National and Kapodistrian University of Athens

Master's Thesis

Likelihood-Based Inference and Model Selection for Discrete-Time Finite State-Space Hidden Markov Models

Author:
Vasileios Katsianos

Supervisor:
Dr. Loukia Meligkotsidou

September 2018

A thesis submitted in partial fulfilment of the requirements
for the degree of M.Sc. in Statistics and Operations Research

in the

Faculty of Sciences
Department of Mathematics

Approved on _____ by the Evaluation Committee composed of:

Full Name	**Academic Rank**	**Signature**
Antonis Economou	Professor	_____
Loukia Meligkotsidou (supervisor)	Assistant Professor	_____
Samis Trevezas	Lecturer	_____

Abstract

Vasileios Katsianos

Likelihood-Based Inference and Model Selection for
Discrete-Time Finite State-Space Hidden Markov Models

Hidden Markov Models (HMMs) are one of the most fruitful statistical modelling concepts that have appeared in the last fifty years. The use of latent states makes HMMs generic enough to handle a wide array of complex real-world time series, while the relatively straightforward dependence structure still allows for the use of efficient computational procedures. This dissertation concerns itself with the presentation of frequentist and Bayesian methods for statistical inference and model selection in the context of HMMs. These methods are, then, applied on real and simulated data in order to gauge their accuracy and efficiency.

HMMs belong in a general class of models referred to as missing data problems. In the context of frequentist statistics, the Expectation-Maximisation (EM) algorithm approximates the maximum likelihood estimator (MLE) of the parameter vector in a missing data problem, whereas, in the framework of Bayesian statistics, Markov Chain Monte Carlo (MCMC) methods, especially the full-conditional Gibbs sampler, are applicable to approximate the posterior distribution of the parameter vector. These methods are first applied for parameter estimation in finite mixture models, which may be regarded as special cases of HMMs, where no dependence is allowed whatsoever between subsequent observations.

In the case of HMMs some form of forward-backward recursion is additionally required in order to compute the conditional distribution of the hidden variables, given the observations. This so called Forward-Backward algorithm may be combined either with the EM algorithm or some MCMC method for parameter estimation. Lastly, we examine methods for selecting the number of hidden states in an HMM. The frequentist approach usually entails the approximation of the generalised likelihood-ratio (LR) statistics through some bootstrap technique, while the Bayesian approach relies either on trans-dimensional MCMC methods, which incorporate moves between different models along with parameter estimation, or on simulation methods to approximate the marginal likelihoods of the competing models.

Περίληψη

Βασίλειος Κατσιανος

Συμπερασματολογία Βασισμένη στην Πιθανοφάνεια και Επιλογή Μοντέλου για Κρυμμένα Μαρκοβιανά Μοντέλα Διακριτού Χρόνου και Πεπερασμένου Χώρου Καταστάσεων

Τα Κρυμμένα Μαρκοβιανά Μοντέλα (ΚΜΜ) είναι μία από τις πιο καρποφόρες ιδέες στατιστικής μοντελοποίησης που έχει αναπτυχθεί τα τελευταία πενήντα χρόνια. Η χρήση λανθανουσών καταστάσεων καθιστά τα ΚΜΜ αρκετά γενικού χαρακτήρα για να διαχειριστούν ένα ευρύ φάσμα πολύπλοκων πραγματικών χρονοσειρών, ενώ η σχετικά απλή δομή εξάρτησής τους επιτρέπει τη χρήση αποτελεσματικών υπολογιστικών διαδικασιών. Αυτή η διπλωματική εργασία ασχολείται με την παρουσίαση Κλασικών και Μπεϋζιανών μεθόδων συμπερασματολογίας και επιλογής μοντέλου για ΚΜΜ. Στη συνέχεια, αυτές οι μέθοδοι εφαρμόζονται σε πραγματικά και προσομοιωμένα δεδομένα με στόχο να διαπιστωθεί η ακρίβεια και η αποτελεσματικότητά τους.

Τα ΚΜΜ ανήκουν σε μια γενικότερη κλάση μοντέλων που αναφέρονται ως προβλήματα ελλιπών δεδομένων. Στο πλαίσιο της κλασικής στατιστικής, ο αλγόριθμος Expectation-Maximisation (ΕΜ) προσεγγίζει την εκτιμήτρια μέγιστης πιθανοφάνειας (ΕΜΠ) του διανύσματος των παραμέτρων, ενώ, στο πλαίσιο της Μπεϋζιανής στατιστικής, οι μέθοδοι Markov Chain Monte Carlo (MCMC) και συγκεκριμένα ο πλήρως δεσμευμένος δειγματολήπτης Gibbs, εφαρμόζονται για την προσέγγιση της εκ των υστέρων κατανομής του διανύσματος των παραμέτρων. Αυτές οι μέθοδοι εφαρμόζονται πρώτα για την εκτίμηση παραμέτρων σε μοντέλα πεπερασμένων μίξεων κατανομών, τα οποία μπορούν να θεωρηθούν ως ειδικές περιπτώσεις ΚΜΜ, όπου δεν επιτρέπεται καμία εξάρτηση μεταξύ διαδοχικών παρατηρήσεων.

Στην περίπτωση των ΚΜΜ μια μορφή αμφίδρομης αναδρομικής διαδικασίας είναι επιπλέον απαραίτητη για τον υπολογισμό της δεσμευμένης κατανομής των κρυφών μεταβλητών, δεδομένων των παρατηρήσεων. Αυτός ο αλγόριθμος Forward-Backward μπορεί να συνδυαστεί είτε με τον αλγόριθμο ΕΜ είτε με

κάποια μέθοδο MCMC για εκτίμηση παραμέτρων. Τέλος, εξετάζουμε μεθόδους για επιλογή του αριθμού των κρυφών καταστάσεων σε ένα ΚΜΜ. Η κλασική προσέγγιση συνήθως περιλαμβάνει την προσέγγιση των γενικευμένων λόγων πιθανοφανειών μέσω κάποιας τεχνικής bootstrap, ενώ η Μπεϋζιανή προσέγγιση στηρίζεται είτε σε δια-διαστατικές μεθόδους MCMC, οι οποίες περιλαμβάνουν κινήσεις μεταξύ διαφορετικών μοντέλων μαζί με εκτίμηση παραμέτρων, είτε σε μεθόδους προσομοίωσης για την προσέγγιση των περιθωρίων πιθανοφανειών των συγκρινόμενων μοντέλων.

Acknowledgements

It is difficult to overstate my gratitude to my thesis advisor, Asst. Prof. Loukia Meligkotsidou, first and foremost for introducing me to Bayesian statistics, but also for always being able to provide me with a fresh prospect on statistics in general. As far as this thesis is concerned, she consistently allowed it to be my own work and only steered me in the right direction whenever she deemed it necessary.

I also wish to acknowledge the help provided by the rest of my thesis committee, Prof. Antonis Economou and Lecturer Samis Trevezas. Dr. Antonis Economou was my first-year combinatorics and probability professor during my undergraduate studies. His teaching style and enthusiasm made a strong impression on me and I have always carried positive memories of his classes with me, eventually leading me to pursue the M.Sc. in Statistics and Operations Research. Dr. Samis Trevezas was always able to give a more formal perspective upon my field of study and challenged me to continually strive to better myself through his unwavering belief in me. Additionally, even though I have not had the opportunity to work with Prof. Apostolos Burnetas on this project, his work has had a profound impact on my academic studies.

Getting through my thesis required more than academic support and I have many people to thank for listening to me and, at times, having to tolerate me over the past year. I am particularly grateful to my friends for accepting nothing less than excellence from me. Particularly my fellow students Panos Andreou and Thodoris Rousounelos have been steadfast in their personal and professional support during the time I spent studying for my master's degree.

Most importantly, none of this could have happened without my parents and particularly my father, who was the first person to introduce me to the wonderful world of mathematics and who never failed to find a way to deal with my stubbornness. I must express my very profound gratitude to them for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Vasileios Katsianos

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Statistical inference and decision making hinge on the availability of data from which valuable information may be extracted. Most scientific conjectures and deductions are predominately related to the quantity and quality of information available at the time an experiment is formulated or a research is conducted.

One of the grave problems that presents itself, when considering statistical computing, is that, in reality, a part of the data which should be available to us is missing. In practice, missing data occur in a wide array of applications and for a wide variety of reasons. In a clinical study, a subject may drop out during its course or fail to follow up with the doctor responsible for the study, resulting in missing observations at subsequent time points for that particular subject. In various experiments, the data vector may become partially corrupted. In surveys, participants may decline to provide certain answers.

Obviously, the most straightforward way to handle missing data is to ignore them completely and base our inference solely on those records which have been fully observed. When there is no discernible pattern in the missing data and it can be safely assumed that data points are missing completely at random, the above can be viewed as a valid approach to handling the problem at hand. Nevertheless, in every other case, simply ignoring the parts of the data which are incomplete is not a viable alternative, as it can lead to systematically biased results in statistical modelling.

This thesis is concerned with presenting several parameter estimation and model selection methods, in the framework of both classical and Bayesian statistics, especially designed for handling missing data problems. In particular, we are going to implement these methods in order to draw inference on a special class of missing data problems, referred to as Hidden Markov Models (HMMs). For the purposes of this thesis, we are going to limit ourselves to HMMs for which the observations are made in discrete-time and the latent Markov chain has a finite state-space, namely the discrete-time, finite state-space HMMs.

The use of an unobservable sequence of states makes the model generic enough to handle a variety of complex, real-world time series, while the relatively simple Markov dependence structure still permits the use of efficient computational procedures. Our goal is to present a reasonably complete picture of statistical inference for discrete-time, finite state-space HMMs, illustrated with relevant running examples.

In Chapter 2, we present the general idea behind statistical modelling of a missing data problem. We illustrate this by explicating several known classes of missing data problems often used in statistical modelling of non-homogeneous data sets. We start by exploring the class of finite mixture models. Then, we present the basic structure of a hidden Markov model, which, as will be discussed, can be viewed as a generalisation of a finite mixture model. Lastly, we add for further consideration several generalisations of HMMs.

In Chapter 3, we develop the basic algorithm, used within the classical framework, to make inferences on finite mixture models, that is, the Expectation-Maximisation (EM) algorithm. On the other hand, in Chapter 4 we analyse the Bayesian counterpart for parameter estimation of finite mixture models, which is the Gibbs sampling algorithm with Data Augmentation.

In Chapter 5, we begin by dissecting the various methods required in the context of HMMs, in order to additionally draw inferences on the latent Markov chain. We combine these methods, with the methods explained previously in the context of finite mixture models, so as to formulate several methods for parameter estimation of HMMs, both in a classical and a Bayesian framework.

In Chapter 6, we confront the more difficult task of selecting the appropriate hidden Markov model to fit to a given data set. More precisely, within the framework of frequentist statistics, we combine the EM algorithm, utilised for parameter estimation of a hidden Markov model, with bootstrapping procedures, in order to implement the generalised likelihood ratio (LR) tests required for model selection. On the other hand, from the perspective of Bayesian statistics, we present several methods which combine model selection with parameter estimation, in order to infer the posterior probability of each of the predetermined competing models.

Lastly, in Chapter 7, we assess the adequacy of all the methods discussed for model selection and parameter estimation of HMMs, by implementing them on a time series of count data analysed originally in Nhu D. Le et al. (1992), as well as on two simulated data sets.

Chapter 2

Missing Data Problems

2.1 Introduction

In statistical modelling, often the relationship between the parameter $\boldsymbol{\theta}$ and the observed variables $\mathbf{Y}$ is defined in terms of some unobservable variables $\mathbf{X}$. In other words, the observed data $\mathbf{Y}$ is a subset of some partially observable data $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$. Models as such are called missing data problems, latent variable models or, also, models with incomplete data. In what follows, we assume that the joint distribution of $\mathbf{X}$ and $\mathbf{Y}$, for a given parameter value $\boldsymbol{\theta}$, admits a joint probability density function $f(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})$, which is also referred to as the complete-data likelihood $L(\boldsymbol{\theta}|\mathbf{x}, \mathbf{y})$. Obviously for a given value of $\mathbf{y}$ and considered as a function of $\mathbf{x}$ only, f is simply a positive integrable function and not a probability density function.

The actual likelihood of the observations, also referred to as the observed likelihood, is defined as the probability density function of $\mathbf{Y}$ and obtained by marginalisation as

$$L(\boldsymbol{\theta}|\mathbf{y}) = \int f(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) d\mathbf{x}. \tag{2.1}$$

Inference on $\boldsymbol{\theta}$ must obviously be based on the observed-data likelihood, where the missing variables have been integrated out. In practice, however, this is frequently difficult, if not impossible to do, owing to the integration involved in the above calculation. In Chapters 3 and 4, we are going to present methods designed to deal with the above issue, in both the Classical and the Bayesian framework.

2.2 Finite Mixture Models

Finite mixture models are commonly used for modelling densities which exhibit multi-modality, skewness or excess kurtosis and for making inference about classification problems. A finite mixture distribution arises in a natural way as described

below.

We assume that the population under study comprises K subgroups, with relative group sizes $p_1, p_2, \ldots, p_K$ and that interest lies in some random observation $\mathbf{Y}$, which is heterogeneous across all different subgroups. Thus, $\mathbf{Y}$ has a different probability distribution within each group, usually assumed to originate from the same parametric family $f(\mathbf{y}; \boldsymbol{\theta})$, however, with the parameter $\boldsymbol{\theta}$ differing between the groups. The groups may be labelled through a discrete indicator variable S taking values in the set $\{1, 2, \ldots, K\}$.

If we are in a position to record the group indicator S, in addition to the random variable $\mathbf{Y}$, when sampling from such a population, then conditional on having observed from group s, $\mathbf{Y}$ is a random variable following the distribution $f(\mathbf{y}; \boldsymbol{\theta}_s)$, with $\boldsymbol{\theta}_s$ being the parameter in group s. The probability of sampling from the group labelled s is, evidently, equal to p_s, therefore the joint density of $(\mathbf{y}, s)$ is given by

$$f(\mathbf{y}, s; \boldsymbol{\vartheta}) = f(s; \mathbf{p}) f(\mathbf{y}|s; \boldsymbol{\theta}) = p_s f(\mathbf{y}; \boldsymbol{\theta}_s). \tag{2.2}$$

However, in many cases it is impossible to record the group indicator S and so, we arrive at a finite mixture distribution. Indeed, having observed only the random variable $\mathbf{Y}$, the marginal density is given by the law of total probability, as the following mixture density

$$f(\mathbf{y}; \boldsymbol{\vartheta}) = \sum_{k=1}^{K} p_k f(\mathbf{y}; \boldsymbol{\theta}_k). \tag{2.3}$$

In statistical terms, the densities $f(\mathbf{y}; \boldsymbol{\theta}_k)$ are referred to as the component densities, K is called the number of components, the parameters $p_1, p_2, \ldots, p_K$ are called the component weights and the vector $\mathbf{p} = (p_1, p_2, \ldots, p_K)$ is called the weight distribution, which takes values in the unit simplex $\mathcal{E}_K$, defined by the constraints $p_k \geqslant 0$ and $p_1 + p_2 + \cdots + p_K = 1$. The mixture density function is indexed by the parameter $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K, \mathbf{p})$, taking values in the parameter space $\Theta_K = \Theta^K \times \mathcal{E}_K$.

Here, the specification is such that the likelihood of the missing variables $\mathbf{S}$ and the conditional likelihood of $\mathbf{y}$, given $\mathbf{s}$, are both known. Consequently, if the complete data $(\mathbf{y}, \mathbf{s})$ were available, then the complete-data likelihood function would be calculated as follows

$$L(\boldsymbol{\vartheta}|\mathbf{y}, \mathbf{s}) = \prod_{i=1}^{N} p_{s_i} f(\mathbf{y}_i; \boldsymbol{\theta}_{s_i}). \tag{2.4}$$

On the other hand, the mixture likelihood function $f(\mathbf{y}; \boldsymbol{\vartheta})$ of $\boldsymbol{\vartheta}$, given N random observations $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ from the above mixture density, assuming that the

data have been sampled independently and, again, that no information concerning the allocation of $\mathbf{y}_i$ to a certain component is available, takes the form

$$L(\boldsymbol{\vartheta}|\mathbf{y}) = \prod_{i=1}^{N} \left[\sum_{k=1}^{K} p_k f(\mathbf{y}_i; \boldsymbol{\theta}_k) \right], \tag{2.5}$$

which is impossible to either maximise to derive the MLE of $\boldsymbol{\vartheta}$ or to handle in a conjugate Bayesian setting for $K \geqslant 2$, due to the summation involved over all possible components.

2.3 Hidden Markov Models (HMMs)

Hidden Markov Models comprise another well-known category of latent variable models, where the values of a stochastic process, observed at discrete times, depend on some unobserved mechanism. This mechanism is nothing else but a Markov chain, denoted by $\{\mathbf{X}_k\}_{k\geqslant 0}$, which is not available to the observer. This hidden Markov chain governs the distribution of the observable process $\{\mathbf{Y}_k\}_{k\geqslant 0}$, that is to say, the distribution of Y_k is defined through the value of $\mathbf{X}_k$. For instance, $\mathbf{Y}_k$ may have a Gaussian distribution, whose mean and variance is determined by $\mathbf{X}_k$.

More formally, an HMM is a bivariate discrete-time process $\{\mathbf{X}_k, \mathbf{Y}_k\}_{k\geqslant 0}$, where $\{\mathbf{X}_k\}_{k\geqslant 0}$ is a Markov chain, $\{\mathbf{Y}_k\}_{k\geqslant 0}$ is a sequence of independent random variables conditional on $\{\mathbf{X}_k\}_{k\geqslant 0}$ and the conditional distribution of $\mathbf{Y}_k$ depends solely on $\mathbf{X}_k$. The underlying Markov chain is sometimes called the regime or state process. It is clear at this point that all statistical inference on the parameters of the model, even on the Markov chain itself, must be done in terms of $\{\mathbf{Y}_k\}_{k\geqslant 0}$ only. Figure 2.1 summarises the dependence structure of an HMM.

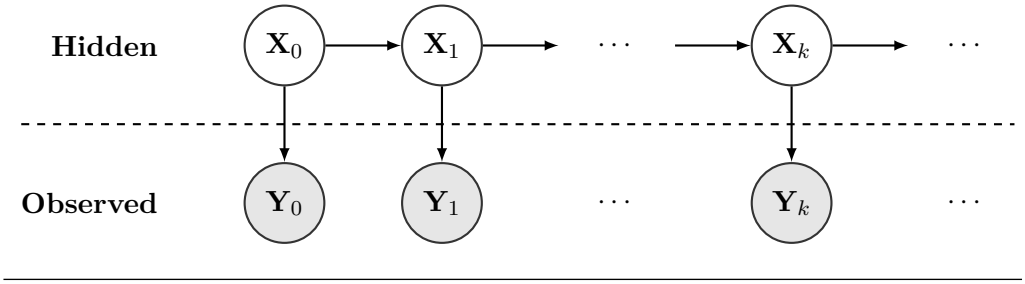

FIGURE 2.1: Hidden Markov Model - Graphical Representation of the Dependence Structure

More precisely, the underlying process obviously has the Markov property. The distribution of $\mathbf{X}_{k+1}$ conditional on the past values of the chain, $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_k$, is determined by $\mathbf{X}_k$ only. Likewise, the distribution of $\mathbf{Y}_k$ conditional on the past observations $\mathbf{Y}_0, \mathbf{Y}_1, \ldots, \mathbf{Y}_{k-1}$ and the history of the chain $\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_k$ depends

solely on the value of $\mathbf{X}_k$. However, even though the variables $\mathbf{Y}_k$ are conditionally independent given $\{\mathbf{X}_k\}_{k\geqslant 0}$, $\{\mathbf{Y}_k\}_{k\geqslant 0}$ is not an independent sequence, due to the dependence on $\{\mathbf{X}_k\}_{k\geqslant 0}$.

The Markov chain is often assumed to be homogeneous and to take values in a finite set, identified by $S = \{1, 2, \ldots, r\}$. Hence, $\{X_k\}_{k\geqslant 0}$ comes to be a homogeneous, discrete-time Markov chain on a finite state-space, with transition probability matrix $\mathbf{P} = [p_{ij}]$, where $p_{ij} = P(X_{k+1} = j | X_k = i)$ for $i, j = 1, 2, \ldots, r$, initial distribution $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_r)$ and, if also ergodic, unique stationary distribution $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_r)$.

A notable example of this is a Normal HMM, where $(Y_k | X_k = i) \sim \mathcal{N}(\mu_i, \sigma_i^2)$. In this model, the conditional distributions of Y_k, given X_k, all belong to the same parametric family, with parameters governed, as mentioned above, by X_k. One may remark at this point, that the marginal distribution of Y_k is a mixture of r Gaussian distributions, much like the case of a finite mixture of Normal distributions. Thus, one way to view HMMs is as an extension of independent mixture models, in which we allow for dependence between subsequent observations through a latent Markov chain.

For the sake of conciseness, in what follows, we will utilise the notation $\mathbf{Y}_{l:m}$ to denote the collection of consecutively indexed variables $\mathbf{Y}_l, \mathbf{Y}_{l+1}, \ldots, \mathbf{Y}_m$. We also denote by f_i the conditional distribution of $\mathbf{Y}_k$ given that $X_k = i$, which we assume to be parametrised by $\boldsymbol{\theta}_i$, and by $\boldsymbol{\vartheta} = (\mathbf{P}, \boldsymbol{\nu}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_r)$ the vector of all unknown parameters. Then, the complete-data likelihood of this model is given as

$$L(\boldsymbol{\vartheta} | \mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = \nu(x_0; \mathbf{P}) \cdot \prod_{k=0}^{n-1} p(x_k, x_{k+1}) \cdot \prod_{k=0}^{n} f(\mathbf{y}_k; \boldsymbol{\theta}_{x_k}). \tag{2.6}$$

However, compared to finite mixture models, where computation of the observed likelihood through the complete-data likelihood simply involves adding over all possible components for each observation $\mathbf{y}_i$, the dependence structure of HMMs leads to some additional computational complexity, owing to the fact that $\{\mathbf{Y}_k\}_{k\geqslant 0}$ is not an independent sequence. Using the chain rule and then applying the law of total probability to condition on every possible state for each observation $\mathbf{y}_i$, we obtain the following expression for the observed likelihood

$$L_n(\boldsymbol{\vartheta}) = \prod_{k=0}^{n} f(\mathbf{y}_k | \mathbf{y}_{0:k-1}; \boldsymbol{\vartheta}) = \prod_{k=0}^{n} \left[\sum_{i=1}^{r} P(X_k = i | \mathbf{y}_{0:k-1}; \mathbf{P}, \boldsymbol{\nu}) f(\mathbf{y}_k; \boldsymbol{\theta}_i) \right]. \tag{2.7}$$

Calculation of the probabilities $P(X_k = i | \mathbf{y}_{0:k-1}; \mathbf{P}, \boldsymbol{\nu})$, involved in the above expression, is not straightforward, but requires some form of forward recursion, which will

be discussed at length in Chapter 5.

Another way to view HMMs is, as an extension of Markov chains, where the observation Y_k of the state X_k, which is itself unavailable to the observer, is distorted through some additional, independent source of randomness. Returning to the previous example, we can verify this notion, by writing the model as $Y_k = \mu_{X_k} + \sigma_{X_k} V_k$, where $\{V_k\}_{k \geqslant 0}$ is an i.i.d. sequence of $\mathcal{N}(0,1)$ variables. Moreover, we can easily obtain a similar functional representation for the Markov chain

$$X_{k+1} = \min \left\{ j = 1, 2, \ldots, r : U_k < \sum_{\ell=1}^{j} p_{X_k \ell} \right\},$$

where $\{U_k\}_{k \geqslant 0}$ is similarly defined as an i.i.d. sequence of Uniform random variables on the interval $(0, 1)$.

It is noteworthy that the example presented above is not merely a singular case, but, in great generality, any HMM can be equivalently defined through the following functional representation, known as a general state-space model

$$\mathbf{X}_{k+1} = \mathbf{a}(\mathbf{X}_k, \mathbf{U}_k), \tag{2.8}$$

$$\mathbf{Y}_k = \mathbf{b}(\mathbf{X}_k, \mathbf{V}_k), \tag{2.9}$$

where $\{\mathbf{U}_k\}_{k \geqslant 0}$ and $\{\mathbf{V}_k\}_{k \geqslant 0}$ are mutually independent i.i.d. sequences of random variables, which are independent of $\mathbf{X}_0$, and $\mathbf{a}$, $\mathbf{b}$ are measurable functions. The former is called the state or dynamics equation, while the latter is referred to as the observation equation.

Hidden Markov Models generally boast a wide array of applications, including economics, speech recognition, image analysis, biology and genetics. In some applications, the underlying process has a clear interpretation, while in others it is just a figment to represent heterogeneity.

Most examples given throughout this thesis are based on HMMs for which the underlying Markov chain is ergodic, that is to say irreducible with a unique stationary distribution. Such models can produce an infinitely long sequence of output for observation and have a typically small number of states. For this reason, inference on ergodic HMMs is usually based on a single long observed sequence of output.

However, HMMs applied in fields such as speech recognition correspond to a different class of models, referred to as left-to-right HMMs. The Markov chain in such models begins in a particular initial state and, when traversing the intermediate states, may not go backwards, toward the initial state, but only forwards, toward the

final state. Because they generally have a considerable number of states, compared to ergodic HMMs, inference on them requires many independent sequences of output.

Example 2.1 ***(Stochastic Volatility Models)*** *Let S_k be the price of a financial asset, such as a share price or stock index, at time k. Instead of the prices, it is more customary to consider the relative returns, $\frac{S_k - S_{k-1}}{S_{k-1}}$, or the log-returns, $\log\left(\frac{S_k}{S_{k-1}}\right)$, which both describe the relative change over time of the price process.*

Data from financial markets clearly indicate that the distribution of returns usually has tails which are heavier than those of a Normal distribution. In addition, even though the returns are approximately uncorrelated over times, they are not independent. Lastly, the variance of returns tends to change over time. Large changes tend to be followed by large changes and small changes tend to be followed by small changes, a phenomenon often referred to as volatility clustering.

Most models assume that the process $\{\sigma_k\}_{k\geqslant 0}$, where σ_k represents the volatility (standard deviation) of the returns at time k, is a function of past values. The simplest model assumes that σ_k is a function of the squares of previous observations. This leads to the celebrated Autoregressive Conditional Heteroscedasticity (ARCH) model developed by Engle (1982).

An alternative to the ARCH/GARCH framework is a model in which the variance is specified to follow some latent stochastic process. Such models are referred to as Stochastic Volatility (SV) models. In contrast to GARCH models, there is no direct dependence on past returns. The canonical SV model for discrete-time data is

$$X_{k+1} = \phi X_k + \sigma U_k, \quad U_k \sim \mathcal{N}(0,1), \tag{2.10}$$

$$Y_k = \beta \exp\left\{\frac{X_k}{2}\right\} V_k, \quad V_k \sim \mathcal{N}(0,1), \tag{2.11}$$

where the observations $\{Y_k\}_{k\geqslant 0}$ are the log-returns, $\{X_k\}_{k\geqslant 0}$ is the log-volatility process, which is assumed to follow a stationary $AR(1)$ model, and $\{U_k\}_{k\geqslant 0}$, $\{V_k\}_{k\geqslant 0}$ are i.i.d. sequences. ■

2.3.1 Gaussian Linear State-Space Models (GLSSMs)

As a case of Hidden Markov Models with non-finite state-space, we consider the following general state-space model

$$\mathbf{X}_{k+1} = \mathbf{A}\mathbf{X}_k + \mathbf{R}\mathbf{U}_k, \tag{2.12}$$

$$\mathbf{Y}_k = \mathbf{B}\mathbf{X}_k + \mathbf{S}\mathbf{V}_k, \tag{2.13}$$

where

- $\{\mathbf{U}_k\}_{k\geqslant 0}$, $\{\mathbf{V}_k\}_{k\geqslant 0}$ are i.i.d. sequences of multivariate Normal random variables. The former is called the state or process noise, while the latter is referred to as the measurement noise.
- $\mathbf{A}$ is the state transition matrix, $\mathbf{B}$ is the measurement transition matrix, $\mathbf{R}$ is the square-root of the state noise covariance and $\mathbf{S}$ is the square-root of the measurement noise covariance.
- $\mathbf{X}_0$ is Gaussian with mean vector $\boldsymbol{\mu}_\mathbf{a}$ and covariance matrix $\boldsymbol{\Sigma}_\mathbf{a}$ and is independent of the processes $\{\mathbf{U}_k\}_{k\geqslant 0}$, $\{\mathbf{V}_k\}_{k\geqslant 0}$.

The above model, is very popular both in engineering and in time series literature. In addition to its practical importance, this model is a rare example of a non-finite state-space HMM for which exact and reasonably efficient numerical procedures are available to compute the distribution of the underlying chain given the observations.

Example 2.2 ***(Noisy Autoregressive Process)*** *We define a p^{th} order autoregressive process $\{Z_k\}_{k\geqslant 0}$ through the equation*

$$Z_{k+1} = \phi_1 Z_k + \cdots + \phi_p Z_{k-p+1} + U_k,$$

where $\{U_k\}_{k\geqslant 0}$ is standard Gaussian white noise, i.e. $U_k \sim \mathcal{N}(0,1)$ independent for $k \geqslant 0$. We also assume that the autoregressive process is observable only through $Y_k = Z_k + SV_k$, where $\{V_k\}_{k\geqslant 0}$ is the measurement noise and S the square-root of the corresponding covariance.

Define the lag-vector $\boldsymbol{X}_k = (Z_k, \ldots, Z_{k-p+1})^T$, $\boldsymbol{R} = (1, 0, \ldots, 0)^T \in \mathbb{R}^{p\times 1}$ and the so-called companion matrix of the autoregressive coefficients $\boldsymbol{\phi} = (\phi_1, \phi_2, \ldots, \phi_d)$

$$\boldsymbol{A} = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_{p-1} & \phi_p \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Then, the noisy autoregressive process can be equivalently written in state-space form

$$\boldsymbol{X}_{k+1} = \boldsymbol{A}\boldsymbol{X}_k + \boldsymbol{R}U_k, \tag{2.14}$$

$$Y_k = \boldsymbol{R}^T \boldsymbol{X}_k + SV_k. \tag{2.15}$$

It is sensible to assume that the state and measurement noises are independent, so the above noisy auto-regression corresponds to the form of a general Gaussian linear state-space model. ■

2.3.2 Conditionally Gaussian Linear State-Space Models (CGLSSMs)

In the case of conditionally Gaussian linear state-space models, the state $\mathbf{X}_k$ is comprised of two components, $\mathbf{C}_k$ and $\mathbf{W}_k$, where the former is finite-valued, whereas the latter is a continuous, possibly vector-valued, variable. We generally refer to the variables $\mathbf{C}_k$ as the indicator or latent variables and to $\mathbf{W}_k$ as the state variables.

The term "conditionally Gaussian linear state-space models" signifies the fact that, when conditioned on the finite-valued process $\{\mathbf{C}_k\}_{k\geqslant 0}$, the CGLSSM reduces to a GLSSM. It is also common to refer to such models as jump Markov models, where the term jump refers to the instants k, at which the value of $\mathbf{C}_k$ differs from that of $\mathbf{C}_{k-1}$.

CGLSSMs belong to a class of models, referred to as hierarchical hidden Markov models. This means that the variable $\mathbf{C}_k$, which is the highest in the hierarchy, influences not only the value of $\mathbf{Y}_k$, but also the transition from $\mathbf{W}_{k-1}$ to $\mathbf{W}_k$, as demonstrated in Figure 2.2.

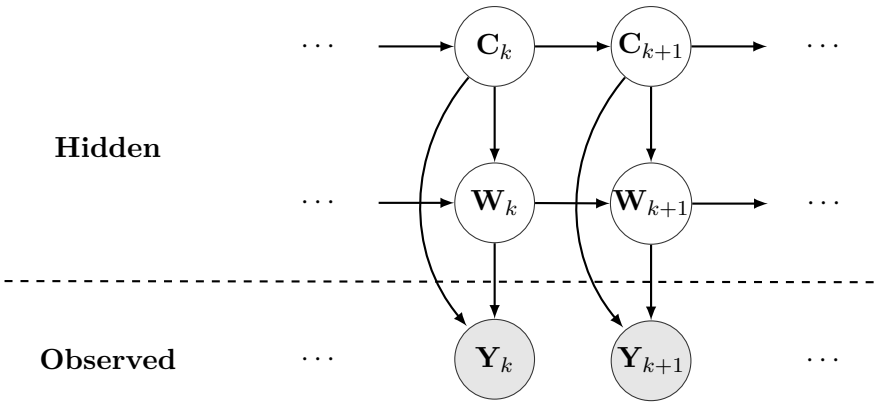

FIGURE 2.2: Conditionally Gaussian Linear State-Space Model - Graphical Representation of the Dependence Structure

It is often advantageous to treat the intermediate sequence $\{\mathbf{W}_k\}_{k\geqslant 0}$ as a nuisance parameter, which means integrating out the influence of $\{\mathbf{W}_k\}_{k\geqslant 0}$, given $\{\mathbf{C}_k\}_{k\geqslant 0}$.

Example 2.3 ***(Change Point Detection)*** *In a GLSSM, the dynamics of the state depend on the state transition matrix and on the noise covariance. These quantities may change over time and, if they do so randomly and at unknown time points, then the problem that arises is referred to as a change point problem.*

In the simplest change point problems, the state variable represents the level of a quantity of interest, which is modelled as a step function. To model this situation, we put $C_k = 0$, if there is no change point at time index k and $C_k = 1$, if a jump has occurred at time index k. The resulting state-space model is

$$\boldsymbol{W}_{k+1} = \boldsymbol{A}_{C_{k+1}} \boldsymbol{W}_k + \boldsymbol{R}_{C_{k+1}} \boldsymbol{U}_k, \tag{2.16}$$

$$\boldsymbol{Y}_k = \boldsymbol{B}\boldsymbol{W}_k + \boldsymbol{S}\boldsymbol{V}_k, \tag{2.17}$$

where $\boldsymbol{A}_0 = \boldsymbol{I}$, $\boldsymbol{R}_0 = \boldsymbol{0}$, $\boldsymbol{A}_1 = \boldsymbol{0}$ and $\boldsymbol{R}_1 = \boldsymbol{R}$.

The simplest way to visualise this is assuming that $\{C_k\}_{k\geqslant 0}$ is an i.i.d. sequence of Bernoulli(p) random variables. The time between two subsequent change points is, thus, distributed as a Geometric random variable with mean value $1/p$, while

$$\boldsymbol{W}_{k+1} = \begin{cases} \boldsymbol{W}_k & \text{with probability } p \\ \boldsymbol{U}_k & \text{with probability } 1-p \end{cases}.$$

It is also possible to allow for a more general distribution of the time between consequent jumps by introducing some form of dependence among the variables C_k.■

Example 2.4 ***(Observational Outliers and Heavy-Tailed Noise)*** *Another interesting application of CGLSSMs pertains to the field of robust statistics. In the course of model building, statisticians are often confronted with the presence of outliers. Routinely ignoring outlying observations is obviously not statistically sound, as they may contain valuable information about measurement errors, system characteristics that have been left out of the model and so forth.*

For example, if we confirm the presence of outliers in the data, we can account for them, by utilising the following model

$$W_{k+1} = A_{C_{k+1,1}} W_k + R_{C_{k+1,1}} U_k, \quad U_k \sim \mathcal{N}(0,1), \tag{2.18}$$

$$Y_k = \mu_{C_{k,2}} + B_{C_{k,2}} W_k + S_{C_{k,2}} V_k, \quad V_k \sim \mathcal{N}(0,1), \tag{2.19}$$

where $C_{k,1}, C_{k,2} \in \{0,1\}$ are indicators of a change point and of the presence of outliers respectively.

Similarly to the previous example, we set $A_0 = 1, R_0 = 0, A_1 = 0$ and $R_1 = \sigma_U$. When there is no outlier, we assume that the level is observed in additive Gaussian noise, therefore $\mu_0 = 0, B_0 = 1$ and $S_0 = \sigma_{V,0}$. In the presence of an outlier, however, the measurement does no longer carry information about the current value of the

level, that is $B_1 = 0$, and the measurement noise is assumed to follow a Gaussian distribution with negative mean $\mu_1 = \mu$ and large standard deviation $S_1 = \sigma_{V,1}$.

The simplest model for $C_{k,2}$ would be a Bernoulli model, in which we would include information about the ratio of outliers/non-outliers in the success probability. ■

2.4 Markov Switching Models

Perhaps the most significant generalisation of HMMs are the so-called Markov Switching Models. In these models, the conditional distribution of $\mathbf{Y}_{k+1}$, given all past variables, depends not only on $\mathbf{X}_{k+1}$, but also on $\mathbf{Y}_k$. Hence, conditional on the state process $\{\mathbf{X}_k\}_{k\geqslant 0}$, the sequence $\{\mathbf{Y}_k\}_{k\geqslant 0}$ forms a non-homogeneous Markov process. Figure 2.3 summarises the dependence structure of a Markov Switching Model.

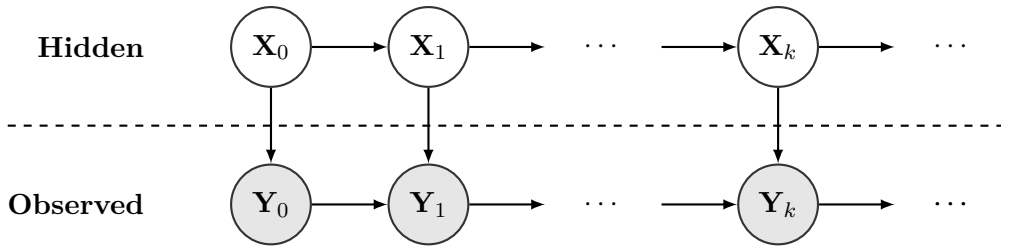

FIGURE 2.3: Markov Switching Model - Graphical Representation of the Dependence Structure

In state space form, a Markov Switching Model may be given as

$$\mathbf{X}_{k+1} = \mathbf{a}(\mathbf{X}_k, \mathbf{U}_k), \tag{2.20}$$

$$\mathbf{Y}_{k+1} = \mathbf{b}(\mathbf{X}_{k+1}, \mathbf{Y}_k, \mathbf{V}_k). \tag{2.21}$$

Markov switching models have much in common with basic HMMs and virtually identical computational machinery may be used for both classes of models. However, their statistical analysis is much more intricate, owing to the fact that the properties of the observed process are not fully controlled by the underlying process. Specifically, $\{\mathbf{Y}_k\}_{k\geqslant 0}$ is an infinite memory process, whose dependence structure may be even stronger than that of the hidden Markov chain.

