2017

Model Fit Diagnostics for Hidden Markov Models

Kadhem, Safaa K.

http://hdl.handle.net/10026.1/9966

http://dx.doi.org/10.24382/1218
University of Plymouth

Model Fit Diagnostics for Hidden Markov Models

by

Safaa K. Kadhem

A thesis submitted to Plymouth University in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Electronics and Mathematics
Faculty of Science and Engineering

Plymouth University

January 2017

Model Fit Diagnostics for Hidden Markov Models

Safaa K. Kadhem

Abstract

Hidden Markov models (HMMs) are an efficient tool to describe and model the underlying behaviour of many phenomena. HMMs assume that the observed data are generated independently from a parametric distribution, conditional on an unobserved process that satisfies the Markov property. The model selection or determining the number of hidden states for these models is an important issue which represents the main interest of this thesis. Applying likelihood-based criteria for HMMs is a challenging task as the likelihood function of these models is not available in a closed form. Using the data augmentation approach, we derive two forms of the likelihood function of a HMM in closed form, namely the observed and the conditional likelihoods. Subsequently, we develop several modified versions of the Akaike information criterion (AIC) and Bayesian information criterion (BIC) approximated under the Bayesian principle. We also develop several versions for the deviance information criterion (DIC). These proposed versions are based on the type of likelihood, i.e. conditional or observed likelihood, and also on whether the hidden states are dealt with as missing data or additional parameters in the model. This latter point is referred to as the concept of focus. Finally, we consider model selection from a predictive viewpoint. To this end, we develop the so-called widely applicable information criterion (WAIC). We assess the performance of these various proposed criteria via simulation studies and real-data applications.

In this thesis, we apply Poisson HMMs to model the spatial dependence analysis in count data via an application to traffic safety crashes for three highways in the UK. The ultimate interest is in identifying highway segments which have distinctly higher crash rates. Selecting an optimal number of states is an important part of the interpretation. For this purpose, we employ model selection criteria to determine the optimal number of states. We also use several goodness-of-fit checks to assess the model fitted to the data. We implement an MCMC algorithm and check its convergence. We examine the sensitivity of the results to the prior specification, a potential problem given small sample sizes. The Poisson HMMs adopted can provide a different model for analysing spatial dependence on networks. It is possible to identify segments with a higher posterior probability of classification in a high risk state, a task that could prioritise management action.

Contents

List of Figures

List of Tables

Acknowledgements

Completing this work would not have been easy were it not for the support and encouragement that was provided by my Director of Studies, Dr Paul Hewson. So, I must thank him and as I am truly indebted to him for his help. Also, I would like to thank Dr Irene Kaimi, as a second supervisor, for her continuous help during my studies. I am so thankful for her reviewing several drafts of this thesis and for her helpful comments. Also, I have to thank my new supervisor, Professor David McMullan, for helping me at the last stages of submission of my thesis and arranging the final examination.

I would like to thank the Iraqi Ministry of Higher Education and Scientific Research and Al Muthanna University for financing the scholarship that has enable me to complete this thesis.

Personally, I would also like to thank all my friends, PhD students, in Plymouth and all members of the School of Mathematics and Statistics. Also, I would like to thank all members of Plymouth University.

Finally, I am so thankful for my parents and family who have supported me since my first days of starting my thesis and until this moment.

To my dear friend, Dr. Muhammed Al-Mallah, who passed away before this work was finished. I will never forget you. You will always remain in my memory.

Author's declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award. Work submitted for this research degree at Plymouth University has not formed part of any other degree either at Plymouth University or at another establishment.

Relevant scientific seminars and conferences were regularly attended at which work was often presented. One paper has been accepted for publication in refereed journal.

Publications:

Kadhem, S. K., Hewson, P. & Kaimi, I. (2016). Recursive Deviance Information Criterion for the Hidden Markov Model. *International Journal of Statistics and Probability, 5(1)*, 61-78. DOI: http://dx.doi.org/10.5539/ijsp.v5n1p61

Kadhem, S. K., Hewson, P. & Kaimi, I. (2016). Using hidden Markov models to model spatial dependence in a network. (*Under review: Spatial Statistics*)

Book reviews

Kadhem, S. K. (2017). Handbook of discrete-valued time series by Richard A. Davis, Scott H. Holan, Robert B. Lund, and Nalini Ravishanker, *Journal of the Royal Statistical Society A, Volume 180, Issue 2, February 2017, Pages 682-683.*

Posters and conference presentations:

Kadhem, S. K. Bayesian Criteria for Model Choice for Hidden Markov Models. *International Conference of the Royal Statistical Society*. Exeter University, UK, 7-10 Sept. 2015

Conferences and Courses attended:

- Academy for PhD training in statistics

- Statistical Computing and Statistical Inference, University of Cambridge, Cambridge, UK, 2013.
- Applied Stochastic Process and Computer Intensive Statistics, Leeds University, Leeds, UK, 2014.

Word count for the main body of this thesis: **55456**

Signed: ______________________________

Date: ______________________________

Chapter 1

Introduction

This chapter includes a brief overview of relevant Bayesian theory and the aims and structure of the thesis. In Section 1.1, we present the general concepts of Bayesian inference and some related topics such as the prior and posterior specification as well as the use of the sampling MCMC algorithms for simulation based inference. In Section 1.2 we summarize the aims and the outline of this thesis.

1.1 Bayesian inference

This section reviews the basic principle of the Bayesian approach and also Bayes law. It also considers computational methods which make a Bayesian approach possible. We consider the Markov Chain Monte Carlo (MCMC) approach, the most widely used procedure in Bayesian sampling. In addition, the section considers how to diagnose convergence of an MCMC sampler.

1.1.1 Basics of Bayesian analysis

In general, statistical inference is the process of drawing conclusions about populations or scientific truths from data, $\mathbf{y}$. To conduct statistical inference, we specify a statistical model, characterized by model parameter(s), θ, that explains the data according to a probability distribution. For example, for a single data point, y_t, we may assume

$$\mathrm{y}_t \sim Pr(\mathrm{y}_t|\theta), \tag{1.1}$$

where $Pr(\mathrm{y}_t|\theta) = L(\theta;\mathrm{y}_t)$ is a function of an unknown parameter(s) θ, which is also called the "likelihood" function. Given a sequence of observations, denoted by the vector $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$, the likelihood function of the entire sequence of observations can be defined as:

$$L(\theta;\mathbf{y}) = Pr(\mathbf{y}|\theta) = \prod_{t=1}^{T} Pr(\mathrm{y}_t|\theta). \tag{1.2}$$

For example, if we assume that observation y_t has been generated from a Normal distribution with parameters $\theta = (\mu, \sigma^2)$ as

$$Pr(\mathrm{y}_t|\theta) = \phi(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\mathrm{y}_t - \mu)^2\right\}, \tag{1.3}$$

the likelihood of the entire independent and identically distributed (*iid*) sample, $Pr(\mathbf{y}|\theta) = Pr(\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T|\theta)$, is

$$L(\theta;\mathbf{y}) = \prod_{t=1}^{T} Pr(\mathrm{y}_t|\theta) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^T \exp\left\{-\frac{1}{2\sigma^2}\sum_{t=1}^{T}(\mathrm{y}_t - \mu)^2\right\}. \tag{1.4}$$

Note that the concept of likelihood function in the Bayesian approach has a different meaning from that in the frequentist approach. In Bayesian approach, the likelihood, $Pr(\mathbf{y}|\theta)$, is viewed as a conditional probability function that varies with data $\mathbf{y}$ at fixed values of θ, whereas in the frequentist approach the likelihood, $L(\theta;\mathbf{y})$, is a mathematical function of the parameter θ for fixed data, $\mathbf{y}$ (Kroese and Chan, 2014, p.228).

Suppose we are interested in making inferences about an unknown quantity, θ. This can be performed using either the frequentist or the Bayesian approach. In a Maximum Likelihood (ML) approach, this is achieved by finding a value $\theta = \theta^*$ that maximizes the likelihood of $\mathbf{y}$ with respect to θ, as follows:

$$\frac{\partial L(\theta;\mathbf{y})}{\partial\theta} = 0,$$

which satisfies

$$\frac{\partial^2 L(\theta;\mathbf{y})}{\partial^2\theta}|_{\theta=\theta^*} < 0. \tag{1.5}$$

Alternatively, the inference about θ can be implemented using the Bayesian approach, when θ is treated as a random variable represented by a probability distribution. This involves the specification of a prior distribution:

$$\theta \sim Pr(\theta).$$

According to the Bayes' theorem, by combining information from the prior distribution and information about the observed data from the likelihood, we can obtain

$$Pr(\theta|\mathbf{y}) = \frac{Pr(\mathbf{y}|\theta)Pr(\theta)}{Pr(\mathbf{y})} \propto Pr(\mathbf{y}|\theta)Pr(\theta), \tag{1.6}$$

where $Pr(\theta|\mathbf{y})$, which represents the probability statement about the unknown parameters given the data, is known as the *posterior* distribution and forms the core of Bayesian inference

(Gelman et al., 2014, p.7). Note that the term $Pr(\mathbf{y}) = \sum_\theta Pr(\theta)Pr(\mathbf{y}|\theta)$ for discrete or $Pr(\mathbf{y}) = \int_\theta Pr(\theta)Pr(\mathbf{y}|\theta)d\theta$ in the case of continuous θ, can be viewed as the marginal probability of observing the data. It behaves as a normalizing constant which ensures that the posterior distribution is a probability distribution and thus integrates of 1 over all values of θ. The normalizing constant, $Pr(\mathbf{y})$, in Equation (1.6) is generally not of interest as it does not depend on the parameter θ, and thus it can be ignored during parameter estimation.

Many efficient procedures have been proposed for approximating the posterior in Equation (1.6). One of which is known as the Markov chain Monte Carlo (MCMC) method (Geyer, 2011).

1.1.2 The prior distributions

As discussed in Section (1.1.1), the posterior distribution is based on two main components, the likelihood model and the prior distribution. However, the functional form of the prior distribution is often unknown. In this case, the choice of prior is commonly based on assumptions (Carlin and Louis, 2009; Gelman et al., 2014). Such assumptions are often based on various factors such as physical considerations, degree of knowledge and, more controversially, mathematical convenience. The choice of prior distribution is an essential part in Bayesian analysis in view of its ability in simplifying the posterior manipulations.

Prior distributions can be classified as *informative* and *non-informative* priors. Informative priors are used when prior knowledge is available. On the other hand, the non-informative priors are used when no *a priori* knowledge is available. Such priors have a minimal effect on inference as they provide little prior information for the unknown parameters of the model. Hence, the data will be mostly responsible for the posterior distribution, or as described by (Gelman et al., 2014) "to let the data speak for themselves", so that inference is not affected by external information. Examples of priors intended to be non-informative are flat priors (e.g. that a parameter is uniformly distributed between $-\infty$ and $+\infty$, or between 0 and $+\infty$), reference priors (Berger and Bernardo, 1989) and Jeffreys's prior (Jeffreys, 1961) which is expressed as

$$Pr(\theta) \propto |I(\theta)|^{\frac{1}{2}}, \quad \text{where } I(\theta) = -\mathrm{E}\left\{\frac{-\partial^2 \log f(\mathbf{y}|\theta)}{\partial\theta'\partial\theta}\right\}.$$

The term $I(\theta)$ is called Fisher's information matrix and the expectation is taken with respect to the sampling distribution of $\mathbf{y}$. The Jeffreys' prior gives an automated method for finding a non-informative prior for any parametric model. Also, it is known that the Jeffreys's prior is invariant to transformation.

It is said that the prior is a *conjugate* prior if the posterior distribution follows the same distribution family as the chosen prior. A common class of distributions that all their members have conjugate priors is the exponential family. The exponential family of distributions can be written as:

$$Pr(\mathrm{y}_t|\theta) = f(\mathrm{y}_t)g(\theta)\exp\left\{\phi(\theta)' u(\mathrm{y}_t)\right\}. \tag{1.7}$$

In general, the factors $\phi(\theta)$ and $u(\mathrm{y}_t)$ are vectors of the same dimension as θ. The factor $\phi(\theta)$ is called the natural parameter of the exponential family. The likelihood of the whole sequence of *iid* variables, a function of θ, can be then written as

$$Pr(\mathbf{y}|\theta) = \left(\prod_{t=1}^{n} f(\mathrm{y}_t)\right) g(\theta)^T \exp\left\{\phi(\theta)' \sum_{t=1}^{T} u(\mathrm{y}_t)\right\}, \tag{1.8}$$

which has a fixed form

$$Pr(\mathbf{y}|\theta) \propto g(\theta)^T \exp\left\{\phi(\theta)' h(\mathbf{y})\right\}, \tag{1.9}$$

where $h(\mathbf{y}) = \sum_{t=1}^{n} u(\mathrm{y}_t)$ denotes a sufficient statistic for θ, because the likelihood for θ depends on the data $\mathbf{y}$ only through the value of $h(\mathbf{y})$. If the prior density is specified as:

$$Pr(\theta) \propto g(\theta)^\eta \exp\left\{\phi(\theta)' v\right\}, \tag{1.10}$$

then the posterior distribution is

$$Pr(\theta|\mathbf{y}) \propto g(\theta)^{\eta+T} \exp\left\{\phi(\theta)'(v + h(\mathbf{y}))\right\}, \tag{1.11}$$

which shows that this choice of prior density is conjugate. For the Normal distribution, where variance parameter σ^2 is known while the mean parameter μ is unknown, the conjugate prior of the unknown mean parameter can take the form of a Normal distribution as

$$\mu \sim N(\mu_0, \sigma_0^2), \tag{1.12}$$

which leads to the posterior distribution

$$\begin{aligned} Pr(\mu|\mathbf{y},\sigma^2,\mu_0,\sigma_0^2) &\propto \exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma_0^2}+\frac{T}{\sigma^2}\right)\left(\mu-\frac{\sigma_0^2 T\bar{\mathbf{y}}+\mu\sigma^2}{\sigma_0^2\sigma^2}\right)^2\right\}, \\ &\sim N\left(\frac{\sigma_0^2 T\bar{\mathbf{y}}+\mu\sigma^2}{\sigma_0^2\sigma^2},\frac{\sigma_0^2\sigma^2}{\sigma_0^2+\sigma^2}\right), \end{aligned} \tag{1.13}$$

where T denotes the sample size. In addition to the Normal distribution, there are many widely used distributions that belong to the exponential family, for example, the Poisson, Gamma, Beta, Dirichlet, binomial and Multinomial distributions (Gelman et al., 2014).

1.1.3 Markov Chain Monte Carlo method

The main aim of Bayesian inference is to approximate the posterior distribution, $Pr(\theta|\mathbf{y})$, in Equation (1.6), as a function of θ. However, in high–dimensional models, where θ is a multi–dimensional vector, we may often face the problem of obtaining the marginal posterior distribution for a single given parameter such as θ_i (where $1 < \theta_i < k$). In principle, the marginal posterior density of θ_i is the integral of the joint posterior density of all elements of θ except θ_i. In practice, evaluating such integrals is analytically difficult. It is possible to evaluate these integrals numerically using Markov Chain Monte Carlo (MCMC) methods, in which a Markov chain is used to sample from the posterior distribution. The main idea behind the MCMC method is that it provides an approximation to the posterior distribution by generating sequentially sampled values, where the posterior distribution depends on its previous sampled value for each unknown parameter (Gelman et al., 2014). The MCMC approach is based on two key aspects, namely, the Markov chain and Monte Carlo integration. So, to understand more about the MCMC methods, it is useful to have a look at these two concepts.

1.1.3.1 Markov chains

The MCMC method works by creating a Markov chain that represents the posterior distribution of interest. A Markov chain can be defined as a particular type of discrete time Markov process,$\left\{X^{(t)};t\geq 0\right\}$, with state space $S=\left\{s_j;j=1,2,...,K\right\}$, $K\leqslant\infty$. A sequence $X_0,X_1,X_2...$ of random variables is a *Markov chain* if the conditional distribution of X_{t+1} given $X_0,...,X_t$ depends only on X_t (Geyer, 2011). We can write this as

$$Pr(X_{t+1}=s_j|X_t=s_i,X_{t-1}=s_{i_{t-1}},...)=Pr(X_{t+1}=s_j|X_t=s_i), \tag{1.14}$$

for all $s_j, s_i, s_{i_{t-1}}, \ldots \in S$ and $t = 0, 1, 2, \ldots$. In other words, the future and past states are independent, given the current state. This property is called the Markov property. The probability

$$p_{ij} = Pr(X_{t+1} = s_j | X_t = s_i); \quad s_i, s_j, \in S, \tag{1.15}$$

is called the transition probability from state s_i to state s_j. If there are K possible states, then

$$\mathbf{P} = p_{ij}; \quad i, j = 1, 2, \ldots, K, \tag{1.16}$$

represents the transition matrix of dimension $K \times K$ that provides the various probabilities of all possible moves among these states for every i, $\sum_{j=1}^{K} p_{ij} = 1$. Let $\pi_j(t) = Pr(X_t = s_j)$ denote the probability that the chain is in state s_j at time t, and $\boldsymbol{\pi}(t) = \{\pi_1(t), \pi_2(t), \ldots, \pi_K(t)\}$ denote the K-length vector of these state probabilities at time t. The probability that the chain is in state s_j at time (step) $t+1$ is given by

$$\begin{aligned} \pi_j(t+1) &= Pr(X_{t+1} = s_j), \\ &= \sum_i Pr(X_{t+1} = s_j | X_t = s_i) Pr(X_t = s_i), \\ &= \sum_i p_{ij} \pi_i(t), \end{aligned} \tag{1.17}$$

where Equation (1.17) describes the evolution of the chain using a number of successive iterations. Using matrix notation, Equation (1.17) can be written as

$$\boldsymbol{\pi}^{(t+1)} = \boldsymbol{\pi}^{(t)} \mathbf{P}. \tag{1.18}$$

It follows that

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{0} \mathbf{P}^{(t)}. \tag{1.19}$$

The n-step transition probability, $p_{ij}^{(n)}$, is the probability that the chain is in the state s_j given that n steps earlier it was in the state s_i, i.e.,

$$p_{ij}^{(n)} = Pr(X_{t+n} = s_j | X_t = s_i), \tag{1.20}$$

where $p_{ij}^{(n)}$ is just the i,jth element of $\mathbf{P}^{(n)}$. A Markov chain is said to be *irreducible* if there exists a positive integer n_{ij} such that $p_{ij}^{n_{ij}} > 0$, for all $i,j = 1,2,...,K$. That is, one can move from any state to any other state in S possible states in a finite number of steps.

A state in a Markov chain is classified as *absorbing*, *transient*, or *recurrent* to characterize how often the state is visited or the time between visits. Let $f_{ij}^{(n)}$ denote the probability that the chain first visits the state s_j at step n, when it started in the state s_i at step 0, i.e.,

$$f_{ij}^{(n)} = Pr(X_1 \neq s_j, X_2 \neq s_j, ..., X_{n-1} \neq s_j, X_n \neq s_j | X_0 = s_i), \tag{1.21}$$

with $f_{ii}^{(0)} = 1$ and $f_{ij}^{(0)} = 0$ for $j \neq i$. Further, define the sum of probabilities of the first visiting times being $n = 1,2,...$,

$$f_{ij} = \sum_{n=1}^{\infty} f_{ij}^{(n)}, \tag{1.22}$$

which is the probability that the chain visits state s_j in finite time if it starts in state s_i. In particular, f_{ii} is the probability of returning to the starting state s_i in finite time. A state s_j is said to be: transient if $f_{jj} < 1$, recurrent if $f_{jj} = 1$, and absorbing if $p_{jj} = 1$. If state s_j is recurrent, then it is said to be positive recurrent if the mean time between revisits is finite, i.e.,

$$\sum_{n=1}^{\infty} n f_{jj}^{(n)} < \infty. \tag{1.23}$$

Otherwise, it is said to be *null recurrent*. If one state in an irreducible Markov chain is positive recurrent, then all the states are positive recurrent. The period of a state s_j is defined as

$$d_j = \gcd\left\{n \geq 1 | p_{jj}^{(n)} > 0\right\}, \tag{1.24}$$

where d_j denotes the greatest common divisor (gcd) of all integers $n \geq 1$. It can be shown that for an irreducible Markov chain, $d_j = d, \forall j$. If $d > 1$, the chain is said to be *periodic* with period d. If $d = 1$, then the chain is said to be *aperiodic*, which means that the chain is not forced into some cycle of fixed length between certain states. It can be seen that if $\mathbf{P}$ has no eigenvalues equal to 1 the chain is aperiodic. The limiting probability $\lim_{n\to\infty} p_{jj}^{(n)}$ may or may not converge. For a transient or null recurrent state s_j, $\lim_{n\to\infty} p_{jj}^{(n)} = 0$, i.e., the probability of the chain being in state s_j eventually goes to zero. If state s_j is positive recurrent and periodic, then $\lim_{n\to\infty} p_{jj}^{(n)}$ will not converge. If s_j is positive recurrent and aperiodic, then $\lim_{n\to\infty} p_{jj}^{(n)}$ will converge to a steady state probability $\pi_j > 0$. A positive recurrent and aperiodic Markov chain approaches a stationary distribution π, where the vector of probabilities of being in any particular given state is independent of the initial condition $\pi^{(0)}$. The stationary distribution satisfies

$$\pi = \pi\mathbf{P}. \tag{1.25}$$

A sufficient condition for a unique stationary distribution is that the detailed balance or time reversibility condition, namely,

$$\pi_i p_{ij} = \pi_j p_{ji}; \quad \forall i, j, \tag{1.26}$$

is satisfied. Indeed, reversibility of a Markov chain for a desired distribution π implies that the Markov chain has π as its stationary distribution. Given a Markov chain X_1, X_2, ..., the transition probability distribution is said to be reversible regarding an initial distribution if the distribution of pairs (X_t, X_{t+1}) is exchangeable. Reversible Markov chain plays a main role in MCMC methods (Fan and Sisson, 2011). Given a reversible Markov chain X_m with the stationary distribution π, it follows that

$$\frac{1}{M}\sum_{m=1}^{M} h(X_m) \longrightarrow \int h(x)\pi(x)dx, \quad \text{as } M \longrightarrow \infty, \tag{1.27}$$

which links Markov chains to Monte Carlo methods, where $m = 1, 2, ..., M$, is a desired period of iterations. Now, it is easy to calculate the posterior quantities using Equation (1.27) because the stationary distribution π is equal to the posterior density $Pr(\theta|\mathbf{y})$.

1.1.3.2 Monte Carlo integration

In practice, inference usually requires the integration of posterior distribution over the parameter space. However, for complicated models, such integration is difficult or impossible to be achieved analytically. For this reason, Monte Carlo integration is often utilized to approximate such integrals (Rizzo, 2008).

For example, assume one is interested in computing the expectation of some function h of a random variable Y that has probability density function $f_Y(\mathrm{y})$:

$$\mathrm{E}[h(Y)] = \int h(\mathrm{y})df_Y = \int h(\mathrm{y})f_Y(\mathrm{y})d\mathrm{y}. \tag{1.28}$$

Hence, an approximation to this integral can be computed by estimating the sample average of random variables $\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T$ drawn from the distribution of Y as follows

$$\widehat{h(\mathrm{y})} = \frac{1}{T}\sum_{t=1}^{T} h(\mathrm{y}_t). \tag{1.29}$$

It follows that the estimate $\widehat{h(\mathrm{y})}$ is a strongly consistent estimate of $\mathrm{E}\left[h(Y)\right]$, such that $\widehat{h(\mathrm{y})} \longrightarrow \mathrm{E}\left[h(Y)\right]$ as the sample size $T \longrightarrow \infty$. Consequently, it can be said that $\widehat{h(\mathrm{y})}$ converges to $\mathrm{E}\left[h(Y)\right]$ with probability 1 as $T \longrightarrow \infty$. Also, according to the Central Limit Theorem,

$$\frac{\widehat{h(\mathrm{y})} - \mathrm{E}\left[h(Y)\right]}{\sigma/\sqrt{T}} \longrightarrow \phi(0,1) \;\; \text{as } T \longrightarrow \infty, \tag{1.30}$$

where $\sigma/\sqrt{T}$ is the standard error of estimate $\widehat{h(\mathrm{y})}$ and $\sigma^2 = Var(h(Y))$ is variance of sample. The main idea of applying the Monte Carlo approach is obtaining the solution of integrals which involve the posterior distribution $Pr(\theta|\mathbf{y})$ mentioned earlier in Equation (1.6).

1.1.4 MCMC sampling techniques

Markov chain Monte Carlo (MCMC) methods are a framework that involves many techniques introduced by Metropolis et al. (1953) and Hastings (1970) for Monte Carlo integration. In this section, we review the well-known Metropolis-Hastings and Gibbs algorithms.

1.1.4.1 The Metropolis-Hastings algorithms

Metropolis-Hastings algorithms (M-H) are a class of Markov Chain Monte Carlo (MCMC) methods. M-H algorithms include many special algorithms such as: the Metropolis Sampler, the independent sampler, the random walk sampler and the Gibbs sampler. The main idea here is to generate a Markov chain $\{X_t;\, t = 0,1,2,...\}$, such that its stationary distribution is the target distribution. This chain must satisfy the regularity conditions discussed in the previous section, i.e. irreducibility, positive recurrence and aperiodicity. In general, the algorithm must specify how to generate the next state X_{t+1}, given state X_t. The M-H algorithms assume candidate values Y that can be generated from some proposal distribution $g(.|X_t)$. If the candidate point is accepted, the chain moves to state Y at time $t+1$ and $X_{t+1} = Y$; otherwise the chain stays in state X_t and $X_{t+1} = X_t$. The choice of proposal distribution is very flexible, but, the chain selected must meet the regularity conditions. Choosing proposal distributions with the same support set as the target distribution will usually satisfy those the regularity conditions (Rizzo, 2008). A Metropolis-Hastings algorithm associated with target density f and conditional density g

produces a Markov chain, $X^{(t)}$; see algorithm (1). The conditional distribution g is called the *proposal density*, the ratio $a(\mathrm{x},\mathrm{y})$ is called the *Hastings ratio* or *acceptance probability* and the step (2) in the algorithm is called *Metropolis rejection*. This algorithm always accepts values y_t such that the ratio $\dfrac{f(\mathrm{y}_t)}{g(\mathrm{y}_t|\mathrm{x}^{(t)})}$ is increased, compared with the previous value $\dfrac{f(\mathrm{x}^{(t)})}{g(x^{(t)}|\mathrm{y}_t)}$. In this algorithm, the chain convergence largely depends on the proposal density. A proposal density with large jumps to places far from the support of the posterior has low acceptance rate and causes the Markov chain to stand still most of the time. On the other hand, a proposal density with small jumps and high acceptance rate may cause the chain to move slowly and to get stuck in one state. Also, in multi-dimensional cases, the M-H algorithm as described above might require a proposal density for the whole vector, which is extremely difficult when the dimension is high (Casella and Robert, 2004).

```
Algorithm 1 : Metropolis-Hastings algorithm
Given $\mathrm{x}^{(t)}$,
    1. Generate $Y_t \sim g(\mathrm{y},\mathrm{x}^{(t)})$.
    2. Take
       $$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } a(\mathrm{x}^{(t)},Y_t) \\ \mathrm{x}^{(t)} & \text{with probability } 1-a(\mathrm{x}^{(t)},Y_t). \end{cases}$$
       where;
       $$a(\mathrm{x},\mathrm{y}) = min\left\{1, \frac{f(\mathrm{y})}{f(\mathrm{x})}\frac{g(\mathrm{x}|\mathrm{y})}{g(\mathrm{y}|\mathrm{x})}\right\}.$$
    3. Stop when convergence is achieved.
```

1.1.4.2 The Gibbs sampler

Gibbs sampler is a special case of the M-H algorithm. Sampling using the Gibbs sampler was proposed by Geman and Geman (1984). This sampler is often applied when the target distribution is multivariate. It regards that all the univariate conditional densities in a multivariate distribution can be specified and that they are easy to sample from. The chain is generated by successively sampling from the conditional distributions of the target distribution (Rizzo, 2008, p.263).

A Gibbs sampler generates a sample from the distribution of each parameter or variable conditioning on the current values of the other parameters or variables. As a simplified example, given an multivariate probability distribution, $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$, we define the $d-1$

dimension random vectors

$$\boldsymbol{\theta}_{(-j)} = (\theta_1, ..., \theta_{j-1}, \theta_{j+1}, ..., \theta_d), \tag{1.31}$$

and denote the corresponding univariate conditional density of θ_j given θ_{-j} by $f(\theta_j|\theta_{-j})$. The Gibbs sampler generates the chain by sampling from each of the densities $f(\theta_j|\theta_{-j})$. Algorithm (2) summarizes the steps of sampling using Gibbs sampler (Marin and Robert, 2014, p.90).

Algorithm 2 : General Gibbs Algorithm

```
I- Initialization: Start with an arbitrary value $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, ..., \theta_d^{(0)})$.
II- Iteration $t$: Given $(\theta_1^{(t-1)}, ..., \theta_d^{(t-1)})$, generate:
    1- $\theta_1^{(t)}$ according to $f_1(\theta_1|\theta_2^{(t-1)}, ..., \theta_d^{(t-1)})$,
    2- $\theta_2^{(t)}$ according to $f_2(\theta_2|\theta_1^{(t)}, \theta_3^{(t-1)}, ..., \theta_d^{(t-1)})$, $\vdots$
    d- $\theta_d^{(t)}$ according to $f_d(\theta_d|\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_{d-1}^{(t)})$
```

1.1.5 Convergence of MCMC methods

MCMC methods have to be carefully implemented. It is important to check that the algorithm explore the posterior distribution fully and that the simulation converges the posterior distribution (Gelman et al., 2014). Some specific implementations are used to check the algorithm include the deciding when to stop sampling or the required length of sampling, the length of burn-in sample that should be discarded and whether a Markov chain has mixed sufficiently. Indeed, there is no single test or diagnostic tool that correctly checks convergence (Gelman et al., 2014). Consequently, many different techniques have been developed, each with a range of advantages and disadvantages, as tools to check convergence. Some of those techniques are claimed to perform well with the M-H algorithm, others are claimed to perform well with the Gibbs sampler (Cowles and Carlin, 1996). Convergence can be diagnosed based on visual methods, including trace density, trace mean, and autocorrelation plots, or using test statistics such as Gelman and Rubin (1992), Geweke (1992) and Raftery and Lewis (1992). We will consider the following convergence tools.

Gelman-Rubin statistic ($\hat{R}$, (Gelman and Rubin, 1992)):

The Gelman-Rubin statistic is based on running multiple chains with different over-dispersed starting points. These chains are then compared by calculating the within chain variance W and between chain variance B. Given W and B, the estimated marginal posterior variance of θ is computed as

$$\widehat{Var} = (1 - \frac{1}{n})W + (\frac{1}{n}B),$$

where n is the number of samples drawn. The statistic assesses whether W and B are different enough to worry about convergence. We then compute the estimated scale reduction using

$$\hat{R} = \sqrt{\frac{\widehat{Var}}{W}}.$$

Both $\widehat{Var}$ and W are expected to converge to the true marginal posterior variance. Hence, values of $\hat{R}$ close to 1 (and often less than 1.1) are considered as an evidence that the chains have converged (Gelman and Hill, 2007, p.358).

Geweke statistic proposed by Geweke (1992):

This statistic is based on a single chain. After discarding a desired burn-in period, Geweke's statistic depends on comparing the difference between the means of the first 10% and 50% of chain over a Z statistic that follows asymptotically a standard-normal distribution:

$$Z = \frac{\hat{\theta}^A - \hat{\theta}^B}{\sqrt{\frac{S^A_\theta}{T_A} + \frac{S^B_\theta}{T_B}}},$$

where $\hat{\theta}^A$ and $\hat{\theta}^B$ denote the means of first and second sample means, respectively, S^A_θ and S^B_θ denote the sample variance corresponding to the first and second part of chain, respectively, and T_A and T_B represent the sizes of first and second part of chain, respectively. A value Z within the interval $[-2, 2]$ denotes no a notable difference between two samples, and hence, convergence is achieved, and vice versa.

Autocorrelation function (ACF):

Because the samples produced by MCMC methods are moving at random small steps which may be correlated, these methods may not be an efficient to represent independent samples of the target distribution (Cowles and Carlin, 1996). Convergence diagnostics can additionally be monitored graphically using the autocorrelation function (ACF) (Ntzoufras, 2009; Gelman et al., 2014). Given a chain of samples, $\mathrm{x}_m : m = 1, 2, ..., M$, generated from an MCMC method, the ACF describes the correlation between successive elements x_m and x_{m+1} of the chain at a different sampling lags l. The ACF is defined as

$$\begin{aligned}\rho(l) &= \frac{Cov(\mathrm{x}_m, \mathrm{x}_{m+1})}{Var(\mathrm{x}_m)}, \\ &= \frac{\sum_{m=1}^{M-1}(\mathrm{x}_m - \bar{\mathrm{x}})(\mathrm{x}_{m+1} - \bar{\mathrm{x}})}{\sum_{m=1}^{M}(\mathrm{x}_m - \bar{\mathrm{x}})^2}.\end{aligned}$$

If ACF drops to zero at lag 1, it implies that there is no correlation between successive samples. The ACF is not a convergence diagnostic tool, but can be helpful indirectly to assess convergence of the MCMC algorithms. The MCMC algorithms produce Markov chains of correlated consecutive samples. These correlated samples are then used to summarize many features, e.g. means, variances and percentiles, which assume to be approximated features to the target distribution. However, these approximations are often less accurate than if they were produced from independent samples (Ntzoufras, 2009; Gelman et al., 2014). In practice, this autocorrelation between samples is often reduced using the so-called *thinning* technique. The thinning method is based on keeping only every m^{th} samples after a pre-specified burn-in period is discarded from the posterior distribution. Thus, inference will be adopted mainly on those thinned chains (Ntzoufras, 2009; Gelman et al., 2014).

1.2 The aims and outlines of the thesis

The main aim of this thesis is to develop Bayesian diagnostic tools for the model selection issue in a Hidden Markov model context. Under the Bayesian perspective, we develop likelihood-based criteria from the AIC, BIC and DIC for HMMs. We extend the original definition of the DIC taking into account the concept of focus and the availability of closed form of the likelihood of HMMs. We also contribute in developing Bayesian modified versions of the AIC and BIC which approximated at the posterior distribution of the model parameters. We also examine the WAIC (Watanabe, 2010), based on the predictive pointwise density.

We also develop a Poisson hidden Markov model (PHMM) to spatially model the traffic crash data. Our methodology is illustrated by application involving the crashes which occurred on several motorways in the UK. We are interested in identifying highway segments which have distinct crash rates (distinct states) of the relative safety process. Selecting an optimal number of states is an important part of the interpretation.

The structure of this thesis is organized as follows:

In Chapter 2 we present briefly the concept of mixture models to give a better understanding of hidden Markov models.

In Chapter 3 we include the fundamental definitions and notations of the HMMs. In addition, we introduce the idea of presenting the HMM as a generative model. We develop an algorithm to explain the mechanism of generating data from a parametric HMM. Furthermore, this chapter presents the concept of forward-backward algorithm, as well as the estimation of the model parameters using the EM approach.

Chapter 4 discusses the inference technique for the unknown parameters of hidden Markov models within the Bayesian framework. We set out a theory of hidden state models and develop the necessary MCMC algorithm. We discuss the problem of estimation of the hidden state sequence of a HMM. In addition, this chapter discusses the problem of label switching. We review the literature relevant to this problem and also its solutions.

In chapter 5 we consider the model selection issue of HMMs. We derive several forms of the likelihood function of a HMM, namely, the observed, complete and conditional likelihood. We develop several conditional and observed likelihood-based versions for the Deviance information criterion (DIC; Spiegelhalter et al., 2002). In addition, we propose several modified versions of the Akaike information criterion (AIC; Akaike, 1973) and the Bayesian information criterion (BIC; Schwarz, 1978) approximated from a Bayesian perspective. Also, this chapter introduces a criterion based on assessing the predictive ability of a HMM, the widely applicable information criterion (WAIC; Watanabe, 2009).

In chapter 6, we introduce simulation studies based on synthetic and real data application to assess the model selection criteria proposed in chapter 5.

in Chapter 7 we presents an application involving the traffic crash data. In this chapter we model the spatial dependency, rather than the temporal dependency, on a highway segment using a Poisson hidden Markov model (PHMM). We apply our methodology to identify the highway segments that have distinct crash rates (distinct states) of the relative safety process. This chapter also includes the process of estimation and model selection taking into account the sensitivity analysis of some priors chosen for the state-specific crash rates.

Finally, chapter 8 dedicated to summarize the work of this thesis and introduce some proposed ideas for future research of HMMs.

Chapter 2

Finite Mixture Models

2.1 Introduction

Hidden Markov models can be considered as an extension of mixture models where the observations are generated independently from some distribution depending on a state or component follows an unobserved Markov process (Cappé et al., 2005). In order to understand the theoretical structure of hidden Markov models, we devote this chapter for reviewing briefly some fundamentals of mixture models. This thesis mainly concentrates on HMMs with a discrete and finite state space.

Mixture models have been developed as a flexible tool to model data with an unobserved heterogeneity, for example, different types of data can form clusters or groups. A finite mixture model (FMM) is generally used when an observation belongs to one of K groups (components) that have distinct features and can be described by different probability distributions (Marin and Robert, 2014). In other words, these models are a weighted average of a finite number of distributions (mixing components). In real life, FMMs may be a finite mixture of distributions such as Gaussian or Poisson distributions (McLachlan and Peel, 2000; Frühwirth-Schnatter, 2006).

Interest in FMMs has increased over the last decades. They can be used for cluster analysis, latent class analysis, discriminant analysis, image analysis, survival analysis, disease mapping and meta analysis. There are many textbooks which have focused in detail on finite mixture models such as McLachlan and Peel (2000); Frühwirth-Schnatter (2006); Schlattmann (2009); Marin and Robert (2014).

Bayesian methods to model these mixtures of distributions have been used widely for inference. The extensive use of these distributions led to the rapid development in posterior simulation techniques such as MCMC methods (McLachlan and Peel, 2000, p.5). Therefore, MCMC procedures have been used to handle the difficulties in the estimation processes of parameters of FMM such as the number of k components (Richardson and Green, 1997), and the effect of label switching (Stephens, 2000; Jasra et al., 2005). Moreover, the Bayesian framework has been employed to simplify these complicated structures by classifying them into a set of simple structures using hidden or latent variables (Marin and Robert, 2014).

2.2 Definition of the finite mixture model

One way of conceptualizing the FMM would be to assume that the data arise from a mixture of pre-specified number of sub-populations in different proportions instead of one population. Let $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$ denote a sample of observed data with length T, the probability density function (pdf) of a mixture model can be defined as a combination of K component pdfs

$$Pr(\mathbf{y}|\Theta) = \sum_{k=1}^{K} \pi_k \; Pr_k(\mathbf{y}|\theta_k), \tag{2.1}$$

where $Pr_k(\mathbf{y}|\theta_k)$ denotes the pdf of the k^{th} component, π_k is the weight of the population k such that

$$0 \leq \pi_k \leq 1, \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1,$$

$\Theta = (\pi; \boldsymbol{\theta}) = (\pi_1, \pi_2, ..., \pi_k; \theta_1, \theta_2, ..., \theta_k)$ denotes a set of all unknown weights and parameters of a mixture model. In many applications, a family of distributions having the density in Equation (2.1) can be called a K-component finite mixture model.

