ has a *finite mixture distribution* and that $\boldsymbol{p}(\cdot)$, defined by (3.1), is a *finite mixture density function.* The parameters $\pi_1, \cdots, \pi_k$ will be called the *mixing weights* or *mixing proportions*, and $\boldsymbol{f}_1(\cdot), \cdots, \boldsymbol{f}_k(\cdot)$ will be referred to as the *component densities* of the mixture. It is straightforward to verify that (3.1) does indeed define a valid p.d.f. In many situations, $\boldsymbol{f}_1(\cdot), \cdots, \boldsymbol{f}_k(\cdot)$ will have specified parametric forms and the right-hand side of (3.1) will have the more explicit representation

$$\pi_1 \boldsymbol{f}_1(\boldsymbol{x}|\phi_1) + \cdots + \pi_k \boldsymbol{f}_k(\boldsymbol{x}|\phi_k) = \sum_{j=1}^{k} \pi_j \boldsymbol{f}_j(\boldsymbol{x}|\phi_j), \tag{3.2}$$

where ϕ_j denotes the parameters occurring in $\boldsymbol{f}_j(\cdot)$. The complete collection of model parameters $\boldsymbol{\theta}$ will therefore be $\boldsymbol{\theta} = (\pi_1, \cdots, \pi_k, \phi_1, \cdots, \phi_k)$. With a slight abuse of notation, we shall write $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_k)$, $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_k)$ and then write $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\phi})$.

Example: If the probability density function of a univariate random variable $\boldsymbol{x}$ can be represented by a two-component mixture of Normal densities with common variance, then we can write

$$\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta}) = \pi \mathcal{N}(\boldsymbol{x}; \mu_1, \sigma) + (1 - \pi)\mathcal{N}(\boldsymbol{x}; \mu_2, \sigma). \tag{3.3}$$

In this case, $\pi_1 = \pi$, $\pi_2 = 1 - \pi$, $\phi_1 = (\mu_1, \sigma)$, $\phi_2 = (\mu_2, \sigma)$, $\boldsymbol{\phi} = (\mu_1, \mu_2, \sigma)$, and the complete collection of model parameters is therefore $\boldsymbol{\theta} = (\pi, \mu_1, \mu_2, \sigma)$.

Note: In both univariate and multivariate settings, *mixtures of normal densities*, also known as *mixtures of Gaussians* or *Gaussian mixtures* have been extensively studied over the years, since they have a wide range of applications. Finite mixtures of Gaussians are indeed the most frequently encountered form of finite mixture distributions.

While there is no requirement that the component densities appearing in (3.2) should all belong to the same parametric family, it must be said that in most applications, this will be the case, and the finite mixture density function will then have the form

$$\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{j=1}^{k} \pi_j \boldsymbol{f}(\boldsymbol{x}|\phi_j), \tag{3.4}$$

where $\boldsymbol{f}(\boldsymbol{x}|\phi_j)$ denotes a generic member of the parametric family under consideration.

3.1.2 General mixture densities

In the case of a finite mixture model defined by (3.4), each of $\phi_1, \cdots, \phi_k$ is an element of the same parameter space Θ. It follows that $\pi = (\pi_1, \cdots, \pi_k)$ may be thought of as defining a probability distribution over Θ, with $\pi_j = \Pr(\phi = \phi_j)$, for $j = 1, \cdots, k$. If $G_\pi(\cdot)$ denotes the probability measure over Θ defined by π, then (3.4) may be formally rewritten as

$$p(x|\theta) = \int_\Theta f(x|\phi) dG_\pi(\phi). \tag{3.5}$$

Throughout our work, we only deal with finite mixtures, which correspond to cases where $G_\pi(\cdot)$ defines a finite, discrete measure over Θ. It is however worth mentioning that equation (3.5) suggests a generalisation to *general mixture densities* by allowing $G_\pi(\cdot)$ to be a more general form of measure over Θ.

3.1.3 Latent structure formulation

Another formulation of finite mixture models, which we shall make extensive use of throughout our work is the latent structure formulation. In fact, given an observation x, a finite mixture model assumes that x was generated from one of k subpopulations, each containing a proportion π_j of elements of the wider population. Each subpopulation is also known as a *component* of the mixture, and can be viewed as a cluster in the input space. Since the component from which the observation originated is not directly observable, it is usually convenient to define a discrete random latent variable y that identifies such a component or cluster, and its distribution is given by $\mathbf{Pr}(y = j) = \pi_j$ for $j = 1, \cdots, k$. If we use the indicator variable version of y (i.e $y^\mathsf{T} = (y_1, \cdots, y_k)$ as defined in chapter 1) it is easy to see that y has a multinomial distribution, $y \sim \mathsf{Mn}(1; \pi_1, \cdots, \pi_k)$. Here, π_j represent the probability that the observation x comes from source j, and play the same role as the mixing proportions or weights that we encountered earlier. We obviously have $\pi_j > 0$ for $j = 1, \cdots, k$, and $\sum \pi_j = 1$. In each subpopulation, x has a specific class conditional density (the component density defined earlier) given by $p(x|y = j)$. With $p(x, y) = p(x|y)p(y)$ and $dP(y) = p(y)dy$, the marginal density

of $\boldsymbol{x}$ is therefore

$$\begin{aligned} p(\boldsymbol{x}) &= \int_{\mathcal{H}} p(\boldsymbol{x},\boldsymbol{y})d\boldsymbol{y} = \int_{\mathcal{H}} p(\boldsymbol{x}|\boldsymbol{y})p(\boldsymbol{y})d\boldsymbol{y} = \int_{\mathcal{H}} p(\boldsymbol{x}|\boldsymbol{y})dP(\boldsymbol{y}) \\ &= \sum_{j=1}^{k} \mathbf{Pr}(\boldsymbol{y}=j)p(\boldsymbol{x}|\boldsymbol{y}=j) = \sum_{j=1}^{k} \pi_j p(\boldsymbol{x}|\boldsymbol{y}=j). \end{aligned} \tag{3.6}$$

As we shall see later, it will be more convenient in most cases to use the complete-data density instead of the above marginal density of the observed-data $\boldsymbol{x}$. The complete-data density for our k-component finite mixture can be written as

$$p(\boldsymbol{x}^*) = p(\boldsymbol{x},\boldsymbol{y}) = \prod_{j=1}^{k} [\pi_j p(\boldsymbol{x}|\boldsymbol{y}=j)]^{y_j} = \prod_{j=1}^{k} \pi_j^{y_j} [p(\boldsymbol{x}|\boldsymbol{y}=j)]^{y_j} \tag{3.7}$$

3.1.4 Aspects, aims and issues in finite mixtures

Clustering: Statistical methods used to analyse finite mixtures generally deliver the clustering of the data in the form of estimates of the expected latent scores $\mathbb{E}[\boldsymbol{y}|\boldsymbol{x}]$.

Discriminant analysis and classification: When the components of a mixture have some physical meaning (interpretation), finite mixtures can be used in pattern recognition as a classification tool.

Parameter estimation: The estimation of parameters in a finite mixture has to overcome the identifiability hurdle that we will be explaining later.

Density estimation: Traditionally, nonparametric methods tend to be preferred when it comes to density estimation. However, finite mixtures can, in some settings, serve as an alternative to these classical nonparametric approaches.

Model selection: To date, determining or estimating the number of components is one of the most complex topics in the analysis of finite mixtures. From a likelihood-based perspective, the task is very similar to the one we encountered with the FA model. Essentially, it consists of a test of significance similar to the one used for goodness-of-fit. The bad news is that such a test is not easy to construct, and to date, the field remains virtually unexplored because of extreme mathematical difficulties in deriving appropriate test statistics and their corresponding reference distributions. Because of these limitations, we adopt a Bayesian approach similar to the one used in Chapter 2.

3.2 Difficulties with finite mixtures

The statistical analysis of finite mixtures generally encounters structural, inferential and computational difficulties, especially when parameter estimation is the main aim of the analysis. In this section, we briefly review some of the most common difficulties.

3.2.1 Identifiability

In this section, we simply give an intuitive definition of identifiability. A more formal and more mathematically grounded definition is provided by Titterington, Smith, and Makov (1985), pages 35-37. In fact, identifiability is inherently linked with parameter estimation, which has to do with the characterisation of the proposed model. In Chapter 1, we presented the general issue of identifiability, and we encountered it again in Chapter 2. When one thinks of parameter estimation, there is an underlying assumption that there exists a unique set of parameters that characterise the model, and the aim of parameter estimation is therefore the determination of that unique set of parameters. This uniqueness is an aspect of identifiability, and when such a unique set does not exist, the model under consideration is said to be non-identifiable. Finite mixtures are essentially non-identifiable, because the value of the density $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})$ remains unchanged for all the $k!$ permutations of the component labels y_j. Thus, if $\boldsymbol{\theta}' = \varsigma(\boldsymbol{\theta})$ is a new set of parameters obtained by a permutation of labels, then $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta}') = \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})$. In other words, given a sample $\mathbf{X}$ assumed to have arisen from a k-component mixture, there are potentially $k!$ different collections of parameters that would equally "characterise" the model. This is obviously not a desirable situation, since one would like to unambiguously determine a unique set of model parameters. As we shall see later, this difficulty, which is structural, also leads to another difficulty which is computational in nature, namely the label switching problem that constitutes one of the bottlenecks of Bayesian sampling for finite mixtures. Many approaches have been used so far to tackle this difficult issue, and we shall touch on some recent solutions later. However, it seems that no fully satisfactory solution exists as yet.

Remark. It is worth stressing that identifiability is such a crucial issue only when parameter estimation (model characterisation) is the aim of the analysis. When finite mixtures are used as alternatives to traditional nonparametric density estimators, the labelling of components is not a critical issue.

Note: Without loss of generality, we shall from now on base all our developments on finite mixtures of Gaussians. In many cases, general aspects apply *mutatis mutandis* to other forms of mixtures.

3.2.2 Unbounded likelihood and singularities

The likelihood function for a Gaussian mixture is unbounded, and allows the existence of singularities when an iterative method like the EM algorithm is used for maximum likelihood estimation (MLE). A simple and widely used illustration of this arises in a mixture of univariate normal distributions where the likelihood tends to infinity if we set μ_1 to $\boldsymbol{x}_1$ and allow σ_1^2 to tend to zero Everitt and Hand (1981). In other words, the first component of such a mixture only contains a single point $\boldsymbol{x}_1$, whatever the number of iterations performed, thereby yielding a partitioning of the population that is meaningless and clearly not of any interest. This aspect of the likelihood function affects both likelihood-based and Bayesian methods of estimation. In fact, in the Bayesian sampling framework, the use of a standard hierarchical prior structure for mixtures of Gaussians often leads to situations where a given component is allocated a very small number of observations, resulting in an almost zero probability for that component to be allocated more observations, or to have some of its few observations allocated to any other component. In fact, it turns out that these almost-absorbing states in MCMC are the analogues of the singularities encountered in the MLE approach. In other words, if a component variance σ_j^2 is allowed to become extremely small (i.e term of very small magnitude with respect to machine precision) at any given sample point, then that component of the mixture will be allocated that single point, with no chance of having any other point allocated to it, since the fixed hyperparameter will obviously never change the state of the chain. In practice, many devices are used to circumvent this

difficult issue, one of them consisting of constraining all the component variances to be equal. However, such a constraint is clearly unrealistic in the majority of cases. While the likelihood-based approach has to rely on ad hoc methods to tackle this issue, the Bayesian framework makes it possible to provide a more principled solution along the lines of Richardson and Green (1997), a solution that consists of adding an extra layer to the hierarchical prior structure in order to allow the hyperparameters of the variances (or covariance matrices) of the components of the mixture to be stochastic quantities. Such an extension allows the covariance matrices to be *similar* without constraining them to be equal, and effectively allows the sampling scheme to explore extensively the current modal region thereby increasing the chance of escaping from trapping states.

3.2.3 The label switching problem

Label switching is a difficulty that arises during the statistical analysis of mixture distributions. In the Bayesian framework, when we combine the use of symmetric priors for model parameters (mixing weights and component parameters) with a likelihood that is invariant to permutations of labels, we end up with a posterior distribution that is also invariant to relabelling. This means that, for a k-component mixture, the posterior essentially has $k!$ modes of equal importance. During the MCMC iterative sampling procedure, samples of parameters drawn from the stationary (equilibrium) distribution are therefore likely to have originated from one of those $k!$ modal regions of the posterior surface. Ideally, for parameter estimates to be meaningful, the samples that provide them have to have been drawn from the same modal region. While label switching is desirable in that it is an indication of good mixing and therefore good exploration of the posterior surface, a careless treatment of its effect would lead to meaningless parameter estimates.

Many strategies have been used to address the difficult issue of label switching. The most natural approach, tested by Diebolt and Robert (1994) ,Richardson and Green (1997), Fokoué (2000), Fokoué and Titterington (2000a) and many others, consists of imposing an ordering a priori to make sure that all the samples of the Markov chain come from

the same modal region of the posterior surface. In practice, one may decide to accept only samples satisfying the constraint $\pi_1 < \pi_2 < \cdots < \pi_k$ or in the univariate setting to impose an ordering on the means of the Gaussians, e.g. $\mu_1 < \mu_2 < \cdots < \mu_k$. Despite its intuitive nature, this approach leads to a poor representation of the geometry of the posterior surface. Besides, it cannot be easily extended to the multivariate setting and, worst of all, it leads to a high rejection rate (especially in multivariate settings) and considerably retards the sampler.

Some other solutions to this problem based on k-means-like clustering algorithms and the use of loss functions have been proposed by Celeux (1998), Celeux, Hurn, and Robert (2000) and Stephens (2000), and tested by Fokoué and Titterington (2000d) and Hurn, Justel, and Robert (2000). In our work, we use an online clustering algorithm Celeux (1998), Celeux, Hurn, and Robert (2000) that consists of isolating one of the $k!$ modes (the mode of reference). The reader is referred to the cited references for details of the methods.

3.2.4 Estimation efficiency and overfitting

When used for density estimation, finite mixtures of Gaussians can be prone to overfitting in high-dimensional spaces. In fact, as the number of mixture components increases, density estimation is greatly improved. However, this increase in the number of components leads to a significant increase in the number of free model parameters when full covariance matrices are used, and this naturally leads to overfitting in the event of small samples. An extreme solution to this problem consists of constraining the covariance matrices to be isotropic. This is obviously not a very realistic solution in most cases. In the next chapter, we will explain how mixtures of factor analysers (MFA) provide a better solution through structured covariance matrices.

3.2.5 Multimodality, local maxima and poor mixing

Because of the invariance to permutation mentioned earlier, the likelihood surface of a mixture model is inherently multimodal. This multimodality can become a serious computational problem with iterative algorithms that may easily get trapped in a local maximum while the true global maximum (if it exists) is actually located at a different modal region. In practice, as we noticed throughout our simulations, label switching does not happen very often when the generic Gibbs sampler is used, since the Gibbs sampler is not very good at jumping between different modal regions of the posterior surface of the parameters. In a sense, this might be good for parameter estimation for reasons given above, but can lead to very poor density estimation. The use in this context of sampling strategies like simulated tempering Celeux, Hurn, and Robert (2000) allows better exploration of all the modal regions of the posterior surface, which results in good mixing and therefore many occurrences of label switching.

3.3 Introduction to Mixtures of Factor Analysers

Unlike the fundamentally linear FA model, the MFA model is more flexible, with its inherent ability to partition a heterogeneous input space into clusters while simultaneously achieving local dimensionality reduction in each of the derived subspaces. Under the assumption of orthogonal factor analysis, the MFA is a reduced-dimensional mixture of multivariate Gaussians that can be used as an approximate method of density estimation in high-dimensional space, especially in cases where samples are of small sizes. In fact, while a plain mixture of multivariate Gaussians with full covariance matrices would be prone to overfitting when the number of mixture components is increased, the MFA model allows one to control or avoid overfitting by varying the dimensionalities of the latent subspaces (i.e. the number of common factors), thereby reducing the number of free model parameters significantly without imposing such strong constraints as forcing the covariance matrices of the local Gaussians to be isotropic.

The MFA model, by its construction and structure, is a rich and interesting extension of

both Factor Analysis and finite mixture models, and therefore has the potential for an even broader range of applications. In fact, in recent years, the study of MFA has received considerable interest. The psychometrics community with its traditional interest in FA and related multivariate models has produced a good number of papers among which Yung (1997), Dolan and Van der Maas (1998), Arminger, Stein, and Wittenberg (1999), all address the fitting of MFAs or closely related models, by various versions of Maximum Likelihood Estimation (MLE). From the Neural Computation community, Ghahramani and Hinton (1997) derived an EM algorithm for parameter estimation within the model. Ghahramani and Beal (2000) later considered a Bayesian treatment of MFA via a variational approximation. Ueda, Nakano, Ghahramani, and Hinton (2000) applied their Split-and-Merge-EM (SMEM) algorithm to the MFA model, and obtained good results in such tasks as image compression and handwritten digits recognition. From the mainstream statistics community, McLachlan and Peel (2000) proposed a variant of the EM algorithm for a study of the MFA model with application to clustering and density estimation. They applied the resulting algorithm to artificial and real data, and obtained good results. If we consider the Bayesian framework, then it emerges that, apart from Fokoué and Titterington (2000a) and Fokoué and Titterington (2000d), only approximate techniques have been used to address the intractability of the functions of interest. While these techniques can be fast in producing reasonably good results, assessing the closeness of approximations to the true values of interest still remains a complex problem.

To the best of our knowledge, the first attempt to use an "exact" technique (no approximation of the functions of interest) for the Bayesian analysis of the MFA model was presented by Fokoué (2000)[1], who constructed an efficient sampling scheme for the posterior simulation of the distributions of interests. The derived Markov Chain Monte Carlo (MCMC) algorithm was essentially a straightforward adaptation of *Data Augmentation* (a two-stage Gibbs sampler) to the complete-data formulation of the MFA inferential task. There have since then been some other developments along the lines

[1]For the writing and the presentation of this paper, I was awarded a Young Researchers' Prize for the best full contribution at the Compstat 2000 conference in Utrecht (The Netherlands).

of stochastic simulation for MFAs, namely in Fokoué and Titterington (2000a). We recently discovered that Utsugi and Kumagai (2001) independently worked on a similar Bayesian sampling scheme for MFAs with known and fixed k and q. Fokoué and Titterington (2000d) presents a more extended treatment where model selection is tackled in the Bayesian framework through the simulation of a continuous time birth-and-death point process; see Section (3.8) below.

3.3.1 What is a Mixture of Factor Analysers?

Definition: Let us once again consider our manifest random variable $\boldsymbol{x} \in \mathcal{X} \subseteq I\!R^p$. Suppose that the sample space $\mathcal{X}$ can be partitioned into k clusters, so that the probability of $\boldsymbol{x}$ to have originated from cluster j is $\pi_j = \Pr(\boldsymbol{y} = j)$ as defined earlier in Section (3.1.3). If we also suppose that in each cluster j, all the assumptions of orthogonal factor analysis hold, so that $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j) = \mathcal{N}_p(\boldsymbol{x}; \mu_j, \Lambda_j\Lambda_j^\intercal + \Sigma_j)$, for $j = 1, \cdots, k$, then it is easy to show that the unconditional probability density function of $\boldsymbol{x}$ can be written as

$$\boldsymbol{p}(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j \mathcal{N}_p(\boldsymbol{x}; \mu_j, \Lambda_j\Lambda_j^\intercal + \Sigma_j). \tag{3.8}$$

The density function $\boldsymbol{p}(\cdot)$ defined by (3.8) is a *mixture of factor analysers* (MFA). The intrinsic dimensionality of the data in cluster j is q_j, so that $\Lambda_j \in I\!R^{p \times q_j}$ for $j = 1, \cdots, k$.

Generative equation: Let $\boldsymbol{z}_j \in I\!R^{q_j}$ and $\boldsymbol{e}_j \in I\!R^p$ respectively denote our random vector of factor scores and our random vector of noise in cluster j. The assumptions here are $\boldsymbol{z}_j \sim \mathcal{N}_{q_j}(0, \mathbf{I}_{q_j})$, $\boldsymbol{e}_j \sim \mathcal{N}_p(0, \Sigma_j)$ and $\mathbb{E}[\boldsymbol{x}|\boldsymbol{y} = j, \boldsymbol{z}_j] = \Lambda_j\boldsymbol{z}_j + \mu_j$. Thus, conditional on $\boldsymbol{y} = j$, the generative equation for the MFA model can be expressed as

$$\boldsymbol{x} = \Lambda_j\boldsymbol{z}_j + \mu_j + \boldsymbol{e}_j, \quad j = 1, \cdots, k, \tag{3.9}$$

Missing data formulation of MFA: As a combination of two latent variable models, the MFA model is obviously a latent variable model itself. As before, the missing data formulation of the model will prove to be crucial for many inferential tasks. Using elements from both FA and finite mixtures, and assuming that $\boldsymbol{y}$ and $\boldsymbol{z}$ are *a priori* independent, it is easy to see that the complete-data density of all the variables of the

model is given by

$$p(x, y, z) \propto \prod_{j=1}^{k} \pi_j^{y_j} [\mathcal{N}_p(x; \mu_j + \Lambda_j z_j, \Sigma_j)]^{y_j}. \tag{3.10}$$

With $\pi = \{\pi_1, \cdots, \pi_k\}$, $\mu = \{\mu_1, \cdots, \mu_k\}$, $\Lambda = \{\Lambda_1, \cdots, \Lambda_k\}$ and $\Sigma = \{\Sigma_1, \cdots, \Sigma_k\}$, our complete collection of model parameters is now given by $\theta \equiv \{\pi, \mu, \Lambda, \Sigma\}$.

Remarks: As the form of (3.8) suggests, a mixture of factor analysers is nothing but a finite mixture of multivariate Gaussians with structured component covariance matrices.

Related work: Closely related to the MFA model are: the Mixture of Probabilistic Principal Component Analysers, studied by Tipping and Bishop (1999) who used the EM algorithm for parameter estimation; and to some extent (although purely univariate at this stage), the Mixture of Regressions studied from a stochastic simulation perspective by Hurn, Justel, and Robert (2000). A good understanding of the MFA model should form a good starting point for estimating Mixtures of Multivariate Regressions.

3.3.2 Why use a Mixture of Factor Analysers?

We have already presented many difficulties underlying the use of both factor analytic and finite mixture models, and it stands to reason that combining these two models naturally results in even more difficulties. Despite all the modelling challenges inherent in it, there are many reasons that make MFAs appealing, two of which can be simply explained as follows:

- *Locally linear but globally nonlinear Factor Analysis:* Combining a finite number of local factor analysers results in a globally nonlinear model that is theoretically more flexible and therefore better able to capture the complexity of the data.

- *Improved density estimation via parsimonious Gaussian mixtures:* When used for density estimation, finite mixtures of Gaussians can be prone to **overfitting** in high-dimensional spaces. In fact, as the number of mixture components increases, density estimation is greatly improved. However, this increase in the number of components leads to a significant increase in the number of free model parameters

when full covariance matrices are used, and this naturally leads to overfitting in the event of small samples. MFAs control or avoid overfitting by using the **intrinsic dimensionalities** of local factor analysers to control the number of model parameters. This is a good trade-off between the use of restrictive isotropic covariance matrices and the use of full covariance matrices.

Note: In the majority of applications where factor analytic models are used, the disturbance (noise) is generally measurement error. It is therefore reasonable and realistic to assume that such a noise has the same distribution across all the components of the MFA model. We shall therefore consider the general case where Σ_j are distinct across clusters, but, for practical applications, we shall assume that $\Sigma_j = \Sigma, \quad j = 1, \cdots, k$. On the other hand, we will treat a general situation where q_j are distinct across clusters, and the special case where $q_j = q, \quad j = 1, \cdots, k$.

3.4 Likelihood function for MFA

From the expression for the marginal density of $\boldsymbol{x}$ in equation (3.8), the observed-data likelihood for a sample of n i.i.d observations is given by

$$L(\boldsymbol{\theta}|\mathbf{X}) \propto \prod_{i=1}^{n} \left(\sum_{j=1}^{k} \pi_j \mathcal{N}_p(\boldsymbol{x}_i; \mu_j, \Lambda_j \Lambda_j^{\mathsf{T}} + \Sigma_j) \right) \tag{3.11}$$

The first thing to notice is that it would be hard if not impossible to derive closed-form expressions from (3.11), if one were inclined to use maximum likelihood estimation for instance. The only way forward would therefore be to resort to Newton-Raphson type algorithms, a solution we have so far avoided for reasons given earlier. On the other hand, (3.11) involves the evaluation of k^n terms corresponding to the different allocations of observations $\boldsymbol{x}_i$ to their corresponding components in the mixture model. If such an evaluation were to be repeated in an iterative algorithm, it would quickly become computationally unrealistic and almost infeasible even for samples of size greater than 40. For both likelihood based and Bayesian estimation methods, the use of the above observed-data likelihood is therefore not attractive, and we shall instead make use of the

complete-data likelihood whose expression is given by

$$L(\theta; \mathbf{X}, \mathbf{Y}, \mathbf{Z}) \propto \prod_{i=1}^{n} \left(\prod_{j=1}^{k} \pi_j^{y_{ij}} \left[\mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \boldsymbol{z}_{ij}, \Sigma_j) \right]^{y_{ij}} \right) \tag{3.12}$$

By "de-mixing" the likelihood function, the complete-data formulation makes it possible to derive closed-form expressions for the EM algorithm, and familiar full conditional posterior densities for Data Augmentation. Moreover, being a typical incomplete-data problem, the inferential task inherent in the MFA model naturally lends itself to two-stage iterative algorithms where the first stage imputes values to the missing (unobserved) data while the second stage performs the estimation based on the complete-data.

3.5 The EM algorithm for the MFA Model

The EM algorithm for mixtures of factor analysers is essentially a straightforward extension of the EM for Factor Analysis that we encountered earlier. On the other hand, since a MFA is a mixture of Gaussians for which the EM is now fairly standard, we avoid lengthy details here, and only provide the main results.

Remarks: (i) At this stage, it is probably fair to point out that, besides its great advantage of guaranteed convergence, the EM algorithm would be an even more appealing alternative to its other Maximum Likelihood estimation counterparts (the Newton-Raphson types) if closed-form expressions could be derived for both the E-step and the M-step. Fortunately, this turns out to be the case as we show later.

(ii) So far, we have, for simplicity, systematically omitted the indication of the parameter set $\boldsymbol{\theta}$ in our expressions (for example, $\boldsymbol{p}(\boldsymbol{x})$ in place of $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})$). While such a simplification is harmless in the likelihood-based framework where parameters are simply fixed quantities to be "plugged-in", it could lead to confusion in the Bayesian paradigm where parameters are stochastic entities, making an expression such as $\boldsymbol{p}(\boldsymbol{x})$ a marginal density of $\boldsymbol{x}$ over $\boldsymbol{\theta}$.

3.5.1 Likelihood-based inference for MFA

The EM algorithm for MFA that we shall be describing in the next section will yield a Maximum Likelihood point estimate $\hat{\boldsymbol{\theta}}_{\mathsf{EM}}$ of $\boldsymbol{\theta}$, and quantities of interest can be estimated by simply plugging in the value of $\hat{\boldsymbol{\theta}}_{\mathsf{EM}}$ in the appropriate expressions.

Density estimation: According to this, the density $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})$ would therefore be estimated by $\boldsymbol{p}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}_{\mathsf{EM}})$, with $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{\theta})$ defined as in equation (3.8).

Classification and Clustering: In the same way, the classification probabilities defined in equation (3.16) could be estimated by $\mathbf{Pr}(y_j = 1|\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{\mathsf{EM}})$.

Data reduction and factor scores estimation: Finally, expected factor scores $\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}, y_j, \boldsymbol{\theta}]$ could be estimated by $\mathbb{E}\left[\boldsymbol{z}|\boldsymbol{x}, y_j, \hat{\boldsymbol{\theta}}_{\mathsf{EM}}\right]$. In the case where $q_j = q$, the marginal expected factor score $\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}]$ is itself a mixture and has the form

$$\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}] = \mathbb{E}_{\boldsymbol{y}}[\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}]] = \sum_{j=1}^{k} \Pr(y_j = 1|\boldsymbol{x})\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}, y_j = 1, \boldsymbol{\theta}]. \tag{3.13}$$

$\mathbb{E}[\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}]$ of (3.13) can then be estimated by $\mathbb{E}\left[\boldsymbol{z}|\boldsymbol{x}, \hat{\boldsymbol{\theta}}_{\mathsf{EM}}\right]$.

3.5.2 Elements of the E-step

The expression of the expected log-likelihood for the MFA model is given by

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}\left[\log\left[\prod_{i=1}^{n}\left(\prod_{j=1}^{k}\pi_j^{y_{ij}}\left[\mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \boldsymbol{z}_{ij}, \Sigma_j)\right]^{y_{ij}}\right)\right]\right]. \tag{3.14}$$

An expansion of (3.14) reveals that a closed-form expression for $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ can be derived easily if closed-form expressions for $\mathbb{E}[y_j\boldsymbol{z}_j|\boldsymbol{x}]$ and $\mathbb{E}\left[y_j\boldsymbol{z}_j\boldsymbol{z}_j^{\mathsf{T}}|\boldsymbol{x}\right]$ exist (see Appendix B for details). It turns out that, if we use $\boldsymbol{y}$ as a vector of indicator variables, $\boldsymbol{y} = (y_i, \cdots, y_k)^{\mathsf{T}}$, then the event $\{\boldsymbol{y} = j\}$ is the same as $\{y_j = 1\}$. With $\boldsymbol{p}(y_j, \boldsymbol{z}_j|\boldsymbol{x}) = \boldsymbol{p}(y_j|\boldsymbol{x})\boldsymbol{p}(\boldsymbol{z}_j|y_j, \boldsymbol{x})$,

$$\mathbb{E}[y_j\boldsymbol{z}_j|\boldsymbol{x}] = \mathbb{E}[y_j|\boldsymbol{x}]\,\mathbb{E}[\boldsymbol{z}_j|\boldsymbol{x}, y_j] \quad \text{and} \quad \mathbb{E}\left[y_j\boldsymbol{z}_j\boldsymbol{z}_j^{\mathsf{T}}|\boldsymbol{x}\right] = \mathbb{E}[y_j|\boldsymbol{x}]\,\mathbb{E}\left[\boldsymbol{z}_j\boldsymbol{z}_j^{\mathsf{T}}|\boldsymbol{x}, y_j\right]. \tag{3.15}$$

$\mathbb{E}[\boldsymbol{z}_j|\boldsymbol{x}, y_j]$ and $\mathbb{E}\left[\boldsymbol{z}_j\boldsymbol{z}_j^{\mathsf{T}}|\boldsymbol{x}, y_j\right]$ are expressions that we derived when studying the EM for Factor Analysis. We therefore only need to find a closed-form expression for $\mathbb{E}[y_j|\boldsymbol{x}]$, which turns out to be quite straightforward as we now show.