Example 2.5 ***(Switching Linear Auto-regression)*** *Similarly to the noisy autoregressive process, a switching linear auto-regression is a model of the form*

$$Y_k = \mu_{X_k} + \sum_{i=1}^{d} a_{i,X_k}(Y_{k-i} - \mu_{X_{k-i}}) + \sigma_{X_k} V_k, \quad k \geqslant 1, \tag{2.22}$$

where $\{X_k\}_{k\geqslant 0}$ is a finite state-space Markov chain, $\{V_k\}_{k\geqslant 0}$ is white noise independent of $\{X_k\}_{k\geqslant 0}$ and the functions μ, a_i, σ describe the dependence of the parameters upon the current state X_k.

Obviously, the conditional distribution of Y_k does not only depend on X_k and Y_{k-1}, but also on other lagged variables. By stacking them in groups of d elements, we can obtain a process, whose conditional distribution depends on just one lagged variable at a time. Define the companion matrix A_i associated with the autoregressive coefficients of state i

$$\boldsymbol{A}_i = \begin{bmatrix} a_{1,i} & a_{2,i} & \cdots & a_{d-1,i} & a_{d,i} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

and the vectors

$$\boldsymbol{Y}_k = (Y_k, Y_{k-1}, \ldots, Y_{k-d+1})^T,$$

$$\boldsymbol{X}_k = (X_k, X_{k-1}, \ldots, X_{k-d+1})^T,$$

$$\boldsymbol{\mu}_{\boldsymbol{X}_k} = (\mu_{X_k}, \mu_{X_{k-1}}, \ldots, \mu_{X_{k-d+1}})^T,$$

$$\boldsymbol{V}_k = (V_k, 0, \ldots, 0)^T.$$

Hence, the stacked observation vector $\boldsymbol{Y}_k$ satisfies the equation

$$\boldsymbol{Y}_k = \boldsymbol{\mu}_{X_k} + \boldsymbol{A}_{X_k}(\boldsymbol{Y}_{k-1} - \boldsymbol{\mu}_{\boldsymbol{X}_{k-1}}) + \sigma_{X_k}\boldsymbol{V}_k. \tag{2.23}$$

This class of auto-regression models finds many uses in econometrics, as it provides a formal statistical representation of the idea that expansion and contraction constitute two distinct economic phases. ■

Chapter 3

Maximum Likelihood Estimation for Missing Data Problems

3.1 Introduction

As is always the case in statistics, the models described in Chapter 2 cannot be fully specified beforehand, in most situations, and so, some of their parameters have to be estimated based on observed data. Except for very simplistic instances, the structures of these models are sufficiently complex to prevent the use of direct estimation methods, hence, exact inference is not possible. Instead, we need to resort to computationally intensive methods to deal with such models.

Specifically, in classical statistics, we cannot rely on estimators provided by moment or least squares methods, but, instead, need to focus on ways to estimate the MLE of the parameter under study. For many of these models, the likelihood function is known or, at least, can be numerically approximated, thus, our main objective is to optimise a possibly quite complex function utilising numerical optimisation methods.

3.2 The Expectation-Maximisation (EM) Algorithm

Under the prism of classical statistics, the task under consideration for missing data problems is the maximisation of the observed likelihood with respect to the parameter $\boldsymbol{\vartheta}$. The most popular method for solving this optimisation problem is the Expectation-Maximisation (EM) algorithm, which is a deterministic algorithm designed to find maximum likelihood estimates in models where there is incomplete data. In many situations, the EM algorithm is very easy to implement from scratch, thus making it a more attractive option than gradient-based methods, such as the Newton-Raphson or Fisher Scoring algorithms.

A first step towards implementing the EM algorithm is defining the following probability density function

$$f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}) = \frac{f(\mathbf{x},\mathbf{y};\boldsymbol{\theta})}{f(\mathbf{y};\boldsymbol{\theta})} = \frac{L(\boldsymbol{\theta}|\mathbf{x},\mathbf{y})}{L(\boldsymbol{\theta}|\mathbf{y})}, \tag{3.1}$$

which, in the case of missing data problems, can be easily identified as the conditional probability density function of X given Y. In what follows, we will also denote by $\ell(\theta)$ the logarithm of the likelihood function $L(\theta)$.

Next, we introduce the so-called intermediate quantity of EM as a function indexed by two separate parameter vectors, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, and defined by

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}') = \int \ell(\boldsymbol{\theta}|\mathbf{x},\mathbf{y}) f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}') d\mathbf{x} = E_{\boldsymbol{\theta}'}[\ell(\boldsymbol{\theta}|\mathbf{X},\mathbf{y})|\mathbf{y}]. \tag{3.2}$$

In order to realise exactly how the intermediate quantity of EM relates to the observed likelihood function, which is the target of our optimisation process, we also need to define the following quantity

$$\mathcal{H}(\boldsymbol{\theta};\boldsymbol{\theta}') = -E_{\boldsymbol{\theta}'}[\log f(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})|\mathbf{y}] = -\int \log f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}) f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}') d\mathbf{x}, \tag{3.3}$$

where $\mathcal{H}(\boldsymbol{\theta}';\boldsymbol{\theta}')$ can be recognised as the entropy of the probability density function $f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}')$. Now, utilising equation (3.1), one may rewrite the observed log-likelihood as follows

$$\ell(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}|\mathbf{y}) = Q(\boldsymbol{\theta};\boldsymbol{\theta}') + \mathcal{H}(\boldsymbol{\theta};\boldsymbol{\theta}'). \tag{3.4}$$

Whereas, in most cases, the intermediate quantity of EM can be calculated in closed form or at least approximated through some form of Monte Carlo simulation, the function $\mathcal{H}(\boldsymbol{\theta};\boldsymbol{\theta}')$ is generally intractable. The quantity $\mathcal{H}(\boldsymbol{\theta};\boldsymbol{\theta}') - \mathcal{H}(\boldsymbol{\theta}';\boldsymbol{\theta}')$ can be recognised as the relative entropy between the probability density function $f(\mathbf{x}|\mathbf{y})$, indexed by $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ respectively. Utilising the fact that the logarithm function is concave along with Jensen's inequality yields us the following result

$$\begin{aligned} \mathcal{H}(\boldsymbol{\theta};\boldsymbol{\theta}') - \mathcal{H}(\boldsymbol{\theta}';\boldsymbol{\theta}') &= -E_{\boldsymbol{\theta}'}\left[\log \frac{f(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})}{f(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}')}\middle|\mathbf{y}\right] \\ &\geqslant -\log E_{\boldsymbol{\theta}'}\left[\frac{f(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})}{f(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}')}\middle|\mathbf{y}\right] \\ &= -\log \int f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}) d\mathbf{x} = 0, \end{aligned} \tag{3.5}$$

which is widely known as the Fundamental Inequality of EM.

The strength of the EM algorithm lies on the fact that the term $\mathcal{H}(\boldsymbol{\theta};\boldsymbol{\theta}')$ can be completely ignored during the optimisation process, while the intermediate quantity

of EM acts as a surrogate for the observed log-likelihood. Indeed, in view of the fundamental inequality of EM, if $\boldsymbol{\theta}'$ represents the current estimate of the MLE, then simply choosing any value of $\boldsymbol{\theta}$, such that $Q(\boldsymbol{\theta};\boldsymbol{\theta}')$ is increased over its baseline $Q(\boldsymbol{\theta}';\boldsymbol{\theta}')$, guarantees an increase to the observed log-likelihood $\ell(\boldsymbol{\theta})$.

The EM algorithm takes its name by the iteration of its two main steps: the E-step, where the conditional expectation of the complete-data log-likelihood given $\mathbf{y}$ is calculated, and the M-step, where its maximisation is performed. It consists of iteratively building a sequence $\left\{\boldsymbol{\theta}^{(\ell)}\right\}_{\ell\geqslant 1}$ of parameter estimates given an initial guess $\boldsymbol{\theta}^{(0)}$. More precisely, each iteration is performed as follows

E-step: Calculate the intermediate quantity $Q\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(\ell)}\right)$.

M-step: Choose $\boldsymbol{\theta}^{(\ell+1)}$ to be any value of $\boldsymbol{\theta}\in\Theta$, that maximises $Q\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(\ell)}\right)$.

It is not obvious at this point that the M-step of the algorithm may in practice be easier to perform than the direct maximisation of the function of interest, which is the observed log-likelihood. However, it is easy to see the decisive argument behind the algorithm. By the very definition of the sequence $\left\{\boldsymbol{\theta}^{(\ell)}\right\}_{\ell\geqslant 1}$, the corresponding sequence $\left\{\ell\left(\boldsymbol{\theta}^{(\ell)}\right)\right\}_{\ell\geqslant 1}$ is non-decreasing, hence EM is a monotone optimisation algorithm.

Moreover, if the algorithm ever converges and the iterations stop at a point $\boldsymbol{\theta}^*$, then that is with certainty a stationary point of the likelihood function, although not necessarily a global maximum. To determine whether the algorithm has found a global, rather than simply a local, maximum it is recommended to use a set of different starting values scattered around the state-space of $\boldsymbol{\theta}$.

The rate of convergence depends on what is known as the fraction of missing information. Informally, this fraction measures the amount of information about $\boldsymbol{\theta}$ which is lost by failing to observe X. Thus, if the complete data is much more informative about $\boldsymbol{\theta}$ than the observed data, then, loosely speaking, the fraction of missing information is large and the EM algorithm converges slowly. Several methods have been proposed in order to accelerate the convergence of the algorithm. Many of them are based on incorporating information about the gradient of the likelihood function into the algorithm.

EM in Exponential Families

In the context of HMMs, the main limitation of the EM algorithm appears in cases where the E-step is not feasible, especially in models for which the state-space is not finite. The basic EM algorithm, as described in the previous section, will generally only be helpful in situations where the following conditions hold

E-step: It is possible to calculate the intermediate quantity $Q(\boldsymbol{\theta};\boldsymbol{\theta}')$ given a value of $\boldsymbol{\theta}'$ at a reasonable computational cost.

M-step: Considered as a function of $\boldsymbol{\theta}$, $Q(\boldsymbol{\theta};\boldsymbol{\theta}')$ is sufficiently simple, to allow for closed-form maximisation.

A rather general context in which both of these requirements are satisfied is when the joint probability density function of the observed and the latent variables belongs to the exponential family of distributions, that is, if it can be expressed in the form

$$f(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) = \exp\left\{[\psi(\boldsymbol{\theta})]^T T(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) - c(\boldsymbol{\theta})\right\} h(\mathbf{x},\mathbf{y}), \tag{3.6}$$

where $T(\mathbf{x},\mathbf{y};\boldsymbol{\theta})$ is referred to as the vector of natural sufficient statistics and $\psi(\boldsymbol{\theta})$ as the natural parametrisation. In this particular case, the intermediate quantity of EM reduces to

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}') = [\psi(\boldsymbol{\theta})]^T \int T(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}')d\mathbf{x} - c(\boldsymbol{\theta}) + \int \log h(\mathbf{x},\mathbf{y}) f(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}')d\mathbf{x}. \tag{3.7}$$

Note that the right-most term does not depend on $\boldsymbol{\theta}$ and so, it might as well be ignored in the maximisation process. Apart from this term, the intermediate quantity takes an explicit form as long as it is possible to evaluate the conditional expectation of the vector of sufficient statistics, given $\mathbf{y}$. The EM algorithm, thus, reduces to

E-step: Calculate $Q\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(\ell)}\right) = E_{\boldsymbol{\theta}^{(\ell)}}[T(\mathbf{X},\mathbf{y};\boldsymbol{\theta})|\mathbf{y}]$.

M-step: Maximise $[\psi(\boldsymbol{\theta})]^T Q\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(\ell)}\right) - c(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta} \in \Theta$, in order to obtain the updated estimate $\boldsymbol{\theta}^{(\ell+1)}$.

3.3 Application to Finite Mixture Models

With the widespread availability of powerful computers and elaborate numerical algorithms, maximum likelihood estimation became the preferred approach to parameter estimation for finite mixture models for many decades. In early papers, the maximum likelihood estimator $\hat{\boldsymbol{\vartheta}}$ is obtained by maximising the mixture likelihood function $f(\mathbf{y}|\boldsymbol{\vartheta})$ with respect to $\boldsymbol{\vartheta}$, using some direct maximisation method such as the Newton-Rhaphson algorithm. Nowadays, the EM algorithm is the most commonly applied method to find the maximum likelihood estimator for a finite mixture model.

To implement the EM algorithm for the estimation of the parameters of a finite mixture model, we first need to rewrite the complete-data likelihood function, given in equation (2.4), in a way that makes it more convenient for us, to calculate the

intermediate quantity of EM. More precisely, we write

$$L(\boldsymbol{\vartheta}|\mathbf{y},\mathbf{s}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\left[p_k f(\mathbf{y}_i;\boldsymbol{\theta}_k)\right]^{\mathbb{1}\{s_i=k\}}. \tag{3.8}$$

Taking the logarithm of the above function and calculating the conditional expectation given $\mathbf{y}$, yields the following result for the intermediate quantity

$$Q\left(\boldsymbol{\vartheta};\boldsymbol{\vartheta}^{(\ell)}\right) = \sum_{i=1}^{N}\sum_{k=1}^{K} P\left(S_i = k \,\middle|\mathbf{y}_i;\boldsymbol{\vartheta}^{(\ell)}\right)\left[\log p_k + \log f(\mathbf{y}_i;\boldsymbol{\theta}_k)\right]. \tag{3.9}$$

As it turns out, in order to carry out the calculation required for the E-step of the algorithm, we are first required to calculate the conditional allocation probability $P\left(S_i = k \,\middle|\mathbf{y}_i;\boldsymbol{\vartheta}^{(\ell)}\right)$ of each observation $\mathbf{y}_i$ for all possible components. In what follows, we are going to utilise the notation $D_{ik}^{(\ell)}$ for these probabilities, which are calculated using Bayes' theorem and the law of total probability as follows

$$D_{ik}^{(\ell)} = P\left(S_i = k \,\middle|\mathbf{y}_i;\boldsymbol{\vartheta}^{(\ell)}\right) = \frac{p_k^{(\ell)} f\left(\mathbf{y}_i;\boldsymbol{\theta}_k^{(\ell)}\right)}{\sum_{j=1}^{K} p_j^{(\ell)} f\left(\mathbf{y}_i;\boldsymbol{\theta}_j^{(\ell)}\right)}, \qquad \begin{matrix} i = 1,2,\ldots,N, \\ k = 1,2,\ldots,K. \end{matrix} \tag{3.10}$$

Indirectly, these classification probabilities permit us to make inference on the hidden allocations of the observations, for example, by assigning each observation to the component that maximises the respective probability.

For an arbitrary mixture model, it is fairly straightforward to maximise the intermediate quantity with respect to the weight distribution. For example, introducing the Lagrange multiplier λ, which corresponds to the constraint $p_1 + p_2 + \cdots + p_K = 1$, and deriving the Lagrangian with respect to the weights, yields the following set of equations

$$p_k^{(\ell+1)} = \frac{1}{\lambda}\sum_{i=1}^{N} D_{ik}^{(\ell)}, \quad k = 1,2,\ldots,K.$$

Additionally, one may easily calculate the multiplier λ as follows

$$1 = \sum_{k=1}^{K} p_k^{(\ell+1)} = \frac{1}{\lambda}\sum_{i=1}^{N}\sum_{k=1}^{K} D_{ik}^{(\ell)} = \frac{N}{\lambda}$$

and so, we obtain the following renewed estimate of the weight distribution

$$p_k^{(\ell+1)} = \frac{1}{N}\sum_{i=1}^{N} D_{ik}^{(\ell)}, \quad k = 1,2,\ldots,K. \tag{3.11}$$

The EM algorithm for finite mixture models consists of iterating the following steps, given an initial guess $\boldsymbol{\vartheta}^{(0)}$ for the model parameters

E-step: Calculate the allocation probabilities $D_{ik}^{(\ell)}$ according to equation (3.10).

M-step: Calculate $\mathbf{p}^{(\ell+1)}$ according to (3.11) and $\boldsymbol{\theta}^{(\ell+1)} = \arg\max_{\boldsymbol{\theta}\in\Theta} Q\left(\boldsymbol{\vartheta}; \boldsymbol{\vartheta}^{(\ell)}\right)$.

The above procedure is usually repeated while the values of the intermediate quantity calculated during successive iterations are not sufficiently close. That is, the algorithm is terminated when $\ell\left(\boldsymbol{\vartheta}^{(L+1)}\right) - \ell\left(\boldsymbol{\vartheta}^{(L)}\right) < \epsilon$ holds for a certain index L, where ϵ is the predetermined precision of the estimation. Afterwards, one may compute the maximum a posteriori component allocations as $\hat{s}_i = \arg\max_{k=1,2,\ldots,K} D_{ik}^{(L)}$.

A disadvantage of the EM algorithm compared to the direct maximisation of the likelihood function is much slower convergence. Several authors use hybrid algorithms for mixture estimation, combining the EM algorithm with Newton's method.

Maximum likelihood estimation may encounter various practical difficulties. First, it may be difficult to find the global maximum of the likelihood function numerically. Several studies report convergence failures, particularly when the sample size is small or the components are not well separated. Recently, more attention has been paid to choosing starting values which increase the chance of convergence.

In particular, for finite mixture models it is common knowledge that the sample size has to be very large for the asymptotic theory of maximum likelihood estimation to apply. The regularity conditions are often violated, including cases of great practical concern, including small data sets, mixtures with small component weights and over-fitting mixtures with too many components.

3.3.1 Finite Mixtures of Poisson Distributions

Over-dispersion occurs for a Poisson random variable if the variance is greater than the mean, since the Poisson distribution is theoretically obligated to have identical mean and variance. One possible reason for over-dispersion is unobserved heterogeneity in the sample, causing the mean to be different among the observed subjects.

A commonly used model in this context is the Poisson mixture model. Applications of mixtures of Poisson distributions appear, in particular, in biology and medicine. The probability mass function of a mixture of Poisson distributions is given as

$$f(y; \boldsymbol{\vartheta}) = \sum_{k=1}^{K} p_k e^{-\lambda_k} \frac{\lambda_k^y}{y!}, \tag{3.12}$$

where $\sum_{k=1}^{K} p_k = 1$, $\lambda_k > 0$ and $0 < p_k < 1$ for $k = 1, 2, \ldots, K$.

We have already seen how the EM algorithm updates the estimates of the weight distribution at every iteration, so, now, we formulate the intermediate quantity of EM specifically for a mixture of Poisson distributions, in order to make inference on the component parameters λ_k. We have

$$Q\left(\boldsymbol{\vartheta}; \boldsymbol{\vartheta}^{(\ell)}\right) = \sum_{i=1}^{N} \sum_{k=1}^{K} D_{ik}^{(\ell)} \left[\log p_k - \lambda_k + y_i \log \lambda_k - \log(y_i!)\right]. \tag{3.13}$$

As it can be seen, deriving the intermediate quantity with respect to λ_k delivers the following set of equations

$$\sum_{i=1}^{N} D_{ik}^{(\ell)} \left(\frac{y_i}{\lambda_k} - 1\right) = 0 \quad k = 1, 2, \ldots, K.$$

Finally, solving the above K equations with respect to λ_k for $k = 1, 2, \ldots, K$ provides the following updated estimates of the parameters

$$\lambda_k^{(\ell+1)} = \frac{1}{n_k^{(\ell)}} \sum_{i=1}^{N} D_{ik}^{(\ell)} y_i, \quad k = 1, 2, \ldots, K, \tag{3.14}$$

where $n_k^{(\ell)} = \sum_{i=1}^{N} D_{ik}^{(\ell)}$.

One could notice that the above updated estimates bear a striking resemblance to the MLE we would obtain from direct maximisation of the complete-data likelihood, in the case where the allocations of each observation to a particular component were known to us. To be precise, the functions $\mathbb{1}\{\mathbf{X}_i = k\}$ in the complete-data maximum likelihood estimates are, in this case, replaced by their conditional expectations, given the actual observations and the available parameter estimate $\boldsymbol{\vartheta}^{(\ell)}$.

3.3.2 Finite Mixtures of Univariate Normal Distributions

A frequently used model for univariate continuous data displaying some kind of heterogeneity is to assume that the observations are i.i.d. realisations from a mixture of K univariate Normal distributions. The density of this distribution is given as

$$f(y; \boldsymbol{\vartheta}) = \sum_{k=1}^{K} p_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{-\frac{(y - \mu_k)^2}{2\sigma_k^2}\right\}. \tag{3.15}$$

where $\sum_{k=1}^{K} p_k = 1$, $p_k > 0$, $\mu_k \in \mathbb{R}$ and $\sigma_k^2 > 0$ for $k = 1, 2, \ldots, K$.

The intermediate quantity of EM can be easily calculated in the case of a univariate Normal mixture distribution as follows

$$Q\left(\boldsymbol{\vartheta};\boldsymbol{\vartheta}^{(\ell)}\right)=\sum_{i=1}^{N}\sum_{k=1}^{K}D_{ik}^{(\ell)}\left[\log p_k-\frac{\log(2\pi)}{2}-\frac{\log\sigma_k^2}{2}-\frac{(y_i-\mu_k)^2}{2\sigma_k^2}\right]. \tag{3.16}$$

Deriving with respect to μ_k and then with respect to σ_k^2 produces the following two sets of equations

$$\sum_{i=1}^{N}D_{ik}^{(\ell)}(y_i-\mu_k)=0,\quad k=1,2,\ldots,K,$$

$$\sum_{i=1}^{N}D_{ik}^{(\ell)}\left[\frac{(y_i-\mu_k)^2}{\sigma_k^2}-1\right]=0,\quad k=1,2,\ldots,K$$

Again, solving the above $2K$ equations with respect to μ_k and σ_k^2 respectively provides the following updated estimates of the parameters

$$\mu_k^{(\ell+1)}=\frac{1}{n_k^{(\ell)}}\sum_{i=1}^{N}D_{ik}^{(\ell)}y_i,\quad k=1,2,\ldots,K, \tag{3.17}$$

$$\left(\sigma_k^2\right)^{(\ell+1)}=\frac{1}{n_k^{(\ell)}}\sum_{i=1}^{N}D_{ik}^{(\ell)}\left(y_i-\mu_k^{(\ell+1)}\right)^2,\quad k=1,2,\ldots,K. \tag{3.18}$$

A certain difficulty with the EM algorithm is that it stops working whenever $(\sigma_k^2)^{(\ell)}$ is numerically close to zero, which happens when $D_{ik}^{(\ell)}$ is close to zero for many observations $\mathbf{y}_i$. Then, the computation of $D_{ik}^{(\ell+1)}$ at the next iteration is no longer possible. Such difficulties arise in particular if the EM algorithm is applied to a mixture of Normals over-fitting the true number of components.

A further difficulty with ML estimation for univariate mixtures of Normal distributions is that the mixture likelihood function is generally unbounded and has many spurious local modes. Thus, the ML estimator as a global maximiser of the mixture likelihood does not exist. Nevertheless, statistical theory guarantees that a particular local maximiser is consistent, efficient and asymptotically Normal, if the mixture is not over-fitting the true number of components.

3.3.3 Finite Mixtures of Multivariate Normal Distributions

Mixtures of Normals can be easily extended to deal with multivariate observations, which consist of d-dimensional vectors. In this case, the various observations typically

measure d distinct features of a unit i. Such a mixture density is given as

$$f(\mathbf{y};\boldsymbol{\vartheta}) = \sum_{k=1}^{K} p_k \left|\mathbf{2\pi\Sigma}_k\right|^{-\frac{1}{2}} \exp\left\{-\frac{(\mathbf{y}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}-\boldsymbol{\mu}_k)}{2}\right\}. \tag{3.19}$$

One has to take into consideration the fact that a multivariate mixture of Normal distributions with general variance-covariance matrices is highly parametrised in terms of $K\left[d+\frac{d(d+1)}{2}+1\right]-1$ distinct model parameters. Hence, an unconstrained multivariate mixture may turn out to be too general to handle in various situations.

Other interesting multivariate finite mixture models are obtained by putting certain constraints on the variance-covariance matrices. For example, in a homoscedastic mixture, the variance-covariance matrices are restricted to be the same in each component, whereas in a spherical mixture, we have $\boldsymbol{\Sigma}_k = \sigma_k^2\mathbf{I}_d$ for all components. Furthermore, similar issues, as in the univariate case, may present themselves, if the matrix $\boldsymbol{\Sigma}_k^{(\ell)}$ is singular, or at least nearly singular, at a certain iteration.

Now, we take the intermediate quantity of EM for a mixture of multivariate Normal distributions in the same way as before

$$Q\left(\boldsymbol{\vartheta};\boldsymbol{\vartheta}^{(\ell)}\right) = \sum_{i=1}^{N}\sum_{k=1}^{K} D_{ik}^{(\ell)}\left[\log p_k - \frac{d\log(2\pi)}{2} + \frac{\log\left|\boldsymbol{\Sigma}_k^{-1}\right|}{2} - \frac{(\mathbf{y}_i-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{y}_i-\boldsymbol{\mu}_k)}{2}\right]. \tag{3.20}$$

Derivation with respect to $\boldsymbol{\mu}_k$ and then with respect to $\boldsymbol{\Sigma}_k^{-1}$ produces the following two sets of equations

$$\boldsymbol{\Sigma}_k^{-1}\sum_{i=1}^{N} D_{ik}^{(\ell)}(\mathbf{y}_i-\boldsymbol{\mu}_k) = 0, \quad k=1,2,\ldots,K,$$

$$\sum_{i=1}^{N} D_{ik}^{(\ell)}\left[\boldsymbol{\Sigma}_k - (\mathbf{y}_i-\boldsymbol{\mu}_k)(\mathbf{y}_i-\boldsymbol{\mu}_k)^T\right] = 0, \quad k=1,2,\ldots,K.$$

And so, the updated estimates we obtain from the above sets of equations perfectly mirror the corresponding estimates given in the univariate case, i.e.

$$\boldsymbol{\mu}_k^{(\ell+1)} = \frac{1}{n_k^{(\ell)}}\sum_{i=1}^{N} D_{ik}^{(\ell)}\mathbf{y}_i, \quad k=1,2,\ldots,K, \tag{3.21}$$

$$\boldsymbol{\Sigma}_k^{(\ell+1)} = \frac{1}{n_k^{(\ell)}}\sum_{i=1}^{N} D_{ik}^{(\ell)}\left(\mathbf{y}_i-\boldsymbol{\mu}_k^{(\ell+1)}\right)\left(\mathbf{y}_i-\boldsymbol{\mu}_k^{(\ell+1)}\right)^T, \quad k=1,2,\ldots,K. \tag{3.22}$$

Chapter 4

Bayesian Inference for Missing Data Problems

4.1 Introduction

The spirit of this chapter is different, in that it covers the fully Bayesian approaches to missing data problems, which means that, besides the latent variables and their conditional, parametrised distributions, the model parameters are assigned probability distributions, called prior distributions. Moreover, inferences on these parameters are of Bayesian nature, that is, based on the posterior distributions of the parameters, given the observations.

We remark that the distinction between data and parameters is somewhat blurred in a Bayesian setting, since both are essentially treated as random variables. For this reason, inference on the parameters of the model, as well as on the latent variables, as discussed in the previous section, must be based on the respective marginal posterior distributions, given the observations $\mathbf{y}$.

In Bayesian problems where there is a single unknown parameter, the notion of conjugacy allows the effect of the data on our prior beliefs to be summarised in terms of a simple, well-known and easy to handle distribution. When dealing specifically with missing data problems in a Bayesian setting, however, three issues have to be considered.

First, in the presence of missing data, the observed likelihood function is difficult to handle, or even impossible to calculate analytically, if we want to compute the posterior distribution of the parameters. One way to solve this problem is to augment the observed data with the latent variables and work with the complete-data likelihood instead.

Secondly, in multi-parameter Bayesian problems, it is necessary to estimate the quantities of interest, such as posterior means or posterior probabilities, using a Monte

Carlo approach. However, simulating from an arbitrary high dimensional distribution is usually difficult, if not impossible, to do directly. Instead, Markov Chain Monte Carlo (MCMC) methods are used to simulate a Markov chain, whose stationary or limiting distribution is the posterior distribution of interest. These sampling algorithms are implemented in Bayesian statistics in order to provide random draws from the posterior distribution of the parameters, which can then in turn be used to approximate any quantity of interest.

Lastly, although in many problems with a single unknown parameter it is usually possible to find useful conjugate priors, in higher dimensional situations, the conjugate families end up being highly complex and difficult to summarise. However, many Bayesian problems, including several missing data problems, exhibit conditional conjugacy. That is, the conditional posterior distributions of the parameters belong to the same families of distributions as the priors. The concept of conditional conjugacy is crucial in the construction of one of the most basic forms of MCMC, the Gibbs sampler, which samples from the conditional posterior distributions of the parameters.

4.2 Gibbs Sampling with Data Augmentation

Data augmentation is a statistical technique which adds further random variables to the model. These can be viewed as data or parameters, depending on the context, but the interpretation is not really relevant for the workings of the method. In missing data problems, the hidden data $\mathbf{x}$ are naturally the additional variables to be included in the model. So, we move from the likelihood $f(\mathbf{y}|\boldsymbol{\theta})$, which is intractable, to $f(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta})$, which is easy to handle. After assigning a prior $\pi(\boldsymbol{\theta})$ to the original parameter vector $\boldsymbol{\theta}$, the joint posterior distribution of $(\boldsymbol{\theta}, \mathbf{x})$ is proportional to

$$\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}) \propto f(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) = f(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})f(\mathbf{x}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}). \tag{4.1}$$

The technique proceeds, by carrying out Gibbs sampling, to sample successively from $\boldsymbol{\theta}$ and $\mathbf{x}$ and produce a sample from this joint posterior distribution. The Gibbs sampler obtains a sample from $\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$ by successively and iteratively simulating from the conditional posterior distributions of each parameter, given the others. Under conditional conjugacy, this simulation step is usually straightforward. Given initial values $\left(\boldsymbol{\theta}^{(0)}, \mathbf{x}^{(0)}\right)$, the algorithm iterates the following steps

- Simulate $\boldsymbol{\theta}^{(\ell+1)}$ from the conditional posterior distribution $\pi\left(\boldsymbol{\theta} \left| \mathbf{x}^{(\ell)}, \mathbf{y} \right.\right)$.
- Simulate $\mathbf{x}^{(\ell+1)}$ from the conditional posterior distribution $\pi\left(\mathbf{x} \left| \boldsymbol{\theta}^{(\ell+1)}, \mathbf{y} \right.\right)$.

Of course, since both $\boldsymbol{\theta}$ and $\mathbf{x}$ are generally vector-valued quantities in missing data problems, the above steps can be broken down so that each parameter θ_i of the vector $\boldsymbol{\theta} = (\theta_1, \theta_1, \ldots, \theta_d)$ is sampled from its conditional posterior distribution, $\pi\left(\theta_i \left| \boldsymbol{\theta}_{1:i-1}^{(\ell+1)}, \boldsymbol{\theta}_{i+1:d}^{(\ell)}, \mathbf{x}^{(\ell)}, \mathbf{y}\right.\right)$, and the same applies to the latent variables $\mathbf{x}$.

Under mild regularity conditions, convergence of the Markov chain to the stationary distribution $\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$ is guaranteed. So, after a burn-in period L_0, that is, a starting number of iterations for which the draws are discarded, the subsequent draws $\left(\boldsymbol{\theta}^{(1)}, \mathbf{x}^{(1)}\right), \left(\boldsymbol{\theta}^{(2)}, \mathbf{x}^{(2)}\right), \ldots, \left(\boldsymbol{\theta}^{(L)}, \mathbf{x}^{(L)}\right)$ can be regarded as realisations from this posterior distribution. Furthermore, it is really important to understand at this point that the i^{th} component of each of the draws $\boldsymbol{\theta}^{(\ell)}$ constitutes a sample from the marginal posterior distribution $\pi(\theta_i|\mathbf{y})$ and not from the conditional posterior distribution $\pi(\theta_i|\boldsymbol{\theta}_{-i}, \mathbf{x}, \mathbf{y})$.

Once we obtain a sample from $\pi(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y})$, we can approximate any feature of the posterior distribution using its empirical counterpart from the simulated draws. For example, we can approximately compute

$$E\left(g(\boldsymbol{\theta})|\mathbf{y}\right)) \approx \frac{1}{L}\sum_{\ell=1}^{L} g(\boldsymbol{\theta}^{(\ell)}), \tag{4.2}$$

for any function $g(\boldsymbol{\theta})$, whose posterior expectation exists.

4.3 Bayesian Inference for a Finite Mixture Model

For a finite mixture model three kinds of statistical inference problems have to be taken into consideration. First, modelling of the data by a finite mixture model requires some specification of the number of components K. Statistical inference for unspecified number of components is a delicate matter, so, for the time being, we are going to assume that K is fixed and known. Second, the component parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K$ and the weight distribution $\mathbf{p}$ should be estimated from the observed data. Finally, each observation $\mathbf{y}_i$ needs to be assigned to a certain component, in order to make inference on the hidden discrete indicators $\mathbf{S}$.

There are various reasons why one might be interested in adopting a Bayesian approach for finite mixture models. First, the inclusion of a proper prior within a Bayesian approach will generally introduce a smoothing effect on the mixture likelihood function and reduce the risk of obtaining spurious modes, in cases where the EM algorithm leads to degenerate solutions. Secondly, as the whole posterior distribution $\pi(\boldsymbol{\vartheta}|\mathbf{y})$ is available, it is much easier to address the issue of parameter uncertainty. Finally, Bayesian estimation does not rely on asymptotic Normality and yields valid

inference even in cases where some regularity conditions are violated, such as small data sets and mixtures with negligible component weights.

Once again, we begin our analysis by rewriting the complete-data likelihood in a way that makes it more convenient to combine it with a suitable prior distribution on the parameter vector $\boldsymbol{\vartheta}$. That is

$$f(\mathbf{y}, \mathbf{s}|\boldsymbol{\vartheta}) = f(\mathbf{s}|\mathbf{p}) f(\mathbf{y}|\mathbf{s}, \boldsymbol{\theta}) = \prod_{k=1}^{K} p_k^{N_k(\mathbf{s})} \cdot \prod_{k=1}^{K} \prod_{i:s_i=k} f(\mathbf{y}_i|\boldsymbol{\theta}_k), \tag{4.3}$$

where $N_k(\mathbf{s}) = \sum_{i=1}^{N} \mathbb{1}\{s_i = k\}$ is counting the number of observations allocated to component k.

When regarded as a function of $\boldsymbol{\vartheta}$, the complete-data likelihood exhibits a rather convenient structure that highly facilitates parameter estimation. It reduces to the product of $K+1$ factors, with the first depending only on the weight distribution $\mathbf{p}$, whereas the last K factors depend on a certain component parameter $\boldsymbol{\theta}_k$. Hence, if the weight distribution and the component parameters are assumed to be independent a priori, then Bayesian estimation can be carried out separately for each of them.

4.3.1 Complete-Data Estimation of the Weight Distribution

For complete-data Bayesian estimation of the weights, the complete-data likelihood $f(\mathbf{s}|\mathbf{p})$ is combined with a prior distribution $\pi(\mathbf{p})$, to obtain the posterior distribution of the weights. Due to the constraint $p_1 + p_2 + \cdots + p_K = 1$, the component weights are not independent.

The complete-data likelihood $f(\mathbf{s}|\mathbf{p})$, when regarded as a function of $\mathbf{s}$, is proportional to the probability mass function of a Multinomial distribution. The conjugate family for the multinomial likelihood is the Dirichlet family of distributions. We assume $\mathbf{p} \sim \mathrm{Dir}(a_1, a_2, \ldots, a_K)$, where

$$\pi(\mathbf{p}) \propto \prod_{k=1}^{K} p_k^{a_k - 1}, \tag{4.4}$$

leading to the following posterior distribution

$$\pi(\mathbf{p}|\mathbf{s}) \propto \prod_{k=1}^{K} p_k^{N_k(\mathbf{s}) + a_k - 1}. \tag{4.5}$$

Of course, equation 4.5 corresponds to the density of a Dirichlet distribution, i.e. $\mathbf{p}|\mathbf{s} \sim \text{Dir}\left(N_1(\mathbf{s}) + a_1, N_2(\mathbf{s}) + a_2, \ldots, N_K(\mathbf{s}) + a_K\right)$. Simulation from this distribution can be easily carried out by drawing $q_1, q_2, \ldots, q_K$ independently, with q_k from a Gamma $\left(N_k(\mathbf{s}) + a_k, 1\right)$ distribution, and setting $q = \sum_{k=1}^{K} q_k$. Then, the vector $\left(\frac{q_1}{q}, \frac{q_2}{q}, \ldots, \frac{q_K}{q}\right)$ has a Dir $\left(N_1(\mathbf{s}) + a_1, N_2(\mathbf{s}) + a_2, \ldots, N_K(\mathbf{s}) + a_K\right)$ distribution.

Under this Dirichlet prior, the marginal posterior of p_k is easily obtained from the joint posterior as follows

$$p_k|\mathbf{s} \sim \text{Beta}\left(N_k(\mathbf{s}) + a_k, N - N_k(\mathbf{s}) + \sum_{j \neq k} a_j\right), \quad k = 1, 2, \ldots, K. \tag{4.6}$$

Hence, even if category k is not directly observed and $N_k(\mathbf{s}) = 0$, the total number of observations in the other categories is highly informative about p_k.

Prior distributions which are common for a Bayesian analysis of latent binary data, where the Dirichlet distribution reduces to a Beta distribution, may be applied, such as the Uniform prior $p_1 \sim \text{Beta}(1, 1) \equiv \mathcal{U}(0, 1)$, Jeffreys' prior $p_1 \sim \text{Beta}\left(\frac{1}{2}, \frac{1}{2}\right)$ or a prior that is uniform in the natural parameter of the exponential family representation, which corresponds to the improper prior $p_1 \sim \text{Beta}(0, 0)$. Dealing with latent multinomial data, one could equivalently utilise the prior $\mathbf{p} \sim \text{Dir}(1, 1, \ldots, 1)$, which is uniform over the unit Simplex $\mathcal{E}_K$, the prior $\mathbf{p} \sim \text{Dir}\left(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\right)$ or the improper prior $\mathbf{p} \sim \text{Dir}(0, 0, \ldots, 0)$.

Generally speaking, reference priors are prior distributions which hold a minimal effect on the final inference, relative to the data. For a mixture of two known densities, the reference prior for p_1 is virtually Jeffreys' prior, when the two densities are well separated, whereas the uniform prior would approximate the reference prior, when the densities are very close. For a mixture of more than two known densities, a Dirichlet distribution with parameters ranging in the interval $\left[\frac{1}{2}, 1\right]$ is a reasonable approximation to the reference prior.

4.3.2 Complete-Data Estimation of the Component-Specific Parameters

The precise prior on the component parameters depends on the distribution family underlying the mixture distribution. Whereas it is not possible to choose simple conjugate priors for the mixture likelihood, $f(\mathbf{y}|\boldsymbol{\vartheta})$, a conjugate analysis is possible for the complete-data likelihood, $f(\mathbf{y}|\mathbf{s}, \boldsymbol{\vartheta})$, if the component densities in the mixture come from the exponential family.