The main idea of mixture model is that the observations $\mathbf{y}$ are generated from K distinct random processes, so that each process is modelled by the density $Pr_k(\mathbf{y}|\theta_k)$, and π_k represents the corresponding proportion of observations from this process. For example, consider a FMM where $Pr(\mathbf{y}|\Theta)$ is constituted from densities which are all Normal distributions. For simplification, assume a mixture of two univariate Normal components with common variance σ^2 and means μ_1 and μ_2 in proportions π_1 and π_2, so that

$$\begin{aligned} \phi(\mathrm{y}_t; \Theta) = \sum_{k=1}^{2} \pi_k \; \phi_k(\mathrm{y}_t; \theta_k) &= \sum_{k=1}^{2} \pi_k \; \phi_k(\mathrm{y}_t; \mu_k, \sigma^2) \\ &= \pi_1 \; \phi_1(\mathrm{y}_t; \mu_1, \sigma^2) + \pi_2 \; \phi_2(\mathrm{y}_t; \mu_2, \sigma^2), \end{aligned} \tag{2.2}$$

where $\Theta = (\theta_k, \pi_k) = (\mu_1, \mu_2, \sigma^2; \pi_1, \pi_2)$ denotes all unknown parameters of a two-component Normal mixture model, and

$$\phi(\mathrm{y}_t; \mu, \sigma^2) = (2\pi)^{-\frac{1}{2}} \sigma^{-1} \exp\left\{ -\frac{1}{2} (\mathrm{y}_t - \mu)^2 / \sigma^2 \right\}, \tag{2.3}$$

denotes a univariate Normal density with mean μ and variance σ^2. If the two Normal densities are sufficiently divergent, then the mixture density $f(\mathrm{y}_t)$ forms a bimodal density. Figure (2.1) shows a Normal mixture model for various values of μ_2 where $\mu_1 = 0$ and $\sigma_1^2 = \sigma_2^2 = 1$ with equal weight proportions $(\pi_1 = \pi_2 = 0.5)$. It can be seen that when values of μ_2 increase, the shape of mixture density $f(\mathrm{y}_t)$ changes from unimodal to bimodal.

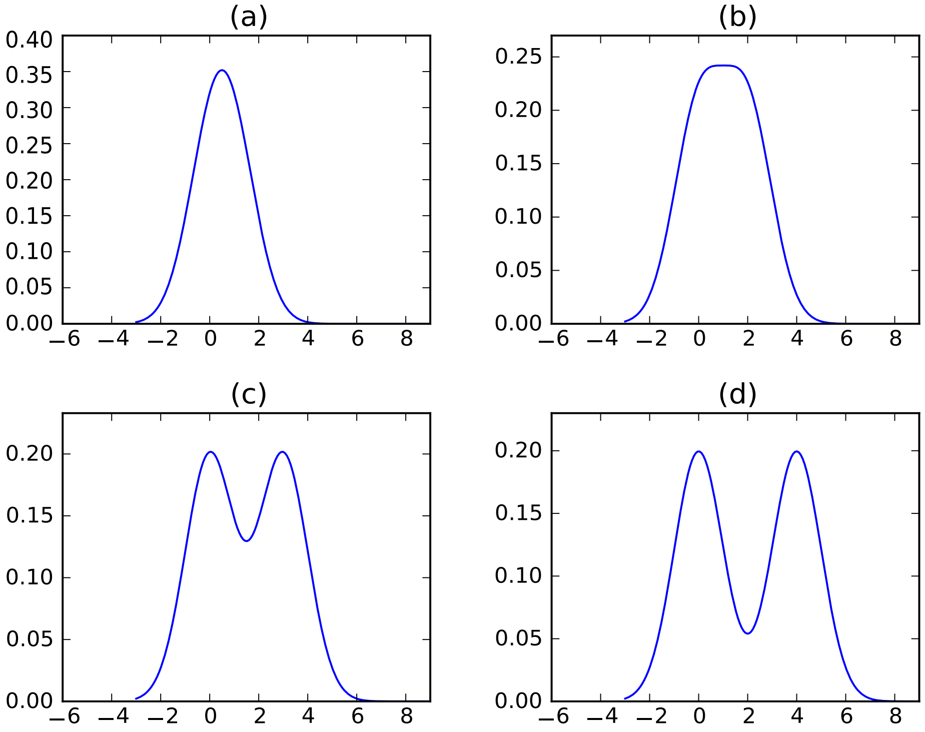


Figure 2.1: Plot of a mixture density of two univariate Normal components with equal weight proportions, common variance $\sigma^2 = 1$, and a fixed mean for the first component, $\mu_1 = 0$, and different values for the mean of second component, $\mu_2 = i$, namely; (a) *i*=1; (b) *i*=2; (c) *i*=3; (d) *i*=4.

2.3 Mixture model estimation

Given an *iid* random sample, $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ...\mathrm{y}_T)$, generated from a K-component mixture model defined in Equation (2.1), the likelihood function of these observations, assuming that y_t is independently distributed, can be written as

$$\begin{aligned} Pr(\mathbf{y}|\Theta) = L(\Theta|\mathbf{y}) &= \prod_{t=1}^{T} \{\pi_1 Pr_1(\mathrm{y}_t;\theta_1) + \pi_2 Pr_2(\mathrm{y}_t;\theta_2) + ... + \pi_K Pr_K(\mathrm{y}_t;\theta_k)\} \\ &= \prod_{t=1}^{T} \sum_{k=1}^{K} \pi_k \, Pr_k(\mathrm{y}_t|\theta_k). \end{aligned} \tag{2.4}$$

The maximum likelihood estimator of Θ is defined to be

$$\hat{\Theta} = \arg\max_{\Theta} L(\Theta|\mathbf{y}). \tag{2.5}$$

The main task is to find the parameter vector Θ that maximizes $L(\Theta|\mathbf{y})$. However, the function in Equation (2.4) is difficult to maximize directly as it involves a summation inside the logarithm operator. Additionally, it is not clear which component of the mixture generated each data point and thus which parameters require adjusting to fit that data point. Consequently, a number of methods have been developed for maximizing the log-likelihood function either using the expectation-maximization (EM) or Bayesian methods.

2.3.1 Expectation-Maximization algorithm (EM) of FMMs

One common approach to estimate the parameters $\Theta = (\pi, \boldsymbol{\theta})$ with respect to the observed data $\mathbf{y}$ is by maximizing the likelihood function using Expectation Maximization (EM) algorithm (Dempster et al., 1977). The EM procedure has been used to maximize the likelihood function when there are missing values or latent variables. The key idea of the EM algorithm is the data augmentation procedure (Tanner and Wong, 1987). In the mixture model context, the missing data is represented by a set of random discrete indicators $\mathbf{z} = (\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T)$, where $\mathrm{z}_t \in \{1, ..., K\}$ indicates which mixture component generated the observation y_t. Mathematically, given $\mathbf{z} = (\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T)$, the complete-data log-likelihood, $L_c(.)$, can be written as

$$\begin{aligned} L_c(\Theta|\mathbf{y},\mathbf{z}) = Pr(\mathbf{y},\mathbf{z}|\Theta) &= Pr(\mathbf{y}|\mathbf{z},\Theta)Pr(\mathbf{z}|\Theta), \\ &= \prod_{t=1}^{T}\prod_{k=1}^{K}\{\pi_k Pr_k(\mathrm{y}_t|\theta_k)\}^{\mathbb{I}(\mathrm{z}_t=1)}, \\ &= \prod_{t=1}^{T}\prod_{k=1}^{K}\pi_k^{\mathbb{I}(\mathrm{z}_t=1)} Pr_k(\mathrm{y}_t|\theta_k)^{\mathbb{I}(\mathrm{z}_t=1)}, \end{aligned} \tag{2.6}$$

where $\mathbb{I}(\mathrm{z}_t = 1) = 1$ if $\mathrm{z}_t = k$ holds, where $k = 1, 2, ..., K$, and $\mathbb{I}(\mathrm{z}_t = 1) = 0$ otherwise. The complete log-likelihood, $\ell_c(.)$, as

$$\begin{aligned} \ell_c(\Theta|\mathbf{y},\mathbf{z}) &= \sum_{t=1}^{T}\sum_{k=1}^{K} \mathrm{z}_{kt}\log[\pi_k f(\mathrm{y}_t;\theta_k)], \\ &= \sum_{t=1}^{T}\sum_{k=1}^{K} \mathrm{z}_{kt}\log\pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K} \mathrm{z}_{kt}\log f(\mathrm{y}_t,\theta_k). \end{aligned} \tag{2.7}$$

The key idea behind the EM algorithm is to set an upper bound function $Q-$function on the negative log-likelihood of the observed variables by introducing distributions over the latent variables. The EM algorithm involves two steps to maximize the above log-likelihood which are: Expectation step (**E-step**) and maximization step (**M-step**). **E-step** includes the finding the expectation with respect to the conditional distribution of latent variables $\mathbf{z}$ given data points $\mathbf{y}$ and the current estimation of parameters $\Theta^{(m)}$:

$$Q(\Theta|\Theta^{(m)}) = E_{\mathbf{z}\ |(\mathbf{y},\Theta^{(m)})}[\ell_c(\Theta|\mathbf{y},\mathbf{z})]. \tag{2.8}$$

In **M-step**, a new $\Theta = \Theta^{(m+1)}$ is computed, which maximizes the Q-function that is obtained in **E-step**:

$$\Theta^{(m+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(m)}). \tag{2.9}$$

Initially, the **M-step** requires the maximization of $Q(\Theta;\Theta^{(0)})$ with respect to Θ over the parameter space. This implies choosing $\Theta^{(1)}$ such that

$$Q(\Theta^{(1)};\Theta^{(0)}) \geqslant Q(\Theta;\Theta^{(0)}). \tag{2.10}$$

The **E-step** and the **M-step** are then implemented again with $\Theta^{(0)}$ replaced by the current fit $\Theta^{(1)}$. On the $(m+1)^{th}$ iteration the **E-step** and the **M-step** are alternated repeatedly until the changes in the log-likelihood values are less than some specified threshold, (McLachlan and Peel, 2000, p.24). The EM algorithm is numerically stable with each EM iteration increasing the likelihood value as

$$Q(\Theta^{(m+1)}) \geqslant Q(\Theta^{(m)}). \tag{2.11}$$

Before that, we need to define the posterior probability of latent variable z_{kt}. The E-step of the EM algorithm is carried out by replacing the z_{kt} by their expected values given the data $\mathbf{y}$ and current estimate of the model parameters $\Theta^{(m)}$. According to the Bayes theorem (McLachlan and Peel, 2000), we obtain

$$E(\mathrm{z}_{kt}|\mathbf{y},\Theta^{m}) = Pr(\mathrm{z}_{kt}=1|\mathrm{y}_t,\Theta^{m}) = \frac{Pr(\mathrm{y}_t|\mathrm{z}_{kt}=1)Pr(\mathrm{z}_{kt}=1)}{\sum_l Pr(\mathrm{y}_t|\mathrm{z}_{lt}=1)Pr(\mathrm{z}_{lt}=1)} = \frac{\pi_k f(\mathrm{y}_t,\theta_k)}{\sum_l \pi_l f(\mathrm{y}_t,\theta_l)} = \mathrm{w}_{kt}. \tag{2.12}$$

By substituting z_{kt} by w_{kt} in Equation (2.7), the **E-step** of the EM algorithm is then given as

$$Q(\Theta;\Theta^{(m)}) = E_{\Theta^{(m)}}\left\{\ell_c(\Theta|\mathbf{y},\mathbf{z})\right\}, \tag{2.13}$$

$$= \sum_{t=1}^{T}\sum_{k=1}^{K} \mathrm{w}_{kt}\log\pi_k + \sum_{t=1}^{T}\sum_{k=1}^{K} \mathrm{w}_{kt}\log f(\mathrm{y}_t,\theta_k). \tag{2.14}$$

The **M-step** involves maximization of the current expected complete data likelihood. Since π_k appears only in the first term, and θ_k only in the second term, we can maximize the two terms separately. To maximize the expression for π_k, we introduce the Lagrange multiplier $\mathbf{L}$ with the constraint that $\sum_{k=1}^{K}\pi_k = 1$, and solve the following equation (Bilmes, 1998):

$$\frac{\partial}{\partial\pi_k}\left[\sum_{t=1}^{T}\sum_{k=1}^{K}\mathrm{w}_{kt}\log\pi_k + \mathbf{L}(\sum_k \pi_k - 1)\right] = 0 \tag{2.15}$$

or

$$\sum_{t=1}^{T}\frac{1}{\pi_k}\mathrm{w}_{kt} + \mathbf{L} = 0. \tag{2.16}$$

Summing both sides over k, we get that $\mathbf{L} = -T$ resulting in:

$$\pi_k^{(m+1)} = \frac{1}{T}\sum_{t=1}^{T}\mathrm{w}_{kt}, \tag{2.17}$$

where $\mathrm{w}_{kt} = \dfrac{\pi_k f(\mathrm{y}_t,\theta_k)}{\sum_l \pi_l f(\mathrm{y}_t,\theta_l)}$. If the component parameters are unknown, they are estimated by finding the maximum likelihood estimator for the second sum of the expected complete data likelihood:

$$\theta_k^{(m+1)} = \frac{\sum_{t=1}^{T}\mathrm{w}_{kt}\ \mathrm{y}_t}{\sum_{t=1}^{T}\mathrm{w}_{kt}}. \tag{2.18}$$

For instance, in the case of a Poisson mixture model, Equation (2.18) can be written as

$$\lambda_k^{(m+1)} = \frac{\sum_{t=1}^{T}\mathrm{w}_{kt}\ \mathrm{y}_t}{\sum_{t=1}^{T}\mathrm{w}_{kt}}, \tag{2.19}$$

whereas, in the case of a Normal mixture model, for the parameters μ, the **M-step** can be written as

$$\mu_k^{(m+1)} = \frac{\sum_{t=1}^{T}\mathrm{w}_{kt}\ \mathrm{y}_t}{\sum_{t=1}^{T}\mathrm{w}_{kt}}, \tag{2.20}$$

and for σ

$$\sigma_k^{(m+1)} = \frac{\sum_{t=1}^{T}\mathrm{w}_{kt}\ (y_t - \mu_k^{(m+1)})}{\sum_{t=1}^{T}\mathrm{w}_{kt}}. \tag{2.21}$$

To illustrate the process of parameter estimation for a FMM via the EM algorithm, we implemented a fitting process, with threshold = 0.000001, to a two-component Normal mixture model with weights $\pi_1 = 0.3$ and $\pi_2 = 0.7$, equal variances $\sigma_1^2 = \sigma_2^2 = 1$ and different means, $\mu_1 = 2$ and $\mu_2 = 5$. Both Table (2.1) and Figure (2.2) show results of the estimation process using the EM procedure.

Parameter	π_1	μ_1	μ_2	σ_1^2	σ_2^2
True	0.3	2	5	1	1
Estimated	0.291	1.928	4.948	0.902	1.012

Table 2.1: The true and estimated values of a two-component Normal mixture model using EM algorithm for a simulated sample of 1000 observations.

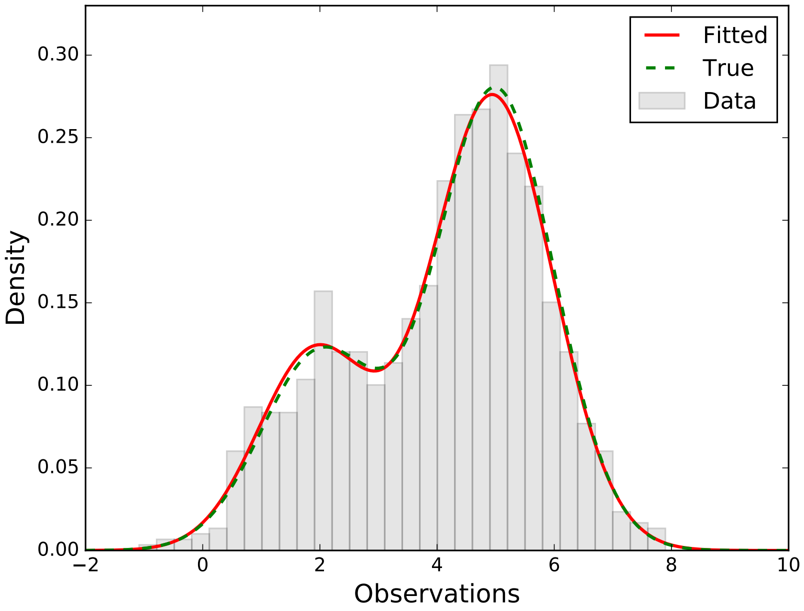


Figure 2.2: Fitting a two-component Normal mixture model using the EM algorithm.

2.3.2 The Bayesian estimation of FMMs

In the FMM in Equation (2.1), the unknown parameter vector $\Theta = (\pi;\boldsymbol{\theta})$ needs to be estimated. In order to obtain the posterior distribution of Θ, we need to combine the data-dependent likelihood function $L(\Theta;\mathbf{y})$ of the mixture model and the prior distribution of the unknown parameters $\Theta = (\pi;\boldsymbol{\theta})$. By assuming the independence of the prior distributions of the model parameters; $\boldsymbol{\theta}$ and π, the posterior distribution $Pr(\Theta|\mathbf{y})$ can be given as

$$Pr(\Theta|\mathbf{y}) \propto L(\boldsymbol{\theta},\pi;\mathbf{y})Pr(\pi)Pr(\boldsymbol{\theta}), \tag{2.22}$$

where $L(\boldsymbol{\theta},\pi;\mathbf{y}) = \prod_{t=1}^{T} Pr(\mathrm{y}_t|\boldsymbol{\theta},\pi) = \prod_{t=1}^{T}\{\sum_{k=1}^{K}\pi_k f(\mathrm{y}_t|\theta_k)\}$ is the likelihood, and $Pr(\boldsymbol{\theta},\pi) = Pr(\pi)Pr(\boldsymbol{\theta})$ is the joint prior distribution of $\boldsymbol{\theta}$ and π.

An efficient method to simplify the sampling from the posterior distribution is the data augmentation method proposed by Tanner and Wong (1987). The principle of this technique is based on sampling from the complete data posterior distribution $Pr(\Theta,\mathbf{z}|\mathbf{y})$ rather than $Pr(\Theta|\mathbf{y})$ by proposing auxiliary variables, called $\mathbf{z}$, also referred as latent indicator variables. If we know $\mathbf{y}$ and $\mathbf{z}$, then the analysis will be straightforward.

We assume that there are discrete latent indicators, $\mathbf{z} = \{\mathrm{z}_{kt}\}$, associated with each observation of the vector $\mathbf{y} = (\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_T)$. Since these indicators in real life are unknown parameters, the inference about a mixture model requires estimating two unknown quantities: the component indicators, $\mathbf{z}$, and the component parameters, $\Theta = (\pi,\boldsymbol{\theta})$. In the Bayesian perspective, in order to obtain those quantities, these can be sampled from the following complete data posterior

$$Pr(\mathbf{z},\pi,\theta|\mathbf{y}) \propto L_c(\boldsymbol{\theta},\pi;\mathbf{y},\mathbf{z})Pr(\pi)Pr(\boldsymbol{\theta}), \tag{2.23}$$

where $L_c(\boldsymbol{\theta},\pi;\mathbf{y},\mathbf{z})$ is the complete data likelihood of a finite mixture model, $Pr(\boldsymbol{\theta})$ and $Pr(\pi)$ are independent prior distributions of the parameter $\boldsymbol{\theta}$ and of the components weights π respectively.

The complete-data likelihood can be written as

$$\begin{aligned} L_c(\boldsymbol{\theta},\pi;\mathbf{y},\mathbf{z}) &= \prod_{t=1}^{T}\pi_{\mathrm{z}_t}Pr(\mathrm{y}_t|\theta_{\mathrm{z}_t}) \\ &= \prod_{k=1}^{K}\prod_{t:\ \mathrm{z}_t=k}\pi_k Pr(\mathrm{y}_t|\theta_k) \\ &= \prod_{k=1}^{K}\pi_k^{\sum_{t=1}^{T}\mathbb{I}(\mathrm{z}_t=k)}\prod_{t:\ \mathrm{z}_t=k}Pr(\mathrm{y}_t|\theta_k). \end{aligned} \tag{2.24}$$

To complete the Bayesian specification of the model, we need to specify priors for the unknown parameters of the model: π and θ. The prior on the component weights is represented by a Dirichlet distribution as

$$Pr(\pi) = \prod_{k=1}^{K}\pi_k \propto \prod_{k=1}^{K}\pi_k^{\delta_k-1} = Dir(\delta_1,\delta_2,...,\delta_K), \tag{2.25}$$

where δ_k, $k = 1,2,...,K$ are the positive ($\delta_k > 0$) hyper-parameters of the Dirichlet distribution. The prior on the component-specific parameter, θ, based on the form of the parametric distribution assumed for observations, $\mathbf{y}$. As a general case for representing the prior on the component-specific parameter, θ, we can write the following expression

$$\theta \sim Pr(\theta|\varphi), \tag{2.26}$$

where φ is referred to a collection of the hyper-parameters governing the shape of the prior distribution of θ. Common MCMC approaches can be employed. We use the Gibbs sampler to simulate from the full conditional posterior distributions of the FMM.

2.3.2.1 Estimation using the Gibbs sampler

The posterior distribution in Equation (2.23) involves three full conditional distributions which can be written as

$$\begin{aligned} \mathbf{z} &\sim Pr(\mathbf{z}|\mathbf{y},\pi,\boldsymbol{\theta}), \\ \pi &\sim Pr(\pi|\mathbf{y},\mathbf{z}), \\ \boldsymbol{\theta} &\sim Pr(\boldsymbol{\theta}|\mathbf{y},\mathbf{z}). \end{aligned} \tag{2.27}$$

It is easy to implement the Gibbs sampler to sample from those distributions. In Bayesian inference for FMMs, the mixing proportion $\{\pi_1, \pi_2, ..., \pi_k\}$ can be viewed as the prior distribution that one observation belongs to sub-population k. Given the observations, y_t, the full conditional posterior distribution of z_t can be obtained as

$$\begin{aligned} Pr(\mathrm{z}_t = k|\mathrm{y}_t, \pi, \theta) &\propto \pi_k Pr(\mathrm{y}_t|\theta_k) \\ &= \frac{\pi_k\; Pr(y_t|\theta_k)}{\sum_{l=1}^{K} \pi_l\; Pr(\mathrm{y}_t|\theta_l)}. \end{aligned} \tag{2.28}$$

From Equation (2.28), the marginal distribution of the z_t is a multinomial distribution

$$\mathrm{z}_t \sim Multi\{Pr(\mathrm{z}_t = 1), Pr(\mathrm{z}_t = 2), ..., Pr(\mathrm{z}_t = K)\}. \tag{2.29}$$

Given component indicators $\mathbf{z}$, the full conditional posterior of the component weights, π, can be sampled as follows

$$\begin{aligned} Pr(\pi|\mathbf{y},\mathbf{z},\delta) &\propto L_c(\boldsymbol{\theta},\pi;\mathbf{y},\mathbf{z})Pr(\pi|\delta) \\ &\propto \prod_{k=1}^{K} \pi_k^{\sum_{t=1}^{T}\mathbb{I}(\mathrm{z}_t=k)} \prod_{t:\ \mathrm{z}_t=k} Pr(\mathrm{y}_t|\theta_k) \prod_{k=1}^{K} \pi_k^{\delta_k-1} \\ &\propto \prod_{k=1}^{K} \pi_k^{\sum_{t=1}^{T}\mathbb{I}(\mathrm{z}_t=k)+\delta_k-1} \\ \pi &\sim Dir(n_1+\delta_1, n_2+\delta_2, \ldots, n_K+\delta_K), \end{aligned} \tag{2.30}$$

where $n_k = \sum_{l=1}^{T} \mathbb{I}_{\mathrm{z}_l=k},\ k=1,2,\ldots,K$, denote the allocation sizes. Given component indicators $\mathbf{z}$ and observation $\mathbf{y}$, the posterior of $\boldsymbol{\theta}$ is

$$\begin{aligned} Pr(\boldsymbol{\theta}|\mathbf{y},\mathbf{z}) &\propto L_c(\boldsymbol{\theta},\pi;\mathbf{y},\mathbf{z})Pr(\boldsymbol{\theta}) \\ &\sim Pr(\boldsymbol{\theta}) \prod_{t:\mathrm{z}_t=k} Pr(\mathrm{y}_t|\theta_k). \end{aligned} \tag{2.31}$$

Algorithm (3) provided by Marin and Robert (2014, p.183) describes the steps of sampling from the full conditional posterior distributions of a mixture model. Note that according to the posterior given in Equation (2.31), if density $f(\mathrm{y}_t|\theta_k)$ belongs to an exponential family of standard form,

Algorithm 3 : Gibbs Sampler for a K-component finite mixture model

```
Initialization: Choose $\pi^{(0)}$ and $\boldsymbol{\theta}^{(0)}$ arbitrarily
Iteration $m$ $(m \geqslant 1)$:
    1- Generate $\mathrm{z}_t^{(m)}$ $(t = 1,\ldots,T)$ from
        $Pr(\mathrm{z}_t^{(m)} = k|\pi_k^{(m-1)}, \theta_k^{(m-1)}, \mathrm{y}_t) \propto \pi_k^{(m-1)} f(\mathrm{y}_t|\theta_k^{(m-1)});\ k = 1,2,\ldots,K.$
    2- Generate $\pi^{(m)}$ from $Pr(\pi|\mathrm{z}^{(m)})$,
    3- Generate $\boldsymbol{\theta}^{(m)}$ from $Pr(\boldsymbol{\theta}|\mathrm{z}^{(m)}, \mathrm{y}_t)$.
```

$$f(\mathbf{y}|\theta_k) = h(\mathbf{y})\exp\{\theta_k\, R(\mathbf{y}) - \Psi(\theta_k)\}, \tag{2.32}$$

we can use a conjugate prior on each θ_k,

$$Pr(\theta_k) \propto \exp\{\theta_k\eta_k - \zeta_k\Psi(\theta_k)\}. \tag{2.33}$$

The $\theta_k{}'s$ are then independent of one another, given $\mathbf{z}$ and $\mathbf{y}$, with respective distributions

$$Pr(\theta_k|\mathbf{y},\mathbf{z}) \propto \exp\left\{\theta_k\left[\eta_k+\sum_{t=1}^{T}\mathbb{I}_{z_t=k}R(\mathrm{y}_t)\right]-\Psi(\theta_k)(n_k+\zeta_k)\right\}, \tag{2.34}$$

which are available in closed form by the virtue of conjugacy.

2.3.3 Label switching

When the Bayesian approach is applied to estimate the parameters of mixture models, a so-called label switching or non identifiability problem may occur (Stephens, 2000). In the mixture model context, this problem arises because of the invariance of the likelihoods with respect to the permutations of the component labels. In Bayesian analysis, this occurs when the prior distribution does not distinguish the components. Hence, the resulting posterior distribution will be invariant in the permutations of the labels, where it will be proportional to the product of a symmetric likelihood with a symmetric prior distribution (Stephens, 2000; Jasra et al., 2005; Papastamoulis and Iliopoulos, 2010). To explain this problem, let $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$ be independent observations from a finite mixture density with k of known components. Then

$$L(\Theta;\mathbf{y}) = L(\pi,\boldsymbol{\theta};\mathbf{y}) = \pi_1 f(\mathrm{y}_t;\theta_1)+\pi_2 f(\mathrm{y}_t;\theta_1)+...+\pi_k f(\mathrm{y}_t;\theta_k). \tag{2.35}$$

Let ρ_k be the set of permutations of the component indices $\{1,2,...,k\}$, and define

$$\rho(\pi,\boldsymbol{\theta}) = (\pi_{\rho_1},\pi_{\rho_2},...,\pi_{\rho_k},\theta_{\rho_1},\theta_{\rho_2},...,\theta_{\rho_k}). \tag{2.36}$$

We can obtain a mixture model with permutations,

$$\begin{aligned} L(\Theta;\mathbf{y}) = L(\pi,\boldsymbol{\theta};\mathbf{y}) &= \pi_{\rho_1} f(\mathrm{y}_t;\theta_{\rho_1})+\pi_{\rho_2} f(\mathrm{y}_t;\theta_{\rho_2})+...+\pi_{\rho_k} f(\mathrm{y}_t;\theta_{\rho_k}),\\ &= L(\rho(\pi,\boldsymbol{\theta});\mathbf{y}). \end{aligned} \tag{2.37}$$

When the prior distributions are exchangeable, then $Pr(\pi,\boldsymbol{\theta}) = Pr(\rho(\pi,\boldsymbol{\theta}))$ and thus the posterior distribution is $Pr(\pi,\boldsymbol{\theta}|\mathbf{y}) = Pr(\rho(\pi,\boldsymbol{\theta})|\mathbf{y})$. Consequently, when implementing the simulation process, the sampler may encounter a symmetry of the posterior distribution and the ergodic averages of estimating the weights. Hence, the component-specific parameters will lead to unreasonable results because they will be identical. In order to handle this problem, either an alternative prior modelling or a more tailored sophisticated approach are required (Marin and Robert, 2014).

The label switching issue has been addressed in the literature such as Diebolt and Robert (1994); Richardson and Green (1997); Stephens (2000); Celeux et al. (2000); Frühwirth-Schnatter (2001); Hurn et al. (2003) and Marin and Robert (2014).

In our study, we used the Identifiability Constraints (IC) method introduced by Diebolt and Robert (1994) where a constraint is imposed on one of the parameters of the mixture model.

2.3.4 Gibbs sampler for fitting a finite mixture of Normal distributions

In this section we evaluate the performance of Gibbs sampler. For this purpose, we carry out a simulation study to examine the behaviour of the sampler by fitting a finite mixture of Normal distributions using synthetic and real data.

2.3.4.1 A simulation study on synthetic data

In this sub-section, we check the our sampler by fitting independently six models with two components to six synthetic data sets, each one with length $T = 500$ observations. The six data sets have been generated according to the following model:

$$\sum_{k=1}^{2} \pi_k N(\mu_k, \sigma_k^2).$$

Each data set was generated under different mixing weights but with fixed means, $(\mu_1, \mu_2) = (4, 8)$, and variances, $(\sigma_1^2, \sigma_2^2) = (0.2, 1)$. The model can then be written as

$$\pi_1 N(4, 0.2) + \pi_1 N(8, 1).$$

The proposed six weights are shown in the Table (2.2) which also includes the parameters estimates of the six fitted models. Similarly, the parameters of each model are given conjugate priors as follows

$$\sigma_j^2 \sim InvGamma(a_j, b_j), \;\; \mu_j|\sigma_j^2 \sim N(\eta_j, \sigma_j^2|\zeta_j), \;\; \pi_j \sim Dir(\delta_j),$$

where $\eta_j, \zeta_j, a_j, b_j$ and δ_j are known hyper-parameters, $j = 1, 2$. The hyper-parameters need to be specified or endowed with hyper-priors when they cannot be specified (Marin et al., 2005). These hyper-parameters are commonly given non-informative hyper-priors or flat values (Gelman et al., 2014). For instance, the inverse Gamma with parameters $a = 0.001$ and $b = 0.001$ and thus a mean of $a/b = 1$ and a variance of $a/b^2 = 1000$ can give diffuse values of this form. The prior of the mean parameter can be assigned flat values from a Normal distribution with a shape parameter, $\eta = 0$, and a scale parameter, $\zeta = 0.001$, which has a large variance equal to 1000. The weight parameter, π, is given a Dirichlet prior with non-informative value, $\delta_k = 1$, $k = 1,2$. Given the above parametrization on the hyper-parameters of the priors distributions, we follow algorithm (4), given by Marin et al. (2005), to implement the sampling process.

Algorithm 4 : Gibbs Sampler for a two-component Normal mixture model with conjugate priors

```
1. Initialization: Choose $\pi_k^{(0)}$ and $\theta_k^{(0)}, k = 1,2$.
2. Iteration: for $m = 1,2,...,M$
    (a) Generate $\mathrm{z}_t^{(m)}; t = 1,...,T$ from $(k = 1,2)$
            $Pr(\mathrm{z}_t^{(m)} = 1) = 1 - Pr(\mathrm{z}_t^{(m)} = 2) \propto \frac{\pi_k^{(m-1)}}{\sigma_k^{(m-1)}} \exp\left(-\frac{(\mathrm{y}_t - \mu_k^{(m-1)})^2}{2(\sigma_k^2)^{(m-1)}}\right).$
        Compute: $n_k^{(m)} = \sum_{l=1}^{n} \mathbb{I}_{\mathrm{z}_l^{(m)}=k}$ and $s_k^{\mathrm{y}(m)} = \sum_{l=1}^{n} \mathbb{I}_{\mathrm{z}_l^{(m)}=k}\, \mathrm{y}_l.$
    (b) Update $\pi_k^{(m)}$ from $Dir(\delta_1 + n_1^{(m)}, \delta_2 + n_2^{(m)})$.
    (c) Generate $\mu_k^{(m)};\ k = 1,2$ from
            $N\left(\frac{\eta_k \zeta_k + s_k^{\mathrm{y}(m)}}{\zeta_k + (n_k)^{(m)}}, \frac{\sigma_k^{2\,(m-1)}}{\zeta_k + (n_k)^{(m)}}\right).$
        Compute: $s_k^{\nu\,(m)} = \sum_{t=1}^{n} \mathbb{I}_{\mathrm{z}_l^{(m)}=k}(\mathrm{y}_t - \mu_k^{(m)})^2.$
    (d) Generate $\sigma_k^{2\,(m)};\ k = 1,2$ from
            $InvGamma(a_k + 0.5(n_k^{(m)} + 1),\ b_k + 0.5\zeta_k(\mu_k^{(m)} - \eta_k)^2 + 0.5(s_k^{\nu\,(m)})).$
```

2.3.4.2 Results using synthetic data

We run separately the Gibbs sampler for 12000 iterations for each model. We adopted the last 10000 iterations for inference and discarded 2000 iterations as a burn-in period. We used the artificial constraint on the means parameters ($\mu_1 < \mu_2$) to overcome the label switching problem. Table (2.2) shows the resulting posterior estimates of six models; it also provides 95% credible intervals with (2.5%-97.5%) quantiles for each parameter of each model. In addition, we also provide the trace-plots, histograms and autocorrelation function (ACF) for each each parameter of each model as displayed in Figures (2.3-2.8). For each of these figures, the first column involves the trace plots of all model parameters, where the grey colour refers to the produced samples and the dashed black colour refers to the posterior mean, whereas the bold green colour represents the burn-in period. The second column represents the histograms of all samples in grey. The bold dashed red line refers to the posterior means. The last column represents all ACF plots of all model parameters. As can be seen from the results in Table (2.2) and Figures (2.3-2.8), the sampler performs well in estimating the true parameters of all six models. The sampler has a good sampling mixing within the provided credible intervals and also has a rapid behaviour to reach the targeted posterior by very few iterations. Moreover, the ACF plots suggest that there is no correlation between the samples produced by the sampler, except in one case, concerning the sixth model in Figure (2.8). More specifically, for both the mean and variance parameter of the second component there appear to be a slight correlation at the first lag and then began to fade quickly. This can be easily treated by increasing the burn-in period. Finally, we provide the plots of predictive densities of all models fitted under different mixing weights. Figure (2.9) compares fitting each model with the histogram of each corresponding data set.

Model	True Weights: π	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$
1	π_1=0.3, π_2=0.7	0.2995	0.7004	3.9542	7.9869	0.1780	1.0664
		(0.2593, 0.3397)	(0.6602, 0.7406)	(3.8862, 4.022)	(7.8765, 8.0972))	(0.1364, 0.2195)	(0.9004, 1.2323)
2	π_1=0.4, π_2=0.6	0.3991	0.6008	3.9930	8.0017	0.1801	0.9768
		(0.3559, 0.4423)	(0.5576, 0.6440)	(3.9337, 4.0523)	(7.8883, 8.1152)	(0.1433, 0.2168)	(0.8123, 1.1412)
3	π_1=0.6, π_2=0.4	0.6035	0.3965	3.9861	7.8993	0.2439	0.9038
		(0.5611, 0.6460)	(0.3539, 0.4389)	(3.92851, 4.0425)	(7.76311, 8.0355)	(0.20171, 0.2861)	(0.7061, 1.1016)
4	π_1=0.7, π_2=0.3	0.6995	0.3004	3.9861	7.9453	0.1819	1.0333
		(0.6592, 0.7398)	(0.2601, 0.3407)	(3.9411, 4.0311)	(7.7799, 8.1107)	(0.1536, 0.2101)	(0.7772, 1.2894)
5	π_1=0.8, π_2=0.2	0.7979	0.2020	3.9758	8.0514	0.1920	0.9977
		(0.1664, 0.2376)	(0.7623, 0.8335)	(3.9324, 4.0192)	(7.8532, 8.2496)	(0.1648, 0.2193)	(0.6945, 1.3009)
6	π_1=0.9, π_2=0.1	0.8959	0.1040	3.9945	7.9707	0.1848	1.5535
		(0.8685, 0.9233)	(0.0766, 0.1314)	(3.9541, 4.0350)	(7.5830, 8.3584)	(0.1593, 0.2103)	(0.7267, 2.3803)

Table 2.2: Results of the parameter estimation of all two-components Normal mixture models. All models were fitted independently to data set of length $T = 500$, each one with a different mixing weight but fixed mean and variance.

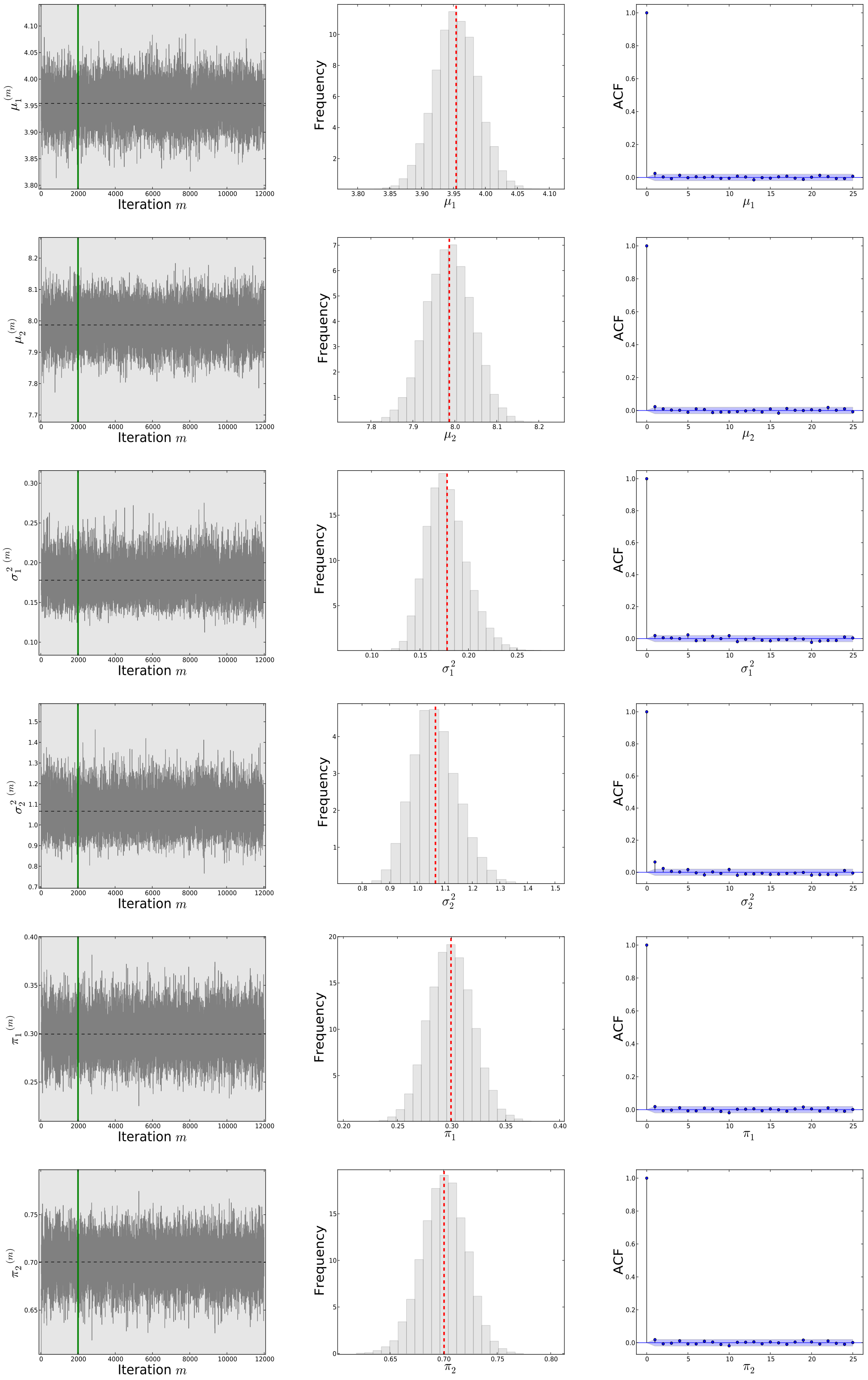


Figure 2.3: Trace-plots, histograms and ACF functions of all posterior parameters of the model with the weights: $\pi_1 = 0.7$ and $\pi_2 = 0.3$.