If we average the complete-data density $\boldsymbol{p}(\boldsymbol{x}, y_j, \boldsymbol{z}_j)$ over $\boldsymbol{z}_j$, we can write $\boldsymbol{p}(\boldsymbol{x}, y_j) = \boldsymbol{p}(y_j|\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{p}(y_j)\boldsymbol{p}(\boldsymbol{x}|y_j)$. With $\mathbf{Pr}(y_j = 1) = \pi_j$ and $\boldsymbol{p}(y_j|\boldsymbol{x}) \propto \boldsymbol{p}(y_j)\boldsymbol{p}(\boldsymbol{x}|y_j)$, we derive $\mathbf{Pr}(y_j = 1|\boldsymbol{x}) \propto \pi_j \mathcal{N}(\boldsymbol{x}; \mu_j, \Lambda_j\Lambda_j^\mathsf{T} + \Sigma_j)$. Introducing the normalising constant, we can therefore express $\mathbf{Pr}(y_j = 1|\boldsymbol{x})$ as follows:

$$\mathbf{Pr}(y_j = 1|\boldsymbol{x}) = \frac{\pi_j \mathcal{N}_p(\boldsymbol{x}; \mu_j, \Lambda_j\Lambda_j^\mathsf{T} + \Sigma_j)}{\sum_{j'=1}^{k} \pi_{j'} \mathcal{N}_p(\boldsymbol{x}; \mu_{j'}, \Lambda_{j'}\Lambda_{j'}^\mathsf{T} + \Sigma_{j'})}. \tag{3.16}$$

Definition: We define $\mathsf{a}_{ij} = \mathbb{E}[y_{ij}|\boldsymbol{x}_i] = \mathbf{Pr}(y_{ij} = 1|\boldsymbol{x}_i)$, for each $\boldsymbol{x}_i$ and the corresponding y_{ij}, $(i = 1, \cdots, n$ and $j = 1, \cdots, k)$, so that $\mathsf{a}_{ij}^{(t)}$ is the current value of a_{ij} at the t-th iteration of the EM algorithm. In the same way, $\mathbf{b}_{ij}^{(t)}$ and $\mathbf{C}_{ij}^{(t)}$ are respectively the values of $\mathbf{b}_{ij} = \mathbb{E}[\boldsymbol{z}_{ij}|\boldsymbol{x}_i, y_{ij}]$ and $\mathbf{C}_{ij} = \mathbb{E}\left[\boldsymbol{z}_{ij}\boldsymbol{z}_{ij}^\mathsf{T}|\boldsymbol{x}_i, y_{ij}\right]$ at the t-th iteration of the algorithm.

3.5.3 Elements of the M-step updates

It turns out that the analytical expression of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ obtained earlier also allows the derivation of closed-form expressions at the M-step. The details of the results of this section can be found in Appendix B. We define

$$n_j = \sum_{i=1}^{n} y_{ij} = \#\{\boldsymbol{x}_i : \boldsymbol{y}_i = j, \quad i = 1, \cdots, n\} \tag{3.17}$$

as the number of observations currently allocated to component j. A pseudocode for one iteration of the corresponding EM algorithm is provided by Algorithm 5.

Remark: As we discussed in Section 3.2, the application of the EM to mixtures generally encounters such problems as: (a) singularities due to the unboundedness of the likelihood; (b) existence of many local maxima due to the usually multimodal nature of the likelihood surface; and (c) sensitivity to initial parameter values. In fact, our simulations reveal (see Section 3.7) that the above EM algorithm, once trapped in a local maximum of the likelihood surface, cannot escape it, thereby yielding solutions that in some cases are very far from the truth and therefore not of great use. Such

limitations stimulate one to resort to an alternative approach. McLachlan and Peel (2000) propose an extension of the EM, while we opt for a Bayesian solution via Data Augmentation.

Algorithm 5: The EM Algorithm for Mixtures of Factor Analysers

- **E-step** - With current $\boldsymbol{\theta}^{(t)}$, compute $\mathsf{a}_{ij}^{(t)}$, $\mathbf{b}_{ij}^{(t)}$ and $\mathbf{C}_{ij}^{(t)}$.

- **M-step** -

$$\pi_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\mathsf{a}_{ij}^{(t)}$$

$$\mu_j^{(t+1)} = \left[\sum_{i=1}^{n}\mathsf{a}_{ij}^{(t)}\left(\boldsymbol{x}_i - \Lambda_j^{(t)}\mathbf{b}_{ij}^{(t)}\right)\right]\left[\sum_{i'=1}^{n}\mathsf{a}_{i'j}^{(t)}\right]^{-1}$$

$$\Lambda_j^{(t+1)} = \left[\sum_{i=1}^{n}\mathsf{a}_{ij}^{(t)}(\boldsymbol{x}_i - \mu_j^{(t+1)})\left(\mathbf{b}_{ij}^{(t)}\right)^{\mathsf{T}}\right]\left[\sum_{i'=1}^{n}\mathsf{a}_{i'j}^{(t)}\mathbf{C}_{i'j}^{(t)}\right]^{-1}$$

if $\Sigma_j = \Sigma, \quad \forall j \in \{1, \cdots, k\}$ then

$$\Sigma^{(t+1)} = \frac{1}{n}\mathsf{diag}\left[\sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}\left(\boldsymbol{x}_i - \mu_j^{(t+1)} - \Lambda_j^{(t+1)}\mathbf{b}_{ij}^{(t)}\right)\left(\boldsymbol{x}_i - \mu_j^{(t+1)}\right)^{\mathsf{T}}\right]$$

if $\Sigma_j = \Sigma_j' \quad \text{for} \quad j \neq j'$ then

$$\Sigma_j^{(t+1)} = \frac{1}{n_j}\mathsf{diag}\left[\sum_{i=1}^{n}\mathsf{a}_{ij}^{(t)}\left(\boldsymbol{x}_i - \mu_j^{(t+1)} - \Lambda_j^{(t+1)}\mathbf{b}_{ij}^{(t)}\right)\left(\boldsymbol{x}_i - \mu_j^{(t+1)}\right)^{\mathsf{T}}\right]$$

Note: The details of the derivation of the above algorithm are given in Appendix (B).

3.6 Bayesian inference for MFA

As we discussed earlier in Section 2.5, our main ingredient for inference in the Bayesian framework is the posterior density $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$. All quantities of interest in this case are obtained by averaging over the parameter space, with averages weighted by $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$.

Predictive density: For a future (unseen) observation $\boldsymbol{x}_{\text{new}}$, the predictive density $\boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\mathbf{X})$ of $\boldsymbol{x}_{\text{new}}$ given $\mathbf{X}$ is provided by

$$\boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\mathbf{X}) = \int_{\Theta} \boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}. \tag{3.18}$$

Expected factor scores: For observations contained in the sample used for parameter estimation, the expected factor score $\mathbb{E}[\boldsymbol{z}|\mathbf{X}]$ is given by

$$\mathbb{E}[\boldsymbol{z}|\mathbf{X}] = \mathbb{E}_{\boldsymbol{\theta}}[\mathbb{E}[\boldsymbol{z}|\mathbf{X},\boldsymbol{\theta}]] = \int_{\Theta} \mathbb{E}[\boldsymbol{z}|\mathbf{X},\boldsymbol{\theta}]\, \boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}. \tag{3.19}$$

For a future (unseen) observation $\boldsymbol{x}_{\text{new}}$, the expected factor score is given by

$$\begin{aligned}\mathbb{E}[\boldsymbol{z}_{\text{new}}|\boldsymbol{x}_{\text{new}},\mathbf{X}] &= \mathbb{E}_{\boldsymbol{\theta}}[\mathbb{E}[\boldsymbol{z}_{\text{new}}|\boldsymbol{x}_{\text{new}},\mathbf{X},\boldsymbol{\theta}]] = \int_{\Theta} \mathbb{E}[\boldsymbol{z}_{\text{new}}|\boldsymbol{x}_{\text{new}},\boldsymbol{\theta}]\, \boldsymbol{p}(\boldsymbol{\theta}|\boldsymbol{x}_{\text{new}},\mathbf{X})d\boldsymbol{\theta} \\ &= \frac{1}{\boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\mathbf{X})} \int_{\Theta} \mathbb{E}[\boldsymbol{z}_{\text{new}}|\boldsymbol{x}_{\text{new}},\boldsymbol{\theta}]\, \boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}. \end{aligned} \tag{3.20}$$

Clustering and classification: The classification probabilities for all the observations in the sample are provided by

$$\Pr(\boldsymbol{y}_i = j|\mathbf{X}) = \int_{\Theta} \Pr(\boldsymbol{y}_i = j|\boldsymbol{x}_i,\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}, \tag{3.21}$$

where $\Pr(\boldsymbol{y}_i = j|\boldsymbol{x}_i,\boldsymbol{\theta})$ is defined by (3.16). Finally, for a future observation $\boldsymbol{x}_{\text{new}}$, the classification probability will be

$$\begin{aligned}\Pr(\boldsymbol{y}_{\text{new}} = j|\boldsymbol{x}_{\text{new}},\mathbf{X}) &= \int_{\Theta} \Pr(\boldsymbol{y}_{\text{new}} = j|\boldsymbol{x}_{\text{new}},\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\boldsymbol{x}_{\text{new}},\mathbf{X})d\boldsymbol{\theta} \\ &= \frac{1}{\boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\mathbf{X})} \int_{\Theta} \Pr(\boldsymbol{y}_{\text{new}} = j|\boldsymbol{x}_{\text{new}},\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{x}_{\text{new}}|\boldsymbol{\theta})\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} \end{aligned} \tag{3.22}$$

Note: All the above averages cannot be obtained in closed-form, because of the intractability of the integrals involved. To remedy this, we draw samples from $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$ via Data Augmentation, and then we compute approximations of the above integrals.

3.6.1 Elements of Data Augmentation for MFA

In Chapter 2, we used Data Augmentation for the analysis of the single factor model, along the lines of Lopes and West (1999), Martin and McDonald (1981) and Ihara and Kano (1995), and we showed that the derived scheme was efficient. Diebolt and Robert

(1994), Richardson and Green (1997), Escobar and West (1995), Celeux (1998) and Hurn, Justel, and Robert (2000) among others, have applied the method to the analysis of finite mixtures. Combining ideas from both these camps, it turns out that Data Augmentation (two-stage Gibbs sampler) provides a natural framework for the derivation of an efficient sampling scheme for the MFA model. In fact, if we assume a fully specified prior density $p(\theta)$ for θ, we can then use the complete-data likelihood of equation (3.12) to form the complete-data posterior density for the MFA model, which is given by

$$p(\theta, \mathbf{Y}, \mathbf{Z}|\mathbf{X}) \propto \left[\prod_{i=1}^{n}\left(\prod_{j=1}^{k}\pi_j^{y_{ij}}\left[\mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \boldsymbol{z}_{ij}, \Sigma_j)\right]^{y_{ij}}\right)\right] p(\theta). \tag{3.23}$$

The good news here is that our complete-data likelihood function makes it possible to use a conjugate prior structure, which in turn allows us to derive full conditional posteriors that are standard and easy to simulate. The resulting sampling scheme for $p(\theta, \mathbf{Y}, \mathbf{Z}|\mathbf{X})$ is therefore an efficient one. Note that this sampling scheme is essentially the same as the one derived in Section 2.5, with the main difference that we now need additional sampling for $\boldsymbol{y}$ and the corresponding π.

With $\theta^{(t)}, \mathbf{Y}^{(t)}, \mathbf{Z}^{(t)}$ as the current values of the chain the algorithm has the form

---

Algorithm 6: The Data Augmentation Algorithm for MFA.

Imputation step: Impute some values for the missing latent variables.

Simulate $\mathbf{Y}^{(t+1)} \sim p(\mathbf{Y}|\theta^{(t)}, \mathbf{X}, \mathbf{Z}^{(t)})$

Simulate $\mathbf{Z}^{(t+1)} \sim p(\mathbf{Z}|\theta^{(t)}, \mathbf{X}, \mathbf{Y}^{(t+1)})$

Posterior step: Draw new parameter values given the augmented data.

Simulate $\theta^{(t+1)} \sim p(\theta|\mathbf{X}, \mathbf{Y}^{(t+1)}, \mathbf{Z}^{(t+1)})$

In the spirit of the Gibbs sampler, the equilibrium distribution reached by Algorithm 6 should provide samples from posterior marginals $p(\theta|\mathbf{X})$, $p(\boldsymbol{z}|\mathbf{X})$ and $p(\boldsymbol{y}|\mathbf{X})$ that can then be used to obtain estimates of parameters, estimates of expected factor scores and estimates of classification probabilities respectively.

3.6.2 Bayesian inference via Data Augmentation

As we said earlier, once the Markov chain constructed by Data Augmentation has converged to the equilibrium distribution, it provides ingredients for a variety of inferential tasks. The key advantage here is that all these ingredients are obtained simultaneously, and many aspects of inference in this case are just by-products of the same process, with little extra computation needed. Before giving a detailed description of the sampling scheme, we first address some important issues related to inference. First and foremost, if we assume that problems like label switching and convergence have been dealt with, Bayesian parameters estimates are straightforward. With M useful MCMC samples retained after "burn-in", the chain $\{\boldsymbol{\theta}^{(t)} : t = 1, \cdots, M\}$ provides a sample of draws from $\boldsymbol{p}(\boldsymbol{\theta}|\mathbf{X})$, and, just as before, the estimate $\hat{\boldsymbol{\theta}}_{\mathsf{DA}}$ of $\boldsymbol{\theta}$ is given by

$$\hat{\boldsymbol{\theta}}_{\mathsf{DA}} = \frac{1}{M}\sum_{t=1}^{M}\boldsymbol{\theta}^{(t)}. \tag{3.24}$$

Expected Factor scores: With the sample path $\{\mathbf{Z}^{(t)} : t = 1, \cdots, M\}$ produced by Algorithm 6, it is natural and straightforward to compute the corresponding ergodic averages, and to use them as Bayesian estimates for $\mathbb{E}[\boldsymbol{z}|\mathbf{X}]$. However, for problems of even moderate intrinsic dimensionalities, having to store these vectors of common factors can quickly become explosive in terms of the storage capacity required. This is indeed a major drawback. Two straightforward solutions that avoid such explosive storages can be adopted in this case: (a) *On-line estimation of expected factor scores,* which consists of updating averages such $\frac{1}{M}\sum_{t=1}^{M}\boldsymbol{z}^{(t)}$ at each iteration. (b) *auxiliary latent variables.* Instead of storing the $\boldsymbol{z}^{(t)}$ so as to compute averages later, one can simply use the current draw to "augment" the data for the sake of parameter estimation. Once convergence is achieved, simply use the chain of parameters to compute the corresponding estimates of expected factor scores. For instance, this would mean combining the expression for $\mathbb{E}[\boldsymbol{z}|\mathbf{X}, \boldsymbol{\theta}]$ of equation (3.13) with (3.19) to produce an approximation of $\mathbb{E}[\boldsymbol{z}|\mathbf{X}]$ given by

$$\mathbb{E}[\boldsymbol{z}|\mathbf{X}] \approx \frac{1}{M}\sum_{t=1}^{M}\mathbb{E}\left[\boldsymbol{z}|\mathbf{X}, \boldsymbol{\theta}^{(t)}\right]. \tag{3.25}$$

In this way, the $z^{(t)}$'s appear in the sampling scheme just as auxiliary variables to facilitate the estimation of parameters.

Classification probabilities and clustering: Each set $\mathbf{Y}^{(t)} = \{y_i^{(t)} : i = 1, \cdots, n\}$ in the chain $\{\mathbf{Y}^{(t)} : t = 1, \cdots, M\}$ provides a possible clustering of the data, corresponding to the current set $\theta^{(t)}$ of model parameters. For the same reasons as before, it is inefficient to store the chain $\{\mathbf{Y}^{(t)} : t = 1, \cdots, M\}$. We simply use each draw $y^{(t)}$ just as an instrumental variable for the completion of the data in view of parameter estimation. Using the chain $\{\theta^{(t)} : t = 1 \cdots, M\}$, compute estimates of our classification probabilities

$$\Pr(y_i = j|\mathbf{X}) \approx \frac{1}{M}\sum_{t=1}^{M} \Pr(y_i = j|x_i, \theta^{(t)}). \tag{3.26}$$

Estimates provided by (3.26) can then be used for *soft* or probabilistic clustering of the data. More specifically, *soft* clustering is achieved by drawing the label y_i of x_i from a multinomial distribution with parameters k and $\Pr(y_i = j|\mathbf{X}), j = 1, \cdots, k$. A *hard* or outright clustering can also be obtained by assigning each observation to the component for which the posterior probability $\Pr(y_i = j|\mathbf{X})$ is the highest.

Density estimation: Last but not least, it is worth pointing out that density estimates for future observations in this case are straightforward, and are given by

$$p(x_{\text{new}}|\mathbf{X}) \approx \frac{1}{M}\sum_{t=1}^{M} p(x_{\text{new}}|\theta^{(t)}), \tag{3.27}$$

which can also be used for all the inferential tasks involving future observations.

3.6.3 Hierarchical structure specification

It is worth remarking that the MFA model lends itself to a hierarchical structure specification. The natural approach to prior specification in this context would be to use the standard hierarchical prior structure as given by

$$p(\theta) = p(\pi|\delta)p(\mu|\xi, \kappa)p(\Sigma|\alpha, \tau)p(\Lambda|\eta, \Omega), \tag{3.28}$$

where $\delta, \xi, \kappa, \alpha, \tau, \eta, \Omega$ are hyperparameters. With the prior defined by (3.28), the hierarchical structure of the MFA model is given by Figure (3.1). However, as reported

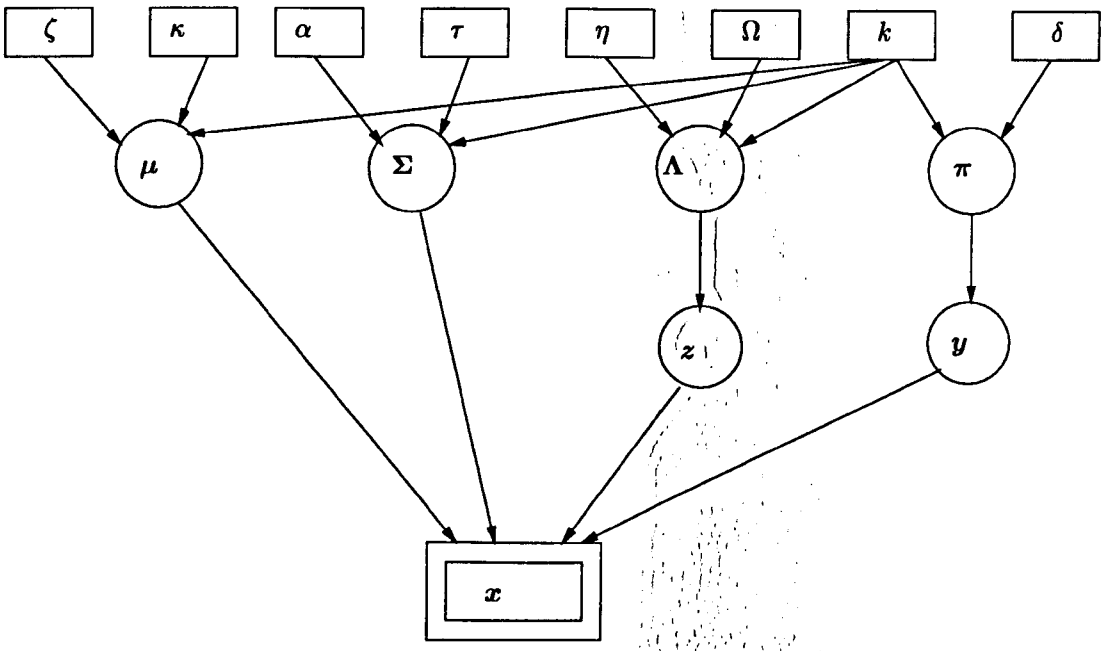


Figure 3.1: Direct Acyclic Graph (DAG) showing the hierarchical structure of the MFA model. A circle indicates an unknown random quantity, while a square (or rectangle) represents a constant. The double box is dedicated to the observed data.

by Robert and Casella (2000) and also noticed in our simulations, the use of this standard hierarchical prior structure leads to singularities and trapping states in sampling as explained in Section (3.2.2). To simplify our description, we first restrict ourselves to the case where $\Sigma_j = \Sigma$, and $q_j = q$, $\forall j \in \{1, \cdots, k\}$. Adaptation to the other cases is straightforward. In fact, if one of the component covariance matrices $\Lambda_j \Lambda_j^T + \Sigma$ is allowed to become extremely small (i.e have terms of very small magnitude) at any given sample point, then that component of the MFA will be allocated that single point, with no chance of having any other point allocated to it, since the fixed hyperparameter will obviously never change the state of the chain. Instead of using the standard hierarchical prior structure[2] of equation (3.28), we use the extended structure of equation (3.29), where an extra layer allows the component covariance matrices to be explored at least locally through the stochasticity of the hyperparameters of Λ:

$$p(\theta) = p(\pi|\delta)p(\mu|\xi,\kappa)p(\Sigma|\alpha,\tau)p(\Lambda|\eta,\Omega)p(\Omega|g,h), \tag{3.29}$$

[2]It is fair to point out that, while this is feasible via the use of an extended prior structure, such a flexible and principled solution is not available in the deterministic setting of the EM algorithm.

where g and h are the hyperparameters of the hyperprior Ω. The extended prior defined in (3.29) modifies the hierarchical structure of the MFA model, and the new DAG is given by Figure (3.2). As far as prior distributions are concerned, we use conjugate

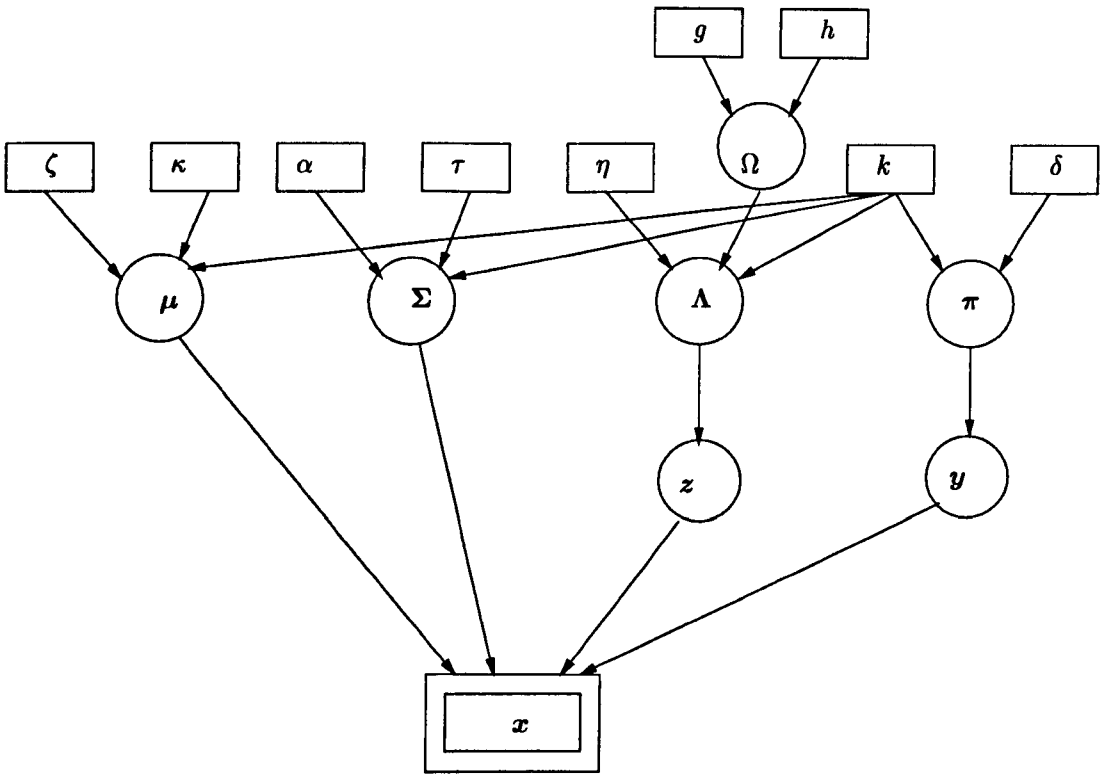


Figure 3.2: DAG of the extended hierarchical structure for the MFA model.

priors as we did before for the FA model. The main difference from the FA model in this regard is the discrete categorical latent variable $\boldsymbol{y}$ which has a multinomial unconditional distribution, allowing us to use a *Dirichlet* conjugate prior for the mixing weights $\boldsymbol{\pi}$.

3.6.4 Construction of the sampling scheme

The sampling scheme for the MFA model is essentially just an extension of the scheme that we constructed for the FA model in Chapter 2.

Imputation step for MFA: This step consists of simulating samples from the conditional posterior distributions of the latent variables. It can be easily shown that $\boldsymbol{y}$ has a multinomial conditional posterior distribution, denoted here by Mn. With $\boldsymbol{z}$ having a Gaussian distribution, imputation here has the following form:

$$[y_i|\cdots] \sim \mathsf{Mn}(1, \pi^*_{1i}, \cdots, \pi^*_{ki}) \text{ with } \pi^*_{ji} \propto \pi_j \mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \boldsymbol{z}_i, \Sigma)$$
$$[\boldsymbol{z}_{i:y_i=j}|\cdots] \sim \mathcal{N}_q((\mathbf{I}_q + \Lambda_j^\mathsf{T}\Sigma^{-1}\Lambda_j)^{-1}\Lambda_j^\mathsf{T}\Sigma^{-1}(\boldsymbol{x}_i - \mu_j), (\mathbf{I}_q + \Lambda_j^\mathsf{T}\Sigma^{-1}\Lambda_j)^{-1}).$$

For $i = 1, \cdots, n$ and $j = 1, \cdots, k$

Full Conditional posteriors: The derivation of full conditional posteriors for the MFA model follows naturally from the construction of conditional posteriors for the FA model as encountered in Chapter 2, with the only exception that we now have to include the mixing proportions. Since we gave ample details of the construction in Chapter 2, we will just present the key elements of our extended full conditional posteriors.

Mixing proportions: From the expression of the joint posterior, we have for $\boldsymbol{\pi}$

$$\boldsymbol{p}(\boldsymbol{\pi}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Lambda}, \mathbf{X}^*) \propto \left[\prod_{j=1}^{k} \pi_j^{\sum_{i=1}^{n} y_{ij}}\right] \boldsymbol{p}(\boldsymbol{\pi}) = \left[\prod_{j=1}^{k} \pi_j^{n_j}\right] \boldsymbol{p}(\boldsymbol{\pi}). \tag{3.30}$$

The expression of the likelihood function allows us to use a symmetric *Dirichlet* prior distribution as the natural conjugate prior distribution for our mixing proportions. More specifically, we have $\boldsymbol{\pi} \sim \mathsf{Di}(\delta, \cdots, \delta)$, and we write $\boldsymbol{p}(\pi) \propto \pi_1^\delta \cdots \pi_k^\delta = \prod_{j=1}^{k} \pi_j^\delta$. Combining the *Dirichlet* prior with the complete-data likelihood yields a *Dirichlet* full conditional posterior:

$$[\boldsymbol{\pi}|\cdots] \sim \mathsf{Di}(\delta + n_1, \cdots, \delta + n_k),$$

Component means μ_j: Details of this derivation are the same as described in Chapter 2. For each μ_j, we use the Gaussian prior $\mu_j \sim \mathcal{N}(\xi, \kappa)$ and the full conditional posterior for μ_j is therefore Gaussian, that is $[\mu_j|\cdots] \sim \mathcal{N}_p(m_{\mu_j}, C_{\mu_j})$, with

$$C_{\mu_j}^{-1} = \kappa^{-1} + n_j\Sigma^{-1} \quad \text{and} \quad m_{\mu_j} = C_{\mu_j}(\kappa^{-1}\xi + \Sigma^{-1}\xi_{\boldsymbol{x}_j}). \tag{3.31}$$

In the above equation (3.31), $\xi_{\boldsymbol{x}_j} = \sum_{i:y_i=j}^{n} (\boldsymbol{x}_i - \Lambda_j \boldsymbol{z}_i)$, for $j = 1, \cdots, k$.