To formulate a joint prior for $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K$, the component parameters are assumed a priori independent, given a fixed hyper-parameter $\boldsymbol{\delta}$. We have

$$\pi(\boldsymbol{\theta}) = \prod_{k=1}^{K} \pi(\boldsymbol{\theta}_k). \tag{4.7}$$

Then, the conditional posterior $\pi(\boldsymbol{\theta}_k|\mathbf{y}, \mathbf{s})$ is given by

$$\pi(\boldsymbol{\theta}_k|\mathbf{y}, \mathbf{s}) \propto \pi(\boldsymbol{\theta}_k) \prod_{i:s_i=k} f(\mathbf{y}_i|\boldsymbol{\theta}_k), \quad k = 1, 2, \ldots, K. \tag{4.8}$$

In our case studies, we often make use of improper priors as uninformative priors. Special care has to be given in such cases, in order to ensure that the posterior obtained is, indeed, a proper distribution.

Results from a Bayesian analysis of finite mixture models are often highly dependent on particular choices of hyper-parameters. In particular for mixtures with components of small sizes, the posterior distribution of the parameters may be sensitive to specific choices of the hyper-parameter $\boldsymbol{\delta}$. To reduce prior sensitivity, it is common practice to use hierarchical priors, which treat $\boldsymbol{\delta}$ as an unknown quantity with a prior $\pi(\boldsymbol{\delta})$ of its own. As a result $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K$ are dependent a priori, with

$$\pi(\boldsymbol{\theta}, \boldsymbol{\delta}) = \pi(\boldsymbol{\delta})\pi(\boldsymbol{\theta}|\boldsymbol{\delta}) = \pi(\boldsymbol{\delta}) \prod_{k=1}^{K} \pi(\boldsymbol{\theta}_k|\boldsymbol{\delta}). \tag{4.9}$$

Of course, conditional on $\boldsymbol{\delta}$, the component parameters are a priori independent, so Gibbs sampling may be carried out independently for each component parameter. If we combine the above prior distribution with $f(\mathbf{y}|\mathbf{s}, \boldsymbol{\theta})$, we obtain the same conditional posterior distribution for the component parameters

$$\pi(\boldsymbol{\theta}_k|\mathbf{y}, \mathbf{s}, \boldsymbol{\delta}) \propto \pi(\boldsymbol{\theta}_k|\boldsymbol{\delta}) \prod_{i:s_i=k} f(\mathbf{y}_i|\boldsymbol{\theta}_k), \quad k = 1, 2, \ldots, K. \tag{4.10}$$

Additionally, the conditional posterior distribution of the hyper-parameter $\boldsymbol{\delta}$, given $\boldsymbol{\theta}$, is calculated as

$$\pi(\boldsymbol{\delta}|\boldsymbol{\theta}) \propto \pi(\boldsymbol{\delta}) \prod_{k=1}^{K} \pi(\boldsymbol{\theta}_k|\boldsymbol{\delta}). \tag{4.11}$$

Partially proper priors are hierarchical priors where the prior $\pi(\boldsymbol{\delta})$ of the hyper-parameter is improper. Although, marginally, the prior $\pi(\boldsymbol{\theta}_k)$ is improper, the posterior distribution is proper.

4.3.3 Classification for Known Component Parameters

It is common to assume that the allocations, like the data, are independent, given the component parameters. Thus, the joint posterior classification distribution $\pi(\mathbf{s}|\boldsymbol{\vartheta}, \mathbf{y})$ is decomposed as

$$\pi(\mathbf{s}|\mathbf{y}, \boldsymbol{\vartheta}) \propto f(\mathbf{s}|\mathbf{p}) f(\mathbf{y}|\mathbf{s}, \boldsymbol{\theta}) = \prod_{i=1}^{N} f(s_i|\mathbf{p}) f(\mathbf{y}_i|\boldsymbol{\theta}_{s_i}).$$

As a result, joint allocation of all observations may be carried out independently for each individual observation.

Classification of a single observation $\mathbf{y}_i$ aims at deriving the conditional probability, $P(S_i = k|\mathbf{y}_i, \boldsymbol{\vartheta})$, for all possible values of k. According to Bayes' theorem, this probability is computed as

$$P(S_i = k|\mathbf{y}_i, \boldsymbol{\vartheta}) = \frac{p_k f(\mathbf{y}_i|\boldsymbol{\theta}_k)}{f(\mathbf{y}_i|\boldsymbol{\vartheta})} \propto p_k f(\mathbf{y}_i|\boldsymbol{\theta}_k), \qquad \begin{matrix} i = 1, 2, \ldots, N, \\ k = 1, 2, \ldots, K. \end{matrix} \tag{4.12}$$

A common classification rule, also called the naive Bayes' classifier, assigns each observation to the component with highest posterior probability, since this minimises the expected misclassification risk. The performance of this classification rule depends on the difference between the true parameter values in the various mixture components.

Rearranging the above equation and substituting it into (3.8), allows us to once again rewrite the complete-data likelihood, as

$$f(\mathbf{y}, \mathbf{s}|\boldsymbol{\vartheta}) = \prod_{i=1}^{N} f(\mathbf{y}_i; \boldsymbol{\vartheta}) \cdot \prod_{i=1}^{N} \prod_{k=1}^{K} P(S_i = k|\mathbf{y}_i, \boldsymbol{\vartheta})^{\mathbb{1}\{S_i = k\}}.$$

Taking the logarithm of both sides yields the following relation between the mixture likelihood and the complete-data likelihood functions

$$\log f(\mathbf{y}|\boldsymbol{\vartheta}) = \log f(\mathbf{y}, \mathbf{s}|\boldsymbol{\vartheta}) - \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}\{S_i = k\} \log P(S_i = k|\mathbf{y}_i, \boldsymbol{\vartheta}).$$

The second right-hand term is a measure of loss of information in the mixture likelihood, compared to the complete-data likelihood function, which becomes zero, when the mixture model enables perfect allocation.

A general way of assessing the quality of the classification based on Bayes' rule, is to consider the entropy $H(\mathbf{S}|\mathbf{y}, \boldsymbol{\vartheta})$ of $f(\mathbf{S}|\mathbf{y}, \boldsymbol{\vartheta})$, which is the expectation of the aforementioned term with respect to the classification distribution $f(\mathbf{S}|\mathbf{y}, \boldsymbol{\vartheta})$. It is

defined as

$$H(\mathbf{S}|\mathbf{y},\boldsymbol{\vartheta}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} P(S_i = k|\mathbf{y}_i,\boldsymbol{\vartheta}) \log P(S_i = k|\mathbf{y}_i,\boldsymbol{\vartheta}) \geq 0. \tag{4.13}$$

For a fixed value of $\boldsymbol{\vartheta}$, the entropy is a measure of how well the data are classified given a mixture distribution defined by $\boldsymbol{\vartheta}$. The entropy is 0 for a perfect classification, where for all observations $P(S_i = k_i|\mathbf{y}_i,\boldsymbol{\vartheta}) = 1$ for a certain value of k_i, otherwise the entropy may be considerably larger.

4.4 Application to Finite Mixture Models

We have already discussed the factorisation of the complete-data likelihood and, as seen in the previous section, a similar structure has been assumed for the prior density $\pi(\boldsymbol{\vartheta},\boldsymbol{\delta})$

$$\pi(\boldsymbol{\vartheta},\boldsymbol{\delta}) = \pi(\mathbf{p})\pi(\boldsymbol{\delta})\pi(\boldsymbol{\theta}|\boldsymbol{\delta}) \propto \pi(\boldsymbol{\delta}) \prod_{k=1}^{K} \pi(\boldsymbol{\theta}_k|\boldsymbol{\delta}) p_k^{a_k-1}. \tag{4.14}$$

Under this prior choice, the complete-data posterior $\pi(\boldsymbol{\vartheta},\boldsymbol{\delta},\mathbf{s}|\mathbf{y})$ factorises in the same convenient way

$$\begin{aligned}\pi(\boldsymbol{\vartheta},\boldsymbol{\delta},\mathbf{s}|\mathbf{y}) &\propto f(\mathbf{s}|\mathbf{p}) f(\mathbf{y}|\mathbf{s},\boldsymbol{\theta})\pi(\mathbf{p})\pi(\boldsymbol{\delta})\pi(\boldsymbol{\theta}|\boldsymbol{\delta}) \\ &\propto \pi(\boldsymbol{\delta}) \cdot \prod_{k=1}^{K} p_k^{N_k(\mathbf{s})+a_k-1} \cdot \prod_{k=1}^{K} \left[\pi(\boldsymbol{\theta}_k|\boldsymbol{\delta}) \prod_{i:s_i=k} f(\mathbf{y}_i|\boldsymbol{\theta}_k) \right].\end{aligned} \tag{4.15}$$

Sampling from this posterior distribution is most commonly implemented by the following Gibbs sampling scheme, where the conditional posterior distributions required are calculated exactly as discussed in the previous section. In other words, the component parameters $\boldsymbol{\vartheta}$ are sampled conditional on knowing the allocations $\mathbf{s}$, whereas the allocations are sampled conditional on knowing the component parameters. Starting with some initial values $\left(\boldsymbol{\delta}^{(0)}, \mathbf{s}^{(0)}\right)$, we iterate the following steps

- Sample the weight distribution $\mathbf{p}^{(\ell+1)}$ from the complete-data posterior distribution $\text{Dir}\left(N_1\left(\mathbf{s}^{(\ell)}\right) + a_1, N_2\left(\mathbf{s}^{(\ell)}\right) + a_2, \ldots, N_K\left(\mathbf{s}^{(\ell)}\right) + a_K\right)$, given by (4.5).
- Sample the component parameters $\boldsymbol{\theta}_1^{(\ell+1)}, \boldsymbol{\theta}_2^{(\ell+1)}, \ldots, \boldsymbol{\theta}_K^{(\ell+1)}$ independently from the conditional posteriors $\pi\left(\boldsymbol{\theta}_k \left| \mathbf{y}, \mathbf{s}^{(\ell)}, \boldsymbol{\delta}^{(\ell)} \right.\right)$, given by (4.10).
- Sample the hyper-parameter $\boldsymbol{\delta}^{(\ell+1)}$ from the conditional posterior $\pi\left(\boldsymbol{\delta} \left| \boldsymbol{\theta}^{(\ell+1)} \right.\right)$, given by (4.11).

- Sample the allocations $s_1^{(\ell+1)}, s_2^{(\ell+1)}, \ldots, s_N^{(\ell+1)}$ independently from the discrete distribution $\pi\left(S_i \left| \mathbf{y}_i, \boldsymbol{\vartheta}^{(\ell+1)}\right.\right)$, given by (4.12).

4.4.1 Finite Mixtures of Poisson Distributions

For mixtures of Poisson distributions, Bayesian inference for the complete data problem leads to the conditionally conjugate prior $\lambda_k|B \sim \text{Gamma}(b, B)$, where both b and B have to be positive to obtain a proper posterior distribution. A hierarchical prior is obtained by assuming that B is a random parameter with a prior of its own, that is, $B \sim \text{Gamma}(g, G)$.

Each of the conditional posteriors $\pi(\lambda_k|\mathbf{y}, \mathbf{s}, B)$ can be handled within this conjugate setting. From Bayes' theorem we obtain

$$\pi(\lambda_k|\mathbf{y}, \mathbf{s}, B) \propto \lambda_k^{S_k(\mathbf{y},\mathbf{s})+b-1} e^{-(N_k(\mathbf{s})+B)\lambda_k}, \quad k = 1, 2, \ldots, K, \tag{4.16}$$

which is a Gamma $(S_k(\mathbf{y}, \mathbf{s}) + b, N_k(\mathbf{s}) + B)$ distribution. Notice that we have introduced the notation $S_k(\mathbf{y}, \mathbf{s}) = \sum_{i:s_i=k} y_i$.

Under this hierarchical prior, B has to be sampled from the conditional posterior distribution $\pi(B|\boldsymbol{\lambda})$, also given by Bayes' theorem as

$$\pi(B|\boldsymbol{\lambda}) \propto B^{Kb+g-1} \exp\left\{ - \left(\sum_{k=1}^{K} \lambda_k + G \right) B \right\}, \tag{4.17}$$

which is a Gamma $\left(Kb + g, \sum_{k=1}^{K} \lambda_k + G \right)$ distribution.

Estimation is rather insensitive to the choice of the parameter g, so it could be chosen as $g = 0.5$. Additionally, fixing b around 1, one could choose $G = \frac{g\bar{y}}{b}$. Full-conditional Gibbs sampling proceeds along the lines indicated by the algorithm for general mixtures of distributions, where the results of this section are used to sample the parameter in each group. Hence, MCMC estimation of a mixture of Poisson distributions, with initial values $\left(B^{(0)}, \mathbf{s}^{(0)}\right)$, consists of iterating the following steps

- Sample the weight distribution $\mathbf{p}^{(\ell+1)}$ from the complete-data posterior distribution $\text{Dir}\left(N_1\left(\mathbf{s}^{(\ell)}\right) + a_1, N_2\left(\mathbf{s}^{(\ell)}\right) + a_2, \ldots, N_K\left(\mathbf{s}^{(\ell)}\right) + a_K\right)$.
- Sample the component parameters $\lambda_1^{(\ell+1)}, \lambda_2^{(\ell+1)}, \ldots, \lambda_K^{(\ell+1)}$ independently from the posteriors $\pi\left(\lambda_k \left| \mathbf{y}, \mathbf{s}^{(\ell)}, B^{(\ell)}\right.\right)$, given by (4.16).
- Sample the hyper-parameter $B^{(\ell+1)}$ from the posterior $\pi\left(B \left| \boldsymbol{\lambda}^{(\ell+1)}\right.\right)$, given by (4.17).

- Sample $s_1^{(\ell+1)}, s_2^{(\ell+1)}, \ldots, s_N^{(\ell+1)}$ independently according to the posterior probabilities $P\left(S_i = k \left| y_i, \boldsymbol{\vartheta}^{(\ell+1)}\right.\right) \propto p_k^{(\ell+1)} e^{-\lambda_k^{(\ell+1)}} \left(\lambda_k^{(\ell+1)}\right)^{y_i}$ for $i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$.

4.4.2 Finite Mixtures of Univariate Normal Distributions

In mixtures of univariate Normal distributions, when holding the variance σ_k^2 fixed, the complete-data likelihood function, regarded as a function of μ_k, is the kernel of a univariate Normal distribution. Under the conjugate prior $\mu_k | b \sim \mathcal{N}(b, B)$, the conditional posterior density of μ_k is

$$\pi\left(\mu_k \left| \mathbf{y}, \mathbf{s}, \sigma_k^2, b\right.\right) \propto \exp\left\{-\frac{BN_k(\mathbf{s}) + \sigma_k^2}{2B\sigma_k^2}\mu_k^2 + \frac{BS_k(\mathbf{y}, \mathbf{s}) + b\sigma_k^2}{B\sigma_k^2}\mu_k\right\}, \tag{4.18}$$

for $k = 1, 2, \ldots, K$, which corresponds to a $\mathcal{N}\left(\frac{BS_k(\mathbf{y}, \mathbf{s}) + b\sigma_k^2}{BN_k(\mathbf{s}) + \sigma_k^2}, \frac{B\sigma_k^2}{BN_k(\mathbf{s}) + \sigma_k^2}\right)$ distribution.

On the other hand, when holding the mean μ_k fixed, the complete-data likelihood, regarded as a function of σ_k^2, is the kernel of an inverse Gamma density. Under the conjugate prior $\sigma_k^2 \left| C \sim \text{Inv-Gamma}(c, C)\right.$, the posterior density of σ_k^2 is

$$\pi\left(\sigma_k^2 \left| \mathbf{y}, \mathbf{s}, \mu_k, C\right.\right) \propto \left(\sigma_k^2\right)^{-\frac{N_k(\mathbf{s})}{2} - c - 1} \exp\left\{-\left(\frac{V_k(\mathbf{y}, \mathbf{s})}{2} + C\right)\frac{1}{\sigma_k^2}\right\}, \tag{4.19}$$

for $k = 1, 2, \ldots, K$ which is an Inv-Gamma $\left(\frac{N_k(\mathbf{s})}{2} + c, \frac{V_k(\mathbf{y}, \mathbf{s})}{2} + C\right)$ distribution with $V_k(\mathbf{y}, \mathbf{s}) = \sum_{i:s_i=k} (y_i - \mu_k)^2$.

The hyper-parameter C can be treated as unknown with a prior of its own, that is, $C \sim \text{Gamma}(g, G)$. For this hierarchical prior, sampling has to be carried out from the conditional posterior density $\pi\left(C \left| \boldsymbol{\sigma}^2\right.\right)$, given by Bayes' theorem as

$$\pi\left(C \left| \boldsymbol{\sigma}^2\right.\right) \propto C^{Kc+g-1} \exp\left\{-\left(\sum_{k=1}^{K} \frac{1}{\sigma_k^2} + G\right) C\right\}, \tag{4.20}$$

which is obviously the kernel of a Gamma $\left(Kc + g, \sum_{k=1}^{K} \frac{1}{\sigma_k^2} + G\right)$ distribution.

A partly proper prior for the variances arises for $g = G = 0$, which leads to the standard improper prior $\pi(C) \propto C^{-1}$ for the scale parameter. Even though the marginal prior distribution of each σ_k^2 also ends up with the usual improper reference prior, $\pi\left(\sigma_k^2\right) \propto \sigma_k^{-2}$, the posterior distribution is proven to be proper.

Similarly, if b is an unknown hyper-parameter with improper prior $\pi(b) \propto 1$, then b is sampled from $\pi(b|\boldsymbol{\mu})$, given by

$$\pi(b|\boldsymbol{\mu}) \propto \exp\left\{-\frac{K}{2B}b^2 + \frac{b}{B}\sum_{k=1}^{K}\mu_k\right\}, \tag{4.21}$$

which corresponds to a $\mathcal{N}\left(\frac{1}{K}\sum_{k=1}^{K}\mu_k, \frac{B}{K}\right)$ distribution.

For our studies, we select $c = 2$, $g = 0.2$, fix B to the square of the range of the observation interval and set $G = \frac{100g}{cB}$. Starting with some initial values $\left(\left(\boldsymbol{\sigma}^2\right)^{(0)}, C^{(0)}, b^{(0)}, \mathbf{s}^{(0)}\right)$, we iterate the following steps

- Sample the weight distribution $\mathbf{p}^{(\ell+1)}$ from the complete-data posterior distribution $\text{Dir}\left(N_1\left(\mathbf{s}^{(\ell)}\right) + a_1, N_2\left(\mathbf{s}^{(\ell)}\right) + a_2, \ldots, N_K\left(\mathbf{s}^{(\ell)}\right) + a_K\right)$.
- Sample the component means $\mu_1^{(\ell+1)}, \mu_2^{(\ell+1)}, \ldots, \mu_K^{(\ell+1)}$ independently from the posteriors $\pi\left(\mu_k \middle| \mathbf{y}, \mathbf{s}^{(\ell)}, \left(\sigma_k^2\right)^{(\ell)}, b^{(\ell)}\right)$, given by (4.18).
- Sample the component variances $\left(\sigma_1^2\right)^{(\ell+1)}, \left(\sigma_2^2\right)^{(\ell+1)}, \ldots, \left(\sigma_K^2\right)^{(\ell+1)}$ independently from the posteriors $\pi\left(\sigma_k^2 \middle| \mathbf{y}, \mathbf{s}^{(\ell)}, \mu_k^{(\ell+1)}, C^{(\ell)}\right)$, given by (4.19).
- Sample the variance scale hyper-parameter $C^{(\ell+1)}$ from the conditional posterior $\pi\left(C \middle| \left(\boldsymbol{\sigma}^2\right)^{(\ell+1)}\right)$, given by (4.20).
- Sample the mean position hyper-parameter $b^{(\ell+1)}$ from the conditional posterior $\pi\left(b \middle| \boldsymbol{\mu}^{(\ell+1)}\right)$, given by (4.21).
- Sample the allocations $s_1^{(\ell+1)}, s_2^{(\ell+1)}, \ldots, s_N^{(\ell+1)}$ independently according to

$$P\left(S_i = k \middle| y_i, \boldsymbol{\vartheta}^{(\ell+1)}\right) \propto \frac{p_k^{(\ell+1)}}{\sigma_k^{(\ell+1)}} \exp\left\{-\frac{\left(y_i - \mu_k^{(\ell+1)}\right)^2}{2\left(\sigma_k^2\right)^{(\ell+1)}}\right\}, \qquad \begin{array}{l} i = 1, 2, \ldots, N, \\ k = 1, 2, \ldots, K. \end{array}$$

4.4.3 Finite Mixtures of Multivariate Normal Distributions

Similarly to the univariate case, the complete-data likelihood function of a multivariate Normal mixture, regarded as a function of $\boldsymbol{\mu}_k$, when holding the variance-covariance matrix $\boldsymbol{\Sigma}_k$ fixed, is the kernel of a multivariate Normal distribution. Under the conjugate prior $\boldsymbol{\mu}_k \sim \mathcal{N}_d(\mathbf{b}, \mathbf{B})$, the conditional posterior density of $\boldsymbol{\mu}_k$ is

$$\pi(\boldsymbol{\mu}_k|\mathbf{y}, \mathbf{s}, \boldsymbol{\Sigma}_k) \propto \exp\left\{-\frac{\boldsymbol{\mu}_k^T\left(N_k(\mathbf{s})\boldsymbol{\Sigma}_k^{-1} + \mathbf{B}^{-1}\right)\boldsymbol{\mu}_k}{2} + \boldsymbol{\mu}_k^T\left(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_k(\mathbf{y}, \mathbf{s}) + \mathbf{B}^{-1}\mathbf{b}\right)\right\}, \tag{4.22}$$

for $k = 1, 2, \ldots, K$, which is a $\mathcal{N}_d\left(\mathbf{B}_k(\mathbf{s}, \boldsymbol{\Sigma}_k)\left(\boldsymbol{\Sigma}_k^{-1}\mathbf{S}_k(\mathbf{y}, \mathbf{s}) + \mathbf{B}^{-1}\mathbf{b}\right), \mathbf{B}_k(\mathbf{s}, \boldsymbol{\Sigma}_k)\right)$ distribution with $\mathbf{B}_k(\mathbf{s}, \boldsymbol{\Sigma}_k) = \left(N_k(\mathbf{s})\boldsymbol{\Sigma}_k^{-1} + \mathbf{B}^{-1}\right)^{-1}$.

On the other hand, when holding the mean $\boldsymbol{\mu}_k$ fixed, the complete-data likelihood, regarded as a function of $\boldsymbol{\Sigma}_k$, is the kernel of an inverse Wishart density. Under the conjugate inverse Wishart prior, $\boldsymbol{\Sigma}_k|\mathbf{C} \sim \mathcal{W}_d^{-1}(\mathbf{C}, c)$, the posterior density of $\boldsymbol{\Sigma}_k$ is

$$\pi\left(\boldsymbol{\Sigma}_k|\mathbf{y}, \mathbf{s}, \boldsymbol{\mu}_k, \mathbf{C}\right) \propto |\boldsymbol{\Sigma}_k|^{-\frac{N_k(\mathbf{s})+c+d+1}{2}} \exp\left\{-\frac{1}{2}\text{tr}\left[\left(\mathbf{V}_k(\mathbf{y}, \mathbf{s}) + \mathbf{C}\right)\boldsymbol{\Sigma}_k^{-1}\right]\right\}, \tag{4.23}$$

for $k = 1, 2, \ldots, K$, which is also an inverse Wishart $\mathcal{W}_d^{-1}\left(\mathbf{V}_k(\mathbf{y}, \mathbf{s}) + \mathbf{C}, N_k(\mathbf{s}) + c\right)$ distribution with $\mathbf{V}_k(\mathbf{y}, \mathbf{s}) = \sum\limits_{i:s_i=k} (\mathbf{y}_i - \boldsymbol{\mu}_k)(\mathbf{y}_i - \boldsymbol{\mu}_k)^T$.

Again, the hyper-parameter $\mathbf{C}$ can be treated as unknown with a Wishart prior $\mathbf{C} \sim \mathcal{W}_d(\mathbf{G}, g)$. The conditional posterior density $\pi(\mathbf{C}|\boldsymbol{\Sigma})$ is given by

$$\pi(\mathbf{C}|\boldsymbol{\Sigma}) \propto |\mathbf{C}|^{\frac{Kc+g-d-1}{2}} \exp\left\{-\frac{1}{2}\text{tr}\left[\left(\sum_{k=1}^{K}\boldsymbol{\Sigma}_k^{-1} + \mathbf{G}^{-1}\right)\mathbf{C}\right]\right\}, \tag{4.24}$$

which is obviously equal to the kernel of a Wishart $\mathcal{W}_d\left(\mathbf{G}(\boldsymbol{\Sigma}), Kc + g\right)$ distribution with $\mathbf{G}(\boldsymbol{\Sigma}) = \left(\sum\limits_{k=1}^{K}\boldsymbol{\Sigma}_k^{-1} + \mathbf{G}^{-1}\right)^{-1}$.

We select $c = 2.5 + \frac{d-1}{2}$, $g = 0.1 \cdot c$, fix each element of $\mathbf{b}$ to the midpoint of the observation interval of the corresponding component of $\mathbf{y}$, each diagonal element of $\mathbf{B}$ to the square of the range of the observation interval of the corresponding component of $\mathbf{y}$ and set $G = 10 \cdot \mathbf{B}^{-1}$. Finally, to implement the Gibbs sampler, we begin with some initial values $\left(\boldsymbol{\Sigma}^{(0)}, \mathbf{C}^{(0)}, \mathbf{s}^{(0)}\right)$ and iterate the following steps

- Sample the weight distribution $\mathbf{p}^{(\ell+1)}$ from the complete-data posterior distribution $\text{Dir}\left(N_1\left(\mathbf{s}^{(\ell)}\right) + a_1, N_2\left(\mathbf{s}^{(\ell)}\right) + a_2, \ldots, N_K\left(\mathbf{s}^{(\ell)}\right) + a_K\right)$.
- Sample the component means $\boldsymbol{\mu}_1^{(\ell+1)}, \boldsymbol{\mu}_2^{(\ell+1)}, \ldots, \boldsymbol{\mu}_K^{(\ell+1)}$ independently from the posteriors $\pi\left(\boldsymbol{\mu}_k \middle| \mathbf{y}, \mathbf{s}^{(\ell)}, \boldsymbol{\Sigma}_k^{(\ell)}\right)$, given by (4.22).
- Sample the covariance matrices $\boldsymbol{\Sigma}_1^{(\ell+1)}, \boldsymbol{\Sigma}_2^{(\ell+1)}, \ldots, \boldsymbol{\Sigma}_K^{(\ell+1)}$ independently from the posteriors $\pi\left(\boldsymbol{\Sigma}_k \middle| \mathbf{y}, \mathbf{s}^{(\ell)}, \boldsymbol{\mu}_k^{(\ell+1)}, \mathbf{C}^{(\ell)}\right)$, given by (4.23).
- Sample the hyper-parameter $\mathbf{C}^{(\ell+1)}$ from the posterior $\pi\left(\mathbf{C} \middle| \boldsymbol{\Sigma}^{(\ell+1)}\right)$, given by (4.24).
- Sample the allocations $s_1^{(\ell+1)}, s_2^{(\ell+1)}, \ldots, s_N^{(\ell+1)}$ independently according to

$$P\left(S_i = k \middle| \mathbf{y}_i, \boldsymbol{\vartheta}^{(\ell+1)}\ s\right) \propto$$
$$p_k^{(\ell+1)}\left|\boldsymbol{\Sigma}_k^{(\ell+1)}\right|^{-1/2} \exp\left\{-\frac{\left(\mathbf{y}_i - \boldsymbol{\mu}_k^{(\ell+1)}\right)^T\left(\boldsymbol{\Sigma}_k^{(\ell+1)}\right)^{-1}\left(\mathbf{y}_i - \boldsymbol{\mu}_k^{(\ell+1)}\right)}{2}\right\},$$

for $i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$.

4.5 Identifiability Issues

Now, we have to broach the more formal issue of identifiability of a mixture distribution, which is essential for parameter estimation. For mixtures of probability distributions, one has to distinguish among three types of non-identifiability. Non-identifiability due to invariance to relabelling the components of the mixture distribution and due to potential over-fitting may be ruled out through formal identifiability constraints. The last type of non-identifiability is a generic property of certain classes of mixture distributions.

4.5.1 Label Switching

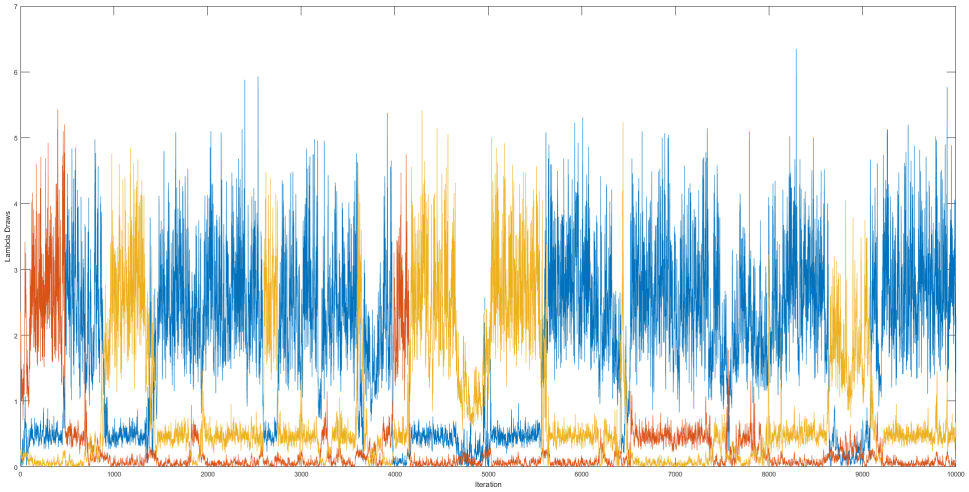

FIGURE 4.1: MCMC Draws Displaying Label Switching

The term label switching refers to the invariance of the mixture distribution to relabelling the components of the mixture. For a general finite mixture distribution with K components, there exist $K!$ equivalent ways of arranging these components. Each of them may be described in terms of a permutation $\sigma : \{1, 2, \ldots, K\} \to \{1, 2, \ldots, K\}$, where the value $\sigma(k)$ is assigned to each value $k \in \{1, 2, \ldots, K\}$.

Let $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, p_1, \ldots, p_,)$ be an arbitrary point in the parameter space $\Theta_K = \Theta^K \times \mathcal{E}_K$. Any point $\boldsymbol{\vartheta}^* = \left(\boldsymbol{\theta}_{\sigma(1)}, \ldots, \boldsymbol{\theta}_{\sigma(K)}, p_{\sigma(1)}, \ldots, p_{\sigma(K)}\right)$ generates the same mixture density as $\boldsymbol{\vartheta}$, which is easily seen by rearranging the components of the mixture density according to the permutation σ

$$\begin{aligned} f(y|\boldsymbol{\vartheta}) &= p_1 f(y|\boldsymbol{\theta}_1) + \cdots + p_K f(y|\boldsymbol{\theta}_K) \\ &= p_{\sigma(1)} f\left(y|\boldsymbol{\theta}_{\sigma(1)}\right) + \cdots + p_{\sigma(K)} f\left(y|\boldsymbol{\theta}_{\sigma(K)}\right) = f(y|\boldsymbol{\vartheta}^*). \end{aligned}$$

There exist $K!$ such different parameters $\boldsymbol{\vartheta}^*$, if and only if all K component parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ are distinct points in Θ. The set contains only $\frac{K!}{L!}$ distinct parameters $\boldsymbol{\vartheta}^*$, if L among the K component parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ of $\boldsymbol{\vartheta}$ are identical. Thus, for each $\boldsymbol{\vartheta}$ with at east two distinct component parameters $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_j$, the corresponding mixture distribution is non-identifiable.

Because the components in a mixture density may be arbitrarily arranged, it is usual to choose priors which reflect this attribute by being themselves invariant to relabelling the components. A prior density $\pi(\boldsymbol{\vartheta})$ is invariant to relabelling the components of the mixture model if the identity $\pi(\boldsymbol{\vartheta}^*) = \pi(\boldsymbol{\vartheta})$ holds for all $\boldsymbol{\vartheta}$ and for any of the $K!$ permutations σ of $\{1, 2, \ldots, K\}$.

Label switching is of no concern for maximum likelihood estimation, where the goal is to find one of the equivalent modes of the likelihood function. In the context of Bayesian estimation, however, label switching has to be addressed explicitly, as, in the course of sampling from the mixture posterior distribution, the labelling of the unobserved categories may potentially change.

Invariance of the Mixture Posterior Distribution

The mixture posterior density $\pi(\boldsymbol{\vartheta}|\mathbf{y})$ is, to a large extent, dominated by the mixture likelihood function $f(\mathbf{y}|\boldsymbol{\vartheta})$. Under an invariant prior distribution, the mixture posterior distribution inherits the invariance of the mixture likelihood and it holds that $\pi(\boldsymbol{\vartheta}^*|\mathbf{y}) = \pi(\boldsymbol{\vartheta}|\mathbf{y})$.

The invariance property of the mixture posterior density causes state independence of many functionals derived from the posterior distribution, which at first sight appear to be component-specific. Consider, as an example, the marginal distribution of the component parameter $\boldsymbol{\theta}_k$, which is defined as

$$\begin{aligned}\pi(\boldsymbol{\theta}_k|\mathbf{y}) &= \int_{\Theta^{K-1}\times\mathcal{E}_K} \pi(\boldsymbol{\vartheta}|\mathbf{y}) d\mathbf{p} d\boldsymbol{\theta}_1 \cdots d\boldsymbol{\theta}_{k-1} d\boldsymbol{\theta}_{k+1} \cdots d\boldsymbol{\theta}_K \\ &= \int_{\Theta^{K-1}\times\mathcal{E}_K} \pi(\boldsymbol{\vartheta}^*|\mathbf{y}) d\mathbf{p} d\boldsymbol{\theta}_{\sigma(1)} \cdots d\boldsymbol{\theta}_{\sigma(k-1)} d\boldsymbol{\theta}_{\sigma(k+1)} \cdots d\boldsymbol{\theta}_{\sigma(K)} = \pi(\boldsymbol{\theta}_{\sigma(k)}|\mathbf{y}).\end{aligned}$$

Because this applies for all permutations σ of $\{1, 2, \ldots, K\}$, the marginal posterior densities $\pi(\boldsymbol{\theta}_k|\mathbf{y})$ are actually state-independent and the identity $\pi(\boldsymbol{\theta}_k|\mathbf{y}) = \pi(\boldsymbol{\theta}_j|\mathbf{y})$ holds for all $k, j \in \{1, 2, \ldots, K\}$. It could be proven, in a similar way, that the marginal posterior density of the component weight p_k is state-independent and that the identity $\pi(p_k|\mathbf{y}) = \pi(p_j|\mathbf{y})$ is valid for all $k, j \in \{1, 2, \ldots, K\}$.

The posterior mean is a commonly used point estimator, which is optimal with respect to a quadratic loss function. It follows, from the previous discussion, that the

seemingly component-specific means of $\boldsymbol{\theta}_k$ and p_k are also actually state-independent, i.e. $E(\boldsymbol{\theta}_k|\mathbf{y}) = E(\boldsymbol{\theta}_j|\mathbf{y})$ and $E(p_k|\mathbf{y}) = E(p_j|\mathbf{y})$ for any $k, j \in \{1, 2, \ldots, K\}$. As a result, the mean $E(\boldsymbol{\vartheta}|\mathbf{y})$ of the mixture posterior is not a sensible point estimator for the component parameters and the weight distribution.

State invariance occurs also for the seemingly component-dependent allocations $\mathbf{S}$. The marginal posterior distribution $\pi(\mathbf{s}|\mathbf{y})$ is defined as

$$\pi(\mathbf{s}|\mathbf{y}) = \int_{\Theta_K} \pi(\mathbf{s}, \boldsymbol{\vartheta}|\mathbf{y}) d\boldsymbol{\vartheta} = \int_{\Theta_K} \pi(\mathbf{s}^*, \boldsymbol{\vartheta}^*|\mathbf{y}) d\boldsymbol{\vartheta}^* = \pi(\mathbf{s}^*|\mathbf{y}),$$

where $\mathbf{s}^* = (\sigma(s_1), \sigma(s_2), \ldots, \sigma(s_K))$. Consequently, it turns out that the marginal posterior probability $P(S_i = k|\mathbf{y})$ is state-independent and equal to $\frac{1}{K}$ regardless of the data.

Model Identification

Inference on functionals of $\boldsymbol{\vartheta}$ which are not invariant to relabelling the components of the finite mixture is sensible only if the posterior draws come from a unique labelling subspace of the unconstrained parameter space. Gibbs sampling may lead to implicit model identification if the $K!$ modal regions of the mixture posterior density are well-separated and the sampler is trapped in one of them. Nevertheless, this is not always the case.

To achieve model identification, one strategy is to relabel the posterior draws $\boldsymbol{\vartheta}^\ell$ in such a way that draws from a unique labelling subspace result. A common reaction to the label switching problem is to impose some formal identifiability constraint within sampling-based Bayesian estimation.

An inequality constraint on the component parameters forces a unique labelling. For mixtures with a univariate component parameter θ_k, this condition evidently reads $\theta_1 < \theta_2 < \cdots < \theta_K$. For mixtures with a multivariate component parameter, one could require only that any two parameters $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_j$ differ in at least one element, which does not need be the same for all components. Naturally, the identification of a valid constraint in higher dimensions may be somewhat of a challenge.

A straightforward method to impose a constraint on the posterior draws is to post-process the MCMC draws which were generated from the mixture posterior. Whenever a draw does not satisfy the constraint, one permutes the labelling of the components in such a way that the constraint is fulfilled. It can be proven that this method actually delivers a sample from the constrained posterior.

As a more automatic procedure, it is suggested to permute the MCMC draws obtained from unconstrained sampling by utilising a clustering procedure. For example, a k-means clustering algorithm with $K!$ clusters, which is initialised from the first 100 draws, after reaching burn-in, by defining $K!$ reference centres from these draws, could be used. For each MCMC draw $\boldsymbol{\theta}^{\ell}$, the distance to each of the $K!$ centres is computed, which is then used to permute the labels.

4.5.2 Potential Over-Fitting

A further identifiability problem is non-identifiability due to potential over-fitting. Any mixture with $K-1$ components defines a non-identifiability subset in the larger parameter space Θ_K, where either one component is empty or two components are equal.