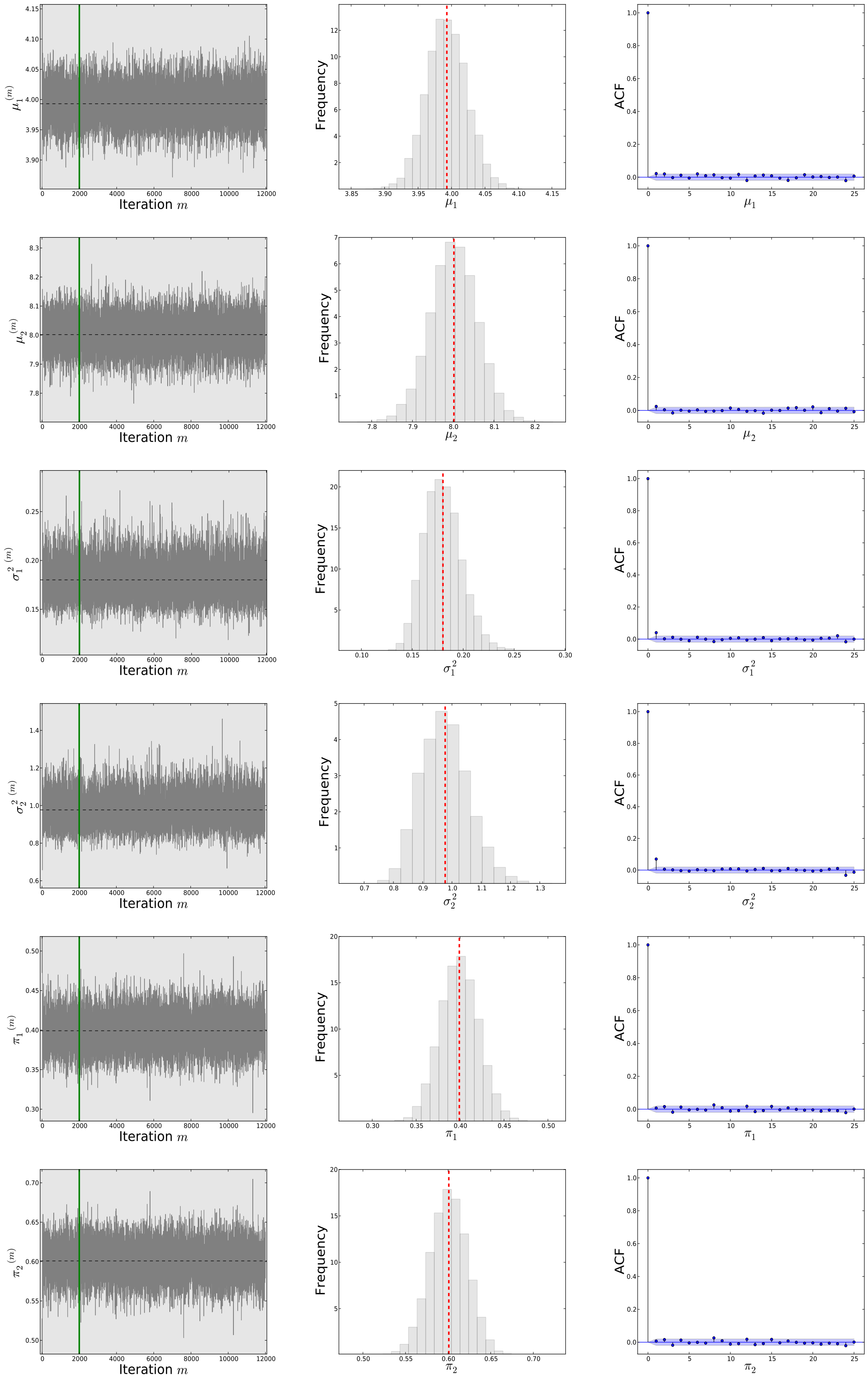


Figure 2.4: Trace-plots, histograms and ACF functions of all posterior parameters of the model with the weights: $\pi_1 = 0.4$ and $\pi_2 = 0.6$.

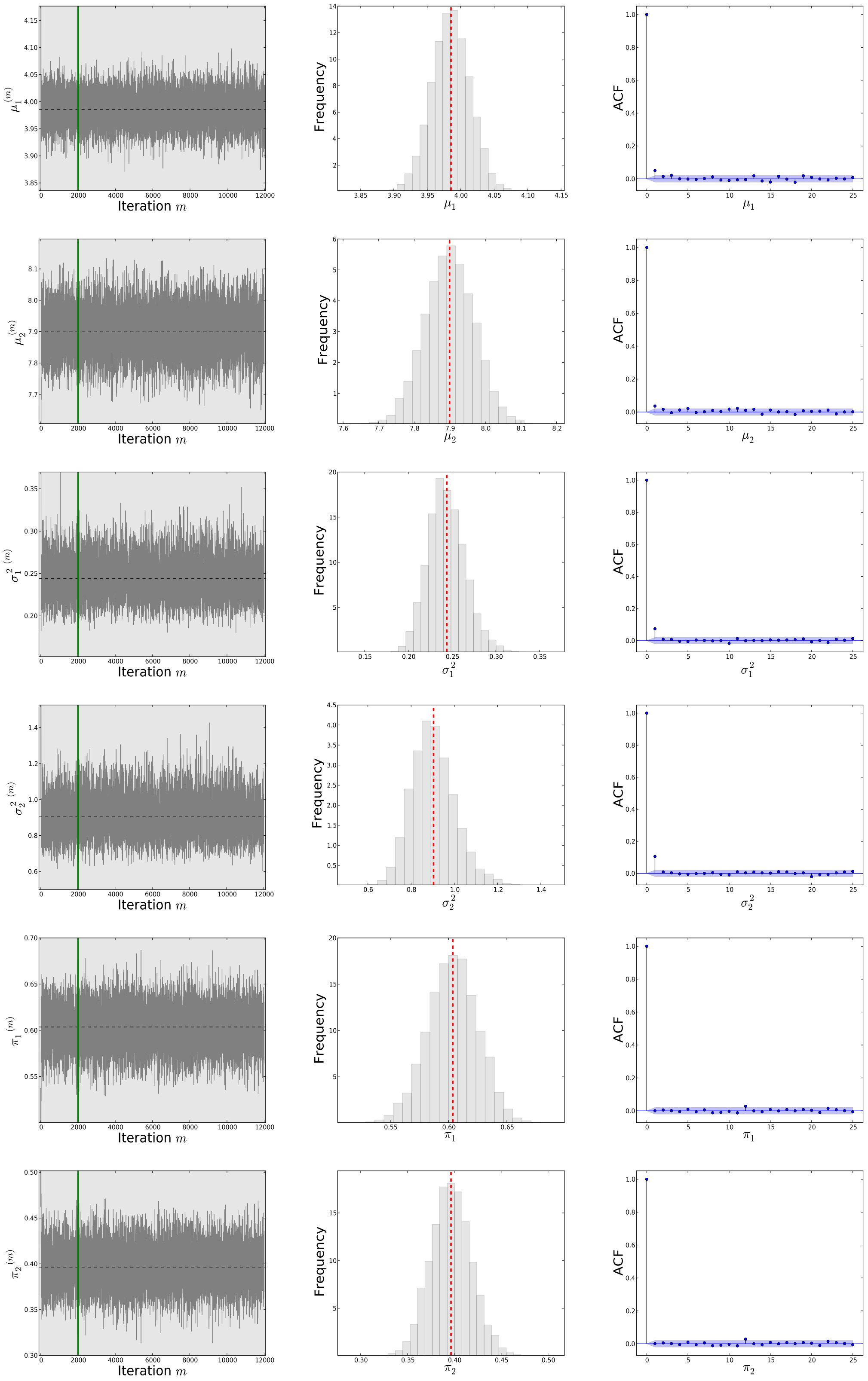


Figure 2.5: Trace-plots, histograms and ACF functions of all posterior parameters of the model with the weights: $\pi_1 = 0.6$ and $\pi_2 = 0.4$.

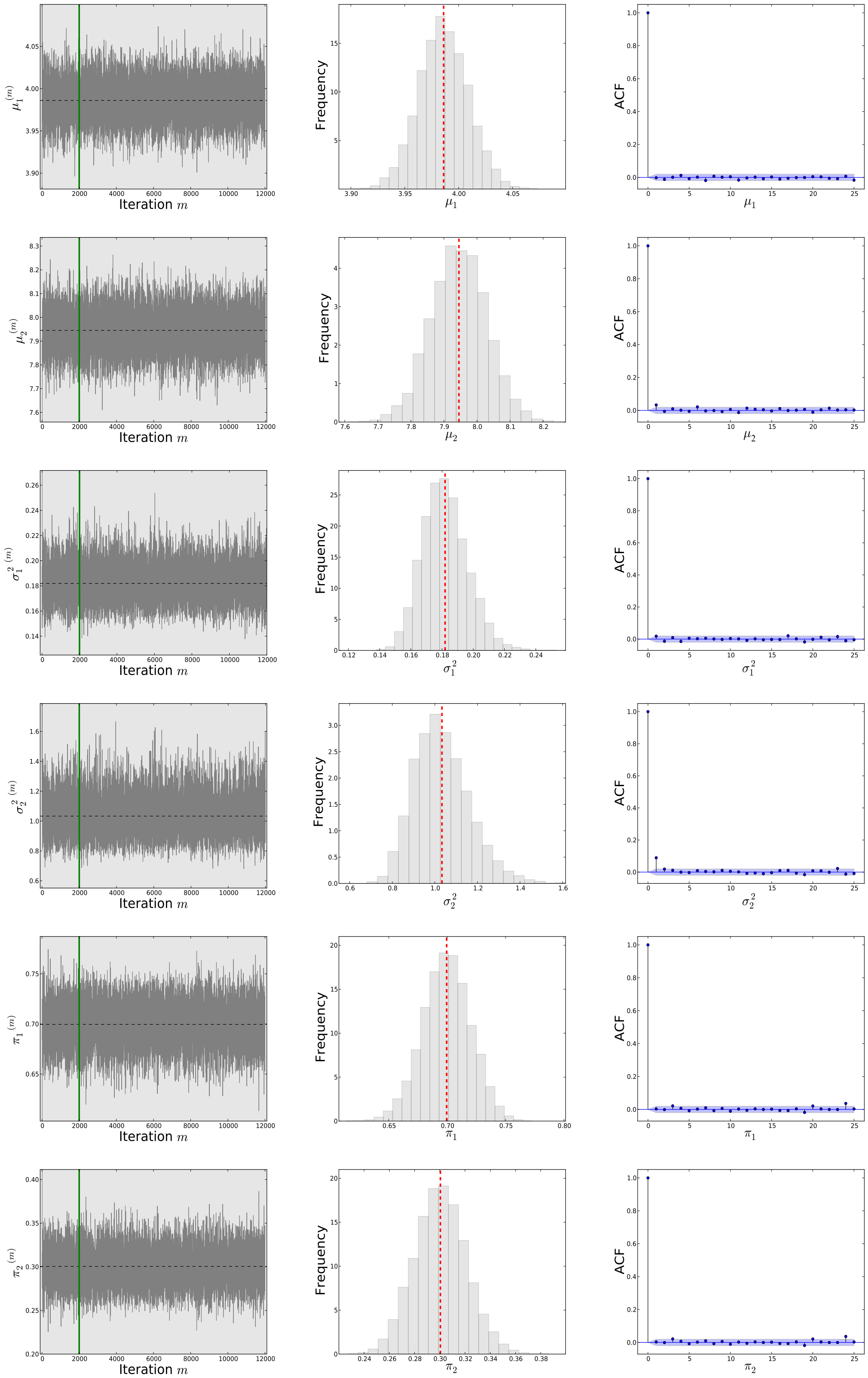


Figure 2.6: Trace-plots, histograms and ACF functions of all posterior parameters of the model with the weights: $\pi_1 = 0.7$ and $\pi_2 = 0.3$.

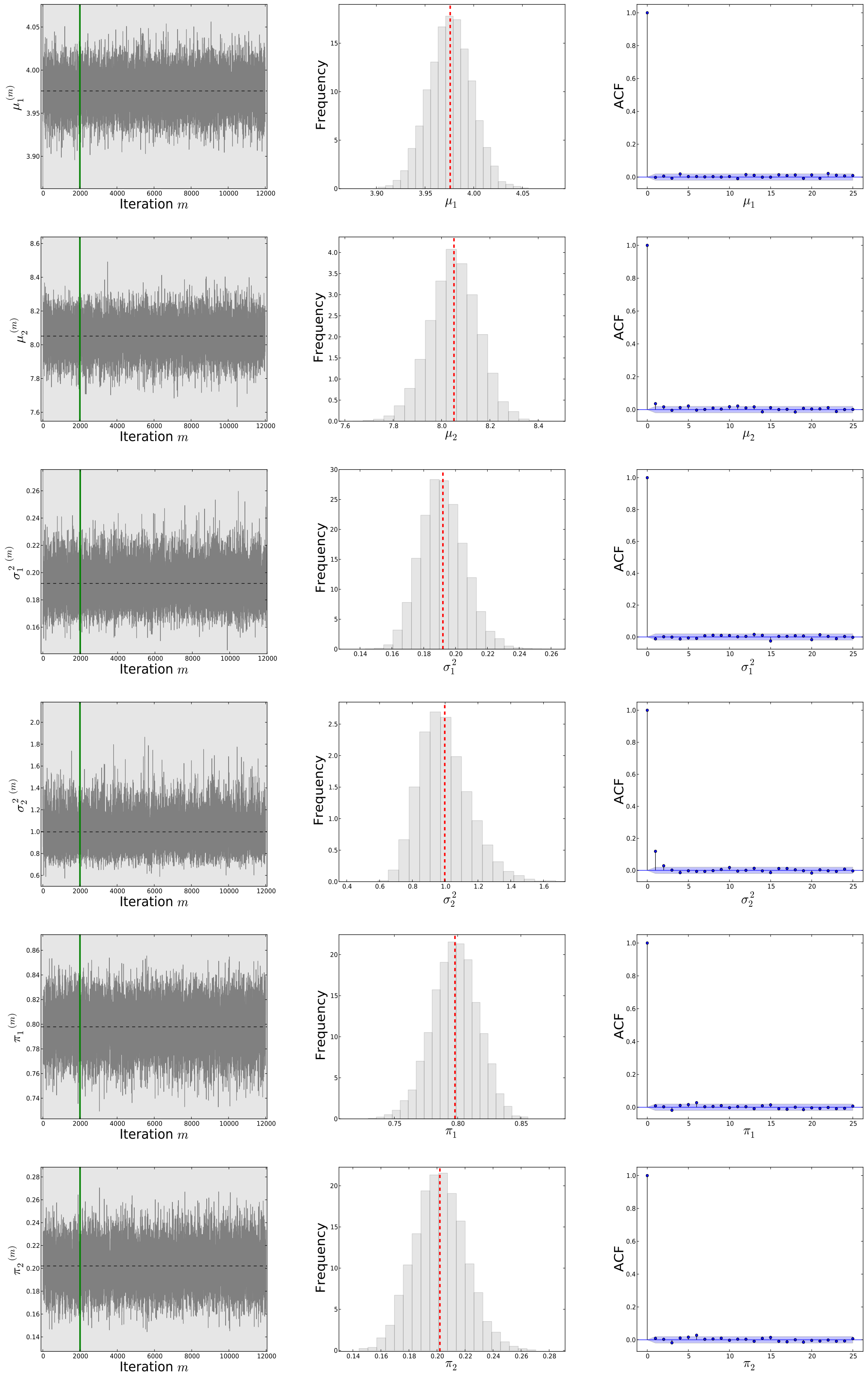


Figure 2.7: Trace-plots, histograms and ACF functions of all posterior parameters of the model with the weights: $\pi_1 = 0.8$ and $\pi_2 = 0.2$.

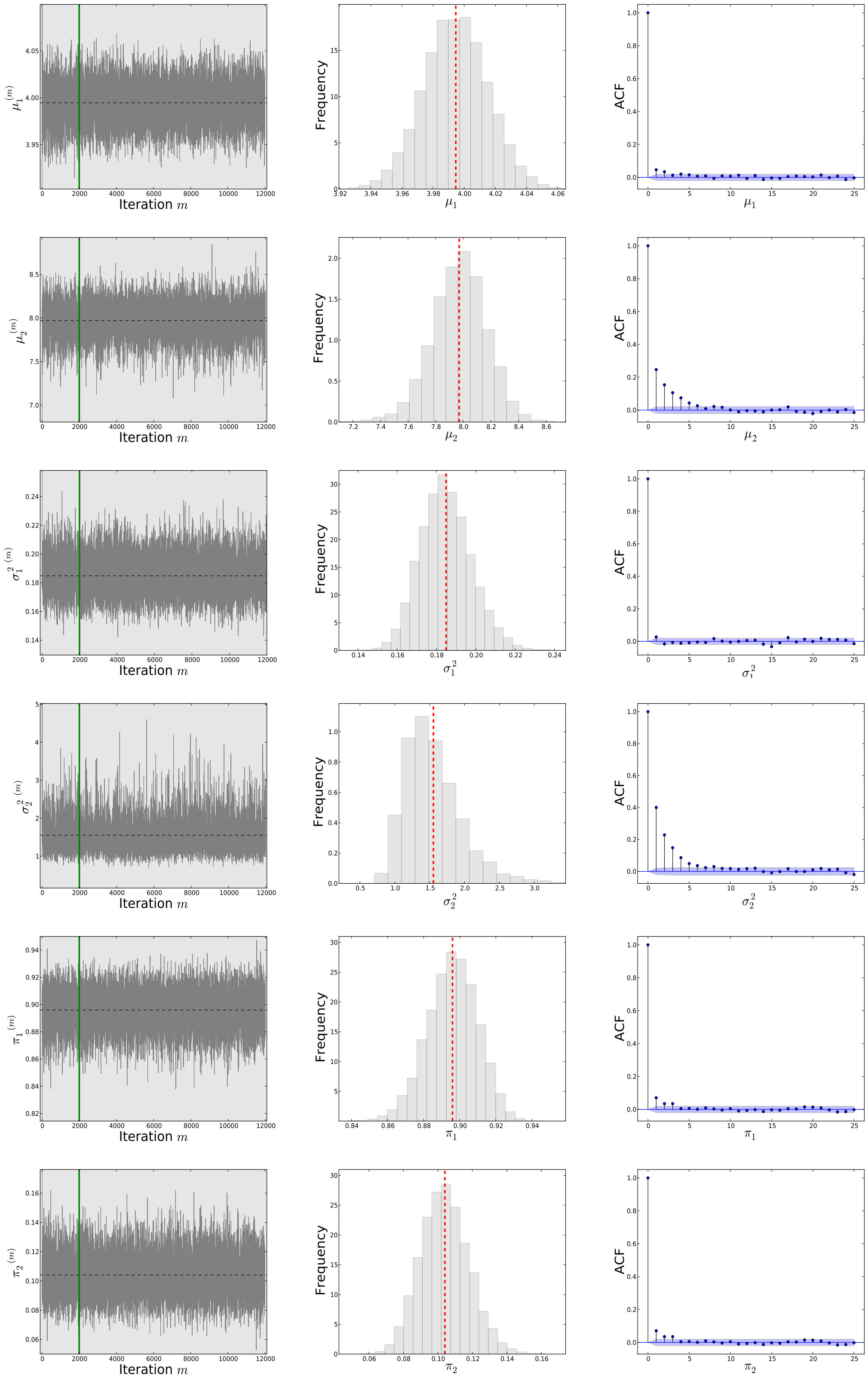


Figure 2.8: Trace-plots, histograms and ACF functions of all posterior parameters of the model with the weights: $\pi_1 = 0.9$ and $\pi_2 = 0.1$.

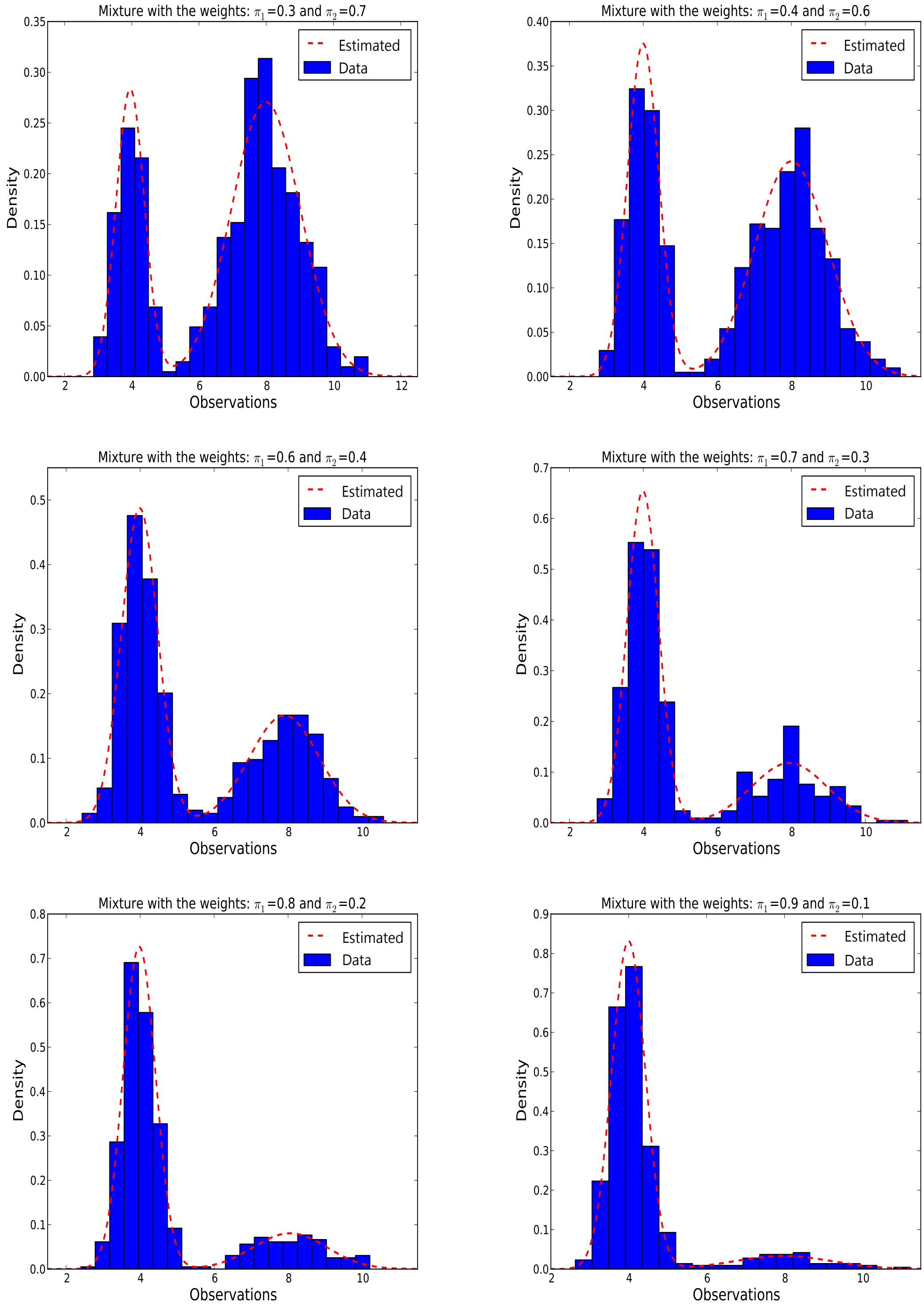


Figure 2.9: Fitting of predictive density of all model to synthetic sets of length 500 observation based on the two-component Normal mixture with true weights as labelled in the title each figure.

2.3.4.3 Fitting a finite mixture model to real data

In this section, we also perform the Gibbs sampler using a real application involving the Acidity data. We fitted a two-component Normal mixture model to these data. The acidity data ($\bar{y} = 5.1051$ and $s = 1.0384$) consist of 155 acidity measurements made on lakes in North-central Wisconsin. These data were analysed by many researchers. For example, Crawford (1994) used these data to fit a Normal mixture model on the log-scale. Richardson and Green (1997) analysed these data with unknown number of components using reversible jump MCMC. McLachlan and Peel (2000) fitted a mixture of two unrestricted Normal components for these data using the bootstrap method. Figure 2.10 shows the histogram of acidity data and a plot of the predictive density fitted to those data.

We follow the same information used with the synthetic data in sub-section (2.3.4.1) with respect to the prior specification of the model parameters. We also ran the sampler for 12000 iterations as an original sample and kept the last 10000 iterations (the first 2000 iterations were discarded as a burn-in period) for summarizing the results. We summarize the parameter estimates of the model in Table (2.3), which shows almost % 60 of and % 40 of data are concentrated in the first and second component respectively. Figure (2.10) show the model fit.

Estimated Parameters	Values
$\hat{\mu}$	(4.3237, 6.2141)
$\hat{\sigma}$	(0.1379, 0.3229)
$\hat{\pi}$	(0.5848, 0.4151)

Table 2.3: Parameter estimates for two-component Normal mixture model fitted to the Acidity data.

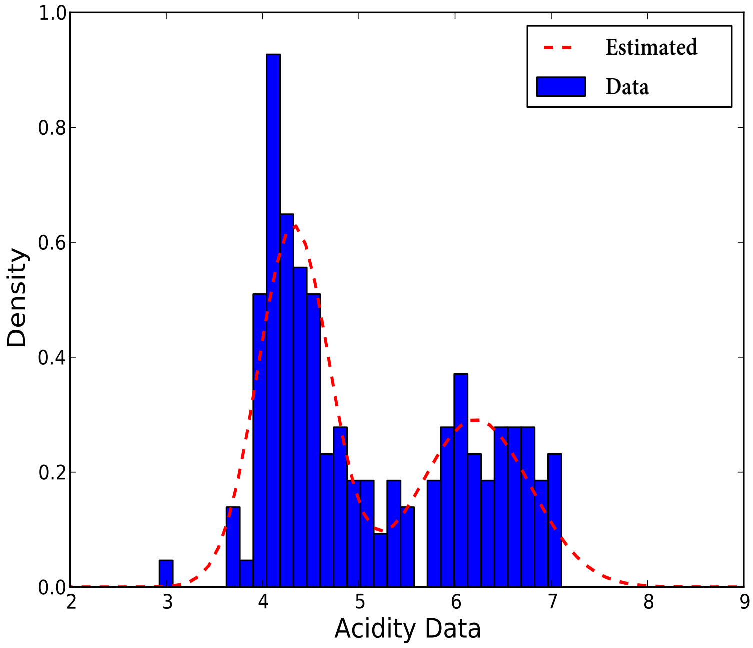


Figure 2.10: A two-component Normal mixture model fitted to the Acidity data.

2.4 Summary

In this chapter, we provided a general definition for FMMs with known number of components. We presented two methods for estimating the parameters of FMMs, the Expectation Maximization (EM) method as well as MCMC methods using the Gibbs sampler. This has allowed us to set out core ideas such as label switching. We implemented a simulation study to fit a two-component Normal mixture model. In this study, we introduced six models with different weights to evaluate the performance of the Gibbs sampler. Also, the sampler was employed to fit a two-component Normal mixture using a real application involving the Acidity data.

Chapter 3

Hidden Markov Models

3.1 Introduction

There are many phenomena and systems that involve sequentially correlated data. Thus, the mixture models described in the previous chapter, where data are assumed independent of each other, do not take into account this serial dependency between observations. Alternatively, hidden Markov models (HMMs) are described as a more powerful tool to account for such dependency. Therefore, the HMMs are considered as a generalization of mixture models. The serial dependence between the data can be described using an unobserved process, the so called the Markov process, which can be thought to explain another observed process. In this thesis, we consider mainly discrete-time and finite state space HMMs, where the unobserved process is discrete in both time and state space.

In this chapter, we review some relevant literature on the applications of HMMs. This chapter also provides the basic definitions of hidden Markov models. We also review the estimation of HMM parameters using the so called Baum-Welch algorithm.

3.2 Literature review

A theory of HMMs was introduced in the late 1960s through a series of papers published by Baum and Petrie (1966); Baum and Eagon (1967); Baum et al. (1970). This class of models has been successfully used for modelling and classifying dynamic behaviours. HMMs may be applied for different types of data: discrete, continuous, univariate, multivariate, mixed and mixture data. Consequently, they have been widely used in many fields, such as; econometrics (Hamilton, 1989; Billio et al., 1999); finance (Bhar and Hamori, 2006); speech recognition, image analysis, and time series prediction (Derrode et al., 2006; Rabiner, 1989); and psychology (Raijmakers and Molenaar, 2004; Visser et al., 2002). We next give some examples of applications for these models.

There are several challenges in psychology and medicine concerning the diagnosis and determining the real behaviour of patients. These challenges arise because of population heterogeneity, cohabitation of different patients and medical diseases, and diagnostic uncertainty. Hence, it is not easy to measure behavioural indicators of those phenomena where we are interested in the behaviour of a particular disease or of a patient. Therefore, HMMs are more appropriate models in these cases due to their flexibility in the presence of unobservable behaviour.

For HMMs applications, Jackson et al. (2003) proposed a Multi-stage Markov Model to describe aortic aneurysm patients. Sometimes the process of checking patients is not without mistakes and misclassification problems. Hence, HMMs were suggested to estimate transmission averages and likelihoods of state misclassification. A generalized regression model was used to model explanatory variables for transitions between statuses and probabilities of misclassification. In order to reduce uncertainty, Jackson et al. (2003) introduced Hazard Models to link transitions with the age variable for detecting whether there is an age effect. The findings proved that HMM models were sensitive to the assumptions of the study and suggested that the older adults are at increased risk of aortic aneurysm compared to younger adults. The Weibull distribution was proposed as an alternative for the exponential distribution in the estimation process of the sojourn states (Jackson et al., 2003).

Visser et al. (2002) used HMMs for psychological studies. The model was used to quantify knowledge that subjects express in an implicit learning task. The suggested procedure for comparing models with different constrains imposed on their parameters was the Maximum Likelihood method. They introduced a discrete-time HMM model instead of a continuous time HMM model due to the former being more convenient. Simulation experiments were implemented for the evaluation and model selection. Several candidate criteria were introduced for selecting the best model. They suggested, in addition of the standard criteria: AIC and BIC, two criteria which are the Adjusted Akaike Information Criterion (A-AIC) and Adjusted Bayesian Information Criterion (A-BIC). Their results proved that AIC and BIC are inappropriate in evaluating large models. After having chosen a final model, they used a prediction error measure to test the validity of chosen model (Visser et al., 2002). Nevertheless, these criteria; A-AIC and A-BIC will not be considered in this thesis as do not take into account the uncertainty about the model parameters.

Wall and Li (2009) performed a study to describe unobserved behaviour. They referred to two types of models to describe the effects of some of the unobserved variables on alcoholism addiction: the Multiple Indicator Hidden Markov Model and the Univariate Hidden Markov

Model. They determined two variables; "healthy" and "unhealthy" which were associated with each case (patient) as latent variables. The study used two kinds of data; longitudinal data that were classified depending on type of patient (Alcohol Specific, Alcohol Chronic, Alcohol Acute, and Not Alcohol), and monthly total data. Since the observations are monthly counts of medical visits, they proposed a two-state Poisson hidden Markov model (HMM). The purpose of the study was to investigate whether medical care reduces the probability of entry to unhealthy state that is identified by the medical visits (Wall and Li, 2009).

Furthermore, Hidden Markov Models have been used to analyse clustering and longitudinal data in describing some diseases. Scott et al. (2005) introduced a hidden Markov model for investigating the effect of an anti-psychotic drug and clozapine for schizophrenia patients. The univariate analysis for complicated medical diseases is not suitable for revealing detailed characteristics about the disease under study because of the presence of heterogeneity. This heterogeneity can be interpreted as the different features among patients. Hence, a longitudinal multivariate analysis is more suitable to describe the disease. A clustering method has been used as it is suitable for identifying complicated relationships among medical cases. However, there are some obstacles related to the nature of the data, and the procedure followed in estimating the parameters. Scott et al. (2005) offered a HMM to address those issues because of its ability in dealing with temporal data when estimating model parameters and classifying observations. HMMs in turn have some problems regarding time-homogeneity. Sometimes hypotheses of time-homogeneity may not be valid, particularly since there are unequal temporally intervals during treatment process. Therefore, the authors proposed a non-homogeneous HMM for this purpose. The findings of the study suggest that the clozapine is more effective than haloperidol in antipsychotic therapy.

Various studies have considered Hidden Markov Models as an accurate early-warning system. Rafei et al. (2012) used a Hidden Markov Model to identify the abnormal cases of a pulmonary disease, rampant tuberculosis, in Iran over the period 2005-2011. The study sample was based on data gathered weekly from sputum smear of patients. The model's parameters were estimated using maximum likelihood estimation and the Bayesian framework. The data were presented as a weekly bivariate discrete sequence. The usual phase represented what was expected in the diagnosis process of the disease, and the abnormal phase represented what exceeds the normal phase. Since the data were discrete, Rafei et al. (2012) proposed a Poisson mixture model to fit the data, and introduced two methods for estimating the model parameter (λ); one without seasonal trends, and one with seasonal trends. HMMs were applied for both methods. The basic idea of the study was based on the abnormal increase in counts of patients that exceeds normal diagnosis phase. The authors relied on multiple regression models proposed by Serfling (1963), derived from Fourier Equations. The two models below represent a function of the model parameters:

$$\lambda_{1t} = E\left(Y_t|S_t = 1\right) = \beta_0 + \beta_1 t + \beta_2 \sin\left(\frac{2\pi t}{r}\right) + \beta_3 \cos\left(\frac{2\pi t}{r}\right).$$

$$\lambda_{2t} = E\left(Y_t|S_t = 2\right) = \left(\beta_0 + \beta_e\right) + \beta_1 t + \beta_2 \sin\left(\frac{2\pi t}{r}\right) + \beta_3 \cos\left(\frac{2\pi t}{r}\right).$$

The Bayesian Information Criterion (BIC) and adjusted R-squared were proposed by the authors as criteria to select the best model. Finally, Rafei et al. (2012) concluded from their findings that the results using either criterion suggested that the HMM model with seasonal trend was better than the model without seasonal trend.

HMMs have also been used in epidemiology. Cooper and Lipsitch (2004) used HMMs to analyse hospital infection data. They presented a new method for parameter estimation of structured hidden Markov models for hospital infections data compared to an unstructured (standard) HMM and used their model to evaluate their method. They analysed monthly infection counts that followed a Poisson distribution. They found that both methods can offer considerable improvements over currently used approaches when hospital infection spread is important. Compared to the standard hidden Markov model, the new approach is more biologically plausible, and allows key epidemiological parameters to be estimated.

3.3 Definition of hidden Markov models

The hidden Markov model (HMM) is a statistical model that involves two stochastic processes. The first process $(\mathbf{Z}_t;\ t = 1, 2, ..., T)$ is an unobserved or hidden process (state process), satisfying the Markov property, where $\mathbf{z} = (\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T)$ denotes its corresponding realizations. The second process $(\mathbf{Y}_t;\ t = 1, 2, ..., T)$ is an observed process (state-dependent process) and $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$ represents one possible realization. When $\mathbf{Z}_t$ is known, the distribution of $\mathbf{Y}_t$ can be determined based only on $\mathbf{Z}_t$ (Zucchini and MacDonald, 2009). The dependency between Markovian hidden states and observed state can be illustrated in the directed graph in Figure (3.1).

We can summarize the relationship between those two processes under the following assumptions:

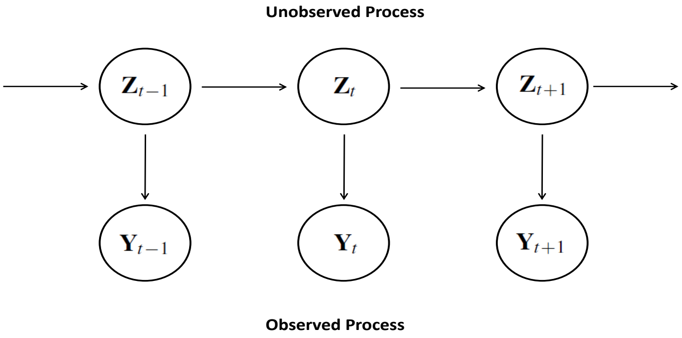


Figure 3.1: Graphical representation of the dependence structure of a discrete-time finite state-space HMM.

1. Markov assumption:

$$Pr(\mathbf{Z}_t = \mathrm{z}_t | \mathbf{Z}_{t-1} = \mathrm{z}_{t-1}, \mathbf{Z}_{t-2} = \mathrm{z}_{t-2}, ..., \mathbf{Z}_1 = \mathrm{z}_1) = Pr(\mathbf{Z}_t = \mathrm{z}_t | \mathbf{Z}_{t-1} = \mathrm{z}_{t-1}); \quad (3.1)$$

2. Conditional independence:

$$Pr(\mathbf{Y}_t = \mathrm{y}_t | \mathbf{Y}_{t-1} = \mathrm{y}_{t-1}, \mathbf{Y}_{t-2} = \mathrm{y}_{t-2}, ..., \mathbf{Y}_1 = \mathrm{y}_1, \mathbf{Z}_t = \mathrm{z}_t) = Pr(\mathbf{Y}_t = \mathrm{y}_t | \mathbf{Z}_t = \mathrm{z}_t). \quad (3.2)$$

HMMs can be described as homogeneous in the sense that the Markov chain $\{\mathbf{Z}_t\}$ and the conditional independence of $\mathbf{Y}_t$ given $\mathbf{Z}_t$ both are homogeneous and do not depend on t either. There are several theoretical texts on HMMs, for instance Cappé et al. (2005); Frühwirth-Schnatter (2006); Zucchini and MacDonald (2009); Visser (2011); Dymarski (2011).

We are interested in the parametric HMMs, where the observed process follows a parametric distribution (state-dependent distribution), given an unobserved process. In order to specify a HMM completely, the following elements have to be given:

1. The number of states in the model, K. We denote the individual states as:

$$k = (1, 2, ..., K); \quad (3.3)$$

where the hidden state chain, $\mathbf{Z}_t = \mathrm{z}_t$, can take values from the set of all possible states in

Equation (3.3) at any time.

2. The state transition probability distribution, $\mathbf{A} = \{a_{jk}\}$, where

$$a_{jk} = Pr\{\mathrm{z}_t = k | \mathrm{z}_{t-1} = j\}; \quad 1 \leq j,k \leq K, \tag{3.4}$$

is the probability that the state at time t is k, given that the state at time $t-1$ is j. The matrix of all transition probabilities of order $K \times K$ can expressed as

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,K} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{K,1} & a_{K,2} & \cdots & a_{K,K} \end{bmatrix}$$

3. The initial state distribution $\pi = \{\pi_k\}$ where π_k is the probability that the model is in state k at the time $t = 1$ with

$$\pi_k = Pr\{\mathrm{z}_1 = k\} \quad 1 \leq k \leq K. \tag{3.5}$$

4. The state-specific distribution, where observations can be modelled as a sequence of the random variables, that either take a discrete or continuous nature and follows some distribution that is parametrized by the parameter θ_k (Cappé et al., 2005; Zucchini and MacDonald, 2009). In this case, the model will be called a parametric HMM and its parameters will be referred to as $\Theta = (\pi, \mathbf{A}, \theta)$. The parameter θ here accounts for a generic parameter that can be a single parameter, or a vector of the model's parameters. For a discrete case, the observations can be modelled, for example, by the Poisson distribution with a probability mass function *(pmf)* as

$$p(\mathbf{y}|\theta_k) = \frac{\mathrm{e}^{-\lambda_k} \lambda_k^{\mathbf{y}}}{\mathbf{y}!}, \quad \mathbf{y} \in \mathbb{N}, \tag{3.6}$$

where $\theta_k = \lambda_k$ denotes the state-specific mean parameter of Poisson distribution. In the continuous case, the observations can follow the probability density function (*pdf*) of, for example, Normal or Gaussian distribution

$$f(\mathrm{y}_t; \theta_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{-\frac{(\mathrm{y}_t - \mu_k)^2}{2\sigma_k^2}\right\}, \tag{3.7}$$

where $\theta_k = (\mu_k, \sigma_k^2)$; $k = 1, 2, ..., K$. Note that we replaced the general notation of the observed process, $Pr(.)$, in Equation (3.2) by the *pmf* and *pdf* for the examples given above.