Specific covariance Σ: Essentially, the only new aspect here is that we now need to redefine the matrix S as $S = \sum_{j=1}^{k} \sum_{i:y_i=j} (\boldsymbol{x}_i - \Lambda_j \boldsymbol{z}_i - \mu_j)(\boldsymbol{x}_i - \Lambda_j \boldsymbol{z}_i - \mu_j)^\mathsf{T}$. Since $\Sigma^{-1} = \mathsf{diag}(\sigma_1^{-2}, \cdots, \sigma_p^{-2})$, we use independent Gamma conjugate priors for each σ_r^{-2}, namely $\sigma_r^{-2} \sim \mathsf{Ga}(\alpha, \tau)$, for $r = 1, \cdots, p$. From all that, we easily derive a Gamma full conditional conjugate posterior of the following form:

$$[\sigma_r^{-2}|\cdots] \sim \mathsf{Ga}(\alpha + n/2, \tau + S_{rr}/2).$$

Factor loading matrices Λ_j: We define the column vector $\Lambda_{jr.} \in I\!R^q$, made up of the r-th row of the j-th matrix of factor loadings. We also define $\mathbf{Z}_j \in I\!R^{n_j \times q}$ to be the $n_j \times q$ matrix containing all the factor scores currently allocated to component j. We use the zero mean Gaussian prior $\Lambda_{jr.} \sim \mathcal{N}(0, \Omega)$ for $j = 1, \cdots, k$ and $r = 1, \cdots, p$. If we use the same details of derivation as in Chapter 2, this gives a Gaussian full conditional posterior $[\Lambda_{jr.}|\cdots] \sim \mathcal{N}_q(m_{\Lambda_{jr.}}, C_{\Lambda_{jr.}})$, with

$$C_{\Lambda_{jr.}}^{-1} = \Omega^{-1} + \sigma_r^{-2}(\mathbf{Z}_j^{\mathsf{T}}\mathbf{Z}_j) \quad \text{and} \quad m_{\Lambda_{jr.}} = C_{\Lambda_{jr.}}(\sigma_r^{-2}\mathbf{Z}_j^{\mathsf{T}}\ddot{\mathbf{X}}_{.jr}) \tag{3.32}$$

where $\ddot{\mathbf{X}}$ is the data matrix obtained from $\ddot{\mathbf{x}}_j = \boldsymbol{x} - \mu_j$, and $\ddot{\mathbf{X}}_{.jr}$ is a column vector containing the elements of its r-th row in component j.

Note: A full conditional posterior for an MFA model with constrained underlying factor analysers is easily obtained *mutatis mutandis* as for the factor model of Chapter 2.

Posterior for Ω: We assume Ω to be diagonal. More precisely $\Omega^{-1} = \mathsf{diag}(\omega_1^{-2}, \cdots, \omega_q^{-2})$. We also define $B = \sum_{j=1}^{k}\sum_{r=1}^{p} \Lambda_{jr.}\Lambda_{jr.}^{\mathsf{T}}$. Since each $\Lambda_{jr.}$ has a Gaussian distribution, we use an independent Gamma conjugate prior for each ω_c^{-2}, for $c = 1, \cdots, q$. Finally, with $\omega_c^{-2} \sim \mathsf{Ga}(g, h)$, we easily derive a Gamma full conditional conjugate posterior distribution for ω_c^{-2}:

$$[\omega_c^{-2}|\cdots] \sim \mathsf{Ga}(g + kp/2, h + B_{cc}/2).$$

If we combine all the above elements, one step of Data Augmentation for MFA would be given by Algorithm 7. In this algorithm, all the full conditional posterior distributions of interest are standard and therefore easy to simulate, making the algorithm an efficient sampling scheme. However, it must noted that for moderately large values of q (e.g. $q > 7$) and k (e.g. $k > 10$), the algorithm can become extremely slow to converge, for the simple reason that the amount of "missing-data" to be "filled-in" then grows accordingly. Another consequence of having to "fill-in" these latent variables in sampling is the poor mixing due to the fact the sampler tends to remain for too long in a tiny local region of the parameter support. Faced with a similar problem while analysing mixture

models similar to MFAs, Celeux, Hurn, and Robert (2000) and Hurn, Justel, and Robert (2000) have used versions of Langevin Metropolis and random walk in univariate settings. During a personal communication, the above authors revealed to us that they did not achieve any significant gain in performance. Besides, it is not easy to extend their attempts to multivariate settings such as ours. We have tried in our study to construct hybrid schemes along the lines of Nobile (1998) in order to improve both mixing and convergence, but it is fair to say that we did not achieve any progress in this regard.

Algorithm 7: Data Augmentation for Mixtures of Factor Analysers

- **I-step** - For $i = 1, \cdots, n$ and $j = 1, \cdots, k$

$$
\begin{aligned}
[y_i|\cdots] &\sim \mathsf{Mn}(1, \pi^*_{1i}, \cdots, \pi^*_{ki}) \quad \text{with} \quad \pi^*_{ji} \propto \pi_j \mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \boldsymbol{z}_i, \Sigma) \\
[\boldsymbol{z}_{i:y_i=j}|\cdots] &\sim \mathcal{N}_q((\mathbf{I}_q + \Lambda_j^\mathsf{T}\Sigma^{-1}\Lambda_j)^{-1}\Lambda_j^\mathsf{T}\Sigma^{-1}(\boldsymbol{x}_i - \mu_j), (\mathbf{I}_q + \Lambda_j^\mathsf{T}\Sigma^{-1}\Lambda_j)^{-1}
\end{aligned}
$$

- **P-step** -

$$
\begin{aligned}
[\boldsymbol{\pi}|\cdots] &\sim \mathsf{Di}(\delta + n_1, \cdots, \delta + n_k) \\
[\mu_j|\cdots] &\sim \mathcal{N}_p(m_{\mu_j}, C_{\mu_j}), \quad j = 1, \cdots, k \\
[\sigma_r^{-2}|\cdots] &\sim \mathsf{Ga}(\alpha + n/2, \tau + S_{rr}/2), \quad r = 1, \cdots, p \\
[\omega_c^{-2}|\cdots] &\sim \mathsf{Ga}(g + kp/2, h + B_{cc}/2), \quad c = 1, \cdots, q \\
[\Lambda_{jr\cdot}|\cdots] &\sim \mathcal{N}_q(m_{\Lambda_{jr\cdot}}, C_{\Lambda_{jr\cdot}}), \quad j = 1, \cdots, k \quad r = 1, \cdots, p
\end{aligned}
$$

3.6.5 On-line clustering for label switching

Although the poor mixing of the two-stage Gibbs sampler drastically reduces the occurrence of label switching in our essentially multivariate setting, we still have encountered it in some of our problems. Since the simple and intuitive ordering constraints often do not agree with the geometry of the parameter surface, they often fail to isolate one of the $k!$ modes of the posterior. We have opted for Celeux (1998)'s online clustering, and we briefly describe it in this section, using the author's notation. The main advantage of

this algorithm is that it does not require the MCMC sample path to be stored, as all the inferential quantities of interest are computed online. Here θ is a d-dimensional vector containing the complete collection of model parameters, so that if we have a total of b individual scalar parameters in a k-component mixture, then θ would be kb-dimensional. The procedure is initialised with m MCMC samples, $\theta^1, \theta^2, \cdots, \theta^m$, where m is chosen such that label switching has not yet occurred, which in practice may require an inspection of the samples. [3] The method then defines reference centres $\bar{\theta}_i$ together with their corresponding componentwise variances s_i for all the parameters as follows:

$$\bar{\theta}_i = \frac{1}{m}\sum_{j=1}^{m}\theta_i^j \quad \text{and} \quad s_i = \frac{1}{m}\sum_{j=1}^{m}(\theta_i^j - \bar{\theta}_i)^2.$$

It then sets $s_i^{[0]} = s_i, i = 1, \cdots, d$. With $\bar{\theta}_1^{[0]} = \bar{\theta}$, the $(k! - 1)$ other centres $\bar{\theta}_2^{[0]}, \bar{\theta}_3^{[0]}, \cdots$ are deduced by permutation, and one run of the r-th iteration is given by:

Algorithm 8: On-line clustering for label switching

1. Allocate θ^{m+r} to the cluster j^* (where $j = 1, \cdots, k!$ that minimises the normalised squared distance

$$\|\theta^{m+r} - \bar{\theta}_j^{[r-1]}\|^2 = \sum_{i=1}^{d}\frac{(\theta_i^{m+r} - \bar{\theta}_{ij}^{[r-1]})^2}{s_i^{(r-1)}},$$

where $\bar{\theta}_{ij}^{[r-1]}$ is the i-th coordinate of $\bar{\theta}_j^{[r-1]}$. If $j^* \neq 1$, then permute the coordinates of θ^{m+r} to get $j^* = 1$.

2. Update the $k!$ centres and the d normalising coefficients

   (a) Compute

$$\bar{\theta}_1^{[r]} = \frac{m+r-1}{m+r}\bar{\theta}_1^{[r-1]} + \frac{1}{m+1}\theta^{m+r}$$

   (b) Derive the $(k! - 1)$ other centres by permutation

   (c) Update the variances for $i = 1, \cdots, d$

$$\begin{aligned} s_i^{[r]} &= \frac{m+r-1}{m+r}s_i^{[r-1]} + \frac{m+r-1}{m+r}(\bar{\theta}_{i1}^{[r-1]} - \bar{\theta}_{i1}^{[r]})^2 \\ &+ \frac{1}{m+r}(\theta_i^{m+r} - \bar{\theta}_{i1}^{[r]})^2. \end{aligned}$$

[3] Typically, m must be large enough (this choice is not very sensitive, generally $m = 100$ or so works well) to ensure that the initial estimates are a reasonable crude approximation of the posterior means.

Remark: Because of its $k!$ computational complexity, the above algorithm would not be interesting for mixtures with more than $k = 6$ components, for that would mean a huge extra computational burden at each iteration to isolate the good mode.

3.6.6 A decision-theoretic solution to label switching

An alternative to the above algorithm is the decision-theoretic solution proposed by Celeux, Hurn, and Robert (2000) and used by Hurn, Justel, and Robert (2000). The method mainly consists of specifying "suitable" loss functions $\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$ for the inferential tasks at hand[4]. These loss functions are chosen so that they do not rely on the labelling of the components, and are therefore not affected by the lack of identifiability due to invariance to relabelling. This section is purely informative, and we therefore do not give ample details here. The reader is referred to either Celeux, Hurn, and Robert (2000) or Hurn, Justel, and Robert (2000) for a more complete description of the method. Once a "suitable" loss function is chosen according to the inferential issue of interest, the method essentially consists of the following:

Algorithm 9: Decision-theoretic approach to label switching

1. Compute the expected loss $\mathbb{E}_{\boldsymbol{\theta}|\mathbf{x}}\left[\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})\right]$.
2. Find $\hat{\boldsymbol{\theta}}^*$ that minimises the above expected loss $\mathbb{E}_{\boldsymbol{\theta}|\mathbf{x}}\left[\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})\right]$, ie

$$\hat{\boldsymbol{\theta}}^* = \underset{\hat{\boldsymbol{\theta}}}{\operatorname{argmin}} \quad \mathbb{E}_{\boldsymbol{\theta}|\mathbf{x}}\left[\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})\right]$$

As reported by Hurn, Justel, and Robert (2000), it is not possible to find $\hat{\boldsymbol{\theta}}^*$ explicitly for many choices of $\mathcal{L}$. The good news however is that, for a large class of loss functions, a computationally feasible two-step procedure due to Rue (1995) helps overcome the drawback. Below is a description of the two-step procedure:

[4]The authors report that such a choice is not easy, and in many cases can render some calculations analytically intractable.

Step 1: Use MCMC ergodic averages to approximate $\mathbb{E}_{\boldsymbol{\theta}|\mathbf{x}}\left[\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})\right]$ for a given $\hat{\boldsymbol{\theta}}$.

Step 2: Perform the optimisation of the above estimated expected loss over $\hat{\boldsymbol{\theta}}$.

Note: We did not implement this decision-theoretic approach in our study, mainly because of the difficulty encountered in choosing suitable loss functions in our setting.

3.7 Implementation and Numerical results

Amongst some of the key issues that arise in the numerical application of our sampling scheme are the suitable choice of the fixed values of hyperparameters and the choice of initial parameter values that would speed up convergence. As far as hyperparameters are concerned, Richardson and Green (1997) used data-dependent hyperpriors for mixtures of univariate normals. We simply extend and adapt some of their ideas to our multivariate context along the lines of Stephens (2000). For our mixing proportions $\boldsymbol{\pi}$, we use a Dirichlet prior with $\delta = k$. Such a choice tends to favour a model in which all the components *a priori* have equal weights [5]. Taking $\delta = 1$ would cause the weights to differ significantly. For Σ, we use the same initial values as in Chapter 2, and we choose our fixed hyperparameters α and τ such that $\alpha/\tau = \hat{\sigma}_i^2$, where $\hat{\sigma}_i^2$ is Jöreskog (1975)'s initialisation of Chapter 2. For $\boldsymbol{\mu}$, we use $\xi = (\xi_1, \cdots, \xi_p)^\mathsf{T}$ where ξ_i is the midpoint of the observed range of x_i, for $i = 1, \cdots, p$. We also define R_i as the observed range of x_i, for $i = 1, \cdots, p$, which allows us to choose $\kappa = \mathsf{diag}(\mathsf{R}_1, \cdots, \mathsf{R}_p)$. We use a zero mean ($\eta = 0$) prior for $\boldsymbol{\Lambda}$, and we choose values of g and h that favour draws of Λ_j that are similar (but not equal). Our initial values for Λ_j are the same as in Chapter 2.

3.7.1 Artificial data: Example 2 revisited

It is fair to stress here that this example is purely illustrative. Hence, despite its simplicity shown by well separated components in Figure (2.3), we use it to compare the

[5] This choice is sensible in our context because we mainly aimed at analysing datasets for which the number of mixture component is assumed to be small.

performances of the EM algorithm and Data Augmentation in parameter estimation and density estimation. We saw in Chapter 2 that our single factor model was unable to learn the underlying structure of this 3-component MFA. Recall that $p = 9$ and $q = 2$ in this case. The data in this example come from an MFA with $\pi_1 = 0.3, \pi_2 = 0.45, \pi_3 = 0.25$. With $n = 300$, this corresponds to $n_1 = 90, n_2 = 135, n_3 = 75$. The matrices of factor loadings, the vectors of means and the specific covariance matrix Σ are the same as in Chapter 2. The observed-data log-likelihood for this toy problem is $\ell(\boldsymbol{\theta}, \mathbf{X}) = -2256.03$

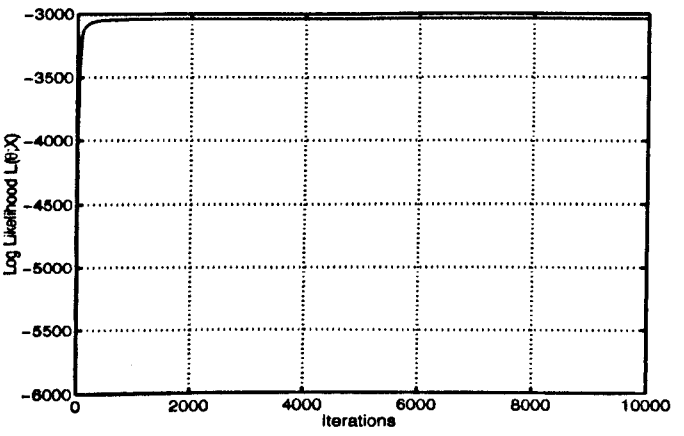


Figure 3.3: Observed-data log-likelihood for Example 2 from a 3-component MFA.

EM solution: Despite the use of many different starting values, the EM algorithm consistently ends up getting trapped in a rather meaningless local maximum. As Figure (3.3) shows, the algorithm falls into the local maximum after fewer than 500 iterations, and despite the 9500 subsequent iterations, it remains trapped and never gets out, leading to a rather poor performance on such a simple task. For instance, as far as density estimation is concerned, the observed-data log-likelihood produced by this EM solution is $\hat{\ell} = -3040.00$, which is very far from the global maximum $\ell(\boldsymbol{\theta}, \mathbf{X}) = -2256.03$. Similarly, the clustering produced by such a meaningless local maximum is equally very unsatisfactory. In fact, despite the apparently good estimates of mixing proportions $\hat{\pi_1} = 0.27, \hat{\pi_2} = 0.30, \hat{\pi_3} = 0.42$, the corresponding factor loading matrices are very inaccurate, and the misclassification rate quite high (more than 25%). One could argue

that trying more starting points could lead to a better solution, but it is clear that this sensitivity to initialisation and this inability to escape local maxima constitute serious weaknesses of this algorithm.

Data Augmentation: As expected, the performance of Data Augmentation on this simple task turns out to be very satisfactory. With $T_o = 9500$ burn-in iterations and $M = 1500$ MCMC samples, the algorithm yields very good density estimation, accurate and precise parameter estimates and perfect clustering.

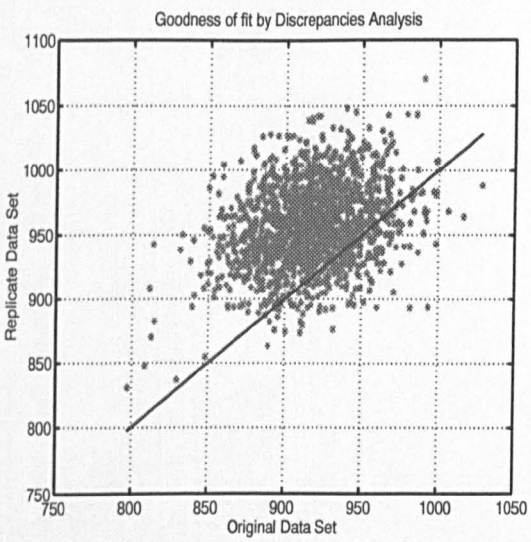
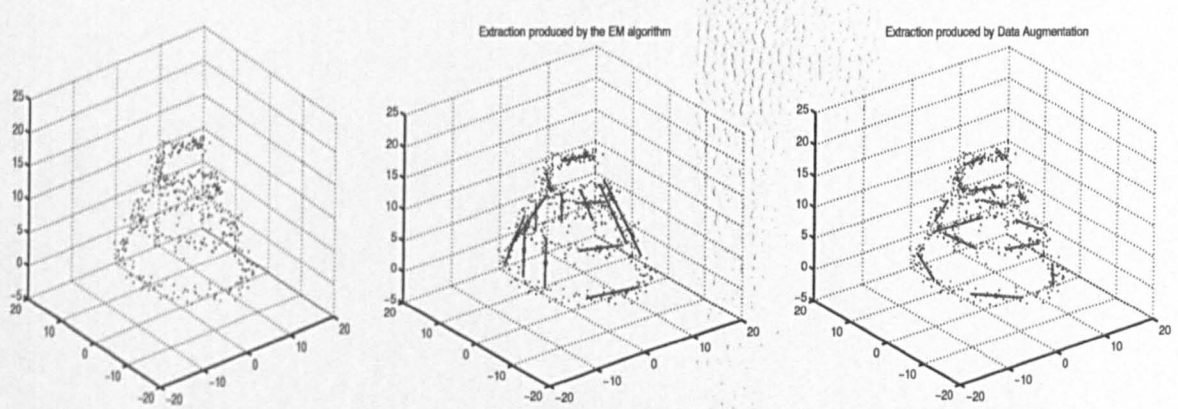


Figure 3.4: Scatterplot of discrepancies for Example 2 analysed with a 3-component MFA.

In fact, here the estimated observed-data log-likelihood produced by Data Augmentation is $\hat{\ell} = -2324.00$ which is much closer to the true value $\ell(\boldsymbol{\theta}, \mathbf{X}) = -2256.03$ than the EM estimate $\hat{\ell} = -3040.00$ is. On the other hand, the estimated mixing weights exactly equal the true values, and the algorithm achieves 100% good clustering rate. As shown by Figure (3.4), the posterior predictive assessment of model fitness looks very much in favour of the plausibility of our proposed 3-component MFA model. Overall, Data Augmentation clearly outperforms the EM algorithm on this task.

3.7.2 The noisy shrinking spiral

This problem was proposed by Ueda, Nakano, Ghahramani, and Hinton (2000) whose aim was apparently to show how one could extract a one-dimensional manifold from a 3-dimensional system. The MFA model appeared to be a good candidate for modelling such a task. The authors in their original work used an MFA with $k = 14$ components, and $q = 1$ (one-dimensional). They compared the performance of their Split-and-Merge EM with the generic EM algorithm. They found that SMEM could easily escape local maxima and extract the one-dimensional manifold satisfactorily, while the generic EM algorithm produced a poor extraction as a result of its inability to escape local maxima. In their variational approximation approach to MFAs, Ghahramani and Beal (2000) also used the same example. In this section, we compare the performances of the generic EM algorithm and Data Augmentation on extracting that one-dimensional manifold from this very same three-dimensional noisy shrinking spiral.



(a) Original spiral. (b) EM Algorithm. (c) Data Augmentation.

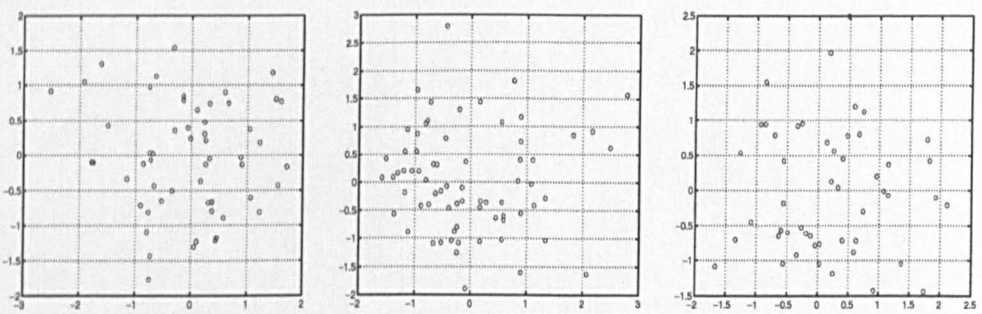
Figure 3.5: Extraction of a one-dimensional manifold from a shrinking spiral.

Comments: We use $k = 14$, and sample size $n = 500$. The lines in (Figure 3.11-(b) and (c)) that are used to plot an estimate of the one-dimensional manifold are obtained using estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$. More specifically, the centre of each line is $\hat{\mu_j}$ (which is the centre or mean of the corresponding local Gaussian), and the direction of the line is given by $\hat{\Lambda_j}$ which in this case is 3-dimensional column vector of factor loadings. For the EM algorithm, we run $T = 10000$ iterations. Despite many starting values, the algorithm

gets trapped in poor local maxima. (Fig 3.11-(b)) shows the poor extraction produced by the local maximum reached by the EM algorithm. A run of Data Augmentation with $T_o = 9500$ and $M = 1500$ MCMC samples produces a very satisfactory extraction as shown by (Fig 3.11-(c)). As expected, Data Augmentation outperforms the generic EM algorithm on this task.

3.7.3 Wine data set (revisited)

We encountered the wine dataset in Chapter 2, and we used the BDMCMC algorithm to find that an estimate of the intrinsic dimensionality of these data would be $q = 6$. We also established the existence of three groups or classes in the data. In this section, we model the data using a 3-component MFA, and we reconsider parameter estimation, density estimation and clustering. In fact, with $p = 13$, the full covariance matrix for each component would have 91 free parameters to be estimated, an estimation that would be very inefficient and prone to overfitting with the small sample of only $n = 178$ observations. The use of the MFA model is therefore justified for this task. We assume that all three hypothetical classes (components) have the same intrinsic dimension q. Using $q = 2$, Data Augmentation produces the estimated posterior expectations of factor



(a) Class 1. (b) Class 2. (c) Class 3.

Figure 3.6: Estimated posterior means of factor scores

scores shown in (Fig 3.6). The good news here is that none of the plots in (Fig 3.6) shows any group structure, meaning that each class is homogeneous. The assumption of the existence of three classes therefore seems reasonable.

Data Augmentation: If we use different values of q (i.e. $q = 1, \cdots, 6$), Data Augmentation yields an overall very good performance in clustering, with the percentage of correct clustering ranging from 95.00% to 98.31%. As expected, the highest log-likelihood is obtained with the value of q found by the BDMCMC, namely $q = 6$, suggesting that the best density estimates would come from an MFA with $q = 6$.

EM algorithm: The performance of the generic EM algorithm on this task is very unsatisfactory. For example, $T = 10000$ iterations of the EM with $q = 6$ lead to a very poor local maximum, yielding a data log-likelihood equal to $\hat{\ell} = -6200.00$ compared to $\hat{\ell} = -3190.00$ produced by Data Augmentation. The corresponding clustering produced by the EM is equally unsatisfactory. Once again, the Data Augmentation algorithm clearly outperforms the EM algorithm.

3.8 Stochastic model selection for MFA

In our treatment of the MFA model, we have so far assumed the number of mixture components k known and fixed. In some cases, we have had to use our BDMCMC for FA to determine the intrinsic dimensionality q of the data, especially in cases where we assumed q to be the same across all the components. At the root of model complexity determination for finite mixtures lie difficult questions such as: (a) *what makes a component a separate and homogeneous entity?* (b) Isn't there always the possibility of a hierarchy of clusters, with a given cluster being made up of its own inner clusters? In Section (2.7), we addressed the stochastic estimation of q. In this section, we concentrate on learning the number of mixture components k, which is generally unknown in practical applications. The problem is very similar to the one treated in Section (2.7), and almost all the difficulties explained in Section (2.7) for the FA model apply *mutatis mutandis* to finite mixtures. For example, the application of classical tests in this context is made very difficult and almost impossible because of the complexity involved in deriving test statistics and their reference distributions. In the Bayesian paradigm, the stochastic simulation approach used in Section (2.7), seems to offer an attractive and

promising framework where such a complex problem is made tractable, and we present elements of such an approach in this section.

FA and finite mixture models have in common the fact that they both involve posterior distributions that are invariant to permutations of the labelling of some of their parameters. In both cases, the collection of parameters can be viewed as a random configuration or point process. Throughout this section, we also treat the case where the number of common factors varies across the clusters. In such a case, each local factor analyser has its own intrinsic dimensionality, $q_j, \quad j = 1, \cdots, k$, and we define the k-dimensional vector $\mathbf{q} = \{q_1, \cdots, q_k\}$. If we assume that $\mathbf{q}$ and k are unknown a priori, then the complete collection of our model parameters becomes $\boldsymbol{\theta} = \{k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma\}$, and our aim in parameter estimation from a stochastic simulation perspective now extends to the construction of an ergodic Markov chain with the joint posterior distribution $\boldsymbol{p}(k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma|\mathbf{X})$ as its equilibrium distribution. In a previous section dedicated to Data Augmentation for MFA, we constructed a Markov chain with $\boldsymbol{p}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma|k, \mathbf{q}, \mathbf{X})$ as its equilibrium distribution. In Chapter 2, we treated FA with unknown q, and we used BDMCMC to estimate q. The extension we are considering here must accommodate our two *counting* random variables k and $\mathbf{q}$. Intuitively, we are in the presence of a two-level nested counting process:

- *Between factor analysers:* simulate a birth-death Markov point process to estimate the number of components k.
- *Within a factor analyser:* simulate a birth-death Markov point process to estimate the number of common factors q_j in each local factor analyser.

If we knew k, then at each iteration we would simply simulate a birth-and-death point process for each j to estimate q_j as described by Algorithm 3. With k unknown, we first need to simulate a current value for k through a process similar to the one described in Section (2.7). If we have reason to assume that q is the same across all the components of the mixture, then the overall process need not be nested.

3.8.1 Model selection between factor analysers

The derivation of the algorithm needed in this section is essentially the same as in Section (2.7). The main differences are the form of the detailed balance equation to be satisfied, and the elements of the configuration set $\mathbf{c}$. We saw earlier that the likelihood $\boldsymbol{L}(k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma|\mathbf{X})$ is invariant under permutations of component labels. It is also easy to see that the prior distribution $\boldsymbol{p}(k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma)$ does not depend on the ordering of component labels. As a result, the posterior

$$\boldsymbol{p}(k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma|\mathbf{X}) \propto \boldsymbol{L}(k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma|\mathbf{X})\boldsymbol{p}(k, \mathbf{q}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \Sigma)^{6}$$

is invariant under permutations of component labels. We are therefore in the presence of a *point process.* If we assume $\mathbf{q}$ known as will be the case at each iteration of our overall MCMC sampling scheme, then the key ingredient for the estimation of k would be the posterior density $\boldsymbol{p}(k, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\mathbf{q}, \mathbf{X})$. For simplicity, we use exactly the same notation as before in the description of the birth-and-death point process. Typically, each point or random configuration in the corresponding point process would therefore be of the form $v_j = (\pi_j, \mu_j, \Lambda_j)$. However, since each Λ_j presupposes a current value for q_j, we use points of the form $v_j = (q_j, \pi_j, \mu_j, \Lambda_j)$ for clarity. We therefore define our configuration variable as

$$\mathbf{c} = \{(q_1, \pi_1, \mu_1, \Lambda_1), (q_2, \pi_2, \mu_2, \Lambda_2), \cdots, (q_k, \pi_k, \mu_k, \Lambda_k)\} = \{v_1, \cdots, v_k\}.$$

The principle behind the simulation of the birth-and-death process is exactly the same as before: *a birth increases the number of mixture components by one ($k \longrightarrow k+1$), while a death decreases it by one ($k \longrightarrow k-1$).* In this case, births should occur in such a way that the mixing proportions in the new configuration sum up to 1. To satisfy this constraint, the birth density is defined as $b(\mathbf{c}, v) = k(1-\pi)^{k-1}\boldsymbol{p}(v)$, where $\boldsymbol{p}(v)$ is the prior density of one element of the configuration $v = (\pi, \mu, \Lambda)$. The death rate is computed in the same way as before. When a new component $v = (\pi, \mu, \Lambda)$ is born, the process jumps into a configuration characterised by the set of points $\mathbf{c} \cup \{v\}$ defined by

[6] For economy of notation, we ignore Σ in the description of our point process, since we are dealing with the case where Σ is the same for all the components, and is therefore fixed in the point process.

$$\mathbf{c} \cup \{v\} := \{(q_1, \pi_1(1-\pi), \mu_1, \Lambda_1), \cdots, (q_k, \pi_k(1-\pi), \mu_k, \Lambda_k), (q, \pi, \mu, \Lambda)\}.$$

When a new component is born, we set the intrinsic dimension q of its corresponding factor analyser to $q = 1$. When a new component $v_j = (\pi_j, \mu_j, \Lambda_j)$ is selected to die, the new configuration of the process is characterised by the set $\mathbf{c}\backslash\{v_j\}$ defined by

$$\mathbf{c}\backslash\{v_j\} := \{(q_1, \frac{\pi_1}{1-\pi_j}, \mu_1, \Lambda_1), \cdots, (q_{j+1}, \frac{\pi_{j+1}}{1-\pi_j}, \mu_{j+1}, \Lambda_{j+1}), (q_k, \frac{\pi_k}{1-\pi_j}, \mu_k, \Lambda_k)\}.$$

With the above birth-and-death process so defined, all we need for its simulation is to make sure that its corresponding Markov chain is irreducible and aperiodic. The following theorem from Stephens (2000) presents a "detailed balance" equation whose satisfaction theoretically guarantees the convergence of the chain to the limiting distribution of interest. We use the function $\boldsymbol{h}$ in the same sense as in Section (2.7).