Indeed, any mixture of $K-1$ distributions may be written as a mixture of K distributions by adding another component with weight $p_K = 0$ as evidenced below

$$\begin{aligned} f(y|\boldsymbol{\vartheta}) &= p_1 f(y|\boldsymbol{\theta}_1) + \cdots + p_{K-1} f(y|\boldsymbol{\theta}_{K-1}) \\ &= p_1 f(y|\boldsymbol{\theta}_1) + \cdots + p_{K-1} f(y|\boldsymbol{\theta}_{K-1}) + 0 \cdot f(y|\boldsymbol{\theta}_K). \end{aligned}$$

In the parameter space Θ_K, the parameter $\boldsymbol{\vartheta}$ corresponding to this mixture lies in a non-identifiability set, as the density $f(y|\boldsymbol{\vartheta})$ is the same for arbitrary values of $\boldsymbol{\theta}_K$.

The same non-identifiability set results if a mixture of K distributions is generated by splitting one component of a mixture of $K-1$ distributions into two as follows

$$\begin{aligned} f(y|\boldsymbol{\vartheta}) &= p_1 f(y|\boldsymbol{\theta}_1) + \cdots + p_{K-1} f(y|\boldsymbol{\theta}_{K-1}) \\ &= p_1 f(y|\boldsymbol{\theta}_1) + \cdots + (p_k - p_K) f(y|\boldsymbol{\theta}_k) + \cdots + p_{K-1} f(y|\boldsymbol{\theta}_{K-1}) + p_K f(y|\boldsymbol{\theta}_k). \end{aligned}$$

Again, the parameter vector $\boldsymbol{\vartheta}$ lies in a non-identifiability set, as the density $f(y|\boldsymbol{\vartheta})$ is the same for arbitrary values of p_K with $0 \leqslant p_K \leqslant p_k$. Furthermore, this is the same non-identifiability set as the above. A positivity constraint on the weights avoids non-identifiability due to empty components, whereas an inequality condition on the component parameters avoids non-identifiability due to equal components.

Unfortunately, label switching is unavoidable for a model which is over-fitting the number of components. Consequently, it is not sensible to try model identification for such a mixture. Rather, it may be desirable to apply a prior which is informative enough to bound the posterior away from the non-identifiability sets.

To avoid sampling mixtures with empty components, it is sensible to select a $\mathrm{Dir}(a_1, a_2, \ldots, a_k)$ prior for the weight distribution with $a_k > 1$ for $k = 1, 2, \ldots, K$.

Consequently, two of the component parameters will be pulled together to capture over-fitting and observations are then allocated more or less randomly between these components. Therefore, sampling of component parameters from the prior is avoided, which increases the stability of the sampler.

To avoid sampling mixtures with two identical components, it often helps to modify the prior on the component parameters, by increasing prior shrinkage for elements of the component parameter $\boldsymbol{\theta}_k$ which are not extremely distinctive between the components. Simultaneously, too strong a shrinkage for elements of the component parameter $\boldsymbol{\theta}_k$ which differ between the components should be avoided.

4.5.3 Generic Identifiability

Generic identifiability of a certain family of finite mixture distributions is a class property, which does not respond to formal identifiability constraints. It has been shown that a family of mixture distributions is generically identifiable if and only if the members of the underlying distribution are linearly independent over the field of real numbers. However, it is often easier to verify identifiability through some transform of the underlying distribution, such as the characteristic or the moment-generating function.

As an example, for a mixture of Normal distributions with common variance, the identity

$$\sum_{k=1}^{K} p_k e^{i\mu_k z} e^{-\sigma^2 z^2/2} = 0,$$

pertaining to the characteristic function of the component densities, is possible for all $z \in \mathbb{R}$ if and only if $p_1 = p_2 = \cdots = p_k = 0$.

Many mixtures of univariate continuous densities, such as mixtures of Normal or Gamma distributions, are generically identifiable. These results may be extended to multivariate families, such as multivariate mixtures of Normal distributions. On the other hand, not all mixtures of discrete distributions are identifiable, as can be demonstrated for a mixture of $\text{Bin}(n, p)$ distributions with $n < 2K - 1$. Mixtures of Poisson or Negative Binomial distributions though are, indeed, identifiable.

Chapter 5

Applications to Hidden Markov Models

5.1 Introduction

We now turn to the application of the algorithms discussed in Chapters 3 and 4 on the class of discrete-time, finite state-space hidden Markov models. A fundamental issue in hidden Markov modelling, is drawing inference on the unobserved state sequence $\mathbf{X}$. As already glimpsed in Chapter 2, calculation of the conditional distribution of a state $\mathbf{X}_k$ given the observations $\mathbf{y}$, which is generally referred to as a smoothing distribution, may prove to be a considerable task.

There exist a variety of smoothing approaches with computational cost that only increases linearly with the number of observations. This is only made possible by the fact that, conditional on the observations $\mathbf{y}$, the state sequence still constitutes a Markov chain, albeit a non-homogeneous one.

Of course exact numerical evaluation of the quantities involved in the smoothing recursions discussed in this chapter is only feasible in particular classes of HMMS, like the class of finite state-space HMMs and the Gaussian linear state-space models. In all other cases, one must consider approximate smoothing methods based on Monte Carlo simulations.

Smoothing is crucial in the implementation of all the algorithms discussed in the previous chapters to make inferences on HMMs. Having calculated and stored the aforementioned conditional distributions of each state $\mathbf{X}_k$, one may then utilise them to calculate the intermediate quantity of EM for implementation of the EM algorithm or propose a sampling scheme for the realisations of the underlying Markov chain within a Gibbs sampler.

5.2 State Inference

In what follows, we denote by $\phi_{k:\ell|m}$ the conditional distribution of $\mathbf{X}_{k:\ell}$ given $\mathbf{y}_{0:m}$ for any positive indices k, ℓ and m with $\ell \geqslant k$. Specific choices of k, ℓ and m give rise to particular quantities of interest, such as

- **Joint Smoothing**: $\boldsymbol{\phi}_{0:n|n}$ for $n \geqslant 0$.
- **Marginal Smoothing**: $\boldsymbol{\phi}_{k|n}$ for $n \geqslant k \geqslant 0$.
- **Prediction**: $\boldsymbol{\phi}_{k|k-1}$ for $k \geqslant 1$. It is convenient to extend this notation, to use $\boldsymbol{\phi}_{0|-1}$ as a synonym for the initial distribution $\boldsymbol{\nu}$.
- **Filtering**: $\boldsymbol{\phi}_{k|k}$ for $k \geqslant 0$. Because the use of filtering will be prominent in the following, this notation will be abbreviated to $\boldsymbol{\phi}_k$.

Now, for example, the conditional probabilities involved in equation (2.7) for the calculation of the observed likelihood of an HMM can easily be recognised as predictive probabilities.

5.2.1 The Forward-Backward Algorithm

The structure of an HMM is straightforward enough in its design that efficient instances of this smoothing approach can all be decomposed into two systematic phases: one in which the graph of the HMM is scanned systematically from left to right, referred to as the forward pass, and one in which the graph is scanned in reverse order, called the backward pass.

Forward Filtering

The objective of the forward pass is to calculate the forward filtering probabilities, $\phi_k(i) = P(X_k = i|\mathbf{y}_{0:k})$, for $k = 0, 1, \ldots, n$, through the use of a recursive scheme. These probabilities can be expressed through Bayes' theorem in the following way

$$\phi_k(i) = \frac{P(X_k = i|\mathbf{y}_{0:k-1}) f(\mathbf{y}_k|X_k = i)}{f(\mathbf{y}_k|\mathbf{y}_{0:k-1})} = \frac{\phi_{k|k-1}(i) f_i(\mathbf{y}_k)}{c_k}, \quad \begin{matrix} k = 0, 1, \ldots, n, \\ i = 1, 2, \ldots, r, \end{matrix} \tag{5.1}$$

where $c_k = f(\mathbf{y}_k|\mathbf{y}_{0:k-1}) = \sum_{j=i}^{r} \phi_{k|k-1}(j) f_j(\mathbf{y}_k)$.

As it can be seen, in order to actually calculate the forward filtering probabilities, we first need to calculate the predictive probabilities, $\phi_{k|k-1}(i) = P(X_k = i|\mathbf{y}_{0:k-1})$, for $k = 1, 2, \ldots, n$, always keeping in mind that $\phi_{0|-1}(i) = \nu_i$. This is achieved

through the law of total probability

$$\phi_{k|k-1}(i) = \sum_{j=1}^{r} P(X_{k-1} = j|\mathbf{y}_{0:k-1})P(X_k = i|X_{k-1} = j) = \sum_{j=1}^{r} \phi_{k-1}(j)p_{ji}, \quad (5.2)$$

for $k = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, r$.

Hence, by utilising the initial distribution $\boldsymbol{\nu}$, one may first initialise the filtering probabilities, $\phi_0(i)$, for $i = 1, 2, \ldots, r$. Afterwards, for $k = 1, 2, \ldots, n$, one can calculate the predictive probabilities, based on the filtering probabilities stored from the previous iteration, and, then, calculate and store the next set of filtering probabilities, using the newly calculated predictive probabilities. This is the essence of the so-called forward filtering algorithm, described in detail below

- **Initialisation**: Calculate
  - The normalisation constant $c_0 = \sum_{j=i}^{r} \nu_j f_j(\mathbf{y}_0)$;
  - The filtering probabilities $\phi_0(i) = \frac{\nu_i f_i(\mathbf{y}_0)}{c_0}$ for $i = 1, 2, \ldots, r$.
- **Forward Recursion**: For $k = 1, 2, \ldots, n$, calculate
  - The predictive probabilities $\phi_{k|k-1}(i)$ for $i = 1, 2, \ldots, r$ according to (5.2);
  - The normalisation constant $c_k = \sum_{j=i}^{r} \phi_{k|k-1}(j) f_j(\mathbf{y}_k)$;
  - The filtering probabilities $\phi_k(i)$ for $i = 1, 2, \ldots, r$ according to (5.1).

The computational cost of this filtering method is thus proportional to the number of observations n, and scales like r^2, because of the r vector-matrix products corresponding to equation (5.1).

Of course, having calculated and stored the aforementioned normalisation constants, c_k, for $k = 0, 1, \ldots, n$, one may immediately evaluate the observed likelihood, by rewriting equation (2.7) as $L_n = L(\boldsymbol{\vartheta}|\mathbf{y}_{0:n}) = \prod_{k=0}^{n} c_k$. However, one should systematically opt to calculate the observed likelihood on the log scale, according to $\ell_n = \log L_n = \sum_{k=0}^{n} \log c_k$, rather than on a linear scale, as this form is robust to numerical under or over-flow.

Markovian Backward Smoothing

On the other hand, the backward pass aims at calculating the marginal smoothing probabilities $\phi_{k|n}(i) = P(X_k = i|\mathbf{y}_{0:n})$ and bivariate smoothing probabilities $\phi_{k:k+1|n}(i, j) = P(X_k = i, X_{k+1} = j|\mathbf{y}_{0:n})$ by relying exclusively on the filtering probabilities calculated from the forward pass. More precisely, the bivariate smoothing

distribution can easily be obtained using of the chain rule

$$\phi_{k:k+1|n}(i,j) = P(X_{k+1} = j|\mathbf{y}_{0:n})P(X_k = i|X_{k+1} = j, \mathbf{y}_{0:k}) = \phi_{k+1|n}(j)B_k(j,i), \tag{5.3}$$

for $k = 0, 1, \ldots, n-1$, $i = 1, 2, \ldots, r$ and $j = 1, 2, \ldots, r$, with the probabilities denoted by $B_k(j,i) = P(X_k = i|X_{k+1} = j, \mathbf{y}_{0:k})$ constituting the backward transition kernel of the underlying Markov chain.

From there, it is straightforward to compute the marginal smoothing probabilities through the law of total probability as follows

$$\phi_{k|n}(i) = \sum_{j=1}^{r} \phi_{k:k+1|n}(i,j), \qquad \begin{matrix} k = 0, 1, \ldots, n, \\ i = 1, 2, \ldots, r. \end{matrix} \tag{5.4}$$

The backward transition probabilities $B_k(j,i)$ can also be proven to depend solely on the filtering distributions themselves and not on the actual data. Specifically, applying Bayes' theorem one may obtain

$$B_k(j,i) = \frac{P(X_k = i|\mathbf{y}_{0:k})P(X_{k+1} = j|X_k = i)}{P(X_{k+1} = j|\mathbf{y}_{0:k})} = \frac{\phi_k(i)p_{ij}}{\sum\limits_{\ell=1}^{r} \phi_k(\ell)p_{\ell j}}, \tag{5.5}$$

for $k = 0, 1, \ldots, n-1$ and $i, j = 1, 2, \ldots, r$.

In particular cases where the above denominator happens to be equal to zero for a certain index j, then, it can be shown that the smoothing probability $\phi_{k+1|n}(j)$ will also be equal to zero, so, for the purposes of this algorithm, the corresponding probabilities $B_k(j,i)$ can be set to arbitrary values for this particular value of j. The Markovian backward smoothing algorithm generally proceeds as follows

- **Initialisation**: For $i = 1, 2, \ldots, r$, set $\phi_{n|n}(i) = \phi_n(i)$.
- **Backward Recursion**: For $k = n-1, n-2, \ldots, 0$, calculate
  - The backward transition probabilities $B_k(j,i)$ for $i, j = 1, 2, \ldots, r$ according to equation (5.5);
  - The bivariate smoothing probabilities $\phi_{k:k+1|n}(i,j)$ for $i, j = 1, 2, \ldots, r$ according to equation (5.3);
  - The marginal smoothing probabilities $\phi_{k|n}(i)$ for $i = 1, 2, \ldots, r$ according to equation (5.4).

Of course the Forward-Backward algorithm may easily be replaced by an equivalent Backward-Forward algorithm, where the filtering distributions are first computed

via a backward recursion and, then, a forward Markovian decomposition is utilised to compute the forward transition kernel.

Finally, according to the Markov property, the joint smoothing distribution can be written as a product of the form

$$\begin{aligned}\phi_{0:n|n}(\mathbf{x}_{0:n}) &= P(X_n = x_n|\mathbf{y}_{0:n}) \prod_{k=0}^{n-1} P(X_k = x_k|X_{k+1} = x_{k+1}, \mathbf{y}_{0:k}) \\ &= \phi_n(x_n) \prod_{k=0}^{n-1} B_k(x_{k+1}, x_k).\end{aligned} \tag{5.6}$$

5.2.2 The Viterbi Algorithm

In the case of finite state-space HMMs, it turns out that one may possibly achieve a different kind of inference concerning the underlying sequence of states. This kind of inference is non-probabilistic, in the sense that it does not provide a distributional statement concerning the unobservable states. The result obtained is, rather, the jointly optimal, in terms of maximal conditional probability, sequence of unknown states, given the corresponding observations, which, in some sense, is much stronger a result than just the marginally optimal sequence of states, given by the Forward-Backward algorithm. It may even be that a transition $x_k \to x_{k+1}$ of the marginally optimal sequence is disallowed, in the sense that $p(x_k, x_{k+1}) = 0$.

The algorithm that makes it possible to efficiently compute the a posteriori most likely sequence of states is known as the Viterbi algorithm. It is based on the widely known dynamic programming principle. The key observation is the following recursive equation for the complete-data likelihood of an HMM, which we immediately present in log form

$$\ell(\boldsymbol{\vartheta}|\mathbf{x}_{0:k+1}, \mathbf{y}_{0:k+1}) = \ell(\boldsymbol{\vartheta}|\mathbf{x}_{0:k}, \mathbf{y}_{0:k}) + \log p(x_k, x_k + 1) + \log f_{x_{k+1}}(\mathbf{y}_{k+1}).$$

Making use of the observed likelihood, ℓ_k, and the joint smoothing distribution, $\boldsymbol{\phi}_{0:k|k}$, discussed beforehand, we receive the following equation

$$\ell_{k+1} + \log \phi_{0:k+1|k+1}(\mathbf{x}_{0:k+1}) = \ell_k + \log \phi_{0:k|k}(\mathbf{x}_{0:k}) + \log p(x_k, x_k + 1) + \log f_{x_{k+1}}(\mathbf{y}_{k+1}).$$

The pre-eminent feature of this recursive equation is that, ignoring the observed log-likelihoods, since they do not depend on the state sequence, the posterior log-probability of the subsequence $\mathbf{x}_{0:k+1}$ is equal to that of $\mathbf{x}_{0:k}$ up to terms that solely

involve the pair (x_k, x_{k+1}). Hence, one may define

$$T_k(i) = \ell_k + \max_{\mathbf{x}_{0:k-1} \in \mathbf{X}^k} \log \phi_{0:k|k}(x_0, x_1, \ldots, x_{k-1}, i), \qquad \begin{matrix} k = 0, 1, \ldots, n, \\ i = 1, 2, \ldots, r, \end{matrix}$$

that is, up to a number independent of the state sequence, the maximal conditional log-probability of a sequence up to time k and ending with state i. We also define $B_k(i) = \arg\max_{x_{k-1} \in \mathbf{X}} T_k(i)$, that is, the second final state in an optimal state sequence of length $k+1$ and ending with state i. Substituting $T_k(i)$ into the previous equation, results in the simple recursive equation

$$T_{k+1}(j) = \max_{i=1,2,\ldots,r} [T_k(i) + \log p_{ij}] + \log f_j(\mathbf{y}_{k+1}), \qquad \begin{matrix} k = 0, 1, \ldots, n-1, \\ j = 1, 2, \ldots, r, \end{matrix} \tag{5.7}$$

where $B_{k+1}(j)$ is the index i for which the maximum $T_k(i) + \log p_{ij}$ is achieved.

The above observations immediately lead us to formulate the Viterbi algorithm for the computation of the a posteriori most likely sequence of states as follows

- **Forward Recursion**: Computation of the optimal conditional probabilities.
  - For $i = 1, 2, \ldots, r$, let $T_0(i) = \log \nu_i + \log f_i(\mathbf{y}_0)$.
  - For $k = 0, 1, \ldots, n-1$, compute $T_{k+1}(j)$ for all possible states j according to equation (5.7).
- **Backward Recursion**: Computation of the optimal sequence of states.
  - Let $\hat{x}_n$ be the state j for which $T_n(j)$ is maximised.
  - For $k = n-1, n-2, \ldots, 0$, let $\hat{x}_k = B_{k+1}(\hat{\mathbf{x}}_{k+1})$.

In other words, the backward recursion first identifies the final state of the optimal state sequence. Then, the next to final one can be determined as the state that maximises the probability of sequences ending with the the now known final state, and so forth. Hence, the algorithm requires storage of all the $T_k(j)$ computed during the forward recursion.

In cases where there is no unique optimal state i, there may be no unique optimal state sequence either and $B_{k+1}(\hat{x}_{k+1})$ can, then, be taken arbitrarily within the set of maximising indices i.

5.3 Maximum Likelihood Estimation

Estimation of the weight distribution, as discussed for finite mixture models, is substituted in HMMs by estimation of the transition probability matrix and the initial distribution of the underlying Markov chain. While inference on the transition probabilities p_{ij} presents no challenge, there are several choices to be considered concerning the estimation of the initial distribution, in particular. The first option is to simply consider that $\boldsymbol{\nu}$ is fixed and known with no need to be estimated whatsoever.

A second choice would be to consider that $\boldsymbol{\nu}$ is fully determined by the parameter $\boldsymbol{\vartheta} = (\mathbf{P}, \boldsymbol{\theta})$. A typical example of this is assuming that $\boldsymbol{\nu}$ is the stationary distribution associated with the transition matrix $\mathbf{P}$, if it exists. Obviously, this is a particularly attractive alternative in the case of ergodic HMMs, in which the Markov chain generally admits a unique stationary distribution. However, this option is generally practicable only in the simplest of models, because of the lack of analytical expressions relating the stationary distribution to the transition matrix for general parametrised underlying Markov chains.

The last alternative would be to consider $\boldsymbol{\nu}$ as an independent parameter inside the vector $\boldsymbol{\vartheta}$ and aim to estimate it separately from the other parameters of the model. However, because we mainly consider estimation of the HMM parameter vector from a single long sequence of observations, there is no hope to estimate $\boldsymbol{\nu}$ consistently, since there is only one random variable $\mathbf{X}_0$ drawn from this density and it is not even observed. As a result, this option is much more appealing in left-to-right HMMs, where the model is estimated from several independent sequences of observations and the initial distribution is often a key parameter. Furthermore, handling the case of multiple observational sequences is straightforward, since the quantities corresponding to different sequences simply need to be added together, thanks to the independence assumption.

5.3.1 The Baum-Welch Algorithm

In the following, we are going to assume the last alternative and consider the initial distribution as an independent parameter to be estimated. For the finite state-space HMM under consideration, the complete-data likelihood in (2.6) may be rewritten as

$$L(\boldsymbol{\vartheta}|\mathbf{y}, \mathbf{x}) = \prod_{i=1}^{r} \nu_i^{\mathbb{1}\{x_0=i\}} \cdot \prod_{i=1}^{r}\prod_{j=1}^{r}\prod_{k=0}^{n-1} p_{ij}^{\mathbb{1}\{x_k=i, x_{k+1}=j\}} \cdot \prod_{i=1}^{r}\prod_{k=0}^{n} f(\mathbf{y}_k; \boldsymbol{\theta}_i)^{\mathbb{1}\{x_k=i\}}.$$

Evaluation of the intermediate quantity of EM demonstrates that, in great generality, the only quantities required for the algorithm are the marginal and bivariate

smoothing distributions, given the parameter vector $\boldsymbol{\vartheta}^{(\ell)}$, which may be computed using the Forward-Backward approach presented in the previous section. Moreover, estimation of the initial distribution, the transition probabilities and the state-specific parameters $\boldsymbol{\theta}$ can all be carried out separately within the M-step. Indeed, the intermediate quantity of EM assumes the following additive structure

$$\begin{aligned} Q\left(\boldsymbol{\vartheta};\boldsymbol{\vartheta}^{(\ell)}\right) = \sum_{i=1}^{r} \phi_{0|n}\left(i;\boldsymbol{\vartheta}^{(\ell)}\right) \log \nu_i + \sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{k=0}^{n-1} \phi_{k:k+1|n}\left(i,j;\boldsymbol{\vartheta}^{(\ell)}\right) \log p_{ij} \\ + \sum_{i=1}^{r}\sum_{k=0}^{n} \phi_{k|n}\left(i;\boldsymbol{\vartheta}^{(\ell)}\right) \log f(\mathbf{y}_k;\boldsymbol{\theta}_i). \end{aligned} \tag{5.8}$$

In order to maximise the intermediate quantity with respect to $\boldsymbol{\nu}$, we have to take into consideration the constraint $\nu_1 + \nu_2 + \cdots + \nu_r = 1$ and introduce the Lagrange multiplier λ. Then, we derive the Lagrangian to obtain

$$\nu_i^{(\ell+1)} = \frac{\phi_{0|n}\left(i;\boldsymbol{\vartheta}^{(\ell)}\right)}{\lambda}, \quad i = 1, 2, \ldots, r.$$

Normalisation of these probabilities leads to $\lambda = 1$ and so the updated estimates for the initial distribution are

$$\nu_i^{(\ell+1)} = \phi_{0|n}\left(i;\boldsymbol{\vartheta}^{(\ell)}\right), \quad i = 1, 2, \ldots, r. \tag{5.9}$$

Quite similarly, we introduce the Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_r$ which correspond to the equality constraints $p_{i1} + p_{i2} + \cdots + p_{ir} = 1$ for $i = 1, 2, \ldots, r$. This time, derivation of the Lagrangian yields

$$p_{ij}^{(\ell+1)} = \frac{1}{\lambda_i} \sum_{k=0}^{n-1} \phi_{k:k+1|n}\left(i,j;\boldsymbol{\vartheta}^{(\ell)}\right), \quad i, j = 1, 2, \ldots, r.$$

Summation over all possible states j leads to the computation of the Lagrange multipliers in the following way

$$1 = \sum_{j=1}^{r} p_{ij}^{(\ell+1)} = \frac{1}{\lambda_i} \sum_{k=0}^{n-1}\sum_{j=1}^{r} \phi_{k:k+1|n}\left(i,j;\boldsymbol{\vartheta}^{(\ell)}\right) = \frac{1}{\lambda_i} \sum_{k=0}^{n-1} \phi_{k|n}\left(i;\boldsymbol{\vartheta}^{(\ell)}\right), i = 1, 2, \ldots, r,$$

which in turn lead to the updated estimates for the transition probabilities

$$p_{ij}^{(\ell+1)} = \frac{\sum_{k=0}^{n-1} \phi_{k:k+1|n}\left(i,j;\boldsymbol{\vartheta}^{(\ell)}\right)}{\sum_{k=0}^{n-1} \phi_{k|n}\left(i;\boldsymbol{\vartheta}^{(\ell)}\right)}, \quad i, j = 1, 2, \ldots, r. \tag{5.10}$$

These equations are emblematic of the intuitive form taken by the parameter update formulas derived through the EM strategy. These equations are indeed the maximum likelihood equations for the complete model, except that the functions $\mathbb{1}\{X_k = i\}$ and $\mathbb{1}\{X_k = i, X_{k+1} = j\}$ are replaced by their conditional expectations, given the actual observations and the available parameter estimate $\boldsymbol{\vartheta}^{(\ell)}$. Of course, this behaviour fundamentally displays itself in cases where the probability density functions associated with the complete model form an exponential family.

The Baum-Welch algorithm essenitally consists of iterating the following steps, given an initial guess $\boldsymbol{\vartheta}^{(0)}$ for the model parameters

E-step: Run a Forward-Backward algorithm to calculate the marginal, $\phi_{k|n}\left(i; \boldsymbol{\vartheta}^{(\ell)}\right)$, and bivariate, $\phi_{k:k+1|n}\left(i, j; \boldsymbol{\vartheta}^{(\ell)}\right)$, smoothing distributions for all possible values of i, j, k.

M-step: Calculate $\boldsymbol{\nu}^{(\ell+1)}$ according to equation (5.9), $\mathbf{P}^{(\ell+1)}$ according to equation (5.10) and $\boldsymbol{\theta}^{(\ell+1)} = \underset{\boldsymbol{\theta}\in\Theta}{\arg\max}\, Q\left(\boldsymbol{\vartheta}; \boldsymbol{\vartheta}^{(\ell)}\right)$.

The algorithm is terminated, when $\ell\left(\boldsymbol{\theta}^{(L+1)}\right) - \ell\left(\boldsymbol{\theta}^{(L)}\right) < \epsilon$ for a certain index L, where ϵ is the predetermined precision of the estimation. After the termination of the algorithm, one may implement the Viterbi algorithm to calculate the maximum a posteriori state sequence $\hat{\mathbf{x}}$, given parameter vector $\boldsymbol{\vartheta}^{(L+1)}$.

Example 5.1 ***(Baum-Welch for Normal Hidden Markov Models)*** *In the Normal HMM, the additive term of the intermediate quantity which corresponds to the probability density function of each observation takes the form*

$$-\sum_{i=1}^{r}\sum_{k=0}^{n}\phi_{k|n}\left(i; \boldsymbol{\vartheta}^{(\ell)}\right)\left[\frac{\log(2\pi)}{2} + \frac{\log\sigma_i^2}{2} + \frac{(y_k - \mu_i)^2}{2\sigma_i^2}\right].$$

Derivation with respect to μ_i and σ_i leads to the following update formulas for the means and the variances

$$\mu_i^{(\ell+1)} = \frac{1}{N_i^{(\ell)}}\sum_{k=0}^{n}\phi_{k|n}\left(i; \boldsymbol{\vartheta}^{(\ell)}\right) y_k, \quad i = 1, 2, \ldots, r, \tag{5.11}$$

$$\left(\sigma_i^2\right)^{(\ell+1)} = \frac{1}{N_i^{(\ell)}}\sum_{k=0}^{n}\phi_{k|n}\left(i; \boldsymbol{\vartheta}^{(\ell)}\right)\left(y_k - \mu_i^{(\ell+1)}\right)^2, \quad i = 1, 2, \ldots, r, \tag{5.12}$$

where $N_i^{(\ell)} = \sum_{k=0}^{n}\phi_{k|n}\left(i; \boldsymbol{\vartheta}^{(\ell)}\right)$. ■

5.3.2 The Viterbi Training Algorithm

Viterbi training constitutes a fairly popular HMM method, which provides algorithms for parameter estimation. This is primarily due to the fact that Viterbi training is readily implemented if the Viterbi algorithm is used to generate predictions. Similar to the Baum-Welch algorithm, Viterbi training is an iterative estimation procedure. Unlike the Baum-Welch algorithm, however, which weighs all possible state paths for a given observational sequence in each iteration, Viterbi training only considers a single state path, namely the maximum a posteriori state sequence, when deriving new sets of parameters. In each iteration, a new set of parameter values is derived from the counts of allocations and transitions of the observations in the Viterbi path.

Given the current predicted Viterbi path $\hat{\mathbf{x}}$, we revert to complete-data estimation of the model parameters, by maximising the complete-data log-likelihood $\ell(\mathbf{y}, \hat{\mathbf{x}}; \boldsymbol{\vartheta})$ given by

$$\begin{aligned}\ell(\boldsymbol{\vartheta}|\mathbf{y}, \hat{\mathbf{x}}) = \sum_{i=1}^{r} \mathbb{1}\{\hat{x}_0 = i\} \log \nu_i + \sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{k=0}^{n-1} \mathbb{1}\{\hat{x}_k = i, \hat{x}_{k+1} = j\} \log p_{ij} \\ + \sum_{i=1}^{r}\sum_{k=0}^{n} \mathbb{1}\{\hat{x}_k = i\} \log f(\mathbf{y}_k; \boldsymbol{\theta}_i).\end{aligned} \tag{5.13}$$

As stated beforehand, this complete-data estimation leads to the exact same update formulas as the Baum-Welch algorithm, with the marginal and bivariate smoothing distributions giving place to the corresponding indicator variables. Indeed,

$$\nu_i^{(\ell+1)} = \mathbb{1}\left\{\hat{x}_0^{(\ell)} = i\right\}, \quad i = 1, 2, \ldots, r, \tag{5.14}$$

$$p_{ij}^{(\ell+1)} = \frac{\sum_{k=0}^{n-1} \mathbb{1}\left\{\hat{x}_k^{(\ell)} = i, \hat{x}_{k+1}^{(\ell)} = j\right\}}{\sum_{k=0}^{n-1} \mathbb{1}\left\{\hat{x}_k^{(\ell)} = i\right\}}, \quad i, j = 1, 2, \ldots, r. \tag{5.15}$$

The Viterbi training algorithm begins with an initial guess $\boldsymbol{\vartheta}^{(0)}$ for the model parameters and iterates the following steps

- Run the Viterbi algorithm, given the parameter vector $\boldsymbol{\vartheta}^{(\ell)}$, to calculate the maximum a posteriori state sequence, $\hat{\mathbf{x}}^{(\ell)}$.
- Calculate $\boldsymbol{\nu}^{(\ell+1)}$ according to equation (5.14), $\mathbf{P}^{(\ell+1)}$ according to equation (5.15) and $\boldsymbol{\theta}^{(\ell+1)} = \arg\max_{\boldsymbol{\theta}\in\Theta} \ell\left(\boldsymbol{\vartheta}|\mathbf{y}, \hat{\mathbf{x}}^{(\ell)}\right)$.

The iterations are terminated as soon as the Viterbi paths of successive estimations no longer change. The Viterbi training algorithm sacrifices some of Baum-Welch's generality for computational efficiency and so, leads to much faster computation times. In general, the Baum-Welch algorithm will give parameters that lead to better performance, although there are examples where this is not the case.

Example 5.2 ***(Viterbi Training for Normal Hidden Markov Models)*** *The complete-data estimation of the parameters of the Normal HMM, entailed in the Viterbi training algorithm, leads to the usual maximum likelihood estimators for the Normal distribution*

$$\mu_i^{(\ell+1)} = \frac{1}{N_i^{(\ell)}} \sum_{k:\hat{x}_k^{(\ell)}=i}^{n} y_k, \quad i = 1, 2, \ldots, r, \tag{5.16}$$

$$\left(\sigma_i^2\right)^{(\ell+1)} = \frac{1}{N_i^{(\ell)}} \sum_{k:\hat{x}_k^{(\ell)}=i}^{n} \left(y_k - \mu_i^{(\ell+1)}\right)^2, \quad i = 1, 2, \ldots, r, \tag{5.17}$$

where $N_i^{(\ell)} = \sum_{k=0}^{n} \mathbb{1}\left\{\hat{x}_k^{(\ell)} = i\right\}$ *counts the number of observations allocated to state* i*.* ■

5.4 Fully Bayesian Approaches

This section covers the fully Bayesian processing of HMMs, which means that, besides the hidden states and their conditional distributions, the model parameters are also assigned prior distributions.

We begin by rewriting, the complete-data likelihood in a way that makes it more convenient to combine it with a suitable prior distribution on parameter vector $\boldsymbol{\vartheta}$, as follows

$$f(\mathbf{y}, \mathbf{x}|\boldsymbol{\vartheta}) = \prod_{i=1}^{r} \nu_i^{n_i(x_0)} \cdot \prod_{i=1}^{r}\prod_{j=1}^{r} p_{ij}^{N_{ij}(\mathbf{x})} \cdot \prod_{i=1}^{r}\prod_{k:x_k=i} f(\mathbf{y}_k; \boldsymbol{\theta}_i), \tag{5.18}$$

where $n_i(x_0) = \mathbb{1}\{x_0 = i\}$ is counting the number of initial observations allocated to state i and $N_{ij}(\mathbf{x}) = \sum_{k=0}^{n-1} \mathbb{1}\{x_k = i, x_{k+1} = j\}$ the number of transitions from state i to state j.

In the specific set-up of HMMs, there are typically three separate entities within the parameter vector $\boldsymbol{\vartheta}$. That is, it can be decomposed as $\boldsymbol{\vartheta} = (\mathbf{P}, \boldsymbol{\nu}, \boldsymbol{\theta})$, where $\mathbf{P}$ parametrises the transition distribution, $\boldsymbol{\nu}$ parametrises the initial distribution and $\boldsymbol{\theta}$ parametrises the conditional distribution of $\mathbf{Y}$ given $\mathbf{X}$. When conditioned on the latent chain, the parameter vector $\boldsymbol{\theta}$ is estimated as in a regular, non-latent model,

whereas the parameters $\mathbf{P}, \boldsymbol{\nu}$ only depend on the chain $\mathbf{X}$. Furthermore, given $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{P}$ and $\boldsymbol{\theta}$ are conditionally independent.

For the time being, we are going to entertain the possibility that $\mathbf{X}_0$ is random, unknown and $\boldsymbol{\nu}$ is parametrised by a separate parameter from $\mathbf{P}$. As a result, Bayesian inference about $\mathbf{P}$, $\boldsymbol{\nu}$ and $\boldsymbol{\theta}$ can be conducted separately, conditional on the latent chain, leading to the formulation of a Gibbs sampler for this model.

In the case where the latent variables are finite-valued, the Dirichlet distribution is a conjugate prior for the rows of the transition probability matrix $\mathbf{P}$ of the latent Markov chain in the following sense. Assuming that each row of $\mathbf{P}$ has a prior distribution that is Dirichlet, i.e. $(p_{i1}, p_{i2}, \ldots, p_{ir}) \sim \text{Dir}(a_1, a_2, \ldots, a_r)$, with the rows being a priori independent, then, given $\mathbf{x}$, the rows are conditionally independent and

$$\pi(p_{i1}, p_{i2}, \ldots, p_{ir}|\mathbf{x}) \propto \prod_{j=1}^{r} p_{ij}^{N_{ij}(\mathbf{x})+a_j-1}, \quad i = 1, 2, \ldots, r. \tag{5.19}$$

Furthermore, the unknown initial distribution $\boldsymbol{\nu}$ may also be equipped with a $\text{Dir}(e_1, e_2, \ldots, e_r)$ prior, usually with all e_i equal. Conditional on the initial value x_0 of the Markov chain, the conditional posterior distribution of $\boldsymbol{\nu}$ is given by

$$\pi(\boldsymbol{\nu}|x_0) \propto \prod_{i=1}^{r} \nu_i^{n_i(x_0)+e_i-1}. \tag{5.20}$$

Of course, the state-specific parameter vector $\boldsymbol{\theta}$ is updated by a Gibbs sampler conditional on the underlying Markov chain and the observations. Consequently, Bayesian estimation of these state-specific parameters is identical to the case of finite mixture models, where the hidden allocations are replaced by the hidden states of the Markov chain. Indeed, we may fit the following hierarchical prior on the parameter vector $\boldsymbol{\theta}$

$$\pi(\boldsymbol{\theta}, \boldsymbol{\delta}) = \pi(\boldsymbol{\delta}) \prod_{i=1}^{r} \pi(\boldsymbol{\theta}_i|\boldsymbol{\delta}), \tag{5.21}$$

which, upon combination with the complete-data likelihood, $f(\mathbf{y}, \mathbf{x}|\boldsymbol{\vartheta})$, leads to the following conditional posterior distributions for the parameters $\boldsymbol{\theta}_i$ and the hyper-parameter $\boldsymbol{\delta}$

$$\pi(\boldsymbol{\theta}_i|\mathbf{y}, \mathbf{x}, \boldsymbol{\delta}) \propto \pi(\boldsymbol{\theta}_i|\boldsymbol{\delta}) \prod_{k:x_k=i} f(\mathbf{y}_k|\boldsymbol{\theta}_i), \quad i = 1, 2, \ldots, r, \tag{5.22}$$

$$\pi(\boldsymbol{\delta}|\boldsymbol{\theta}) \propto \pi(\boldsymbol{\delta}) \prod_{i=1}^{r} \pi(\boldsymbol{\theta}_i|\boldsymbol{\delta}). \tag{5.23}$$

Combining all the previously formulated priors on the separate entities of the parameter vector $\boldsymbol{\vartheta}$, leads to the formulation of the joint prior of $(\boldsymbol{\vartheta}, \boldsymbol{\delta})$ as

$$\pi(\boldsymbol{\vartheta}, \boldsymbol{\delta}) \propto \pi(\boldsymbol{\delta}) \cdot \prod_{i=1}^{r} \nu_i^{e_i - 1} \cdot \prod_{i=1}^{r} \left[\prod_{j=1}^{r} p_{ij}^{a_j - 1} \right] \cdot \prod_{i=1}^{r} \pi(\boldsymbol{\theta}_i | \boldsymbol{\delta}). \tag{5.24}$$

Under this prior choice, the complete-data posterior $\pi(\boldsymbol{\vartheta}, \boldsymbol{\delta}, \mathbf{x}|\mathbf{y})$ factorises in the same convenient way as the complete-data likelihood function, i.e.

$$\begin{aligned} \pi(\boldsymbol{\vartheta}, \boldsymbol{\delta}, \mathbf{x}|\mathbf{y}) \propto \pi(\boldsymbol{\delta}) \cdot \prod_{i=1}^{r} \nu_i^{n_i(x_0) + e_i - 1} \cdot \prod_{i=1}^{r} \left[\prod_{j=1}^{r} p_{ij}^{N_{ij}(\mathbf{x}) + a_j - 1} \right] \\ \cdot \prod_{i=1}^{r} \left[\pi(\boldsymbol{\theta}_i | \boldsymbol{\delta}) \prod_{k: x_k = i} f(\mathbf{y}_k | \boldsymbol{\theta}_i) \right]. \end{aligned} \tag{5.25}$$

5.4.1 The Gibbs Sampler with Local Updating

An earlier and more rudimentary version of the Gibbs sampler entails the update of only one hidden variable X_k at a time. This is referred to as local updating of the hidden chain, because each state x_k is updated conditional upon its neighbours only. The conditional posterior distribution $\pi(x_k|\mathbf{y}, \boldsymbol{\vartheta}, \mathbf{x}_{-k})$ reduces to

$$\begin{aligned} &\pi(x_0|\mathbf{y}_0, \boldsymbol{\vartheta}, x_1) \propto \nu(x_0) p(x_0, x_1) f(\mathbf{y}_0; \boldsymbol{\theta}_{x_0}), \\ &\pi(x_k|\mathbf{y}_k, \boldsymbol{\vartheta}, x_{k-1}, x_{k+1}) \propto p(x_{k-1}, x_k) p(x_k, x_{k+1}) f(\mathbf{y}_k; \boldsymbol{\theta}_{x_k}), \quad k = 1, 2, \ldots, n-1, \\ &\pi(x_n|\mathbf{y}_n, \boldsymbol{\vartheta}, x_{n-1}) \propto p(x_{n-1}, x_n) f(\mathbf{y}_n; \boldsymbol{\theta}_{x_n}). \end{aligned} \tag{5.26}$$

The Bayesian analysis conducted in this section, in conjunction with this local updating of the hidden chain, leads to the formulation of the following Gibbs sampler, which starts with some initial values $\left(\mathbf{x}^{(0)}, \boldsymbol{\delta}^{(0)}\right)$ and repeats the following

- Sample the rows of $\mathbf{P}^{(\ell+1)}$ from the complete-data conditional posterior distributions Dir $\left(N_{i1}\left(\mathbf{x}^{(\ell)}\right) + a_1, N_{i2}\left(\mathbf{x}^{(\ell)}\right) + a_2, \ldots, N_{ir}\left(\mathbf{x}^{(\ell)}\right) + a_r\right)$, given by (5.19).
- Sample the initial distribution $\boldsymbol{\nu}^{(\ell+1)}$ from the conditional posterior distribution Dir $\left(n_1(x_0^{(\ell)}) + e_1, n_2(x_0^{(\ell)}) + e_2, \ldots, n_r(x_0^{(\ell)}) + e_r\right)$, given by (5.20).
- Sample the parameters $\boldsymbol{\theta}_1^{(\ell+1)}, \boldsymbol{\theta}_2^{(\ell+1)}, \ldots, \boldsymbol{\theta}_r^{(\ell+1)}$ independently from the conditional posterior distributions $\pi\left(\boldsymbol{\theta}_i \left| \mathbf{y}, \mathbf{x}^{(\ell)}, \boldsymbol{\delta}^{(\ell)} \right.\right)$, given by (5.22).
- Sample the hyper-parameter $\boldsymbol{\delta}^{(\ell+1)}$ from $\pi\left(\boldsymbol{\delta} \left| \boldsymbol{\theta}^{(\ell+1)} \right.\right)$, given by (5.23).
- Sample the states $x_0^{(\ell+1)}, x_1^{(\ell+1)}, \ldots, x_n^{(\ell+1)}$ from the conditional posterior distributions $\pi\left(x_k \left| \mathbf{y}, \boldsymbol{\vartheta}^{(\ell+1)}, \mathbf{x}_{0:k-1}^{(\ell+1)}, \mathbf{x}_{k+1:n}^{(\ell)} \right.\right)$, given by (5.26).