3.4 HMM as a generative model

In this section we develop an algorithm to generate a sequence of observations from some parametric HMM. The motivation behind this subsection is first, to give a wider understanding to how the mechanism of HMM works and secondly, to provide synthetic databases that can be used, for analysis and inference, for HMMs when real data are difficult to obtain. In addition, some literature have been only concentrated on generating a sequence of symbols from non-parametric HMMs, see for instance Rabiner (1989); Bishop (2006), in which each symbol is simulated according to an *emission probabilities* matrix.

We explain the data generating mechanism from a parametric HMM as follows. Consider a sequence of observations, $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$, generated from a parametric distribution with parameter θ, underlying to the sequence of hidden states $\mathbf{z} = (\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T)$, as

$$\mathrm{y}_t \sim \mathrm{y}_t | \theta_{\mathrm{z}_t = k},$$

where $\mathrm{y}_t | \theta_{\mathrm{z}_t = k}$ represents the distribution of observation y_t at time t, given the parameter θ characterized by the underlying state k at time t. As the states that emit the observations are assumed to be unobserved in reality and are hence unknown, appending those states in the generating of observations here is merely for illustration of how an observation is emitted from a hidden behavior of some phenomenon. Given a HMM with parameters $\Theta = (\pi, \mathbf{A}, \theta)$ and K hidden states, we first choose an initial hidden variable at $t = 1$, z_1, with probability governed by the parameter π_k and then sample the corresponding observation y_1. After that, we choose the next hidden state variable z_2 according to the transition probability $Pr(\mathrm{z}_2|\mathrm{z}_1)$ using the value that has already determined to z_1. Thus, suppose that the sample for z_1 corresponds to state j. Then we choose the state k of z_2 with probabilities a_{jk} for $k = 1, ..., K$. Once z_2 is known, one can draw a sample for y_2, sample the next hidden variable z_3 and so on. We developed the algorithm provided in Rabiner (1989) by replacing the emission probability matrix by the state-specific distribution θ_k. Algorithm (5) illustrates the generating process of a sequence of observations from a parametric HMM. This algorithm illustrates the dependence structure in Figure (3.1) between the states, and states and observations. In this case, we can see that the distribution functions of the observed process $\mathbf{Y}_t$ are not deterministic functions,

Algorithm 5 : Generate observations sequence from a parametric HMM

```
1. Suppose initial parameters of a HMM, $\Theta = (\pi, \mathbf{A}, \theta)$.
2. Set $t$=1.
      a. Choose an initial state, $z_1$, based on the initial state distribution $\pi$.
      b. Generate $y_1$ based on state $z_1$.
3. For $t = 2, 3, ..., T$.
      a. Choose a next state, $z_t$, based on row $a_{\{z_{t-1}, \cdot\}}$.
      b. Generate an observation, $y_t$, based on the current state, $z_t$.
      c. Increment $t$.
      d. If $t < T$, return to step 3 , otherwise stop.
```

but rather probability density functions. In other words, when $y_t|\theta_k$ is a deterministic one-to-one function, mapping states $\mathbf{Z}_t$ into observations $\mathbf{Y}_t$, the process $\mathbf{Z}_t$ becomes observed rather than hidden, and hence the model reduces from a HMM to a Markov model (Visser, 2011).

Figures (3.2 - 3.5) display a sequence of length $T = 200$ of the observations and states generated from a 2- and 3- Normal and Poisson HMMs, respectively, given the following parameterization:

$$y_t \sim N(\Theta_2);\ \Theta_2 = \left(\pi = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}, \mu = \begin{bmatrix} 10 \\ 20 \end{bmatrix}, \sigma^2 \begin{bmatrix} 0.7 \\ 1.6 \end{bmatrix}\right),$$

$$y_t \sim N(\Theta_3);\ \Theta_3 = \left(\pi = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.2 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}, \mu = \begin{bmatrix} 5 \\ 10 \\ 20 \end{bmatrix}, \sigma^2 \begin{bmatrix} 1 \\ 0.7 \\ 1.6 \end{bmatrix}\right),$$

$$y_t \sim Poi(\Theta_2);\ \Theta_2 = \left(\pi = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0.6 & 0.4 \\ 0.1 & 0.9 \end{bmatrix}, \lambda = \begin{bmatrix} 1.5 \\ 4 \end{bmatrix}\right),$$

$$y_t \sim Poi(\Theta_3);\ \Theta_3 = \left(\pi = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.2 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}, \lambda = \begin{bmatrix} 1.5 \\ 4 \\ 9 \end{bmatrix}\right).$$

Note that the observations generated from the Poisson HMM were plotted by a stepped line to refer to discrete observations.

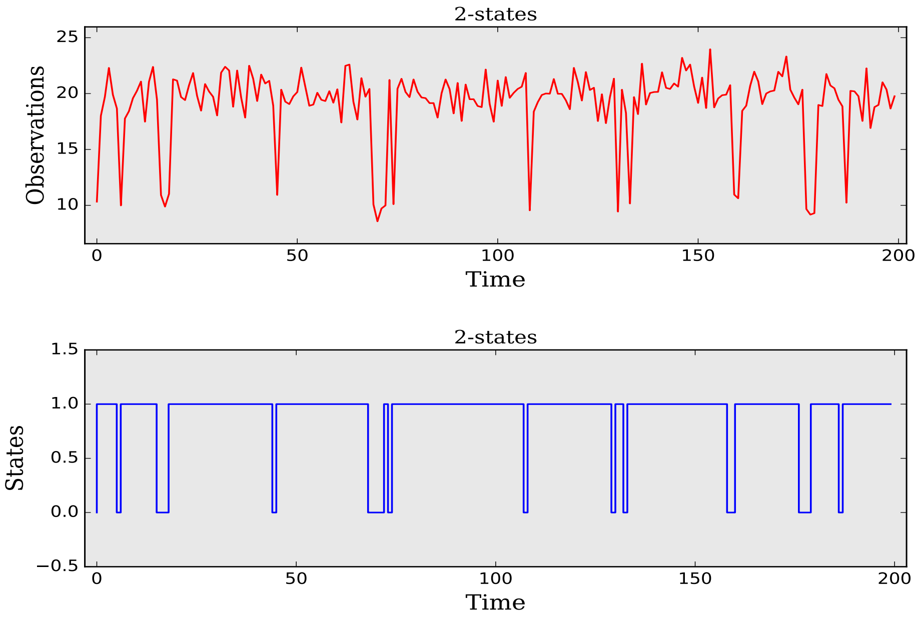


Figure 3.2: A sequence of observations, of length ($T = 200$), generated from 2-state Normal HMM in the top and the corresponding sequence of hidden states with the same length in the bottom.

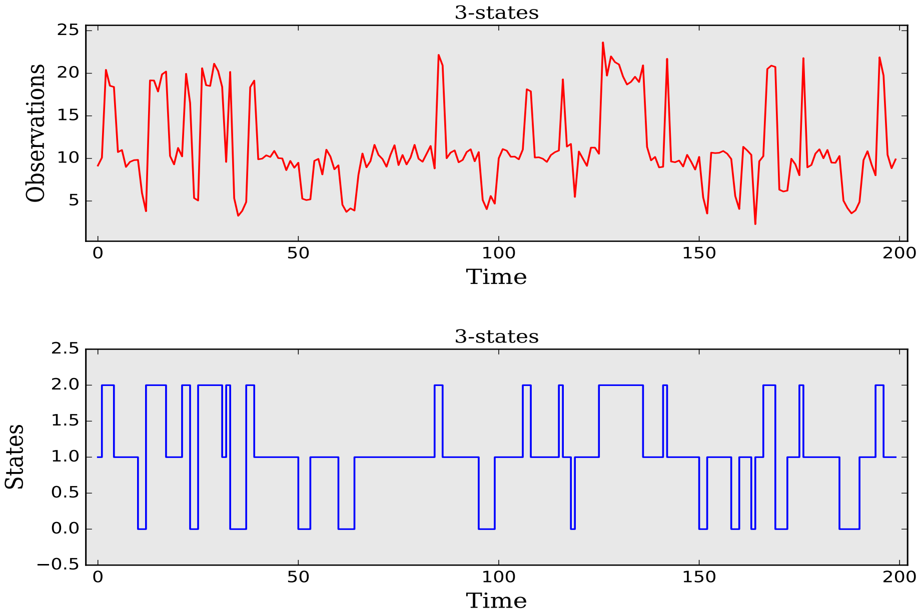


Figure 3.3: A sequence of observations, of length ($T = 200$), generated from 3-state Normal HMM in the top and the corresponding sequence of hidden states with the same length in the bottom.

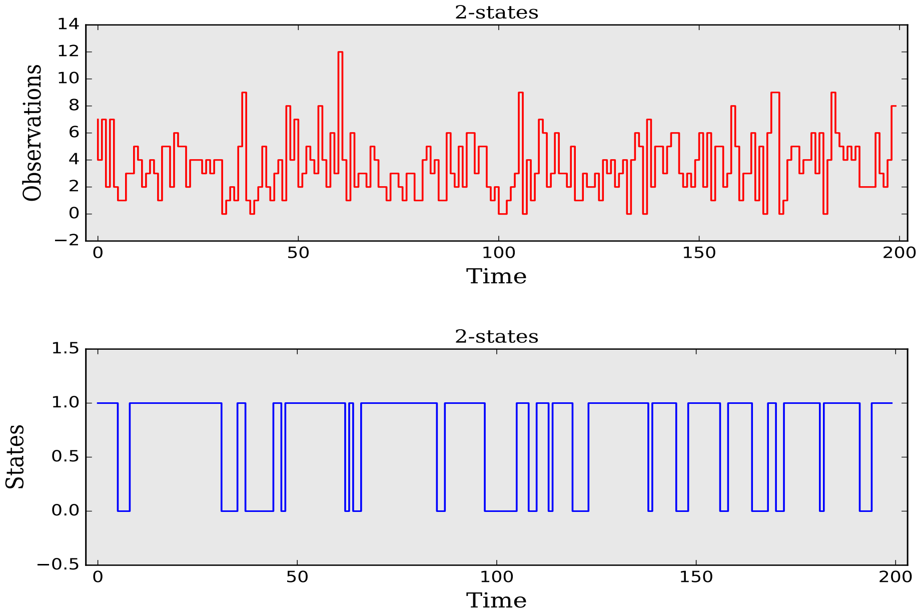


Figure 3.4: A sequence of observations, of length ($T = 200$), generated from 2-state Poisson HMM in the top and the corresponding sequence of hidden states with the same length in the bottom.

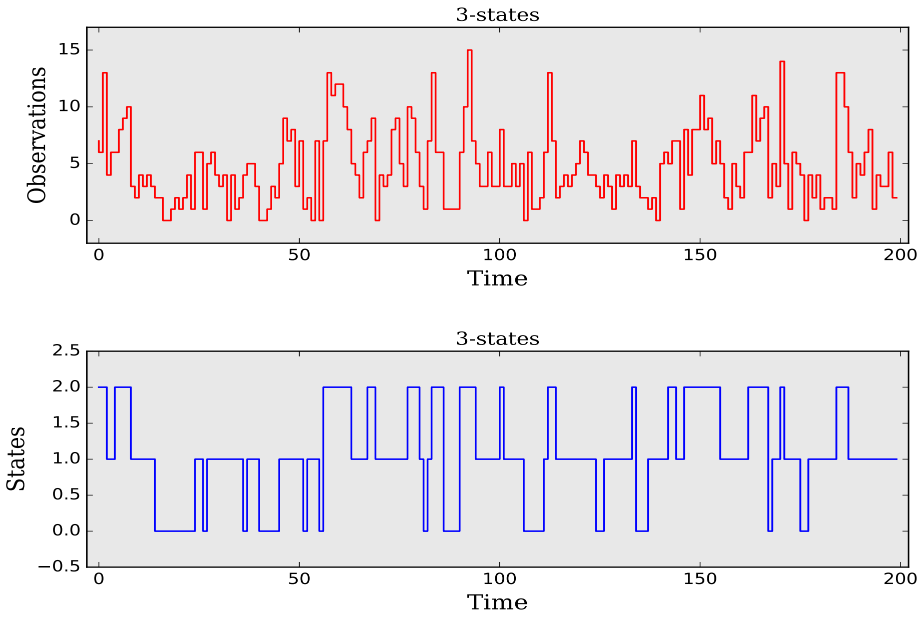


Figure 3.5: A sequence of observations, of length ($T = 200$), generated from 3-state Poisson HMM in the top and the corresponding sequence of hidden states with the same length in the bottom.

3.5 Problems in HMMs

It has been said that HMMs are associated with three basic problems (Rabiner, 1989). Solving these problems effectively means that we can obtain an adequate model to represent sequences of observations.

- **The evaluation problem**: Given a sequence of observations and a model, what is the probability that this observation sequence was produced by that model?

- **The decoding problem**: Given a sequence of observations and a model, what is the most likely state sequence that can meaningfully explain the observations?

- **The estimation problem**: Given a sequence of observations and a model, what should the model parameters be so that the model has a high probability of generating the observations?

The first problem requires that we obtain a solution for the likelihood function of a HMM. The last two problems concerning the estimation of hidden states, $\mathbf{z}$, and all model parameters, $\Theta = (\pi, \mathbf{A}, \theta)$, will be further developed in Chapter 4.

3.6 Likelihood function for the hidden Markov model

In this section we concentrate on the evaluation of the likelihood function of a HMM. Given a sequence of observations, $\mathbf{y} = (y_1, y_2, ...y_T)$, generated from a HMM with parameters $\Theta = (\pi, \mathbf{A}, \theta)$, the probability of this sequence of observations can be expressed as the observed likelihood function, $L(\Theta; \mathbf{y})$. The calculation of $L(\Theta; \mathbf{y})$, or so-called the evaluation problem for a HMM is fairly difficult task (Bishop, 2006, pp. 616-634). A solution for this problem can be achieved via so-called the *augmentation data* strategy (Tanner and Wong, 1987). The concept behind this strategy involves expanding the parameter space by adding auxiliary or "missing data". The HMMs can be viewed as a missing data problem. Hence, the technique of augmentation data can facilitate the evaluation process of the likelihood of HMMs. This can be made via summing or integrating out the missing data so that the likelihood function becomes tractable. Assume that we extent the parameter space Θ by adding a sequence of missing data (hidden states), the likelihood function of a HMM can be then given as

$$\begin{aligned} L(\Theta; \mathbf{y}) &\propto Pr(\mathbf{y}|\Theta) \\ &= \sum_{\forall \mathbf{z}} Pr(\mathbf{y}, \mathbf{z}|\Theta) \\ &= \sum_{\forall \mathbf{z}} Pr(\mathbf{y}|\mathbf{z}, \Theta) Pr(\mathbf{z}|\Theta) \\ &= \sum_{\forall \mathbf{z}} Pr(\mathbf{y}|\mathbf{z}, \theta) Pr(\mathbf{z}|\pi, \mathbf{A}) \\ &= \sum_{\forall \mathbf{z}} L_c(\Theta, \mathbf{z}|\mathbf{y}) \end{aligned} \tag{3.8}$$

where the term $L_c(\Theta, \mathbf{z}|\mathbf{y})$ in Equation (3.8) is called the *complete data likelihood*, whereas the first term, $Pr(\mathbf{y}|\mathbf{z}, \Theta)$, represents the *conditional likelihood* multiplied by the probability density of hidden states, $Pr(\mathbf{z}|\Theta)$. Note that the complete data likelihood, $L_c(\Theta, \mathbf{z}|\mathbf{y})$, can be written in a way so that the observed part can be distinguished from the hidden part of HMM. Note also that we write the observed part as $Pr(\mathbf{y}|\mathbf{z}, \theta)$ due to the fact that the observed process is only inferred directly by the state-specific parameter, θ, given hidden states $\mathbf{z}$ which it is in turn only inferred by the parameters of the hidden part, π and $\mathbf{A}$. Therefore, it is convenient to separate between these two kind of parameters, i.e., the parameter of the observed part θ and the parameters of unobserved part π and $\mathbf{A}$ ($Pr(\mathbf{z}|\pi, \mathbf{A})$). According to the definition of HMM, we can define separately the hidden part, $Pr(\mathbf{z}|\pi, \mathbf{A})$, as

$$\begin{aligned} Pr(\mathbf{z}|\pi,\mathbf{A}) &= Pr(\mathrm{z}_t, \mathrm{z}_{t-1}, ..., \mathrm{z}_1; \mathbf{A}, \pi) \\ &= Pr(\mathrm{z}_t|\mathrm{z}_{t-1}, \mathrm{z}_{t-2}, ..., \mathrm{z}_1; \mathbf{A}) Pr(\mathrm{z}_{t-1}|\mathrm{z}_{t-2}, ..., \mathrm{z}_1; \mathbf{A}), ... Pr(\mathrm{z}_1|\mathrm{z}_0; \pi). \end{aligned}$$

Based on the Markov property, we obtain

$$\begin{aligned} Pr(\mathbf{z}|\pi,\mathbf{A}) &= Pr(\mathrm{z}_t|\mathrm{z}_{t-1}; \mathbf{A}) Pr(\mathrm{z}_{t-1}|\mathrm{z}_{t-2}; \mathbf{A}), ... Pr(\mathrm{z}_2|\mathrm{z}_1; \mathbf{A}) Pr(\mathrm{z}_1|\mathrm{z}_0; \pi) \\ &= Pr(\mathrm{z}_1|\pi) \prod_{t=2}^{T} Pr(\mathrm{z}_t|\mathrm{z}_{t-1}; \mathbf{A}). \end{aligned} \tag{3.9}$$

The observed process, given hidden states, can be given as

$$\begin{aligned} Pr(\mathbf{y}|\mathbf{z}, \theta) &= Pr(\mathrm{y}_1|\theta_{\mathrm{z}_1}) \times Pr(\mathrm{y}_2|\theta_{\mathrm{z}_2}) \times ... \times Pr(\mathrm{y}_T|\theta_{\mathrm{z}_T}) \\ &= \prod_{t=1}^{T} Pr(\mathrm{y}_t|\theta_{\mathrm{z}_t}). \end{aligned} \tag{3.10}$$

As mentioned earlier that the observed process can follow a parametric distribution. For convenience, we will refer to the state-depended observed process in general as $f_{\mathrm{z}_t}(\mathrm{y}|\theta_{\mathrm{z}_t})$, to distinguish from the unobserved process. The complete data likelihood can be then given as

$$L_c(\Theta; \mathbf{z}, \mathbf{y}) = Pr(\mathrm{z}_1|\pi) \prod_{t=2}^{T} Pr(\mathrm{z}_t|\mathrm{z}_{t-1}; \mathbf{A}) \prod_{t=1}^{T} f_{\mathrm{z}_t}(\mathrm{y}_t|\theta_{\mathrm{z}_t}). \tag{3.11}$$

The *observed data likelihood* of a HMM can then be obtained by summing all possible hidden states in the complete data likelihood given in Equation (3.11), i.e.,

$$L(\Theta; \mathbf{y}) = \sum_{\forall \mathbf{z}} L_c(\Theta; \mathbf{z}, \mathbf{y}) = \sum_{\forall \mathbf{z}} \left[p(\mathrm{z}_1|\pi) \prod_{t=2}^{T} p(\mathrm{z}_t|\mathrm{z}_{t-1}, \mathbf{A}) \prod_{t=1}^{T} f_{\mathrm{z}_t}(\mathrm{y}|\theta_{\mathrm{z}_t}) \right]. \tag{3.12}$$

A solution to the likelihood in Equation (3.12) is analytically infeasible or computationally expensive, as it requires the summing over high-dimensional sequences of hidden states $\mathbf{z}$ (Rabiner, 1989). In other words, it involves a total of $2TK^T$ calculations, since at every $t = 1, 2, ..., T$, there are K possible states (i.e. K^T possible state sequences) which can be reached, and for each such state sequence about $2T$ calculations are required for each term in the sum of Equation (3.12). Hence, the calculation is impractical even for small values of K and T. For example, for a model with K=5 states and T=100 observations, it takes $2 \times 100 \times 5^{100} \approx 10^{72}$ computational operations. Therefore, the calculation of likelihood,

$L(\Theta;\mathbf{y})$, for a HMM requires more efficient methods. The high-dimensional summations can be readily computed with the minimum number of computational operations by using the so-called the *forward recursion* proposed by Rabiner (1989). The forward recursion is part of a full algorithm, the *forward-backward algorithm*. Next, we briefly review the forward-backward algorithm and mainly concentrate on the forward recursion to evaluate the likelihood function.

3.6.1 Forward-Backward algorithm

The forward-backward, or Baum-Welch algorithm (Baum, 1972) is a standard algorithm used for HMMs training which is considered a special case of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) developed to calculate the likelihood function. This algorithm is based on two steps. In the first step, named the forward recursion, the algorithm computes a set of forward probabilities which give the probability of ending up in any particular state, given a partial observation sequence, i.e. $Pr(\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_t,\mathrm{z}_t|\Theta)$. In the second step, the algorithm computes a set of backward probabilities which provide the probability of observing the remaining observations, given any starting point, i.e. $Pr(\mathrm{y}_{t+1},\mathrm{y}_{t+2},...,\mathrm{y}_T|\mathrm{z}_t,\Theta)$. Given a sequence of observations, $\mathbf{y} = (\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_T)$, the parameters of the model, $\Theta = (\pi,\mathbf{A},\theta)$, and a sequence of hidden states $\mathbf{z} = (\mathrm{z}_1,\mathrm{z}_2,...,\mathrm{z}_T)$, the forward variable $\alpha_t(j)$ is defined as

$$\alpha_t(j) = Pr(\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_t,\mathrm{z}_t = j|\pi,\mathbf{A},\theta). \tag{3.13}$$

Recursively, the $\alpha_t(j)$ can be computed as follows:

1. Initialization:

$$\alpha_1(j) = \pi_j f_j(\mathrm{y}_1|\theta_j); \qquad 1 \le j \le K. \tag{3.14}$$

2. Induction:

$$\begin{aligned}\alpha_t(k) &= Pr(\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_t,\mathrm{z}_t = j|\pi,\mathbf{A},\theta)\\ &= \sum_{j=1}^{K}\left[\alpha_{t-1}(j)a_{jk}\right] f_k(\mathrm{y}_t|\theta_k); \quad t = 2,3,...,T, \quad 1 \le k \le K.\end{aligned} \tag{3.15}$$

3. Termination:

$$L(\Theta;\mathbf{y}) = \sum_{j=1}^{K} \alpha_T(j). \tag{3.16}$$

Step (1) initializes the forward probabilities as the product of the initial state probability of state j and the observation probability of y_1 at state j. The induction step (2) computes $\alpha_t(k)$ by summing over all K possible states j at time $t-1$ that are reachable to state k at time t via state transitions, and then multiplying the resulting sum by the observation probability $f_k(\mathrm{y}_t|\theta_k)$. The termination step (3) gives the desired computation of $L(\Theta;\mathbf{y})$ as the sum of terminal forward variables $\alpha_T(j)$. Note that the resulting calculations from the forward algorithm have computational complexity of $O(K^2T)$. Figure (3.6) shows the computational operations required for computing the forward variables.

On the other hand, the backward variable, denoted as $\beta_t(j)$, can be defined as the probability of partial observation sequence from $t+1$ to the end, given the state j at time t and the model parameters, Θ, (Rabiner, 1989):

$$\beta_t(j) = Pr(\mathrm{y}_{t+1}, \mathrm{y}_{t+2}, ..., \mathrm{y}_T, |\mathrm{z}_t = j, \pi, \mathbf{A}, \theta). \tag{3.17}$$

The backward variable $\beta_t(j)$ can recursively be computed as follows:

1. Initialization:

$$\beta_T(j) = 1; \quad 1 \leq j \leq K. \tag{3.18}$$

2. Induction:

$$\begin{aligned} \beta_t(j) &= Pr(\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T|\mathrm{z}_t = j, \pi, \mathbf{A}, \theta), \\ &= \sum_{k=1}^{K} a_{jk} f_k(\mathrm{y}_{t+1}|\theta_k)\beta_{t+1}(k); \quad t = T-1, T-2, ..., 1, \quad 1 \leq j \leq K. \end{aligned} \tag{3.19}$$

3. Termination:

$$\begin{aligned} L(\Theta;\mathbf{y}) &= \sum_{j=1}^{K} Pr(\mathrm{z}=j,\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_T|\pi,\mathbf{A},\theta), \\ &= \sum_{j=1}^{K} \pi_j f_j(\mathrm{y}_1|\theta_j).Pr(\mathrm{y}_2,\mathrm{y}_3,...,\mathrm{y}_T|\mathrm{z}=j,\pi,\mathbf{A},\theta), \\ &= \sum_{j=1}^{K} \pi_j f_j(\mathrm{y}_1|\theta_j)\beta_1(j). \end{aligned} \tag{3.20}$$

In the initialization step (1), all the backward probabilities arbitrarily initialize to 1 at $t=T$. The induction step (2) shows that in order to have been in the state j at time t, and to account for the observation sequence from time $t+1$ on, one has to consider all possible states k at time $t+1$, accounting for the transition from j to k (the a_{jk} term), as well as the observation y_{t+1} in the state k (the $f_k(\mathrm{y}_{t+1}|\theta_j)$ term). Then we need to account for the remaining partial observations sequence from state j (the $\beta_{t+1}(j)$ term). The termination step (3) gives the desired computation of $L(\Theta;\mathbf{y})$ as the summation of multiplying the initial backward variables at time $t=1$, $\beta_1(j)$, by the quantity $\pi_j f_j(\mathrm{y}_1|\theta_j)$. Note that the resulting calculations from the backward algorithm also have computational complexity equal to $O(K^2T)$. Figure (3.7) shows the computational operations required for computing the backward variables.

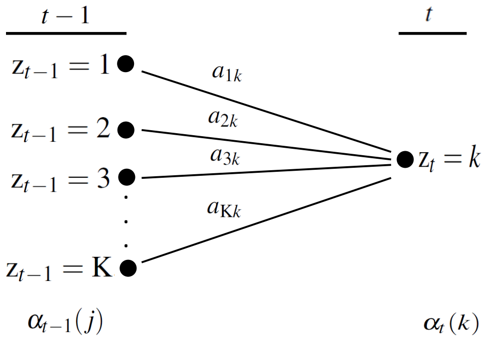


Figure 3.6: The computational operations required for computing the forward variable $\alpha_t(j)$.

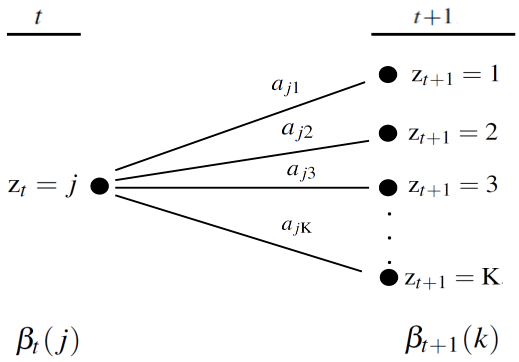


Figure 3.7: The computational operations required for computing the backward variable $\beta_t(j)$.

3.6.2 Scaling procedure

As pointed out by Rabiner (1989), the α and β require to be scaled in the case of the long sequences of observations since the product of probabilities quickly tends to zero, resulting in underflow issues. For example, with size $T > 100$ the probability that sequence will exceed the precision range of essentially any machine even in double precision (Rabiner, 1989). Beginning with the forward variables, the scaling technique involves the multiplication of the forward variable $\alpha_t(j)$ by a scaling factor c_t at each time index t. Thus, the recursion used to compute the probability of forward variables in Equations (3.14) and (3.15) is replaced by a recursion of scaled variables. Here the unscaled forward variable is denoted by $\alpha_t(j)$ and $\hat{\alpha}_t(j)$ denotes the scaled variables, that would be proportional to $\alpha_t(j)$ and sum to 1 over all possible states. That is

$$\sum_{j=1}^{K} \hat{\alpha}_t(j) = 1.$$

From initialization (Equation (3.14)), we have

$$\alpha_1(j) = \pi_j f_j(\mathrm{y}_1|\theta_j), \quad j = 1,2,...,K, \tag{3.21}$$

and the scaling factor c_t can be defined as

$$c_1 = \frac{1}{\sum_{j=1}^{K} \alpha_1(j)} \tag{3.22}$$

$$\hat{\alpha}_1(j) = \frac{\alpha_1(j)}{\sum_{j=1}^{K} \alpha_1(j)} = c_1 \alpha_1(j), \quad j = 1, 2, ..., K. \tag{3.23}$$

$$\alpha_t(k) = \left[\sum_{j=1}^{K} \hat{\alpha}_{t-1}(j) a_{jk} \right] f_k(\mathrm{y}_t | \theta_j), \;\; t = 2, 3, ..., T, \; k = 1, 2, ..., K. \tag{3.24}$$

$$c_t = \frac{1}{\sum_{j=1}^{K} \alpha_t(j)}, \tag{3.25}$$

$$\hat{\alpha}_t(j) = c_t \alpha_t(j), \quad j = 1, 2, ..., K, \tag{3.26}$$

and by induction we obtain

$$\hat{\alpha}_t(j) = c_1 c_2 ... c_t \alpha_t(j). \tag{3.27}$$

For $t = T$, and taking the sum over states gives:

$$\sum_{j=1}^{K} \hat{\alpha}_T(j) = \left[\prod_{t=1}^{T} c_t \right] \sum_{j=1}^{K} \alpha_T(j). \tag{3.28}$$

However, $\sum_{i=1}^{K} \hat{\alpha}_T(j) = 1$ according to the definition of the scaling coefficients, and $\sum_{j=1}^{K} \alpha_T(j) = L(\Theta; \mathbf{y})$ according to Equation (3.16). Thus, we have

$$\prod_{t=1}^{T} c_t L(\Theta; \mathbf{y}) = 1. \tag{3.29}$$

As a result, the likelihood can be written as

$$L(\Theta; \mathbf{y}) = \frac{1}{\prod_{t=1}^{T} c_t}. \tag{3.30}$$

By taking the logarithm of Equation (3.30), we obtain

$$\ell_{\text{rec}}(\Theta|\mathbf{y}) = \log L(\Theta;\mathbf{y}) = -\sum_{t=1}^{T} \log c_t, \tag{3.31}$$

where $\ell_{\text{rec}}(\Theta|\mathbf{y})$ represents the recursive log-likelihood function.

Rabiner (1989) uses the same scaling factors, c_t, used with forward variables to define the scaled backward variables, i.e. the induction step of the recursion of backward variables mentioned in the Equation (3.19) is given as

$$\hat{\beta}_t(j) = c_t\beta_t(j) = \frac{\beta_t(j)}{\sum_{j=1}^{K}\alpha_t(j)}, \tag{3.32}$$

where $\hat{\beta}_t(j)$ represents the scaled backward variable and $c_t = \frac{1}{\sum_{j=1}^{K}\alpha_t(j)}$, represents the scaling factor used with the forward variables. However, Equation (3.32) gives values outside the probability range of the definition of the scaled variables (i.e. $\sum_{j=1}^{K}\hat{\beta}_t(j) \neq 1, \forall t \in T$). Rabiner (1989) justifies the use of the same of scaling factors to scale the backward variables that it is an effective way of keeping the computation within reasonable bounds. Rahimi (2000) confirmed that this the equation is not correct and used an independent scaling factor, D_t, that can be obtained from the backward values itself. We also denote the scaled backward variable as $\hat{\beta}$. We can then define the scaling factor concerning the backward variables as

$$D_t = \frac{1}{\sum_{j=1}^{K}\beta_t(j)}, \tag{3.33}$$

where

$$\begin{aligned}
\hat{\beta}_T(j) &= \frac{\beta_T(j)}{\sum_{j=1}^{K}\beta_T(j)} \\
\beta_t(j) &= \sum_{k=1}^{K} a_{jk} f_k(y_{t+1}|\theta_k)\hat{\beta}_{t+1}(k) \\
\hat{\beta}_t(j) &= D_t\beta_t(j) = \frac{\beta_t(j)}{\sum_{j=1}^{K}\beta_t(j)}.
\end{aligned} \tag{3.34}$$

3.7 Maximum likelihood estimation

One of common methods for estimating the HMM parameters is the Maximum Likelihood estimation (MLE) using the Expectation Maximization algorithm (EM). In general, given a sequence of observations, $\mathbf{y}$, generated from some model with likelihood function $L(\Theta;\mathbf{y})$, the

ML estimator of the model parameter Θ can be given as

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}}\, L(\Theta;\mathbf{y}). \tag{3.35}$$

3.7.1 HMM parameter estimation using the EM algorithm

In the case of HMMs, the EM algorithm is called as the Baum-Welch algorithm (Baum et al., 1970). The Baum-Welch algorithm is the first version of the EM algorithm used in HMMs. It was published by Baum et al. (1970) before it was generalised by Dempster et al. (1977). Baum et al. (1970) and his co-workers investigated the asymptotic properties of EM algorithm with respect to HMMs. It is used for maximizing the likelihood function when some data is missing which corresponds to the hidden states in the context of HMMs.

This is based on two main steps, namely, the E-step and M-step. The E-step includes the computation of conditional expectation of the hidden states $\mathbf{z}$, given the data $\mathbf{y}$ and the model parameters Θ. This step is done by using the forward-backward algorithm. The M-step maximizes the expectation of the logarithm of the complete data likelihood function, given the data, $\mathbf{y}$, and the expected states.

Before formulating the EM algorithm, we need to define first the complete data log-likelihood function, denoted as $\ell_c(\Theta)$, by taking the logarithm of compete data likelihood in Equation (3.11):

$$\ell_c(\Theta) = \log L_c(\Theta;\mathbf{z},\mathbf{y}) = \log\left[\pi_{z_1}\prod_{t=2}^{T} a_{z_t|z_{t-1}}\prod_{t=1}^{T} f(y_t;\theta_{z_t})\right]. \tag{3.36}$$

The EM procedure considers the HMMs as a missing data problem. That means, the hidden states of Markov chain, $z_1, z_2, ..., z_T$, are treated as missing data. So, we need to replace them by indicator variables defined as

$$\gamma_j(t) = \begin{cases} 1 & \text{if } z_t = j \\ 0 & \text{if } z_t \neq j \end{cases} \tag{3.37}$$

$$\xi_{jk}(t) = \begin{cases} 1 & \text{if } z_{t-1} = j \text{ and } z_t = k \\ 0 & \text{if } z_{t-1} \neq j \text{ or } z_t \neq k \end{cases} \tag{3.38}$$

Consequently, the complete data log-likelihood function can be written as

$$\begin{aligned}\ell_c(\Theta) &= \log\left[\prod_{j=1}^{K}\pi_j^{\gamma_j(1)}\prod_{t=2}^{T}\prod_{j=1}^{K}\prod_{k=1}^{K}a_{jk}^{\xi_{jk}(t)}\prod_{t=1}^{T}\prod_{j=1}^{K}(f(\mathrm{y}_t;\theta_j))^{\gamma_j(t)}\right]\\ &= \sum_{j=1}^{K}\gamma_j(1)\log\pi_j+\sum_{t=2}^{T}\sum_{j=1}^{K}\sum_{k=1}^{K}\xi_{jk}(t)\log a_{jk}+\sum_{t=1}^{T}\sum_{k=1}^{K}\gamma_j(t)\log f(\mathrm{y}_t;\theta_j).\end{aligned}\tag{3.39}$$

According to the E-step, the conditional expectation is taken for the complete data log-likelihood function $\ell_c(\Theta)$, given the observations, $\mathbf{y}$, and the current estimate of the model parameter, $\Theta^{(m)}$. Thus, we obtain

$$\begin{aligned}Q(\Theta,\Theta^{(m)}) &= E\left[\ell_c(\Theta)|\mathbf{y},\Theta^{(m)}\right]\\ &= \sum_{j=1}^{K}E[\gamma_j(1)|\mathbf{y},\Theta^{(m)}]\log\pi_j+\sum_{t=2}^{T}\sum_{j=1}^{K}\sum_{k=1}^{K}E[\xi_{jk}(t)|\mathbf{y},\Theta^{(m)}]\log a_{jk}\\ &+\sum_{t=1}^{T}\sum_{k=1}^{K}E[\gamma_j(t)|\mathbf{y},\Theta^{(m)}]\log f(\mathrm{y}_t;\theta_j).\end{aligned}\tag{3.40}$$

Based on the forward-backward recursion, we obtain the expected hidden states as

$$\hat{\gamma}_j(t)=E[\gamma_j(t)|\mathbf{y},\Theta^{(m)}]=Pr(\mathbf{z}_t=j|\mathbf{y},\Theta^{(m)})=\frac{\alpha_t^{(m)}(j)\beta_t^{(m)}(j)}{\sum_{j=1}^{K}\alpha_t^{(m)}(j)\beta_t^{(m)}(j)},\tag{3.41}$$

and

$$\begin{aligned}\hat{\xi}_{jk}(t) &= E[\xi_{jk}(t)|\mathbf{y},\Theta^{(m)}]=Pr(\mathrm{z}_t=j,\mathrm{z}_{t+1}=k|\mathbf{y},\Theta^{(m)})\\ &= \frac{\alpha_t^{(m)}(j)^{(m)}a_{jk}^{(m)}f^{(m)}(\mathrm{y}_{t+1})\beta_{t+1}^{(m)}(j)^{(m)}}{\sum_{j=1}^{K}\sum_{k=1}^{K}\alpha_t^{(m)}(j)^{(m)}a_{jk}^{(m)}f^{(m)}(\mathrm{y}_{t+1})\beta_{t+1}^{(m)}(j)}.\end{aligned}\tag{3.42}$$

Hence, the $Q(\Theta,\Theta^{(m)})$ is given as

$$Q(\Theta,\Theta^{(m)})=\sum_{j=1}^{K}\hat{\gamma}_j^{(m)}(1)\log\pi_j+\sum_{t=2}^{T}\sum_{j=1}^{K}\sum_{k=1}^{K}\hat{\xi}_{jk}^{(m)}(t)\log a_{jk}+\sum_{t=1}^{T}\sum_{k=1}^{K}\hat{\gamma}_j^{(m)}(t)\log f(\mathrm{y}_t;\theta_j).\tag{3.43}$$

The derivation of variables $\gamma_j(t)$ and $\xi_{jk}(t)$ is provided in the Appendix A.

The M-step maximizes $Q(\Theta,\Theta^{(m)})$ with respect to the model parameters Θ, i.e.

$$\Theta^{(m)}=\underset{\Theta}{\operatorname{argmax}}\ Q(\Theta,\Theta^{(m)}).\tag{3.44}$$

Note that Equation (3.43) involves three terms, each of which includes one of the model parameters of interesting; π, $\mathbf{A}$ and θ. In order to obtain an estimate for each parameter of the model, we need to maximize each term separately.

For updating the initial state parameter π, we need to compute the partial derivative of $Q(\Theta, \Theta^{(m)})$ with respect to π_j under the constraint $\sum_{l=1}^{K} \pi_l = 1$:

$$\begin{aligned} 0 &= \frac{\partial}{\partial \pi_j}\left[Q(\Theta, \Theta^{(m)}) - \mathbf{L}\left(\sum_{l=1}^{K} \pi_l - 1\right)\right] \\ &= \frac{1}{\pi_j}.Pr(z_t = 1, \mathbf{y}|\Theta^{(m)}) - \mathbf{L} \end{aligned} \tag{3.45}$$

where $\mathbf{L}$ is the Lagrange multiplier. Multiplying Equation (3.45) by π_j and summing over j, we obtain

$$\mathbf{L} = Pr(\mathbf{y}|\Theta^{(m)}) \propto L(\Theta^{(m)}; \mathbf{y}). \tag{3.46}$$

By inserting Equation (3.46) into Equation (3.45), we can obtain the updated probability for state j at time $t = 1$ as

$$\begin{aligned} 0 &= \frac{1}{\pi_j}.Pr(z_t = j, \mathbf{y}|\Theta^{(m)}) - Pr(\mathbf{y}|\Theta^{(m)}) \\ \pi_j^{(m+1)} &= \frac{Pr(z_t = j, \mathbf{y}|\Theta^{(m)})}{Pr(\mathbf{y}|\Theta^{(m)})} = \hat{\gamma}_j^{(m)}(1). \end{aligned} \tag{3.47}$$

Updating for the transition probabilities, $\mathbf{A} = \{a_{jk}\}$ can be also obtained by taking the partial derivative of $Q(\Theta, \Theta^{(m)})$ with respect to a_{jk} under the constraint $\sum_{l=1}^{K} a_{jl} = 1$:

$$\begin{aligned} 0 &= \frac{\partial}{\partial a_{jk}}\left[Q(\Theta, \Theta^{(m)}) - \mathbf{L}\left(\sum_{l=1}^{K} a_{jl} - 1\right)\right] \\ &= \frac{1}{a_{jk}} \sum_{t=1}^{T-1} Pr(z_t = j, z_{t+1} = k, \mathbf{y}|\Theta^{(m)}) - \mathbf{L} \end{aligned} \tag{3.48}$$

Multiplying Equation (3.48) by a_{jk} and summing over k, we obtain

$$\mathbf{L} = \sum_{t=1}^{T-1} Pr(z_t = j, \mathbf{y}|\Theta^{(m)}). \tag{3.49}$$

By inserting Equation (3.49) into Equation (3.48), we can obtain a new update to the transition probabilities:

$$\begin{aligned} 0 &= \frac{1}{a_{jk}} \sum_{t=2}^{T} Pr(z_t = j, z_{t+1} = k, \mathbf{y}|\Theta^{(m)}) - \sum_{t=1}^{T-1} Pr(z_t = j, \mathbf{y}|\Theta^{(m)}) \\ a_{jk}^{(m+1)} &= \frac{\sum_{t=1}^{T-1} Pr(z_t = j, z_{t+1} = k, \mathbf{y}|\Theta^{(m)})}{\sum_{t=1}^{T-1} Pr(z_t = j, \mathbf{y}|\Theta^{(m)})} \\ &= \frac{\sum_{t=1}^{T-1} \hat{\xi}_{jk}^{(m)}(t)}{\sum_{t=1}^{T-1} \hat{\gamma}_j^{(m)}(t)} \end{aligned} \tag{3.50}$$

The estimate of a_{jk} above is product of dividing the expected number of transitions from state j to state k on the expected number of transitions from state j.