Theorem 3.1 *If the birth density b and the death density d satisfy*

$$(k+1)d(\mathbf{c} \cup \{v\}; v)\boldsymbol{h}(\mathbf{c} \cup \{v\})k(1-\pi)^{k-1} = \beta(\mathbf{c})b(\mathbf{c}; v)\boldsymbol{h}(\mathbf{c}) \tag{3.33}$$

for all configurations $\mathbf{c}$ and all points v, then the birth-and-death process defined above has $\boldsymbol{p}(k, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\mathbf{q}, \mathbf{X})$ as its stationary distribution.

A description of the algorithm used to simulate the above birth-and-death MCMC scheme for mixtures is given by Algorithm 11, and the overall stochastic model selection for MFA can be described as follows:

Algorithm 10: Stochastic model selection for MFA.

Assuming a current set $(k^{(t)}, \mathbf{q}^{(t)}, \boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Lambda}^{(t)})$ of parameters,
Simulate $k^{(t+1)}$ through a run of Algorithm 11 .
For $j = 1, \cdots, k^{(t+1)}$
  Simulate $q_j^{(t+1)}$ through a run of Algorithm 3.
End
Set $\mathbf{q}^{(t+1)} = (q_1^{(t+1)}, \cdots, q_{k^{(t+1)}}^{(t+1)})$
Simulate $(\boldsymbol{\pi}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Lambda}^{(t+1)})$ via Algorithm 7 given $(k^{(t+1)}, \mathbf{q}^{(t+1)})$

As before, we use a Poisson prior for k, with hyperparameter ϱ. At each iteration t of the overall algorithm, the continuous-time birth-and-death process is simulated as follows:

Algorithm 11: Birth-and-death MCMC mixtures.

Set $\beta(\mathbf{c}) = \beta$, $t_{mm} = 0$ and $k = k^{(t-1)}$
Repeat
  Compute $\delta_j(\mathbf{c}) := \dfrac{\boldsymbol{L}(\mathbf{c}\backslash v_j)}{\boldsymbol{L}(\mathbf{c})}\dfrac{\beta}{\varrho}$ for $j = 1, \cdots, k$
  Compute $\delta(\mathbf{c}) := \sum_{j=1}^{q} \delta_j(\mathbf{c})$
  Simulate $s \sim \mathsf{Exp}(1/(\beta(\mathbf{c}) + \delta(\mathbf{c})))$ and Set $t_{mm} := t_{mm} + s$
  If $(\mathsf{Ber}(\beta(\mathbf{c})/(\beta(\mathbf{c}) + \delta(\mathbf{c})) = 1)$ /* It is a birth */
    Set $k = k + 1$
    Simulate (μ, π, Λ) from $\boldsymbol{p}(v|\iota)$ and set $q = 1$
    Set $\mathbf{c} := \mathbf{c} \cup \{(q, \pi, \mu, \Lambda)\}$
  Else /* It is a death */
    Simulate $j' = \mathsf{Mn}(\delta_1(\mathbf{c})/\delta(\mathbf{c}), \cdots, \delta_k(\mathbf{c})/\delta(\mathbf{c}))$
    Set $\mathbf{c} := \mathbf{c}\backslash\{(q_{j'}, \pi_{j'}, \mu_{j'}, \Lambda_{j'})\}$
    Set $k := k - 1$
Until $(t_{mm} \geq \rho)$

The algorithm combines the simulation of birth-death point processes between and within factor analysers with the basic Bayesian sampling to concurrently perform model selection and the corresponding parameter estimation.

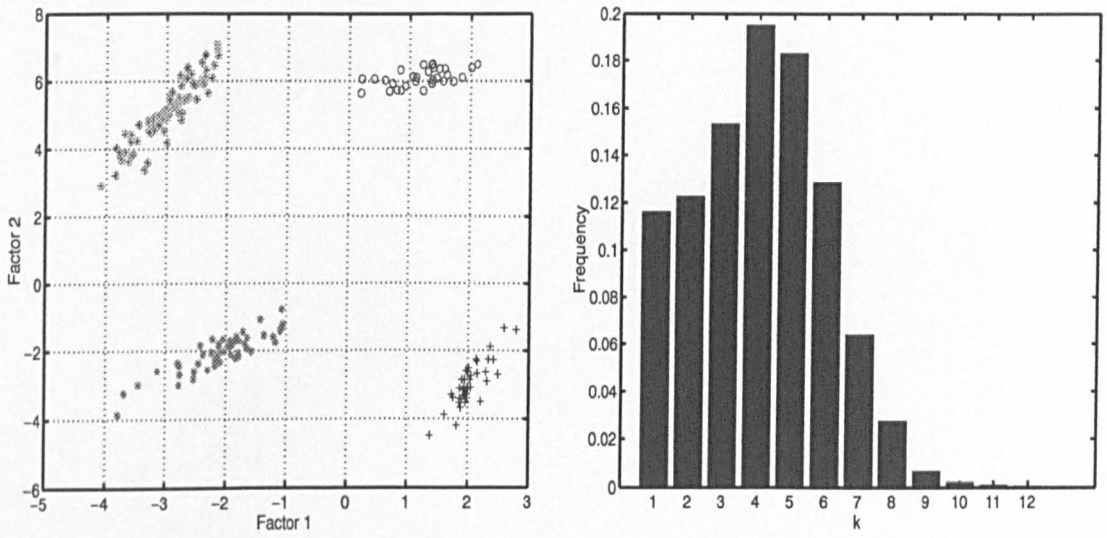
3.9 Numerical examples of model selection

In this section, we have mainly analysed examples where the intrinsic dimensionality q, although unknown, is the same across all components. Examples with different intrinsic dimensionalities require only minor changes, and will be treated in our future work.

3.9.1 Artificial problem with 4 components

This first example is purely illustrative, and is mainly aimed at testing the performance of our scheme on a simple task. At each iteration t of the overall process, the birth-and-death process is run for a time period $\rho = 1.618$. The overall constant birth rate in this case is $\beta = 0.618$, and we take hyperparameter $\varrho = \beta$. The components of the mixture in this case are well separated, and the dimensionalities of both $\boldsymbol{x}$ and $\boldsymbol{z}$ are small ($p = 2$ and $q = 1$). The task at hand is therefore simple enough for the scheme to be able to

solve it easily. Since we assume that q is the same across all the components, we first run



(a) 2D Projection of the data

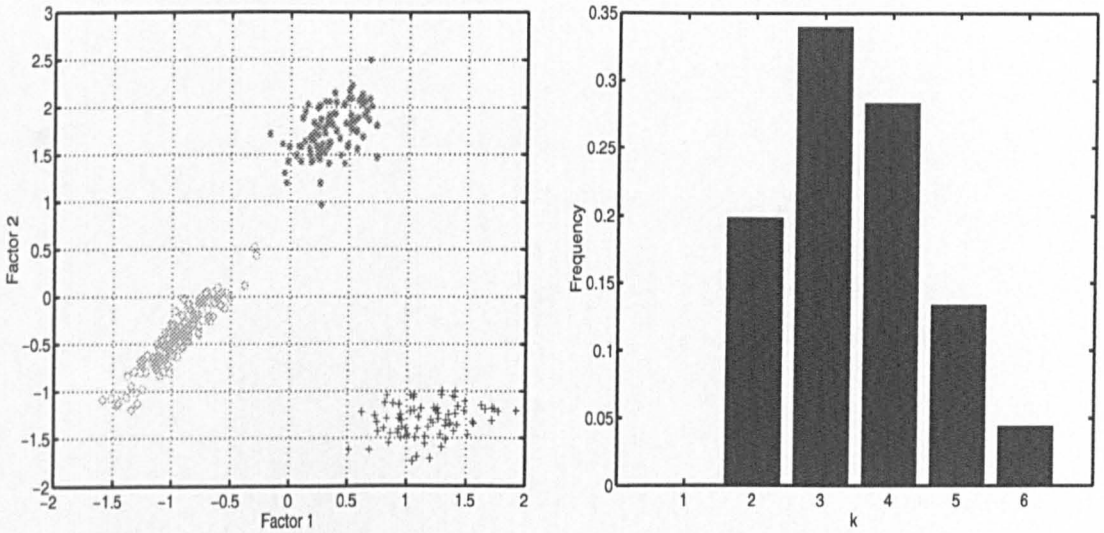
(b) Estimation of $\Pr(k = i|\mathbf{X})$

Figure 3.7: 4-component MFA with histograms approximating $\mathbf{Pr}(k = i|\mathbf{X})$

Algorithm 4 separately, and it finds no problem determining that q for these data is equal to 1. If we use the known value of q, the overall stochastic simulation scheme for MFA produces an estimation of $\Pr(k = i|\mathbf{X})$ as shown in Figure (3.7) after 30000 iterations. The estimation of the number of components k in this case is satisfactorily. However, it is worth pointing out the fact that other values of k do have reasonably high frequency in the chain. This could be attributed to a birth-rate allowing a good exploration of all possible configurations, in which case a larger number of iterations would be required to reach an equilibrium distribution that peaks on the true value of k.

3.9.2 Artificial data: Example 2 visited yet again

We have already encountered this example twice now. In this section, we use the estimated value of q, and we test the performance of the birth-and-death process in determining the number of components of our mixture. $\beta = 0.618$ turns out to be a good value for our overall constant birth rate. After 30000 iterations, the algorithm produces a very good approximation of $\Pr(k = i|\mathbf{X})$ as shown in Figure (3.8), and as expected, the estimated value of k, that is $k = 3$, is correct.



(a) 2D Projection of the data

(b) Estimation of $\Pr(k = i|\mathbf{X})$

Figure 3.8: 3-component MFA with histogram approximating $\mathbf{Pr}(k = i|\mathbf{X})$

3.9.3 Wine data set (revisited)

As we said earlier, it is believed that there are three (3) types of wines in these data. Our aim in this subsection is to estimate k, and to compare this estimated value to the hypothesised $k = 3$. With $q = 3$ and $\beta = 0.15$, the approximation of $\Pr(k = i|\mathbf{X})$ after 15000 iterations is given by Figure (3.9).

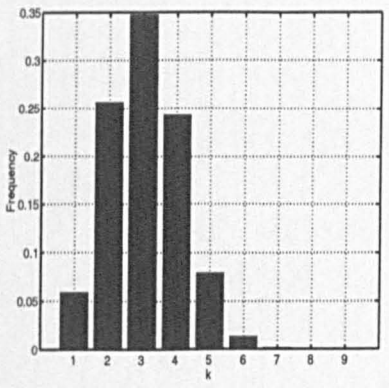
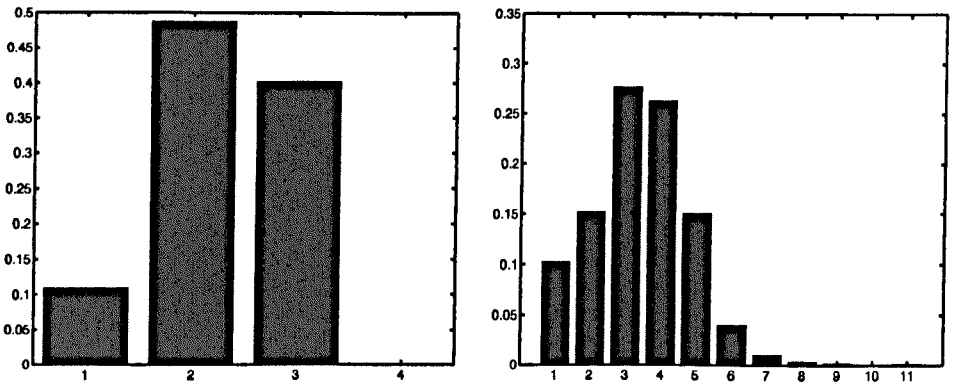


Figure 3.9: Estimation of $\mathbf{Pr}(k = i|\mathbf{X})$ for the wine data.

3.9.4 Iris data set

This is probably one of the most used datasets in statistical analysis. With $p = 4$, it is fair to recognise that this is clearly not a high-dimensional data. However, for the sake of illustration, we shall use our scheme to estimate both q and k for this dataset.



(a) Estimation of $\mathbf{Pr}(q = i|\mathbf{X})$. (b) Estimation of $\Pr(k = i|\mathbf{X})$.

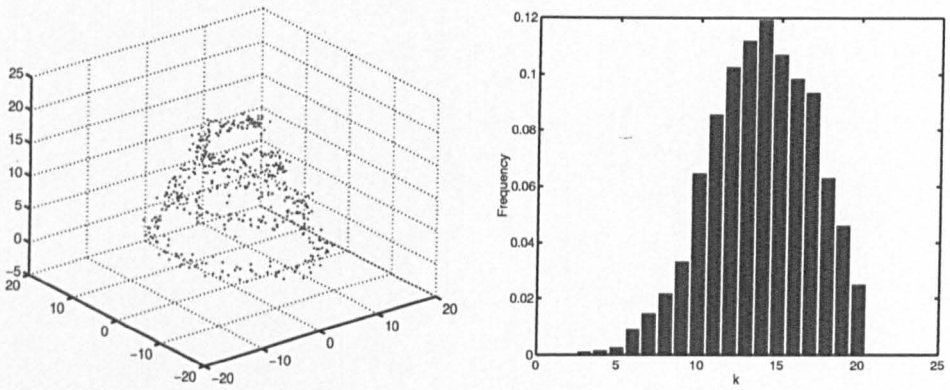
Figure 3.10: Plots for the Iris data

We first assume that all the 3 hypothesised classes of iris have the same intrinsic dimensionality q, and we use our BDMCMC for Factor Analysis to estimate q. As seen on Figure (3.10)-left, our simulations seem to be suggesting that $q = 2$ could be the intrinsic dimensionality of the iris data. If we use $q = 2$ and $\beta = 0.618$, a run of 10000 iterations of the BDMCMC scheme for MFA yields an approximate distribution for k as shown in Figure (3.10)-right. From these findings, It seems therefore pretty likely that $k = 3$ could be the number of types of iris.

3.9.5 Model selection for the spiral data

We encountered the spiral data earlier, and we successfully extracted the underlying one-dimensional intrinsic manifold using Data Augmentation on an MFA with $k = 14$. In this last part of our numerical examples, we use the birth-and-death process to estimate k. For this example, $\beta = 3.3$ turns out to be a "good" overall constant birth rate for the process. Compared to the much smaller birth rates used so far, this birth rate causes the overall scheme to run for much longer before producing the output. With a run

of $T = 2000$ iterations, the corresponding approximation to the posterior distribution $\Pr(k|\mathbf{X})$ of k is given below



(a) The spiral data. (b) Estimation of $\mathbf{Pr}(k = i|\mathbf{X})$.

Figure 3.11: The spiral data: how many components?

Figure (3.11)-right clearly suggests that k for the spiral data is likely to be between 12 and 15, with $k = 14$ having the highest frequency as expected.

3.10 Discussion

We have developed a stochastic simulation based algorithm for the analysis of the Mixture of Factor Analysers model. Our experiments show that our approach performs well in parameter estimation, clustering, density estimation and model selection. We have not yet tested our sampling scheme on very high-dimensional tasks like handwritten digits recognition or image reconstruction, but we are actively working on devising faster sampling schemes that should handle such tasks in practically acceptable computing times. The bulk of our computational burden lies in sampling from the Gamma and the Multivariate Gaussian distributions. For large values of q, this sampling can be computationally very demanding. There is therefore a need to concentrate on this important computational issue, maybe by devising more efficient ways to sample from a multivariate Gaussian. One of the main drawbacks of Data Augmentation in this very high-dimensional context is the fact that the computation of ergodic averages for factor scores requires the storage of a large number of high-dimensional variables through-

out the sampling process. This easily becomes explosive even for problems with latent spaces of moderately high dimensions. We showed earlier how this difficulty can be circumvented by avoiding the storage of latent variables during the sampling process. Our simulations reveal that the use of an extra layer in the hierarchical prior structure effectively eliminates singularities and therefore achieves an advantage over the EM algorithm, for which we noticed many occurrences of singularities. However, despite escaping singularities, we still noticed rather poor mixing of the chains when k and q were known and fixed. An improvement on this might come from the use of tempered transitions, and we are exploring a simulated tempering version of our algorithm to achieve better exploration of the posterior surface. We only used vague conjugate priors throughout our study. We did this partly for computationally convenience, but also because these priors have produced excellent results in similar contexts Richardson and Green (1997), Diebolt and Robert (1994) and have somehow become standard. It would be nice to be fully Bayesian and consider the use of more informative priors, but their incorporation in the sampling scheme could be very difficult and could destroy some nice properties of the Markov chains. Our adaptation of BDMCMC to Factor Analysis is probably the aspect of our proposed method that does not require much extra work, apart from the need to use data-dependent birth rates. It works very well so far on both synthetic and real-life tasks. The nested scheme, however, requires some improvements, especially on the derivation of an adaptive birth rate that would evolve dynamically as a likelihood-related function, allowing only likely models to be born. We are exploring ideas from van Lieshout (1994), Stoyan, Kendall, and Mecke (1995), Barndorff-Nielsen, Kendall, and van Lieshout (1999) to find solutions to this problem. Overall, our results suggest that the scheme we have proposed is a good alternative to other existing methods such as the EM algorithm, the SMEM and Variational approximations. We believe that a careful study of the limitations noticed so far would lead to better sampling schemes that would then be fully applicable to truly high-dimensional Machine Learning tasks.

Chapter 4

Analysis of the Effect of Covariates

Creativity, as has been said, consists largely of rearranging what we know in order to find out what we do not know... Hence, to think creatively, we must be able to look afresh at what we normally take for granted.
George Kneller

We present an extension of the Mixture of Factor Analysers model that investigates the effect of fixed observed covariates on both the continuous latent variable (common factor) and the discrete categorical latent variable (component label). The extended model allows us to study, not just the relationship between the manifest and the latent variables, but also the influence of external fixed observed covariates on the latent variables. Such an extension gives more ingredients and greater flexibility in developing a better and more realistic model. We assume a linear model with some Gaussian noise relating the continuous latent variable to its corresponding covariate, and we use a polytomous logistic regression model to link the discrete categorical latent variable to its corresponding covariate. We then derive an EM algorithm for estimating the parameters of the new model. Application of the algorithm to synthetic tasks yields good performance under suitably chosen initial conditions. This chapter is essentially an extension of Fokoué and Titterington (2000c).

4.1 Introduction

In the previous chapter, we presented a Bayesian sampling approach to the analysis of the generic MFA model, and we reviewed the main ingredients of the EM algorithm used for the Maximum Likelihood estimation of parameters. However, the MFA model, as we have studied it so far, focuses solely on the relationship between the manifest variables and the latent variables. This can lead to a neglect of useful information when the latent and/or manifest variables are related to fixed observable covariates. In this chapter, we only model the effect of covariates on the latent variables. An extension that allows covariates on the manifest variables is straightforward. Our extension of the MFA model is similar to previous work by various authors. Lee and Shi (1999) have studied an extension of the Structural Equation Model (SEM) by allowing fixed observed covariates on both the manifest and the latent variables, and have used a Bayesian sampling approach for inference and estimation. Thompson, Smith, and Boyle (1998) have incorporated concomitant information into fixed observed covariates on both the manifest and the latent variables in their assessment of diagnostic criteria for diabetes using a two-component finite mixture model. Muthén and Shedden (1999) use fixed covariates in their study of the extension of finite mixture models with mixture outcomes. Finally, Sammel, Ryan, and Legler (1997) also found it useful to incorporate fixed covariates in their study of latent variable models for mixed discrete and continuous outcomes. The use of fixed observed covariates in the MFA model therefore seems to be justified by such great practical interest. In the first part of this chapter, we give a brief review of some key ingredients of the MFA model needed in this context. We then present the mechanisms by which the covariates are incorporated into the model, after which we give a description of how the EM algorithm is derived for the extended MFA model together with some expressions used in the iterative EM process. The last part is dedicated to simulations on artificial tasks.

4.2 Modelling the Effect of Covariates

Since we are studying an extension of the MFA model, we shall refer to our previous form of the model as the *generic MFA* model. To simplify our description, we shall restrict ourselves to generic MFA models with $q_j = q$ and $\Sigma_j = \Sigma$ for $j = 1, \cdots, k$. The generative equation of the corresponding generic MFA model in such cases is therefore

$$\boldsymbol{x} = \Lambda_j \boldsymbol{z} + \mu_j + \boldsymbol{e}, \quad j = 1, \cdots, k. \tag{4.1}$$

The main motivation for incorporating covariates into the model can be simply stated as follows: *latent variables are related to manifest variables via the mechanism that we have so far modelled with the generic MFA model. However, situations may arise in which those same latent variables are also related to other observables via other mechanisms.* As far as the MFA model is concerned, we shall focus in this section on the introduction of two such additional mechanisms: one for the continuous latent variable $\boldsymbol{z}$ and the other for the discrete categorical latent variable $\boldsymbol{y}$. Throughout this chapter, we shall assume that k and q are known and fixed.

We first assume that each continuous latent variable $\boldsymbol{z_i}$ is related to a fixed observed covariate $\boldsymbol{w_i} \in I\!R^r$ through the multivariate linear regression model

$$\boldsymbol{z_i} = \Phi \boldsymbol{w_i} + \boldsymbol{v_i}, \tag{4.2}$$

where Φ is the $q \times r$ matrix of regression parameters, and $\boldsymbol{v_i} \in I\!R^q$ is the error or disturbance term with, $\boldsymbol{v_i} \sim \mathcal{N}(0, \Psi)$. As earlier, we restrict ourselves to an orthogonal[1] factor structure, and we therefore assume Ψ to be diagonal, that is $\Psi = \mathsf{diag}(\psi_1, \cdots, \psi_q)$. Moreover, since the estimation equations in factor analysis are invariant with respect to scale changes in the factors, as we explained in Chapter 2, we retain only the simplest covariance matrix for $\boldsymbol{z_i}$, that is $\Psi = \mathbf{I}_q$. Thus, each $\boldsymbol{z_i}$ has a multivariate Gaussian distribution, $\boldsymbol{z_i} \sim \mathcal{N}_q(\Phi \boldsymbol{w_i}, \mathbf{I}_q)$. Essentially, the change brought by the covariate is that the factor score now has a *nonzero* mean, as opposed to the zero mean assumption used for the generic MFA model. It is possible to imagine a more general extension in

[1] We assume factor scores to be uncorrelated.

which there is a different Φ_j for each component j of the mixture, and where Ψ is a full variance-covariance matrix reflecting the fact that factors are allowed to be correlated. We restrict ourselves to the case of identical Φ and $\Psi = \mathbf{I}_q$.

We also assume that the discrete categorical latent variable $\boldsymbol{y}$ is subject to the influence of a fixed observed covariate, $\boldsymbol{u}$, say. Since $\boldsymbol{y}$ takes its values from $\{1, \cdots, k\}$, a good candidate for dealing with this is the widely used polytomous logistic regression model. Given a vector $\boldsymbol{u} \in I\!R^s$ of covariates, the unconditional classification probabilities are therefore defined through the logit model as follows:

$$\log\left[\frac{\Pr(\boldsymbol{y}=j|\boldsymbol{u})}{\Pr(\boldsymbol{y}=k|\boldsymbol{u})}\right] = \phi_{0j} + \tilde{\boldsymbol{\phi}}_j^{\mathsf{T}}\tilde{\boldsymbol{u}} = \boldsymbol{\phi}_j^{\mathsf{T}}\boldsymbol{u} = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{\phi}_j \quad \text{for} \quad j = 1, \cdots, k-1, \tag{4.3}$$

where $\tilde{\boldsymbol{\phi}}_j^{\mathsf{T}} = (\phi_{1j}, \cdots, \phi_{s-1,j}) \in I\!R^{s-1}$ and $\boldsymbol{\phi}_j^{\mathsf{T}} = (\phi_{0j}, \tilde{\boldsymbol{\phi}}_j^{\mathsf{T}}) = (\phi_{0j}, \phi_{1j}, \cdots, \phi_{s-1,j})^{\mathsf{T}} \in I\!R^s$, for $j = 1, \cdots, k-1$. In the same way, $\boldsymbol{u}^{\mathsf{T}} = (1, \tilde{\boldsymbol{u}}^{\mathsf{T}}) \in I\!R^s$, with $\tilde{\boldsymbol{u}} \in I\!R^{s-1}$. For identifiability, we set $\boldsymbol{\phi}_k = 0$. It is easy to show from (4.3) that the classification probabilities are given by

$$\Pr(\boldsymbol{y}=j|\boldsymbol{u}) = \begin{cases} \dfrac{\exp(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{\phi}_j)}{1+\sum_{j'=1}^{k-1}\exp(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{\phi}_{j'})} & \text{for} \quad j = 1, \cdots, k \\ \dfrac{1}{1+\sum_{j'=1}^{k-1}\exp(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{\phi}_{j'})} & \text{for} \quad j = k. \end{cases} \tag{4.4}$$

For simplicity and convenience, we define $\pi_{ij}(\boldsymbol{u}_i, \boldsymbol{\phi}_j) = \Pr(\boldsymbol{y}_i = j|\boldsymbol{u}_i)$ for $j = 1, \cdots, k$ and $i = 1, \cdots, n$. As we shall see later, it turns out to be more convenient to reformulate our model here as a multivariate Generalised Linear Model (GLM) for multicategorical responses. More specifically, we now consider the $(k-1)$-dimensional vector of indicator variables $\boldsymbol{y}_i = (y_{i1}, \cdots, y_{i,k-1})^{\mathsf{T}}$. We define $\boldsymbol{U}_i \in I\!R^{(k-1)s \times (k-1)s}$, $\boldsymbol{\phi} \in I\!R^{(k-1)s}$, and $\boldsymbol{\pi}_i \in I\!R^{k-1}$ as follows:

$$\boldsymbol{U}_i = \begin{bmatrix} \boldsymbol{u}_i^{\mathsf{T}} & & & \\ & \boldsymbol{u}_i^{\mathsf{T}} & & \\ & & \ddots & \\ & & & \boldsymbol{u}_i^{\mathsf{T}} \end{bmatrix}, \quad \boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{\phi}_1 \\ \boldsymbol{\phi}_2 \\ \vdots \\ \boldsymbol{\phi}_{k-1} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\pi}_i = \begin{bmatrix} \pi_{i1} \\ \pi_{i2} \\ \vdots \\ \pi_{i,k-1} \end{bmatrix}. \tag{4.5}$$

From the above definitions, the **systematic component** of our GLM for a given covariate $\boldsymbol{u}_i$ is the vector $\boldsymbol{\eta}_i = \boldsymbol{U}_i\boldsymbol{\phi} = (\eta_{i1}, \cdots, \eta_{i,k-1})^\mathsf{T}$, with $\eta_{ij} = \boldsymbol{u}_i^\mathsf{T}\boldsymbol{\phi}_j$, for $j = 1, \cdots, k-1$. The response function here is a vector-valued function $\boldsymbol{f} = (\boldsymbol{f}_1, \cdots, \boldsymbol{f}_{k-1})$, with

$$\boldsymbol{f}_j(\boldsymbol{\eta}_i) = \boldsymbol{f}_j(\eta_{i1}, \cdots, \eta_{i,k-1}) = \frac{\exp(\eta_{ij})}{1 + \sum_{j'=1}^{k-1} \exp(\eta_{ij'})}, \quad j = 1, \cdots, k-1, \tag{4.6}$$

which allows us to express the $\boldsymbol{\pi}_i$ of equation (4.5) as $\boldsymbol{\pi}_i = \boldsymbol{f}(\boldsymbol{\eta}_i) = \boldsymbol{f}(\boldsymbol{U}_i\boldsymbol{\phi})$. Expressed in terms of the link function of the logit model, we have $\boldsymbol{\eta}_i = \boldsymbol{g}(\boldsymbol{\pi}_i) = \boldsymbol{U}_i\boldsymbol{\phi}$, where $\boldsymbol{g} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_{k-1})$ is a vector-valued function such that

$$\boldsymbol{g}_j(\boldsymbol{\pi}_i) = \boldsymbol{g}_j(\pi_{i1}, \cdots, \pi_{i,k-1}) = \log\left[\frac{\pi_{ij}}{1 - (\pi_{i1} + \cdots + \pi_{i,k-1})}\right]. \tag{4.7}$$

The variance-covariance matrix for a given categorical variable $\boldsymbol{y}_i = (y_{i1}, \cdots, y_{i,k-1})^\mathsf{T}$ is

$$\mathsf{cov}(\boldsymbol{y}_i) = \boldsymbol{C}_i = \boldsymbol{C}_i(\boldsymbol{\phi}) = \begin{bmatrix} \pi_{i1}(1-\pi_{i1}) & -\pi_{i1}\pi_{i2} & \cdots & -\pi_{i1}\pi_{i,k-1} \\ -\pi_{i2}\pi_{i1} & \pi_{i2}(1-\pi_{i2}) & \cdots & -\pi_{i2}\pi_{i,k-1} \\ \vdots & \vdots & \ddots & \\ -\pi_{i,k-1}\pi_{i1} & \cdots & \cdots & \pi_{i,k-1}(1-\pi_{i,k-1}) \end{bmatrix}. \tag{4.8}$$

It is easy to verify that $\boldsymbol{C}_i = \mathsf{diag}(\boldsymbol{\pi}_i) - \boldsymbol{\pi}_i\boldsymbol{\pi}_i^\mathsf{T}$. With $\boldsymbol{\phi} = \{\boldsymbol{\phi}_1, \cdots, \boldsymbol{\phi}_{k-1}\}$, our complete collection of model parameters is now $\boldsymbol{\theta} = \{\boldsymbol{\phi}, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\Phi}\}$.