5.4.2 The Gibbs Sampler with Global Updating

The simplest version of the Gibbs sampler, which will be referred to as global updating of the hidden chain, replaces the last step of the previous algorithm with one that updates the trajectory of the hidden chain as a whole from its joint conditional distribution, given by equation (5.6). Obviously, sampling from this distribution requires a Forward-Backward recursive scheme similar to the one implemented in the Baum-Welch algorithm.

To be precise, the Forward Filtering remains as it is, whereas the Markovian Backward Smoothing is replaced by a Backward sampling scheme, as follows

- **Forward Recursion**: Compute and store the forward filtering distributions $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n$ according to the Forward Filtering algorithm.
- **Backward Simulation**: Sample
  - The final state x_n from $\boldsymbol{\phi}_n$;
  - The rest of the states x_k, for $k = n-1, n-2, \ldots, 0$, from the backward transition distribution $B_k(x_{k+1}, i)$, given by (5.5).

The backward simulation pass in this algorithm is much simpler than its smoothing counterpart in the Markovian Backward Smoothing algorithm, as one is not required to either evaluate $B_k(j, i)$ for all possible states j or compute the marginal and bivariate smoothing distributions.

The local updating of the Markov chain is simpler and less time-consuming to implement than the Backward Simulation described in this section, due to the necessity of computing all the filtering distributions. Especially in models where the number of states is very large, it may be the case that implementing the Backward Simulation is overwhelming, whereas the local updating of the underlying chain is still feasible.

On the other hand, the Monte Carlo simulations obtained by this algorithm are independent, which is not the case for those produced by the Gibbs sampler with local updating. As a result, the Gibbs sampler with local updating should mix and explore the posterior surface much more slowly than when global updating is implemented. It is thus difficult to make a firm recommendation on which updating scheme to use.

5.4.3 A Metropolis-Hastings Step for Stationary Markov Chains

For a stationary latent Markov chain, the initial distribution usually is equal to the stationary distribution and, thus, depends directly on the transition matrix. Gibbs sampling from the conditional posteriors $\pi(p_{i1}, p_{i2}, \ldots, p_{ir}|\mathbf{x})$ is no longer feasible,

since the rows of $\mathbf{P}$ are no longer independent a posteriori, due to the joint dependence on $\boldsymbol{\nu}$. The joint conditional posterior $\pi(\mathbf{P}|\mathbf{x})$ takes the form

$$\pi(\mathbf{P}|\mathbf{x}) \propto \nu(x_0) \prod_{i=1}^{r} \left[\prod_{j=1}^{r} p_{ij}^{N_{ij}(\mathbf{x})+a_j-1} \right], \quad i = 1, 2, \ldots, r. \tag{5.27}$$

To sample $\mathbf{P}$, one could replace the first step of the Gibbs sampler by a Metropolis-Hastings step with Dir $\left(N_{i1}\left(\mathbf{x}^{(\ell)}\right) + a_1, N_{i2}\left(\mathbf{x}^{(\ell)}\right) + a_2, \ldots, N_{ir}\left(\mathbf{x}^{(\ell)}\right) + a_r\right)$ being the proposal density for the i^{th} row. Starting from the current transition matrix $\mathbf{P}^{(\ell)}$, a new transition matrix $\mathbf{P}^{\text{new}}$ is proposed, by drawing all rows from the aforementioned Dirichlet proposal density, denoted by $q(\mathbf{P}|\mathbf{x})$. The acceptance rate for this Metropolis-Hastings step is equal to $\min\{1, A\}$, where

$$A = \frac{\pi(\mathbf{P}^{\text{new}}|\mathbf{x}^{(\ell)}) q(\mathbf{P}^{(\ell)}|\mathbf{x}^{(\ell)})}{\pi(\mathbf{P}^{(\ell)}|\mathbf{x}^{(\ell)}) q(\mathbf{P}^{\text{new}}|\mathbf{x}^{(\ell)})} = \frac{\nu^{\text{new}}(x_0^{(\ell)})}{\nu^{(\ell)}(x_0^{(\ell)})}. \tag{5.28}$$

Drawing $U \sim \mathcal{U}(0,1)$, if $U < \min\{1, A\}$, we accept $\mathbf{P}^{\text{new}}$ and set $\mathbf{P}^{(\ell+1)} = \mathbf{P}^{\text{new}}$. Otherwise, we reject $\mathbf{P}^{\text{new}}$ and set $\mathbf{P}^{(\ell+1)} = \mathbf{P}^{(\ell)}$.

Of course, sampling from the initial distribution, in the second step of the Gibbs sampler, is, then, also replaced by an analytic computation of the stationary distribution of the transition matrix $\mathbf{P}^{\ell+1}$, as the sole normalised eigenvector which corresponds to the eigenvalue 1.

5.5 Maximum a Posteriori Estimation

Rather than simulating from the posterior distribution of the parameters, we now consider maximising it to determine the so-called maximum a posteriori point estimate. In contrast to the Baum-Welch and Viterbi Training methods, which could also be used in this context, the techniques to be discussed explicitly use parameter simulation, in addition to hidden state simulation. The primary objective of these techniques is not only to compensate for the lack of exact smoothing computations in many models of interest, but also to perform some form of random search optimisation, which is hopefully more robust to the presence of local maxima.

Simulated annealing is a non-homogeneous variant of MCMC algorithms used to perform global optimisation, that is, convergence to the global maxima of the function of interest. It is a random search technique, which explores the parameter space, using a non-homogeneous Markov chain, whose transition kernels are tailored

to have invariant probability density functions $\pi_{M_\ell}(\boldsymbol{\vartheta}|\mathbf{y}) \propto [\pi(\boldsymbol{\vartheta}|\mathbf{y})]^{M_\ell}$, with $\{M_\ell\}_{\ell \geqslant 1}$ being a positive increasing sequence tending to infinity.

The intuition behind this technique is that, as M_ℓ tends to infinity, $[\pi(\boldsymbol{\vartheta}|\mathbf{y})]^{M_\ell}$ concentrates itself upon the set of global modes of the posterior distribution. It has been shown, under various assumptions, that convergence to the set of global maxima is, indeed, ensured for sequences $\{M_\ell\}_{\ell \geqslant 1}$ growing at a logarithmic rate. The sequence $\{M_\ell\}_{\ell \geqslant 1}$ is often called a cooling schedule.

For HMMs, the invariant density is only available in closed form in models where exact smoothing is feasible, such as HMMs with finite state-space. To overcome this difficulty, a novel approach named State Augmentation for Marginal Estimation has been developed as a multiple-imputation Metropolis version of the EM algorithm.

5.5.1 The State Augmentation for Marginal Estimation (SAME) Algorithm

The key argument behind SAME is that, upon restricting M_ℓ to be integers, the probability density function π_{M_ℓ} may be viewed as the marginal posterior in an artificially augmented probability model. Hence, one may use standard MCMC techniques to draw from this augmented probability model and implement the simulated annealing strategy for general missing data models. The concentrated distribution π_{M_ℓ} is obtained by artificially replicating the latent variables in the model.

To be precise, consider M artificial copies of the hidden state sequence, denoted by $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(M)$. The model postulates that these sequences are independent with common parameter vector $\boldsymbol{\vartheta}$ and observed sequence $\mathbf{y}$, leading to the posterior

$$\pi_M\left(\boldsymbol{\vartheta}, \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(M) | \, \mathbf{y}\right) \propto \prod_{m=1}^{M} \pi\left(\boldsymbol{\vartheta}, \mathbf{x}(m) | \, \mathbf{y}\right). \tag{5.29}$$

This distribution does not correspond to a real phenomenon, but is a properly defined density, in that it is positive and the right-hand side can be normalised, so that it integrates to unity. Now, the marginal posterior distribution of $\boldsymbol{\vartheta}$ is obtained by integration over all replications of $\mathbf{x}$, as

$$\pi_M(\boldsymbol{\vartheta}|\mathbf{y}) \propto \prod_{m=1}^{M} \int \pi\left(\boldsymbol{\vartheta}, \mathbf{x}(m) | \, \mathbf{y}\right) d\mathbf{x}(m) = [\pi(\boldsymbol{\vartheta}|\mathbf{y})]^M. \tag{5.30}$$

As a result, an MCMC algorithm in the augmented space with invariant distribution $\pi_{M_\ell}\left(\boldsymbol{\vartheta}, \mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(M) | \, \mathbf{y}\right)$ is such that the simulated sequence of parameters $\left\{\boldsymbol{\vartheta}^\ell\right\}_{\ell \geqslant 1}$ marginally admits $\pi_{M_\ell}(\boldsymbol{\vartheta}|\mathbf{y})$ as invariant distribution.

An important point here is that, when an MCMC sampler is available for the density $\pi(\boldsymbol{\vartheta}, \mathbf{x}|\mathbf{y})$, it is easy to also construct an MCMC sampler with target density (5.29), as the replications of $\mathbf{x}$ are independent, given $\boldsymbol{\vartheta}$. Indeed,

$$\pi_M\left(\mathbf{x}(1), \ldots, \mathbf{x}(M) | \, \mathbf{y}, \boldsymbol{\vartheta}\right) = \prod_{m=1}^{M} \pi\left(\mathbf{x}(m) | \, \mathbf{y}, \boldsymbol{\vartheta}\right), \tag{5.31}$$

$$\pi_M\left(\boldsymbol{\vartheta} \, | \mathbf{y}, \mathbf{x}(1), \ldots, \mathbf{x}(M)\right) \propto \prod_{m=1}^{M} \pi\left(\boldsymbol{\vartheta} \, | \mathbf{y}, \mathbf{x}(m)\right) \propto [\pi(\boldsymbol{\vartheta})]^M \prod_{m=1}^{M} f\left(\mathbf{y}, \mathbf{x}(m) | \, \boldsymbol{\vartheta}\right). \tag{5.32}$$

According to (5.31), the sampling step for $\mathbf{x}(m)$ is identical to its counterpart in a standard data augmentation sampler with target distribution $\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\vartheta})$. Additionally, if $\pi(\boldsymbol{\vartheta}|\mathbf{y}, \mathbf{x})$ belongs to an exponential family of distributions, then (5.32) is also a member of this exponential family, so sampling from it is straightforward. In other cases, the vector $\boldsymbol{\vartheta}$ can be simulated using a Metropolis-Hastings step.

Assuming the same conjugate Dirichlet priors on the rows of the transition matrix, as in the previous section, we find that the full-conditional distribution of $\mathbf{P}$ is such that the rows are conditionally independent for $i = 1, 2, \ldots, r$, with

$$\begin{aligned}&(p_{i1}, p_{i2}, \ldots, p_{ir}) \, |\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(M) \sim \\ &\quad \text{Dir}\left(\sum_{m=1}^{M} N_{i1}(\mathbf{x}(m)) + M(a_1 - 1) + 1, \ldots, \sum_{m=1}^{M} N_{ir}(\mathbf{x}(m)) + M(a_r - 1) + 1\right).\end{aligned} \tag{5.33}$$

For a stationary Markov chain, we could use a Metropolis-Hastings step, with (5.33) as the proposal density for the i^{th} row and acceptance rate $\min\{1, A\}$, where

$$A = \prod_{m=1}^{M} \frac{\nu^{\text{new}}(x_0^{(\ell)}(m))}{\nu^{(\ell)}(x_0^{(\ell)}(m))}. \tag{5.34}$$

Example 5.3 ***(SAME for Normal Hidden Markov Models)*** *Assuming the same conjugate priors, we find that the full-conditional posterior distributions are*

$$\mu_i \left| \boldsymbol{y}, \sigma_i^2, b, \boldsymbol{x}(1), \ldots, \boldsymbol{x}(M) \right. \sim \mathcal{N}\left(\frac{BS_i + Mb\sigma_i^2}{BN_i + M\sigma_i^2}, \frac{B\sigma_i^2}{BN_i + M\sigma_i^2}\right), \tag{5.35}$$

with $N_i = \sum_{m=1}^{M} \sum_{k=0}^{n} \mathbb{1}\{x_k(m) = i\}$ *and* $S_i = \sum_{m=1}^{M} \sum_{k:x_k(m)=i} y_k$, *whereas*

$$\sigma_i^2 \left| \, \boldsymbol{y}, \mu_i, C, \boldsymbol{x}(1), \ldots, \boldsymbol{x}(M) \right. \sim \textit{Inv-Gamma}\left(\frac{N_i}{2} + M(c+1) - 1, \frac{V_i}{2} + MC\right), \tag{5.36}$$

with $V_i = \sum_{m=1}^{M} \sum_{k:x_k(m)=i} (y_k - \mu_i)^2$ *for* $i = 1, 2, \ldots, r$. *Furthermore, the conditional posterior distributions of the hyper-parameters are given by*

$$C \,|\boldsymbol{\sigma}^2 \sim Gamma\left(M(rc+g-1)+1, \sum_{i=1}^{r} M\frac{}{\sigma_i^2} + MG\right), \tag{5.37}$$

$$b|\boldsymbol{\mu} \sim \mathcal{N}\left(\frac{1}{r}\sum_{i=1}^{r}\mu_i, \frac{B}{Mr}\right). \tag{5.38}$$

We initialise the algorithm with $\left(\boldsymbol{P}^{(0)}, \boldsymbol{\nu}^{(0)}, \boldsymbol{\mu}^{(0)}, \left(\boldsymbol{\sigma}^2\right)^{(0)}, C^{(0)}, b^{(0)}\right)$ *and select a cooling schedule* $\{M_\ell\}_{\ell \geqslant 1}$. *Then we iterate the following steps*

- *Sample the* $M_{\ell+1}$ *chain replications* $\boldsymbol{x}^{(\ell+1)}(1), \boldsymbol{x}^{(\ell+1)}(2), \ldots, \boldsymbol{x}^{(\ell+1)}(M_{\ell+1})$ *independently using the Forward Filtering-Backward Sampling recursion.*
- *Sample the rows of* $\boldsymbol{P}^{(\ell+1)}$ *independently according to (5.33).*
- *Sample the means* $\mu_1^{(\ell+1)}, \mu_2^{(\ell+1)}, \ldots, \mu_r^{(\ell+1)}$ *according to (5.35).*
- *Sample the variances* $\left(\sigma_1^2\right)^{(\ell+1)}, \left(\sigma_2^2\right)^{(\ell+1)}, \ldots, \left(\sigma_r^2\right)^{(\ell+1)}$ *according to (5.36).*
- *Sample the hyper-parameters* $C^{(\ell+1)}$ *and* $b^{(\ell+1)}$ *according to (5.37) and (5.38) respectively.*

It is of great interest that the above conditional posterior distributions, from which simulation is carried out in the SAME approach, all have variances that decrease proportionally to M^{-1}. *Hence, the distributions get more and more concentrated around their modes, as the number of replications increases.* ■

5.6 Identifiability Issues

For a hidden Markov model, there exists non-identifiability due to invariance to relabelling the states of the underlying Markov chain, as well as generic identifiability. An inequality constraint similar to the one discussed for finite mixture models, requiring that the state-specific parameters $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ differ in at least one element, will rule out the first identifiability issue.

One necessary condition for generic identifiability of a hidden Markov model consisting of the distributions $g_i(\boldsymbol{\theta})$ is that the corresponding finite mixture of $g_i(\boldsymbol{\theta})$ distributions is generically identifiable. A second necessary condition is that the latent Markov chain is irreducible and aperiodic. It is, however, not necessary to assume that $\mathbf{X}_0$ was drawn from the stationary distribution of the latent chain.

Chapter 6

Statistical Inference Under Model Specification Uncertainty

6.1 Introduction

The decision to fit a hidden Markov model to data will often result from careful consideration. Sometimes, however, alternatives to hidden Markov models will be available, which then should be compared with each other. Even if we stay within a certain family of models, we may face model specification problems, the most important being the choice of the number of states, r. If it is impossible to assign a value to r a priori with complete certainty, we are faced with the problem of estimating r from the data.

In many applications, it is of substantial interest to test hypotheses about r, most importantly, to test homogeneity ($r = 1$) against heterogeneity ($r > 1$). Testing for the number of states in a hidden Markov model is known to be a difficult problem, as it involves inference for an over-fitting model, where the true number of states is less than the number of states in the fitted hidden Markov model.

Many approaches have been proposed to deal with model specification uncertainty. Several informal methods for diagnosing HMMs have been explored, such as mode hunting in the sample histogram or diagnosing goodness-of-fit through implied moments or the predictive performance of the model. As far as formal model comparison is concerned, likelihood-based methods include, in particular, the likelihood ratio statistic, as well as the AIC and BIC information criteria.

On the other hand, there are basically two Bayesian approaches to deal with model specification uncertainty. One approach is to apply trans-dimensional Markov Chain Monte Carlo methods to obtain draws from the joint posterior density of the model indicator and the model parameters of all possible models. The second approach is

to compute the marginal likelihoods of all possible models and to apply Bayes' rule to quantify the posterior evidence in favour of each model.

6.2 Likelihood-Based Methods

This section provides a short review of various likelihood-based methods that have been used to deal with model uncertainty in HMMs. These methods play a central role in testing parametric models and, among these, likelihood ratio tests are usually the preferred ones. Calculation of the observed likelihood function, required for all these methods, is attained through the Forward Filtering algorithm.

6.2.1 The Generalised Likelihood Ratio (LR) Test

Application of the likelihood ratio tests to hidden Markov models creates some difficulty. Consider two nested hidden Markov model $\mathcal{M}_0$ and $\mathcal{M}_1$, with $\mathcal{M}_0$ being the simpler one. A standard approach for testing between nested models is to apply a generalised likelihood ratio test. First, the maximum likelihood estimators $\hat{\boldsymbol{\vartheta}}_0$ and $\hat{\boldsymbol{\vartheta}}_1$, as well as the corresponding likelihood functions are determined for both models. Then, the generalised likelihood ratio test statistic is defined as

$$LR = -2\left(\log f\left(\mathbf{y}; \hat{\boldsymbol{\vartheta}}_0\right) - \log f\left(\mathbf{y}; \hat{\boldsymbol{\vartheta}}_1\right)\right). \tag{6.1}$$

Under regularity conditions, the generalised likelihood ratio test statistic asymptotically follows a χ^2_ν distribution, under the assumption that model $\mathcal{M}_0$ is correct, with ν being equal to the number of constraints imposed on $\mathcal{M}_1$ to obtain $\mathcal{M}_0$. If the two hidden Markov models differ only in the parameter structure, but both assume the correct number of states, these regularity conditions typically hold and the LR statistic may be applied in a straightforward manner.

However, if $\mathcal{M}_0$ and $\mathcal{M}_1$ are stationary hidden Markov models with r and $r+1$ states respectively, then implementing the LR test immediately leads to ambiguity, because model $\mathcal{M}_0$ may be obtained from model $\mathcal{M}_1$ in more than one ways. The number of constraints is equal to $2r+1$, when imposing the constraints $p_{i,r+1} = 0$ for $i = 1, 2, \ldots, r+1$ and $p_{r+1,j} = 0$ for $j = 1, 2, \ldots, r+1$ on $\mathcal{M}_1$, whereas it is equal to $\dim(\boldsymbol{\theta})$, when imposing the constraints $\boldsymbol{\theta}_r = \boldsymbol{\theta}_{r+1}$ on $\mathcal{M}_1$.

Asymptotic theory has not lent itself to any practically useful numerical approximations to critical levels or p-values. Instead, the approach usually taken in the literature is bootstrapping the LR test. This bootstrapping can be either non-parametric or parametric.

6.2.2 Information Criteria

Using heuristic arguments on how to account for model complexity, Akaike proposed a general criterion for model selection, which is equivalent to choosing the model that minimises

$$AIC_r = -2 \log f\left(\mathbf{y}; \hat{\boldsymbol{\vartheta}}_r\right) + 2d_r, \tag{6.2}$$

where $d_r = r \dim(\boldsymbol{\theta}) + r(r-1)$ is equal to the dimension of the stationary hidden Markov model and acts as a correction term without which, one would simply choose the model that maximises the likelihood function. This correction term, introduces a severe penalty for high-dimensional models, which provide little additional fit, in terms of increasing the likelihood function in comparison to simpler models.

Using asymptotic expansions, rather than heuristics, Schwarz arrived at the conclusion to select the model for which

$$BIC_r = -2 \log f\left(\mathbf{y}; \hat{\boldsymbol{\vartheta}}_r\right) + d_r \log(n+1) \tag{6.3}$$

is minimised. Quantitatively, AIC and BIC differ only in the factor by which d_r is multiplied. Qualitatively, both criteria provide a mathematical formulation of the principle of parsimony in model building, although for large data sets their behaviour is rather different.

As the first term in both AIC and BIC measures the goodness-of-fit, whereas the second term penalises model complexity, one selects the model that minimises either AIC or BIC. For $n > e^2 - 1$, Akaike's criterion favours more complex models than Schwarz's criterion and has been shown to be inconsistent, choosing too complex models, even asymptotically.

6.3 Trans-Dimensional Markov Chain Monte Carlo

In general, a variable dimension model is, to quote Peter Green, "a model where one of the things you do not know is the number of things you do not know". In other words, it pertains to a statistical model where the dimension of the parameter space is unknown and must be estimated from the data.

More formally, a variable dimension model is defined as a collection of models $\mathcal{M}_r$, or equivalently parameter spaces Θ_r, for $r = 1, 2, \ldots, R$, associated with a collection of priors $\pi_r(\boldsymbol{\vartheta}_r)$ on these spaces and a prior distribution $\pi(r)$ on the indices of these spaces. In the following, we shall consider that a variable dimension model is associated with a probability distribution on the space $\Theta = \bigcup_{r=1}^{R} (\{r\} \times \Theta_r)$. An

element $\boldsymbol{\vartheta}$ of Θ may thus always be written as $\boldsymbol{\vartheta} = (r, \boldsymbol{\vartheta}_r)$, where $\boldsymbol{\vartheta}_r$ is an element of Θ_r. The prior on Θ will be denoted by $\pi(\boldsymbol{\vartheta}) = \pi(r, \boldsymbol{\vartheta}_r) = \pi(r)\pi_r(\boldsymbol{\vartheta}_r)$.

For HMMs, the space Θ_r is, in general, that of the parameters of HMMs with r states for the latent Markov chain. In the Bayesian framework, the dimension r of the model is treated as a usual parameter. The aim is to address the two problems of deciding which model is best and determining the parameters of the best fitting model simultaneously. This is achieved by deriving the posterior density $\pi(r, \boldsymbol{\vartheta}_r|\mathbf{y})$, using Bayes' theorem

$$\pi(r, \boldsymbol{\vartheta}_r|\mathbf{y}) \propto \pi(r)\pi_r(\boldsymbol{\vartheta}_r)f(\mathbf{y}|\boldsymbol{\vartheta}_r). \tag{6.4}$$

Interestingly, by integrating out the index part of the model, we simply arrive at a mixture representation of the observed likelihood and the predictive distribution

$$f(\mathbf{y}|\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \ldots, \boldsymbol{\vartheta}_R) = \sum_{r=1}^{R} \pi(r) f(\mathbf{y}|\boldsymbol{\vartheta}_r),$$

$$f(\mathbf{y}_{\text{new}}|\mathbf{y}) = \sum_{r=1}^{R} \left[\pi(r|\mathbf{y}) \int f(\mathbf{y}_{\text{new}}|\boldsymbol{\vartheta}_r)\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y}) d\boldsymbol{\vartheta}_r \right].$$

This mixture representation, referred to as model averaging in the Bayesian literature, is interesting, because it suggests the use of predictors that are not obtained by selecting a particular model from the R possible ones, but rather consists of taking all possible options into account simultaneously, weighting them by their posterior odds. The variability due to the selection of the model is thus accounted for.

Given a variable dimension model, there is an additional computational difficulty in simulating the posterior distribution, in that the sampler must move both within and between models $\mathcal{M}_r$. Although the former pertains to previous developments, the latter requires a sound measure-theoretic basis to lead to correct MCMC moves.

6.3.1 Reversible Jump Markov Chain Monte Carlo

The reversible jump technique is basically of Metropolis-Hastings type with specific trans-dimensional proposals, carefully designed to move between different models in a way that is consistent with the desired stationary distribution of the MCMC algorithm. The sampler jumps between different models by making moves from a current model $\mathcal{M}_r$ to a new model $\mathcal{M}_s$, while retaining detailed balance which ensures the correct limiting distribution, provided that the chain is irreducible and aperiodic.

To capture potential relations between $\boldsymbol{\vartheta}_r$ and $\boldsymbol{\vartheta}_s$, with $r < s$, the Metropolis-Hastings algorithm proposes values for $\boldsymbol{\vartheta}_s$ via a mapping $\boldsymbol{\vartheta}_s = \mathbf{g}(\boldsymbol{\vartheta}_r, \mathbf{u})$, that also

depends on some auxiliary random vector $\mathbf{u}$. The only requirement is that $\mathbf{g}$ is differentiable with an inverse mapping $\mathbf{g}^{-1}$ which is also differentiable.

As a general rule, when moving to the higher dimension model $\mathcal{M}_s$, the auxiliary random vector needs to be drawn from a non-degenerate proposal density $q(\mathbf{u})$ with $\dim(\mathbf{u}) = \dim(\boldsymbol{\vartheta}_s) - \dim(\boldsymbol{\vartheta}_r)$. This random vector is, then, used to construct $\boldsymbol{\vartheta}_s$ from $\boldsymbol{\vartheta}_r$ through the mapping $\mathbf{g}$. The reverse move, from $\mathcal{M}_s$ to $\mathcal{M}_r$, is deterministic and simply consists of jumping back to $(\boldsymbol{\vartheta}_r, \mathbf{u}) = \mathbf{g}^{-1}(\boldsymbol{\vartheta}_s)$. We shall assume that, when in model $\mathcal{M}_r$, the move to $\mathcal{M}_s$ is attempted with probability $P_{r,s}$.

For reasons of simplicity, in what follows, we are going to assume that the initial distribution of the underlying Markov chain is fixed and known. In the case of the probability transition matrix $\mathbf{P}$, the moves may prove to be quite complex, due to it being a stochastic matrix. One way to overcome this difficulty would be to re-parametrise each row $(p_{i1}, p_{i2}, \ldots, p_{ir})$ as $(q_{i1}, q_{i2}, \ldots, q_{ir})$ with

$$p_{ij} = \frac{q_{ij}}{\sum_{\ell=1}^{r} q_{i\ell}}, \quad i, j = 1, 2, \ldots, r, \tag{6.5}$$

so that the summation constraints on the rows of $\mathbf{P}$ do not hinder the move from one model to another. Obviously, the q_{ij} are not identifiable, but, as we are only interested in the p_{ij}, this poses no hindrance. On the opposite, using over-parametrised representations often helps with the mixing properties of the corresponding MCMC algorithm, since they are less constrained by the data set.

This re-parametrisation of the model forces us to select a prior distribution on the q_{ij}, rather than on the p_{ij}. The choice $q_{ij} \sim \text{Gamma}(a, 1)$ is natural, in that it leads to a $\text{Dir}(a, a, \ldots, a)$ distribution on the corresponding rows $(p_{i1}, p_{i2}, \ldots, p_{ir})$. It is also noteworthy that, given $\mathbf{x}$, $(p_{i1}, p_{i2}, \ldots, p_{ir})$ and $\sum_{\ell=1}^{r} q_{i\ell}$ are conditionally independent, which means that the re-parametrisation does nothing, but introduce a new parameter for each row, which is independent of everything else and, hence, totally irrelevant for inference.

To implement the reversible jump algorithm, a first step is to design a strategy for moving between models with different number of states. If the current model is comprised of $r \in \{2, 3, \ldots, R-1\}$ states, then it is usual to reduce the searching strategy to moves that either preserve the number of states or lead to a model with $r-1$ or $r+1$ states. Jumps are achieved by adding new states, deleting existing states and splitting or combining existing states. The various moves could be scanned systematically or could be selected randomly. Ideally, the dimension-changing moves are designed to have high probability of acceptance, so that the sampler explores the different models adequately.

For practical implementation in the context of HMMs, it is sufficient to choose a mapping function $\boldsymbol{\vartheta}_{r+1} = \mathbf{g}(\boldsymbol{\vartheta}_r, \mathbf{u})$ together with a proposal density $q(\mathbf{u})$ to perform the move from $\mathcal{M}_r$ to $\mathcal{M}_{r+1}$ and the reverse move $(\boldsymbol{\vartheta}_r, \mathbf{u}) = \mathbf{g}^{-1}(\boldsymbol{\vartheta}_{r+1})$ from $\mathcal{M}_{r+1}$ to $\mathcal{M}_r$. Such moves form a reversible pair with acceptance probabilities $\min\{1, A_{r,r+1}\}$ and $\min\left\{1, A_{r,r+1}^{-1}\right\}$ respectively, where

$$A_{r,r+1} = \frac{f(\mathbf{y}|\boldsymbol{\vartheta}_{r+1})}{f(\mathbf{y}|\boldsymbol{\vartheta}_r)} \cdot \frac{(r+1)!\pi(r+1)\pi_{r+1}(\boldsymbol{\vartheta}_{r+1})}{r!\pi(r)\pi_r(\boldsymbol{\vartheta}_r)} \cdot \frac{P_{r+1,r}}{P_{r,r+1}q(\mathbf{u})} \cdot \left|\frac{\partial \mathbf{g}(\boldsymbol{\vartheta}_r, \mathbf{u})}{\partial(\boldsymbol{\vartheta}_r, \mathbf{u})}\right|. \quad (6.6)$$

Here, the first term constitutes the likelihood ratio, the second term constitutes the prior ratio, the third term represents the proposal ratio and the last term constitutes the absolute value of the determinant of the Jacobian matrix associated with the mapping $\mathbf{g}$. The observed likelihood of the two models may be computed on the log scale via the Forward Filtering algorithm.

The factorials arise from the fact that, as the prior is invariant under permutation of states, we cannot distinguish between parameters which are identical up to such permutations. Thus, our effective parameter space of r-state HMMs is that of equivalence classes of parameters which are identical up to permutations. The prior of such an equivalence class is $r!$ times the original prior of one of its representations. This distinction between a parameter and its equivalence class becomes essential, when r is allowed to vary, as ignoring it would lead to incorrect acceptance ratios.

Designing Birth and Death Moves

In a birth move, the order of the Markov chain is increased by one, by adding a new state, and the death move works in the reverse way, by deleting an existing state. Suppose that the current model is $\mathcal{M}_r$ and that we attempt to add a new state, denoted by i_0, to the HMM. We first draw the random variables

$$q_{i,i_0} \sim \text{Gamma}(a, 1), \quad i = 1, 2, \ldots, r+1, \quad (6.7)$$

$$q_{i_0,j} \sim \text{Gamma}(a, 1), \quad j = 1, 2, \ldots, r+1, \quad (6.8)$$

$$\boldsymbol{\theta}_{i_0} \sim \pi(\boldsymbol{\theta}),$$

all independently. In other words, all the parameters of the new state are drawn from their respective prior distributions. These parameters constitute the auxiliary random vector $\mathbf{u}_{\text{birth}}$ for the birth move. The remaining parameters are simply copied to the proposed new model $\mathcal{M}_{r+1}$. Therefore, the corresponding mapping $\mathbf{g}_{\text{birth}}$ is simply the identity mapping.

In the death move, the auxiliary random vector of the associated birth move is trivially recovered, since it just consists of the components of the state i_0, which is proposed to be deleted.

The probability of proposing a birth move, when the current number of states is r, is denoted by $P_b(r)$, whereas by $P_d(r+1)$ we denote the probability of proposing a death move, when the current number of states is $r+1$. So, $\frac{P_d(r+1)}{r+1}$ is the probability of proposing to kill the specific state i_0 out of the $r+1$ possible states.

Because the mapping $\mathbf{g}_{\text{birth}}$ is the identity mapping, its Jacobian is the identity matrix, with determinant $J_{\text{birth}} = 1$. The remaining factors of the acceptance ratio of the birth move become

$$\begin{aligned} A_{\text{birth}} &= \frac{f(\mathbf{y}|\boldsymbol{\vartheta}_{r+1})}{f(\mathbf{y}|\boldsymbol{\vartheta}_r)} \cdot \frac{(r+1)\pi(r+1)\pi_{r+1}(\boldsymbol{\vartheta}_{r+1})}{\pi(r)\pi_r(\boldsymbol{\vartheta}_r)} \cdot \frac{P_d(r+1)}{(r+1)P_b(r)q(\mathbf{u}_{\text{birth}})} \\ &= \frac{P_d(r+1)\pi(r+1)f(\mathbf{y}|\boldsymbol{\vartheta}_{r+1})}{P_b(r)\pi(r)f(\mathbf{y}|\boldsymbol{\vartheta}_r)}. \end{aligned} \tag{6.9}$$

Since the proposal densities are identical to the prior distributions of the corresponding new parameters and the rest of the parameters remain unchanged in model $\mathcal{M}_{r+1}$, there were cancellations in the acceptance ratio of the birth move, leading to the above simplified expression.

Obviously, the acceptance ratio of the death move is the inverse of the above, which completes the description of the birth-death move.

Designing Split and Combine Moves

The split move takes an existing state and splits it in two, whereas the combine move takes a pair of states and merges them into one. Starting with the split move, suppose that the current model is $\mathcal{M}_r$ and that we attempt to split a random state i_0 into two new states i_1 and i_2. The parameters corresponding to state i_0 must, then, be split. This can be done as follows

- Split column i_0 as

$$\begin{aligned} q_{i,i_1} &= u_i q_{i,i_0}, \\ q_{i,i_2} &= (1-u_i) q_{i,i_0}, \end{aligned} \tag{6.10}$$

  where $u_i \sim \text{Beta}(a,a)$ for $i = 1,2,\ldots,r$ with $i \neq i_0$.

- Split row i_0 as

$$\begin{aligned} q_{i_1,j} &= 2v_j q_{i_0,j}, \\ q_{i_2,j} &= 2(1-v_j) q_{i_0,j}, \end{aligned} \tag{6.11}$$

  where $v_j \sim \text{Beta}(a,a)$ for $j = 1,2,\ldots,r$ with $j \neq i_0$.

- Split q_{i_0,i_0} as

$$\begin{aligned} q_{i_1,i_1} &= 2w_1w_2q_{i_0,i_0}, \\ q_{i_1,i_2} &= 2(1-w_1)w_3q_{i_0,i_0}, \\ q_{i_2,i_1} &= 2w_1(1-w_2)q_{i_0,i_0}, \\ q_{i_2,i_2} &= 2(1-w_1)(1-w_3)q_{i_0,i_0}, \end{aligned} \tag{6.12}$$

  where $w_1 \sim \text{Beta}(2a, 2a)$ and $w_2, w_3 \sim \text{Beta}(a, a)$.

- To satisfy the remaining degrees of freedom, a random vector $\mathbf{z}$ of dimension $\dim(\boldsymbol{\theta})$ with non-degenerate proposal density $q(\mathbf{z})$ is chosen. It is then used to construct $\boldsymbol{\theta}_{i_1}$ and $\boldsymbol{\theta}_{i_2}$ from $\boldsymbol{\theta}_{i_0}$ through $\boldsymbol{\theta}_{i_1} = \mathbf{g}_1(\boldsymbol{\theta}_{i_0}, \mathbf{z})$ and $\boldsymbol{\theta}_{i_2} = \mathbf{g}_2(\boldsymbol{\theta}_{i_0}, \mathbf{z})$.