The maximization of third term of the complete-data log-likelihood in Equation (3.40) is based on the nature of the state-dependent distribution θ_j. This requires a solution to the equation:

$$\frac{\partial Q(\Theta, \Theta^{(m)})}{\partial \theta_j} = \frac{\partial}{\partial \theta_j} \left[\sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t) \log f(y_t; \theta_j) \right] = 0, \text{ for } j = 1, 2, ..., K. \tag{3.51}$$

In the case of Normal HMM where

$$f(y_t; \theta_j) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{ -\frac{(y_t - \mu_k)^2}{2\sigma_k^2} \right\}, \ \theta_j = (\mu_j, \sigma_j^2), \tag{3.52}$$

the parameters of the state-specific distribution are updated as

$$\begin{aligned} \frac{\partial Q(\Theta, \Theta^{(m)})}{\partial \theta_j} &= \frac{\partial}{\partial \mu_j \partial \sigma_j^2} \left[\sum_{t=1}^{T} \sum_{j=1}^{K} \hat{\gamma}_j^{(m)}(t) \log \left(\frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left\{ -\frac{(y_t - \mu_j)^2}{2\sigma_j^2} \right\} \right) \right] \\ &= \frac{\partial}{\partial \mu_j \partial \sigma_j^2} \left[\sum_{t=1}^{T} \sum_{j=1}^{K} \hat{\gamma}_j^{(m)}(t) (-\frac{1}{2} \log 2\pi\sigma_j^2 - \frac{1}{2\sigma_j^2}(y_t - \mu_j)^2) \right] = 0 \end{aligned} \tag{3.53}$$

For the mean parameter, we obtain

$$\frac{\partial Q(\Theta, \Theta^{(m)})}{\partial \mu_j} = \sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)(y_t - \mu_j) = 0 \tag{3.54}$$

The yields a new update for the state-dependent mean parameter:

$$\hat{\mu_j}^{(m+1)} = \frac{\sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t) y_t}{\sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)}. \tag{3.55}$$

For the variance parameter

$$\frac{\partial Q(\Theta, \Theta^{(m)})}{\partial \sigma_j^2} = \sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)(-\frac{1}{2\sigma_j^2} + \frac{1}{2(\sigma_j^2)^2}(\mathrm{y}_t - \mu_j)^2) = 0 \tag{3.56}$$

which leads to:

$$\hat{\sigma_j^2}^{(m+1)} = \frac{\sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)(\mathrm{y}_t - \hat{\mu_j}^{(m+1)})^2}{\sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)}. \tag{3.57}$$

In case of a Poisson distribution, where

$$Pr(\mathrm{y}_t; \lambda_j) = \frac{\mathrm{e}^{-\lambda_j} \lambda_j^{\mathrm{y}_t}}{\mathrm{y}_t!}, \tag{3.58}$$

the ME estimate of the state-dependent mean parameters λ_j, can be obtained as:

$$\begin{aligned} \frac{\partial Q(\Theta, \Theta^{(m)})}{\partial \theta_j} &= \frac{\partial}{\partial \lambda_j} \left[\sum_{t=1}^{T} \sum_{j=1}^{K} \hat{\gamma}_j^{(m)}(t) \log \left(\frac{\mathrm{e}^{-\lambda_j} \lambda_j^{\mathrm{y}_t}}{\mathrm{y}_t!} \right) \right] \\ &= \frac{\partial}{\partial \lambda_j} \left[\sum_{t=1}^{T} \sum_{j=1}^{K} \hat{\gamma}_j^{(m)}(t)(\mathrm{y}_t \log \lambda_j - \log(\mathrm{y}_t!) - \lambda_j) \right] = 0 \end{aligned} \tag{3.59}$$

and leads to

$$\frac{\partial Q(\Theta, \Theta^{(m)})}{\partial \lambda_j} = \frac{1}{\lambda_j} \sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)(\mathrm{y}_t - \lambda_j). \tag{3.60}$$

Hence, a new value for the λ_j is obtain as

$$\hat{\lambda_j}^{(m+1)} = \frac{\sum_{t=1}^{T} \hat{\gamma}_j^{(m)}(t)\mathrm{y}_t}{\hat{\gamma}_j^{(m)}(t)}. \tag{3.61}$$

Given the new parameter estimates of the model, we use them again in the E-step and repeat the algorithm until some convergence criterion has been achieved, e.g. until the resulting change in Θ is less than some threshold

$$\parallel \Theta^{(m+1)} - \Theta^{(m)} \parallel < \varepsilon$$

where $\varepsilon > 0$ is a per-specified value and $\parallel . \parallel$ is the Euclidean distance.

The EM algorithm has some limitations. It is slow to converge. Furthermore, it is very sensitive to the starting points as the likelihood values tend to have multiple local maxima, and does not provide any guarantee about the convergence to a global maximum of the likelihood function (Cappé et al., 2005; Zucchini and MacDonald, 2009). It may converge to a global maximum depending on the ways of selection the starting values $\Theta^{(l)}$ (Baum et al., 1970). This requires to choose carefully starting points.

One possible advantage of using the Bayesian theory is the reduction of risk of obtaining spurious modes in cases where the EM algorithm leads to degenerate solutions (Frühwirth-Schnatter, 2006). In Chapter (4), we will introduce Bayesian inference for parameter estimation of HMMs using MCMC algorithms.

3.8 Summary

In this chapter, we have given some definitions and provided notation for Hidden Markov Models. Additionally, we have reviewed some relevant literature on the applications of HMMs.

We have mainly concentrated on the issue of the computation of the likelihood function of a HMM due to its importance for model selection problem that will be introduced later in the Chapter 5. We have illustrated the computation of the likelihood of a HMM based on the forward recursion. Finally, we have reviewed the estimation of HMM parameters using the Baum-Welch algorithm as a spacial case of the Expectation-maximization (EM) algorithm.

Chapter 4

Bayesian Estimation of Hidden Markov Models

4.1 Introduction

The general aim of this chapter is to explain the implementation of Bayesian estimation of Hidden Markov models (HMMs).

In section 4.2, we describe the Bayesian HMM and specify prior and posterior distributions. Section 4.3 demonstrates the estimation of model parameters using the Gibbs sampler. In section 4.4 we present the estimation methods of hidden states. Section 4.5 develops sampling algorithms for parametric HMMs such as the Normal and Poisson HMMs. In section 4.6 we discuss the label switching problem. In section 4.7 we conduct a simulation study to assess the Gibbs sampler. In section 4.8, the sampler is also assessed using real application data involving the waiting time of Old Faithful geyser data.

4.2 The Bayesian HMM

Before implementing Bayesian analysis for HMMs, let us begin by defining the Bayesian model. The posterior distribution for the model parameters according to Bayes Theorem can be written as

$$Pr(\Theta|\mathbf{y}) \propto L(\Theta;\mathbf{y})Pr(\Theta). \tag{4.1}$$

where $L(\Theta;\mathbf{y})$ as defined before is the observed data likelihood. The complexity of this the posterior distribution may preclude the possibility of obtaining a fully analytical solution. Therefore, MCMC methods are used to generate samples from the posterior distribution. The so-called data augmentation procedure (Tanner and Wong, 1987) is often used with MCMC methods in Bayesian analysis of HMMs where extra or auxiliary variables are added to the models in order to facilitate the estimation process of the parameters of the model

(Frühwirth-Schnatter, 2006). In other words, the hidden states are introduced as "missing data" and augmented to the parameter space of the sampler. This allows posterior inference for the model parameters $\mathbf{\Theta}$ to be obtained by averaging over the distribution of the hidden states, $\mathbf{z}$. Thus, the process of sampling from the posterior distribution will be more flexible by writing the posterior in Equation (4.1) as

$$\begin{aligned} Pr(\Theta, \mathbf{z}|\mathbf{y}) &\propto L_c(\Theta; \mathbf{y}, \mathbf{z}) Pr(\Theta), \\ &\propto Pr(\mathbf{y}|\mathbf{z}, \Theta) Pr(\mathbf{z}|\Theta) Pr(\Theta), \end{aligned} \tag{4.2}$$

where the term $L_c(\Theta; \mathbf{y}, \mathbf{z}) = Pr(\mathbf{y}, \mathbf{z}|\Theta)$ represents the complete data likelihood which can be written according to Bayes' rules as $Pr(\mathbf{y}|\mathbf{z}, \Theta)Pr(\mathbf{z}|\Theta)$ and $Pr(\Theta)$ represents a prior distribution on Θ. The joint or complete data posterior above represents a Bayesian model for the HMM that needs to be sampled using a particular MCMC sampler. In the next section, a Bayesian HMM is introduced.

4.2.1 Specification of the model and priors

In order to construct the model, we have to understand the role of each parameter in the model. The parameters of the model $\Theta = (\boldsymbol{\pi}, \mathbf{A}, \theta)$ can be classified into two parts. The first part includes the parameters related to the underlying unobserved process, $\mathbf{z}$, which are the initial and transition probability parameters, $\boldsymbol{\pi}$ and $\mathbf{A}$ respectively. On the other hand, the second part involves the parameters that are related to the observed process, $\mathbf{y}$, which are called the state-dependent parameters, θ, in which these parameters are allocated according to a given hidden state.

According to the Bayesian theory, the model parameters $\boldsymbol{\pi}, \mathbf{A}$, and θ are unknown quantities and need to be estimated. Hence, priors should be specified for these quantities. As explained by Cappé et al. (2005, p.475), we assume that the initial state, z_1, is random and its distribution $\boldsymbol{\pi}$ does not depend on $\mathbf{A}$. In this case, the distribution $\boldsymbol{\pi}$ is unknown and therefore regarded as another parameter about which inference is to be made. A natural choice of prior on the initial state distribution $\boldsymbol{\pi}$ and transition matrix $\mathbf{A}$ is the Dirichlet distribution. The Dirichlet distribution, denoted by $Dir(\delta)$, is a continuous distribution with the density function for the vector $\mathbf{u} = (u_1, u_2, ..., u_K)$ given by

$$f(\mathbf{u}; \delta) = \frac{1}{B(\delta)} \prod_{k=1}^{K} u_k^{\delta_k - 1} \propto \prod_{k=1}^{K} u_k^{\delta_k - 1}, \tag{4.3}$$

subject to $\sum_{k=1}^{K} u_k = 1;\ \ 0 < u_k < 1$, and $\delta_k > 0$. Note that the normalizing constant $B(\delta)$ is a multinomial Beta function and can be expressed in terms of the Gamma function as

$$B(\delta) = \frac{\prod_{j=1}^{K} \Gamma(\delta_j)}{\Gamma(\sum_{j=1}^{K} \delta_j)}, \quad \delta = (\delta_1, \delta_2, ..., \delta_K), \tag{4.4}$$

with a hyperparameter δ. The initial distribution π, given a Dirichlet prior, is

$$Pr(\pi) = \prod_{k=1}^{K} \pi_k \propto \prod_{k=1}^{K} \pi_k^{\delta_k - 1} = Dir(\delta_1, \delta_2, ..., \delta_K), \ \text{ where } \delta_k = 1 \ \forall\, k = 1, 2, ..., K. \tag{4.5}$$

Regarding the transition matrix $\mathbf{A}$, each row of $\mathbf{A}$, $\{a_{j.}\}$, is independently given a Dirichlet prior

$$Pr(\mathbf{A}) = \prod_{j=1}^{K} a_{j.} \propto \prod_{j=1}^{K} a_{j.}^{\delta_j - 1} = Dir(\delta_1, \delta_2, ..., \delta_K); \ \ j,k = 1, 2, \ ..., K. \tag{4.6}$$

The hyper-parameter δ, used either with the initial state distribution or rows of transition matrix, can be considered as a prior number of staying or transiting between states respectively.

Regarding the state-dependent parameter θ, we choose priors for θ, expressed by $Pr(\theta|\varphi)$, to be conjugate, i.e. to have the same parametric distributions for the priors and the likelihood, and hence the posterior. φ refers to collectively as the hyper-parameters governing the shape of the prior distribution of θ. Figure (4.1) shows the parameters of the HMM and all their associated priors and hyper-parameters.

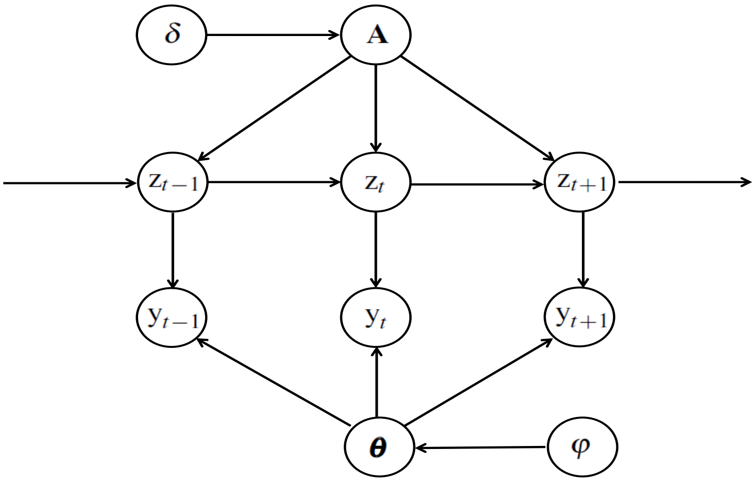


Figure 4.1: A Graphical model of the prior distributions and hyper-parameters relevant to the main parameters of a Bayesian HMM.

According to Bayes' theorem, we can simply specify from Figure (4.1) the joint posterior distribution of hidden states, $\mathbf{z}$, and all parameters of the model, $\Theta = \{\boldsymbol{\pi}, \mathbf{A}, \theta\}$. Given the likelihood function in Equation (4.2), priors and hyper-parameters, and by assuming *a priori* independence between the priors related to the hidden states (i.e. $\boldsymbol{\pi}$, $\mathbf{A}$) and those concerning to observed data, θ, the joint posterior distribution of the HMM can be written as

$$\begin{aligned} Pr(\Theta,\mathbf{z}|\mathbf{y}) &\propto Pr(\mathbf{y}|\mathbf{z},\Theta)Pr(\mathbf{z}|\Theta)Pr(\Theta), \\ &= Pr(\mathbf{y}|\mathbf{z},\theta)Pr(\mathbf{z}|\boldsymbol{\pi},\mathbf{A})Pr(\boldsymbol{\pi})Pr(\mathbf{A})Pr(\theta) \\ &= Pr(\mathbf{y}|\mathbf{z},\theta)Pr(\mathbf{z}|\boldsymbol{\pi},\mathbf{A}) \times Dir(\boldsymbol{\pi}|\delta)\prod_{j=1}^{K} Dir(a_{j.}|\delta)Pr(\theta|\varphi). \end{aligned} \tag{4.7}$$

4.3 Sampling the posterior using MCMC

A full Bayesian analysis can be obtained by sampling the joint posterior distribution (4.7) using a MCMC method called the Gibbs sampling. This requires partitioning the joint posterior into blocks (conditional distributions), and thus the sampling process will be simpler (Geman and Geman, 1984).

The hidden states $\mathbf{z}$ and all parameters Θ of the HMM can be estimated using the Gibbs sampler by sampling from their conditional distributions instead from their joint distribution. Thus, the parameter space is broken into individual blocks and a two-stage Gibbs sampler is introduced as a sampling strategy for HMMs. The Gibbs sampler is then implemented by alternating between drawing $\mathbf{z}$ from the conditional posterior distribution $Pr(\mathbf{z}|\Theta,\mathbf{y})$ (data augmentation) and drawing Θ from the conditional posterior distribution $Pr(\Theta|\mathbf{z},\mathbf{y})$ (Casella and Robert, 2004). The general form of the two-stage Gibbs sampler is outlined in algorithm (6).

Algorithm 6 : Sampling from the joint posterior for $(\Theta,\mathbf{z})$ using the two-stage Gibbs sampling

```
1. Start with initial samples $\mathbf{z}^{(0)}$ and $\Theta^{(0)}$:
2. For $m = 1,2,...,M$ iterations:
    (a) Generate $\mathbf{z}^{(m)} \sim Pr(\mathbf{z}|\mathbf{y},\Theta^{(m-1)})$;
    (b) Generate $\Theta^{(m)} \sim Pr(\Theta|\mathbf{y},\mathbf{z}^{(m)})$.
```

From algorithm (6), it can be seen that sampling from Θ is simplified via integrating the hidden states out, where the hidden states in this stage are treated as missing data

$$\Theta^{(m)} \sim Pr(\Theta|\mathbf{y}) = \sum_{\mathbf{z}} Pr(\Theta,\mathbf{z}|\mathbf{y}). \tag{4.8}$$

As pointed out by Casella and Robert (2004), the joint sampling from $(\Theta^{(m)}, \mathbf{z}^{(m)})$ forms a Markov chain, of which the transition kernel ($\mathbb{K}$) over the joint variables,

$$\mathbb{K}((\Theta^{'},\mathbf{z}^{'})|(\Theta,\mathbf{z})) = Pr(\Theta^{'}|\mathbf{z}^{'},\mathbf{y})Pr(\mathbf{z}^{'}|\Theta,\mathbf{y}), \tag{4.9}$$

has a stationary distribution of $Pr(\Theta,\mathbf{z}|\mathbf{y})$, and sampling from their conditional distributions, i.e. $\Theta^{(m)}$ and $\mathbf{z}^{(m)}$, is also a Markov chain. For example, the sub-chain $\Theta^{(m)}$ can be produced with the transition kernel

$$\mathbb{K}(\Theta^{'}|\Theta) = \sum_{\mathbf{z}} Pr(\Theta^{'}|\mathbf{z}^{'},\mathbf{y})Pr(\mathbf{z}^{'}|\Theta,\mathbf{y}). \tag{4.10}$$

Analogously, the sequence $\mathbf{z}^{(m)}$ is also a Markov chain (Albert and Chib, 1993), with transition kernel density, i.e. the conditional density of $\mathbf{z}^{(m)} = \mathbf{z}^{'}$ given $\mathbf{z}^{(m-1)} = \mathbf{z}$ is,

$$\mathbb{K}(\mathbf{z}^{'}|\mathbf{z}) = \int_{\Theta} Pr(\Theta|\mathbf{z},\mathbf{y})Pr(\mathbf{z}^{'}|\Theta,\mathbf{y})d\Theta. \tag{4.11}$$

A good review for the derivation and properties of the Gibbs sampler can be found in Casella and Robert (2004, Ch. 9).

Particularly, step (b) in algorithm (6) can be broken into several conditional distributions with respect to the HMM parameters, i.e.

$$\begin{aligned} \pi &\sim Pr(\pi|\mathbf{z},\mathbf{y}), \\ \mathbf{A} &\sim Pr(\mathbf{A}|\mathbf{y},\pi,\mathbf{z}), \end{aligned} \tag{4.12}$$

and the state-dependent parameter

$$\theta \sim Pr(\theta|\mathbf{z},\mathbf{y}). \tag{4.13}$$

The hidden states in step (a) in algorithm (6) can be given by

$$\mathbf{z} \sim Pr(\mathbf{z}|\mathbf{y},\pi,\mathbf{A},\theta). \tag{4.14}$$

4.4 State sequence estimation

Sampling from the conditional posterior of hidden states, $Pr(\mathbf{z}|\mathbf{y},\boldsymbol{\pi},\mathbf{A},\boldsymbol{\theta})$, mentioned in Equation (4.14), is often achieved using one of two common methods, namely, the Direct Gibbs (DG) algorithm or *local* updating of hidden states (Robert et al., 1993) and the so-called *global* updating or the forward-backward Gibbs (FBG) algorithm (Chib, 1996). In this regard, Cappé et al. (2005, p.484) commented that:

> *" It is thus difficult to make a firm recommendation on which updating scheme to use. One may start by running local updating, and if its mixing behavior is poor, try global updating as well. "*

The above comment was based on results reported by Robert et al. (1999) who implemented a comparison study to check the convergence of the two methods. They concluded there is no evidence in favour of the global compared to the local method. The only difference, which can be in favor of the DG sampler, is that the FBG sampler uses the forward-backward recursion, which requires longer time than that is consumed in the DG sampler (Robert et al., 1993). We therefore use the DG sampler to sample the hidden states of the model.

4.4.1 Sampling the hidden states using the direct Gibbs sampler

Sampling the hidden states using the direct Gibbs (DG) sampler was firstly proposed by Robert et al. (1993) and it has since been widely used by many authors, for example, Albert and Chib (1993), Chib (1996), Robert and Titterington (1998), Robert et al. (1999).
According to Chib (1996), We describe the sampling hidden states using DG sampler. Consider a sequence of discrete hidden states, $\mathrm{z}_t \in \{1,2,...,K\}$, which evolves according to a Markov chain of first order:

$$\mathrm{z}_t|\mathrm{z}_{t-1} \sim \mathrm{Markov}(\pi_1,\mathbf{A}), \tag{4.15}$$

where $\mathbf{A} = \{a_{jk}\}$ is the one-step transition probability matrix of the chain, and π_1 is the initial state distribution at t. At each observation point t, a realization of the state occurs. Then, given that $\mathrm{z}_t = k$, the observation y_t is drawn from the population given by the conditional density

$$\mathrm{y}_t|\mathbf{y}_{t-1},\theta_k \sim f(\mathrm{y}_t|\mathbf{y}_{t-1},\theta_k); \quad k = 1,2,...,K, \tag{4.16}$$

where $\mathbf{y}_{t-1} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_{t-1})$, $f(.|\mathbf{y}_{t-1}, \theta_k)$ is a density (or mass) function with respect to the finite measure σ, and θ_k is the state-specific parameter of the k^{th} state. Thus, the observation at time t is generated according to a finite mixture distribution

$$f(\mathrm{y}_t|\mathbf{y}_{t-1}, \mathrm{z}_{t-1}, \theta) = \begin{cases} \sum_{k=1}^{K} f(\mathrm{y}_t|\mathbf{y}_{t-1}, \theta_k)\pi_1(\mathrm{z}_1 = k), & t = 1 \\ \sum_{k=1}^{K} f(\mathrm{y}_t|\mathbf{y}_{t-1}, \theta_k)Pr(\mathrm{z}_t = k|\mathrm{z}_{t-1}), & t \geq 2. \end{cases} \tag{4.17}$$

The sampling process is implemented using the sampler by simulating the states one by one from their full conditional distributions. Hence, the joint posterior of hidden states $\mathbf{z}$ in Equation (4.14) can be written as

$$Pr(\mathbf{z}|\mathbf{y}, \theta, \mathbf{A}) = Pr(\mathrm{z}_T|\mathbf{y}, \theta, \mathbf{A}) \times ... \times Pr(\mathrm{z}_2|\mathrm{z}_3, ..., \mathrm{z}_T, \mathbf{y}, \theta, \mathbf{A})Pr(\mathrm{z}_1|\mathrm{z}_2, ..., \mathrm{z}_T, \mathbf{y}, \theta, \mathbf{A}). \tag{4.18}$$

Consequently, the simulation of the hidden state sequence requires the calculation of the probability mass function of each state, i.e., the Gibbs sampler is implemented with T univariate component blocks. In other words, the hidden states are sampled sequentially from their full conditional distributions $Pr(\mathrm{z}_t|\mathbf{z}_{-t}, \mathbf{y}, \Theta)$ for $t = 1, 2, ..., T$, where $\mathbf{z}_{-t} = (\mathrm{z}_1, ..., \mathrm{z}_{t-1}, \mathrm{z}_{t+1}, ..., \mathrm{z}_T)$ denotes the whole state chain of $\mathbf{z}$ except the state z_t. According to the conditional independence and Markov property assumptions shown from the graphical representation of the dependence structure of a HMM in Figure (4.1), the conditional posterior probability distribution $Pr(\mathrm{z}_t|\mathbf{z}_{-t}, \mathbf{y}, \Theta)$ is precisely obtained using:

$$\begin{aligned} Pr(\mathrm{z}_t|\mathbf{z}_{-t}, \mathbf{y}, \Theta) &\propto Pr(\mathrm{z}_t = k|\mathrm{z}_{t+1}, \mathrm{z}_{t-1}, \mathrm{y}_t, \mathbf{A}, \theta) \\ &\propto Pr(\mathrm{z}_t = k|\mathrm{z}_{t-1}, \mathbf{A})Pr(\mathrm{z}_{t+1}|\mathrm{z}_t = k, \mathbf{A})Pr(\mathrm{y}_t|\mathrm{z}_t = k, \theta) \\ &\propto a_{\mathrm{z}_{t-1}k}a_{k\mathrm{z}_{t+1}}f(\mathrm{y}_t|\theta_k), \end{aligned} \tag{4.19}$$

that is, for $2 < t < T$ the full conditional distribution of z_t is

$$Pr(\mathrm{z}_t|..., \mathrm{z}_{t-1}, \mathrm{z}_{t+1}, ..., \mathbf{y}, \Theta) = \frac{a_{\mathrm{z}_{t-1}k}a_{k\mathrm{z}_{t+1}}f(\mathrm{y}_t|\theta_k)}{\sum_{l=1}^{K} a_{\mathrm{z}_{t-1}l}a_{l\mathrm{z}_{t+1}}f(\mathrm{y}_t|\theta_l)}, \tag{4.20}$$

while at time $t = 1$ and $t = T$, the full conditional distributions of state z_1 and the state z_T respectively can be obtained using

$$Pr(\mathrm{z}_1|\mathrm{z}_2, ..., \mathbf{y}, \Theta) = \frac{\pi_k a_{k\mathrm{z}_2}f(\mathrm{y}_1|\theta_k)}{\sum_{l=1}^{K} \pi_l a_{l\mathrm{z}_2}f(\mathrm{y}_1|\theta_l)}, \tag{4.21}$$

and

$$Pr(\mathrm{z}_T|...\mathrm{z}_{T-1},\mathbf{y},\Theta) = \frac{a_{\mathrm{z}_{T-1,k}} f(\mathrm{y}_T|\theta_k)}{\sum_{l=1}^{K} a_{\mathrm{z}_{T-1}l} f(\mathrm{y}_T|\theta_l)}. \tag{4.22}$$

Practically, the DG algorithm is initialized by choosing at random a sequence of hidden states $\mathbf{z}^{(0)}$ with a desired length, T, and initial values for the model parameters, $\Theta^{(0)} = (\boldsymbol{\pi}^{(0)}, \mathbf{A}^{(0)}, \theta^{(0)})$. Then new hidden states $\mathrm{z}_t^{(m)}; m = 1,2,...,M$, are sampled one by one from a nominal distribution, given the allocation probabilities corresponding to each observation point y_t.

We summarize the sampling process of hidden states using the DG sampler in algorithm (7).

Algorithm 7 : Sampling hidden states using the direct method

```
Initialization: Start with initial samples at $m = 0$: $\mathbf{z}^{(0)}, \Theta^{(0)} = (\boldsymbol{\pi}^{(0)}, \mathbf{A}^{(0)}, \theta^{(0)})$:
For $m = 1,2,...,M$ iterations:
    1. For $t = 1,\ k = 1,2,...,K$;
        (a) Compute
            $\mathbb{P}_{k1}^{(m)} = Pr(\mathrm{z}_1^{(m)} = k|\mathrm{z}_2^{(m-1)}, \boldsymbol{\pi}^{(m-1)}, \mathbf{A}^{(m-1)}, \theta^{(m-1)}, \mathbf{y}) \propto \frac{\pi_k^{(m-1)} a_{k\mathrm{z}_2}^{(m-1)} f(\mathrm{y}_1|\theta_k^{(m-1)})}{\sum_{l=1}^{K} \pi_l^{(m-1)} a_{l\mathrm{z}_2}^{(m-1)} f(\mathrm{y}_1|\theta_l^{(m-1)})}$,
        (b) Sample $\mathrm{z}_1^{(m)} \sim \text{Multi}\left\{Pr(\mathrm{z}_1^{(m)} = k)\right\}$.
    2. For $t = 2,3,...,T-1,\ k = 1,2,...,K$;
        (a) Compute
            $\mathbb{P}_{kt}^{(m)} = Pr(\mathrm{z}_t^{(m)} = k|\mathrm{z}_{t+1}^{(m-1)}, \mathrm{z}_{t-1}^{(m)}, \boldsymbol{\pi}^{(m-1)}, \mathbf{A}^{(m-1)}, \theta^{(m-1)}, \mathbf{y}) \propto \frac{a_{\mathrm{z}_{t-1}k}^{(m)} a_{k\mathrm{z}_{t+1}}^{(m-1)} f(\mathrm{y}_t|\theta_k^{(m-1)})}{\sum_{l=1}^{K} a_{\mathrm{z}_{t-1}l}^{(m)} a_{l\mathrm{z}_{t+1}}^{(m-1)} f(\mathrm{y}_t|\theta_l^{(m-1)})}$;
        (b) Sample $\mathrm{z}_t^{(m)} \sim \text{Multi}\left\{Pr(\mathrm{z}_t^{(m)} = k)\right\}$.
    3. For $t = T,\ k = 1,2,...,K$;
        (a) Compute
            $\mathbb{P}_{kT}^{(m)} = Pr(\mathrm{z}_T^{(m)} = k|\mathrm{z}_T^{(m)}, \boldsymbol{\pi}^{(m-1)}, \mathbf{A}^{(m-1)}, \theta^{(m-1)}, \mathbf{y}) \propto \frac{a_{\mathrm{z}_{T-1}k}^{(m)} f(\mathrm{y}_T|\theta_k^{(m-1)})}{\sum_{l=1}^{K} a_{\mathrm{z}_{T-1}l}^{(m)} f(\mathrm{y}_T|\theta_l^{(m-1)})}$;
        (b) Sample $\mathrm{z}_T^{(m)} \sim \text{Multi}\left\{Pr(\mathrm{z}_T^{(m)} = k)\right\}$.
    4. Increment $m$.
```

From above algorithm, the allocation probabilities can be stored in a matrix of dimension $(K \times T)$, called $\mathbb{P}$, where its rows represent the number of states K and its columns represent the length of data, T. The first hidden state, z_1, will be sampled given the first column of the matrix $\mathbb{P}$. For $t = 2,3,...,T-1$, hidden states will be successively sampled one by one given the columns from $t = 2$ until $t = T-1$. Finally the last hidden state z_T, is sampled based on the last column in the matrix of allocation probabilities $\mathbb{P}$.

After an MCMC run, it is possible to obtain a marginal estimate for each hidden state using

$$\hat{P}_r(\mathrm{z}_t = k|\mathbf{y}) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}(\mathrm{z}_t^{(m)} = k); \quad m = 1,2,...,M. \tag{4.23}$$

These probabilities are essentially averages that are obtained by dividing the frequency of each state k over the number of iterations M.

4.4.2 The most likely state sequence

To estimate the optimal hidden state sequence, the maximum a posteriori (MAP) estimator is usually applied. This estimator can be obtained using an effective approach called the Viterbi algorithm (Viterbi, 1967) which is essentially based on the forward-backward computations. This algorithm is based on maximizing the conditional posterior distribution, $Pr(\mathbf{z}|\mathbf{y},\boldsymbol{\pi},\mathbf{A},\theta)$, given in Equation (4.14):

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmax}} Pr(\mathbf{z}|\mathbf{y},\Theta).$$

However, this approach suffers from the same stability issues as the forward-backward recursions. Furthermore, it does not fully take into account the uncertainty in the model parameters. A better method that includes estimating the hidden state sequence marginally over the model parameters:

$$Pr(\mathbf{z}|\mathbf{y}) = \int Pr(\mathbf{z}|\mathbf{y},\Theta)Pr(\Theta)d\Theta \approx \frac{1}{M}\sum_{m=1}^{M} Pr(\mathbf{z}|\mathbf{y},\Theta^{(m)}),$$

where $\Theta^{(m)}; m = 1,2,...,M$, is a sequence of sampled model parameters from Markov chain. Scott (2002) pointed out that such an approach, averaging over Θ, result in destroying the Markov structure of the model. He also added that maximizing $Pr(\mathbf{z}|\mathbf{y})$ rather than $Pr(\mathbf{z}|\mathbf{y},\Theta)$, using Viterbi algorithm, is still difficult. A simple solution is to compute the most frequent state at time t of each of the drawn hidden states sequences $\mathbf{z}^t$ over an MCMC run, where $\mathbf{z}^t = (\mathbf{z}_1^t,...,\mathbf{z}_M^t)$ for $t = 1,2,...,T$.

4.5 Bayesian parametric distributions-based HMMs

In this section we describe HMMs when the state-dependent distribution can take each of the two most widely used exponential family distributions, namely, the Normal and Poisson distributions.

4.5.1 Bayesian Normal HMM

Consider a K-state Normal HMM, that involves conditionally independent Normal variables, $\mathrm{y}_t; t = 1,2,...,T$, whose parameters (μ_k, σ_k^2), depend on a hidden state z_t such that $\mathbf{z} = (\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T)$ is a Markov chain defined on the state-space $\{1,2,...,K\}$. The Markov chain $\mathrm{z}_t;\ t = 1,2,...,T$, is modelled with a transition matrix $\mathbf{A} = \{a_{jk}\}$, and an initial state distribution π_k where $j,k = 1,2,...,K$, are such that

$$\mathrm{z}_t \sim Pr(\pi_j);\ \ t = 1$$
$$\mathrm{z}_t|\mathrm{z}_{t-1} \sim Pr(a_{\mathrm{z}_{t-1}}.);\ \ t = 2,3,...,T,$$

and

$$\mathrm{y}_t|\mathrm{z}_t = k, \mu_k, \sigma_k^2 \sim \phi_k(\mathrm{y}_t|\mu_k, \sigma_k^2),$$

where $\phi_k(\mathrm{y}_t|\mu_k, \sigma_k^2)$ represent the k^{th} state-dependent probability density function of the Normal distribution

$$\phi_k(\mathrm{y}_t|\mu_k, \sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{ -\frac{(\mathrm{y}_t - \mu_k)^2}{2\sigma_k^2} \right\}.$$

To simplify of the notation, we parametrize the Normal distribution using the precision parameter τ rather than the variance, i.e. $\sigma^2 = \frac{1}{\tau}$, such that

$$\phi_k(\mathrm{y}_t|\mu_k, \tau_k^{-1}) = \frac{\sqrt{\tau_k}}{\sqrt{2\pi}} \exp\left\{ -\frac{\tau_k\,(\mathrm{y}_t - \mu_k)^2}{2} \right\}. \tag{4.24}$$

Sampling from the Normal HMM requires specification of priors on the parameters $\Theta = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\tau}^{-1})$. Beginning with the initial state distribution $\boldsymbol{\pi}$ and the matrix of probability transitions $\mathbf{A}$, we assume that the initial state distribution $\boldsymbol{\pi}$ and each row $a_{j.}$ in $\mathbf{A}$, where $\mathbf{A} = \{a_{jk}\};\ j,k = 1,2,...,K$, follows independently a Dirichlet distribution with parameter

$\delta = (\delta_1, \delta_2, ..., \delta_K)$,

$$\begin{aligned} \pi &\sim Dir(\delta_1, \delta_2, ..., \delta_K), \\ \{a_{j\cdot}\} &\sim Dir(\delta_1, \delta_2, ..., \delta_K), \ \text{ for } j = 1, 2, ..., K. \end{aligned} \tag{4.25}$$

For the parameters of the Normal distribution μ and τ^{-1}, we assume a conjugate Normal prior for each μ_k with mean η and variance ζ^{-1}, i.e.

$$\mu_k \sim N(\eta, \zeta^{-1}), \tag{4.26}$$

and a Gamma prior for each precision τ_k^{-1} with parameters κ and ν, i.e.

$$\tau_k^{-1} \sim Gamma(\kappa, \nu), \ \text{independently} \ \forall\, k = 1, 2, ..., K, \tag{4.27}$$

where η, ζ, κ and ν are hyper-parameters. To alleviate the influence on the resulting inference, values or hyper-priors on those hyper-parameters are often given non-informative priors (Cappé et al., 2005, p.474). The priors can then be introduced as

$$Pr(\pi|\delta) \propto \prod_{k=1}^{K} \pi_k^{\delta_k - 1}. \tag{4.28}$$

$$Pr(\mathbf{A}|\delta) \propto \prod_{j=1}^{K} \prod_{k=1}^{K} a_{jk}{}^{\delta_k - 1}. \tag{4.29}$$

$$Pr(\mu_k|\zeta, \eta) \propto \exp\left\{-\frac{\zeta(\mu_k - \eta)^2}{2}\right\}. \tag{4.30}$$

$$Pr(\tau_k|\kappa, \nu) \propto \tau_k^{\kappa - 1} \exp\{-\nu\tau_k\}. \tag{4.31}$$

Given a sequence of hidden states $\mathbf{z}$, the observed likelihood of Normal HMM with K-states is defined as

$$\begin{aligned} L_c(\pi, \mathbf{A}, \mu, \tau; \mathbf{z}, \mathbf{y}) &= \pi_{z_1} \phi_{z_1}(y_1|\mu_{z_1}, \tau_{z_1}) a_{z_1 z_2} \phi_{z_2}(y_2|\mu_{z_2}, \tau_{z_2}), ..., a_{z_{T-1} z_T} \phi_{z_T}(y_T|\mu_{z_T}, \tau_{z_T}) \\ &= \prod_{k=1}^{K} \pi_k^{N_k} \prod_{k=1}^{K} \prod_{j=1}^{K} a_{jk}^{N_{jk}} \prod_{k=1}^{K} \prod_{t:z_t=k}^{T} \phi_k(y_t|\mu_k, \tau_k), \end{aligned} \tag{4.32}$$

where $N_k = \sum_{t=1}^{T} \mathbb{I}_{(z_t=k)}$ denotes the number of transitions from state k at time t and $N_{jk} = \sum_{t=1}^{T-1} \mathbb{I}_{(z_t=k, z_{t-1}=j)}$ denotes the number of transitions from state j at time $t-1$ into the state k at time t. Note that the first two terms in Equation (4.32), $\prod_{k=1}^{K} \pi_k^{N_k}$ and $\prod_{k=1}^{K} \prod_{j=1}^{K} a_{jk}^{N_{jk}}$, are used for making inference about the hidden part of the model, whereas the last, $\prod_{k=1}^{K}\prod_{t:\mathrm{z}_t=k}^{T}\phi_k(\mathrm{y}_t|\mu_k,\tau_k)$, is used for inference about the state-dependent Normal distribution. The joint distribution for all parameters of a Normal HMM can then be written as

$$Pr(\boldsymbol{\pi},\mathbf{A},\mu_k,\tau_k) \propto \prod_{k=1}^{K}\pi_k^{N_k}\prod_{k=1}^{K}\prod_{j=1}^{K}a_{jk}^{N_{jk}}\prod_{k=1}^{K}\prod_{t:\mathrm{z}_t=k}^{T}\sqrt{\tau_k}\exp\left\{-\frac{\tau_k(\mathrm{y}_t-\mu_k)^2}{2}\right\}\times$$
$$\prod_{k=1}^{K}\pi^{\delta_k-1}\prod_{k=1}^{K}\prod_{j=1}^{K}a_{jk}^{\delta_k-1}\times\exp\left\{-\frac{\zeta(\mu_k-\eta)^2}{2}\right\}\times\tau_k^{\kappa-1}\exp\{-\nu\tau_k\}. \quad (4.33)$$

Using the Gibbs sampler, the joint distribution in Equation (4.33) can be decomposed into full conditional posterior distributions, independently for each $k\in\{1,2,...,K\}$, of the parameters of the model as follows

$$Pr(\pi_k|\mathbf{y},\mathbf{z},\mu,\tau) \propto \prod_{k=1}^{K}\pi^{N_k+\delta_k-1} = Dir(N_k+\delta_k), \quad (4.34)$$

$$Pr(\mathbf{A}|\mathbf{y},\mathbf{z},\mu,\tau) \propto \prod_{j=1}^{K}\prod_{k=1}^{K}a_{jk}{}^{N_{jk}+\delta_k-1} = Dir(N_{j.}+\delta_k), \quad (4.35)$$

$$Pr(\mu_k|\mathbf{y},\mathbf{z},\tau,\mathbf{A}) = N\left\{\frac{\eta\zeta+\tau_k\sum_{t:\mathrm{z}_t=k}\mathrm{y}_t}{T_k\tau_k+\eta},\frac{1}{T_k\tau_k+\eta}\right\}, \quad (4.36)$$

where $T_k=\sum_{t=1}^{T}\mathbb{I}_{(\mathrm{z}_t=k)}$ denotes the number of observations generated from the state k and $\sum_{t:\mathrm{z}_t=k}\mathrm{y}_t$ denotes the summation of observations at state k. The full conditional posterior of the precision parameter is

$$Pr(\tau_k|\mathbf{y},\mathbf{z},\mu,\mathbf{A}) = Gamma\left\{\kappa+0.5T_k,\nu+0.5\mathrm{s}^{2(\nu)}\right\}, \quad (4.37)$$

where $\mathrm{s}^{2(\nu)}=\sum_{t=1}^{T}\mathbb{I}_{(\mathrm{z}_t=k)}(\mathrm{y}_t-\mu_k)^2;\quad k=1,2,...,K$. For more details about deriving the full conditional posterior distribution of Normal HMM, see Appendix (A).