Note: For economy of notational space, we shall omit the explicit mention of covariates and parameters in many of our expressions of probability densities and expectations, unless a need for clarity requires it. For instance, we shall simply write $[\boldsymbol{x}_i|\boldsymbol{y}_i = j]$ instead of $[\boldsymbol{x}_i|\boldsymbol{y}_i = j, \boldsymbol{w}_i, \boldsymbol{\theta}]$, and $\Pr(\boldsymbol{y} = j)$ instead of $\Pr(\boldsymbol{y} = j|\boldsymbol{u}, \boldsymbol{\theta})$.

4.3 Elements of estimation and inference

Modelling the effect of covariates on latent variables can only be fully justified if the estimation of latent scores plays a central (key) role in the statistical analysis being carried out. It is therefore important in this context to concentrate a large amount of effort on addressing the estimation of both posterior expectations of factor scores

and posterior classification probabilities. Parameter estimation obviously remains the prime focus, since the other inferential tasks depend on it. If we use all the basic assumptions of the traditional factor model as seen in the previous chapters, it is easy to see that $[\boldsymbol{x}_i|\boldsymbol{y}_i = j] \sim \mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \Phi \boldsymbol{w}_i, \Lambda_j \Lambda_j^\mathsf{T} + \Sigma)$. Since $\boldsymbol{z}$ and $\boldsymbol{e}$ are assumed to be independent,

$$\text{cov}(\boldsymbol{z}, \boldsymbol{x}) = \mathbb{E}\left[\boldsymbol{z}\boldsymbol{x}^\mathsf{T}\right] - \mathbb{E}\left[\boldsymbol{z}\right]\left(\mathbb{E}\left[\boldsymbol{x}\right]\right)^\mathsf{T} = \Lambda_j^\mathsf{T} - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T}\Lambda_j^\mathsf{T} = (\mathbf{I}_q - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T})\Lambda_j^\mathsf{T}, \quad (4.9)$$

and, as a result, in each component j of the mixture, we have the following distribution:

$$\begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{x} \end{bmatrix} \sim \mathcal{N}_{(q+p)}\left(\begin{bmatrix} \Phi \boldsymbol{w} \\ \mu_j + \Lambda_j \Phi \boldsymbol{w} \end{bmatrix}, \begin{bmatrix} \mathbf{I}_q & (\mathbf{I}_q - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T})\Lambda_j^\mathsf{T} \\ \Lambda_j(\mathbf{I}_q - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T}) & \Sigma + \Lambda_j\Lambda_j^\mathsf{T} \end{bmatrix}\right).$$

From theorem (A.1), we derive $[\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y} = j] \sim \mathcal{N}_q(m_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y}=j}, C_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y}=j})$, where

$$\begin{aligned} m_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y}=j} &= \Phi \boldsymbol{w} + (\mathbf{I}_q - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T})\Lambda_j^\mathsf{T}\left(\Lambda_j\Lambda_j^\mathsf{T} + \Sigma\right)^{-1}(\boldsymbol{x} - \mu_j - \Lambda_j \Phi \boldsymbol{w}). \\ C_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y}=j} &= \mathbf{I}_q - (\mathbf{I}_q - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T})\Lambda_j^\mathsf{T}\left(\Lambda_j\Lambda_j^\mathsf{T} + \Sigma\right)^{-1}\Lambda_j(\mathbf{I}_q - \Phi \boldsymbol{w}\boldsymbol{w}^\mathsf{T}\Phi^\mathsf{T}). \quad (4.10) \end{aligned}$$

Thus, given an observation $\boldsymbol{x}_i$, a covariate $\boldsymbol{w}_i$, an assumed value y_{ij} of the label of $\boldsymbol{x}_i$ and a set of parameters $\boldsymbol{\theta}$, an estimate of the expected factor score is given by

$$\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{w}_i, \boldsymbol{x}_i, \boldsymbol{y}_i = j\right] = \Phi \boldsymbol{w}_i + (\mathbf{I}_q - \Phi \boldsymbol{w}_i\boldsymbol{w}_i^\mathsf{T}\Phi^\mathsf{T})\Lambda_j^\mathsf{T}\left(\Lambda_j\Lambda_j^\mathsf{T} + \Sigma\right)^{-1}(\boldsymbol{x}_i - \mu_j - \Lambda_j \Phi \boldsymbol{w}_i). \quad (4.11)$$

It is also easy to show that the posterior classification probabilities are now given by

$$\Pr(y_{ij} = 1|\boldsymbol{x}_i) = \frac{\pi_{ij}\mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \Phi \boldsymbol{w}_i, \Lambda_j\Lambda_j^\mathsf{T} + \Sigma)}{\sum_{j'=1}^{k} \pi_{ij'}\mathcal{N}_p(\boldsymbol{x}_i; \mu_{j'} + \Lambda_{j'} \Phi \boldsymbol{w}_i, \Lambda_{j'}\Lambda_{j'}^\mathsf{T} + \Sigma)}, \quad (4.12)$$

where $\pi_{ij} = \pi_{ij}(\boldsymbol{u}_i, \boldsymbol{\phi}_j) = \Pr(y_{ij} = 1)$. Here, y_{ij} is an indicator variable as defined earlier, and it is easy to see that $\mathbb{E}\left[y_{ij}|\boldsymbol{x}_i\right] = \Pr(y_{ij} = 1|\boldsymbol{x}_i)$.

4.4 Parameter estimation via the EM algorithm

The above estimates of posterior expected factor scores (4.11) and posterior classification probabilities (4.12) presuppose the existence of a set of parameter estimates. In this

chapter, we only tackle parameter estimation from a likelihood-based perspective via the EM algorithm. The EM algorithm for this extended MFA model makes extensive use of elements from Chapter 3. In fact, the joint density of all the variables is now

$$\boldsymbol{p}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{u})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{w}), \tag{4.13}$$

and the corresponding complete-data log-likelihood of the model is therefore given by

$$\ell(\boldsymbol{\theta}; \mathbf{X}^*) = \sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij} \log \boldsymbol{p}(\boldsymbol{x}_i|y_{ij} = 1, \boldsymbol{z}_i) + \sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij} \log \pi_{ij} + \sum_{i=1}^{n} \log \boldsymbol{p}(\boldsymbol{z}_i|\boldsymbol{w}_i). \tag{4.14}$$

4.4.1 Constructing the E-step

As usual, we need to construct an analytical expression of the expectation of the complete-data log-likelihood $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ with respect to the joint conditional distribution of our latent variables $(\boldsymbol{y}, \boldsymbol{z})$ given $\mathbf{X}$ and $\boldsymbol{\theta}^{(t)}$, which is defined as

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{(\boldsymbol{y},\boldsymbol{z})}\left[\ell(\boldsymbol{\theta}, \mathbf{X}^*)|\mathbf{X}, \boldsymbol{\theta}^{(t)}\right] = \int_{\mathcal{H}} \ell(\boldsymbol{\theta}, \mathbf{X}^*)\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\mathbf{X}, \boldsymbol{\theta}^{(t)})d\boldsymbol{y}d\boldsymbol{z}. \tag{4.15}$$

Our expectations are taken with respect to the joint distribution of $(\boldsymbol{y}, \boldsymbol{z})$ conditional on $\mathbf{X}$ and $\boldsymbol{\theta}^{(t)}$, and we therefore simply use $\mathbb{E}$ instead of $\mathbb{E}_{(\boldsymbol{y},\boldsymbol{z})}$. Based on the expression of $\ell(\boldsymbol{\theta}, \mathbf{X}^*)$ in equation (4.14), the formation of an analytical expression for $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ in (4.15) requires analytical expressions for $\mathrm{a}_{ij}^{(t)} = \mathbb{E}\left[y_{ij}|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right]$, $\mathrm{b}_{ij}^{(t)} = \mathbb{E}\left[\boldsymbol{z}_i|y_{ij} = 1, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right]$, $\mathbf{C}_{ij}^{(t)} = \mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|y_{ij} = 1, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right]$, $\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right]$ and finally $\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right]$. From the fact that $\mathbb{E}_{(\boldsymbol{y},\boldsymbol{z})}\left[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right] = \mathbb{E}_{\boldsymbol{y}}\left[\mathbb{E}_{\boldsymbol{z}}\left[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{\theta}^{(t)}\right]\right]$, we easily derive

$$\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right] = \sum_{j=1}^{k} \mathrm{a}_{ij}^{(t)}\mathrm{b}_{ij}^{(t)} \quad \text{and} \quad \mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}\right] = \sum_{j=1}^{k} \mathrm{a}_{ij}^{(t)}\mathbf{C}_{ij}^{(t)}. \tag{4.16}$$

With the above expressions clearly defined, the derivation of the expression of $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ turns out to be straightforward, making the E-step an easy one in this case. However, as we shall see later, some of the parameters do not allow direct analytical updating at the M-step. Nevertheless, the good news is that the Newton-Raphson iteration used to find new updates turns out to behave well, thanks to the good properties of the function of interest.

4.4.2 Estimating ϕ to obtain the mixing proportions

With the incorporation of fixed observed covariates into our model, we now have to obtain the mixing proportions through their corresponding parameters ϕ_j. As a function of ϕ, our expected log-likelihood function Q can be written as

$$Q(\phi) = \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\log(\pi_{ij}(\boldsymbol{u}_i,\phi_j))\right] = \sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)}\log(\pi_{ij}(\boldsymbol{u}_i,\phi_j)). \tag{4.17}$$

Recall that our aim at the M-Step is to find a new ϕ that maximises $Q(\phi)$ subject to

$$\sum_{j=1}^{k}\pi_{ij} = 1 \quad \text{and} \quad \sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)} = 1. \tag{4.18}$$

Estimation of ϕ for a 2-component mixture

We first restrict ourselves to a 2-component mixture in order to gain more insights into the estimation of ϕ. In fact, if we only have two components, then $\boldsymbol{y}$ has a Bernoulli distribution $\mathsf{Ber}(\pi)$, where $\pi = \pi(\phi,\boldsymbol{u})$ is a function of $\boldsymbol{u}$ and ϕ defined as follows:

$$\Pr(\boldsymbol{y}_i = 1|\boldsymbol{u}_i) = \pi_i = \frac{\exp(\boldsymbol{u}_i^{\mathsf{T}}\phi)}{1+\exp(\boldsymbol{u}_i^{\mathsf{T}}\phi)}. \tag{4.19}$$

From (4.19) and (4.18), our expected log-likelihood function Q in this binary case is now

$$Q(\phi) = \sum_{i=1}^{n}\mathsf{a}_i^{(t)}\log(\pi_i) + (1-\mathsf{a}_i^{(t)})\log(1-\pi_i). \tag{4.20}$$

It is easy to see that $Q(\phi)$ is a nonlinear function of ϕ. On the other hand, it is important to note that the form of $Q(\phi)$ does not allow the derivation of a closed-form expression for its maximiser. We use Newton-Raphson iteration to find the maximiser, which in this case is obtained by solving the equation $\dfrac{\partial Q}{\partial\phi} = 0$.

$$\frac{\partial Q}{\partial\pi_i} = \frac{\mathsf{a}_i^{(t)}-\pi_i}{\pi_i(1-\pi_i)} \quad \text{and} \quad \frac{\partial\pi_i}{\partial\phi} = \pi_i(1-\pi_i)\boldsymbol{u}_i \tag{4.21}$$

If we use the chain rule $\dfrac{\partial Q}{\partial\phi} = \dfrac{\partial Q}{\partial\pi_i}\dfrac{\partial\pi_i}{\partial\phi}$, it is straightforward to find that

$$\frac{\partial Q}{\partial\phi} = \sum_{i=1}^{n}(\mathsf{a}_i^{(t)}-\pi_i)\boldsymbol{u}_i = F(\phi). \tag{4.22}$$

The Jacobian matrix $J(\phi)$ in this case is given by

$$J(\phi) = \frac{\partial F}{\partial \phi} = -\sum_{i=1}^{n} \pi_i(1-\pi_i)\boldsymbol{u}_i\boldsymbol{u}_i^{\mathsf{T}}. \tag{4.23}$$

With F and J thus defined, the update $\phi^{(t+1)}$ of ϕ at iteration $t+1$ of the EM algorithm is obtained by the following Newton-Raphson iteration.

Algorithm 12: Newton-Raphson Iteration for updating $\phi^{(t)}$

Set $m := 1$ and $\phi^{new}(m) := \phi^{(t)}$, and choose Tol
Repeat
    $m := m + 1$;
    $\phi^{new}(m) := \phi^{new}(m-1) - J^{-1}(\phi^{new}(m-1))F(\phi^{new}(m-1))$;
Until $(\|\phi^{new}(m) - \phi^{new}(m-1)\| < \text{Tol})$ or $(m = m_{\max})$
$\phi^{(t+1)} := \phi^{new}(m)$

At each iteration of the EM algorithm, Algorithm 12 is applied to $\phi^{(t)}$. It is important to point out that, although simple in its formulation, the behaviour of Algorithm 12, in terms of convergence and stability, depends heavily on the accuracy of initial guesses and the existence of $J^{-1}(\phi)$.

Property 4.1 *According to a standard Newton-Raphson property,* Algorithm 12 *achieves local quadratic convergence if its initial values are accurate enough and* $J^{-1}(\phi_j)$ *exists.*

Proposition 4.1 *The Jacobian matrix* $J(\phi)$ *defined by* (4.23) *is negative definite.*

Proof: Since $\boldsymbol{u}_i\boldsymbol{u}_i^{\mathsf{T}}$ is a positive definite matrix, and the term $\pi_i(1-\pi_i)$ is a positive number, the sum $\sum_{i=1}^{n} \pi_i(1-\pi_i)\boldsymbol{u}_i\boldsymbol{u}_i^{\mathsf{T}}$ is therefore a positive definite matrix, and as a result, $J(\phi)$ is a negative definite matrix. □

Remark: Since $J(\phi)$ is negative definite, $J^{-1}(\phi)$ exists, and Algorithm 12 should therefore require very few iterations to yield the desired updates.

Estimation of ϕ for a k-component mixture

If we use the GLM formulation of Section (4.2), then we can rewrite $Q(\phi)$ as

$$Q(\phi) \propto \sum_{i=1}^{n} \left[[\mathsf{a}_i^{(t)}]^{\mathsf{T}}\boldsymbol{\eta}_i - \mathsf{b}(\boldsymbol{\eta}_i) \right], \tag{4.24}$$

where $\mathsf{a}_i^{(t)} = (\mathsf{a}_{i1}^{(t)}, \cdots, \mathsf{a}_{ik-1}^{(t)})^\mathsf{T}$ and $\mathsf{b}(\eta_i) = \log(1 + \sum_{j=1}^{k-1} \exp(\eta_{ij}))$, so that $\frac{\partial \mathsf{b}(\eta_i)}{\partial \eta_i} = \pi_i$. It is also easy to show that $\frac{\partial \pi_i}{\partial \eta_i} = \frac{\partial^2 \mathsf{b}(\eta_i)}{\partial \eta_i^2} = C_i(\phi)$, where $C_i(\phi)$ is as defined in (4.8). We use the chain rule $\frac{\partial \pi_i}{\partial \phi} = \frac{\partial \pi_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \phi} = \frac{\partial \pi_i}{\partial \eta_i} U_i = C_i(\phi) U_i$. From the above definition of Q in (4.24), and considering the fact that our logistic link function is a canonical link function, a well established result in GLM theory McCullagh and Nelder (1989), Fahrmeir and Tutz (1994) allows us to easily derive F and J as follows:

$$F(\phi) = \frac{\partial Q}{\partial \phi} = \sum_{i=1}^{n} U_i^\mathsf{T} \left[\mathsf{a}_i^{(t)} - \pi_i\right] \quad \text{and} \quad J(\phi) = \frac{\partial F}{\partial \phi} = -\sum_{i=1}^{n} U_i^\mathsf{T} C_i(\phi) U_i. \tag{4.25}$$

Proposition 4.2 *The Jacobian matrix $J(\phi)$ defined by (4.25) is negative definite.*

The proof of the above proposition is straightforward. Since $J(\phi)$ is negative, $J^{-1}(\phi)$ exists, and a conveniently extended version of Algorithm 12 should have quadratic local convergence to the update $\phi^{(t+1)}$.

4.4.3 Estimating the regression parameters Φ

As a function of Φ, Q can be written as

$$Q(\Phi) = \sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)} w_i^\mathsf{T} \Phi^\mathsf{T} \Psi^{-1} \mathsf{b}_{ij}^{(t)} - \frac{1}{2}\sum_{i=1}^{n} w_i^\mathsf{T} \Phi^\mathsf{T} \Psi^{-1} \Phi w_i.$$

The maximiser of $Q(\Phi)$ is given by

$$\Phi^{(t+1)} = \left[\sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)} \mathsf{b}_{ij}^{(t)} w_i^\mathsf{T}\right] \left[\sum_{i'=1}^{n} w_{i'} w_{i'}^\mathsf{T}\right]^{-1}.$$

The equations for updating the rest of parameters μ, Λ, Σ have the same form as the ones derived in Chapter 3 for the generic MFA model, although it must be noted that $\mathsf{a}_{ij}^{(t)}$, $\mathsf{b}_{ij}^{(t)}$ and $\mathbf{C}_{ij}^{(t)}$ are computed differently when covariates are used. The details of the derivation are provided in Sections B.2.3, B.2.4 and B.2.5 of Appendix (B).

$$
\begin{aligned}
\mu_j^{(t+1)} &= \left[\sum_{i=1}^{n} \mathsf{a}_{ij}^{(t)} \left(\boldsymbol{x}_i - \Lambda_j^{(t)} \mathbf{b}_{ij}^{(t)}\right)\right] \left[\sum_{i'=1}^{n} \mathsf{a}_{i'j}^{(t)}\right]^{-1} \\
\Lambda_j^{(t+1)} &= \left[\sum_{i=1}^{n} \mathsf{a}_{ij}^{(t)} (\boldsymbol{x}_i - \mu_j^{(t+1)}) \left(\mathbf{b}_{ij}^{(t)}\right)^{\mathsf{T}}\right] \left[\sum_{i'=1}^{n} \mathsf{a}_{i'j}^{(t)} \mathbf{C}_{i'j}^{(t)}\right]^{-1} \\
\Sigma^{(t+1)} &= \frac{1}{n} \mathsf{diag}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)} \left(\boldsymbol{x}_i - \mu_j^{(t+1)} - \Lambda_j^{(t+1)} \mathbf{b}_{ij}^{(t)}\right) \left(\boldsymbol{x}_i - \mu_j^{(t+1)}\right)^{\mathsf{T}}\right]
\end{aligned}
$$

4.4.4 Identifiability and other estimation difficulties

To gain insights into the extent of our identifiability problem, recall that we now have

$$[\boldsymbol{x}_i | \boldsymbol{y}_i = j] \sim \mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \Phi \boldsymbol{w}_i, \Lambda_j \Lambda_j^{\mathsf{T}} + \Sigma), \tag{4.26}$$

the corresponding marginal density of $\boldsymbol{x}_i$ being a mixture with the density

$$p(\boldsymbol{x}_i) = \sum_{j=1}^{k} \pi_{ij} \mathcal{N}_p(\boldsymbol{x}_i; \mu_j + \Lambda_j \Phi \boldsymbol{w}_i, \Lambda_j \Lambda_j^{\mathsf{T}} + \Sigma). \tag{4.27}$$

As we discussed extensively earlier, the generic MFA model itself already poses two main identifiability problems, one of which is brought about by the factor model, while the other is caused by the invariance of the mixture density to relabelling. Our approach has so far consisted and will once again consist of restricting the model to allow the determination of a unique set of parameters characterising our model. Besides the inherent lack of identifiability of the generic MFA model on which our extension is based, we have to contend here with new aspects of identifiability. As remarked by Titterington, Smith, and Makov (1985), it is difficult to give general rules for model identification, so that this difficult issue is always tackled according to the task at hand. Let us consider an unconstrained underlying local FA model, and a $q \times q$ orthogonal transformation Γ such that $\Gamma^{\mathsf{T}}\Gamma = \Gamma\Gamma^{\mathsf{T}} = \mathbf{I}_q$. Given our set $\boldsymbol{\theta} = \{\boldsymbol{\phi}, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \Sigma, \Phi\}$ of parameters, we apply the following transformations: $\tilde{\Phi} = \Gamma^{\mathsf{T}}\Phi$ and $\tilde{\Lambda}_j = \Lambda_j\Gamma$. It is easy to see that both the mean and the covariance matrix in (4.26) remain unchanged if we substitute Λ_j and Φ by $\tilde{\Lambda}_j$

and $\tilde{\Phi}$ respectively. The parameter set $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\phi}, \tilde{\Lambda}, \boldsymbol{\mu}, \Sigma, \tilde{\Phi}\}$ is therefore equivalent to $\boldsymbol{\theta}$, and we conclude that the model as defined is not identifiable. However, if we constrain each local factor analyser as we did in Section (2.2.3), $\tilde{\boldsymbol{\theta}} = \{\boldsymbol{\phi}, \tilde{\Lambda}, \boldsymbol{\mu}, \Sigma, \tilde{\Phi}\}$ will define an entirely new model, since transformations will lead to a violation of our restrictions on the structure with parameters not satisfying our constraints. The identifiability of our extended model is therefore achieved by the constraints imposed on the local factor analysers.

4.5 Application to synthetic tasks

Our examples in this chapter are all based on synthetic datasets. Since our fixed observed covariates are all assumed to be continuous variables, we generate datasets of covariates from multivariate Gaussians with some chosen mean and variance. Once the two sets of covariates are formed, the generation of $\boldsymbol{x}$ follows easily. As the derivation of our EM algorithm shows, the estimation equations for $\boldsymbol{\mu}$, Λ, Σ are very much the same as those obtained for the EM for the generic MFA model. On the other hand, the estimation equation for Φ is very straightforward. We shall therefore only concentrate on the estimates of $\boldsymbol{\phi}$, since the estimation is done via a new mechanism that we wish to explain and interpret.

4.5.1 Example 1

We first consider a relatively simple case where the underlying factor model has intrinsic dimensionality $q = 1$. For this toy problem, we choose $p = 3$, $r = 1$, and $s = 2$. Our true parameters are the following: $\boldsymbol{\phi}_1^{\mathsf{T}} = (-0.3, 0.9)$, $\boldsymbol{\phi}_2^{\mathsf{T}} = (0.60, -0.40)$, $\Phi = 2.7$ and $\Sigma = \mathsf{diag}(0.01, 0.05, 0.02)$. We use the above parameters to generate $n = 155$ observations from a $k = 3$-component mixture of factor analysers. We also generate the corresponding covariates for $\boldsymbol{z}$ and $\boldsymbol{y}$. A 3-D plot of the data is given below in Figure (4.1). In our artificial dataset, we have $n_1 = 69$, $n_2 = 52$ and $n_3 = 34$, which translates into the following mixing proportions: $\pi_1 = 0.44$, $\pi_2 = 0.35$ and $\pi_3 = 0.21$.

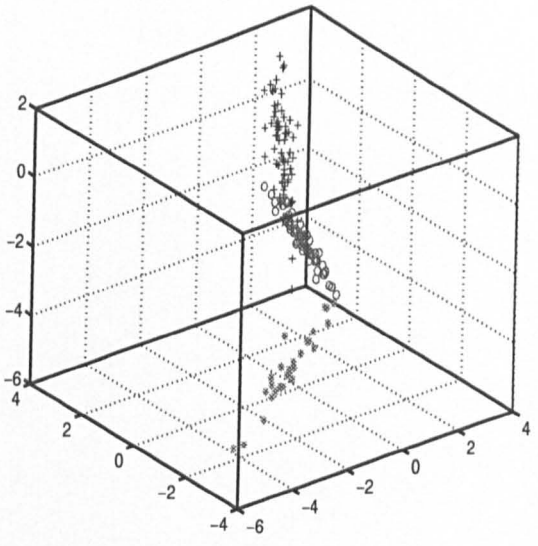


Figure 4.1: 3D plot of a 3-component MFA, with $q = 1$.

The good news. The application of our estimation scheme to this task yields encouraging results. It is particularly encouraging to point out that the Newton-Raphson iteration used to update the $\boldsymbol{\phi}_j$'s had quadratic local convergence. In fact, in many cases, fewer than 3 Newton-Raphson iterations are required to produce the update $\boldsymbol{\phi}^{(t+1)}$ at each EM iteration, up to a point where one could think of using a *one-step* Newton-Raphson updating instead of *full* Newton-Raphson described by Algorithm 12. For this particular task, we obtain $\hat{\boldsymbol{\phi}}_1^{\mathsf{T}} = (-0.42, 0.97)^{\mathsf{T}}$ and $\hat{\boldsymbol{\phi}}_2^{\mathsf{T}} = (0.74, -0.36)^{\mathsf{T}}$, which are very accurate. As far as the estimation of expected latent scores is concerned, it is also encouraging to note that the scheme achieves 100% correct clustering for the training data, with the above estimates of $\boldsymbol{\phi}$ allowing us to find very accurate estimates of the mixing proportions, namely $\hat{\pi}_1 = 0.44$, $\hat{\pi}_2 = 0.34$ and $\hat{\pi}_3 = 0.22$.

4.5.2 Example 2

Our second example is also a toy problem, with the only difference that we consider more components and more covariates on the component label than earlier. Here, $\boldsymbol{\phi}_1^{\mathsf{T}} = (-0.3, 0.5, 0.10)$, $\boldsymbol{\phi}_2^{\mathsf{T}} = (0.60, -0.40, -0.20)$ and $\boldsymbol{\phi}_3^{\mathsf{T}} = (-0.50, 0.40, -0.30)$. We use $s = 3$, and $k = 4$. Our mixing proportions in this case are $\pi_1 = 0.352$, $\pi_2 = 0.252$, $\pi_3 = 0.160$ and $\pi_4 = 0.236$, which correspond to $n_1 = 88$, $n_2 = 63$, $n_3 = 40$ and

$n_4 = 59$ for our sample of $n = 250$ observations. Again, the algorithm achieves good local quadratic convergence when initial values are close enough to the true values of interest. For instance, using either of $\boldsymbol{\phi}_1^{(0)} = (-0.40, 0.90, 0.00)^\mathsf{T}$, $\boldsymbol{\phi}_2^{(0)} = (0.60, 0.40, 0.00)^\mathsf{T}$, and $\boldsymbol{\phi}_3^{(0)} = (0.00, 0.00, 0.00)^\mathsf{T}$, or $\boldsymbol{\phi}_1^{(0)} = (0.00, 0.00, 0.00)^\mathsf{T}$, $\boldsymbol{\phi}_2^{(0)} = (0.00, 0.00, 0.00)^\mathsf{T}$, and $\boldsymbol{\phi}_3^{(0)} = (0.00, 0.00, 0.00)^\mathsf{T}$ as initial estimates yields, $\hat{\boldsymbol{\phi}}_1^\mathsf{T} = (-0.21, 0.41, 0.11)^\mathsf{T}$, $\hat{\boldsymbol{\phi}}_2^\mathsf{T} = (0.65, -0.56, -0.08)^\mathsf{T}$ and $\hat{\boldsymbol{\phi}}_3^\mathsf{T} = (-0.30, 0.05, -0.14)^\mathsf{T}$, corresponding to $\hat{\pi}_1 = 0.345$, $\hat{\pi}_2 = 0.258$, $\hat{\pi}_3 = 0.160$ and $\hat{\pi}_3 = 0.236$, all of which are very good estimates.

The bad news. However, the scheme still suffers from the weaknesses of both the EM and the Newton-Raphson algorithms, namely

- **Dependence on initial conditions.** Although we obtain relatively accurate estimates for this example using two different sets of initial values, we also experience total lack of convergence with some sets of initial values, especially those not close enough to the neighbourhood of the true values of interest. For instance, $\boldsymbol{\phi}_1^{(0)} = (-2.00, 0.00, 0.00)^\mathsf{T}$, $\boldsymbol{\phi}_2^{(0)} = (0.00, -1.00, 0.00)^\mathsf{T}$, and $\boldsymbol{\phi}_3^{(0)} = (0.00, 0.00, -1.00)^\mathsf{T}$ fails to produce any meaningful set of estimates. This is typical of Newton-Raphson iteration, because of its essentially local behaviour.

- **Inability to escape local maxima.** Once a fixed point is found, the scheme tends to remain there, no matter how long we iterate. In the event that the fixed point is far from the true value sought, the algorithm fails and yields very inaccurate estimates.

4.6 Outline of a Bayesian treatment

A natural alternative to the EM algorithm that we have just studied is the Bayesian treatment of the model. In our analysis of the generic MFA model, we found that the model allowed the use of conjugate priors, and we used Bayesian sampling on the complete-data posterior to perform estimation and inference. If we consider our set of parameters $\boldsymbol{\theta}$ and the form of the likelihood for the extended MFA model, it is easy to see that we can still use the same priors for $\boldsymbol{\mu}$, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}$. As far as the two newcomers

ϕ and Φ are concerned, a Gaussian prior on the columns or rows of Φ should lead to a full conditional posterior that is also Gaussian. The only parameter that could demand extra concentration of effort in this case is ϕ. In fact, a good candidate prior for each ϕ_j is a Gaussian prior. Let us consider deriving the corresponding full conditional posterior

$$p(\phi_j|\cdots) \propto \left[\prod_{i=1}^{n} [p(x_i|y_i, z_i)\mathbf{Pr}(y_i = j|u_i)]^{y_{ij}}\right] p(\phi_j). \tag{4.28}$$

In (4.28), $p(\phi_j)$ is Gaussian, and $p(x_i|y_i, z_i)$ is also Gaussian, but the logistic distribution function $\mathbf{Pr}(y_i = j|u_i)$ is non-Gaussian, so that the derivation of $p(\phi_j|\cdots)$ is not straightforward. One of the classical solutions to this problem is the use of approximations, namely the Laplace approximation. This Laplace approximation consists of approximating the logistic function by a Gaussian, which then allows the derivation of an approximate Gaussian full conditional posterior $p(\phi_j|\cdots)$. We will be exploring this Bayesian treatment in our future work.