The auxiliary random vector $\mathbf{u}_{\text{birth}}$ is comprised of $\mathbf{u} = (u_i)_{i \neq i_0}$, $\mathbf{v} = (v_i)_{i \neq i_0}$, $\mathbf{w} = (w_1, w_2, w_3)$ and $\mathbf{z}$. Now, we describe the combine move, which reverses the above operations. Two distinct states i_1 and i_2 are selected at random and we attempt to combine them into a single state i_0 as follows

$$\begin{aligned} q_{i,i_0} &= q_{i,i_1} + q_{i,i_2}, \quad i = 1, 2, \ldots, r, \quad i \neq i_0, \\ q_{i_0,j} &= \frac{q_{i_1,j} + q_{i_2,j}}{2}, \quad j = 1, 2, \ldots, r, \quad j \neq i_0, \\ q_{i_0,i_0} &= \frac{q_{i_1,i_1} + q_{i_2,i_1}}{2} + \frac{q_{i_1,i_2} + q_{i_2,i_2}}{2}. \end{aligned} \tag{6.13}$$

To compute the acceptance rate of the combine move, the auxiliary random vector of the associated split move has to be reconstructed. We may obtain $\boldsymbol{\theta}_{i_0}$ and $\mathbf{z}$ by inverting the mapping functions g_1 and g_2 defined above, while the remaining auxiliary variables are given by

$$\begin{aligned} u_i &= \frac{q_{i,i_1}}{q_{i,i_1} + q_{i,i_2}}, \quad i = 1, 2, \ldots, r, \quad i \neq i_0, \\ v_j &= \frac{q_{i_1,j}}{q_{i_1,j} + q_{i_2,j}}, \quad j = 1, 2, \ldots, r, \quad j \neq i_0, \\ w_1 &= \frac{q_{i_1,i_1} + q_{i_2,i_1}}{q_{i_1,i_1} + q_{i_1,i_2} + q_{i_2,i_1} + q_{i_2,i_2}}, \\ w_2 &= \frac{q_{i_1,i_1}}{q_{i_1,i_1} + q_{i_2,i_1}}, \\ w_3 &= \frac{q_{i_1,i_2}}{q_{i_1,i_2} + q_{i_2,i_2}}. \end{aligned} \tag{6.14}$$

Finally, we denote by $P_s(r)$ the probability of proposing a split move, when the current number of states is r and by $P_c(r+1)$ the probability of proposing a combine move, when the current number of states is $r+1$. Then, $\frac{P_s(r)}{r}$ and $\frac{2P_c(r+1)}{r(r+1)}$ are the probabilities to propose to split a specific state out of r and to propose to combine a specific pair of states out of $\frac{r(r+1)}{2}$ possible ones respectively.

Example 6.1 ***(Reversible Jump for Poisson Hidden Markov Models)*** *For a Poisson HMM, a single additional auxiliary random variable z is needed to split λ_{i_0}. The choice of z, however, is not totally free, as the new parameters λ_{i_1} and λ_{i_2} are subject to non-negativity constraints. One way to overcome this difficulty is to propose the same split move as for the rows of the probability transition matrix*

$$\begin{aligned} \lambda_{i_1} &= 2z\lambda_{i_0}, \\ \lambda_{i_2} &= 2(1-z)\lambda_{i_0}, \end{aligned} \tag{6.15}$$

where $z \sim Beta(b, b)$. On the other hand, the combine move, which reverses the above operations, yields

$$\begin{aligned} \lambda_{i_0} &= \frac{\lambda_{i_1} + \lambda_{i_2}}{2}, \\ z &= \frac{\lambda_{i_1}}{\lambda_{i_1} + \lambda_{i_2}}. \end{aligned} \tag{6.16}$$

In this transformation, most states, namely all that are not associated with state i_0 which is split, remain unaffected and do not affect, in turn, any of the other states of the new model. In effect, this means that the Jacobian determinant equals the Jacobian determinant associated with the states actually involved in the split. Analysing this part, we notice that the Jacobian takes the form of a block-diagonal matrix, since the sets of parameters and auxiliary variables involved in each of the steps comprising the split move are disjoint. The determinant of the Jacobian will be the product of the determinants given below

- *The Jacobian is further block-diagonal with respect to each $i \neq i_0$. For each such i, the transformation takes (q_{i,i_0}, u_i) into (q_{i,i_1}, q_{i,i_2}) with Jacobian*

$$\begin{bmatrix} u_i & q_{i,i_0} \\ 1-u_i & -q_{i,i_0} \end{bmatrix}$$

  and determinant q_{i,i_0} in absolute value. The overall absolute value of the Jacobian determinant of this step is $\prod_{\substack{i=1 \\ i \neq i_0}}^{r} q_{i,i_0}$.

- *The Jacobian is also further block-diagonal with respect to each $j \neq i_0$. For each such j, the transformation takes $(q_{i_0,j}, v_j)$ into $(q_{i_1,j}, q_{i_2,j})$ with Jacobian*

$$2 \cdot \begin{bmatrix} v_j & q_{i_0,j} \\ (1-v_j) & -q_{i_0,j} \end{bmatrix}$$

  and determinant $4q_{i_0,j}$ in absolute value. The overall absolute value of the Jacobian determinant of this step is $4^{r-1} \prod_{\substack{j=1 \\ j \neq i_0}}^{r} q_{i_0,j}$.

- *This step takes* $(q_{i_0,i_0}, w_2, w_2, w_3)$ *into* $(q_{i_1,i_1}, q_{i_1,i_2}, q_{i_2,i_1}, q_{i_2,i_2})$ *with Jacobian*

$$2 \cdot \begin{bmatrix} w_1 w_2 & w_2 q_{i_0,i_0} & w_1 q_{i_0,i_0} & 0 \\ (1-w_1)w_3 & -w_3 q_{i_0,i_0} & 0 & (1-w_1) q_{i_0,i_0} \\ w_1(1-w_2) & (1-w_2) q_{i_0,i_0} & -w_1 q_{i_0,i_0} & 0 \\ (1-w_1)(1-w_3) & -(1-w_3) q_{i_0,i_0} & 0 & -(1-w_1) q_{i_0,i_0} \end{bmatrix}$$

  and determinant $16 w_1 (1-w_1) q_{i_0,i_0}^3$ *in absolute value.*

- *Lastly, this step takes* (λ_{i_0}, z) *into* $(\lambda_{i_1}, \lambda_{i_2})$ *with Jacobian*

$$2 \cdot \begin{bmatrix} z & \lambda_{i_0} \\ 1-z & -\lambda_{i_0} \end{bmatrix}$$

  and determinant $4\lambda_{i_0}$ *in absolute value.*

Finally, we arrive at the absolute value of the overall Jacobian determinant of the split move

$$J_{split} = 4^{r+2} \lambda_{i_0} w_1 (1-w_1) q_{i_0,i_0} \prod_{i=1}^{r} q_{i,i_0} \prod_{j=1}^{r} q_{i_0,j}, \tag{6.17}$$

while the acceptance ratio of the split move is not as easily computed as for the birth move, since the components of the new state are not drawn from their respective priors

$$\begin{aligned} A_{split} &= \frac{\pi(r+1) f(\boldsymbol{y}|\boldsymbol{\vartheta}_{r+1})}{\pi(r) f(\boldsymbol{y}|\boldsymbol{\vartheta}_r)} \cdot \frac{(r+1)\pi_{r+1}(\boldsymbol{\vartheta}_{r+1})}{\pi_r(\boldsymbol{\vartheta}_r)} \cdot \frac{2rP_c(r+1)}{r(r+1)P_s(r)} \cdot \frac{1}{2q(\boldsymbol{u}_{split})} \cdot J_{split} \\ &= 4^{a(r+1)+b} \cdot \frac{P_c(r+1)\pi(r+1) f(\boldsymbol{y}|\boldsymbol{\vartheta}_{r+1})}{P_s(r)\pi(r) f(\boldsymbol{y}|\boldsymbol{\vartheta}_r)} \cdot \frac{[\Gamma(a)]^{2r-1}}{\Gamma(4a)\,[\Gamma(2a)]^{2(r-1)}} \\ &\quad \cdot \left[q_{i_0,i_0} \prod_{i=1}^{r} q_{i,i_0} \prod_{j=1}^{r} q_{i_0,j} \right]^a \cdot \exp\left\{ -\sum_{j=1}^{r} q_{i_0,j} \right\} \cdot \frac{\Gamma(b)}{\Gamma(2b)} \cdot (B\lambda_{i_0})^b \cdot e^{-B\lambda_{i_0}}. \end{aligned} \tag{6.18}$$

Here, the proposal density is initially multiplied by 2, since there are two different combinations of auxiliary random variables which have equal density and result in identical parameters after the split. Of course, the acceptance rate for the combine move is the inverse of the above. ■

Example 6.2 ***(Reversible Jump for Normal Hidden Markov Models)*** *For a Normal HMM, two additional auxiliary random variables* z_1 *and* z_2 *are needed to split* $\left(\mu_{i_0}, \sigma_{i_0}^2\right)$*. One possible way to simulate the parameters of the new states is through simple random walk proposals*

- *Split* μ_{i_0} *as*

$$\begin{aligned} \mu_{i_1} &= \mu_{i_0} - z_1, \\ \mu_{i_2} &= \mu_{i_0} + z_1, \end{aligned} \tag{6.19}$$

where $z_1 \sim \mathcal{N}(0, \tau_\mu)$ *and* τ_μ *is a parameter which may be adjusted to optimise the performance of the split move.*

- *Split* $\sigma^2_{i_0}$ *through a multiplicative random walk proposal as*

$$\begin{aligned} \sigma^2_{i_1} &= \sigma^2_{i_0} z_2, \\ \sigma^2_{i_2} &= \sigma^2_{i_0} z_2^{-1}, \end{aligned} \tag{6.20}$$

where $z_2 \sim$ *Lognormal*$(0, \tau_\sigma)$ *and* τ_σ *is a parameter which may also be adjusted.*

On the other hand, the move which reverses the above operations, that is, the combine move, goes as follows

$$\begin{aligned} \mu_{i_0} &= \frac{\mu_{i_1} + \mu_{i_2}}{2}, \\ z_1 &= \frac{\mu_{i_2} - \mu_{i_1}}{2}, \\ \sigma^2_{i_0} &= \sigma_{i_1}\sigma_{i_2}, \\ z_2 &= \sigma_{i_1}\sigma^{-1}_{i_2}. \end{aligned} \tag{6.21}$$

As far as the Jacobian of the transformation is concerned, the only part that differs is the diagonal block which takes $\left(\mu_{i_0}, z_1, \sigma^2_{i_0}, z_2\right)$ *into* $\left(\mu_{i_1}, \mu_{i_2}, \sigma^2_{i_1}, \sigma^2_{i_2}\right)$*, with Jacobian*

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & z_2 & \sigma^2_{i_0} \\ 0 & 0 & \frac{1}{z_2} & -\left[\frac{\sigma_{i_0}}{z_2}\right]^2 \end{bmatrix}$$

and determinant $\frac{4\sigma^2_{i_0}}{z_2}$ *in absolute value.*

The absolute value of the overall Jacobian determinant of the split move is

$$J_{split} = \frac{4^{r+2}\sigma^2_{i_0} w_1(1-w_1) q_{i_0,i_0}}{z_2} \prod_{i=1}^{r} q_{i,i_0} \prod_{j=1}^{r} q_{i_0,j}, \tag{6.22}$$

while the acceptance ratio A_{split} *is given below*

$$\begin{aligned} A_{split} = 4^{a(r+1)+1} &\cdot \frac{P_c(r+1)\pi(r+1)f(\boldsymbol{y}|\boldsymbol{\vartheta}_{r+1})}{P_s(r)\pi(r)f(\boldsymbol{y}|\boldsymbol{\vartheta}_r)} \cdot \frac{C^c}{\Gamma(c)} \cdot \sigma^{-2c}_{i_0} \cdot \frac{[\Gamma(a)]^{2r-1}}{\Gamma(4a)\left[\Gamma(2a)\right]^{2(r-1)}} \\ &\cdot \left[q_{i_0,i_0} \prod_{i=1}^{r} q_{i,i_0} \prod_{j=1}^{r} q_{i_0,j}\right]^a \cdot \exp\left\{-\sum_{j=1}^{r} q_{i_0,j}\right\} \cdot \exp\left\{-\frac{(\mu_{i_0}-b)^2 + 2z_1^2}{2B}\right\} \\ &\cdot \exp\left\{-\left[z_2 - 1 + \frac{1}{z_2}\right] \cdot \frac{C}{\sigma^2_{i_0}}\right\} \cdot \sqrt{\frac{2\pi\tau_\mu\tau_\sigma}{B}} \cdot \exp\left\{\frac{z_1^2}{2\tau_\mu}\right\} \cdot \exp\left\{\frac{[\log z_2]^2}{2\tau_\sigma}\right\}. \end{aligned} \tag{6.23}$$

Of course, the acceptance rate for the combine move of two states is the inverse of the above. ■

Just as for MCMC algorithms with fixed r, several types of moves are typically combined into a sweep of the Reversible Jump algorithm. We begin with some initial values for all possible model-specific parameter vectors $\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \ldots, \boldsymbol{\vartheta}_R$ and a specific value for the model index r. For the current algorithm, a sweep may look as follows

- Propose a birth move or a death move, with probabilities $P_b(r)$ and $P_d(r)$ respectively.
- Propose a split move or a combine move, with probabilities $P_s(r)$ and $P_c(r)$ respectively.
- Update the current hidden states $\mathbf{X}$ through the global updating algorithm.
- Update the model-specific parameter vector $\boldsymbol{\vartheta}_r$ through the Gibbs sampling algorithm.

Obviously $P_b(r) + P_d(r) = 1$ and $P_s(r) + P_c(r) = 1$ must hold for all r. Typically, all these probabilities are set to 0.5, except for $P_b(1) = P_s(1) = P_d(R) = P_c(R) = 1$ and $P_b(R) = P_s(R) = P_d(1) = P_c(1) = 0$. The posterior probability $\pi(r|\mathbf{y})$ of model $\mathcal{M}_r$ may be estimated from the draws $r^{(1)}, r^{(2)}, \ldots, r^{(L)}$ of the model indices as

$$\hat{\pi}_{\mathrm{RJ}}(r|\mathbf{y}) = \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{1}\left\{r^{(\ell)} = r\right\}. \tag{6.24}$$

Based on these posterior probabilities, the framework of Bayesian decision theory is used for model selection. In the absence of specific information about the actual loss incurred by a wrong decision, it is customary to consider the $0-1$ loss function, which leads to selecting the model $\mathcal{M}_r$ with the highest posterior probability $\hat{\pi}_{\mathrm{RJ}}(r|\mathbf{y})$.

As a last remark, we note that combining a reference prior, such as the uniform $\mathrm{Dir}(1, 1, \ldots, 1)$ prior, on the rows of the transition matrix with a uniform prior on the number of states, leads to a high risk of selecting too many states, even for quite large data sets. On the other hand, a truncated Poisson(1) prior, which is proportional to $\frac{1}{r!}$, in combination with a $\mathrm{Dir}(4, 4, \ldots, 4)$ prior on the rows of the transition matrix seems to be optimal, if the major objective is to avoid over-fitting.

Of course, the reverse is also true. For small data sets, this prior carries some risk of under-fitting, as opposed to the combination of the uniform priors, which minimises this risk. Quite generally, prior distributions which appear only weakly informative for the state-specific parameters of the hidden Markov model may be highly informative about the number of states of the mixture.

6.3.2 Birth-Death Markov Chain Monte Carlo

A hidden Markov model may be viewed, in an abstract sense, as a marked point process in a general space. To sample from the posterior distribution of a hidden Markov model with an unknown number of states, modified simulation methods, which regard a spatial point-process as the invariant distribution of a continuous-time spatial birth and death Markov process, have been developed.

In a birth and death Markov process, births and deaths occur in continuous time. A birth occurs at a constant rate λ_b. On the other hand, for each state $i = 1, 2, \ldots, r$, a death occurs at a rate d_i, which is low for states that are important for explaining the data, but high for states that do not help to explain the data. This relevance is mainly measured in terms of the observed likelihood $f(\mathbf{y}|\boldsymbol{\vartheta})$ of the current hidden Markov model, in relation to the observed likelihood $f(\mathbf{y}|\boldsymbol{\vartheta}_{-i})$ of a hidden Markov model without state i.

For the Birth-Death MCMC, a sweep of the algorithm may look as follows. Select a Poisson(λ_r) prior for the number of states r, as well as a fixed time t_0 for running the birth and death process. Begin with a specific value for the model index r and some initial values for the model-specific parameter vector $\boldsymbol{\vartheta}$ of model $\mathcal{M}_r$. Then, iterate the following steps

- Simulate $(r, \boldsymbol{\vartheta})$ by running a birth and death process for fixed time t_0. In other words, set $t = 0$ and repeat the following steps while $t < t_0$
  - If $r > 1$, determine the actual death rate d_i for each possible state as

$$d_i = \frac{\lambda_b \pi(r-1) f(\mathbf{y}|\boldsymbol{\vartheta}_{-i})}{r\pi(r) f(\mathbf{y}|\boldsymbol{\vartheta})} = \frac{\lambda_b f(\mathbf{y}|\boldsymbol{\vartheta}_{-i})}{\lambda_r f(\mathbf{y}|\boldsymbol{\vartheta})}, \tag{6.25}$$

    by utilising the Forward Filtering algorithm, in order to determine the observed log-likelihood of each model. Also determine the overall death rate $\lambda_d = \sum_{i=1}^{r} d_i$.
  - Simulate the arrival time $t^{\text{new}} = t + \text{Exp}(\lambda_b + \lambda_d)$ to the next jump.
  - If $t^{\text{new}} < t_0$, simulate the type of jump with the appropriate probabilities

$$P(\text{birth}) = \frac{\lambda_b}{\lambda_b + \lambda_d}, \quad P(\text{death of state } i) = \frac{d_i}{\lambda_b + \lambda_d}. \tag{6.26}$$

  - If $t^{\text{new}} < t_0$, also adjust the hidden Markov model to reflect either the birth of a new state or the death of state i, in the same way as for the birth-death move in the Reversible Jump algorithm.
  - Set $t = t^{\text{new}}$.

- Run several steps of full-conditional Gibbs sampling for the current number of states r.
  - Update the hidden states $\mathbf{X}$ via the global updating algorithm and the hyper-parameter $\boldsymbol{\delta}$ of the state-specific prior distributions.
  - Update the transition probability matrix $\mathbf{P}$ and the state-specific parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_r$.

Doubling the birth rate λ_b is equivalent to doubling t_0, thus one is free to choose $t_0 = 1$. Larger values of λ_b will result in better mixing properties, but will require more computation time. To a certain degree, birth and death methods appear to be more natural and elegant than reversible jump methods, since they avoid calculating the Jacobian of the transformation.

6.4 Marginal Likelihoods for Hidden Markov Models

The marginal posterior distribution $\pi(r|\mathbf{y})$, which provides the posterior probability of the various models $\mathcal{M}_r$ given the data, may, alternatively, be calculated through Bayes' theorem as

$$\pi(r|\mathbf{y}) \propto \pi(r) f(\mathbf{y}|\mathcal{M}_r), \tag{6.27}$$

where the so-called marginal likelihood $f(\mathbf{y}|\mathcal{M}_r)$ is given by

$$f(\mathbf{y}|\mathcal{M}_r) = \int_{\Theta_r} \pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r) d\boldsymbol{\vartheta}_r. \tag{6.28}$$

For hidden Markov models, the marginal likelihood $f(\mathbf{y}|\mathcal{M}_r)$ is not available in closed form and obtaining a good numerical approximation to it is quite a challenging integration problem. Marginal likelihoods have been approximated using a multitude of simulation-based methods, such as importance sampling, reciprocal importance sampling and bridge sampling. Although these methods prove to be useful for a wide range of statistical models, most of them are apt to fail in the case of hidden Markov models.

If a sampling-based estimator relies on MCMC draws, it is essential to use an MCMC technique which explores all the modes of the unconstrained posterior, since the behaviour of the Gibbs sampler is somewhat unpredictable. It may get trapped at one modal region or otherwise fail to explore the whole posterior distribution, if label switching took place only from time to time, in an unbalanced manner. Estimating

the marginal density from the MCMC draws may lead to a poor estimate, when unbalanced label switching takes place.

A simple but efficient solution to obtain a sampler which explores the full posterior distribution is to force balanced label switching, by concluding each MCMC draw by a randomly selected permutation of the labelling. This method is called random permutation MCMC sampling and is described below

- Run a sweep of the Gibbs sampler with global updating to obtain $\left(\boldsymbol{\vartheta}^{(\ell)}, \mathbf{x}^{(\ell)}\right)$.
- Select randomly one of the $r!$ possible permutations σ of the current labelling.
  - Each element $p_{ij}^{(\ell)}$ of the simulated transition matrix is substituted by $p_{\sigma(i),\sigma(j)}^{(\ell)}$ for $i, j = 1, 2, \ldots, r$.
  - The state-specific parameter $\boldsymbol{\theta}_i^{(\ell)}$ is substituted by $\boldsymbol{\theta}_{\sigma(i)}^{(\ell)}$ for $i = 1, 2, \ldots, r$.
  - The hidden states $x_k^{(\ell)}$ are substituted by $\sigma(x_k^{(\ell)})$ for $k = 0, 1, \ldots, n$.

For all of these sampling-based techniques, one has to select an importance density $q(\boldsymbol{\vartheta}_r)$, from which it is easy to sample and which provides a rough approximation to the marginal posterior density $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$. As manual tuning of the importance density for each model under consideration is rather tedious, a method for selecting sensible importance densities in an unsupervised manner is required.

For hidden Markov models, where the posterior $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$ is multi-modal, a similarly multi-modal importance density arises in quite a natural way within the data augmentation framework. Evidently, this posterior density may be expressed in the following way

$$\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y}) = \sum_{\mathbf{x}\in\mathbf{X}} \left[\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\vartheta}_r) \pi_r(\mathbf{P}|\mathbf{x}) \prod_{i=1}^{r} \pi_r(\boldsymbol{\theta}_i|\mathbf{y}, \mathbf{x}) \right]. \tag{6.29}$$

Thus, a random subsequence $\mathbf{x}^{(t)}$, $t = 1, 2, \ldots, T$ of the MCMC draws of the state vector $\mathbf{x}$ could be used to construct the following importance density

$$q(\boldsymbol{\vartheta}_r) = \frac{1}{T} \sum_{t=1}^{T} \left[\pi_r\left(\mathbf{P}|\mathbf{x}^{(t)}\right) \prod_{i=1}^{r} \pi_r\left(\boldsymbol{\theta}_i|\mathbf{y}, \mathbf{x}^{(t)}\right) \right]. \tag{6.30}$$

6.4.1 Importance Sampling

A simple Monte Carlo approximation of the marginal likelihood given in (6.28) may be obtained by

$$\hat{f}_{\mathrm{MC}}(\mathbf{y}|\mathcal{M}_r) = \frac{1}{L} \sum_{\ell=1}^{L} f\left(\mathbf{y}|\boldsymbol{\vartheta}_r^{(\ell)}\right), \tag{6.31}$$

where $\boldsymbol{\vartheta}_r^{(1)}, \boldsymbol{\vartheta}_r^{(2)}, \ldots, \boldsymbol{\vartheta}_r^{(L)}$ is a sample from the prior $\pi_r(\boldsymbol{\vartheta}_r)$. The resulting estimator is rather inefficient, if the likelihood is more informative compared to the prior. Importance sampling may be used to obtain a better approximation to the marginal likelihood by rewriting the marginal likelihood as

$$f(\mathbf{y}|\mathcal{M}_r) = \int \frac{\pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r)}{q(\boldsymbol{\vartheta}_r)} q(\boldsymbol{\vartheta}_r) d\boldsymbol{\vartheta}_r, \tag{6.32}$$

where $q(\boldsymbol{\vartheta}_r)$ is a suitably chosen importance density, such as the one described above. If a sample $\tilde{\boldsymbol{\vartheta}}_r^{(1)}, \tilde{\boldsymbol{\vartheta}}_r^{(2)}, \ldots, \tilde{\boldsymbol{\vartheta}}_r^{(M)}$ from $q(\boldsymbol{\vartheta}_r)$ is available, then the marginal likelihood is estimated by

$$\hat{f}_{\mathrm{IS}}(\mathbf{y}|\mathcal{M}_r) = \frac{1}{M} \sum_{m=1}^{M} \frac{\pi_r(\tilde{\boldsymbol{\vartheta}}_r^{(m)}) f(\mathbf{y}|\tilde{\boldsymbol{\vartheta}}_r^{(m)})}{q(\tilde{\boldsymbol{\vartheta}}_r^{(m)})}. \tag{6.33}$$

A sufficient but not necessary condition for this estimator to have finite variance, is that the ratio $\frac{\pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r)}{q(\boldsymbol{\vartheta}_r)}$ is bounded, which implies that the tails of $q(\boldsymbol{\vartheta}_r)$ should be fat, when compared to the tails of the posterior density $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$.

6.4.2 Reciprocal Importance Sampling

The marginal likelihood is not directly available as an expectation with respect to the posterior density, thus straightforward approximations to the marginal likelihood from the MCMC output are not available. A tricky method which expresses the marginal likelihood with respect to the posterior is given by

$$\frac{1}{f(\mathbf{y}|\mathcal{M}_r)} = \frac{\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})}{\pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r)}.$$

By multiplying both sides with the arbitrary density $q(\boldsymbol{\vartheta}_r)$ and integrating with respect to $\boldsymbol{\vartheta}_r$, one obtains the following identity

$$\frac{1}{f(\mathbf{y}|\mathcal{M}_r)} = \int \frac{q(\boldsymbol{\vartheta}_r)}{\pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r)} \pi_r(\boldsymbol{\vartheta}_r|\mathbf{y}) d\boldsymbol{\vartheta}_r.$$

Therefore, the inverse of the marginal likelihood is equal to the posterior expectation of the ratio of an arbitrary importance density $q(\boldsymbol{\vartheta}_r)$ and the non-normalised posterior density. This yields the following estimator of the marginal likelihood

$$\hat{f}_{\mathrm{RIS}}(\mathbf{y}|\mathcal{M}_r) = \left[\frac{1}{L} \sum_{\ell=1}^{L} \frac{q(\boldsymbol{\vartheta}_r^{(\ell)})}{\pi_r(\boldsymbol{\vartheta}_r^{(\ell)}) f(\mathbf{y}|\boldsymbol{\vartheta}_r^{(\ell)})} \right]^{-1}, \tag{6.34}$$

where $\boldsymbol{\vartheta}_r^{(1)}, \boldsymbol{\vartheta}_r^{(2)}, \ldots, \boldsymbol{\vartheta}_r^{(L)}$ are draws from the posterior $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$. Note that the importance density is only evaluated at the MCMC draws, but no draws from the

importance density are required. A sufficient but not necessary condition for this estimator to have finite variance, is that the ratio $\frac{q(\boldsymbol{\vartheta}_r)}{\pi_r(\boldsymbol{\vartheta}_r)f(\mathbf{y}|\boldsymbol{\vartheta}_r)}$ is bounded, which implies that the tails of $q(\boldsymbol{\vartheta}_r)$ should be thin, when compared to the tails of the posterior $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$.

6.4.3 Bridge Sampling

Bridge sampling was introduced into statistics as a simulation-based technique for computing ratios of normalising constants. It generalises the method of importance sampling and, similarly to importance sampling, it is based on an i.i.d. sample from an importance density. However, this sample is combined with the MCMC draws from the posterior density, in an appropriate way. An important advantage of bridge sampling is that the variance of the resulting estimator depends on a ratio which is bounded regardless of the tail behaviour of the underlying importance density. This allows for more flexibility in the construction of the importance density.

Let $q(\boldsymbol{\vartheta}_r)$ be the importance density, as for the simulation-based methods discussed earlier, and $a(\boldsymbol{\vartheta}_r)$ be an arbitrary function such that

$$\int a(\boldsymbol{\vartheta}_r)q(\boldsymbol{\vartheta}_r)\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})d\boldsymbol{\vartheta}_r > 0.$$

Bridge sampling is based on the following result

$$\int a(\boldsymbol{\vartheta}_r)\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})q(\boldsymbol{\vartheta}_r)d\boldsymbol{\vartheta}_r = \int a(\boldsymbol{\vartheta}_r)q(\boldsymbol{\vartheta}_r)\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})d\boldsymbol{\vartheta}_r.$$

Substituting $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y}) = \dfrac{\pi_r(\boldsymbol{\vartheta}_r)f(\mathbf{y}|\boldsymbol{\vartheta}_r)}{f(\mathbf{y}|\mathcal{M}_r)}$ into the left-hand term yields the key identity for bridge sampling

$$f(\mathbf{y}|\mathcal{M}_r) = \frac{\int a(\boldsymbol{\vartheta}_r)\pi_r(\boldsymbol{\vartheta}_r)f(\mathbf{y}|\boldsymbol{\vartheta}_r)q(\boldsymbol{\vartheta}_r)d\boldsymbol{\vartheta}_r}{\int a(\boldsymbol{\vartheta}_r)q(\boldsymbol{\vartheta}_r)\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})d\boldsymbol{\vartheta}_r}. \tag{6.35}$$

To estimate the marginal likelihood for a given function $a(\boldsymbol{\vartheta}_r)$, the integrals at the right-hand side are substituted by sample averages. The numerator is approximated using i.i.d. draws $\tilde{\boldsymbol{\vartheta}}_r^{(1)}, \tilde{\boldsymbol{\vartheta}}_r^{(2)}, \ldots, \tilde{\boldsymbol{\vartheta}}_r^{(M)}$ from $q(\boldsymbol{\vartheta}_r)$, whereas the denominator is approximated using Markov chain Monte Carlo draws $\boldsymbol{\vartheta}_r^{(1)}, \boldsymbol{\vartheta}_r^{(2)}, \ldots, \boldsymbol{\vartheta}_r^{(L)}$ from $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$. The resulting estimator $\hat{f}(\mathbf{y}|\mathcal{M}_r)$ is called the general bridge sampling estimator

$$\hat{f}(\mathbf{y}|\mathcal{M}_r) = \frac{M^{-1}\sum_{m=1}^{M} a(\tilde{\boldsymbol{\vartheta}}_r^{(m)})\pi_r(\tilde{\boldsymbol{\vartheta}}_r^{(m)})f(\mathbf{y}|\tilde{\boldsymbol{\vartheta}}_r^{(m)})}{L^{-1}\sum_{\ell=1}^{L} a(\boldsymbol{\vartheta}_r^{(\ell)})q(\boldsymbol{\vartheta}_r^{(\ell)})}. \tag{6.36}$$

The simulation-based methods discussed earlier result as special cases for appropriate choices of $a(\boldsymbol{\vartheta}_r)$, namely the importance sampling estimator for $a(\boldsymbol{\vartheta}_r) = \frac{1}{q(\boldsymbol{\vartheta}_r)}$ and the reciprocal importance sampling estimator for $a(\boldsymbol{\vartheta}_r) = \frac{1}{\pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r)}$. An asymptotically optimal choice for $a(\boldsymbol{\vartheta}_r)$, which minimises the expected relative mean squared error of the estimator $\hat{f}(\mathbf{y}|\mathcal{M}_r)$ would be

$$a(\boldsymbol{\vartheta}_r) \propto \frac{1}{Mq(\boldsymbol{\vartheta}_r) + L\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})}.$$

We refer to the corresponding estimator $\hat{f}_{\text{BS}}(\mathbf{y}|\mathcal{M}_r)$ as the bridge sampling estimator. As it turns out, this optimal choice of $a(\boldsymbol{\vartheta}_r)$ depends on the normalised posterior $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$, which, in turn, depends on the marginal likelihood $f(\mathbf{y}|\mathcal{M}_r)$. Thus, to estimate the marginal likelihood, we first need to know the marginal likelihood.

In order to solve this issue, we may apply an iterative procedure and obtain $\hat{f}_{\text{BS}}(\mathbf{y}|\mathcal{M}_r)$ as the limit of a sequence $\hat{f}_{\text{BS}}^{(t)}$ for $t \to \infty$. Based on the most recent estimate $\hat{f}_{\text{BS}}^{(t-1)}$ of the marginal likelihood, the posterior $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$ is normalised and a new estimate $\hat{f}_{\text{BS}}^{(t)}$ is obtained from (6.36). This leads to the recursion given below

- **Simulation Step**: Select the importance density $q(\boldsymbol{\vartheta}_r)$.
  - Run a permutation MCMC sampler to obtain draws $\boldsymbol{\vartheta}_r^{(1)}, \boldsymbol{\vartheta}_r^{(2)}, \ldots, \boldsymbol{\vartheta}_r^{(L)}$ from the posterior $\pi_r(\boldsymbol{\vartheta}_r|\mathbf{y})$.
  - Obtain independent $\tilde{\boldsymbol{\vartheta}}_r^{(1)}, \tilde{\boldsymbol{\vartheta}}_r^{(2)}, \ldots, \tilde{\boldsymbol{\vartheta}}_r^{(M)}$ draws from the importance density $q(\boldsymbol{\vartheta}_r)$.
- **Evaluation Step**: Evaluate both the non-normalised posterior $\pi_r(\boldsymbol{\vartheta}_r) f(\mathbf{y}|\boldsymbol{\vartheta}_r)$ and the importance density $q(\boldsymbol{\vartheta}_r)$ at all draws from the posterior, as well as at all draws from the importance density.
- **Iteration Step**: Use the computed values to determine a starting value $\hat{f}_{\text{BS}}^{(0)}$ and run the following recursion until convergence

$$\hat{f}_{\text{BS}}^{(t)} = \frac{M^{-1} \sum_{m=1}^{M} \frac{\pi_r(\tilde{\boldsymbol{\vartheta}}_r^{(m)}) f(\mathbf{y}|\tilde{\boldsymbol{\vartheta}}_r^{(m)})}{Mq(\tilde{\boldsymbol{\vartheta}}_r^{(m)}) + \pi_r(\tilde{\boldsymbol{\vartheta}}_r^{(m)}) f(\mathbf{y}|\tilde{\boldsymbol{\vartheta}}_r^{(m)})/\hat{f}_{\text{BS}}^{(t-1)}}}{L^{-1} \sum_{\ell=1}^{L} \frac{q(\boldsymbol{\vartheta}_r^{(\ell)})}{Mq(\boldsymbol{\vartheta}_r^{(\ell)}) + L\pi_r(\boldsymbol{\vartheta}_r^{(\ell)}) f(\mathbf{y}|\boldsymbol{\vartheta}_r^{(\ell)})/\hat{f}_{\text{BS}}^{(t-1)}}}. \tag{6.37}$$

This iteration is typically very fast in practice. Either the importance sampling estimator or the reciprocal importance sampling estimator may be used as starting values $\hat{f}_{\text{BS}}^{(0)}$. Since both estimators use the same values as the bridge sampling estimator, their calculation is possible with practically no additional computational effort.

Chapter 7

Numerical Results

7.1 Lamb Data

We consider the Lamb Data, a time series of count data analysed originally in Nhu D. Le et al. (1992) and re-analysed by Früwirth-Schnatter (2004). The data plotted in Figure 7.1 are the number of movements by a fetal lamb in 240 consecutive five-second intervals.

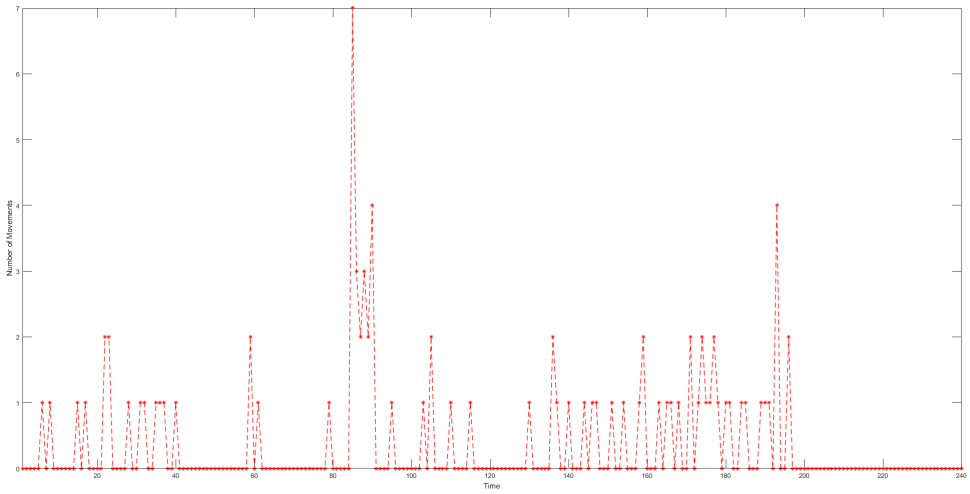

FIGURE 7.1: Time Series Plot of Lamb Data

Assuming that the counts are i.i.d. realisations from a Poisson distribution implies that the mean is equal to the variance. This assumption, however, is violated, since the sample variance, $s^2 = 0.6577$, is nearly twice the sample mean, $\bar{y} = 0.3583$. To capture over-dispersion, a finite mixture of Poisson distributions could be applied. However, the plots of the empirical autocorrelation and partial autocorrelation functions in Figure 7.2 also indicate stochastic dependence between subsequent observations. In order to capture both over-dispersion and autocorrelation, a Poisson hidden Markov model, with an unknown number of states, is applied to the data.

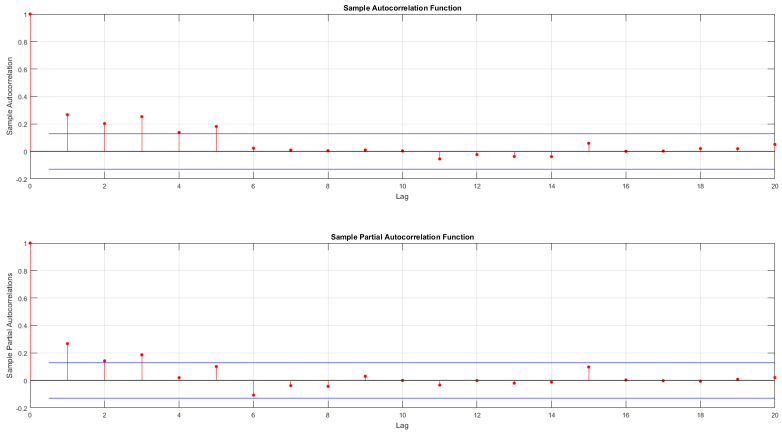

FIGURE 7.2: Lamb Data - Autocorrelation and Partial Autocorrelation Plots

7.1.1 Model Selection

Classical Inference

As far as likelihood ratio testing is concerned, we employed parametric bootstrap for testing r vs. $r+1$ hidden states. In other words, we computed the MLEs $\hat{\boldsymbol{\vartheta}}_r$ and $\hat{\boldsymbol{\vartheta}}_{r+1}$ for the models with r and $r+1$ states, respectively, and the corresponding observed LR statistic, LR_{obs}. Furthermore, we simulated B bootstrapped samples of size equal to the original data from the model with parameter vector $\hat{\boldsymbol{\vartheta}}_r$. For each of those, a bootstrapped LR statistic $\text{LR}^{(b)}_{\text{boot}}$, $b = 1, 2, \ldots, B$, was computed by proceeding exactly as above for the original data. An estimated p-value of the generalised LR test for r vs. $r+1$ hidden states is given by $\frac{N+1}{B+1}$, where N is the number of bootstrapped samples for which $\text{LR}^{(b)}_{\text{boot}} > \text{LR}_{\text{obs}}$.