By following algorithm (7), a sequence of hidden states $\mathbf{z}$ of a Normal HMM can be locally drawn using the DG sampler as follows

$$Pr(\mathrm{z}_1|\mathrm{z}_2,...,\mathbf{y},\boldsymbol{\pi},\mathbf{A},\mu,\tau^{-1}) = \frac{\pi_k a_{k\mathrm{z}_2}\phi(\mathrm{y}_1|\mu_k,\tau_k^{-1})}{\sum_{l=1}^{K}\pi_l a_{l\mathrm{z}_2}\phi(\mathrm{y}_1|\mu_l,\tau_l^{-1})},\ \text{for } t=1, \quad (4.38)$$

for $2<t<T$ the full conditional distribution of z_t is

$$Pr(\mathrm{z}_t|...,\mathrm{z}_{t-1},\mathrm{z}_{t+1},...,\mathbf{y},\mathbf{A},\mu,\tau^{-1}) = \frac{a_{\mathrm{z}_{t-1}k}a_{k\mathrm{z}_{t+1}}\phi(\mathrm{y}_t|\mu_k,\tau_k^{-1})}{\sum_{l=1}^{K}a_{\mathrm{z}_{t-1}l}a_{l\mathrm{z}_{t+1}}\phi(\mathrm{y}_t|\mu_l,\tau_l^{-1})}, \quad (4.39)$$

and for $t = T$

$$Pr(z_T|...z_{T-1}, \mathbf{y}, \mathbf{A}, \mu, \tau^{-1}) = \frac{a_{z_{T-1}k}\phi(y_T|\mu_k, \tau_k^{-1})}{\sum_{l=1}^{K} a_{z_{T-1}l}\phi(y_T|\mu_l, \tau_l^{-1})}. \tag{4.40}$$

We extended algorithm (6) described in section (4.3) to involve the sampling process of full conditional distributions of the parameters of K-state Normal HMM: π, $\mathbf{A}$, μ, and τ. Algorithm (8) illustrates the full Gibbs sampling process of the parameters of a Normal HMM with K states.

Algorithm 8 : The full Gibbs sampling of K-state Normal HMM.

```
Initialization: Start with initial samples at $m = 0$: $\mathbf{z}^{(0)}, \Theta^{(0)} = (\pi^{(0)}, \mathbf{A}^{(0)}, \mu^{(0)}, \tau^{(0)})$:
For $m = 1, 2, ..., M$ iterations:
    1. Compute the sufficient statistics required;
       $T_k = \sum_{t=1}^{T} \mathbb{I}_{(z_t^{(m-1)}=k)}$, $\sum_{t:z_t^{(m-1)}=k} y_t = \sum_{t=1}^{T} \mathbb{I}_{(z_t^{(m-1)}=k)} y_t$, and ,
    2. update $\mu_k^{(m)}$ from Equation (4.36).
    3. Compute $s_k^{(v)(m)} = \sum_{t=1}^{T} \mathbb{I}_{(z_t^{(m-1)}=k)} (y_t - \mu_k^{(m)})^2$;  $k = 1, 2, ..., K$, and
    4. update $\tau_k{}^{(m)}$ from Equation (4.37).
    5. Compute the transition counts $N_j^{(m)}$, $N_{jk}^{(m)}$, and,
    6. update $\pi^{(m)}$ and $\mathbf{A}^{(m)}$ from Equations (4.34) and (4.35) respectively.
    7. Sample the hidden states $\mathbf{z}$ using algorithm (7).
    8. Increment $m$.
```

4.5.2 Bayesian Poisson HMM

In this section, we describe a Bayesian HMM, in which the observed process follows a state-specific Poisson distribution. As with the Bayesian Normal HMM, we can construct a Bayesian PHMM. The priors and posteriors of the parameters π and $\mathbf{A}$ are the same for both models. The only difference is in deriving the likelihood and posterior of the state-based distribution of the PHMM.

To specify the model, assume $\mathbf{y} = (y_1, y_2, ..., y_T)$ are realizations (counts) of an observed process Y defined on the observed discrete space, $\mathbb{N}$, where $\mathbb{N}$ denotes non-negative integer values. The resulting realizations y_t, with parameter of interest λ, can follow a Poisson distribution as

$$y_t \sim Poi(\lambda);\ \lambda > 0. \tag{4.41}$$

The probability mass function of the observed count y_t is then given by

$$Pr(\mathrm{y}_t|\lambda) \sim \frac{e^{-\lambda}\lambda^{\mathrm{y}_t}}{\mathrm{y}_t!}; \; \mathbf{y} \geq 0. \tag{4.42}$$

In the PHMMs context, the observed process at any time is Poisson with a mean parameter λ, which depends only on an underlying hidden state, z_t, which in turn follows a Markov property and takes discrete values k defined on the state space $\{1,2,..,K\}$. Thus, the standard density in Equation (4.42) can be modified by defining the state-dependent probability density, given by

$$Pr(\mathrm{y}_t|\mathrm{z}_t = k, \lambda_t) = \frac{e^{-\lambda_{\mathrm{z}_t}}\lambda_{\mathrm{z}_t}^{\mathrm{y}_t}}{\mathrm{y}_t!}; \; j = 1,2,...,K. \tag{4.43}$$

A Bayesian PHMM is given by

$$\begin{aligned} Pr(\boldsymbol{\pi},\mathbf{A},\lambda|\mathbf{y},\mathbf{z}) &\propto L_c(\boldsymbol{\pi},\mathbf{A},\lambda;\mathbf{y},\mathbf{z})Pr(\boldsymbol{\pi},\mathbf{A},\lambda), \\ &\propto Pr(\mathbf{y}|\mathbf{z},\lambda)Pr(\mathbf{z}|\boldsymbol{\pi},\mathbf{A})Pr(\boldsymbol{\pi})Pr(\mathbf{A})Pr(\lambda). \end{aligned} \tag{4.44}$$

where $L_c(\boldsymbol{\pi},\mathbf{A},\lambda;\mathbf{y},\mathbf{z})$ represents the complete likelihood function of the model. As with the Bayesian Normal HMM derived earlier, the priors of both $\boldsymbol{\pi}$ and $\mathbf{A}$ are given a Dirichlet distribution with parameter $\boldsymbol{\delta}$. For the state-based parameter λ, we assume a Gamma distribution as a conjugate prior on the parameter λ (Gelman and Hill, 2007, p.52), such that

$$\lambda \sim Gamma(\kappa, \nu),$$

where κ and ν are hyper-parameters. Thus,

$$Pr(\lambda) \propto \lambda_k^{\kappa-1}e^{-\nu\lambda_k}. \tag{4.45}$$

The Bayesian PHMM can be then given by

$$Pr(\boldsymbol{\pi},\mathbf{A},\lambda|\mathbf{y},\mathbf{z}) \propto \prod_{k=1}^{K}\pi_k^{N_k}\prod_{k=1}^{K}\prod_{j=1}^{K}a_{jk}^{N_{jk}}\prod_{k=1}^{K}\prod_{t:\mathrm{z}_t=k}^{T}e^{-\lambda_{\mathrm{z}_t}}\lambda_{\mathrm{z}_t}^{\mathrm{y}_t} \times \prod_{k=1}^{K}\pi^{\delta_k-1}\prod_{k=1}^{K}\prod_{j=1}^{K}a_{jk}^{\delta_k-1}\lambda_k^{\kappa-1}e^{-\nu\lambda_k}. \tag{4.46}$$

The full conditional posterior of λ_k, independently for each $k \in \{1,2,...,K\}$, is given by

$$\begin{aligned} Pr(\lambda_k|\mathbf{y},\mathbf{z}) &\propto \prod_{t=1}^{T} Pr(\mathrm{y}_t|\lambda_k)Pr(\lambda_k), \\ &\propto \prod_{t=1}^{T} \left[e^{(-\lambda_k)}(\lambda_k)^{\mathrm{y}_t}\right] \times \left[\lambda_k^{\kappa-1} e^{-\nu\lambda_k}\right], \\ &= \lambda_k^{\sum_{t:\mathrm{z}_t=k} \mathrm{y}_t + \kappa - 1} e^{-(\nu + \sum_{t:\mathrm{z}_t=k} T_k)\lambda_k}, \\ &= Gamma(\kappa + \sum_{t:\mathrm{z}_t=k} \mathrm{y}_t, \nu + \sum_{t:\mathrm{z}_t=k} T_k), \end{aligned} \tag{4.47}$$

where T_k is the state-based observations size and $\sum_{t:\mathrm{z}_t=k} \mathrm{y}_t$ is the sum of the observations generated, given state k. The full conditional posteriors of $\boldsymbol{\pi}$ and $\mathbf{A}$ used with PHMM are the same as those used with Normal HMM (Equations (4.34-4.35)). For more details about deriving the full conditional posterior distribution of a Poisson HMM, see Appendix (A).

A local updating for the sequence of hidden states $\mathbf{z}$ of the PHMM can be introduced as follows (Robert and Titterington, 1998):

$$Pr(\mathrm{z}_1 = j|\mathrm{z}_2, ..., \mathbf{y}, \boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\lambda}) = \frac{\pi_j a_{j\mathrm{z}_2} \lambda_j^{\mathrm{y}_1}.e^{-\lambda_j}}{\sum_{l=1}^{K} \pi_l a_{l\mathrm{z}_2} \lambda_l^{\mathrm{y}_1}.e^{-\lambda_l}}, \text{ for } t = 1, \tag{4.48}$$

for $2 < t < T$ the full conditional distribution of z_t is

$$Pr(\mathrm{z}_t = j|..., \mathrm{z}_{t-1}, \mathrm{z}_{t+1}, ..., \mathbf{y}, \mathbf{A}, \boldsymbol{\lambda}) = \frac{a_{\mathrm{z}_{t-1}j} a_{j\mathrm{z}_{t+1}} \lambda_l^{\mathrm{y}_t}.e^{-\lambda_j}}{\sum_{l=1}^{K} a_{\mathrm{z}_{t-1}l} a_{l\mathrm{z}_{t+1}} \lambda_l^{\mathrm{y}_t}.e^{-\lambda_l}}, \tag{4.49}$$

and for $t = T$

$$Pr(\mathrm{z}_T = j|...\mathrm{z}_{T-1}, \mathbf{y}, \mathbf{A}, \boldsymbol{\lambda}) = \frac{a_{\mathrm{z}_{T-1}j} \lambda_j^{\mathrm{y}_{T-1}}.e^{-\lambda_j}}{\sum_{l=1}^{K} a_{\mathrm{z}_{T-1}l} \lambda_l^{\mathrm{y}_{T-1}}.e^{-\lambda_l}}. \tag{4.50}$$

Algorithm (9) shows the full Gibbs sampling for the parameters and hidden states of PHMM with K states.

Algorithm 9 : The full Gibbs sampling of K-state PHMM.

```
Initialization: Start with initial samples at $m = 0$: $\mathbf{z}^{(0)}, \Theta^{(0)} = (\boldsymbol{\pi}^{(0)}, \mathbf{A}^{(0)}, \lambda^{(0)})$:
For $m = 1, 2, \ldots, M$ iterations:
    1. Compute the sufficient statistics required;
       $T_k = \sum_{t=1}^{T} \mathbb{I}_{(\mathrm{z}_t^{(m-1)}=k)}$, $\sum_{t:\mathrm{z}_t^{(m-1)}=k} \mathrm{y}_t = \sum_{t=1}^{T} \mathbb{I}_{(\mathrm{z}_t^{(m-1)}=k)} \mathrm{y}_t$, and
    2. update $\lambda_k^{(m)}$ from Equation (4.47).
    3. Compute the counts $N_j^{(m)}$, $N_{jk}^{(m)}$, and,
    4. update $\boldsymbol{\pi}^{(m)}$ and $\mathbf{A}^{(m)}$ from Equations (4.34) and (4.35) respectively.
    5. Sample the hidden states $\mathbf{z}$ using Equations (4.48-4.50).
    6. Increment $m$.
```

4.6 Label switching

A problem that substantially affects the MCMC outputs, where the marginal posterior distributions of all components or state-specific parameters are identical, is label switching or non-identifiability (Stephens, 2000; Jasra et al., 2005; Papastamoulis and Iliopoulos, 2010). In the HMMs context, the main reason for this problem is that the likelihood of these models

$$L(\Theta; \mathbf{y}) = \sum_{\forall \mathbf{z}} L_c(\Theta; \mathbf{y}, \mathbf{z}),$$

is invariant under arbitrary permutations of labels of hidden state z_t. In other words, let $\mathscr{P}_k$ be the set of $k!$ permutations of the state labels $\{1, \ldots, k\}$, where $\mathrm{z}_t = k \; \forall t \in T$, and $\rho \in \mathscr{P}_k$ is some arbitrary permutation, then the observed likelihood

$$\begin{aligned} L(\rho(\Theta); \mathbf{y}) &= \sum_{\forall \mathbf{z}} L_c(\Theta; \mathbf{y}, \rho(\mathbf{z})) \\ &= \sum_{\forall \mathbf{z}} L_c(\rho(\Theta); \mathbf{y}, \mathbf{z}) \\ &= L(\theta_{\rho(1)}, \ldots, \theta_{\rho(k)}, a_{\rho(1),.}, \ldots, a_{\rho(k),.}, \pi_{\rho(1)}, \ldots, \pi_{\rho(k)}; \mathbf{y}); \; k = 1, 2, \ldots, K, \\ &= L(\Theta; \mathbf{y}), \end{aligned} \tag{4.51}$$

will stay invariant under any other relabelling of the states. It follows that if the parameters for two states are exchangeable in their prior distribution, then the resulting posterior distribution will be identical for all k (Jasra et al., 2005; Papastamoulis and Iliopoulos, 2010).

Figure (4.2) illustrates the consequence of the non-identifiability of the posterior distribution of a Normal HMM with 6-states fitted to the Galaxy data. It can observe, especially, that the middle four marginal posterior distributions for the state-specific mean parameters, μ_k, are subject to label switching, induced by random permutations of the hidden state labels occur over an MCMC run.

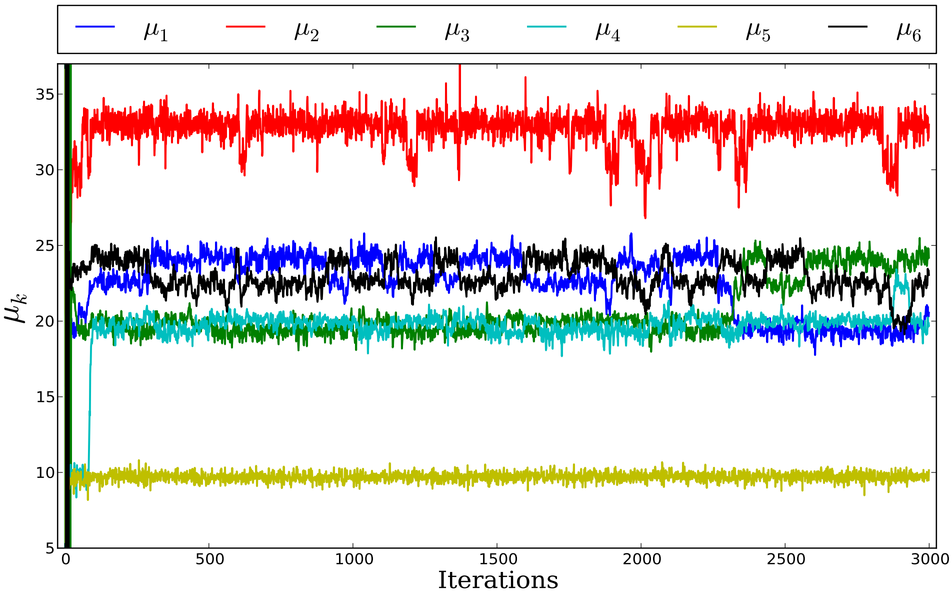


Figure 4.2: Label switching of Galaxy data fitted to 6-state Normal HMM using the unconstrained DG sampler.

A simple way that is introduced by Diebolt and Robert (1994) is to re-order the posteriors of the parameters by imposing artificial identifiability constraints (IC) on one or more of the parameters, which aims at breaking the symmetry in the prior and thus the posterior distribution. We adopt this method throughout this thesis. For example, we impose a constraint on the state-specific mean parameter of the Normal HMM fitted to the Galaxy data

$$\mu_1 < \mu_2 < \ldots < \mu_k.$$

Figure (4.3) shows the solution of label switching problem, given above the constraint on μ_k.

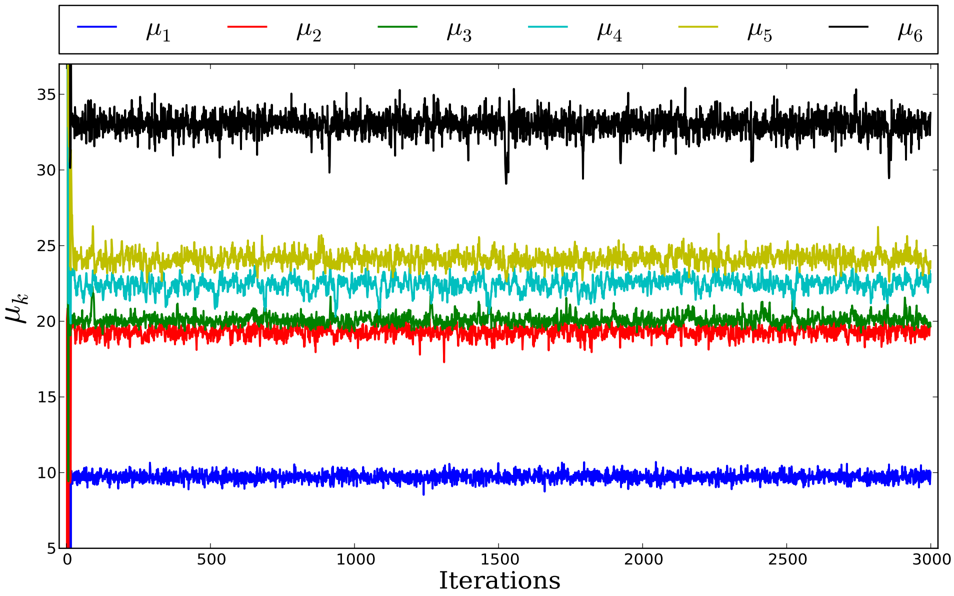


Figure 4.3: Using the IC method to solve the label switching for a Normal HMM with six states fitted to the Galaxy data.

4.7 A simulation study

In this section, we carry out a simulation study to assess our sampler using a synthetic data set of length $T = 300$ generated from a 3-state Normal HMM with parameters

$$\Theta = \left(\boldsymbol{\pi} = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.2 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} 5 \\ 10 \\ 20 \end{bmatrix}, \boldsymbol{\sigma^2} = \begin{bmatrix} 1 \\ 0.7 \\ 1.6 \end{bmatrix} \right),$$

in which we observe Normal variables $\mathrm{y}_t \sim N(\mu_k, \sigma_k^2)$; $k = 1,2,3$, whose parameters (μ_k, σ_k^2) depend on a hidden state z_t such that $\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T$ is a Markov chain. These values of the model parameters were made to reflect real life situations where the three components have variances that makes distinguishing them difficult. The histogram of the synthetic data set is displayed in Figure (4.4). By following algorithm (8), we then fit a 3-state Normal HMM to the synthetic data set simulated from the above model. We specify non-informative priors for the model parameters. We set a Gamma distribution with parameterization: $Gamma(0.001, 0.001)$, on the precision parameter, $\boldsymbol{\tau}$, and a Normal distribution with parameterization: $N(0, 0.001)$ on the mean parameter, $\boldsymbol{\mu}$. Regarding the initial state, $\boldsymbol{\pi}$ and transition distribution $\mathbf{A}$, we assign a Dirichlet prior with hyper-parameter δ with a value equal to 1. We use identifiability constrains on the mean parameters so that $\mu_1 < \mu_2 < ... < \mu_k$ to treat the occurrence of the label switching problem.

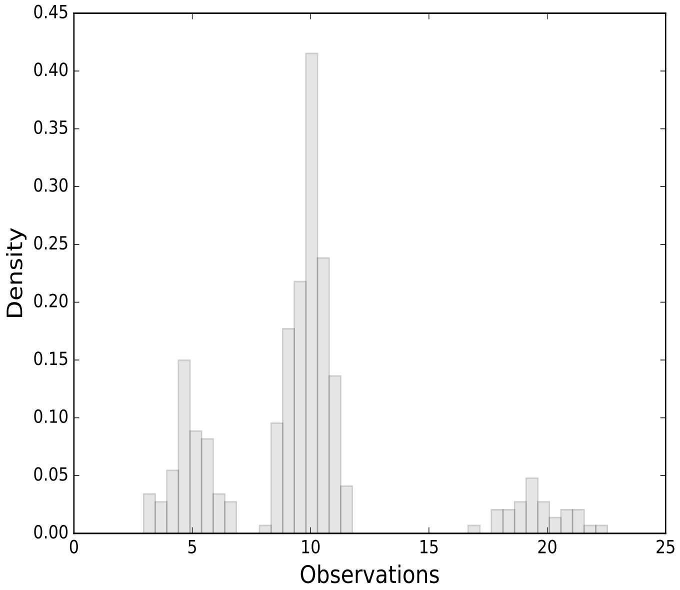


Figure 4.4: A histogram of the synthetic data generated from a Normal HMM with 3 states.

Case I:

We run 10 parallel chains, with different dispersed starting points, of length 1000 iterations each, for each parameter. To monitor the behaviour of produced chains from the sampler, we present all 1000 iterations, without burn-in or thinning. The 3-state Normal HMM adopted in this study includes 18 parameters; 3 for the initial state distribution, 9 for the transition distribution, 3 for mean parameter and 3 for variance parameter. Therefore, $10 \times 18 = 180$ chains will be generated.

4.7.1 Results of Case I

Figures (4.5-4.8) display trace-plots of 10 parallel chains generated using the DG sampler of all 18 parameters; π_k, a_{jk}, μ_k and σ_k, where $j,k = 1,2,3$, respectively. Using visual inspection, we can see from the trace-plots concerning the DG sampler that all the posteriors of the model parameters traversed rapidly their target distributions with very few steps and were not influenced by the high dispersed starting points. In addition, they do not show any particular pattern and appear to mix well. It can be noted that the sampler has reached its target distribution almost at the 50^{th} iteration. As a result, one can adopt the first 10% of the total of iterations as a burn-in period, i.e. the first 100 iterations. To carefully examine convergence, we calculated the Gelman-Rubin statistic, $\hat{R}$, from the last 900 iterations. It can be seen from

Tables (4.1-4.4) that all values of Gelman-Rubin statistic, $\hat{R}$, were less than 1.1 for all 18 posterior of the model parameters. This would give another indicator of convergence of the chains to their posterior distributions.

	Mean Parameter		
Method	μ_1	μ_2	μ_3
DG	0.999	0.999	1.001

Table 4.1: Gelman and Rubin's statistics, R, for the mean parameters obtained using DG algorithm. Value less than 1.1 suggests that we could assume the convergence of the MCMC chains.

	Variance Parameter		
Method	σ_1^2	σ_2^2	σ_3^2
DG	0.999	1.004	0.999

Table 4.2: Gelman and Rubin's statistics, R, for the variance parameters obtained using DG algorithm. Value less than 1.1 suggests that we could assume the convergence of the MCMC chains.

	Initial Parameter		
Method	π_1	π_2	π_3
DG	1.000	1.000	0.999

Table 4.3: Gelman and Rubin's statistics, R, for the initial parameters obtained using DG algorithm. Value less than 1.1 suggests that we could assume the convergence of the MCMC chains.

	Transition Parameter								
Method	a_{11}	a_{12}	a_{13}	a_{21}	a_{22}	a_{23}	a_{31}	a_{32}	a_{33}
DG	0.999	0.999	0.999	0.999	1.005	0.999	0.999	0.999	1.001

Table 4.4: Gelman and Rubin's statistics, R, for the transition parameters obtained using DG algorithm. Value less than 1.1 suggests that we could assume the convergence of the MCMC chains.

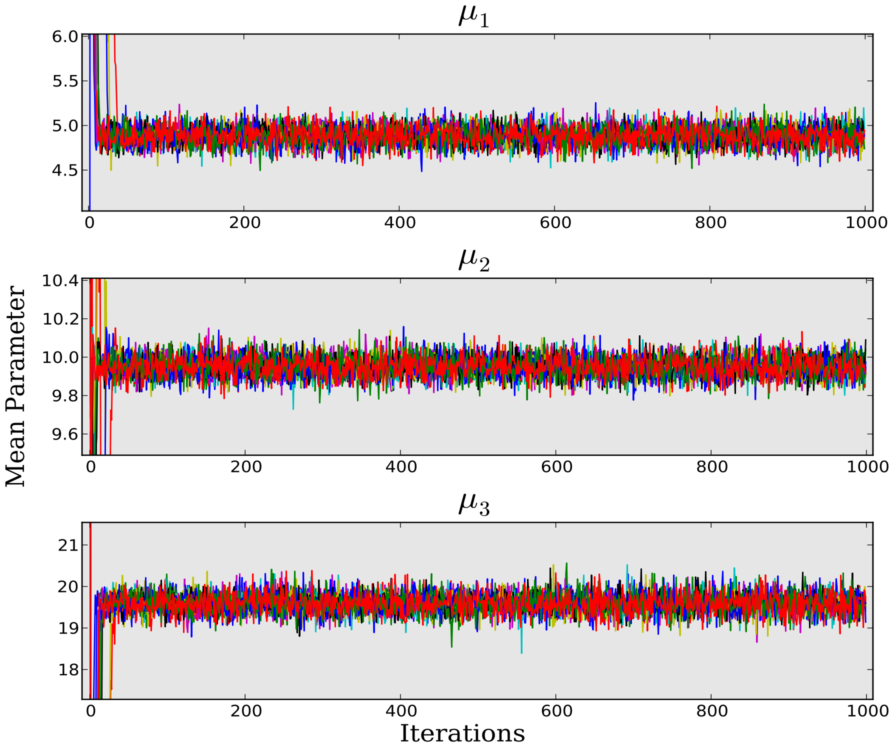


Figure 4.5: Trace-plots of 10 parallel MCMC runs of the mean parameter, $\mu_j;\ j = 1,2,3$.

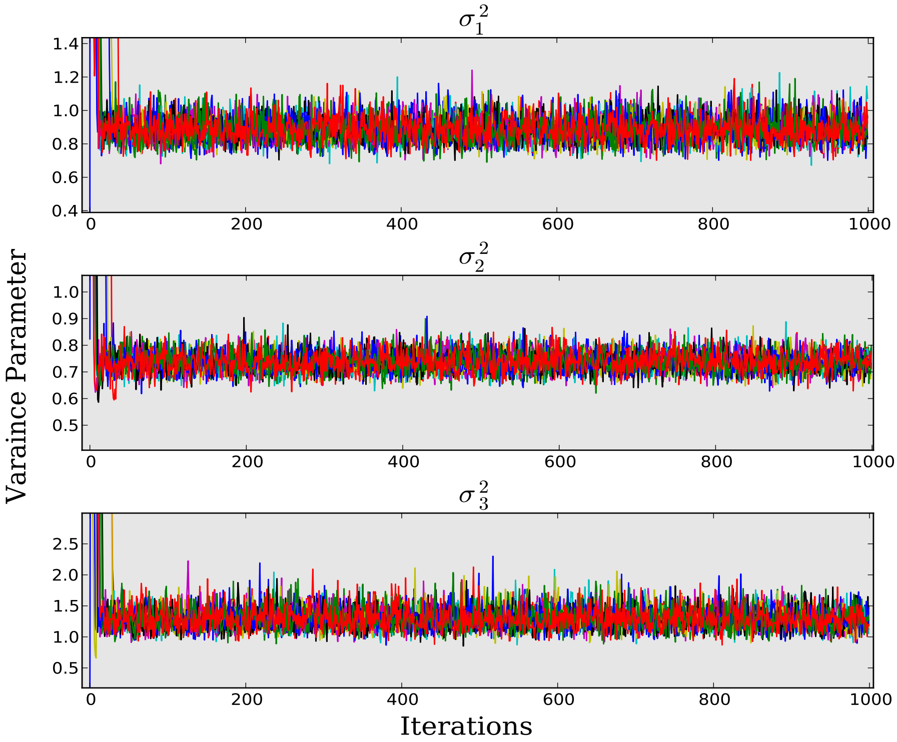


Figure 4.6: Trace-plots of 10 parallel MCMC runs of the variance parameter, $\sigma_j^2;\ j = 1,2,3$.

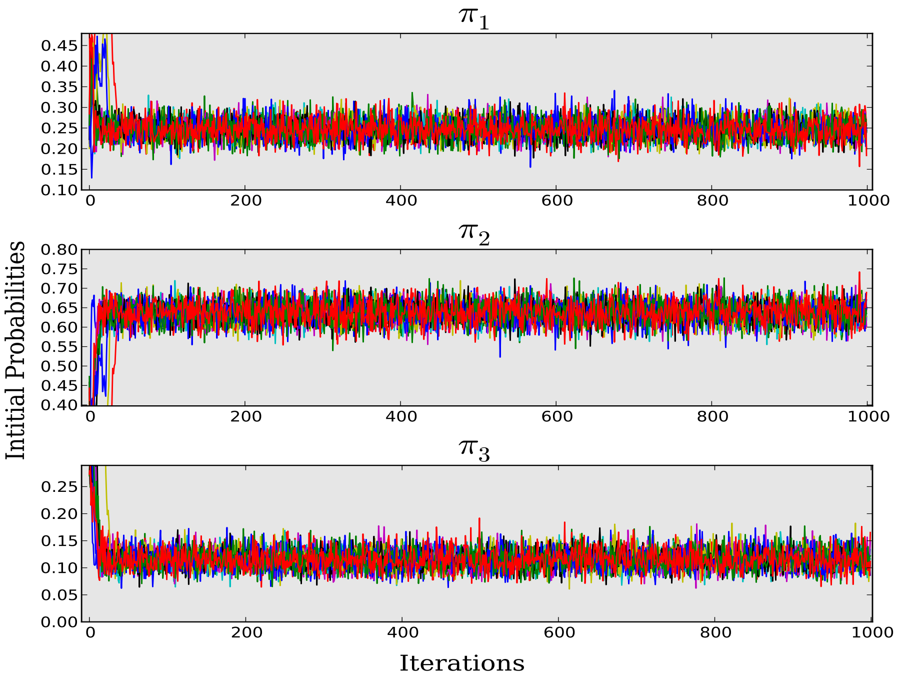


Figure 4.7: Trace-plots of 10 parallel MCMC runs of the initial distribution parameter, $\pi_j;\ j = 1,2,3$.

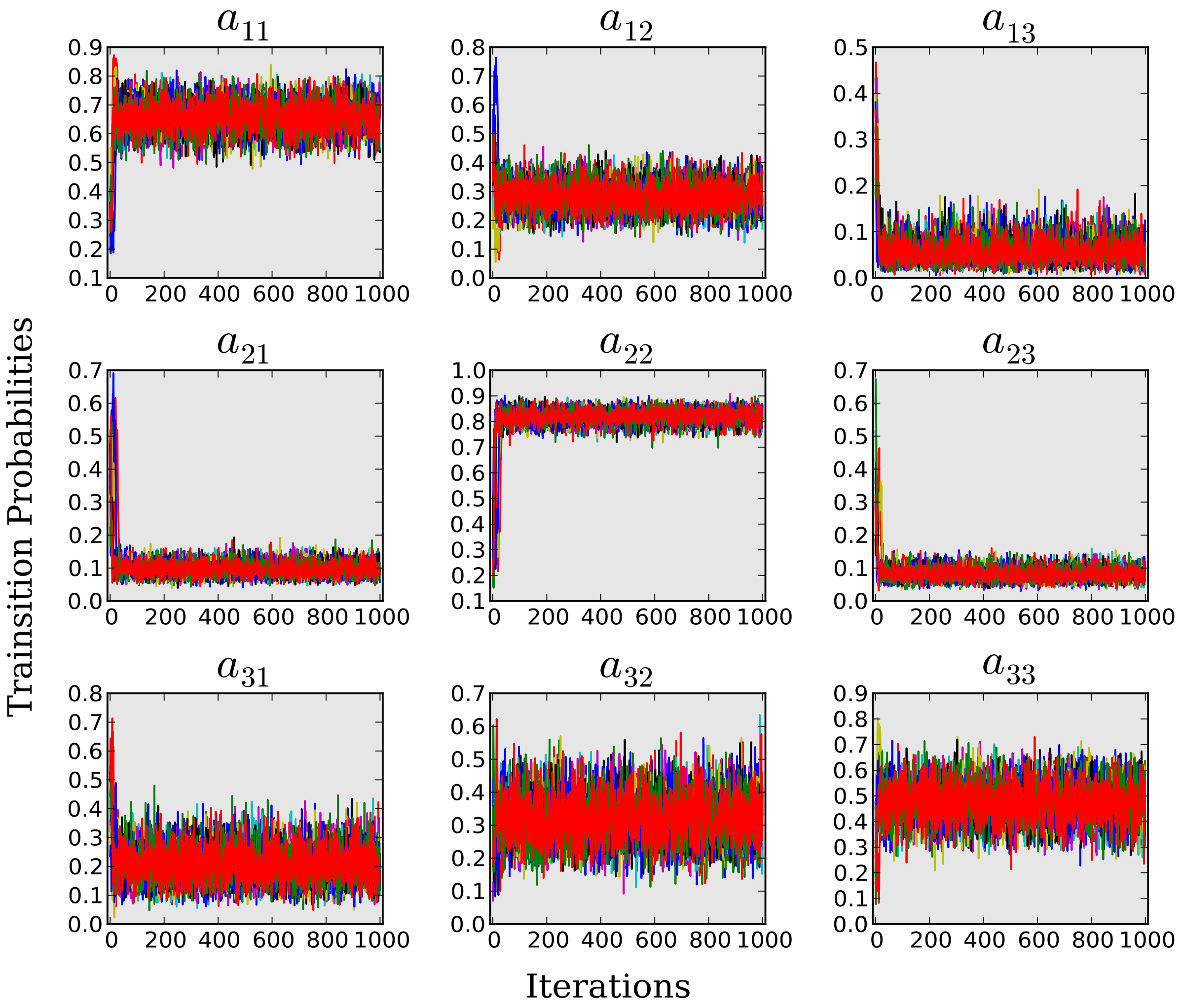


Figure 4.8: Trace-plots of 10 parallel MCMC runs of the probability transition parameters, $a_{jk};\ j,k = 1,2,3$.

Case II:

In the **Case II** we continue to assess the sampler using the synthetic data of the previous subsection, but inference for the model parameters will be implemented using a larger number of iterations. Thus, we adopt the same information provided in the preceding sub-section with respect to the priors on the model parameters. We also follow algorithm (8) for sampling from a Bayesian Normal HMM, mentioned in subsection (4.5.1). In the previous sub-section, we ran 10 chains, each one with 1000 iterations. Here, we run only one long chain and summarize the results based on that chain. We run the sampler for an iterative period of length M=10000 iterations (without thinning). We discard the first 1000 iterations from the original iterations as a burn-in period; the last 9000 iterations will be adopted to summarize the posterior results. We calculate the posterior means of the model parameters and also their corresponding 95% credible intervals. We apply the IC method on the mean parameter, μ_k to treat the non-identifiability problem. We provide the trace-plots of posterior distributions of all model parameters. We also provide the plots of the ACF functions for all parameters to check whether there is inherent correlation in the resulting posterior samples.

4.7.2 Results of Case II

For all model parameters, we display the results of posterior means and their corresponding 95% CIs as shown in Tables (4.5-4.8). Figure (4.9) displays visually only the estimation results of the mean parameter $\boldsymbol{\mu}$. It can be seen that the trace-plots of μ_j; $j = 1,2,3$, induced by the DG sampler were faster to reach their target distributions using only very few steps. Also, the sampler has good mixing along marginal chains. Figure (4.10) shows the fitting results of 3-state NHMM to the synthetic data.

The summaries of posterior estimates in Tables (4.5-4.8) suggest a close consistency between the estimated parameters and data generating mechanism for the most parameters. Note that the estimates of the mean parameter, $\hat{\mu}_j$; $j = 1,2,3$, were somewhat consistent with the parameters of the data generating mechanism. However, this was at the cost of the variance parameter, e.g. $\hat{\sigma}_3^2$=1.296 that was far from the true parameter (1.6). The same thing can be observed for the $\hat{\pi}_3$=0.108 and $\hat{a}_{33}$=0.514. Nevertheless, they were all within the range of parameter space.

Parameter	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\mu}_3$
True	5	10	20
Estimated	4.883	9.951	19.595
95%CI	(4.679, 5.088)	(9.847, 10.054)	(19.137, 20.052)

Table 4.5: Results of the estimation of mean parameter μ_k; $k = 1,2,3$, with 95% CI.

Parameter	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\sigma}_3^2$
True	1	0.7	1.6
Estimated	0.8875	0.7377	1.2956
95%CI	(0.739, 1.035)	(0.664, 0.811)	(0.957, 1.634)

Table 4.6: Results of the estimation of variance parameter σ_k^2; $k = 1,2,3$, with 95% CI.

Parameter	$\hat{\pi}_1$	$\hat{\pi}_2$	$\hat{\pi}_3$
True	0.2	0.6	0.2
Estimated	0.244	0.647	0.108
95%CI	(0.196,0.292)	(0.593, 0.701)	(0.074, 0.144)

Table 4.7: Results of the estimation of initial state parameter π_k; $k = 1,2,3$, with 95% CI.

Parameter	$\hat{a}_{11}$	$\hat{a}_{12}$	$\hat{a}_{13}$
True	0.6	0.3	0.1
Estimated	0.696	0.277	0.026
95%CI	(0.594, 0.799)	(0.177, 0.376)	(0.009, 0.062)
Parameter	$\hat{a}_{21}$	$\hat{a}_{22}$	$\hat{a}_{23}$
True	0.1	0.8	0.1
Estimated	0.086	0.842	0.071
95%CI	(0.046, 0.125)	(0.791, 0.894)	(0.034, 0.107)
Parameter	$\hat{a}_{31}$	$\hat{a}_{32}$	$\hat{a}_{33}$
True	0.1	0.3	0.6
Estimated	0.171	0.314	0.514
95%CI	(0.049, 0.292)	(0.162, 0.466)	(0.351, 0.678)

Table 4.8: Results of the estimation of transition parameters a_{jk}; $j,k = 1,2,3$, with 95% CI.