4.7 Conclusion and discussion

In this chapter, we have studied an extension of the MFA model motivated by the possibility that latent variables could be affected by fixed observed covariates. The EM algorithm for this extended model is found to perform well, despite the need for approximate Newton-Raphson updates. Despite some of the weaknesses of the EM algorithm and the Newton-Raphson iterations, the scheme allows us to obtain reasonably accurate parameter estimates. Last but not least, it is worth mentioning that the Newton-Raphson iteration provides an extra advantage which is an estimate of the variance-covariance matrix of the maximum likelihood estimate.

While it is possible to extend the covariate mechanism on z by allowing a different Φ_j for each component, it must be noted that such an extension could run into greater identifiability problems, partly because of the invariance to permutations of labels.

We have so far tested our inference and estimation algorithm only on artificial tasks, but we would like to use it on real life applications. In our future investigations, we plan to address identifiability by implementing a constrained version of the EM algorithm.

Chapter 5

MFA models with mixed outcomes

An expert is someone who knows some of the worst mistakes, which can be made, in a very narrow field.
Niels Bohr

In all our analyses so far, we have treated the manifest variable $\boldsymbol{x}$ in the pure spirit of traditional factor analysis which assumes $\boldsymbol{x}$ to be a vector of continuous attributes. While there are many practical applications for which this is the case, fields such as social science, psychology and psychometrics are full of applications where the manifest variable is made up of attributes of various different types (continuous, categorical, counts). Many authors have studied various models allowing the observed quantities to be a mix of continuous and non-continuous random variables. Sammel, Ryan, and Legler (1997) for instance have studied *latent variable models for mixed discrete and continuous outcomes*, and have used their scheme on medical applications. Along the same lines, Shi and Lee (2000) have explored the analysis of *latent variable models with mixed continuous and polytomous data*, with applications to a variety of problems in psychology. Finally, in their study of *finite mixture modelling with mixture outcomes using the EM Algorithm*, Muthén and Shedden (1999) touched on some extensions of latent variable models that allow the manifest variable to be made up of attributes of different types. Extending our MFA model so as to allow it to handle such applications is therefore fully justified.

In this chapter, we introduce and study such an extension of the MFA model, and

in particular we examine such issues as parameter estimation and prediction from a likelihood-based perspective via the EM algorithm. Thanks to the axiom of conditional independence that we introduced in Chapter 1, we are able to treat each manifest attribute (variable) separately, and this allows the whole extended model to be treated as a collection of Generalised Linear Models (GLM). Although the resulting model does not allow the derivation of closed-form expressions for both the E-step and the M-step of the corresponding EM algorithm, it turns out that relatively simple Monte Carlo approximations make it possible to compute parameter estimates efficiently. This chapter is an extension of Fokoué and Titterington (2000b).

5.1 Introducing Mixed Outcomes

As we saw in Chapter 1, one of the pillars of latent variable modelling is the axiom of conditional independence introduced and explained in Section (1.6). Intuitively, this means that, under the factor analysis assumptions, the variables that constitute the observed vector become independent once the common factors are known, since these common factors account for the inter-dependence among the observations. In other words, in the particular case of the MFA model, the conditional distribution of the p-dimensional observed vector $\boldsymbol{x}^\top = (x_1, \cdots, x_p)$ given both the common factors and the component labels is the product (5.1) of the conditional distributions of each of its individual attributes:

$$p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}) = \prod_{h=1}^{p} p(x_h|\boldsymbol{y}, \boldsymbol{z}). \tag{5.1}$$

Thanks to this conditional independence, the conditional distribution of each outcome (attribute of the manifest vector) can therefore be modelled separately.

5.1.1 Model for a single outcome

Before we embark on the analysis of mixed outcomes, it makes sense to go back to the generic MFA model and see how we can model the conditional distribution $p(x_h|\boldsymbol{y}, \boldsymbol{z})$

of each $x_h, h = 1, \cdots, p$, separately under the normality assumptions of the traditional factor analysis model. In the context of equation (3.9), we will end up with p equations of classical linear models for normal responses, namely

$$x_h = \Lambda_{jh.}^{\mathsf{T}} \boldsymbol{z} + \mu_{jh} + e_h = \boldsymbol{z}^{\mathsf{T}} \Lambda_{jh.} + \mu_{jh} + e_h, \tag{5.2}$$

where $e_h \sim \mathcal{N}(0, \sigma_h^2)$, since the disturbance vector $\boldsymbol{e}$ is distributed as $\mathcal{N}_p(0, \Sigma)$ with $\Sigma = \mathsf{diag}(\sigma_1^2, \cdots, \sigma_p^2)$. In this simple and essentially illustrative case, all the outcomes have the same conditional distribution, namely the normal distribution. In the next section, we examine a generalisation of this simple case, by allowing the conditional distribution of each x_h to be different.

Note: Given a sample $\mathbf{X} = \{\boldsymbol{x}_i, \cdots, \boldsymbol{x}_n\}$ of i.i.d observations, we shall first concentrate on the specification of the distributional aspects of x_{ih} which is the h-th outcome of the i-th observation $\boldsymbol{x}_i$, with $i = 1, \cdots, n$ and $h = 1, \cdots, p$. For economy of notation, we shall from now on simply write $\Pr(x_{ih}|\cdots)$ or $\mathbb{E}[x_{ih}|\cdots]$ instead of the full $\Pr(x_{ih}|\boldsymbol{y}_i = j, \boldsymbol{z}_i, \boldsymbol{\theta})$ or $\mathbb{E}[x_{ih}|\boldsymbol{y}_i = j, \boldsymbol{z}_i, \boldsymbol{\theta}]$ respectively.

5.1.2 Generalised Linear Model formulation

In order to fully specify the conditional distribution of the single outcome x_h, we reformulate the conditional model for x_h as a Generalised Linear Model McCullagh and Nelder (1989),Fahrmeir and Tutz (1994). We first consider the simple case of a normally distributed x_h, which in a sense is equivalent to simply extracting each attribute of the manifest variable from the factor analysis model as we did in the previous section.

1. **The Random Component:** this is represented here by the random disturbance term e_h, and comes from our distributional assumption about the model. For some types of outcome (categorical for example), we may not be able to write an equation for the outcome as in (5.2), and specifying the random component of the GLM would therefore simply mean indicating the assumed probability distribution of the outcome, and defining the corresponding mean $\mathsf{m}_{jh} = \mathbb{E}[x_h|\cdots]$.

2. **The Systematic Component:** this is represented here by the linear expression in the parameters $\eta_{jh} = z^{\mathsf{T}}\Lambda_{jh.} + \mu_{jh}$. If we define $\tilde{z}^{\mathsf{T}} = (z, 1)$ and $\beta_{jh}^{\mathsf{T}} = (\Lambda_{jh.}^{\mathsf{T}}, \mu_{jh})$ then we can write $\eta_{jh} = \tilde{z}^{\mathsf{T}}\beta_{jh}$, where $\tilde{z}$ and β_{jh} are both $(q+1)$-dimensional column vectors. This is a result of our structural assumption about the model.

3. **The Link between systematic and random components**
   $\mathsf{m}_{jh} = f(\eta_{jh})$ where $f(\cdot)$ is referred to as the *response function.* We can also write $\eta_{jh} = g(\mathsf{m}_{jh})$ where $g(\cdot)$ is referred to as the *link function.*

We further assume that the density of each outcome x_{ih} can be expressed as a regular exponential family density with canonical parameterisation as follows:

$$p(x_{ih}|\cdots) = \exp\left[(x_{ih}\eta_{jh} - \mathsf{b}(\eta_{jh}))/\varphi_h + \mathsf{c}(x_{ih}, \varphi_h)\right]. \tag{5.3}$$

In (5.3), $\mathsf{b}(\cdot)$ and $\mathsf{c}(\cdot)$ are specific functions defining the type of exponential family under consideration, φ_h is an additional scale or dispersion parameter, and η_{jh} is referred to as the *natural parameter.* The canonical parameterisation of (5.3) offers the great advantage that it is a general formulation, and can therefore be used for the analysis of different types of outcomes by simply specifying φ_h, $\mathsf{b}(\eta_{jh})$ and $\mathsf{c}_h(x_{ih}, \varphi_h)$ for the outcome of interest. In many of our subsequent developments, we will need expressions for the mean $\mathsf{m}_{jh} = \mathbb{E}[x_{ih}|\cdots]$ and the variance $\mathsf{V}_{jh} = \mathbb{V}[x_{ih}|\cdots]$ of x_{ih}. In fact, we have

$$\mathsf{m}_{jh} = \mathsf{b}'(\eta_{jh}) = \frac{\partial \mathsf{b}(\eta_{jh})}{\partial \eta_{jh}} \quad \text{and} \quad \mathsf{V}_{jh} = \mathsf{b}''(\eta_{jh})) = \frac{\partial^2 \mathsf{b}(\eta_{jh})}{\partial \eta_{jh}^2}. \tag{5.4}$$

For all the types of outcomes that we shall consider, Fahrmeir and Tutz (1994) provides a table that offers all the above ingredients, namely $\varphi_h, \mathsf{b}(\eta_{jh}), \mathsf{c}(\eta_{jh}), \mathsf{m}_{jh}, \mathsf{V}_{jh}$.

5.2 Exploring different types of outcomes

Our introduction to GLM in Section (5.1.2) gives an example of continuous outcome with Gaussian random noise. We now present a more detailed description of the GLMs for the different types of outcome that we intend to analyse.

- **Categorical outcome:** If x_h is categorial, then a good candidate for modelling it is the logistic regression model. For a simple binary case, we have $x_h \in \{0, 1\}$, and x_h follows a Bernoulli distribution with

$$\mathsf{m}_{jh} = \mathbb{E}[x_{ih}|\cdots] = \Pr(x_{ih} = 1|\cdots) = \frac{\exp(\eta_{jh})}{1 + \exp(\eta_{jh})}, \tag{5.5}$$

so that our response function is the *logistic* function $f(\cdot) = \dfrac{\exp(\cdot)}{1 + \exp(\cdot)}$. This can also be expressed in terms of the *logit* link function as $\eta_{jh} = \log\left[\dfrac{\mathsf{m}_{jh}}{1 - \mathsf{m}_{jh}}\right]$, which defines the log odds. For a Bernoulli distributed outcome, $\varphi_h = 1$, and $\mathsf{b}(\eta_{jh}) = \log(1 + \exp(\eta_{jh}))$, so that $\mathsf{V}_{jh} = \mathsf{m}_{jh}(1 - \mathsf{m}_{jh})$.

- **Gaussian outcome:** As we saw earlier, the Gaussian outcome is the most natural of all. In fact, in the Gaussian case, both the link function and the response function are simply *identity* functions, and we have

$$\mathsf{m}_{jh} = \mathbb{E}[x_{ih}|\cdots] = \eta_{jh} \quad \text{and} \quad p(x_{ih}|\cdots) = \mathcal{N}(x_{ih}; \eta_{jh}, \sigma_h^2). \tag{5.6}$$

For a normally distributed outcome, $\varphi_h = \sigma_h^2$, and $\mathsf{b}(\eta_{jh}) = \eta_{jh}^2/2$.

- **Poisson outcome:** For an x_h following a Poisson distribution, we need to make sure that $\mathsf{m}_{jh} > 0$ since this m_{jh} is both the mean and the variance of the Poisson distributed random outcome. A good candidate for the response function is obviously the exponential function, and this leads us to the log-linear Poisson model. We can therefore write $\eta_{jh} = \log \mathsf{m}_{jh}$ which is equivalent to

$$\mathsf{m}_{jh} = \mathbb{E}[x_{ih}|\cdots] = \exp(\eta_{jh}) \quad \text{and} \quad \Pr(x_{ih} = a|\cdots) = \frac{\mathsf{m}_{jh}^a \mathrm{e}^{-\mathsf{m}_{jh}}}{a!}. \tag{5.7}$$

For a Poisson distributed outcome, $\varphi_h = 1$, and $\mathsf{b}(\eta_{jh}) = \exp(\eta_{jh})$.

5.3 Elements of Estimation and Inference

Among the variety of interesting issues that could be addressed for this extended model, the estimation of latent scores (categorical and continuous) occupies a central place.

However, the estimation of the parameters that characterise the relationship between each outcome and the latent variables remains the most important issue. We define $\beta_j = (\beta_{j1}, \cdots, \beta_{jp})$, $\beta = (\beta_1, \cdots, \beta_k)$ and $\varphi = (\varphi_1, \cdots, \varphi_p)$, so that the complete collection of all the parameters of the model is $\theta = \{\beta, \varphi, \pi\}$. In this chapter, we examine the maximum likelihood estimation of parameters via the EM algorithm. As we shall see in the next section, the EM algorithm in this case is not as straightforward as the one constructed in Chapter 3, but the extra computational effort required is not alarming. If we simply consider the extension of the generic MFA model, then our complete-data likelihood is

$$L(\theta; \mathbf{X}^*) = \prod_{i=1}^{n} \left[\prod_{j=1}^{k} \left[\pi_j^{y_{ij}} \left(\prod_{h=1}^{p} p(x_{ih}|z_i, y_{ij}) \right)^{y_{ij}} \right] \right]. \tag{5.8}$$

If we incorporate the effects of covariates, then the complete-data likelihood becomes

$$L(\theta; \mathbf{X}^*) = \prod_{i=1}^{n} \left[\prod_{j=1}^{k} \left[\left(\prod_{h=1}^{p} p(x_{ih}|z_i, y_{ij}) \right) \Pr(y_{ij} = 1|u_i) \right]^{y_{ij}} p(z_i|w_i) \right]. \tag{5.9}$$

For simplicity, we ignore the effects of fixed covariates in this analysis of mixed outcomes. Thus, we shall concentrate on the complete-data likelihood of equation (5.8) throughout this chapter. The analysis of an extension that allows covariates follows from this one. The complete-data loglikelihood corresponding to (5.8) is given by

$$\ell(\theta, \mathbf{X}^*) = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{h=1}^{p} y_{ij} \log p(x_{ih}|y_{ij}, z_i) + \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log \pi_j. \tag{5.10}$$

If we use the GLM general formulation of equation (5.3), $\ell(\theta, \mathbf{X}^*)$ becomes

$$\ell(\theta, \mathbf{X}^*) = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{h=1}^{p} y_{ij} \left[(x_{ih}\eta_{jh} - \mathsf{b}(\eta_{jh}))/\varphi_h + \mathsf{c}(x_{ih}, \varphi_h) \right] + \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log \pi_j. \tag{5.11}$$

We define the scores of the i-th observation and the entire sample respectively as

$$S_i(\theta) = \frac{\partial}{\partial \theta} \ell(\theta, x_i^*) \quad \text{and} \quad S(\theta) = \frac{\partial}{\partial \theta} \ell(\theta, \mathbf{X}^*) = \sum_{i=1}^{n} S_i(\theta). \tag{5.12}$$

5.4 An EM Algorithm for the model

Our main ingredient for the derivation of the EM algorithm is $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$, the expectation (5.13) of $\ell(\boldsymbol{\theta}; \mathbf{X}^*)$ with respect to the conditional distribution $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ of $\boldsymbol{y}$ and $\boldsymbol{z}$ given the observed data $\mathbf{X}$ and the current set of parameter estimates $\boldsymbol{\theta}^{(t)}$.

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}\left[\ell(\boldsymbol{\theta}; \mathbf{X}^*)|\mathbf{X}, \boldsymbol{\theta}^{(t)}\right] = \int\int_{\mathcal{H}} \ell(\boldsymbol{\theta}; \mathbf{X}^*)\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})d\boldsymbol{y}d\boldsymbol{z}. \tag{5.13}$$

In our previous analyses, the manifest variable $\boldsymbol{x}$ had a Gaussian distribution, and this made the computation of the conditional expectations $\mathbb{E}[y_{ij}|\boldsymbol{x}_i]$, $\mathbb{E}[\boldsymbol{z}_i y_{ij}|\boldsymbol{x}_i]$ and $\mathbb{E}\left[\boldsymbol{z}_i \boldsymbol{z}_i^\top y_{ij}|\boldsymbol{x}_i\right]$ straightforward, allowing us to have a closed-form expression for $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ by direct manipulation of expectations (see Chapters 2, 3, and 4). In this new extended model, the manifest variable $\boldsymbol{x}$ is no longer Gaussian. Moreover, the situation is made even more complicated by the "mixed" nature of $\boldsymbol{x}$ that does not allow a closed-form expression for $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$. As a result, all expectations with respect to $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ involve integrals that are intractable. We therefore need to "efficiently" compute (hopefully) accurate approximations to the expectations of interest.

5.4.1 Notations and remarks

All our expectations are taken with respect to $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$. We therefore simply write $\mathbb{E}[g(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})]$ instead of $\mathbb{E}\left[g(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}\right]$, so that we have

$$\mathbb{E}[g(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})] = \int\int_{\mathcal{H}} g(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})d\boldsymbol{y}d\boldsymbol{z}. \tag{5.14}$$

Under the regularity conditions that allow the interchange of differentiation and integration, we have

$$\frac{\partial}{\partial\boldsymbol{\theta}}\mathbb{E}[\ell(\boldsymbol{\theta}, \mathbf{X}^*)] = \mathbb{E}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\ell(\boldsymbol{\theta}, \mathbf{X}^*)\right] = \mathbb{E}[S(\boldsymbol{\theta})]. \tag{5.15}$$

At the M-step, our aim is to solve $\frac{\partial}{\partial\boldsymbol{\theta}}Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \frac{\partial}{\partial\boldsymbol{\theta}}\mathbb{E}[\ell(\boldsymbol{\theta}, \mathbf{X}^*)] = 0$. As we shall see later, it will turn out to be more convenient (mathematically) in many cases to use the above regularity conditions and instead solve the expected score equation $\mathbb{E}[S(\boldsymbol{\theta})] = 0$.

5.4.2 Approximating intractable expectations

As we remarked earlier, the integrals needed for the computation of our expectations are all high-dimensional non-Gaussian integrals for which closed-form expressions cannot be obtained. The two main approximation methods that are most commonly used to tackle this intractability are: (a) the Monte Carlo approximation and (b) the Gauss-Hermite quadrature approximation. Sammel, Ryan, and Legler (1997) examined both approximations in their work, and found the Gauss-Hermite quadrature to be faster in reaching convergence. Our main focus in this chapter is on the Monte Carlo approximation. We give an outline of the Gauss-Hermite quadrature approximation at the end of the chapter.

5.4.3 Monte Carlo E-Step

If we could sample directly from $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$, then we would simply draw a sample $(\boldsymbol{y}_1, \boldsymbol{z}_1), \cdots, (\boldsymbol{y}_D, \boldsymbol{z}_D)$, and a straightforward Monte Carlo approximation to $\mathbb{E}\left[g(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})\right]$ would be given by

$$\widehat{\mathbb{E}\left[g(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{z})\right]} = \frac{1}{D}\sum_{d=1}^{D} g(\boldsymbol{x}; \boldsymbol{y}_d, \boldsymbol{z}_d). \tag{5.16}$$

However, since we do not have a closed-form expression for $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ in our context, such a direct sampling is not possible. To solve the problem, we write $\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)})$ as

$$\boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) = \frac{\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})}{\int\int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})d\boldsymbol{y}d\boldsymbol{z}}. \tag{5.17}$$

Since $\boldsymbol{y}$ is a discrete categorical random variable with $\mathbf{Pr}(\boldsymbol{y} = j|\boldsymbol{\theta}^{(t)}) = \pi_j^{(t)}$, we can write

$$\boldsymbol{p}(\boldsymbol{y} = j, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}^{(t)}) = \frac{\pi_j^{(t)}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j, \boldsymbol{z}, \boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})}{\sum_{j'=1}^{k}\int_{\mathcal{Z}} \pi_{j'}^{(t)}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j', \boldsymbol{z}, \boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})d\boldsymbol{z}}. \tag{5.18}$$

Equations (5.17) and (5.18) allow us to rewrite $\mathbb{E}[g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})]$ as

$$\begin{aligned}\mathbb{E}[g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})] &= \frac{\int\int_{\mathcal{H}} g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z},\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})d\boldsymbol{y}d\boldsymbol{z}}{\int\int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z},\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})d\boldsymbol{y}d\boldsymbol{z}} \\ &= \frac{\sum_{j=1}^{k}\int_{\mathcal{Z}} \pi_j^{(t)} g(\boldsymbol{x};\boldsymbol{y}=j,\boldsymbol{z})\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}=j,\boldsymbol{z},\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})d\boldsymbol{z}}{\sum_{j'=1}^{k}\int_{\mathcal{Z}} \pi_{j'}^{(t)}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}=j',\boldsymbol{z},\boldsymbol{\theta}^{(t)})\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})d\boldsymbol{z}} \qquad (5.19)\end{aligned}$$

for which the Monte Carlo approximation is given by

$$\mathbb{E}[g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})] \approx \frac{\sum_{d=1}^{D} g(\boldsymbol{x};\boldsymbol{y}_d,\boldsymbol{z}_d)\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d,\boldsymbol{z}_d,\boldsymbol{\theta}^{(t)})}{\sum_{d=1}^{D}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d,\boldsymbol{z}_d,\boldsymbol{\theta}^{(t)})}, \qquad (5.20)$$

and where the samples $(\boldsymbol{y}_d,\boldsymbol{z}_d)$ with $d = 1,\cdots,D$ are samples drawn from $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})$ and $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})$ respectively. $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\theta}^{(t)})$ is the multinomial distribution $\boldsymbol{y} \sim \mathsf{Mn}(k;\boldsymbol{\pi})$ and $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)})$ is the standard multivariate Gaussian distribution, both of which are distributions that can be simulated easily. It is also possible to avoid sampling $\boldsymbol{y}$, in which case the Monte Carlo approximation is given by

$$\mathbb{E}[g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})] \approx \frac{\sum_{j=1}^{k}\sum_{d=1}^{D} g(\boldsymbol{x};\boldsymbol{y}=j,\boldsymbol{z}_d)\pi_j^{(t)}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}=j,\boldsymbol{z}_d,\boldsymbol{\theta}^{(t)})}{\sum_{j'=1}^{k}\sum_{d=1}^{D}\pi_{j'}^{(t)}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}=j',\boldsymbol{z}_d,\boldsymbol{\theta}^{(t)})}. \qquad (5.21)$$

Remark: The obvious advantage of (5.21) over (5.20) is that with (5.21) we avoid the sampling of $\boldsymbol{y}$. However, it must be said that this is done at the expense of an extra loop on k which could become computationally burdensome for large k and large D.

Note : One of the main drawbacks of this Monte Carlo approach is its slowness to converge. In fact, to get accurate approximations, large samples are required, and this is very time consuming in multivariate settings such as ours. Sammel, Ryan, and Legler (1997) have reported $T = 10000$ as the number of iterations needed to get accurate estimates.

In pseudocode form, computing (5.20) can be described as follows:

Algorithm 13: Computing expectation $\mathbb{E}[g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})]$

Set $\mathbf{N} := 0$ and $\mathbf{D} := 0$;
For $d := 1$ to D
    Simulate $\boldsymbol{y}_d \sim \mathsf{Mn}(k, \boldsymbol{\pi}^{(t)})$; /* Choose component */
    Simulate $\boldsymbol{z}_d \sim \mathcal{N}_q(0, \mathbf{I}_q)$; /* Draw a common factor */
    Compute $g(\boldsymbol{x}; \boldsymbol{y}_d, \boldsymbol{z}_d)$;
    Compute $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d, \boldsymbol{z}_d)$;
    Set $\mathbf{N} := \mathbf{N} + g(\boldsymbol{x}; \boldsymbol{y}_d, \boldsymbol{z}_d)\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d, \boldsymbol{z}_d)$;
    Set $\mathbf{D} := \mathbf{D} + \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d, \boldsymbol{z}_d)$;
End;
Set $\mathbb{E}[g(\boldsymbol{x};\boldsymbol{y},\boldsymbol{z})] \approx \mathbf{N}/\mathbf{D}$;

5.4.4 Constructing the Maximisation step

As we said earlier, our M-step consists of solving the expected score equation

$$\mathbb{E}[S(\boldsymbol{\theta})] = \sum_{i=1}^{n} \mathbb{E}[S_i(\boldsymbol{\theta})] = 0, \tag{5.22}$$

where

$$\mathbb{E}[S_i(\boldsymbol{\theta})] = \int\int_{\mathcal{H}} S_i(\boldsymbol{\theta}) \boldsymbol{p}(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}) d\boldsymbol{y} d\boldsymbol{z}. \tag{5.23}$$

Clearly, our complete-data loglikelihood function (5.11) has two main parts: (a) one that contains the conditional distribution of the manifest variables, and the other (b) that contains elements of the distribution of the categorical latent variable $\boldsymbol{y}$. For the second part (b) of $\boldsymbol{\ell}(\boldsymbol{\theta}, \mathbf{X}^*)$, the expression of the update $\pi_j^{(t+1)}$ of $\pi_j^{(t)}$ can be easily written. However, because of the nonlinearity of some of the terms involved, we will resort to the Fisher scoring method to find updates $\boldsymbol{\theta}^{(t+1)}$ of $\boldsymbol{\theta}^{(t)}$ for part (a) of the loglikelihood.

5.4.5 Updating the mixing proportions

To determine the updates $\pi_j^{(t+1)}$ of the mixing proportions at the M-step, we consider

$$\boldsymbol{\ell}(\boldsymbol{\pi}, \mathbf{X}^*) = \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log \pi_j. \tag{5.24}$$

If we use the Lagrange multiplier method to maximise $\ell(\boldsymbol{\pi}, \mathbf{X}^*)$ subject to the constraint $\sum_{j=1}^{k} \pi_j = 1$, then it is easy to see that

$$\pi_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[y_j | \boldsymbol{x}_i\right]. \tag{5.25}$$

Recall that y_j is a indicator variable. Therefore y_j follows a Bernoulli distribution with parameter $\pi_j^{(t)}$. Hence $\mathbb{E}\left[y_j|\boldsymbol{x}_i\right] = \mathbf{Pr}(\boldsymbol{y} = j|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}) = \int_{\mathcal{Z}} \boldsymbol{p}(\boldsymbol{y} = j, \boldsymbol{z}|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)})d\boldsymbol{z}$, allowing us to write

$$\mathbb{E}\left[y_j|\boldsymbol{x}_i\right] = \frac{\int_{\mathcal{Z}} \pi_j^{(t)} \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j, \boldsymbol{z}, \boldsymbol{\theta}^{(t)}) \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)}) d\boldsymbol{z}}{\sum_{j'=1}^{k} \int_{\mathcal{Z}} \pi_{j'}^{(t)} \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j', \boldsymbol{z}, \boldsymbol{\theta}^{(t)}) \boldsymbol{p}(\boldsymbol{z}|\boldsymbol{\theta}^{(t)}) d\boldsymbol{z}}. \tag{5.26}$$

The Monte Carlo approximation to $\mathbb{E}\left[y_j|\boldsymbol{x}_i\right]$ in this case is therefore given by

$$D_{ij} = \frac{\sum_{d=1}^{D} \pi_j^{(t)} \boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{y} = j, \boldsymbol{z}_d, \boldsymbol{\theta}^{(t)})}{\sum_{j'=1}^{k} \sum_{d=1}^{D} \pi_{j'}^{(t)} \boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{y} = j', \boldsymbol{z}_d, \boldsymbol{\theta}^{(t)})}. \tag{5.27}$$

If we choose to also sample $\boldsymbol{y}$, then an alternative to (5.27) would be

$$D_{ij} = \frac{\sum_{d=1, y_d=j}^{D} \boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{y}_d, \boldsymbol{z}_d, \boldsymbol{\theta}^{(t)})}{\sum_{d=1}^{D} \boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{y}_d, \boldsymbol{z}_d, \boldsymbol{\theta}^{(t)})}.$$

An update of $\pi_j^{(t+1)}$ via Monte Carlo approximation would therefore be

$$\pi_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} D_{ij}. \tag{5.28}$$

5.4.6 Updating the GLM parameters

Because of the nonlinearity of the functionals of interest, we will use a Newton-Raphson type scheme to update our GLM parameters at each step. In general, we need to estimate the parameter $\boldsymbol{\beta}_{jh}$ that characterises the effect of the h-th outcome in the j-th component

of the mixture. If we focus on the part of the loglikelihood function that involves the GLM model, it is easy to show that a general expression for the expected score is

$$G(\beta_{jh}) = \mathbb{E}\left[S(\beta_{jh})\right] = \mathbb{E}\left[\frac{\partial}{\partial\beta_{jh}}\ell(\beta_{jh}, \mathbf{X}^*)\right] = \frac{1}{\varphi_h}\sum_{i=1}^{n}\mathbb{E}\left[y_{ij}\left[x_{ih} - \mathsf{m}_{jh}(\beta_{jh})\right]\tilde{\boldsymbol{z}}_i\right]$$
$$= \frac{1}{\varphi_h}\sum_{i=1}^{n}\mathbb{E}\left[y_{ij}|\boldsymbol{x}_i\right]\mathbb{E}\left[[x_{ih} - \mathsf{m}_{jh}(\beta_{jh})]\tilde{\boldsymbol{z}}_i|y_{ij}, \boldsymbol{x}_i\right] \tag{5.29}$$

Solving the expected score equation, $\mathbb{E}\left[S(\beta_{jh})\right] = 0$, then requires the computation of

$$J(\beta_{jh}) = \frac{\partial}{\partial\beta_{jh}}\mathbb{E}\left[S(\beta_{jh})\right] = \mathbb{E}\left[\frac{\partial^2\ell(\beta_{jh}, \mathbf{X}^*)}{\partial\beta_{jh}\partial\beta_{jh}^{\mathsf{T}}}\right] = -\frac{1}{\varphi_h}\sum_{i=1}^{n}\mathbb{E}\left[y_{ij}\mathsf{V}_{jh}(\beta_{jh})\tilde{\boldsymbol{z}}_i\tilde{\boldsymbol{z}}_i^{\mathsf{T}}\right]$$
$$= -\frac{1}{\varphi_h}\sum_{i=1}^{n}\mathbb{E}\left[y_{ij}|\boldsymbol{x}_i\right]\mathbb{E}\left[\mathsf{V}_{jh}(\beta_{jh})\tilde{\boldsymbol{z}}_i\tilde{\boldsymbol{z}}_i^{\mathsf{T}}|\boldsymbol{x}_i, y_{ij}\right] \tag{5.30}$$

With (5.29) and (5.30), the update of β_{jh} is given by the following Newton-Raphson scheme:

$$\beta_{jh}^{(t+1)} = \beta_{jh}^{(t)} - J^{-1}(\beta_{jh}^{(t)})G(\beta_{jh}^{(t)}). \tag{5.31}$$

It turns out that $J(\beta_{jh}) = -F(\beta_{jh})$ where $F(\beta_{jh})$ is the expected Fisher information matrix. For this reason, (5.31) can instead be written as

$$\beta_{jh}^{(t+1)} = \beta_{jh}^{(t)} + F^{-1}(\beta_{jh}^{(t)})G(\beta_{jh}^{(t)}). \tag{5.32}$$

Note: It is interesting to note that the above framework for updating the GLM parameters is very general, and therefore makes it easy to compute updates for any type of outcome.