We performed this bootstrapping procedure for testing $r = 1$ vs. $r = 2$, $r = 2$ vs. $r = 3$ and $r = 3$ vs. $r = 4$ hidden states, using $B = 200$ bootstrapped samples for each test. For the first two tests, the largest bootstrapped LR statistics were smaller than the observed ones, thus $N = 0$ and the estimated p-values were 0.005. In the last test, the estimated p-value was 0.3632, hence LR testing seems to conclude that a Poisson HMM with $r = 3$ hidden states would be suitable to describe the Lamb Data. It should be noted that we did not attempt to test for $r = 4$ vs. $r = 5$ hidden states.

	$r = 1$ vs. $r = 2$	$r = 2$ vs. $r = 3$	$r = 3$ vs. $r = 4$
p-value	0.0050	0.0050	0.3632

TABLE 7.1: Lamb Data - p-values of LR Tests for r vs. $r+1$ Hidden States

By utilising the log-likelihoods which were estimated in the course of the above procedure, we additionally calculated the AIC and BIC of each of the competing models. The AIC is minimised for the model with $r = 3$ hidden states, a result which is in accordance with the results obtained from LR testing. On the other hand, the BIC, which favours less complex models, is minimised for the model with just $r = 2$ hidden states.

	$r = 1$	$r = 2$	$r = 3$	$r = 4$
AIC	404.0873	364.3148	351.7248	362.3793
BIC	407.5679	378.2374	383.0505	418.0695

TABLE 7.2: Lamb Data - AIC and BIC for $r = 1$ to $r = 4$ Hidden States

The main point we want to make here, however, is that the above computations were slow. As a whole, the above procedure took $5,874$ seconds of CPU time to run using Matlab on a machine with a quad-core processor, running on 3.50 GHz. Obviously, these computations would have been must faster if implemented in another programming language, e.g. C++.

Bayesian Inference

For the Bayesian analysis we employed a reversible jump MCMC sampler. A uniform prior over $\{1, 2, \ldots, R\}$ with $R = 4$ was put on r. The transition probabilities were parametrised by q_{ij} for $i, j = 1, 2, \ldots, r$, where each q_{ij} was given an independent Exponential prior with unit mean. This implies an independent uniform $\text{Dir}(1, 1, \ldots, 1)$ prior over each row of $\mathbf{P}$.

We ran this reversible jump sampler for $100,000$ total sweeps with a burn-in period of $50,000$ sweeps. Using a Matlab implementation, the total computation time was 554 seconds on the same machine as above, which is not even comparable to the computation times required for the previous bootstrap analysis.

The mean acceptance probability was 51% for the birth-death move and 40% for the split-combine move. We remark that obtaining satisfying acceptance rates for dimension-changing moves in HMMs is generally difficult. The estimated posterior probabilities are given in Table 7.3. The degree of belief in $r = 3$ vs. $r = 4$ is comparable to what was obtained with the bootstrap analysis, whereas the result for $r = 2$ vs. $r = 3$ is entirely different. Here, $r = 2$ appears to be a plausible model, though it was firmly rejected by the generalised LR test.

Next, we ran the birth-death sampler with the same specifications as above, but with a truncated Poisson(1) prior over $\{1, 2, 3, 4\}$, in order to avoid sampling from

models with too many states. We also selected the birth rate, $\lambda_b = 1$. Again, using a Matlab implementation, the total computation time was $1,589$ seconds on the same machine as above, which is three times the computation time required for the reversible jump approach, but still much faster than the bootstrap analysis.

The estimated posterior probabilities given by the birth-death sampler are also shown in Table 7.3. We remark that the results obtained by the two different trans-dimensional MCMC samplers are markedly different. The posterior probabilities estimated via the birth-death MCMC sampler seem to concur with the results of the bootstrap analysis. Namely the model with $r = 3$ is greatly favoured against the model with $r = 2$ states, whereas it is slightly favoured against the model with $r = 4$, which seems as a viable alternative.

	$r = 1$	$r = 2$	$r = 3$	$r = 4$
Reversible Jump	0.1165	0.3155	0.3792	0.1889
Birth-Death	0.0000	0.0729	0.5845	0.3426
Importance Sampling	0.0000	0.4784	0.5209	0.0007
Reciprocal Importance Sampling	0.0000	0.7732	0.2268	0.0000
Bridge Sampling	0.0000	0.4394	0.5562	0.0044

TABLE 7.3: Lamb Data - Posterior Probabilities for $r = 1$ to $r = 4$ Hidden States

Lastly, we approximated the marginal likelihoods of models with $r = 1$ to $r = 4$ hidden states. A uniform prior over $\{1, 2, 3, 4\}$ was chosen for r. The posterior distribution of $\boldsymbol{\vartheta}_r$ was estimated through the permutation sampling algorithm based on the priors $(p_{i,1}, \ldots, p_{i,r}) \sim \text{Dir}(1, \ldots, 1)$ and $\lambda_i \sim \text{Gamma}(b, B)$ for $i = 1, 2, \ldots, r$, where $b = \dfrac{\bar{y}^2}{s^2 - \bar{y}^2}$ and $B = \dfrac{b}{\bar{y}}$.

Naturally, the marginal likelihood of the model with $r = 1$ was computed analytically, leading to the following expression

$$f(\mathbf{y}|\mathcal{M}_1) = \frac{B^b}{\Gamma(b)} \cdot \frac{\Gamma(S(\mathbf{y}) + b)}{(B + n + 1)^{S(\mathbf{y})+b}} \cdot \prod_{k=0}^{n} \frac{1}{y_k!}, \tag{7.1}$$

where $S(\mathbf{y}) = \sum_{k=0}^{n} y_k$.

The rest of the marginal likelihoods were estimated given $M = L = 10,000$ draws from the posterior and the importance densities of $\boldsymbol{\vartheta}_r$ respectively. A burn-in of $1,000$ draws was allowed for the permutation sampling algorithm, whereas $T = 100$ randomly selected MCMC draws of the state sequence $\mathbf{x}$ were used to approximate

the proposal density. The importance sampling estimator was chosen as the starting value of the bridge sampling algorithm.

Once again, the results are summarised in Table 7.3 and there is clear evidence against the hypothesis of homogeneity ($r = 1$). The total computation time for the estimation of all the marginal likelihoods was just 296 seconds of CPU time, using the same machine, which is in line with the rest of the Bayesian methods applied for model selection. Nevertheless, the estimated marginal likelihoods appear to be highly sensitive to the estimation method and, more importantly, to the choice of a prior distribution over $\boldsymbol{\vartheta}_r$.

7.1.2 Parameter Estimation

The EM algorithm was run for the model with $r = 3$ hidden states. The initial values of the means were computed as $\lambda_i = \min y_k + \frac{R(\mathbf{y})}{2d} + \frac{(i-1)R(\mathbf{y})}{d}$ for $i = 1, 2, 3$, where $R(\mathbf{y}) = \max y_k - \min y_k$. Afterwards, we employed parametric bootstrap to compute 95% confidence intervals for the MLEs, via the percentiles defined by the empirical distribution of the parameters estimated from $B = 1,000$ bootstrapped samples. The total computation time required for this estimation procedure was $4,390$ seconds on the same machine. The results are available in Table 7.4.

	MLE	**Confidence Interval**	**MAP**	**Posterior Mean**	**Credibility Interval**
λ_1	0.0398	[0.0000, 0.2514]	0.0357	0.0741	[0.0037, 0.2159]
λ_2	0.4937	[0.0315, 2.4582]	0.4919	0.4537	[0.1813, 0.8056]
λ_3	3.4106	[0.4426, 6.4501]	3.0057	2.4549	[1.0890, 4.2023]
$p_{1,1}$	0.9487	[0.0000, 0.9909]	0.9479	0.8366	[0.1334, 0.9757]
$p_{1,2}$	0.0409	[0.0000, 0.8479]	0.0422	0.1265	[0.0045, 0.7547]
$p_{1,3}$	0.0104	[0.0000, 0.5641]	0.0100	0.0369	[0.0016, 0.1477]
$p_{2,1}$	0.0400	[0.0000, 1.0000]	0.0427	0.1357	[0.0069, 0.6753]
$p_{2,2}$	0.9600	[0.0000, 0.9931]	0.9573	0.8204	[0.1789, 0.9765]
$p_{2,3}$	0.0000	[0.0000, 0.9428]	0.0000	0.0439	[0.0010, 0.2126]
$p_{3,1}$	0.1848	[0.0000, 1.0000]	0.1826	0.2157	[0.0101, 0.5585]
$p_{3,2}$	0.0000	[0.0000, 1.0000]	0.0012	0.2171	[0.0077, 0.5865]
$p_{3,3}$	0.8152	[0.0000, 0.9824]	0.8162	0.5673	[0.2323, 0.8597]

TABLE 7.4: Lamb Data - Parameter Estimation for $r = 3$ Hidden States

For the Gibbs sampler, the parameters were initialised as described in Chapters 4 and 5 and the sampler was run for $11,000$ total sweeps, of which the first $1,000$

were discarded as a burn-in period. Even though the EM algorithm required to be coupled with a bootstrapping procedure in order to provide confidence intervals for the estimates, Gibbs sampling naturally gives 95% credibility intervals through the draws of the posterior distribution. Hence, the computation times were just 26 seconds for the Gibbs sampler with local updating and 37 seconds for the Gibbs sampler with global updating. A summary of the results obtained from the global updating algorithm is also available in Table 7.4.

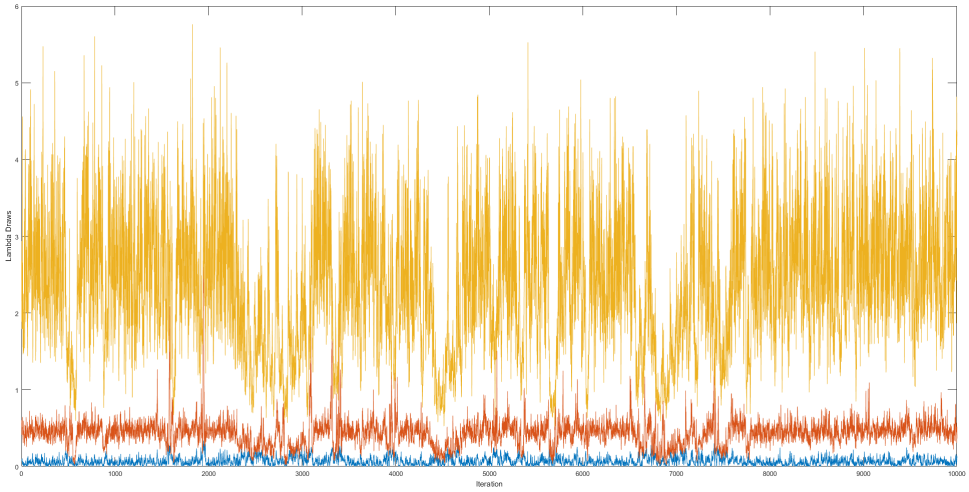

FIGURE 7.3: Lamb Data - Draws of $\boldsymbol{\lambda}$ from the Gibbs Sampler

Lastly, we ran the SAME algorithm for $L = 10,000$ iterations with a linear cooling schedule $M_\ell = [0.01 \cdot \ell + 1]$ for $\ell = 1, 2, \ldots, L$. The computation time required for this method was $1,014$ seconds and the resulting maximum a posteriori (MAP) estimates are available in Table 7.4.

7.2 Simulated Data from a Normal HMM

To further compare the various methods of model selection and parameter estimation developed in previous chapters, we performed the following simulation experiment. We simulated a set of data consisting of 300 observations from a Normal HMM with $r = 3$ states and true parameter values $\boldsymbol{\mu} = (-2, 0, 2)$, $\boldsymbol{\sigma}^2 = (0.25, 2.25, 1)$ and

$$\mathbf{P} = \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.3 & 0.2 & 0.5 \\ 0.5 & 0.4 & 0.1 \end{bmatrix}.$$

The chain is assumed to be stationary. Nevertheless, it is convenient to also include the initial probabilities as separate parameters in the model and we do so even if these are implicitly given by the stationarity assumption. Figure 7.4 displays the densities

of the Normal components, weighted by their stationary probabilities, (dashed lines) and the marginal density of a single observation from this Normal HMM (solid lines).

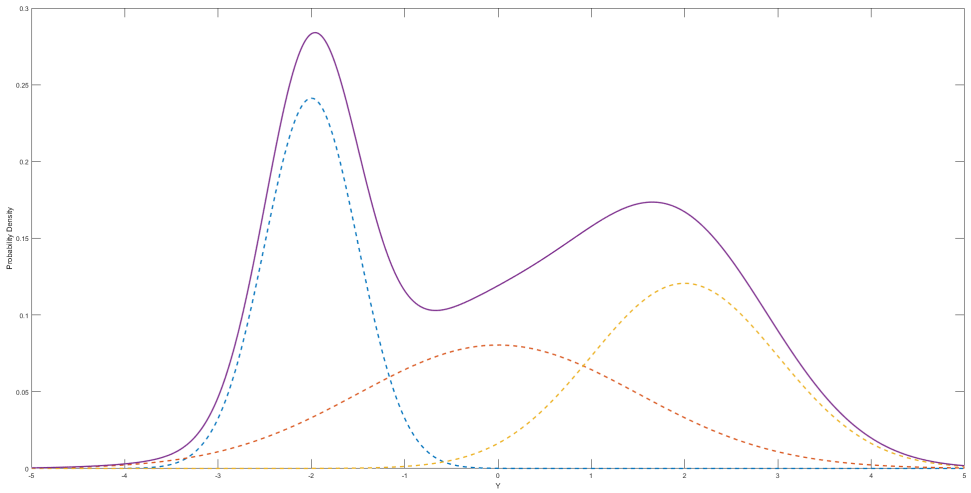

FIGURE 7.4: Weighted Component and Marginal Densities of Normal HMM

7.2.1 Model Selection

Classical Inference

We performed the same bootstrapping procedure for testing $r = 1$ vs. $r = 2$, $r = 2$ vs. $r = 3$ and $r = 3$ vs. $r = 4$ hidden states, using $B = 200$ bootstrapped samples for each test. This procedure took $4,732$ seconds to run using Matlab. In the last test, the estimated p-value was 0.8806, hence LR testing strongly suggests that a Normal HMM with $r = 3$ hidden states would be suitable to describe the simulated data.

	$r = 1$ vs. $r = 2$	$r = 2$ vs. $r = 3$	$r = 3$ vs. $r = 4$
p-value	0.0050	0.0100	0.8806

TABLE 7.5: Normal Data - p-values of LR Tests for r vs. $r+1$ Hidden States

	$r = 1$	$r = 2$	$r = 3$	$r = 4$
AIC	$1,249$	$1,176$	$1,157$	$1,167$
BIC	$1,257$	$1,198$	$1,202$	$1,241$

TABLE 7.6: Normal Data - AIC and BIC for $r = 1$ to $r = 4$ Hidden States

We also calculated the AIC and BIC of each of the competing models. The AIC is minimised for the model with $r = 3$ hidden states, a result which is in accordance

with the results obtained from LR testing. On the other hand, the BIC, which favours less complex models, is minimised for the model with just $r = 2$ hidden states.

Bayesian Inference

For the Bayesian analysis we employed a reversible jump MCMC sampler with the exact same specifications as for the Lamb Data. Using a Matlab implementation, the total computation time was 414 seconds on the same machine as above. The mean acceptance probability was 25% for the birth-death move and 24% for the split-combine move. The estimated posterior probabilities are given in Table 7.3. The results of the reversible jump algorithm are rather inconclusive, since the models with $r = 2$ and $r = 3$ hidden states are both assigned almost equal posterior probabilities.

Next, we ran the birth-death sampler, again, with the same specifications as beforehand. Using Matlab, the total computation time was $1,956$ seconds. The estimated posterior probabilities given by the birth-death sampler are shown in Table 7.7. They seem to concur with the results of the bootstrap analysis. Namely, the model with $r = 3$ is greatly favoured against the models with $r = 2$ and $r = 4$ states.

	$r = 1$	$r = 2$	$r = 3$	$r = 4$
Reversible Jump	0.0010	0.4275	0.4340	0.1375
Birth-Death	0.0000	0.0112	0.7366	0.2522
Importance Sampling	0.0000	0.0467	0.9532	0.0001
Reciprocal Importance Sampling	0.0000	0.3124	0.6876	0.0000
Bridge Sampling	0.0000	0.0601	0.9397	0.0002

TABLE 7.7: Normal Data - Posterior Probabilities for $r = 1$ to $r = 4$ Hidden States

Lastly, we estimated the marginal likelihoods of models with $r = 1$ to $r = 4$ hidden states. The posterior distribution of $\boldsymbol{\vartheta}_r$ was estimated based on the priors $(p_{i,1}, \ldots, p_{i,r}) \sim \text{Dir}(1, \ldots, 1)$, $\sigma_i^2 \sim \text{Inv-Gamma}(c, C)$ and $\mu_i | \sigma_i^2 \sim \mathcal{N}(b, N^{-1}\sigma_i^2)$ for $i = 1, 2, \ldots, r$, where $c = 2.5$, $C = 0.5 \cdot s^2$, $b = \bar{y}$ and $N = 1$.

For this prior specification, the conditional posterior distributions of μ_i and σ_i^2 for $i = 1, 2, \ldots, r$ are given by

$$\pi\left(\sigma_i^2 \middle| \mathbf{y}, \mathbf{x}\right) \propto \left(\sigma_i^2\right)^{-\frac{N_i(\mathbf{x})}{2} - c - 1} \cdot \exp\left\{ - \left[C + \frac{V_i(\mathbf{y}) + Nb^2}{2} - \frac{(S_i(\mathbf{y}) + Nb)^2}{2(N_i(\mathbf{x}) + N)} \right] \frac{1}{\sigma_i^2} \right\},$$

$$\pi\left(\mu_i \middle| \mathbf{y}, \mathbf{x}, \sigma_i^2\right) \propto \exp\left\{ - \frac{N_i(\mathbf{x}) + N}{2\sigma_i^2} \mu_i^2 + \frac{S_i(\mathbf{y}) + Nb}{\sigma_i^2} \mu_i \right\},$$

where $N_i(\mathbf{x}) = \sum_{k=0}^{n} \mathbb{1}\{x_k = i\}$, $S_i(\mathbf{y}) = \sum_{k:x_k=i} y_k$ and $V_i(\mathbf{y}) = \sum_{k:x_k=i} y_k^2$, which correspond to $\sigma_i^2 \big| \mathbf{y}, \mathbf{x} \sim \text{Inv-Gamma}\left(\frac{N_i(\mathbf{x})}{2} + c, C + \frac{V_i(\mathbf{y}) + Nb^2}{2} - \frac{(S_i(\mathbf{y}) + Nb)^2}{2(N_i(\mathbf{x}) + N)}\right)$ and $\mu_i \left| \mathbf{y}, \mathbf{x}, \sigma_i^2 \right. \sim \mathcal{N}\left(\frac{S_i(\mathbf{y}) + Nb}{N_i(\mathbf{x}) + N}, \frac{\sigma_i^2}{N_i(\mathbf{x}) + N}\right)$ for $i = 1, 2, \ldots, r$.

Naturally, the marginal likelihood of the model with $r = 1$ was computed analytically, leading to the following expression

$$f(\mathbf{y}|\mathcal{M}_1) = (2\pi)^{-\frac{n+1}{2}} \cdot \sqrt{\frac{N}{N+n+1}} \cdot \frac{C^c}{\Gamma(C)} \cdot \Gamma\left(c + \frac{n+1}{2}\right) \cdot [C(\mathbf{y})]^{-c-\frac{n+1}{2}}, \quad (7.2)$$

where $C(\mathbf{y}) = C + \frac{V(\mathbf{y}) + Nb^2}{2} - \frac{(S(\mathbf{y}) + Nb)^2}{2(N+n+1)}$, $S(\mathbf{y}) = \sum_{k=0}^{n} y_k$ and $V(\mathbf{y}) = \sum_{k=0}^{n} y_k^2$.

The reciprocal importance sampling estimator was chosen as the starting value of the bridge sampling algorithm. The total computation time for the estimation of all the marginal likelihoods was just 366 seconds of CPU time, using the same machine. Once again, the results are summarised in Table 7.7. All methods for approximating the marginal likelihoods concur that the best fitting model to describe our simulated data is the correct one, namely the one with $r = 3$ hidden states.

7.2.2 Parameter Estimation

We ran the Gibbs sampler for inferring the model with $r = 3$ hidden states. The parameters were initialised as described in Chapters 4 and 5 and the sampler was run for $11,000$ total sweeps, of which the first $1,000$ were discarded as burn-in. The computation times were just 32 seconds for the Gibbs sampler with local updating and 47 seconds for the Gibbs sampler with global updating. A summary of the obtained results through the global updating algorithm is also available in Table 7.8.

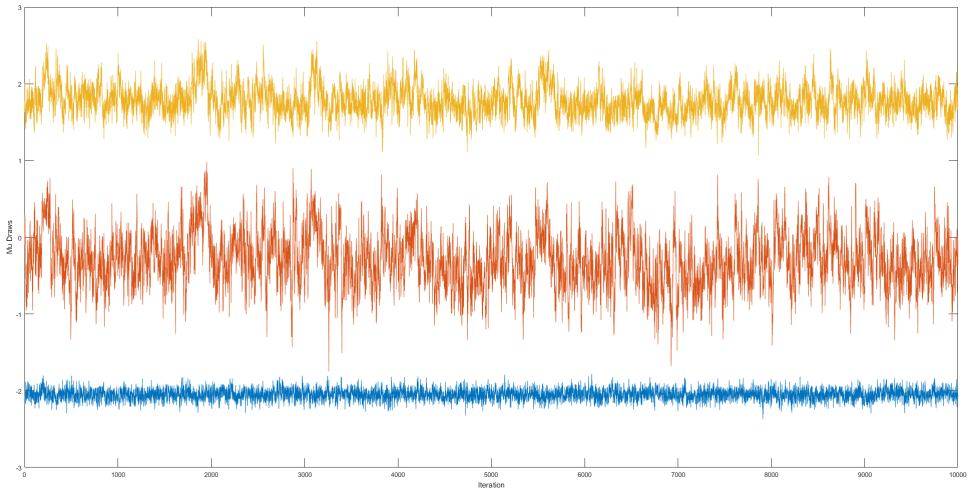

FIGURE 7.5: Normal Data - Draws of $\boldsymbol{\mu}$ from the Gibbs Sampler

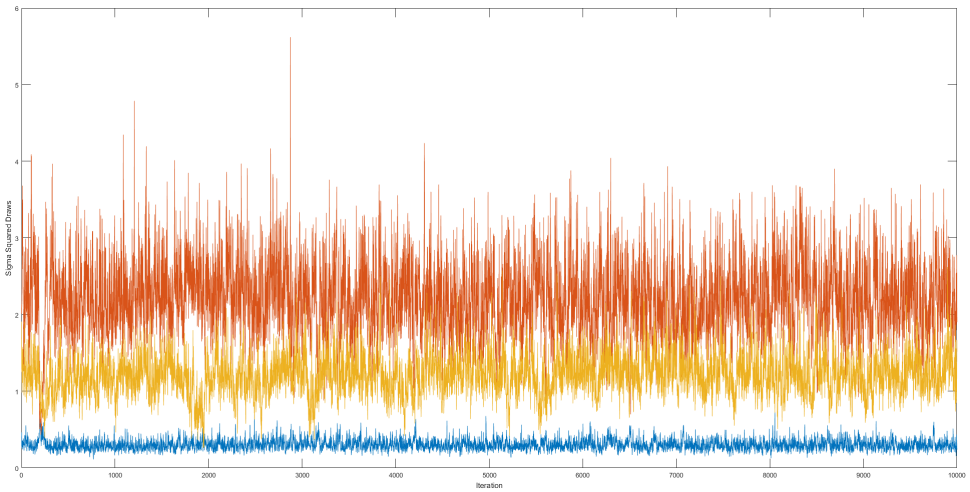

FIGURE 7.6: Normal Data - Draws of $\boldsymbol{\sigma}^2$ from the Gibbs Sampler

	MLE	**Confidence Interval**	**MAP**	**Posterior Mean**	**Credibility Interval**
μ_1	-2.0498	$[-2.2053, -1.9067]$	-2.0574	-2.0489	$[-2.1918, -1.9064]$
μ_2	-0.4799	$[-1.2849, 0.6544]$	-0.5847	-0.2963	$[-0.9656, 0.4315]$
μ_3	1.6546	$[1.3983, 2.1846]$	1.6600	1.7798	$[1.4239, 2.2160]$
σ_1^2	0.2561	$[0.1399, 0.4170]$	0.2624	0.3037	$[0.1939, 0.4656]$
σ_2^2	2.1908	$[0.6515, 3.2395]$	1.9319	2.2009	$[1.3260, 3.2255]$
σ_3^2	1.3470	$[0.6403, 1.7718]$	1.2894	1.2099	$[0.7055, 1.7649]$
$p_{1,1}$	0.1202	$[0.0000, 0.2899]$	0.1159	0.1302	$[0.0145, 0.2728]$
$p_{1,2}$	0.5954	$[0.3104, 0.8880]$	0.6067	0.6071	$[0.3438, 0.8496]$
$p_{1,3}$	0.2845	$[0.0401, 0.4697]$	0.2774	0.2626	$[0.0778, 0.4724]$
$p_{2,1}$	0.1176	$[0.0000, 0.3140]$	0.1327	0.1486	$[0.0136, 0.3161]$
$p_{2,2}$	0.0084	$[0.0000, 0.4578]$	0.0001	0.1775	$[0.0085, 0.4626]$
$p_{2,3}$	0.8739	$[0.3105, 1.0000]$	0.8672	0.6739	$[0.3369, 0.9123]$
$p_{3,1}$	0.5413	$[0.3706, 0.8284]$	0.5217	0.5626	$[0.3937, 0.7527]$
$p_{3,2}$	0.2571	$[0.0000, 0.4700]$	0.2510	0.2699	$[0.0938, 0.4603]$
$p_{3,3}$	0.2015	$[0.0000, 0.4029]$	0.2273	0.1675	$[0.0192, 0.3605]$

TABLE 7.8: Normal Data - Parameter Estimation for $r = 3$ Hidden States

The EM algorithm was also run for the model with $r = 3$ hidden states. The initial values were computed as $\mu_i = \min y_k + \frac{R(\mathbf{y})}{2d} + \frac{(i-1)R(\mathbf{y})}{d}$ and $\sigma_i = s^2$ for $i = 1, 2, 3$, where $R(\mathbf{y}) = \max y_k - \min y_k$. Afterwards, we employed parametric bootstrap to compute 95% confidence intervals for the MLEs, using $B = 1,000$ bootstrapped samples. The total computation time required for this estimation procedure was $3,685$ seconds on the same machine. The results are available in Table 7.8.

Lastly, we ran the SAME algorithm for $L = 10,000$ iterations with a linear cooling schedule $M_\ell = [0.01 \cdot \ell + 1]$ for $\ell = 1, 2, \ldots, L$. The computation time required for this method was $1,270$ seconds and the resulting maximum a posteriori (MAP) estimates are available in Table 7.4.

7.3 Simulated Data from a Multivariate Normal HMM

Lastly, we simulated a set of data consisting of 300 observations from a Multivariate Normal HMM with $d = 2$ dimensions, $r = 2$ hidden states and true parameter values $\boldsymbol{\mu}_1 = (-1, 1)$, $\boldsymbol{\mu}_2 = (1, -1)$,

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 2 & 1 \\ 1 & 5 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \text{ and } \mathbf{P} = \begin{bmatrix} 0.3 & 0.7 \\ 0.8 & 0.2 \end{bmatrix}.$$

The chain is assumed to be stationary. Nevertheless, it is convenient to also include the initial probabilities as separate parameters in the model and we do so even if these are implicitly given by the stationarity assumption. Figure 7.7 shows the marginal density of the Multivariate Normal HMM, whereas Figure 7.8 illustrates the contours of its surface.

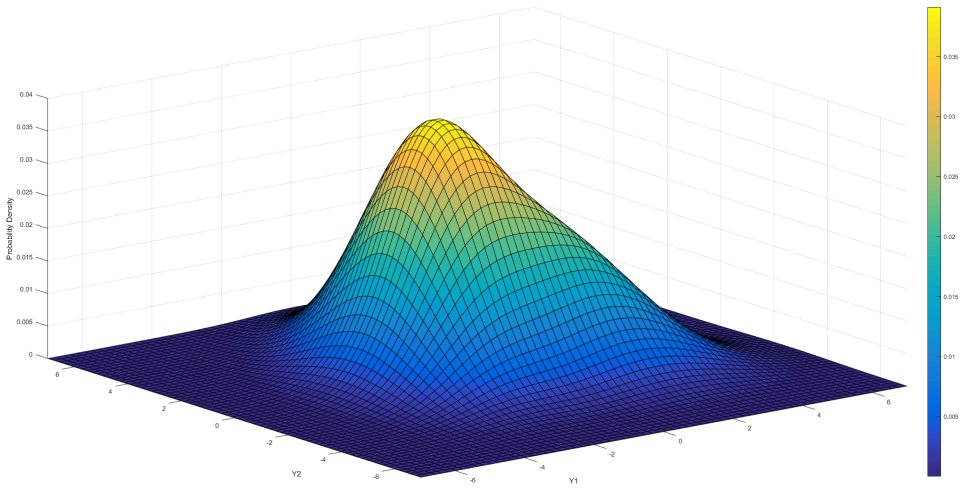

FIGURE 7.7: Marginal Density of Multivariate Normal HMM

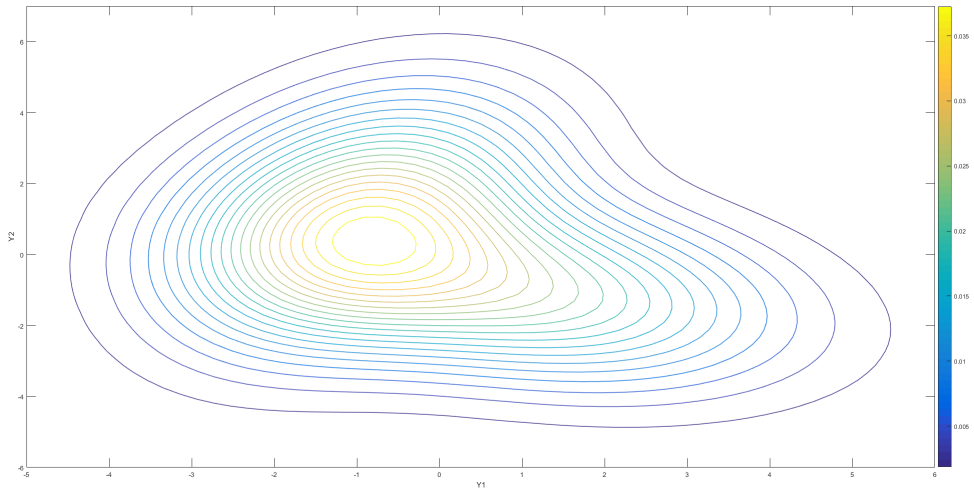

FIGURE 7.8: Contours of the Marginal Distribution of Multivariate Normal HMM

7.3.1 Model Selection

Classical Inference

We performed the same bootstrapping procedure for testing $r = 1$ vs. $r = 2$ and $r = 2$ vs. $r = 3$ hidden states, using $B = 200$ bootstrapped samples for each test. This procedure took $7,895$ seconds to run using Matlab. In the last test, the estimated p-value was 0.4726, hence LR testing seems to conclude that a Normal HMM with $r = 3$ hidden states would be suitable to describe the simulated data.

	$r = 1$ vs. $r = 2$	$r = 2$ vs. $r = 3$
p-value	0.0050	0.4726

TABLE 7.9: Multivariate Normal Data - p-values of LR Tests for r vs. $r + 1$ Hidden States

We note that several complications arose in the bootstrap analysis to estimate the p-value of the LR test for $r = 2$ vs. $r = 3$ hidden states. Since the bootstrapped samples for this test were drawn from a hidden Markov model with $r = 2$ hidden states and parameter vector $\hat{\boldsymbol{\vartheta}}_2$, estimated from the original data, the EM algorithm was sometimes unable to estimate the over-parametrised model with $r = 3$ hidden states, based on these samples. The cause for that was that the smoothing probabilities $\phi_{k|n}(3)$ were close to zero for the majority of observations, leading to the estimated covariance matrix $\boldsymbol{\Sigma}_3$ being close to singular. Hence, the EM algorithm was terminated in those instances, having been unable to properly estimate the required parameter vector.

By utilising the log-likelihoods which were estimated in the course of the above parametric procedure, we additionally calculated the AIC and BIC of each of the

competing models. Both the AIC and the BIC are minimised for the model with $r = 3$ hidden states, results that both contradict the consensus of the previous bootstrap analysis. We note, however, that the values of the AIC and the BIC corresponding to each of the competing models are very close to each other, so no reliable conclusion can be drawn.

	$r = 1$	$r = 2$	$r = 3$
AIC	2,578	2,525	2,511
BIC	2,596	2,570	2,556

TABLE 7.10: Multivariate Normal Data - AIC and BIC for $r = 1$ to $r = 3$ Hidden States

Bayesian Inference

We ran the birth-death sampler with the exact same specifications as for the Lamb Data. Again, using a Matlab implementation, the total computation time was 10,086 seconds. The estimated posterior probabilities given by the birth-death sampler are shown in Table 7.11. They seem to concur with the results of the bootstrap analysis. Namely the model with $r = 2$ is greatly favoured against the model with $r = 1$ states, whereas it is slightly less favoured against the model with $r = 3$.

	$r = 1$	$r = 2$	$r = 3$
Birth-Death	0.0000	0.8110	0.1890

TABLE 7.11: Multivariate Normal Data - Posterior Probabilities for $r = 1$ to $r = 3$ Hidden States

7.3.2 Parameter Estimation

The EM algorithm was run for the model with $r = 2$ hidden states. The initial values were computed as $\boldsymbol{\mu}_{i,\ell} = \min_{0 \leqslant k \leqslant n} y_{\ell,k} + \frac{R(\mathbf{y}_\ell)}{2d} + \frac{(i-1)R(\mathbf{y}_\ell)}{d}$ and $\boldsymbol{\Sigma}_i = \text{Cov}(\mathbf{y})$, where $R(\mathbf{y}_\ell) = \max_{0 \leqslant k \leqslant n} y_{\ell,k} - \min_{0 \leqslant k \leqslant n} y_{\ell,k}$, for $i = 1, 2$ and $\ell = 1, 2$. Afterwards, we employed parametric bootstrap to compute 95% confidence intervals for the MLEs, using $B = 1,000$ bootstrapped samples. The total computation time required for this estimation procedure was 4,536 seconds on the same machine. The results obtained are available in Table 7.12.

For the Gibbs sampler, the parameters were initialised as described in Chapters 4 and 5 and the sampler was run for 11,000 total sweeps, of which the first 1,000 were discarded as a burn-in period. A summary of the obtained results through the global updating algorithm is also available in Table 7.12.

	MLE	**Confidence Interval**	**Posterior Mean**	**Credibility Interval**
$\mu_{1,1}$	-0.8036	$[-1.1508, -0.4871]$	-0.8068	$[-1.0933, -0.5222]$
$\mu_{1,2}$	1.1973	$[-0.2075, 1.6520]$	1.2049	$[0.7924, 1.6380]$
$\mu_{2,1}$	1.2465	$[0.6179, 2.1791]$	1.1713	$[0.6019, 1.8176]$
$\mu_{2,2}$	-1.1709	$[-1.7486, 0.8273]$	-1.0909	$[-1.7300, -0.4367]$
$\sigma^2_{1,1}$	1.9337	$[1.3837, 2.5515]$	1.9334	$[1.4598, 2.4916]$
$\sigma^2_{1,2}$	4.3639	$[2.6003, 5.4314]$	4.4365	$[3.4150, 5.7621]$
Cov_1	0.8710	$[-0.6151, 1.4502]$	0.9061	$[0.3150, 1.5805]$
$\sigma^2_{2,1}$	3.8696	$[2.1820, 5.2536]$	3.9646	$[2.8822, 5.4041]$
$\sigma^2_{2,2}$	2.7126	$[1.7000, 7.4875]$	2.8889	$[1.7614, 4.3581]$
Cov_2	-0.8384	$[-3.0423, 0.1434]$	-0.9144	$[-1.9005, 0.0280]$
$p_{1,1}$	0.3925	$[0.1150, 0.7504]$	0.3556	$[0.0501, 0.5995]$
$p_{1,2}$	0.6075	$[0.2453, 0.8802]$	0.6444	$[0.4004, 0.9497]$
$p_{2,1}$	0.8843	$[0.4236, 1.0000]$	0.8415	$[0.6472, 0.9781]$
$p_{2,2}$	0.1157	$[0.0000, 0.5755]$	0.1585	$[0.0219, 0.3525]$

TABLE 7.12: Multivariate Normal Data - Parameter Estimation for $r = 2$ Hidden States

Chapter 8

Conclusion

We have, thus, compared inference for hidden Markov models primarily based on the EM algorithm and parametric bootstrap, on one hand, and on the Gibbs sampler and other MCMC algorithms, on the other hand. In situations where one requires only a point estimate of the parameter vector or where it suffices to compare models only through some information criterion, then the EM algorithm is typically the quickest and most straightforward way to achieve this. Nevertheless, in the case where a point estimate is not sufficient, the comparison between the two approaches is not an elementary task.

In the examples presented in Chapter 7, we saw that the dependent posterior samples provided by the Gibbs sampling algorithm boast very swift computation times. On the other hand, while the i.i.d. replicates provided by parametric bootstrap require much longer run times, no further analyses of autocorrelations are required in order to assess the precision of the results. The results obtained from all of the proposed algorithms are very promising so the choice between them is left to the reader's taste.

Having said that, it is important to be aware that inference in HMMs, whether frequentist or Bayesian, is usually a complex and far from automated task. Even in simple models the multi-modality of the observed likelihood may potentially be causing the EM algorithm to converge to local, rather than global, maxima and the MCMC algorithms to have poor mixing properties. A further problem is slow convergence of the EM algorithm and slow mixing of the Gibbs sampler occurring when the amount of information carried by the complete data is much greater than the amount carried by the observed data alone.

Although a Bayesian approach to HMMs may be appealing from several perspectives, it is the author's experience that users of HMMs often consider writing the computer code necessary to implement such procedures a prohibitive exercise. In particular reversible jump MCMC algorithms have a somewhat unjustified reputation

for being difficult to derive and implement. Hence, it is clear that readily available software packages would be extremely beneficial for making such methods available to a wider audience of researchers in statistics and other scientific fields. Future work certainly includes the development of such packages for various programming languages, including Matlab, R, Python, C and Java.