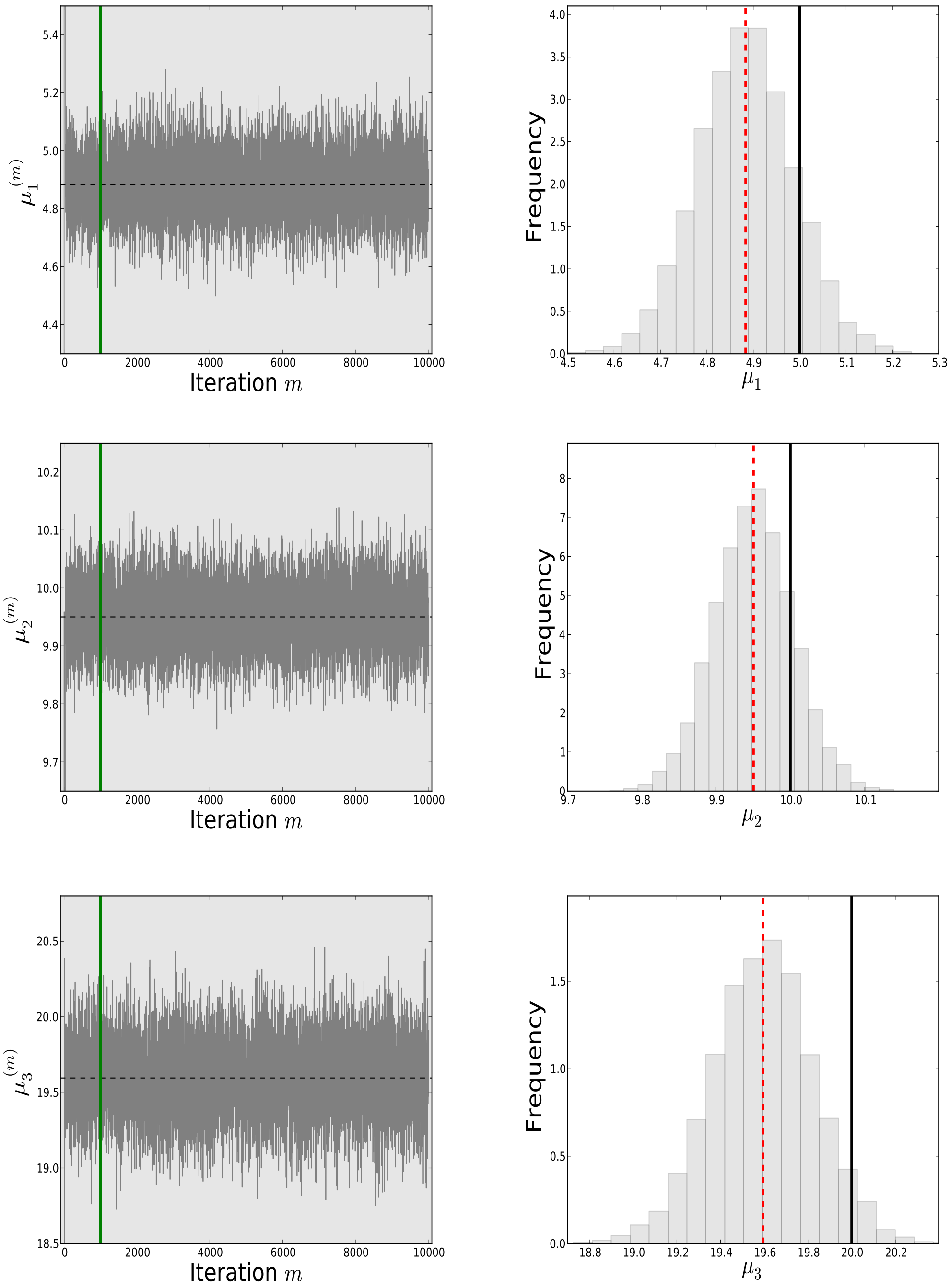


Figure 4.9: The graphs show simulations from the posterior distribution of the mean parameter μ_k; $k = 1,2,3$, of 3-state Normal HMM. The graphs in the first column represent the trace-plots of the mean parameters $(\mu_1 - \mu_3)$. The vertical green line in the trace-plots separates the burned-in samples (M=1000) from those used for the future inference (M=9000), and the horizontal black dashed line shows the true parameter. The graphs in the second column show the histograms of the densities of the mean parameters $(\mu_1 - \mu_3)$. The black solid and red dashed vertical lines show the true parameter and posterior mean respectively. The graphs in the third column show the autocorrelation functions of the mean parameters $(\mu_1 - \mu_3)$.

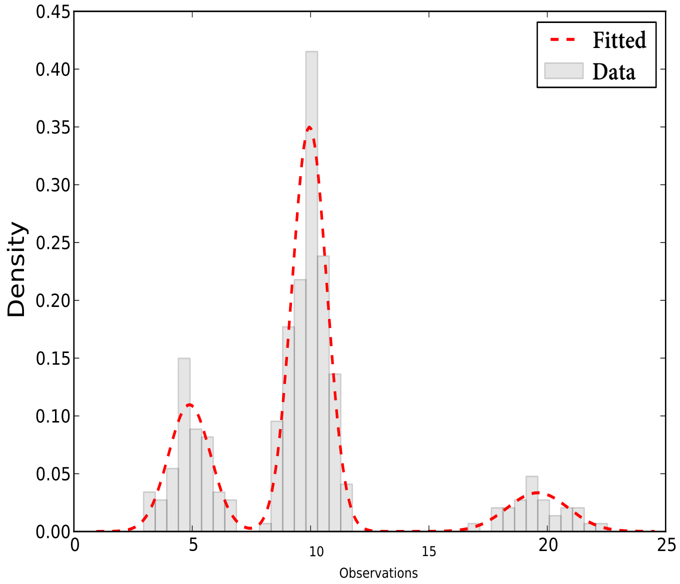


Figure 4.10: Fitting the densities of a Normal HMM with 3 states to the synthetic data using the DG sampler.

4.8 Application to real data

In this section we assess the performance of the sampler using a real application involving the waiting times of the Old Faithful geyser. These data consist of 299 observations which represent the waiting time, in minutes, between two successive eruptions. Old Faithful is a geyser in the Yellowstone National Park in Wyoming, USA (Hardle, 1991). These data have been fitted using Normal HMMs with a different number of states and also using different estimation methods. For instance, Robert and Titterington (1998) used the Bayesian framework to estimate the parameters of model fitted to those data. On the other hand, Zucchini and MacDonald (2009) used the frequentist framework for the same purpose. Figure (4.11) displays the trace-plot of a sequence of the waiting times between eruptions of the Old Faithful geyser. From Figure (4.11), it can note that the sequence appears a high heterogeneity among observations. In addition, these data have a strong serial dependence in the behaviour of the waiting times as shown from the sample autocorrelation function in Figure (4.12).

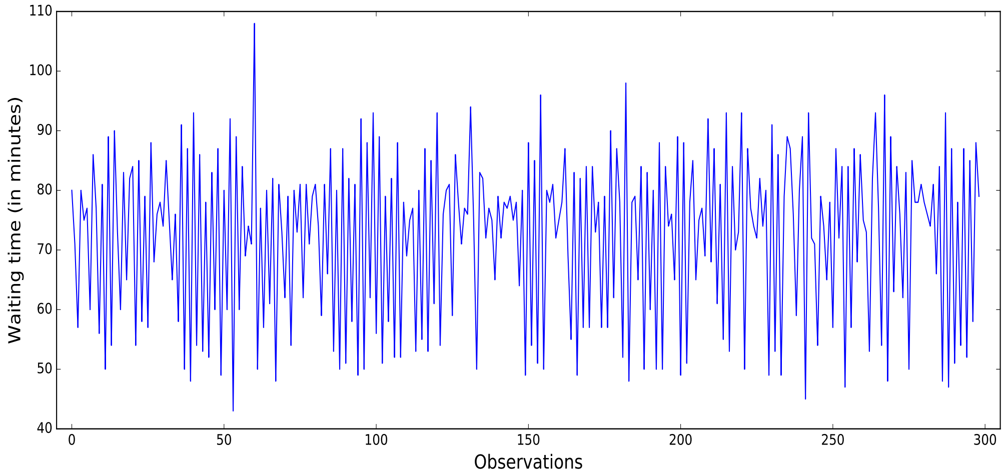


Figure 4.11: The waiting times between successive eruptions of the Old Faithful geyser.

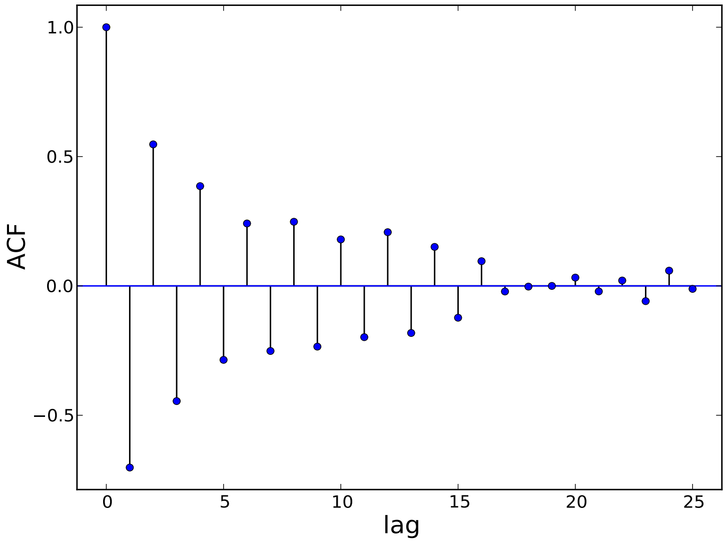


Figure 4.12: Sample ACF of the waiting times of the Old Faithful geyser data.

We consider the same information for the prior distributions on the initial state distribution $\boldsymbol{\pi}$, the rows of transition matrix $\mathbf{A}$, and on the state-dependent Normal parameters, μ and σ^2 as in the previous simulation study. We also run three parallel chains, each of which of 10000 iterations. We discard the first 1000 iterations as a burn-in period and adopt the last 9000 iterations for inference. We apply identifiability constraints on the mean parameter, μ_k, to address the label-switching problem.

4.8.1 Results

We display the results of posterior means with 95% CIs and also the values of Gelman-Rubin statistic, $\hat{R}$, for all model parameters as shown in Table (4.9). For comparison purposes, we also include in the same table the estimation results obtained from the previous studies on these data. We also present only the trace-plots of state-specific mean parameter, μ_j; $j = 1, 2, 3$, as shown in Figure (4.13).

It can be noted that the Gelman-Rubin statistic, $\hat{R}$, provides values less than 1.1 for all model parameters. This may reflect the fact that convergence has been achieved. The results of the parameter estimates obtained from the sampler were somewhat consistent with the results obtained by and Robert and Titterington (1998) as shown from Table (4.9). The results introduced by Zucchini and MacDonald (2009) were obtained using the frequentist approach, whereas those obtained by Robert and Titterington (1998) were using the Bayesian approach. Although the estimates introduced by Zucchini and MacDonald (2009) do not take into account the uncertainty about the model parameters, this consistency in results may be due to the fact that we used more non-informative priors. Hence, the results obtained by the Bayesian approach can be closer to those obtained using the classic approach.

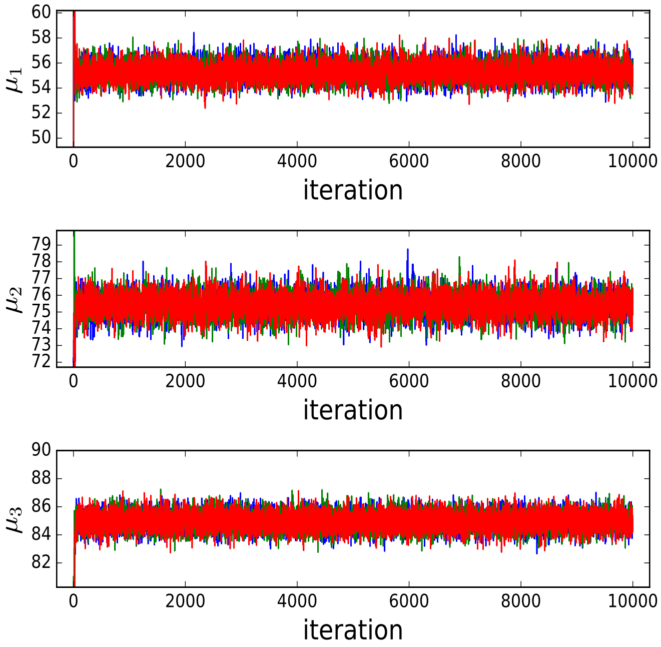


Figure 4.13: Trace-plots of the mean parameter, μ_j, of 3-state Normal HMM fitted to the waiting times of the Old Faithful geyser data.

Figure (4.14) displays state-specific means of the waiting times (at the bottom), plotted according to the most likely state sequence (at the top) which interprets the hidden pattern of eruptions. The figure reveals high levels of eruptions occur at the state1 ($\hat{\mu}_1 = 55.305$) and state 3 ($\hat{\mu}_3 = 84.942$), whereas a low level of eruptions occurs at the state 2 ($\hat{\mu}_2 = 75.418$).

Method	$\hat{\pi}$	$\hat{\mathbf{A}}$	$\hat{\mu}$	$\hat{\sigma}$
DG	$\begin{bmatrix} 0.341(0.284,0.398) \\ \hat{R}=1.006 \\ 0.265(0.193,0.336) \\ \hat{R}=1.008 \\ 0.393(0.325,0.461) \\ \hat{R}=0.999 \end{bmatrix}$	$\begin{bmatrix} 0.009(0.007,0.028) & 0.034(0.019,0.097) & 0.956(0.889,1.022) \\ \hat{R}=0.999 & \hat{R}=0.999 & \hat{R}=0.999 \\ 0.293(0.165,0.422) & 0.557(0.401,0.714) & 0.148(0.009,0.287) \\ \hat{R}=0.999 & \hat{R}=0.999 & \hat{R}=0.999 \\ 0.669(0.574,0.764) & 0.266(0.170,0.362) & 0.064(0.005,0.122) \\ \hat{R}=0.999 & \hat{R}=0.999 & \hat{R}=1.991 \end{bmatrix}$	$\begin{bmatrix} 55.305(53.926,56.684) \\ \hat{R}=1.002 \\ 75.418(74.144,76.692) \\ \hat{R}=1.004 \\ 84.942(83.780,86.104) \\ \hat{R}=1.001 \end{bmatrix}$	$\begin{bmatrix} 5.899(4.837,6.962) \\ \hat{R}=1.006 \\ 4.008(2.695,5.322) \\ \hat{R}=1.008 \\ 5.516(4.748,6.284) \\ \hat{R}=0.999 \end{bmatrix}$
Robert and Titterington (1998)	—	$\begin{bmatrix} 0.004 & 0.001 & 0.995 \\ 0.271 & 0.667 & 0.062 \\ 0.571 & 0.306 & 0.123 \end{bmatrix}$	$\begin{bmatrix} 55.400 \\ 75.400 \\ 84.900 \end{bmatrix}$	$\begin{bmatrix} 5.983 \\ 3.794 \\ 5.468 \end{bmatrix}$
Zucchini and MacDonald (2009)	$\begin{bmatrix} 0.342 \\ 0.259 \\ 0.399 \end{bmatrix}$	$\begin{bmatrix} 0.000 & 0.000 & 1.000 \\ 0.298 & 0.575 & 0.127 \\ 0.662 & 0.276 & 0.062 \end{bmatrix}$	$\begin{bmatrix} 55.300 \\ 75.300 \\ 84.930 \end{bmatrix}$	$\begin{bmatrix} 5.809 \\ 3.808 \\ 5.433 \end{bmatrix}$

Table 4.9: The results of the parameter estimates of 3-state Normal HMM using the DG sampler on the waiting times of the Old Faithful geyser data. The last two rows display the results obtained from previous studies. The numbers in brackets represent the corresponding 95% CIs.

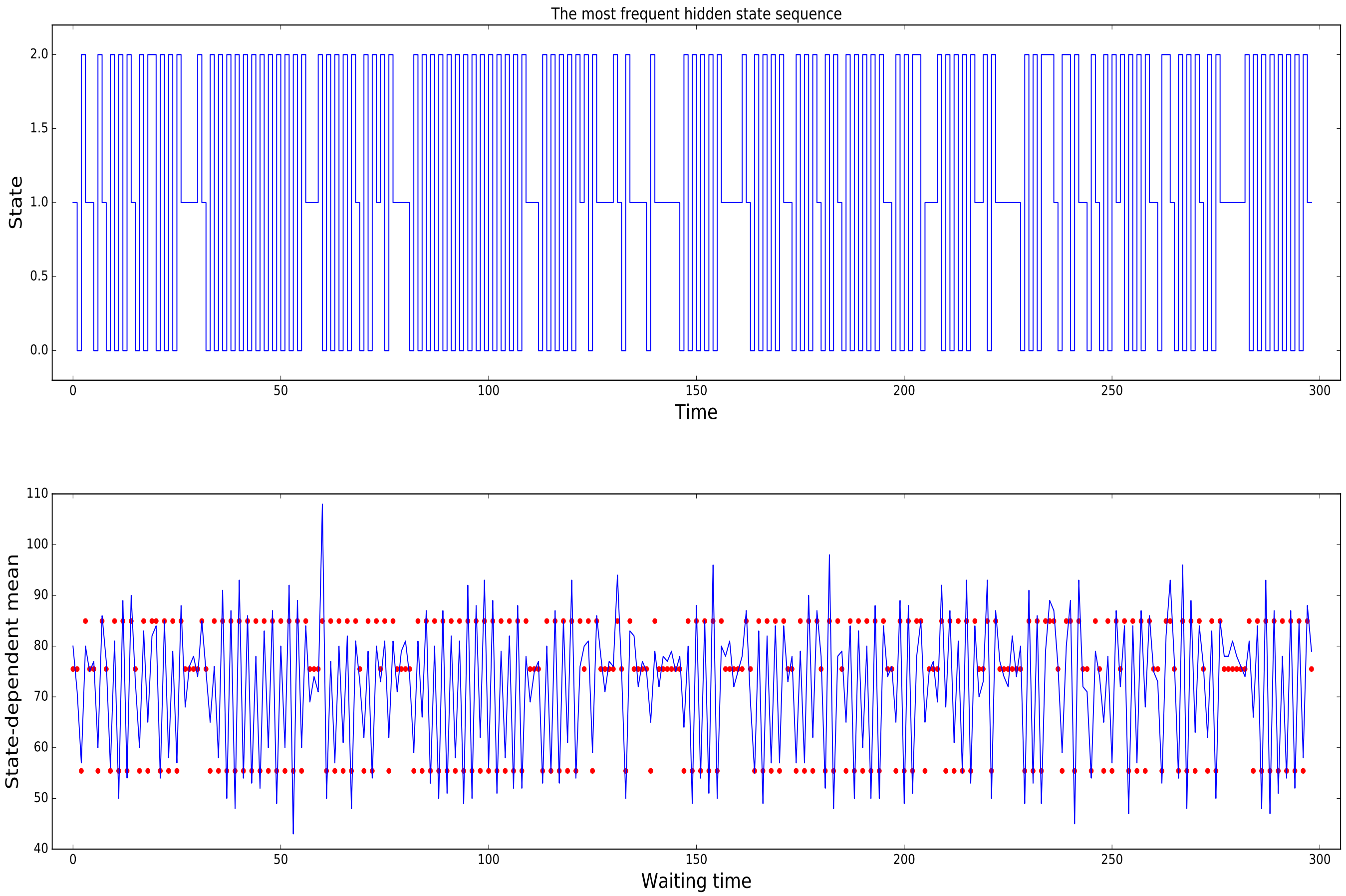


Figure 4.14: Trace-plot of the most frequent hidden state sequence (Top). State-dependent means (dashed horizontal points) vs. observations, depicted by the most likely state sequence (Bottom).

4.9 Summary

In this chapter, we have introduced a parametric HMM from a Bayesian viewpoint. We have specified the priors and full conditional distributions of all parameters of the model. We have developed Bayesian HMMs where observations, conditioned on hidden states, follow a parametric distribution such as a Normal or a Poisson. We have developed the special sampling algorithms of these models. We have also introduced the issue of sampling the hidden state sequences, which was implemented using the Direct Gibbs (DG) sampler or the so-called local updating. In addition, we have illustrated how obtaining the most likely state sequence, given several state sequences sampled over an MCMC run. This would allows us observing the more frequent state sequence over time as well understanding the structural nature of the data. We have assessed the sampler using simulated and real application data. The performance of the sampler was examined based on the trace-plots of all model parameters. We also used the Gelman-Rubin statistics to check convergence of the posterior chains produced from the sampler. The results have shown that the sampler provides satisfactory convergence results and it has a good mixing.

Chapter 5

Bayesian Selection Criteria for HMMs

5.1 Introduction

This chapter develops our contribution towards model selection for HMMs. There are a variety of ways of thinking about HMMs. We will expand on this idea in the next chapter, where we seek to find a model with an optimal number of hidden states.

In this aspect, our contribution involves the introduction of three groups of model selection criteria. The first group includes two of the most commonly used criteria which are the Akaike information criterion (AIC) (Akaike, 1973) and the Bayesian information criterion (BIC) (Schwarz, 1978). More specifically, we contribute in developing several versions for AIC and BIC that are modified using the Bayesian principle as a new application in the HMMs context. Basically, such an idea was first proposed by Brooks (2002) who applied it to autoregressive models. The second group includes the more popular Bayesian metric which is the deviance information criterion (DIC) (Spiegelhalter et al., 2002). We use some versions of this criterion as proposed by Celeux et al. (2006), who examined them on mixture models, for the HMM selection. In addition, we contribute in developing new versions for the same criterion in the HMM context. Finally, the third group addresses the model selection issue from a predictive perspective. In this context, our contribution is based on applying a fully Bayesian criterion called the widely applicable information (WAIC) (Watanabe, 2009) for HMMs. To our knowledge, the use of WAIC for HMMs has not been applied till writing this thesis.

In many applications of HMMs, the number of hidden states may be assumed known *a priori*, either from the context of the application or according to a scientific insight (Scott, 2002). For example, in an attempt to analyse the behavior of the Gross National Product (GNP) of U.S., Hamilton (1989) used Gaussian Markov-switching models to explain the recession status of GNP. He wished to consider two states of recession, namely, “Contraction” and “Expansion”, and estimated these by assuming a hidden Markov chain. Albert (1991) used two states to model a series of counts of myoclonic seizures suffered by one patient on 204 consecutive days. The high or low seizure activity states of the patient were modelled using a two-state Poisson HMM. Using Poisson HMMs, Leroux and Puterman (1992) used data on counts of movements in five-second intervals of one fetal lamb (240 intervals). They classified the physiological state of the fetal lamb into two states; a relaxed state and an excited state.

Conversely, some applications may assume that the number of underlying states is unknown and regard this quantity as a random variable to be estimated along with the other parameters of the model. For example, the reversible jump MCMC method was designed by Richardson and Green (1997) to estimate the number of components in independent mixtures and has been extended by Robert et al. (2000) to HMMs. We will not consider this method further. It is often computationally intensive. It also requires some caution when designing moves to ensure that Markov chains mix well both within model spaces (same number of states, different parameters) and amongst model spaces (different number of states). In addition, it may have some convergence problems. Furthermore, this method imposes challenges on the prior selection for the number of hidden states K (Fan and Sisson, 2011).

However, for our purposes, whilst we do not wish to made strong a priori assumptions about the number of states, the applications in the next chapter assume that we have to interpret the model in relation to a fixed number of states. Therefore, this thesis takes into account the issue of model selection in the HMM context by assuming a fixed but unknown number of hidden states, $K \in (1,2,\dots,K_{\max})$. Consequently, we choose an appropriate maximum number of hidden states for an HMM, fit several models with increasing numbers of states and then select the best model according to a proposed model selection criterion.

Several methods for model selection have been proposed. For instance, the Bayes factors method (BF; Kass and Raftery, 1995) has been introduced as a tool for model selection. The BF approach is based on computing the ratio of the marginal likelihoods of the data for two models under comparison. The evaluation of marginal density can be unsatisfactory for models that include high-dimensional integrals over the parameter space (Han and Carlin, 2011) and BF can be sensitive to the priors specified for the parameters of two models (Ando, 2010; Gelman et al., 2014).

In a frequentist context, the AIC and BIC have been used by Zucchini and MacDonald (2009) to determine the best HMM in many applications. However, these criteria can suffer from irregular behavior of the likelihood function that may cause under-fitting or over-fitting, where the number of hidden states that are analyzed is smaller or greater than the true number of states (Johnson, 2007). Furthermore, assessing the models based only on point estimates of parameters using the above criteria does not naturally incorporate our uncertainty pertaining to those quantities. One of the key advantages of Bayesian inference is its ability to account for various sources of uncertainty. On this basis, Brooks (2002, p. 617) proposed the possibility of developing criteria such as AIC and BIC from a Bayesian viewpoint. He pointed out through his comments on the paper published by Spiegelhalter et al. (2002) that one can use the likelihood or deviance evaluated at the posterior distribution of the model parameters and plug it into the AIC and BIC. Such a proposal was also presented by Ntzoufras (2009, pp.426-428), Carlin and Louis (2009, p.211) and Congdon (2014, p.36). In this chapter, we exploit this proposal and introduce several modified versions for the AIC and BIC developed from Bayesian principles for HMMs.

As an extended Bayesian version of the AIC, the DIC has been developed by Spiegelhalter et al. (2002) to consider the posterior distribution of the log likelihood. Similar to the AIC, DIC trades off a measure of model fit against a penalty for complexity. The popularity of this criterion stems from the ease of computing the posterior distribution of the log-likelihood or deviance and also its implementation in standard software such as `WinBUGS`. Consequently, it has been applied to a wide range of statistical models. However, applying this criterion to latent variables models is problematic as the likelihood function of these models is not available in a closed form (Celeux et al., 2006). For Bayesian modelling using the data augmentation principle (Tanner and Wong, 1987), the process of likelihood estimation for such models can be simplified. In other words, this strategy, whereby the parameter space is augmented by adding latent or missing data, facilitates the MCMC computation of the posterior distributions of those models.

Nevertheless, the DIC remains difficult to apply. As mentioned by Spiegelhalter et al. (2002) the DIC depends on a concept of *focus* that it is not always easily chosen in practice. In the context of latent variable models, Celeux et al. (2006) introduced several possible alternatives to the original DIC, taking into account the nature of inferential focus. They showed how focus can be on the main parameter of a model, or on the latent variables. They also showed that focus changes depending on whether these variables are considered as missing data or extra parameters. Finally, they showed that the focus can depend on the nature of the likelihood used specifically whether it is the observed, complete or a conditional likelihood.

Developing model selection criteria for HMMs requires the availability of the closed form of likelihood of those models (Celeux et al., 2006). This can be achieved by using the data augmentation technique which leads to different forms to the likelihood function of those models as defined by Celeux et al. (2006). We then extend the original definition of the DIC, taking into account the definitions given by Celeux et al. (2006) in a HMM context. We introduce several versions of this criterion based on different definitions of the observed and conditional likelihood of a HMM. Despite using the augmented data, the observed likelihood of a HMM is computationally challenging as it requires integrating out high-dimensional vectors of hidden states. We therefore use an efficient method of evaluation called the forward algorithm (Rabiner, 1989), which reduces the dimensionality in the computation of the likelihood. In contrast, the computation of the conditional likelihood is easier as it is directly applied, given a chosen focus.

Model choice can also be addressed from a predictive viewpoint in terms of selecting the best predictive performance compared with other competing models (Vehtari and Ojanen, 2012; Gelman et al., 2014). Under a Bayesian perspective, several criteria that take the predictive performance into account as a measure for comparing models have been proposed. One criterion is the posterior predictive distribution (PPD) of Laud and Ibrahim (1995), Gelman et al. (1996) and Gelfand and Ghosh (1998). This approach is useful for revealing inconsistency between the model and data. Meng (1994) prefers the use of PPD provided that its use is limited only for measuring the discrepancy between the model and the data and not for model comparison and inference (Carlin and Louis, 2009, p. 87).

In order to avoid the double use of the data, one alternative approach is to assess the predictive ability of the model on out-of-sample data. Out-of-sample data are observations that are not used to fit the model but are used to form predictions from the model. Cross-validation (CV) methods for model assessment and comparison are well established (Stone, 1974; Geisser, 1975; Geisser and Eddy, 1979; Gelfand and Dey, 1994; Gelfand and Ghosh, 1998). The key idea in the cross-validation strategy is to split the full set of data $\mathbf{y}$ into k subsets (k-fold cross-validation). Consider $k = 2$ for 2-fold cross-validation where we have two data subsets, $(\mathbf{y}_{T_1}, \mathbf{y}_{T_2})$ such that $T_1 \cup T_2 = T$. The first set (training sample), $\mathbf{y}_{T_1}$, is used to fit the model and estimate the posterior distributions of interest, while the remaining observations (test sample), $\mathbf{y}_{T_2}$, are used for model evaluation and checking by calculating the cross-validation predictive density. The leave-one-out cross-validation (LOO-CV) is a special variant of k-fold cross-validation, when $k = T$, where each data point is successively excluded from the full data set, a model is fitted using MCMC methods without the excluded point, and then the predictive distribution for the omitted observation is found. This procedure is repeated with each observation as an excluded point.

Overall, the CV methods can be computationally expensive, since the repeated exclusion of observations each time and fitting the model on the remaining data requires a long time that increases exponentially as the sample size increases (Vehtari and Ojanen, 2012).

In the HMM context, Celeux and Durand (2008) used a CV method on HMMs to determine the number of hidden states. They adopted two ways to approximate the cross validated likelihood; a half-sampling and an EM procedure. Their results were based on a comparative study with various versions of the cross-validated likelihood criterion such as the AIC, BIC, the integrated completed likelihood criterion (ICL) proposed by Biernacki et al. (2001) and the penalized marginal likelihood criterion proposed and studied in Gassiat and Keribin (2000). However, their procedures did not take into account any uncertainty about the parameters of the model. In addition, the high computational cost of this approach requires a long time.

Recently, the WAIC was proposed by Watanabe (2009), which is claimed to be a good alternative to the LOO-CV. Watanabe (2010) has shown that the WAIC can be viewed as an asymptotic version of the LOO-CV and can be used to approximate the out-of-sample predictive ability. The WAIC is based on point-wise calculations to approximate the predictive densities of future observations. The main advantage of this criterion, compared with the LOO-CV, is that it has less computational cost as all predictive densities for observations are evaluated at one-time from only one MCMC run (Gelman et al., 2014).

5.2 Likelihood-based criteria

In this section we introduce the general definitions of three likelihood-based criteria, namely, AIC, BIC and DIC, which will be developed later for the HMMs in this thesis.

5.2.1 AIC and BIC

The Akaike information criterion (AIC) was introduced by Akaike (1973) as an approximation to the expected Kullback-Leibler (KL) distance (Kullback and Leibler, 1951) between a true but unknown model and an estimated model. This criterion was originally based on a point estimate, $\tilde{\theta}$, obtained by maximizing the likelihood function and then measuring the deviation of the estimated model from the true model using the KL information function, defined as

$$I\left[f(\mathbf{y}), g(\mathbf{y}, \tilde{\theta})\right] = \int_{\Omega} f(\mathbf{y}) \log\left\{\frac{f(\mathbf{y})}{g(\mathbf{y}, \tilde{\theta})}\right\} d\mathbf{y}, \tag{5.1}$$

where $I[.]$ denotes the loss function called the KL discrepancy that expresses the amount of information lost when the estimated model $g(\mathbf{y}, \tilde{\theta})$ is used to approximate the true model $f(\mathbf{y})$

(Burnham and Anderson, 2002). $\tilde{\theta} \in \Omega$ denotes a maximum likelihood estimator given the data. Given k candidate models being fitted to data set, $\mathbf{y}$, where $k = 1, 2, ..., K$, the aim is to find the model that minimizes the distance $I[.]$ compared to other candidate models (Burnham and Anderson, 2002). In other words, the KL divergence serves as a criterion to measure the lost information induced from using the estimated model $g(\mathbf{y}, \tilde{\theta})$ to approximate $f(\mathbf{y})$. Conceptually, the best model will be the one that loses the least information and corresponds to the smallest KL distance. However, it cannot be used directly as a criterion for model selection, as the true model $f(\mathbf{y})$ is an unknown, as are the estimates, $\tilde{\theta}$, in $g(\mathbf{y}, \tilde{\theta})$. To solve this problem, one has to look at the *relative* KL distance between the estimated and true model. By writing the Equation (5.1) as

$$I\left[f(\mathbf{y}), g(\mathbf{y}, \tilde{\theta})\right] = \int f(\mathbf{y}) \log\{f(\mathbf{y})\} \, d\mathbf{y} - \int f(\mathbf{y}) \log\left\{g(\mathbf{y}, \tilde{\theta})\right\} d\mathbf{y}, \tag{5.2}$$

both terms in the right of Equation (5.2) can be viewed as statistical expectations taken over the true model $f(\mathbf{y})$. Hence, the KL distance can be thought as a difference between two expectations

$$I\left[f(\mathbf{y}), g(\mathbf{y}, \tilde{\theta})\right] = E_{f(\mathbf{y})}\left[\log\{f(\mathbf{y})\}\right] - E_{f(\mathbf{y})}\left[\log\left\{g(\mathbf{y}, \tilde{\theta})\right\}\right]. \tag{5.3}$$

The first term in Equation (5.3) is a constant which depends only on the unknown real distribution of data and does not include the parameter θ (Burnham and Anderson, 2002). The second term, $E_{f(\mathbf{y})}\left[\log\left\{g(\mathbf{y}, \tilde{\theta})\right\}\right]$, in Equation (5.3) is called the relative KL discrepancy from approximating the model $g(\mathbf{y}, \tilde{\theta})$ with $f(\mathbf{y})$. Akaike (1973) showed that the quantity $E_{f(\mathbf{y})}\left[\log\left\{g(\mathbf{y}, \tilde{\theta})\right\}\right]$ cannot be evaluated, but found that one can estimate its expectation as $E_{f(\mathbf{y})}E_{f(\mathbf{y})}\left[\log\left\{g(\mathbf{y}, \tilde{\theta})\right\}\right]$ which he called the *relative expected* KL distance. Based on the empirical log-likelihood function, $\ell(\tilde{\theta}|\mathbf{y})$, Akaike (1973) provided an approach to estimate the relative expected KL distance. He concluded that the value of the maximized log-likelihood, $\ell(\tilde{\theta}|\mathbf{y})$, was a biased estimate of the relative expected KL distance and showed that this bias is approximately equal to h, where h denotes a penalty term equal to the number of estimated parameters used in the approximation process of model $g(\mathbf{y}, \tilde{\theta})$ (Burnham and Anderson, 2002, 2004);

$$\text{relative } \hat{E}(\text{KL}) = \ell(\tilde{\theta}|\mathbf{y}) - h. \tag{5.4}$$

Hence, the selected model will have the highest value for the quantity $\ell(\tilde{\theta}|\mathbf{y})-h$, i.e. the best penalized log-likelihood. For historical reasons, Akaike (1973) proposed to multiply Equation (5.4) by -2 and obtain the minimum value of ($-2\ell(\tilde{\theta}|\mathbf{y})$ plus twice the penalty term) instead. Given k candidate models, $k = 1,2,...,K$, each one with h parameters being estimated, Akaike (1973) introduced his criterion as follows

$$\text{AIC}_k = -2\ell_k(\tilde{\theta}|\mathbf{y}) + 2h_k;\ k = 1,2,...,K, \tag{5.5}$$

where $\ell_k(\tilde{\theta}|\mathbf{y})$ denotes the k^{th} log-likelihood of the k^{th} model fitted to data set, $\mathbf{y}$, given a point estimate, e.g. $\tilde{\theta} = \hat{\theta}_{\text{mle}}$, where $\hat{\theta}_{\text{mle}}$ is the maximum likelihood estimate (MLE) of θ, and h_k represents the number of free parameters of the k^{th} model. The first term is a measure of fit, and decreases with increasing the order of the model, k. The second term is a penalty term which increases with increasing k. Generally, smaller values of the AIC refer to a better model. Another well-known model selection criterion is the Bayesian information criterion (BIC) (Schwarz, 1978). Unlike the AIC, the BIC is not an estimator of relative KL. It arises from a Bayesian standpoint by assuming an equal prior probability for each model with flat priors on the model parameters (Raftery, 1995; Burnham and Anderson, 2002, 2004). Consider k competing models, $k = 1,2,...,K$, and assume that each model k is characterized by a parametric distribution $g_k(\mathbf{y}|\theta_k)$ with prior distribution $Pr_k(\theta_k)$. Given a sequence of observations $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$, the marginal distribution or probability of $\mathbf{y}$ for the k^{th} model is given by (Konishi and Kitagawa, 2008)

$$Pr_k(\mathbf{y}) = \int g_k(\mathbf{y}|\theta_k) Pr_k(\theta_k) d\theta_k, \tag{5.6}$$

where the quantity $Pr_k(\mathbf{y})$ can be considered as the likelihood of the k^{th} model and is referred to as the *marginal likelihood* of the data. Based on Bayes' rule, if we assume that the prior probability of the k^{th} model is $Pr(k)$, then the posterior probability of the k^{th} model is given by

$$Pr(k|\mathbf{y}) = \frac{Pr_k(\mathbf{y})Pr(k)}{\sum_{l=1}^{K} Pr_l(\mathbf{y})Pr(l)};\ l,k = 1,2,...,K, \tag{5.7}$$

which indicates the probability of the data being generated from the k^{th} model when the data set $\mathbf{y}$ are observed. Consequently, the model which has the largest posterior probability will be the preferred model. When assuming the same priors for all models, then the model that maximizes the marginal likelihood $Pr_k(\mathbf{y})$ of the data set must be selected. So, if an approximation to the marginal likelihood, expressed in terms of an integral in (5.6), can readily be obtained, then the need to compute the integral on a problem-by-problem basis will fade, thus making the BIC usable as a suitable criterion for model selection (Konishi and Kitagawa, 2008). The BIC can be then defined as the natural logarithm of the integral in Equation (5.6) multiplied by -2, i.e.,

$$\begin{aligned} \mathrm{BIC} = -2\log Pr_k(\mathbf{y}) &= -2\log\left\{\int g_k(\mathbf{y}|\theta_k)Pr_k(\theta_k)d\theta_k\right\}, \\ &\approx -2\log g_k(\mathbf{y}|\tilde{\theta}_k) + h_k\log(T), \end{aligned} \tag{5.8}$$

where $\tilde{\theta}_k$ is a point MLE estimate, i.e. $\tilde{\theta}_k = \hat{\theta}_{\text{mle}}$, h_k is the number of the k^{th} model parameters and $\log(T)$ is the natural logarithm of sample size T. As with the AIC, a smaller BIC value refers to a better model. It can be noted that the AIC and the BIC have the same quantity in the first term which is based on the maximization of the likelihood. However, they differ in the second term (the model complexity), as the BIC likely depends on the sample size T. The BIC is more likely to favour smaller or more parsimonious models than AIC when T is increased (Burnham and Anderson, 2002).

Overall, a generic formula representing information criteria (IC); AIC and BIC, can be given as (Ntzoufras, 2009)

$$\mathrm{IC}(k) = D(\tilde{\theta}_k) + h_k F, \tag{5.9}$$

where F refers to as a penalty or complexity term imposed on the deviance that increases as more parameters for the model are added. When $F = 2$ we obtain the AIC, and when $F = \log(T)$, we obtain the BIC. For comparing two models, for instance k_1 and k_2, one can select the model that has the lowest value of IC, and can also use the corresponding difference $\Delta\mathrm{IC}_{12}$ between the IC values of two compared models as

$$\Delta\mathrm{IC}_{12} = D(\tilde{\theta}_1) - D(\tilde{\theta}_2) - (h_1 - h_2)F. \tag{5.10}$$

By assuming that $h_1 < h_2$, the model k_1 is selected when the difference $\Delta\mathrm{IC}_{12} < 0$, otherwise, i.e. when $\Delta\mathrm{IC}_{12} > 0$, the model k_2 is selected.