Algorithm 14: Newton-Raphson Iteration for updating $\beta_{jh}^{(t)}$

Set $m := 1$ and $\beta_{jh}^{new}(m) := \beta_{jh}^{(t)}$, and choose Tol
Repeat
  $m := m + 1$;
  $\beta_{jh}^{new}(m) := \beta_{jh}^{new}(m-1) - J^{-1}(\beta_{jh}^{new}(m-1))G(\beta_{jh}^{new}(m-1))$;
Until $(\|\beta_{jh}^{new}(m) - \beta_{jh}^{new}(m-1)\| < \mathsf{Tol})$ or $(m = m_{\mathsf{max}})$
$\beta_{jh}^{(t+1)} := \beta_{jh}^{new}(m)$

Note: Because of the quadratic local convergence of the Newton-Raphson scheme, we can just use a *one-step* Newton-Raphson updating. However, in the event of bad guesses of initial values for β_{jh}, the full Newton-Raphson updating of Algorithm 14 should be preferred.

5.4.7 Updating the scale parameter φ_h

For the Bernoulli and the Poisson outcome, the scale parameter φ_h is equal to 1. For a Gaussian however, we have $\varphi_h = \sigma_h^2$, and it is straightforward to estimate it. Note that we consider the error to have the same distribution across of the components of the model. That is why we only have φ_h instead of φ_{jh}. In fact, for a Gaussian outcome, our loglikelihood for a single outcome x_{ih} simply becomes

$$\ell(\boldsymbol{\theta}, \mathbf{X}^*) = \sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\left[-\frac{1}{2}\frac{1}{\sigma_h^2}(x_{ih} - \tilde{\boldsymbol{z}}_i^\top \boldsymbol{\beta}_{jh})^2 - \frac{1}{2}\log(\sigma_h^2)\right]. \tag{5.33}$$

If we solve the expected score equation $\mathbb{E}\left[S(\sigma_h^2)\right]$ for σ_h^2, it is straightforward to find that for $h = 1, \cdots, p$, the update $(\sigma_h^2)^{(t+1)}$ is given by

$$(\sigma_h^2)^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{E}\left[y_{ij}\left[(x_{ih} - \tilde{\boldsymbol{z}}_i^\top \boldsymbol{\beta}_{jh})^2\right]\right]. \tag{5.34}$$

5.4.8 Aspects of the derived scheme

It is important to note that the "mixed" form of the input space has as an immediate consequence the complication of the whole computational procedure. The computation of densities for instance requires calls to different functions, and this leads to computational inefficiency and the need for a case by case treatment. In other words, the generality enjoyed when dealing with input spaces with variables of the same type is lost. On the other hand, the Monte Carlo approximation that we use here is extremely slow, almost rendering the whole scheme impractical for tasks of moderately high dimension ($q > 6$, $k > 3$, $p > 5$). For instance, the computation of the observed-data loglikelihood alone is a very demanding task in this context. In fact the marginal density $\boldsymbol{p}(\boldsymbol{x})$ of $\boldsymbol{x}$ is given by

$$\boldsymbol{p}(\boldsymbol{x}) = \int\int_{\mathcal{H}} \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})\boldsymbol{p}(\boldsymbol{y})\boldsymbol{p}(\boldsymbol{z})d\boldsymbol{y}d\boldsymbol{z} = \sum_{j=1}^{k}\pi_j \int_{\mathcal{Z}} \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j, \boldsymbol{z})\boldsymbol{p}(\boldsymbol{z})d\boldsymbol{z}. \tag{5.35}$$

Since there is no closed-form expression for the above marginal density, we resort to its Monte Carlo approximation which in this case is given by

$$\boldsymbol{p}(\boldsymbol{x}) \approx \frac{1}{D}\sum_{j=1}^{k}\sum_{d=1}^{D}\pi_j \boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y} = j, \boldsymbol{z}_d) \approx \frac{1}{D}\sum_{d=1}^{D}\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d, \boldsymbol{z}_d). \tag{5.36}$$

If we used the above $\boldsymbol{p}(\boldsymbol{x})$, then the Monte Carlo approximation for the observed-data loglikelihood is given by

$$\boldsymbol{\ell}(\boldsymbol{\theta};\mathbf{X}) \approx \frac{1}{D}\sum_{i=1}^{n}\sum_{j=1}^{k}\sum_{d=1}^{D}\pi_j \boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{y}=j,\boldsymbol{z}_d) \approx \frac{1}{D}\sum_{i=1}^{n}\sum_{d=1}^{D}\boldsymbol{p}(\boldsymbol{x}_i|\boldsymbol{y}_d,\boldsymbol{z}_d). \tag{5.37}$$

If we consider the need to draw samples from the distribution of $\boldsymbol{y}$ and $\boldsymbol{z}$ with all the loops involved in the expression of $\boldsymbol{\ell}(\boldsymbol{\theta};\mathbf{X})$ plus the fact that the computation of $\boldsymbol{p}(\boldsymbol{x}|\boldsymbol{y}_d,\boldsymbol{z}_d)$ is not straightforward, then it becomes obvious that the computational burden is indeed heavy.

5.5 Implementing MFA with mixed outcomes

Our main aim in the numerical examples of this chapter is parameter estimation. We have exclusively used Monte Carlo approximations to tackle the intractability of integrals, and in most cases we have not used large enough Monte Carlo samples, because of the slowness of Matlab 6.0 in dealing with this. We like to emphasize the fact that our simulations in this chapter are essentially experimental, and that the unsuitability of Matlab in this case has prevented us from exploring as many numerical examples as desirable. We will concentrate in our future work on improving the computational scheme so as to make it ready for use in practical applications.

5.5.1 Estimating the $\boldsymbol{\beta}_{jh}$

In this first example, we assume the mixing proportions known, and we focus on the estimation of the GLM parameters $\boldsymbol{\beta}_{jh}$. We consider an artificial problem with a 3-dimensional input space having three different types of outcome, namely one Bernoulli, one Poisson and one Gaussian. Our mixture in this case has 2 components, with $\pi_1 = 0.60$ and $\pi_2 = 0.40$. Our sample size here is $n = 200$. $\Lambda_1 = (0.90, 0.40, 0.50)^\mathsf{T}$, $\Lambda_2 = (0.20, 0.90, 0.40)^\mathsf{T}$, $\mu_1 = (1.10, 1.20, 2.40)^\mathsf{T}$, and $\mu_2 = (-1.00, 2.40, 1.40)^\mathsf{T}$. In other words, the parameters that we wish to estimate have true values $\boldsymbol{\beta}_{11} = (0.90, 1.10)^\mathsf{T}$, $\boldsymbol{\beta}_{12} = (0.40, 1.20)^\mathsf{T}$, $\boldsymbol{\beta}_{13} = (0.50, 2.40)^\mathsf{T}$, $\boldsymbol{\beta}_{21} = (0.20, -1.00)^\mathsf{T}$, $\boldsymbol{\beta}_{22} = (0.90, 2.40)^\mathsf{T}$ and $\boldsymbol{\beta}_{23} =$

$(0.40, 1.40)^T$. After 17 EM iterations, the estimates obtained are the following: $\hat{\beta}_{11} = (0.85, 0.88)^T$, $\hat{\beta}_{12} = (0.44, 1.36)^T$, $\hat{\beta}_{13} = (0.52, 2.47)^T$, $\hat{\beta}_{21} = (0.18, -0.81)^T$, $\hat{\beta}_{22} = (1.11, 2.10)^T$, and $\hat{\beta}_{23} = (0.13, 1.62)^T$. For our Monte Carlo approximations, we used $D = 200$. In principle, one would be expected to use a far larger number of Monte Carlo draws in order to hope for an accurate approximation. The small number we used here seems satisfactory, probably because of the low dimensionality ($p = 3$) of the task at hand, and the fact that the "random" starting values must have happened to be in the region of the parameter space containing the true values of the parameters. The least one can say about these estimates after 17 iterations is that they are reasonably good [1].

5.5.2 Estimating the mixing proportions π

We reconsider the above synthetic task, this time with $k = 3$, $p = 3$, $q = 1$, $\pi_1 = 0.30$, $\pi_2 = 0.45$ and $\pi_3 = 0.25$. We use the same set of parameters β with the β's for component 3 given by $\beta_{31} = (0.40, -2.00)^T$, $\beta_{32} = (0.50, 2.00)^T$, and $\beta_{33} = (0.90, 0.40)^T$. In this section, we assume β known and fixed, and we use the scheme to estimate our mixing proportions. We use $\pi_1^{(0)} = 0.30$, $\pi_2^{(0)} = 0.35$ and $\pi_3^{(0)} = 0.35$ as initial guesses, and $D = 1000$ as our Monte Carlo sample size. The first 30 iterations of the EM algorithm yield $\hat{\pi}_1^{(30)} = 0.40$, $\hat{\pi}_2^{(30)} = 0.32$ and $\hat{\pi}_3^{(30)} = 0.28$, which seem to be converging to a set of mixing proportions reasonably close to the true values.

5.5.3 Estimating both β_{jh} and π

We reconsider the synthetic task of the previous example, but this time $\pi_1 = 0.30$, $\pi_2 = 0.45$ and $\pi_3 = 0.25$. $\beta_{11} = (0.90, 1.10)^T$, $\beta_{12} = (0.40, 1.20)^T$, $\beta_{13} = (0.50, 2.50)^T$, $\beta_{21} = (0.20, -0.50)^T$, $\beta_{22} = (0.90, 1.50)^T$, $\beta_{23} = (0.40, 1.50)^T$, $\beta_{31} = (0.40, -1.10)^T$, $\beta_{32} = (0.50, 1.80)^T$, and $\beta_{33} = (0.90, 0.50)^T$. Here, we use $D = 1000$ as our Monte Carlo

[1] In toy (artificial) problems where we know the true values of the parameters, a reasonable solution is one that is close enough to the true (known) values. In such cases, if a starting point leads to limiting values that are too far from the true values, other starting points are used. In practice (when the true values are unknown), we do something similar. Since the properties of the algorithm guarantee monotonic convergence to a maximum of $\ell(\theta, \mathbf{X})$, we systematically try at least a couple of starting points and pick the one that yields the highest crude approximation of $\ell(\theta, \mathbf{X})$.

sample size. We run our scheme to estimate both β_{jh} and π. Our sample size in this case is $n = 200$. With random starting values for β and $\pi_1^{(0)} = 0.60$, $\pi_2^{(0)} = 0.25$ and $\pi_3^{(0)} = 0.15$ as our initial values for the mixing proportions, the first 30 EM iterations produce $\hat{\pi}_1^{(30)} = 0.37$, $\hat{\pi}_2^{(30)} = 0.45$ $\hat{\pi}_3^{(30)} = 0.18$, and $\hat{\beta}_{11} = (0.81, 1.25)^\mathsf{T}$, $\hat{\beta}_{12} = (0.37, 1.17)^\mathsf{T}$, $\hat{\beta}_{13} = (0.32, 2.50)^\mathsf{T}$, $\hat{\beta}_{21} = (0.12, -0.38)^\mathsf{T}$, $\hat{\beta}_{22} = (0.97, 1.45)^\mathsf{T}$, $\hat{\beta}_{23} = (0.32, 1.71)^\mathsf{T}$, $\hat{\beta}_{31} = (0.25, -0.57)^\mathsf{T}$, $\hat{\beta}_{32} = (0.50, 1.74)^\mathsf{T}$, and $\hat{\beta}_{33} = (0.83, 0.88)^\mathsf{T}$. One of the most striking features in these simulations is the local behaviour of the estimation scheme. In fact, when starting values are not close enough to the true values of the parameters of interest, the scheme is very slow to converge, and in some cases eventually gets trapped in local maxima, producing solutions that are in most cases not satisfactory.

Another important point to note here is the fact that the Monte Carlo approximation can cause the overall scheme to become unstable if the number of Monte Carlo samples is not large enough ($D \geq 1000$) to allow accurate approximations of the integrals of interest at each step of the EM algorithm.

The above reasonable solution is obtained after trying various starting points. This approach which consists of trying different starting values is very commonly used when dealing with the EM algorithm for mixture models, because of the tendency of these EM algorithms to get trapped in local (and often not very useful) regions of the parameter surface. In this particular context, the shape of the parameter surface must be quite complex, and it therefore comes as no surprise if the scheme gets trapped in local regions. On the other hand, to lessen the overall computational burden, we only used the one-step Newton-Raphson in our simulations, but we anticipate that a full Newton-Raphson at each EM iteration should improve the speed of convergence.

It is important to note that convergence problems were less serious when we only estimated either the mixing proportions or the GLM parameters. An obvious reason for that could be the fact that the simultaneous estimation of both π and β is a much more complex task requiring the search of a much larger parameter space. When we studied the generic MFA model, the Bayesian treatment allowed us to circumvent some aspects of these local maxima, as the sampling scheme was less dependent on initial guesses. We

will be addressing these issues in our future work.

As far as the implementation of our scheme is concerned, it is important to note that Matlab 6.0 that we use for our simulations is not particularly suitable when many loops are used in the computations, as is the case here. We therefore plan to implement our scheme in C++ in our future work, and we anticipate that this is likely to improve the computational efficiency.

5.6 Approximation by Gauss-Hermite quadrature

5.6.1 Introduction

Let $\mathcal{Z} = I\!R^q$ be our domain of integration. The use of the Gauss-Hermite quadrature approximation rests on the ability to express the integrand $f(\boldsymbol{z}) : I\!R^q \mapsto I\!R$ as a product of a weight function $w(\boldsymbol{z}) : I\!R^q \mapsto I\!R_+$ and another function, $g(\boldsymbol{z}) : I\!R^q \mapsto I\!R$, say. More precisely, if $f(\boldsymbol{z})$ is to be integrated, this method first expresses it as $f(\boldsymbol{z}) = w(\boldsymbol{z})g(\boldsymbol{z})$, where $w(\boldsymbol{z})$ is referred to as the weight function. The integral is then expressed as

$$\int_{I\!R^q} f(\boldsymbol{z})d\boldsymbol{z} = \int_{I\!R} \cdots \int_{I\!R} w(z_1, \cdots, z_q)g(z_1, \cdots, z_q)dz_1 \cdots dz_q. \tag{5.38}$$

In (5.38), $w(\boldsymbol{z}) = \exp(-\boldsymbol{z}^\mathsf{T}\boldsymbol{z}) = \exp(-z_1^2) \cdots \exp(-z_q^2)$. With the weight function specified, the Gauss-Hermite approximation to (5.38) is given by

$$\int_{I\!R^q} f(\boldsymbol{z})d\boldsymbol{z} \approx \sum_{i_1=1}^{n_1} w_{i_1}^{(1)} \cdots \sum_{i_q=1}^{n_q} w_{i_q}^{(q)} g(z_{i_1}^{(1)}, \cdots, z_{i_q}^{(q)}), \tag{5.39}$$

where $z_{i_r}^{(r)}$ is the i_r-th zero of the Hermite polynomial of degree n_r, and $w_{i_r}^{(r)}$ is the corresponding weight.

All the integrals that we need to calculate in our analysis involve the standard multivariate Gaussian density $\boldsymbol{p}(\boldsymbol{z}) = \mathcal{N}_q(\boldsymbol{z}; 0, \mathbf{I}_q)$. With $\mathcal{N}_q(\boldsymbol{z}; 0, \mathbf{I}_q) = (2\pi)^{-q/2}\exp(-\frac{1}{2}\boldsymbol{z}^\mathsf{T}\boldsymbol{z})$, we readily have an appropriate weight function. In fact, by considering a simple transformation $\tilde{\boldsymbol{z}} = \frac{1}{\sqrt{2}}\boldsymbol{z}$, we easily derive the weight function $w(\tilde{\boldsymbol{z}}) = \exp(-\tilde{\boldsymbol{z}}^\mathsf{T}\tilde{\boldsymbol{z}})$. As a result, the application of the Gauss-Hermite quadrature to our context is relatively straightforward.

5.6.2 Limitations of the Gauss-Hermite quadrature

As clearly explained by Fahrmeir and Tutz (1994), the number of nodes needed for the Gauss-Hermite approximation of (5.39) is $\prod_{r=1}^{q} n_r$. The bad news here is that this number increases exponentially with the number of dimensions q of the domain of integration. This exponential increase makes this method unappealing for integration in higher-dimensional spaces such as those we are interested in. Fahrmeir and Tutz (1994) also report that the method works well in practice for $q \leq 6$. For reasonably lower-dimensional integrals ($q < 6$), the method is found to be computationally far more efficient than Monte Carlo approximation as confirmed by Sammel, Ryan, and Legler (1997) in their study.

5.7 Conclusion and discussion

5.7.1 Modelling strengths

The extension of the MFA that we have studied in this chapter is justified by similar closely related work that we mentioned earlier. Our work in this regard can be seen as an extension of Sammel, Ryan, and Legler (1997) who studied the same type of model for a homogeneous input space, while we have now extended it to finite mixtures for which the input space is heterogeneous.

The good news here is that the use of the GLM machinery allows the derivation of an estimation procedure that is straightforward and easy to interpret. This is a strength, since the model at first sight has two sources of heterogeneity, namely the mixed form of the input variable and the mixture structure of the input space itself. One would imagine a complicated estimating procedure, but it is encouraging to remark that the scheme derived here is fairly straightforward.

5.7.2 Computational weaknesses

The bad news here is the computational inefficiency of the candidate methods used to tackle the intractability of the integrals of interest. The implementation of the algorithms in this case is indeed very inefficient, because it does require many different loops that cannot be vectorised (because of functions of different types) to speed up computations. Because of the heterogeneous nature of the conditional densities involved, it is difficult if not virtually impossible to perform efficient function calls for the computation of densities and likelihood values, and this is a heavy drawback in this context since the bulk of the computation effort lies in the calculation of these densities.

The Monte Carlo approximation that we have used in this context is so slow that it is very close to being impractical, especially when implemented in Matlab where loops cause programs to become inefficient. We anticipate that coding in C or C++ will render the scheme more efficient, and that we will then be able to use large Monte Carlo samples to guarantee accurate estimates of our integrals.

Another point worth stressing is the fact that one cannot visualise anything to assess the plausibility of the result obtained. With traditional models where the observed vector is not "mixed", one can project the original data onto the plane to have some ideas, albeit limited, of the underlying structure of the data. In this case, with different types, it is hard if not impossible to think of a visual representation. Moreover, the geometry of the parameter space for such a fundamentally heterogeneous model is certainly a very complex one, and this is likely to cause major difficulties in estimation procedures.

5.7.3 Future work

Despite the fact that its extension to multivariate settings runs into trouble for $q > 6$, a good alternative to the Monte Carlo approximation in this context is the Gauss-Hermite quadrature that we outlined earlier. We plan to explore this alternative in our future work.

Another alternative worth considering is the Bayesian treatment. As we have seen

throughout our research, the Bayesian approach has proven to be essentially more flexible than its counterparts, and it would certainly offer a good platform for addressing the analysis of a heterogeneous model like this one. Such a Bayesian treatment can be made efficient, especially if we use ingredients encountered in Chapter 3. All our conditional densities being from the regular exponential family, it is straightforward to find the corresponding conjugate priors for the parameters, and to get the conjugate posteriors. It is easy to see that the β_{jh} would naturally have a Gaussian conjugate prior, especially for a normal outcome. This way, it is easy to specify case by case a suitable prior distribution for the parameters of interest. Obviously, the π's in this case still have a Dirichlet distribution, since $\boldsymbol{y}$ clearly has multinomial distribution.

Chapter 6

Conclusion

If a man will begin with certainties, he shall end in doubts; but if he will be content to begin with doubts, he shall end in certainties
Francis Bacon

Throughout this thesis, we have provided a conclusion and a discussion at the end of each chapter. In this final chapter, we present a brief summary of our work, and we outline ideas for future research investigations.

6.1 Justification and relevance

As we have seen throughout, latent variable models can be used to tackle practical problems in a great variety of fields, ranging from social sciences to engineering to physical and biological sciences. Our contribution in this regard has been to propose relatively new models drawing inspiration from existing ones, and motivated by a potential for application to practical problems. Some of the models that we have studied have been around for quite some time now, but we have in every case relevant and useful extensions (or at least alternatives) capable of handling situations not treated by existing models. Chapter 4 and Chapter 5 are particularly illustrative in this regard. Although we have in some cases not provided applications of our work to real data, some of the references cited show that our proposed models can be used for practical applications.

6.2 Complexity of latent variable models

It is important to note that the price to pay for the vast potential of applicability of latent variable models is their inherent complexity.

6.2.1 Structural complexity

All the latent variable models we have studied in this thesis have in common the fact that they are both structurally complex and computationally intensive. In all cases, the number of model parameters tends to grow very fast as the dimensions of the observable and latent spaces are increased. On the other hand, all the models considered have some degree of lack of identifiability, making it difficult to find a *unique* set of parameters that characterise the model. It emerges from our work that there are mainly two ways to approach model identification, namely the use of constraints and the use of a decision-theoretic treatment of the quantities of interest.

6.2.2 Inferential difficulties

The large number of parameters and the lack of identifiability also make *interpretability* very difficult. In such cases therefore, unless parameters have clear and unambiguous physical meanings attached to them, it might well suffice to content oneself with a set of reasonably accurate parameter estimates that can then be used for such tasks as the estimation of expected latent scores. As we have seen throughout, parameter estimation is prone to inefficiency, because of the large number of parameters to be estimated in situations where there might not be much data.

6.2.3 Estimation and Computational difficulties

Throughout our work, we have addressed statistical estimation and inference from both the likelihood-based and Bayesian perspectives. From both perspectives, the structural complexity of the model makes it difficult to use traditional estimation techniques. Our contribution in this regard has been the construction of variants of the EM algorithm and

Data Augmentation, and to some extent the adaptation of recent ideas from stochastic geometry. The main strength of our algorithms is the ease of interpretation that they allow, and the fact that they are all relatively easy to implement.

6.3 Future work

6.3.1 Mixtures of oblique Factor analysers

Throughout our work, we have assumed an underlying orthogonal factor model. A possible future direction would be to consider the assumption of an underlying oblique factor model, and to study the corresponding Mixture of Oblique Factor Analysers. Such an extension of the generic MFA model should have many applications in social sciences, in the light of the example from Press (1972) given in Chapter2.

6.3.2 A sampling alternative to varimax

When it comes to achieving better interpretability of a factor analysis solution, rotation, and in particular Kaiser's varimax rotation, tends to be the most commonly preferred approach. Intuitively expressed, the key idea behind Kaiser's varimax rotation is to find a new solution such that, for each row of the matrix of factor loadings, only one entry has a large magnitude (close to 1 when the analysis is based on the correlation matrix) while all the other entries are as small as possible, so small that they can be set to zero. While Kaiser achieves this through a deterministic optimisation procedure with a well-defined objective function, it is our belief that the same idea can be incorporated more much efficiently in the overall Bayesian sampling scheme. We are currently exploring ways to specify an extra layer in the hierarchical structure of the model, so that a carefully chosen hyperprior on Λ can allow its posterior distribution to concentrate its sampling of rows on realisations for which the above maximum variation criterion is met. A very naive alternative solution to the above extra layer idea would be to perform rejection sampling with a systematic rejection of draws not verifying the condition. Unfortunately

such an alternative is not attractive in multivariate settings since efficiency would be lost due to the fact that rejection slows sampling schemes considerably.

6.3.3 Efficient sampling

Despite the flexibility of the stochastic simulation approach used in this thesis, we have noted its relative computational inefficiency in terms of slow convergence. Part of our future work will be focused on the derivation of more efficient sampling schemes.

6.3.4 Applications

Many of the models that we have studied in this thesis have the potential to be used in practical applications. We are currently in the process of applying the generic Mixture of Factor Analysers model to the analysis of genetic data.

Bibliography

Akaike, H. (1987). Factor Analysis and AIC. *Psychometrika 52*, 317–332.

Anderson, T. W. (1984). *An Introduction Multivariate Statistical Analysis* (Second ed.). Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Arminger, G., P. Stein, and J. Wittenberg (1999). Mixtures of conditional mean and covariance structure models. *Psychometrika 65*, 475–494.

Baddeley, A. (1994). Discussion Representation of Knowledge in Complex Systems *by Grenander & Miller. Journal of the Royal Statistical Society, Series B 56*, 584–585.

Barndorff-Nielsen, O., W. Kendall, and M. van Lieshout (1999). *Stochastic Geometry: Likelihood and Computation.* Monographs on Statistics and Applied Probability. Chapman & Hall.

Bartholomew, D. J. (1987). *Latent Variable Models and Factor Analysis.* Griffin's Statistical Monographs and Courses. Charles Griffin & Company Limited.

Bartlett, M. (1954). A note on the multiplying factors for various χ^2 approximations. *Journal of the Royal Statistical Society, Series B 16*, 296–298.

Beltrando, G. (1990). Space-time variability of rainfall in april and october-november over east africa during the period 1932-1983. *Int. J. Climatol. 10*, 691–702.

Bernardo, J. M. and A. F. M. Smith (1994). *Bayesian Theory.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Bishop, C. M. (1995). *Neural Networks and Pattern Recognition* (First ed.). Oxford University Press.

BIBLIOGRAPHY

Blake, C. and C. Merz (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Box, G. E. P. and G. C. Tiao (1973). *Bayesian Inference in Statistical Analysis.* Addison-Wesley.

Celeux, G. (1998). *Bayesian Inference for Mixtures: The Label Switching Problem.* In *COMPSTAT 1998 Proceedings in Computational Statistics*, Volume of Short Communications and Posters. Springer Verlag.

Celeux, G., M. Hurn, and C. Robert (2000). Computational and Inferential difficulties with Mixture Posterior Distributions. *Journal of the American Statistical Association 95*, 957–970.

Cowles, M. and B. Carlin (1996). Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association 91*, 883–904.

Csató, L., E. Fokoué, M. Opper, B. Schottky, and O. Winther (2000). *Efficient Approaches to Gaussian Process Classification.* In S. A. Solla, T. K. Leen, and K. R. Muller (Eds.), *Advances in Neural Information Processing Systems*, Number 12. MIT Press.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm (with discussion). *Journal of The Royal Statistical Society, Series B 39*, 1–38.

Diebolt, J. and C. P. Robert (1994). Estimation of Finite Mixture Distributions through Bayesian Sampling. *Journal of the Royal Statistical Society, Series B. 56*(2), 363–375.

Dolan, C. and H. Van der Maas (1998). Fitting multivariate normal finite mixtures subject to structural equation modelling. *Psychometrika 63*, 227–253.

Dong, K. and J. M. G. Taylor (1995). The Restricted EM Algorithm for Maximum Likelihood Estimation Under Linear Restrictions on the Parameters. *Journal of the American Statistical Association 90*, 707–716.

Eastment, H. T. and W. J. Krzanowski (1982). Cross-validatory choice of the number of components from a principal component analysis. *Technometrics 24*, 73–77.

Escobar, M. and M. West (1995). Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association 90*, 577–588.

Everitt, B. S. (1984). *An Introduction to Latent Variable Models* (First ed.). Monographs on Statistics and Applied Probability. Chapman and Hall.

Everitt, B. S. and D. J. Hand (1981). *Finite Mixture Distributions* (First ed.). Monographs on Statistics and Applied Probability. Chapman and Hall.

Fahrmeir, L. and G. Tutz (1994). *Multivariate Statistical Modelling Based on Generalized Linear Models.* Springer Series in Statistics. Springer.

Fokoué, E. (1998, September). Mean Field Methods For Gaussian Process Classifiers. Master's thesis, Aston University, Neural Computing Research Group - Department of Computer Science and Applied Mathematics, Aston University, Birmingham, B4 7ET, England, UK.

Fokoué, E. (2000). *A Markov Chain Monte Carlo (MCMC) Approach to the Bayesian Analysis of Mixtures of Factor Analysers.* In W. Jansen and J. G. Bethlehem (Eds.), *Proceedings in Computational Statistics 2000,* Volume of Short Communications and Posters. Statistics Netherlands.

Fokoué, E. and D. M. Titterington (2000a). *Bayesian Sampling for Mixtures of Factor Analysers.* Technical Report 00-3, University of Glasgow, Department of Statistics, University of Glasgow, Glasgow, G12 8QW, Scotland-United Kingdom.

Fokoué, E. and D. M. Titterington (2000b). *Mixtures of Factor Analysers allowing a mix of continuous and non continuous manifest variables.* Technical Report 00-10, University of Glasgow, Department of Statistics, University of Glasgow, Glasgow, G12 8QW, Scotland-United Kingdom.

Fokoué, E. and D. M. Titterington (2000c). *Mixtures of Factor Analysers with fixed observed covariates.* Technical Report 00-4, University of Glasgow, Department of Statistics, University of Glasgow, Glasgow, G12 8QW, Scotland-United Kingdom.

Fokoué, E. and D. M. Titterington (2000d). *Stochastic Model Selection for Mixtures of Factor Analysers.* Technical Report 00-5, University of Glasgow, Department of Statistics, University of Glasgow, Glasgow, G12 8QW, Scotland-United Kingdom.

Fokoué, E. and D. M. Titterington (2001). *Mixtures of Factor Analysers: Bayesian Estimation and Inference by Stochastic Simulation. Machine Learning: Special Issue on MCMC To appear*, To appear.