Appendix A

Markov Chain Monte Carlo Methods

A.1 The Metropolis-Hastings Algorithm

Suppose that we want to simulate a sequence of random variables with probability mass function $\pi_i = B^{-1}b_i$ for $i = 1, 2, \ldots, m$, where $B = \sum_{i=1}^{m} b_i$ is the normalising constant of the distribution. Suppose, also, that m is large and that the normalising constant B is difficult to calculate. One way of simulating a sequence of identically distributed random variables whose common distribution converges to π_i, $i = 1, 2, \ldots, m$, is to construct a Markov chain whose limiting distribution coincides with the distribution we desire to simulate from.

Let $\mathbf{P} = [p_{ij}]$ be an irreducible Markov transition probability matrix over the integers $\{1, 2, \ldots, m\}$. The Metropolis-Hastings algorithm defines a discrete-time stochastic process $\{X_n\}_{n \geqslant 0}$ in the following manner. Given that $X_n = i$, a random variable X is generated according to the probability distribution given by the i^{th} row of $\mathbf{P}$, i.e. $P(X = j) = p_{ij}$ for $j = 1, 2, \ldots, m$. If $X = j$, then

$$X_{n+1} = \begin{cases} X & \text{with probability } a_{ij} \\ X_n & \text{with probability } 1 - a_{ij} \end{cases}.$$

It is, thus, easy to verify that $\{X_n\}_{n \geqslant 0}$ constitutes a discrete-time Markov chain with transition probabilities P_{ij} given by

$$P_{ij} = \begin{cases} p_{ij}a_{ij}, & j \neq i \\ p_{ii} + \sum_{k \neq i} p_{ik}(1 - a_{ik}), & j = i \end{cases}.$$

Now, we demand that $\{X_n\}_{n\geqslant 0}$ be time-reversible with unique stationary distribution π_i, $i = 1, 2, \ldots, m$, given by the detailed balance equations $\pi_i P_{ij} = \pi_j P_{ji}$ for $j \neq i$. It is easy to verify that the detailed balance equations hold if we take

$$a_{ij} = \min\left\{\frac{\pi_j p_{ji}}{\pi_i p_{ij}}, 1\right\} = \min\left\{\frac{b_j p_{ji}}{b_i p_{ij}}, 1\right\}.$$

Hence, we note that knowledge of B is not required to define the Markov chain, since the values of b_i for $i = 1, 2, \ldots, m$ suffice. Lastly, in order for the limiting distribution of $\{X_n\}_{n\geqslant 0}$ to coincide with its stationary distribution, π_i for $i = 1, 2, \ldots, m$, we require that $\{X_n\}_{n\geqslant 0}$ be aperiodic. A sufficient condition is that $P_{ii} > 0$ for some $i \in \{1, 2, \ldots, m\}$.

The following steps sum up the Metropolis-Hastings algorithm for generating a sequence of identically distributed random variables from a given distribution with an intractable normalising constant

1. Choose an irreducible Markov transition probability matrix $\mathbf{P} = [p_{ij}]$ over the integers $\{1, 2, \ldots, m\}$.
2. Set $X_0 = k$ for some integer $k \in \{1, 2, \ldots, m\}$ and $n = 0$.
3. Given that $X_n = i$, generate a random variable X such that $P(X = j) = q_{ij}$.
4. Generate a random variable $U \sim \mathcal{U}(0, 1)$. Given that $X = j$, then

$$X_{n+1} = \begin{cases} X, & \text{if } U < \dfrac{b_j p_{ji}}{b_i p_{ij}} \\ X_n, & \text{otherwise} \end{cases}.$$

5. Set $n = n + 1$ and go to step 3.

A.2 The Gibbs Sampler

Suppose that we want to generate a sequence of identically distributed random vectors $\{\mathbf{X}_n\}_{n\geqslant 0}$ with common probability mass function $q(\mathbf{x}) = Cg(\mathbf{x})$, where C is an unknown multiplicative constant and $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Utilisation of the Gibbs sampler presupposes that we can generate a random variable X from the conditional probability distribution $P(X = x) = P(X_i = x | X_j = x_j, j \neq i)$ for some $i \in \{1, 2, \ldots, m\}$. It operates by using the Metropolis-Hastings algorithm on a Markov chain with states $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and transition probabilities defined as below.

Given a present state $\mathbf{x}$, a coordinate i is chosen out of $\{1, 2, \ldots, m\}$ and a random variable X is generated as previously described. Given that $X = x$, then $\mathbf{y} = (x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_m)$ is considered as the next candidate state. In other words, the transition probabilities of the Markov chain are given by

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{m} \cdot P(X_i = x | X_j = x_j, j \neq i) = \frac{q(\mathbf{y})}{m \cdot P(X_j = x_j, j \neq i)}.$$

Since we desire the limiting distribution of the Markov chain to coincide with $q(\mathbf{x})$, the vector $\mathbf{y}$ is accepted as the new state of the Markov chain with probability

$$a(\mathbf{x}, \mathbf{y}) = \min\left\{\frac{q(\mathbf{y})p(\mathbf{y}, \mathbf{x})}{q(\mathbf{x})p(\mathbf{x}, \mathbf{y})}, 1\right\} = \min\left\{\frac{q(\mathbf{y})q(\mathbf{x})}{q(\mathbf{x})q(\mathbf{y})}, 1\right\} = 1.$$

Hence, when utilising the Gibbs sampler, the candidate state is always accepted as the new state of the Markov chain.

The following steps sum up the Gibbs sampling algorithm for generating a sequence of identically distributed random vectors from a given probability distribution

1. Select a starting vector $\mathbf{X}_0$ and set $n = 0$.
2. Given that $\mathbf{X}_n = \mathbf{x} = (x_1, x_2, \ldots, x_m)$, generate the next candidate state $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ by sampling the components in order, starting from the first component. For $i = 1, 2, \ldots, m$, sample y_i according to the distribution specified by $P(X_i = x | X_1 = y_1, \ldots, X_{i-1} = y_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_m = x_m)$.
3. Set $\mathbf{X}_{n+1} = \mathbf{y}$, $n = n + 1$ and go to step 2.

A.3 Simulated Annealing

Let $\mathcal{A}$ be a finite set of vectors and $V : \mathcal{A} \to [0, \infty)$. Suppose that we are interested in specifying its maximal value $V^* = \max_{\mathbf{x} \in \mathcal{A}} V(\mathbf{x})$, as well as at least one vector at which the maximal value is attained, that is, an element in $\mathcal{M} = \{\mathbf{x} \in \mathcal{A} : V(\mathbf{x}) = V^*\}$.

Let $\lambda > 0$ and consider the following probability mass function on the set of vectors $\mathcal{A}$

$$p_\lambda(\mathbf{x}) = \frac{e^{\lambda V(\mathbf{x})}}{\sum_{\mathbf{y} \in \mathcal{A}} e^{\lambda V(\mathbf{y})}}.$$

By multiplying both the numerator and the denominator by $e^{-\lambda V^*}$, we see that

$$p_\lambda(\mathbf{x}) = \frac{e^{\lambda(V(\mathbf{x}) - V^*)}}{\sum_{\mathbf{y} \in \mathcal{A}} e^{\lambda(V(\mathbf{y}) - V^*)}} = \frac{e^{\lambda(V(\mathbf{x}) - V^*)}}{|\mathcal{M}| + \sum_{\mathbf{y} \notin \mathcal{M}} e^{\lambda(V(\mathbf{y}) - V^*)}},$$

where $|\mathcal{M}|$ denotes the cardinality of $\mathcal{M}$. Since $V(\mathbf{x}) - V^* < 0$ for $\mathbf{x} \notin \mathcal{M}$, then, as $\lambda \to \infty$, we obtain

$$\lim_{\lambda \to \infty} p_\lambda(\mathbf{x}) = \frac{\mathbb{1}\{\mathbf{x} \in \mathcal{M}\}}{|\mathcal{M}|}.$$

Hence, if λ is large enough and we generate a Markov chain whose limiting distribution is $p_\lambda(\mathbf{x})$, then most of the mass of the limiting distribution will be concentrated on $\mathcal{M}$. An approach that is often useful is to introduce the notion of neighbouring vectors and utilise a Metropolis-Hastings algorithm to define such a chain. For instance, we could say that two vectors are neighbouring if they differ in only a single coordinate or if the one can be obtained from the other by interchanging two of its components. Given that the current state is $\mathbf{x}$, we could designate that the next state is randomly chosen out of all the neighbouring vectors of $\mathbf{x}$. If the neighbouring vector $\mathbf{y}$ is selected, then that vector becomes the next state with probability

$$\min\left\{\frac{e^{\lambda V(\mathbf{y})}}{e^{\lambda V(\mathbf{x})}} \cdot \frac{|\mathcal{N}(\mathbf{x})|}{|\mathcal{N}(\mathbf{y})|}, 1\right\},$$

where $\mathcal{N}(\mathbf{x})$ denotes the set of neighbouring vectors of $\mathbf{x}$.

One weakness of the preceding algorithm is that, since λ was chosen to be sufficiently large, when the chain enters a state $\mathbf{x}$ with $V(\mathbf{x}) > V(\mathbf{y})$, $\forall \mathbf{y} \in \mathcal{N}(\mathbf{x})$, then it might take a long time for the chain to move to a different state. Hence, it has been proven useful to permit the value of λ to vary with time. A popular variation of the preceding algorithm, referred to as Simulated Annealing, operates in the following way. If $\mathbf{X}_n = \mathbf{x}$ and the vector $\mathbf{y}$ is chosen out of $\mathcal{N}(\mathbf{x})$, then $\mathbf{X}_{n+1}$ is chosen to be $\mathbf{y}$ with probability

$$\min\left\{\frac{e^{\lambda_n V(\mathbf{y})}}{e^{\lambda_n V(\mathbf{x})}} \cdot \frac{|\mathcal{N}(\mathbf{x})|}{|\mathcal{N}(\mathbf{y})|}, 1\right\},$$

where $\{\lambda_n\}_{n \geqslant 0}$ is a predetermined sequence of values, commonly referred to as a cooling schedule, which start out small, thus initially attaining a large number of transitions, and gradually grow, thus achieving convergence to the set of global maxima.

A computationally useful choice of cooling schedule is to let $\lambda_n = C \log(n+2)$ for $n \geqslant 0$, where C is any fixed positive constant. If we generate N successive states $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$, then we can estimate V^* by $\max\limits_{1 \leqslant i \leqslant N} V(\mathbf{X}_i)$ and an element in $\mathcal{M}$ by X_{i^*}, where $i^* = \arg\max\limits_{1 \leqslant i \leqslant N} V(\mathbf{X}_i)$.

Appendix B

Common Probability Distributions

B.1 Discrete Distributions

B.1.1 Poisson Distribution

The Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time if these events occur with a constant rate and independently of the time since the last event.

A discrete random variable X is said to have a Poisson distribution with parameter $\lambda > 0$ if the probability mass function of X is given by

$$f(k;\lambda) = P(X = k;\lambda) = e^{-\lambda} \cdot \frac{\lambda^k}{k!}, \quad k = 0, 1, 2, \ldots.$$

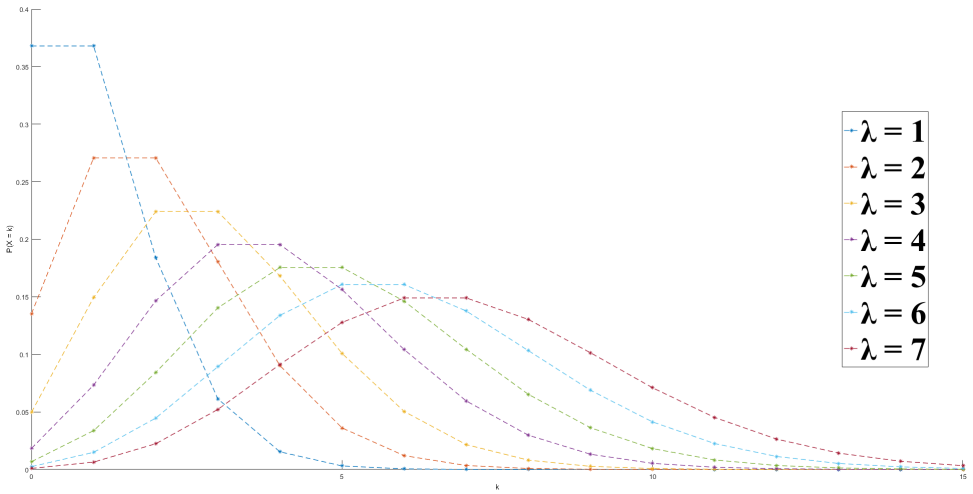

FIGURE B.1: Poisson Distribution - Probability Mass Function

The positive real number λ is equal to the expected value of X, as well as the variance of X, i.e. $E(X) = \mathrm{Var}(X) = \lambda$.

B.2 Continuous Distributions

B.2.1 Beta Distribution

The Beta distribution is a family of continuous probability distributions defined on the interval $[0, 1]$ and parametrised by two positive shape parameters, denoted by α and β. In Bayesian inference, it is the conjugate prior probability distribution for the Bernoulli, Binomial, Negative Binomial and Geometric distributions. For example, it can be used in Bayesian analysis to describe initial knowledge concerning probability of success. It is a suitable model for the random behaviour of percentages and proportions.

The probability density function (pdf) of the Beta distribution is a power function of the variable x and of its reflection $1-x$ as follows

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot x^{\alpha-1} \cdot (1-x)^{\beta-1}, \quad 0 \leqslant x \leqslant 1.$$

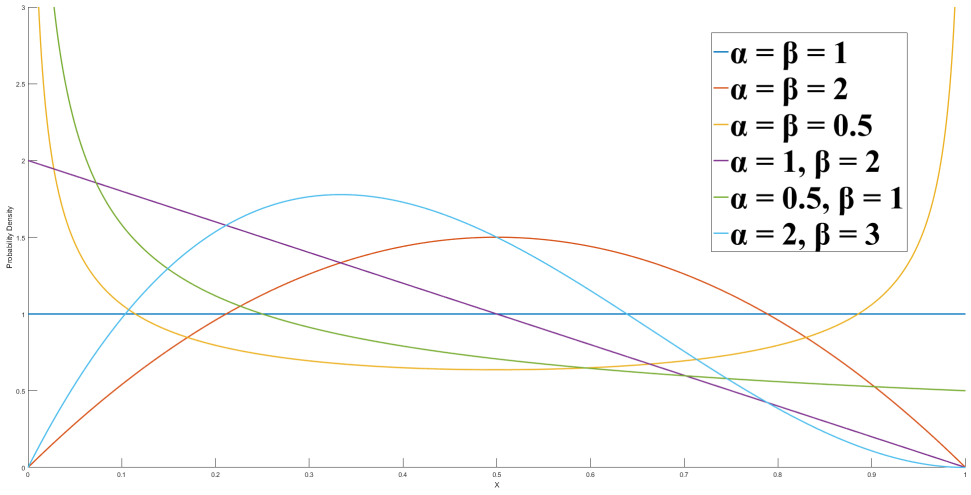

FIGURE B.2: Beta Distribution - Probability Density Function

The mode of a Beta distributed random variable X with $\alpha, \beta > 1$ is given by

$$\text{Mode} = \frac{\alpha - 1}{\alpha + \beta - 2}.$$

When $\alpha, \beta < 1$, this is the anti-mode, i.e. the lowest point of the probability density curve. Letting $\alpha = \beta$, the expression for the mode simplifies to 0.5, showing that for

$\alpha = \beta > 1$ the mode is at the center of the distribution, which is to be expected, since, in that case, it is symmetric around the point $x = 0.5$.

The expected value of a Beta distributed random variable X is a function of only the ratio β/α of the two parameters

$$E(X) = \frac{1}{1 + \beta/\alpha} = \frac{\alpha}{\alpha + \beta}.$$

Letting $\alpha = \beta$ in the above expression, one obtains $E(X) = 0.5$.

The variance of a Beta distributed random variable X is

$$\mathrm{Var}(X) = \frac{\alpha\beta}{(\alpha + \beta)^2 \cdot (\alpha + \beta + 1)}.$$

Letting $\alpha = \beta$ in the above expression one obtains $\mathrm{Var}(X) = \frac{1}{4(2\alpha+1)}$, showing that the variance decreases monotonically as α increases. Approaching the limit at $\alpha = \beta = 0$, one finds the maximum variance, $\mathrm{Var}(X) = 0.25$.

B.2.2 Gamma Distribution

The Gamma distribution is a two-parameter family of continuous probability distributions. One common parametrisation is with a shape parameter α and an inverse scale parameter β, called a rate parameter. The corresponding pdf in the shape-rate parametrisation is

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot x^{\alpha-1} \cdot e^{-\beta x}, \quad x > 0.$$

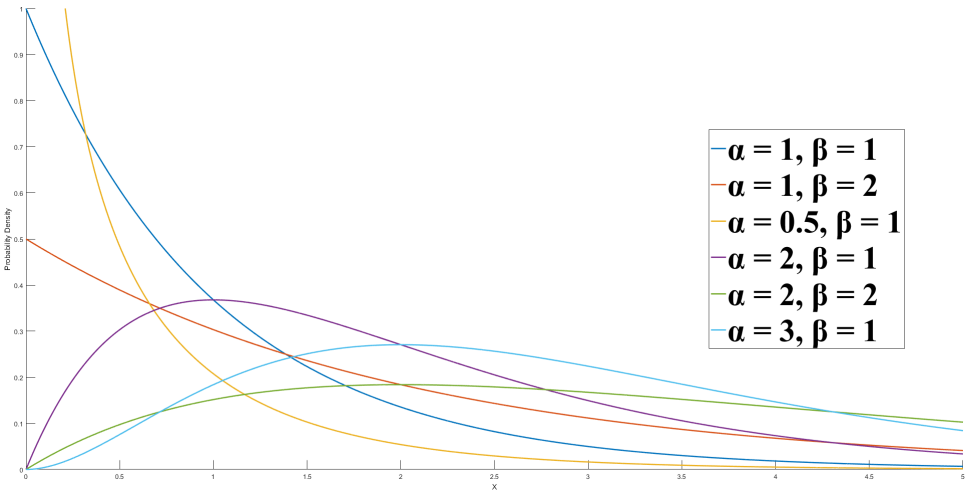

FIGURE B.3: Gamma Distribution - Probability Density Function

We summarise some of the properties of the Gamma distribution with $\alpha, \beta > 0$

$$\text{Mode} = \frac{\alpha - 1}{\beta}, \quad \alpha \geqslant 1,$$

$$E(X) = \frac{\alpha}{\beta},$$

$$\text{Var}(X) = \frac{\alpha}{\beta^2}.$$

B.2.3 Inverse Gamma Distribution

The Inverse Gamma distribution is the continuous probability distribution which acts as the distribution of the reciprocal of a variable distributed according to the Gamma distribution, i.e. $X \sim \text{Gamma}(\alpha, \beta) \Rightarrow X^{-1} \sim \text{Inv-Gamma}(\alpha, \beta)$. Perhaps the chief use of the Inverse Gamma distribution is in Bayesian statistics, where the distribution arises as the conjugate prior for the unknown variance of a Normal distribution. The Inverse Gamma distribution's pdf is defined as

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot x^{-\alpha-1} \cdot \exp\left\{-\frac{\beta}{x}\right\}, \quad x > 0,$$

with shape parameter $\alpha > 0$ and scale parameter $\beta > 0$.

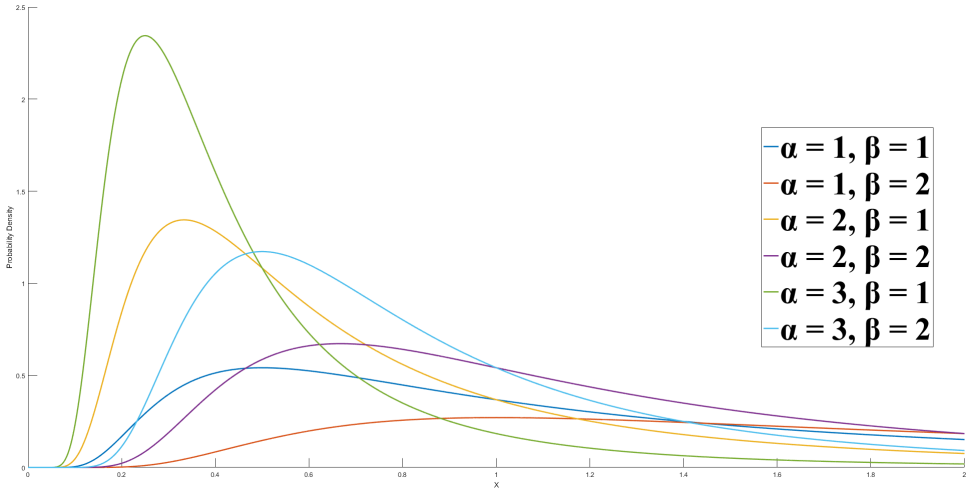

FIGURE B.4: Inverse Gamma Distribution - Probability Density Function

We summarise some of the properties of the Inverse Gamma distribution

$$\text{Mode} = \frac{\beta}{\alpha + 1},$$

$$E(X) = \frac{\beta}{\alpha - 1}, \quad \alpha > 1,$$

$$\mathrm{Var}(X) = \frac{\beta^2}{(\alpha-1)^2 \cdot (\alpha-2)}, \quad \alpha > 2.$$

B.2.4 Log-Normal Distribution

The Log-Normal distribution is the continuous probability distribution of a random variable whose logarithm is Normally distributed, i.e. $X \sim \mathrm{Lognormal}(\mu, \sigma^2) \Leftrightarrow \log X \sim \mathcal{N}(\mu, \sigma^2)$. We have

$$f(x; \mu, \sigma^2) = \frac{1}{x \cdot \sqrt{2\pi\sigma^2}} \cdot \exp\left\{-\frac{(\log x - \mu)^2}{2\sigma^2}\right\}, \quad x > 0.$$

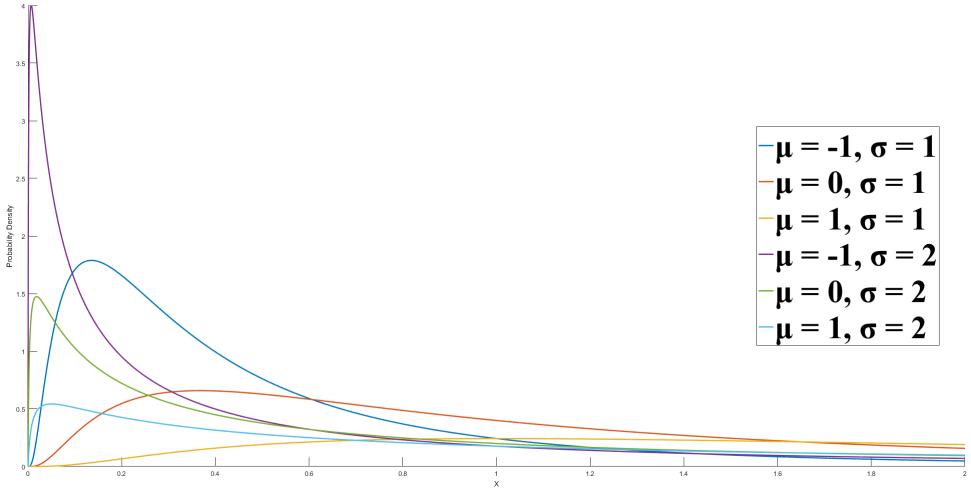

FIGURE B.5: Log-Normal Distribution - Probability Density Function

We summarise some of the properties of the Log-Normal distribution

$$\mathrm{Median} = e^{\mu}, \quad \mathrm{Mode} = \exp\left\{\mu - \sigma^2\right\},$$

$$E(X) = \exp\left\{\mu + \frac{\sigma^2}{2}\right\}, \quad \mathrm{Var}(X) = \left[\exp\left\{\sigma^2\right\} - 1\right] \cdot \exp\left\{2\mu + \sigma^2\right\}.$$

B.2.5 Normal Distribution

The Normal (Gaussian) distribution is a very common continuous probability distribution. The pdf of the Normal distribution is

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}, \quad x \in \mathbb{R}.$$

It is uni-modal and symmetric around the point $x = \mu$, which is at the same time the mode, the median and the mean of the distribution.

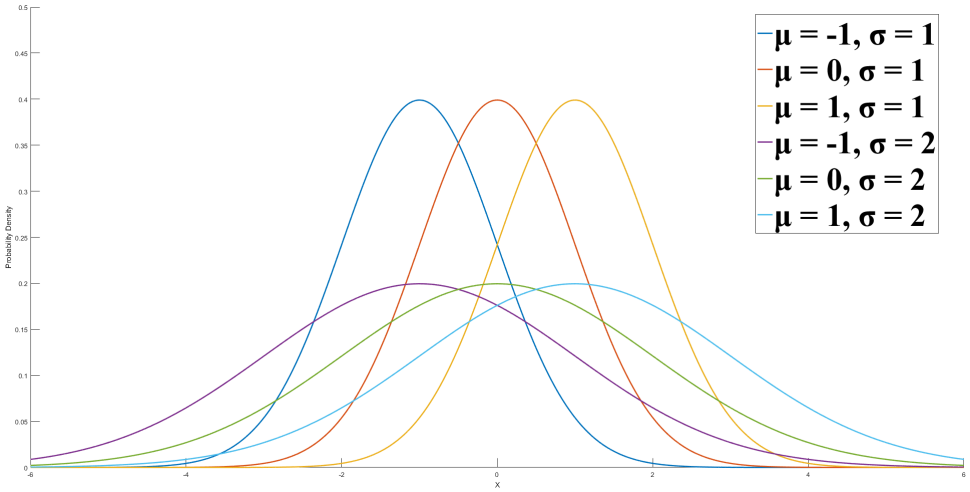

FIGURE B.6: Normal Distribution - Probability Density Function

B.3 Joint Distributions

B.3.1 Dirichlet Distribution

The Dirichlet distribution is a family of continuous multivariate probability distributions parametrised by a vector $\boldsymbol{\alpha} = (a_1, \ldots, a_r)$ of positive real numbers. It is a multivariate generalisation of the Beta distribution. In Bayesian statistics, it is the conjugate prior of the Categorical distribution and Multinomial distributions. The Dirichlet distribution of order $r \geqslant 2$ has a pdf given by

$$f(\mathbf{x}; \boldsymbol{\alpha}) = \frac{1}{B_r(\boldsymbol{\alpha})} \cdot \prod_{i=1}^{r} x_i^{a_i - 1}, \quad \sum_{i=1}^{r} x_i = 1, \quad x_i \geqslant 0, \quad i = 1, 2, \ldots, r,$$

where $\mathbf{x} = (x_1, \ldots, x_r)$ and $B_r(\boldsymbol{\alpha})$ is the multivariate Beta function.

The marginal distributions of $\mathbf{X} = (X_1, X_2, \ldots, X_r)$ are Beta distributions. To be precise, $X_i \sim \text{Beta}(a_i, a_0 - a_i)$ for $i = 1, 2, \ldots, r$, where $a_0 = \sum_{i=1}^{r} a_i$.

The mode of the distribution is the vector

$$\left(\frac{a_1 - 1}{a_0 - r}, \frac{a_2 - 1}{a_0 - r}, \ldots, \frac{a_r - 1}{a_0 - r} \right).$$

B.3.2 Mutlinomial Distribution

The Multinomial distribution is a generalisation of the Binomial distribution. For n independent trials, each of which leads to a success for exactly one of r categories, with each category having a given fixed success probability, the Multinomial distribution gives the probability of any particular combination of numbers of successes for the various categories.

When $r = 2$ and $n = 1$, the Multinomial distribution is the Bernoulli distribution. When $r = 2$ and $n > 1$, it is the Binomial distribution. When $r > 2$ and $n = 1$, it is the Categorical distribution.

Denote the variable which is the number of successes of category i as X_i and denote as p_i the probability of success for category i. The probability mass function of this Multinomial distribution is

$$P(\mathbf{X} = \mathbf{x}; n, \mathbf{p}) = \frac{n!}{x_1! \cdots x_r!} \cdot p_1^{x_1} \cdots p_r^{x_r}, \quad \sum_{i=1}^{r} x_i = n,$$

where $\mathbf{x} = (x_1, \ldots, x_r)$ and $\mathbf{p} = (p_1, \ldots, p_r)$. Each of the r components has a Binomial distribution with parameters n and p_i, i.e. $X_i \sim \text{Bin}(n, p_i)$ for $i = 1, 2, \ldots, r$.

B.3.3 Multivariate Normal Distribution

The Multivariate Normal distribution is a generalisation of the one-dimensional Normal distribution. A random vector is said to have a d-dimensional Normal distribution if every linear combination of its d components has a univariate Normal distribution.

The Multivariate Normal distribution is said to be "non-degenerate" when the symmetric covariance matrix $\boldsymbol{\Sigma}$ is positive definite. In this case, it has pdf

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = |\mathbf{2\pi\Sigma}|^{-\frac{1}{2}} \cdot \exp\left\{-\frac{(\mathbf{x} - \boldsymbol{\mu})^T \cdot \boldsymbol{\Sigma}^{-1} \cdot (\mathbf{x} - \boldsymbol{\mu})}{2}\right\}, \quad \mathbf{x} \in \mathbb{R}^d,$$

where $\mathbf{x} = (x_1, \ldots, x_d)$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)$ is the d-dimensional mean vector.

B.3.4 Wishart Distribution

The Wishart distribution is a generalisation of the Gamma distribution. It is a probability distribution defined over symmetric, non-negative definite, matrix-valued random variables. In Bayesian statistics, the Wishart distribution is the conjugate prior of the inverse covariance matrix (precision matrix) of a Multivariate Normal random vector.

Let $\mathbf{X}$ be a $d \times d$ symmetric matrix of random variables that is positive definite. Let $\mathbf{V}$ be a fixed positive definite matrix of size $d \times d$. Given that $n \geqslant d$, $\mathbf{X}$ has a d-dimensional Wishart distribution with n degrees of freedom if it has a pdf given by

$$f(\mathbf{X}; \mathbf{V}, n) = \frac{1}{\Gamma_d\left(\frac{n}{2}\right)} \cdot |2\mathbf{V}|^{-\frac{n}{2}} \cdot |\mathbf{X}|^{\frac{n-d-1}{2}} \cdot \exp\left\{-\frac{1}{2}\text{tr}\left[\mathbf{V}^{-1}\mathbf{X}\right]\right\},$$

where Γ_d is the d-dimensional Gamma function, defined as

$$\Gamma_d(a) = \pi^{\frac{d(d-1)}{4}} \prod_{i=1}^{d} \Gamma\left(a + \frac{1-i}{2}\right).$$

We summarise some of the properties of the Wishart distribution

$$\text{Mode} = (n-d-1)\cdot\mathbf{V}, \quad n \geqslant d+1, \quad E(\mathbf{X}) = n\cdot\mathbf{V}.$$

B.3.5 Inverse Wishart Distribution

The Inverse Wishart distribution is a probability distribution defined on real-valued, positive-definite matrices. In Bayesian statistics, it is used as the conjugate prior for the covariance matrix of a Multivariate Normal distribution.

We say $\mathbf{X}$ follows an Inverse Wishart distribution, denoted as $\mathbf{X} \sim \mathcal{W}_d^{-1}(\mathbf{V}, n)$, if its inverse, $\mathbf{X}^{-1}$, has a Wishart distribution, i.e. $\mathbf{X}^{-1} \sim \mathcal{W}_d(\mathbf{V}^{-1}, n)$. The pdf of the Inverse Wishart distribution is

$$f(\mathbf{X}; \mathbf{V}, n) = \frac{1}{\Gamma_d\left(\frac{n}{2}\right)} \cdot \left|2\mathbf{V}^{-1}\right|^{-\frac{n}{2}} \cdot |\mathbf{X}|^{-\frac{n+d+1}{2}} \cdot \exp\left\{-\frac{1}{2}\text{tr}\left[\mathbf{V}\mathbf{X}^{-1}\right]\right\}.$$

We summarise some of the properties of the Inverse Wishart distribution

$$\text{Mode} = \frac{1}{n+d+1}\cdot\mathbf{V}, \quad E(\mathbf{X}) = \frac{1}{n-d-1}\cdot\mathbf{V}, \quad n > d+1.$$

Bibliography

[1] José M. Bernardo and Adrian F. M. Smith. *Bayesian Theory*. John Wiley & Sons, 2000.

[2] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer Science & Business Media, 2005.

[3] Olivier Cappé, Christian P. Robert, and Tobias Rydén. "Reversible Jump, Birth-and-Death and More General Continuous Time Markov Chain Monte Carlo Samplers". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.3 (July 2003), pp. 679–700.

[4] Bradley P. Carlin and Thomas A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis. Third Edition*. CRC Press, 2008.

[5] Jiahua Chen. "On Finite Mixture Models". In: *Statistical Theory and Related Fields* 1.1 (May 2017), pp. 15–27. URL: https://www.tandfonline.com/doi/full/10.1080/24754269.2017.1321883.

[6] Paul S. P. Cowpertwait and Andrew V. Metcalfe. *Introductory Time Series with R*. Springer Science & Business Media, 2009.

[7] Arnaud Doucet, Simon J. Godsill, and Christian P. Robert. "Marginal Maximum a Posteriori Estimation Using Markov Chain Monte Carlo". In: *Statistics and Computing* 12.1 (Jan. 2002), pp. 77–84. URL: http://www.cs.ubc.ca/~arnaud/doucet_godsill_robert_marginalmapusingmcmc.pdf.

[8] Bradley Efron. *The Jackknife, the Bootstrap and Other Resampling Plans. CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, 1987.

[9] Robert J. Elliott, Lakhdar Aggoun, and John B. Moore. *Hidden Markov Models. Estimation and Control*. Springer Science & Business Media, 2008.

[10] Sylvia Frühwirth-Schnatter. "Estimating Marginal Likelihoods for Mixture and Markov Switching Models Using Bridge Sampling Techniques". In: *The Econometrics Journal* 7.1 (June 2004), pp. 143–167.

[11] Sylvia Früwirth-Schnatter. *Finite Mixture and Markov Switching Models*. Springer Science & Business Media, 2006.

[12] Sylvia Früwirth-Schnatter. *Finite Mixture and Markov Switching Models. Implementation in MATLAB Using the Package bayesf Version 2.0*. Springer Science & Business Media, 2008.

[13] Andrew Gelman and Xiao-Li Meng. "Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling". In: *Statistical Science* 13.2 (May 1998), pp. 163–185. URL: https://projecteuclid.org/euclid.ss/1028905934.

[14] Phillip I. Good. *Permutation, Parametric and Bootstrap Tests of Hypotheses. Third Edition*. Springer Science & Business Media, 2004.

[15] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and Random Processes. Third Edition*. Oxford University Press, 2001.

[16] Maya R. Gupta and Yihua Chen. *Theory and Use of the EM Algorithm. Foundations and Trends in Signal Processing*. Now Publishers, 2011.

[17] A. Jasra, C. C. Holmes, and D. A. Stephens. "Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling". In: *Statistical Science* 20.1 (Feb. 2005), pp. 50–67. URL: https://projecteuclid.org/euclid.ss/1118065042.

[18] Jae Kwang Kim and Jun Shao. *Statistical Methods for Handling Incomplete Data*. CRC Press, 2013.

[19] Vidyadhar G. Kulkarni. *Modeling and Analysis of Stochastic Systems. Second Edition*. CRC Press, 2009.

[20] Nhu D. Le et al. "Exact Likelihood Evaluation in a Markov Mixture Model for Time Series of Seizure Counts". In: *Biometrics* 48.1 (Mar. 1992), pp. 317–323.

[21] Jüri Lember and Alexey Koloydenko. "The Adjusted Viterbi Training for Hidden Markov Models". In: *Bernoulli Journal* 14.1 (Feb. 2008), pp. 180–206. URL: https://projecteuclid.org/download/pdfview_1/euclid.bj/1202492790.

[22] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, 2008.

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

[24] Loukia Meligkotsidou. *Bayesian Inference*. 2017.

[25] Loukia Meligkotsidou. *Bayesian Inference II*. 2014.

[26] Anthony O'Hagan and Jonathan Forster. *Kendall's Advanced Theory of Statistics. Volume 2B: Bayesian Inference, Second Edition*. Hodder Arnold, 2004.

[27] Yudi Pawitan. *In All Likelihood. Statistical Modelling and Inference Using Likelihood*. Oxford University Press, 2013.

[28] L. R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2 (Feb. 1989), pp. 257–286. URL: https://www.ece.ucsb.edu/Faculty/Rabiner/ece259/.

[29] Sylvia Richardson and Peter J. Green. "On Bayesian Analysis of Mixtures with an Unknown Number of Components". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59.4 (1997), pp. 731–792.

[30] Christian P. Robert. *The Bayesian Choice. From Decision-Theoretic Foundations to Computational Implementation, Second Edition*. Springer Science & Business Media, 2007.

[31] Christian P. Robert, Tobias Rydén, and D. M. Titterington. "Bayesian Inference in Hidden Markov Models Through the Reversible Jump Markov Chain Monte Carlo Method". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 62.1 (Jan. 2002), pp. 57–75.

[32] Christian P. Robert, Tobias Rydén, and D. M. Titterington. "Convergence Controls for MCMC Algorithms with Applications to Hidden Markov Chains". In: *Journal of Statistical Computation and Simulation* 64.4 (Sept. 1998), pp. 327–355.

[33] Sheldon M. Ross. *Simulation. Fourth Edition*. Elsevier Academic Press, 2006.

[34] Sheldon M. Ross. *Stochastic Processes. Second Edition*. John Wiley & Sons, 1996.

[35] Tobias Rydén. "EM Versus Markov Chain Monte Carlo for Estimation of Hidden Markov Models: A Computational Perspective". In: *Bayesian Analysis* 3.4 (Dec. 2008), pp. 659–688. URL: https://projecteuclid.org/euclid.ba/1340370402.

[36] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications. With R Examples, Third Edition*. Springer Science & Business Media, 2011.

[37] Philipp Singer et al. "Detecting Memory and Structure in Human Navigation Patterns Using Markov Chain Models of Varying Order". In: *Plos One* 9.7 (July 2014). URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102070.

[38] Matthew Stephens. "Bayesian Analysis of Mixture Models with an Unknown Number of Components - An Alternative to Reversible Jump Methods". In: *The Annals of Statistics* 28.1 (Feb. 2000), pp. 40–74. URL: https://projecteuclid.org/euclid.aos/1016120364.

[39] C. F. Jeff Wu. "On the Convergence Properties of the EM Algorithm". In: *The Annals of Statistics* 11.1 (Mar. 1983), pp. 95–103. URL: https://projecteuclid.org/euclid.aos/1176346060.

[40] Walter Zucchini and Iain L. MacDonald. *Hidden Markov Models for Time Series. An Introduction Using R*. CRC Press, 2009.