Gelman, A. (1995). Inference and Monitoring Convergence. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in Practice* (First ed.)., Inderdisciplinary Statistics, Chapter 8, pp. 131–143. Chapman & Hall.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (1995). *Bayesian Data Analysis.* Texts in Statistical Science. Chapman and Hall.

Gelman, A., S. L. Meng, and H. Stern (1996). Posterior Predictive Assessment of model fitness via realised discrepancies. *Statistica Sinica 6*, 733–759.

Gelman, A. and D. Rubin (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science 7*, 457–511.

Geyer, C. (1992). Practical Markov chain Monte Carlo (with discussion). *Statistical Science 7*, 473–511.

Ghahramani, Z. and M. Beal (2000). *Variational Inference for Bayesian Mixture of Factor Analysers.* In S. A. Solla, T. K. Leen, and K. R. Muller (Eds.), *Advances in Neural Information Processing Systems*, Number 12. MIT Press.

Ghahramani, Z. and G. E. Hinton (1997). The EM Algorithm for Mixtures of Factor Analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 6 King's College Road, Toronto, Canada, M5S 1A4.

Graybill, F. A. (1969). *Introduction to Matrices with Applications in Statistics* (First ed.). Wadsworth Publishing.

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

BIBLIOGRAPHY

Grenander, U. and M. Miller (1994). Representation of Knowledge in Complex Systems. *Journal of the Royal Statistical Society, Series B 56*, 549–603.

Hurn, M., A. Justel, and C. Robert (2000). Estimating Mixtures of Regressions. Technical report, University of Bath, Department of Mathematical Sciences.

Ihara, M. and Y. Kano (1995). Identifiability of full, marginal and conditional factor analysis model. *Statistics and Probability Letters 23*, 343–350.

Johnson, R. A. and D. W. Wichern (1998). *Applied Multivariate Statistical Analysis* (fourth ed.). Prentice Hall.

Jolliffe, I. T. (1986). *Principal Component Analysis.* Springer.

Jöreskog, K. (1975). Factor Analysis by Least Squares and Maximum Likelihood. In K. Enslein, A. Ralston, and H. Wilf (Eds.), *Statistical Methods for Digital Computers.* John Wiley.

Krane, W. and R. McDonald (1978). Scale invariance and the factor analysis of correlation matrices. *Br. J. Math. Statist. Psychol. 31*, 218–228.

Krzanowski, W. and F. Marriott (1994). *Multivariate Analysis* (First ed.). Kendall's Library of Statistics 1. Edward Arnold.

Krzanowski, W. and F. Marriott (1995). *Multivariate Analysis* (First ed.). Kendall's Library of Statistics 2. Arnold.

Lawley, D. N. and A. E. Maxwell (1971). *Factor Analysis as a Statistical Method* (second ed.). Butterworths.

Lee, S. and J. Shi (1999). *Bayesian Analysis of Structural Equation Model with Fixed Covariates.* Technical report, Department of Statistics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

Liu, C. and D. Rubin (1994). The ECME Algorithm: A simple extension of EM and ECM with fast monotone convergence. *Biometrika 81*, 633–48.

Lopes, H. F. and M. West (1999). Model uncertainty in factor analysis. Technical Report ISDS, Institute of Statistics and Decision Sciences, Duke University.

Manly, B. F. J. (1986). *Multivariate Statistical Methods: A Primer.* Chapman & Hall.

Marriott, F. (1974). *The Interpretation of Multiple Observations.* London:Academic Press.

Martin, J. and R. McDonald (1981). Bayesian estimation in unrestricted factor analysis: A treatment for Heywood cases. *Psychometrika 40*, 505–517.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models* (second ed.). Monographs on Statistics and Applied Probability. Chapman & Hall.

McLachlan, G. and D. Peel (2000). *Finite Mixture Models.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

McLachlan, G. J. and T. Krishnan (1997). *The EM Algorithm and Extensions.* Wiley Series in Probability and Statistics. John Wiley & Sons.

Morgan, B. (2000). *Applied Stochastic Modelling.* Arnold.

Muthén, B. and K. Shedden (1999). Finite Mixture Modelling with Mixture Outcomes using the EM Algorithm. *Biometrics 55*, 463–469.

Nobile, A. (1998). A hybrid markov chain for the bayesian analysis of multinomial probit model. *Statistics and Computing 8*, 229–242.

Phillips, D. and A. Smith (1995). Bayesian model comparison via jump diffusions. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in Practice* (First ed.)., Inderdisciplinary Statistics, Chapter 13, pp. 215–240. Chapman & Hall.

Press, S. J. (1972). *Applied Multivariate Analysis* (First ed.). Holt, Rinehart and Winston, Inc.

Preston, C. (1976). Spatial birth-and-death process. *Bull. Inst. Internat. Statist. 46*, 371–391.

Richardson, S. and P. J. Green (1997). On The Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion). *Journal of the Royal Statistical Society, Series B. 59*, 731–792.

Ripley, B. (1977). Modelling spatial patterns (with discussion). *J. Roy. Statist. Soc. Ser. B 39*, 172–212.

Robert, C. P. and G. Casella (2000). *Monte Carlo Statistical Methods.* Springer Texts in Statistics. Springer.

Rubin, D. and D. Thayer (1982). EM algorithms for ML factor analysis. *Psychometrika 47*, 69–76.

Rubin, D. and D. Thayer (1983). More on EM for ML factor analysis. *Psychometrika 48*, 253–257.

Rue, H. (1995). New loss functions in bayesian imaging. *J. Amer. Statist. Assoc. 90*, 900–908.

Sammel, M. D., L. M. Ryan, and J. M. Legler (1997). Latent variable models for mixed discrete and continuous outcomes. *Journal of the Royal Statistical Society, Series B. 59*, 667–678.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data.* Monographs on Statistics and Applied Probability. Chapman & Hall.

Shi, J.-Q. and S.-Y. Lee (2000). Latent variable models with mixed continuous and polytomous data. *Journal of the Royal Statistical Society, Series B 62*, 77–87.

Spearman, C. (1904). General intelligence, objectively determined and measured. *American Journal of Psychology 19*, 201–293.

Stephens, M. (2000). Bayesian Analysis of Mixtures models with an Unknown Number of Components - An alternative to Reversibe jump methods. *Annals of Statistics 28*, 40–74.

Stoyan, D., W. Kendall, and J. Mecke (1995). *Stochastic Geometry and its Applications* (second ed.). Wiley Series in Probability and Statistics. John Wiley & Sons.

Tanner, M. A. and W. H. Wong (1987). The Calculation of posterior distributions by Data Augmentation (with discussion). *Journal of the American Statistical Association 82*, 528–550.

Thompson, T. J., P. J. Smith, and J. P. Boyle (1998). Finite mixture models with concomitant information: assessing diagnostic criteria for diabetes. *Journal of the Royal Statistical Society, Series C (Applied Statistics) 47*, 393–404.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics 22*, 1701–1762.

Tipping, M. E. and C. M. Bishop (1999). Mixtures of Probabilistic Principal Component Analysers. *Neural Computation 11*, 443–482.

Titterington, D. M., A. F. M. Smith, and U. E. Makov (1985). *Statistical Analysis of Finite Mixture Distributions.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Ueda, N., R. Nakano, Z. Ghahramani, and E. Hinton, G. (2000). SMEM Algorithm for Mixture Models. *Neural Computation 12*, 2019–2128.

Utsugi, A. and T. Kumagai (2001). *Bayesian Analysis of Mixtures of Factor Analysers. Neural Comptation 13*, 993–1002.

van Lieshout, M. (1994). Discussion on *representation of knowledge in complex systems by grenander & miller. Journal of the Royal Statistical Society, Series B 56*, 585.

Velicer, W. F. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika 41*, 321–327.

von Eye, A. and C. Clogg (1994). *Latent Variables Analysis.* Sage Publications.

Wold, S. (1978). Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics 20*, 397–405.

Wu, C. (1983). On the Convergence Properties of the EM Algorithm. *Annals of Statistics 11*, 95–103.

Yung, Y. F. (1997). Finite mixtures in confirmatory factor analysis models. *Psychometrika 62*, 297–330.

Zellner, A. (1971). *An Introduction to Bayesian Inference in Econometrics.* Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Appendix A

General Theorems and Formulae

A.1 General definitions

A.1.1 Direct product of matrices

Let $A = (a_{ij})$ be a $p \times p$ matrix, and $B = (a_{kl})$ be an $n \times n$ matrix. Then the $pn \times pn$ matrix $C = A \otimes B$ is called the direct product (or Kronecker product) of A and B, and is given by

$$C = A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1p}B \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ a_{p1}B & a_{p2}B & \cdots & a_{pp}B \end{bmatrix}$$

Direct products of matrices as can be seen throughout our work are very useful in multivariate statistical analysis, escpecially in multivariate regression.

A.1.2 Conditional normal distributions

Theorem A.1 *Let* $V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$ *be a normally distributed random vector with a mean* $\mu_V = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ *and a covariance matrix* $\Sigma_V = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$. *Then the conditional distribution of* V_1 *given* $(V_2 = v_2)$ *is also normal with a mean vector that is a linear function of*

v_2, and a covariance matrix that is independent of v_2. More specifically, $[V_1|V_2 = v_2] \sim \mathcal{N}(\mu_{1|2}, \Sigma_{1|2})$ where $\mu_{1|2} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(v_2 - \mu_2)$ and $\Sigma_{1|2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.

A.2 Matrix and vector operations

A.2.1 Vector representation of a matrix

Throughout our work, we use vec(A) to denote a vector representation of matrix A. The vector vec(A) stacks in a column vector the columns of the matrix A.

A.2.2 Important Multivariate Derivatives

In this section, we provide general formulae for multivariate derivatives that are used at the M-step of our EM algorithms. Let $\mathbf{x}$ and $\mathbf{u}$ be two p-dimensional vectors. Let $\mathbf{y}$ and $\mathbf{v}$ be two q-dimensional vectors. Let $\mathbf{A} \in I\!R^{p\times p}$ and $\mathbf{N} \in I\!R^{p\times q}$ be two real matrices. (Graybill 1969) and (Press 1972) provide details of the derivation of the following useful identities.

$$
\begin{aligned}
&\frac{\partial \mathbf{x}^T\mathbf{u}}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^T\mathbf{x}}{\partial \mathbf{x}} = \mathbf{u} \\
&\frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x} \\
&\frac{\partial \mathbf{u}^T\mathbf{N}\mathbf{v}}{\partial \mathbf{N}} = \mathbf{u}\mathbf{v}^T \quad \text{and} \quad \frac{\partial \mathbf{v}^T\mathbf{N}^T\mathbf{u}}{\partial \mathbf{N}} = \mathbf{u}\mathbf{v}^T \\
&\frac{\partial \mathbf{x}^T\mathbf{A}\mathbf{x}}{\partial \mathbf{A}} = \frac{\partial \mathbf{x}^T\mathbf{A}^T\mathbf{x}}{\partial \mathbf{A}} = \mathbf{x}\mathbf{x}^T \\
&\frac{\partial \mathbf{v}^T\mathbf{N}^T\mathbf{A}\mathbf{N}\mathbf{y}}{\partial \mathbf{N}} = \mathbf{A}^T\mathbf{N}\mathbf{v}\mathbf{y}^T + \mathbf{A}\mathbf{N}\mathbf{y}\mathbf{v}^T \\
&\frac{\partial \log|\mathbf{A}|}{\partial \mathbf{A}} = \left[\mathbf{A}^T\right]^{-1} = \left[\mathbf{A}^{-1}\right]^T \\
&\frac{\partial \mathbf{tr}\left[\mathbf{A}\mathbf{N}\right]}{\partial \mathbf{A}} = \frac{\partial \mathbf{tr}\left[\mathbf{N}\mathbf{A}\right]}{\partial \mathbf{A}} = \mathbf{N}^T \quad \text{and} \quad \frac{\partial \mathbf{tr}\left[\mathbf{A}^T\mathbf{N}\right]}{\partial \mathbf{A}} = \frac{\partial \mathbf{tr}\left[\mathbf{N}\mathbf{A}^T\right]}{\partial \mathbf{A}} = \mathbf{N} \\
&\frac{\partial \mathbf{tr}\left[\mathbf{A}^T\mathbf{N}\mathbf{A}\right]}{\partial \mathbf{A}} = (\mathbf{N} + \mathbf{N}^T)\mathbf{A}
\end{aligned}
\tag{A.1}
$$

Appendix B

Derivation of Estimation Equations

B.1 Elements of estimation for Factor Analysis

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{x}\boldsymbol{z}^{\mathsf{T}}\right] &= \mathbb{E}\left[(\Lambda\boldsymbol{z}+\boldsymbol{e})\boldsymbol{z}^{\mathsf{T}}\right] = \mathbb{E}\left[(\Lambda\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}+\boldsymbol{e}\boldsymbol{z}^{\mathsf{T}}\right] \\
&= \Lambda\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^{\mathsf{T}}\right]+\mathbb{E}\left[\boldsymbol{z}\boldsymbol{e}^{\mathsf{T}}\right] \\
&= \Lambda \qquad \text{(B.1)} \\
\mathbb{V}[\boldsymbol{x}] &= \mathbb{V}[\Lambda\boldsymbol{z}+\mu+\boldsymbol{e}] \\
&= \mathbb{V}[\Lambda\boldsymbol{z}]+\mathbb{V}[\boldsymbol{e}] \\
&= \Lambda\mathbb{V}[\boldsymbol{z}]\Lambda^{\mathsf{T}}+\mathbb{V}[\boldsymbol{e}] \\
&= \Lambda\Lambda^{\mathsf{T}}+\Sigma \qquad \text{(B.2)}
\end{aligned}$$

At the EM step of the EM algorithm for factor analysis, we form an analytical expression for $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}\left[\ell(\boldsymbol{\theta};\mathbf{X}^*)|\boldsymbol{\theta}^{(t)},\mathbf{X}\right]$, where $\ell(\boldsymbol{\theta};\mathbf{X}^*)$, is given by equation (2.18), i.e

$$\begin{aligned}
\ell(\boldsymbol{\theta};\mathbf{X}^*) \;\propto\; & -\frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Sigma^{-1}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathsf{T}}\right] + \sum_{i=1}^{n}\left[\boldsymbol{x}_i^{\mathsf{T}}\Sigma^{-1}\Lambda\boldsymbol{z}_i\right] + \sum_{i=1}^{n}\left[\boldsymbol{x}_i^{\mathsf{T}}\Sigma^{-1}\mu\right] \\
& - \sum_{i=1}^{n}\left[\mu^{\mathsf{T}}\Sigma^{-1}\Lambda\boldsymbol{z}_i\right] - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}\right] - \frac{1}{2}\sum_{i=1}^{n}\mu^{\mathsf{T}}\Sigma^{-1}\mu.
\end{aligned}$$

If we use identities from (A.1), we can easily compute the necessary partial derivatives needed at the M-step of the EM algorithm for both the FA and the MFA models.

B.1.1 Estimating the mean of the Gaussian

Estimating the mean μ in FA model is straightforward. In fact, as a function of μ, the function Q can be written as

$$Q(\mu) = -\frac{1}{2}\sum_{i=1}^{n}\left[-2\boldsymbol{x}_i^{\mathsf{T}}\Sigma^{-1}\mu + 2\mu^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right] + \mu^{\mathsf{T}}\Sigma^{-1}\mu\right].$$

Since $\frac{\partial\mu^{\mathsf{T}}\Sigma^{-1}\mu}{\partial\mu} = 2\Sigma^{-1}\mu$, $\frac{\partial\boldsymbol{x}_i^{\mathsf{T}}\Sigma^{-1}\mu}{\partial\mu} = \Sigma^{-1}\boldsymbol{x}_i^{\mathsf{T}}$ and $\frac{\partial\mu^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]}{\partial\mu} = \Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]$, it is easy to show that solving $\frac{\partial Q(\mu)}{\partial\mu} = 0$ yields the maximiser of $Q(\mu)$ that is given by

$$\mu^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}_i - \Lambda^{(t)}\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]\right).$$

B.1.2 Estimating the matrix of factor loading

If we treat Q as a function of Λ, we can express it as

$$Q(\Lambda) = \sum_{i=1}^{n}(\boldsymbol{x}_i-\mu)^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right] - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right]\right],$$

and the corresponding partial derivatives with respect to Λ are given by

$$\begin{aligned}\frac{\partial(\boldsymbol{x}_i-\mu)^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]}{\partial\Lambda} &= \Sigma^{-1}(\boldsymbol{x}_i-\mu)\left(\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]\right)^{\mathsf{T}}\\ \frac{\partial\mathbf{tr}(\Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right])}{\partial\Lambda} &= 2\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right].\end{aligned}$$

The solution of $\frac{\partial Q(\Lambda)}{\partial\Lambda} = 0$ yields the maximiser of $Q(\Lambda)$ which is explicitly given by

$$\Lambda^{(t+1)} = \left[\sum_{i=1}^{n}(\boldsymbol{x}_i-\mu^{(t+1)})\left(\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]\right)^{\mathsf{T}}\right]\left[\sum_{i'=1}^{n}\mathbb{E}\left[\boldsymbol{z}_{i'}\boldsymbol{z}_{i'}^{\mathsf{T}}|\boldsymbol{x}_{i'}\right]\right]^{-1}.$$

B.1.3 Estimating the uniquenesses Σ

As a function of Σ, Q can be written as follows:

$$\begin{aligned}Q(\Sigma) &= -\frac{n}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Sigma^{-1}(\boldsymbol{x}_i-\mu)(\boldsymbol{x}_i-\mu)^{\mathsf{T}}\right]\\ &+ \sum_{i=1}^{n}(\boldsymbol{x}_i-\mu)^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right] - \frac{1}{2}\sum_{i=1}^{n}\mathbf{tr}\left[\Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right]\right].\end{aligned}$$

It is easier to derive the partial derivatives of $Q(\Sigma)$ with respect to Σ^{-1}, and they are

$$\begin{aligned}
\frac{\partial \mathbf{tr}\left[\Sigma^{-1}(\boldsymbol{x}_i-\mu)(\boldsymbol{x}_i-\mu)^{\mathsf{T}}\right]}{\partial \Sigma^{-1}} &= (\boldsymbol{x}_i-\mu)(\boldsymbol{x}_i-\mu)^{\mathsf{T}} \\
\frac{\partial \log|\Sigma|}{\partial \Sigma^{-1}} &= -\frac{\partial \log|\Sigma^{-1}|}{\partial \Sigma^{-1}} = -\Sigma \\
\frac{\partial (\boldsymbol{x}_i-\mu)^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]}{\partial \Sigma^{-1}} &= (\boldsymbol{x}_i-\mu)\left(\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]\right)^{\mathsf{T}}\Lambda^{\mathsf{T}} \\
\frac{\partial \mathbf{tr}(\Lambda^{\mathsf{T}}\Sigma^{-1}\Lambda\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right])}{\partial \Sigma^{-1}} &= \Lambda\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right]\Lambda^{\mathsf{T}}.
\end{aligned}$$

The solution of $\frac{\partial Q(\Sigma)}{\partial \Sigma^{-1}} = 0$ yields the maximiser of $Q(\Sigma)$ which is given by

$$\begin{aligned}
\Sigma^{(t+1)} &= \frac{1}{n}\mathsf{diag}\left[\sum_{i=1}^{n}(\boldsymbol{x}_i-\mu^{(t+1)})(\boldsymbol{x}_i-\mu^{(t+1)})^{\mathsf{T}} - 2(\boldsymbol{x}_i-\mu^{(t+1)})[\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]]^{\mathsf{T}}[\Lambda^{(t+1)}]^{\mathsf{T}}\right] \\
&+ \frac{1}{n}\mathsf{diag}\left[\sum_{i=1}^{n}\left[\Lambda^{(t+1)}\mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|\boldsymbol{x}_i\right]\left[\Lambda^{(t+1)}\right]^{\mathsf{T}}\right]\right].
\end{aligned}$$

If we replace $\Lambda^{(t+1)}$ by the expression obtained above, then the update $\Sigma^{(t+1)}$ of Σ is now given by

$$\Sigma^{(t+1)} = \frac{1}{n}\mathsf{diag}\left[\sum_{i=1}^{n}\left(\boldsymbol{x}_i-\mu^{(t+1)}-\Lambda^{(t+1)}\mathbb{E}\left[\boldsymbol{z}_i|\boldsymbol{x}_i\right]\right)\left(\boldsymbol{x}_i-\mu^{(t+1)}\right)^{\mathsf{T}}\right].$$

B.2 Analysis of Mixtures of Factor Analysers

For simplicity, we use $\mathsf{a}_{ij}^{(t)}$, $\mathbf{b}_{ij}^{(t)}$ and $\mathbf{C}_{ij}^{(t)}$ as defined in Section (3.5.2), that is $\mathsf{a}_{ij}^{(t)} = \mathbb{E}\left[y_{ij}|\boldsymbol{x}_i\right]$, $\mathbf{b}_{ij}^{(t)} = \mathbb{E}\left[\boldsymbol{z}_i|y_{ij},\boldsymbol{x}_i\right]$ and $\mathbf{C}_{ij}^{(t)} = \mathbb{E}\left[\boldsymbol{z}_i\boldsymbol{z}_i^{\mathsf{T}}|y_{ij},\boldsymbol{x}_i\right]$.

B.2.1 E-Step for the generic MFA Model

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \mathbb{E}\left[\ell(\boldsymbol{\theta},\mathbf{X}^*)|\mathbf{X},\boldsymbol{\theta}^{(t)}\right] = \mathbb{E}\left[\log\left[\prod_{i=1}^{n}\left(\prod_{j=1}^{k}\pi_j^{y_{ij}}\left[\mathcal{N}(\boldsymbol{x}_i;\Lambda_j\boldsymbol{z}_i,\Sigma)\right]^{y_{ij}}\right)\right]\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{E}\left[y_{ij}|\boldsymbol{x}_i\right]\log(\pi_j) + \sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{E}\left[y_{ij}\log\left[\mathcal{N}(\boldsymbol{x}_i;\Lambda_j\boldsymbol{z}_i,\Sigma)\right]|\boldsymbol{x}_i\right] \quad \text{(B.3)}
\end{aligned}$$

B.2.2 Estimating the mixing proportions

As a function of $\boldsymbol{\pi}$, Q can be written as

$$Q(\boldsymbol{\pi}) = \sum_{i=1}^{n}\sum_{j=1}^{k}\mathbb{E}\left[y_{ij}|\boldsymbol{x}_i\right]\log(\pi_j) = \sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}\log(\pi_j). \quad \text{(B.4)}$$

To find estimates for the mixing proportions, all we need to do is find the set of π_j that Maximize $Q(\boldsymbol{\pi})$ subject to the constraint that $\sum_{j=1}^{k} \pi_j = 1$. By the Lagrange multiplier method, this means maximizing $\tilde{Q}(\boldsymbol{\pi}, \gamma) = Q(\boldsymbol{\pi}) + \gamma\left(1 - \sum_{j=1}^{k} \pi_j\right)$ where γ is the Lagrange multiplier, and $\tilde{Q}(\boldsymbol{\pi}, \gamma)$ the Lagrangian of the original function $Q(\boldsymbol{\pi})$.

$$\frac{\partial \tilde{Q}}{\partial \pi_j} = -\gamma + \frac{1}{\pi_j}\sum_{i=1}^{n} \mathsf{a}_{ij}^{(t)} \quad \text{and} \quad \frac{\partial \tilde{Q}}{\partial \gamma} = 1 - \sum_{j=1}^{k} \pi_j$$

To obtain the maximum, we solve $\frac{\partial \tilde{Q}}{\partial \pi_j} = \frac{\partial \tilde{Q}}{\partial \gamma} = 0$, from which we obtain

$$\pi_j^{(t+1)} = \frac{1}{\gamma}\sum_{i=1}^{n} \mathsf{a}_{ij}^{(t)}$$

Since the set of $\pi_j^{(t+1)}$ must satisfy the maximization constraint, we therefore have $\sum_{j=1}^{k} \pi_j^{(t+1)} = \frac{1}{\gamma}\sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)} = \frac{n}{\gamma} = 1$ which in turn leads to $\gamma = n$. Thus, we have

$$\pi_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \mathsf{a}_{ij}^{(t)} \tag{B.5}$$

B.2.3 Estimating the means of the Gaussians

Estimating the means $\mu_j^{(t+1)}$ of the Gaussians is rather straightforward. In fact, as a function of $\boldsymbol{\mu}$, the function Q can be written as

$$Q(\boldsymbol{\mu}) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)}\left[-2\boldsymbol{x}_i^{\mathsf{T}}\Sigma^{-1}\mu_j + 2\mu_j^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)} + \mu_j^{\mathsf{T}}\Sigma^{-1}\mu_j\right].$$

Since $\frac{\partial \mu_j^{\mathsf{T}}\Sigma^{-1}\mu_j}{\partial \mu_j} = 2\Sigma^{-1}\mu_j$, $\frac{\partial \boldsymbol{x}_i^{\mathsf{T}}\Sigma^{-1}\mu_j}{\partial \mu_j} = \Sigma^{-1}\boldsymbol{x}_i^{\mathsf{T}}$ and $\frac{\partial \mu_j^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)}}{\partial \mu_j} = \Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)}$, it is easy to show that solving $\frac{\partial Q(\boldsymbol{\mu})}{\partial \mu_j} = 0$ yields the maximiser of $Q(\boldsymbol{\mu})$ that is given by

$$\mu_j^{(t+1)} = \left[\sum_{i=1}^{n} \mathsf{a}_{ij}^{(t)}\left(\boldsymbol{x}_i - \Lambda_j^{(t)}\mathbf{b}_{ij}^{(t)}\right)\right]\left[\sum_{i'=1}^{n} \mathsf{a}_{i'j}^{(t)}\right]^{-1}.$$

B.2.4 Estimating the Factor Loadings

If we treat Q as a function of $\boldsymbol{\Lambda}$, we can express it as

$$Q(\boldsymbol{\Lambda}) = \sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)}(\boldsymbol{x}_i - \mu_j)^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{k} \mathsf{a}_{ij}^{(t)}\mathbf{tr}\left[\Lambda_j^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{C}_{ij}^{(t)}\right],$$

and the corresponding partial derivatives with respect to Λ_j are given by

$$\begin{aligned}\frac{\partial(\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)}}{\partial\Lambda_j} &= \Sigma^{-1}(\boldsymbol{x}_i-\mu_j)\left(\mathbf{b}_{ij}^{(t)}\right)^{\mathsf{T}}\\ \frac{\partial\mathbf{tr}(\Lambda_j^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{C}_{ij}^{(t)})}{\partial\Lambda_j} &= 2\Sigma^{-1}\Lambda_j\mathbf{C}_{ij}^{(t)}.\end{aligned}$$

The solution of $\frac{\partial Q(\Lambda)}{\partial\Lambda_j}=0$ yields the maximiser of $Q(\Lambda)$ which is explicitly given by

$$\Lambda_j^{(t+1)}=\left[\sum_{i=1}^{n}\mathsf{a}_{ij}^{(t)}(\boldsymbol{x}_i-\mu_j^{(t+1)})\left(\mathbf{b}_{ij}^{(t)}\right)^{\mathsf{T}}\right]\left[\sum_{i'=1}^{n}\mathsf{a}_{i'j}^{(t)}\mathbf{C}_{i'j}^{(t)}\right]^{-1}.$$

B.2.5 Estimating the uniquenesses Σ

As a function of Σ, Q can be written as follows:

$$\begin{aligned}Q(\Sigma) &= -\frac{n}{2}\log|\Sigma|-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}\mathbf{tr}\left[\Sigma^{-1}(\boldsymbol{x}_i-\mu_j)(\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}\right]\\ &+\sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}(\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)}-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}\mathbf{tr}\left[\Lambda_j^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{C}_{ij}^{(t)}\right].\end{aligned}$$

It is easier to derive the partial derivatives of $Q(\Sigma)$ with respect to Σ^{-1}, and they are

$$\begin{aligned}\frac{\partial\mathbf{tr}\left[\Sigma^{-1}(\boldsymbol{x}_i-\mu_j)(\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}\right]}{\partial\Sigma^{-1}} &= (\boldsymbol{x}_i-\mu_j)(\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}\\ \frac{\partial\log|\Sigma|}{\partial\Sigma^{-1}} &= -\frac{\partial\log|\Sigma^{-1}|}{\partial\Sigma^{-1}}=-\Sigma\\ \frac{\partial(\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{b}_{ij}^{(t)}}{\partial\Sigma^{-1}} &= (\boldsymbol{x}_i-\mu_j)\left(\mathbf{b}_{ij}^{(t)}\right)^{\mathsf{T}}\Lambda_j^{\mathsf{T}}\\ \frac{\partial\mathbf{tr}(\Lambda_j^{\mathsf{T}}\Sigma^{-1}\Lambda_j\mathbf{C}_{ij}^{(t)})}{\partial\Sigma^{-1}} &= \Lambda_j\mathbf{C}_{ij}^{(t)}\Lambda_j^{\mathsf{T}}.\end{aligned}$$

The solution of $\frac{\partial Q(\Sigma)}{\partial\Sigma^{-1}}=0$ yields the maximiser of $Q(\Sigma)$ which is given by

$$\Sigma=\frac{1}{n}\mathsf{diag}\left[\sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}\left((\boldsymbol{x}_i-\mu_j)\left((\boldsymbol{x}_i-\mu_j)^{\mathsf{T}}-2\left[\Lambda_j\mathbf{b}_{ij}^{(t)}\right]^{\mathsf{T}}\right)+\Lambda_j\mathbf{C}_{ij}^{(t)}\Lambda_j^{\mathsf{T}}\right)\right].$$

After simplification, the update $\Sigma^{(t+1)}$ of Σ is now given by

$$\Sigma^{(t+1)}=\frac{1}{n}\mathsf{diag}\left[\sum_{i=1}^{n}\sum_{j=1}^{k}\mathsf{a}_{ij}^{(t)}\left(\boldsymbol{x}_i-\mu_j^{(t+1)}-\Lambda_j^{(t+1)}\mathbf{b}_{ij}^{(t)}\right)\left(\boldsymbol{x}_i-\mu_j^{(t+1)}\right)^{\mathsf{T}}\right].$$

□