We note that both the AIC and BIC require specification of the number of free parameters (model complexity term) of the model. In contrast, as we will see later, this penalty term, called the effective number of the parameters, will be estimated along with the model fit term when using the DIC (Spiegelhalter et al., 2002). These criteria require the availability of a closed form of the likelihood. Therefore, we shall later develop these criteria for HMMs, given a closed form of the likelihood of these models. Next we consider the related DIC.

5.2.2 Deviance information criterion

The deviance information criterion (DIC) was introduced by Spiegelhalter et al. (2002) as a developed version of the AIC from a Bayesian perspective. It is used to measure both the goodness of fit of the model and penalise the model complexity. Spiegelhalter et al. (2002) developed this criterion by introducing the theoretical justification for the concept of effective number of parameters as a measure of the complexity of a model. This criterion is based on the concept of *deviance*. The DIC, as given by Spiegelhalter et al. (2002), can be defined as

$$\text{DIC} = \overline{D(\theta)} + p_{\text{DIC}},$$

where, $\overline{D(\theta)}$, is used as a measure of the goodness of fit and is summarized by the posterior expectation of the deviance,

$$\overline{D(\theta)} = E_{\theta|\mathbf{y}}\left\{D(\theta)\right\} = E_{\theta|\mathbf{y}}\left\{-2\log Pr(\mathbf{y}|\theta)\right\},$$

where $Pr(\mathbf{y}|\theta)$ represents the likelihood of $\mathbf{y}$, given the model parameter θ. The effective number of parameters, p_{DIC}, is used as a measure for model complexity and was originally defined as the difference between the posterior mean of deviance minus the deviance of posterior means, i.e.

$$p_{\text{DIC}} = E_{\theta|\mathbf{y}}\left\{D(\theta)\right\} - D\left\{E_{\theta|\mathbf{y}}(\theta)\right\} = \overline{D(\theta)} - D(\tilde{\theta}),$$

where $\tilde{\theta}$ refers to some generic point estimator of θ which can be one of the justified estimators such as posterior mean, median or mode (Spiegelhalter et al., 2002; Celeux et al., 2006). The DIC can be rewritten as

$$\begin{aligned} \text{DIC} = \overline{D(\theta)} + p_{\text{DIC}} &= \overline{D(\theta)} + \left[\overline{D(\theta)} - D(\tilde{\theta})\right], \\ &= 2\overline{D(\theta)} - D(\tilde{\theta}), \\ &= 2\left[E_{\theta|\mathbf{y}}\left\{-2\log Pr(\mathbf{y}|\theta)\right\}\right] - \left[-2\log Pr(\mathbf{y}|\tilde{\theta})\right], \\ &= -4E_{\theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\theta)\right] + 2\log Pr(\mathbf{y}|\tilde{\theta}), \end{aligned} \tag{5.11}$$

where its the effective number of parameters is

$$p_{\mathrm{DIC}} = -2E_{\theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\theta)\right] + 2\log Pr(\mathbf{y}|\tilde{\theta}). \tag{5.12}$$

Given a set of competing models, a smaller DIC value indicates a good model fit. Unlike AIC and BIC, the penalty term known as the effective number of parameters, p_{DIC}, is estimated by the definition $D(\tilde{\theta})$. Using non-informative priors on the model parameters, Spiegelhalter et al. (2002) concluded that the effective number of parameters, p_{DIC}, can be nearly the same as the actual number of parameters, h, in the AIC.

Based on Spiegelhalter et al. (2002) and Celeux et al. (2006), development of this criterion for latent variable models requires knowing about the focus related to the presence of latent variables $\mathbf{z}$, as well as the availability of the likelihood of these models. Accordingly, we shall later develop this criterion for HMMs. Next we first consider the concept of focus and the likelihood in the context of HMMs .

5.3 The concept of focus and likelihood of HMMs

In order to develop our criteria, this section considers two key aspects. The first aspect is the form of the likelihood and the second aspect relates to the definition of the *focus* used with each form of the likelihood. For latent variable models such as mixture and hidden Markov models, we must define the meaning of missing data introduced in such models. The missing or *augmented* data can be considered as random variables that can take the form of the component membership (independent) or the states (dependent) in mixture and hidden Markov models respectively and thus facilitate the model construction (Frühwirth-Schnatter, 2006). As explained in Chapter (4), the principle of *data augmentation* (Tanner and Wong, 1987) makes MCMC estimation for HMMs easier by introducing the hidden Markov chain, $\mathbf{z}$, as missing data. Thus, the missing data play two roles in HMMs. They can be considered as missing data from a Bayesian modelling viewpoint, and at the same time as hidden states (parameters) inferred through the observational process according to the definition of HMMs. We define the likelihood function of a HMM in Figure (5.1) as a graphical structure, where the hidden states $\mathbf{z}$, are obtained given the parameters related to the unobserved part of the model, namely, $\boldsymbol{\pi}$ and $\mathbf{A}$.

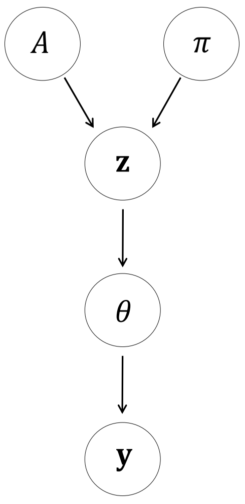


Figure 5.1: The graphical representation of the parameters of HMM.

Conditioning on $\mathbf{z}$, the probability distribution of the observed sequence, $\mathbf{y}$, is obtained given the state-dependent parameter θ. In order to use the DIC for HMMs, the missing data can be dealt with as parameters in focus, along with other model parameters, and these missing data here are named as hidden states. Alternatively, they are treated as missing values being integrated out.

By following the augmented data strategy, a closed form of the likelihood function of HMMs can be obtained as follows. Given a model with parameters $\Theta = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})$ and a sequence of observed data, $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_T)$, augmented with a sequence of missing data, $\mathbf{z} = (\mathrm{z}_1, \mathrm{z}_2, ..., \mathrm{z}_T)$, the joint or complete data distribution can be written as:

$$L_c(\Theta; \mathbf{y}, \mathbf{z}) = Pr(\mathbf{y}, \mathbf{z}|\Theta) = \frac{Pr(\mathbf{y}, \mathbf{z}, \Theta)}{Pr(\Theta)}, \tag{5.13}$$

and

$$\frac{Pr(\mathbf{y}, \mathbf{z}, \Theta)}{Pr(\Theta)} = \frac{Pr(\mathbf{y}, \mathbf{z}, \Theta)}{Pr(\mathbf{z}, \Theta)} . \frac{Pr(\mathbf{z}, \Theta)}{Pr(\Theta)} = Pr(\mathbf{y}|\mathbf{z}, \Theta) Pr(\mathbf{z}|\Theta), \tag{5.14}$$

hence,

$$Pr(\mathbf{y}, \mathbf{z}|\Theta) = Pr(\mathbf{y}|\mathbf{z}, \Theta) Pr(\mathbf{z}|\Theta), \tag{5.15}$$

where $Pr(\mathbf{y}, \mathbf{z}|\Theta)$ denotes the *complete data likelihood* and $Pr(\mathbf{y}|\mathbf{z}, \Theta)$ denotes the *conditional likelihood* multiplied by the density function of hidden variables, $Pr(\mathbf{z}|\Theta)$. The *observed* or *integrated* likelihood function of observations, $Pr(\mathbf{y}|\Theta) = L(\Theta; \mathbf{y})$, is obtained by summing all possible hidden state sequences, in the complete data likelihood,

$$\begin{aligned} Pr(\mathbf{y}|\Theta) &= \sum_{\forall \mathbf{z}} Pr(\mathbf{y}, \mathbf{z}|\boldsymbol{\pi}, \mathbf{A}, \theta), \\ &= \sum_{\forall \mathbf{z}} Pr(\mathbf{y}|\mathbf{z}, \theta) Pr(\mathbf{z}|\boldsymbol{\pi}, \mathbf{A}), \\ &= \sum_{\forall \mathbf{z}} \left[Pr(\mathrm{z}_1|\boldsymbol{\pi}) \prod_{t=2}^{T} Pr(\mathrm{z}_t|\mathrm{z}_{t-1}; \mathbf{A}) \prod_{t=1}^{T} f(\mathrm{y}_t|\mathrm{z}_t, \theta) \right]. \end{aligned} \tag{5.16}$$

Note that in the second line of Equation (5.16), the model parameters were separated. This is due to the fact that the state-specific parameter, θ, does not directly depend on the parameters of the hidden part of the model, $\boldsymbol{\pi}$ and $\mathbf{A}$, but, is affected explicitly by the hidden states (dependence structure), $\mathbf{z}$, that are essentially based on $\boldsymbol{\pi}$ and $\mathbf{A}$ as shown in Figure 5.1.

According to this graphical representation of the model parameters (Figure 5.1), a HMM can be viewed as a hierarchical model. This leads to different versions for the DIC which also take into account different aspects concerning the focus such as whether the hidden state are included, or the availability of the likelihood of the models in closed form. All these considerations will be addressed in the preparation of the proposed criteria in this thesis.

5.4 Modification to AIC and BIC

In conventional HMMs, the AIC and BIC are typically based on the log-likelihood function, $\ell(\hat{\Theta})$, the deviance D; $D = -2(\ell(\hat{\Theta}))$, evaluated at a point estimate, $\hat{\Theta}$, obtained from maximizing the likelihood function using the EM algorithm (Dempster et al., 1977). In this section, we introduce several versions of the AIC and BIC that are based on the observed and conditional log-likelihoods approximated from a Bayesian perspective which considers a new idea in the HMMs context. Developing such expressions are inspired by Brooks (2002, p. 617) who pointed out, in his comments on the article published by Spiegelhalter et al. (2002), that it is possible to obtain approximate estimates for the AIC and BIC based on a deviance evaluated at the posterior draws. Brooks (2002) proposed that the term of model fit in both criteria can be the expected deviance, $\overline{D(\Theta)}$, which is approximated over posterior draws.

In addition to such a proposal, we develop, further, other versions of those criteria for HMMs by evaluating their log-likelihoods at the posterior draws summarized from an MCMC sampling.

As mentioned earlier, these criteria require specification of the number of free parameters or the penalty term. In order to employ such criteria for HMM with different model complexity, we need to determine the number of free parameters, h, of the model. The number of free parameters, h, of a HMM with parameters $\Theta = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})$, is given as (Zucchini and MacDonald, 2009)

$$h = K^2 + sK - 1, \tag{5.17}$$

where K refers to the number of states and s is a single numeric value representing the number of parameters of the underlying distribution of the observation process. For example, $s = 2$ for the Normal distribution (μ and σ^2) and $s = 1$ for the Poisson distribution (λ) (Zucchini and MacDonald, 2009).

5.4.1 Recursive observed likelihood-based AIC and BIC

In this section, we provide modified versions of the AIC and BIC for HMMs based on a recursive or observed likelihood approximated from a Bayesian perspective. By introducing the recursive log-likelihood in closed form into the general definitions of the AIC and BIC provided in the previous section (5.2.1), we introduce three different cases of modified versions of the AIC and BIC. These are referred as AIC$_{rec}$ and BIC$_{rec}$, respectively, as follows:

<u>Case I</u>

$$\begin{aligned}
\text{AIC}_{rec_1} &= E_{\Theta|\mathbf{y}}\left[D_{rec}(\Theta|\mathbf{y})\right] + 2h, \\
&= -2E_{\Theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})\right] + 2h, \\
&= -2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\boldsymbol{\theta}}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})\right] Pr(\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta}|\mathbf{y}, \mathbf{z}) d\boldsymbol{\pi} d\mathbf{A} d\boldsymbol{\theta} + 2h,
\end{aligned} \tag{5.18}$$

$$\begin{aligned}
\text{BIC}_{rec_1} &= E_{\Theta|\mathbf{y}}\left[D_{rec}(\Theta|\mathbf{y})\right] + h\log(T), \\
&= -2E_{\Theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})\right] + h\log(T), \\
&= -2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\boldsymbol{\theta}}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})\right] Pr(\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta}|\mathbf{y}, \mathbf{z}) d\boldsymbol{\pi} d\mathbf{A} d\boldsymbol{\theta} + h\log(T),
\end{aligned} \tag{5.19}$$

where $E_{\Theta|\mathbf{y}}\left[D_{rec}(\Theta|\mathbf{y})\right] = -2E_{\Theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})\right]$ in the above two versions is the expected recursive deviance evaluated at draws from the posterior distribution of all model parameters, $Pr(\pi, \mathbf{A}, \theta|\mathbf{y})$, observed over an MCMC run.

Case II

$$
\begin{aligned}
\mathrm{AIC}_{rec_2} &= D_{rec}(E_{\Theta|\mathbf{y}}(\Theta|\mathbf{y})) + 2h, \\
&= -2\log Pr(\mathbf{y}|E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z}\right]) + 2h, \\
&= -2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\theta} \log Pr(\mathbf{y}|\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\theta})Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\boldsymbol{\pi}d\mathbf{A}d\theta + 2h, \qquad (5.20)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{BIC}_{rec_2} &= D_{rec}(E_{\Theta|\mathbf{y}}(\Theta|\mathbf{y})) + h\log(T), \\
&= -2\log Pr(\mathbf{y}|E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z}\right]) + h\log(T), \\
&= -2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\theta} \log Pr(\mathbf{y}|\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\theta})Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\boldsymbol{\pi}d\mathbf{A}d\theta + h\log(T), \qquad (5.21)
\end{aligned}
$$

where $E_{\Theta|\mathbf{y}}\left[D_{rec}(\Theta|\mathbf{y})\right] = -2E_{\Theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]$ in the above two versions is the expected recursive deviance evaluated at the posterior means of all model parameters summarized from the posterior distribution $Pr(\pi,\mathbf{A},\theta|\mathbf{y})$. These posterior means are marginally approximated from the Gibbs sampler as follows:

$$
\bar{\boldsymbol{\pi}} \approx \frac{1}{M}\sum_{m=1}^{M}\pi_j^{(m)}\ ,\bar{\mathbf{A}} \approx \bar{a}_{jk} = \frac{1}{M}\sum_{m=1}^{M}a_{jk}^{(m)} \text{ and } \bar{\theta} \approx \frac{1}{M}\sum_{m=1}^{M}\theta_j^{(m)}, \text{ for } j,k = 1,2,...,K.
$$

Case III

$$
\begin{aligned}
\mathrm{AIC}_{rec_3} &= E_{\hat{D}_{rec}(.)}\left[D_{rec}(\Theta)\right] + 2h, \\
&= -2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right] + 2h, \\
&= -2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\boldsymbol{\pi}d\mathbf{A}d\theta + 2h, \qquad (5.22)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{BIC}_{rec_3} &= E_{\hat{D}_{rec}(.)}\left[D_{rec}(\Theta)\right] + h\log(T), \\
&= -2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right] + h\log(T), \\
&= -2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\boldsymbol{\pi}d\mathbf{A}d\theta + h\log(T), \qquad (5.23)
\end{aligned}
$$

where $E_{\hat{D}_{rec}(.)}\left[D_{rec}(\Theta)\right] = -2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]$ in both criteria is a minimum expected recursive deviance, $\hat{D}_{rec}(.)$, evaluated at draws from the posterior distribution of all model parameters, $Pr(\pi,\mathbf{A},\theta|\mathbf{y})$, observed over an MCMC run.

We define these three cases. In case **I**, the model fit term in both AIC$_{rec_1}$ and BIC$_{rec_1}$, represents the posterior mean of the recursive deviance. This case is as the same what is proposed by Brooks (2002) for autoregressive models. Furthermore, we contribute in developing the last two versions defined in the cases **II** and **III** as follows. In case **II**, we assume that the model fit term for both versions, AIC$_{rec_2}$ and BIC$_{rec_2}$, represents the recursive deviance evaluated at the plugged-in estimates of the posterior distribution, namely, the posterior means, whereas in the case **III**, the proposed model fit term is inspired by the observed DIC$_3$ introduced by Celeux et al. (2006) who proposed that the model fit term to be a functional estimator (a minimum deviance, or equivalently maximum log-likelihood). Celeux et al. (2006) pointed out that this such an estimator provides more stable evaluations. Furthermore, its density is easily approximated by an MCMC evaluation. This estimator, i.e. a minimum deviance, was also proposed by Richardson (2002) in her discussion of Spiegelhalter et al. (2002). Accordingly, we define the model fit term in both AIC$_{rec_3}$ and BIC$_{rec_3}$ as a minimum recursive deviance observed over an MCMC run.

Further details of the MC approximations for all these versions are provided at the appendix of this chapter.

5.4.2 Conditional likelihood-based AIC and BIC

Given a conditional log-likelihood, it is possible to derive several versions of the fit term of the AIC and BIC. The conditional likelihood-based AIC and BIC will be referred to as AIC$_{con}$ and BIC$_{con}$, respectively. Analogous to criteria based on the recursive observed likelihood, i.e. AICs$_{rec}$ and BICs$_{rec}$, we also introduce three classes of versions of these criteria as follows:

Case I

$$\begin{aligned}\text{AIC}_{con_1} &= E_{\theta,\mathbf{z}}\left[D_{con}(\theta,\mathbf{z})\right]+2h,\\ &= -2E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\theta,\mathbf{z})\right]+2h,\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\theta,\mathbf{z})\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta+2h, \qquad (5.24)\end{aligned}$$

$$\begin{aligned}\text{BIC}_{con_1} &= E_{\theta,\mathbf{z}}\left[D_{con}(\theta,\mathbf{z})\right]+h\log(T),\\ &= -2E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\theta,\mathbf{z})\right]+h\log(T),\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\theta,\mathbf{z})\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta+h\log(T), \qquad (5.25)\end{aligned}$$

where $E_{\theta,\mathbf{z}}[D_{con}(\theta,\mathbf{z})] = -2E_{\theta,\mathbf{z}}[\log Pr(\mathbf{y}|\theta,\mathbf{z})]$ in both criteria is the expected conditional deviance evaluated given draws from the posterior distribution, $Pr(\theta,\mathbf{z}|\mathbf{y})$, of the stale-dependent parameter θ and hidden states $\mathbf{z}$.

Case II

$$\begin{aligned}\text{AIC}_{con_2} &= E_{\theta,\mathbf{z}}\left[D_{con}(\hat{\theta},\hat{\mathbf{z}})\right] + 2h,\\ &= -2E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right] + 2h,\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right] Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + 2h, \end{aligned} \tag{5.26}$$

$$\begin{aligned}\text{BIC}_{con_2} &= E_{\theta,\mathbf{z}}\left[D_{con}(\hat{\theta},\hat{\mathbf{z}})\right] + h\log(T),\\ &= -2E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right] + h\log(T),\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right] Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + h\log(T), \end{aligned} \tag{5.27}$$

where $E_{\theta,\mathbf{z}}\left[D_{con}(\hat{\theta},\hat{\mathbf{z}})\right] = -2E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right]$ in both criteria is the expected conditional deviance, given a joint Maximum a posteriori (MAP) estimator $(\hat{\mathbf{z}},\hat{\theta})$ summarized from the posterior distribution, $Pr(\theta,\mathbf{z}|\mathbf{y})$, of the state-dependent parameters θ and hidden states $\mathbf{z}$. This joint MAP estimator can be approximated by using the best pair among the posterior draws, i.e., the pair that has the highest value of

$$\left(\hat{\mathbf{z}},\hat{\theta}\right) = \underset{\mathbf{z},\Theta}{\operatorname{argmax}}\ Pr(\mathbf{y},\mathbf{z}|\theta)Pr(\mathbf{z}|\boldsymbol{\pi},\mathbf{A})Pr(\theta).$$

Case III

$$\begin{aligned}\text{AIC}_{con_3} &= E_{\hat{D}_{con}(.)}[D_{con}(\theta,\mathbf{z})] + 2h,\\ &= -2E_{\widehat{\log Pr}(.)}[\log Pr(\mathbf{y}|\theta,\mathbf{z})] + 2h,\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\theta,\mathbf{z})\right] Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + 2h, \end{aligned} \tag{5.28}$$

$$\begin{aligned}\text{BIC}_{con_3} &= E_{\hat{D}_{con}(.)}[D_{con}(\theta,\mathbf{z})] + h\log(T),\\ &= -2E_{\widehat{\log Pr}(.)}[\log Pr(\mathbf{y}|\theta,\mathbf{z})] + h\log(T),\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\theta,\mathbf{z})\right] Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + h\log(T), \end{aligned} \tag{5.29}$$

where $E_{\hat{D}_{con}(.)}[D_{con}(\theta,\mathbf{z})] = -2E_{\widehat{\log Pr}(.)}[\log Pr(\mathbf{y}|\theta,\mathbf{z})]$ is a minimum expected conditional deviance, evaluated at draws from the posterior distribution $Pr(\theta,\mathbf{z}|\mathbf{y})$, observed over an MCMC run.

Note that in case **I** we use the expected conditional deviance evaluated over the state-specific parameter, θ, and hidden state, $\mathbf{z}$, as a model fit term of the AIC$_{con_1}$ and BIC$_{con_1}$. In case **II** we use the conditional deviance evaluated at a plugged joint Bayesian estimator: $(\hat{\mathbf{z}},\hat{\theta})$ which can be joint maximum a posteriori (MAP) estimators of $(\mathbf{z},\theta)$. In case **III**, both model fit terms in the AIC$_{con_3}$ and BIC$_{con_3}$ are given a function estimator that represents the minimum conditional deviance value obtained through an MCMC run, given posterior draws of the state-dependent parameter, θ, and hidden states, $\mathbf{z}$. This latter case is based on the same as the idea introduced in the case **III** with respect to the AIC and BIC based on the recursive deviance (sub-section (5.4.1)).

At the end this chapter, we provide all the MC approximations of these versions.

5.5 DIC for HMMs

We now develop several versions of the DIC for HMMs. Specifically, we concentrate mainly on some criteria based on the work of Celeux et al. (2006). Based on the type of the likelihood used, Celeux et al. (2006) introduce three groups of the DIC:

1. The observed DIC which includes three versions, namely the DIC$_1$, DIC$_2$ and DIC$_3$.
2. The complete DIC which includes three versions, namely the DIC$_4$, DIC$_5$ and DIC$_6$.
3. The conditional DIC which involves two versions: DIC$_7$ and DIC$_8$.

We will not investigate here the DIC versions based on the complete likelihood; DIC$_4$, DIC$_5$ and DIC$_6$ as they are logically incoherent with respect to the focus (Li et al., 2015). Celeux et al. (2006) impose different focuses with respect to the latent variables in these versions. In other words, they consider these latent variables as parameters in the first term and at the same time as missing data in the second term for the same criterion. This is also true for the DIC$_8$, where these latent variables are treated as parameters in the first term and at the same time as missing data in the second term in this criterion. Thus, the DIC$_8$ will not be investigated in our work either. Therefore, our work concerns only in investigation the DICs based on observed likelihood as well as the DIC$_7$ as a conditional likelihood-based criterion. In addition, we develop a new conditional version based on a function estimator as an idea inspired by that used with the observed likelihood-based DIC (DIC$_3$).

5.5.1 Recursive DIC

We first define some variations of the observed DIC for HMMs. Given the observed likelihood, $Pr(\mathbf{y}|\Theta)$, in closed form, we can define the DIC$_1$, or DIC$_{rec_1}$, for a HMM as follows

$$\begin{aligned}
\text{DIC}_{rec_1} &= -4E_{\Theta}\left[\log Pr(\mathbf{y}|\Theta)\right] + 2\log Pr(\mathbf{y}|E_{\Theta}\left[\Theta|\mathbf{y},\mathbf{z}\right]),\\
&= -4E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right] + 2\log Pr(\mathbf{y}|E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z}\right]),\\
&= -4\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\theta}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\boldsymbol{\pi}d\mathbf{A}d\theta\\
&+2\int_{\boldsymbol{\pi}}\int_{\mathbf{A}}\int_{\theta}\log Pr(\mathbf{y}|\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\theta})Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\boldsymbol{\pi}d\mathbf{A}d\theta, \qquad (5.30)
\end{aligned}$$

where $Pr(\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z})$ is the joint posterior distribution for all parameters in the HMM, given complete data $(\mathbf{y},\mathbf{z})$. This can be easily broken into marginal posteriors for each parameter of the model as follows:

$$\begin{aligned}
\boldsymbol{\pi} &\sim Pr(\boldsymbol{\pi}|\mathbf{y},\mathbf{z}),\\
\mathbf{A} &\sim Pr(\mathbf{A}|\mathbf{y},\mathbf{z}),
\end{aligned}$$

and the state-dependent parameter as

$$\theta \sim Pr(\theta|\mathbf{y},\mathbf{z}).$$

Posterior draws from these full conditional distributions above can be obtained using MCMC methods such as the Gibbs sampler adopted in this thesis. The joint expectation, $E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[.\right]$, can be partitioned as

$$\begin{aligned}
E_{\Theta}\left[\boldsymbol{\pi}|\mathbf{y},\mathbf{z}\right] &= \int_{\mathbf{z}} Pr(\boldsymbol{\pi}|\mathbf{y},\mathbf{z})d\mathbf{z},\\
E_{\Theta}\left[\mathbf{A}|\mathbf{y},\mathbf{z}\right] &= \int_{\mathbf{z}} Pr(\mathbf{A}|\mathbf{y},\mathbf{z})d\mathbf{z},\\
E_{\Theta}\left[\theta|\mathbf{y},\mathbf{z}\right] &= \int_{\mathbf{z}} Pr(\theta|\mathbf{y},\mathbf{z})d\mathbf{z},
\end{aligned}$$

which can be approximated by averaging their corresponding full conditional posterior distributions obtained from the Gibbs sampler as follows:

$$\bar{\boldsymbol{\pi}} \approx \frac{1}{M}\sum_{m=1}^{M}\pi_j^{(m)}\ ,\bar{\mathbf{A}} \approx \bar{a}_{jk} = \frac{1}{M}\sum_{m=1}^{M}a_{jk}^{(m)}\ \text{ and }\ \bar{\theta} \approx \frac{1}{M}\sum_{m=1}^{M}\theta_j^{(m)},\ \text{ for } j,k = 1,2,...,K.$$

The corresponding effective number of parameters of this version, $p_{\text{DIC}_{rec_1}}$ can be then given by

$$\begin{aligned} p_{\text{DIC}_{rec_1}} &= -2\int_{\pi}\int_{\mathbf{A}}\int_{\theta}[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)]\,Pr(\pi,\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\pi d\mathbf{A}d\theta, \\ &+2\int_{\pi}\int_{\mathbf{A}}\int_{\theta}\log Pr(\mathbf{y}|\bar{\pi},\bar{\mathbf{A}},\bar{\theta})Pr(\pi,\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\pi d\mathbf{A}d\theta. \end{aligned} \tag{5.31}$$

Despite the poor results for DIC_1 as claimed by Celeux et al. (2006), subsequently thought to be due to use of a plugged-in posterior mean, we include this representation of the observed DIC to check its behaviour in the HMM setting.

The DIC_2, as introduced by Celeux et al. (2006) is based on the definition of posterior mode. This version provides unsatisfactory results with respect to its effective number of parameters. We will not include this criterion in this Chapter.

For the DIC_3, Celeux et al. (2006) propose that the focus can be a functional estimator, $\hat{f}(\mathbf{y})$. He pointed out that the functional estimator is the expectation of the mixture density, $Pr(\mathbf{y}|\theta)$, approximated by an MCMC run:

$$\hat{f}(\mathbf{y}) \approx E_{\theta|\mathbf{y}}\left[Pr(\mathbf{y}|\theta)\right],$$

and the DIC_3 and its p_{DIC_3}, therefore, can be given by

$$\text{DIC}_3 = -4E_{\theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\theta)\right] + 2\log\left\{E_{\theta|\mathbf{y}}\left[Pr(\mathbf{y}|\theta)\right]\right\},$$

$$p_{\text{DIC}_3} = -2E_{\theta|\mathbf{y}}\left[\log Pr(\mathbf{y}|\theta)\right] + 2\log\left\{E_{\theta|\mathbf{y}}\left[Pr(\mathbf{y}|\theta)\right]\right\},$$

where the first term is the same as in the DIC_{rec_1}. This version of the DIC was also preferred by Richardson (2002) as the functional estimator, $\hat{f}(\mathbf{y})$, is stable under permutation of the component labels in mixture models. In addition, such an estimator was also used by Gelman et al. (2014) to develop criteria such as the AIC, DIC and WAIC from a predictive perspective.

We expand the version DIC_3, introduced by Celeux et al. (2006), by assuming that the functional estimator is a minimum recursive deviance or equivalently as minus two the recursive maximum log-likelihood obtained at the posterior draws of model parameters. Given that, another observed DIC, therefore, can be developed, denoted by DIC_{rec_2}, as follows

$$\begin{aligned}\mathrm{DIC}_{rec_2} &= -4E_{\Theta}\left[\log Pr(\mathbf{y}|\Theta)\right] + 2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\Theta)\right],\\ &= -4E_{\pi,\mathbf{A},\theta}\left[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)\right] + 2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)\right],\\ &= -4\int_{\pi}\int_{\mathbf{A}}\int_{\theta}\left[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)\right] Pr(\pi,\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\pi d\mathbf{A}d\theta,\\ &+2\int_{\pi}\int_{\mathbf{A}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\pi,\mathbf{A},\theta)\right] Pr(\pi,\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\pi d\mathbf{A}d\theta. \qquad (5.32)\end{aligned}$$

The first term of this version is similar to the first term in the DIC_{rec_1}. The second term, $E_{\widehat{\log Pr}(.)}$, can be readily approximated as a maximum log-likelihood value obtained through an MCMC run, given posterior draws of the model parameters; π, $\mathbf{A}$ and θ. The corresponding effective number of parameters, $p_{\mathrm{DIC}_{rec_2}}$, can be given by

$$\begin{aligned}p_{\mathrm{DIC}_{rec_2}} &= -2E_{\Theta}\left[\log Pr(\mathbf{y}|\Theta)\right] + 2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\Theta)\right],\\ &= -2E_{\pi,\mathbf{A},\theta}\left[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)\right] + 2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)\right],\\ &= -2\int_{\pi}\int_{\mathbf{A}}\int_{\theta}\left[\log Pr(\mathbf{y}|\pi,\mathbf{A},\theta)\right] Pr(\pi,\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\pi d\mathbf{A}d\theta,\\ &+2\int_{\pi}\int_{\mathbf{A}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\pi,\mathbf{A},\theta)\right] Pr(\pi,\mathbf{A},\theta|\mathbf{y},\mathbf{z})d\pi d\mathbf{A}d\theta. \qquad (5.33)\end{aligned}$$

Note that the focus in both versions above is on all model parameters, $(\pi,\mathbf{A},\theta)$, and the hidden states are dealt with as missing data. We compute these versions by using the forward recursion by summing all possible states, given posterior draws of the model parameters. The MC approximations of these two representations are appended at the end of this chapter.

5.5.2 Conditional DIC

We also develop versions of the DIC based on the conditional likelihood, $Pr(\mathbf{y}|\mathbf{z},\Theta)$, for a HMM, where the observations $\mathbf{y}$ are evaluated by conditioning on the hidden states, $\mathbf{z}$, and model parameters, $\Theta = (\pi,\mathbf{A},\theta)$. However, as explained earlier from the graphical representation of HMM parameters shown in Figure (5.1), the state-specific parameter, θ, does not directly depend on the parameters of the hidden part of the model; π and $\mathbf{A}$, but, is affected explicitly by the hidden states (dependence structure), $\mathbf{z}$, that are essentially obtained via integrating out the $\boldsymbol{\pi}$ and $\mathbf{A}$:

$$\begin{aligned} z_1 &\sim \int Pr(z_1|\pi)d\pi, \\ z_t &\sim \int Pr(z_t|z_{t-1},\mathbf{A})d\mathbf{A}. \end{aligned}$$

Consequently, the conditional likelihood of a HMM can be written as $Pr(\mathbf{y}|\mathbf{z},\theta)$. Given this, we can develop conditional likelihood-based DICs for HMMs. We consider version DIC$_7$ introduced by Celeux et al. (2006) because it is more coherent in terms of considering the latent variables as additional parameters in both its terms. This version, as explained by Carlin (2006), considers the latent variables as additional parameters in both terms. Moreover, this version is essentially implemented by WinBUGS as a model selection criterion for models with latent variables.

The DIC$_7$, as defined by Celeux et al. (2006), is given by

$$\text{DIC}_7 = -4E_{\theta,\mathbf{z}}[\log Pr(\mathbf{y}|\mathbf{z},\theta)] + 2\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta}), \tag{5.34}$$

and its p_{DIC_7} as

$$p_{\text{DIC}_7} = -2E_{\theta,\mathbf{z}}[\log Pr(\mathbf{y}|\mathbf{z},\theta)] + 2\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta}), \tag{5.35}$$

where the first term is the expected conditional deviance evaluated over the state-specific parameter, θ, and hidden state, $\mathbf{z}$, whereas the second term is based on a plugged-in joint Bayesian estimator: $(\hat{\mathbf{z}},\hat{\theta})$, which can be the joint maximum a posteriori (MAP) estimators of $(\mathbf{z},\theta)$. However, the MAP estimates, generally, for latent variable models may not be available in closed form and are often approximated by using the best pair among the posterior draws, i.e., the pair that has the highest value of $Pr(\mathbf{y}|\mathbf{z},\theta)Pr(\mathbf{z}|\theta)Pr(\theta)$ (Celeux et al., 2006). Thus, the joint MAP estimator for the state-specific parameter, $\hat{\theta}$, and hidden states, $\hat{\mathbf{z}}$, of HMM can be approximated by

$$(\hat{\mathbf{z}},\hat{\theta}) = \underset{\mathbf{z},\theta}{\text{argmax}}\, Pr(\mathbf{z},\Theta|\mathbf{y}) = \underset{\mathbf{z},\theta}{\text{argmax}}\, Pr(\mathbf{y}|\mathbf{z},\theta)Pr(\mathbf{z}|\boldsymbol{\pi},\mathbf{A})Pr(\theta). \tag{5.36}$$

Accordingly, the first conditional deviance-based version developed for HMMs, and will referred to as DIC$_{con_1}$, can be then approximated as

$$\begin{aligned}\mathrm{DIC}_{con_1} &= -4E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right] + 2\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta}),\\ &= -4\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + 2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta,\end{aligned} \tag{5.37}$$

and,

$$\begin{aligned}p_{\mathrm{DIC}_{con_1}} &= -2E_{\theta,\mathbf{z}}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right] + 2\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta}),\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + 2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\theta})\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta,\end{aligned} \tag{5.38}$$

where $Pr(\theta,\mathbf{z}|\mathbf{y})$ is the joint posterior distribution of the state-specific parameter, θ, and hidden sates, $\mathbf{z}$. We also use a modified version of the DIC$_7$, which we call DIC$_{con_2}$, based on a functional estimator. It is similar to that used with the DIC$_{rec_2}$, but, based on a conditional log-likelihood-based functional estimator. Given that, we can define the DIC$_{con_2}$ as

$$\begin{aligned}\mathrm{DIC}_{con_2} &= -4E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right] + 2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right],\\ &= -4\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + 2\int_{\mathbf{z}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\mathbf{z},\theta)\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta.\end{aligned} \tag{5.39}$$

The first term in Equation (5.39) is the same as the first term in the DIC$_{con_1}$. The second term, $E_{\widehat{\log Pr}(.)}$, can be readily approximated as a minimum conditional deviance or equivalently as a maximum conditional log-likelihood value obtained through an MCMC run, given posterior draws of the state-specific parameter θ, and hidden states, $\mathbf{z}$. The effective number of parameters, $p_{\mathrm{DIC}_{con_2}}$, can be then defined as follows

$$\begin{aligned}p_{\mathrm{DIC}_{con_2}} &= -2E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right] + 2E_{\widehat{\log Pr}(.)}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right],\\ &= -2\int_{\mathbf{z}}\int_{\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta + 2\int_{\mathbf{z}}\int_{\theta}\left[\widehat{\log Pr}(\mathbf{y}|\mathbf{z},\theta)\right]Pr(\theta,\mathbf{z}|\mathbf{y})d\mathbf{z}d\theta,\end{aligned} \tag{5.40}$$

where the first term of this version is the same as the first term used with the DIC$_{con_1}$. We provide all MC approximations of both criteria above at the end of this chapter.

5.6 Widely applicable information criterion (WAIC)

We finally consider the issue of model choice from a predictive viewpoint. Numerous model selection procedures based on the predictive ability of a model are available. We have examined these procedures. One criterion that emerges is the WAIC (widely applicable information criterion), initially proposed by Watanabe (2009). The characteristics of this criterion have not yet been verified in the context of HMMs. We apply this criterion to our model selection problem.

In the next section we will introduce the basic definition of this criterion.

5.6.1 Basic definition of the WAIC

We first define the pointwise predictive density. Consider a sequence of out-of-sample or future data, $\tilde{\mathbf{y}} = (\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_T)$, that have been generated from some predictive distribution. The out-of-sample log-predictive density for a single future observation, as given by Gelman et al. (2014), can be defined as:

$$\log p_{\text{post}}(\tilde{y}_t) = \log E_{\text{post}}\left[Pr(\tilde{y}_t|\theta)\right] = \log \int Pr(\tilde{y}_t|\theta) p_{\text{post}}(\theta|\mathbf{y}) d\theta. \tag{5.41}$$

The second term $p_{\text{post}}(\theta|\mathbf{y})$ on the right-hand side of Equation (5.41) denotes the posterior distribution and $\log p_{\text{post}}(\tilde{y}_t)$ represents the log-predictive density of future point $\tilde{y}_t$ induced by the posterior distribution $p_{\text{post}}(\theta|\mathbf{y})$.

From Equation (5.41) we obtain the expected values of future points $\tilde{y}_t; t = 1, 2, ..., T$. We define the *expected log predictive density* (elpd) as follows:

$$\text{elpd} = E_f\left[\log p_{\text{post}}(\tilde{y}_t)\right] = \int \left[\log p_{\text{post}}(\tilde{y}_t)\right] f(\tilde{y}_t) d\tilde{\mathbf{y}}, \tag{5.42}$$

where $f(.)$ denotes some data distribution. Gelman et al. (2014) noted that the posterior distribution $Pr_{\text{post}}(.)$ is known, but the real data distribution $f(.)$ is unknown and suggested a plugged in estimate for $f(.)$. They define the measure of predictive accuracy for T data points taken one at a time, by summing the expectations in Equation (5.42), as follows:

$$\text{elppd}_{\tilde{\mathbf{y}}} = \sum_{t=1}^{T} E_f\left[\log Pr(\tilde{y}_t|\theta)\right], \tag{5.43}$$

where $\text{elppd}_{\tilde{\mathbf{y}}}$ is called the *expected log pointwise predictive density* of out-of-sample predictive data, $\tilde{\mathbf{y}}$. As explained by Gelman et al. (2014), the quantity $\text{elppd}_{\tilde{\mathbf{y}}}$ in Equation (5.43) cannot be computed directly as the true distribution $f(.)$ is unknown. However, the within-sample data with a bias correction term can be used as an approximation to the elppd$_{\tilde{\mathbf{y}}}$. The log-pointwise predictive density $(\text{lppd}_{\mathbf{y}})$ based on available within-sample data $\mathbf{y}$ is therefore defined as follows:

$$\begin{aligned}\widehat{\text{lppd}}_{\mathbf{y}} = \log \prod_{t=1}^{T} p_{\text{post}}(\mathrm{y}_t) &= \sum_{t=1}^{T} \log E_{\theta}\left[Pr(\mathrm{y}_t|\theta)\right], \\ &= \sum_{t=1}^{T} \log \int Pr(\mathrm{y}_t|\theta)Pr(\theta|\mathbf{y})d\theta, \end{aligned} \tag{5.44}$$

where the integral above can be approximated by integrating out the posterior samples, $\theta^{(m)}$; $m = 1,2,..,M$ of the posterior $Pr(\theta|\mathbf{y})$ from an MCMC run.

The WAIC is an approximation to the out-of-sample expectation given in Equation (5.43) based on the log-pointwise posterior predictive density given in Equation (5.44) after adding an effective number of parameters, or as Gelman et al. (2014) called it, the bias correction, to adjust for overfitting. Gelman et al. (2014) give two definitions of the bias correction, p_{WAIC} based on pointwise calculations. The first definition is given by

$$p_{\text{WAIC}_1} = 2\sum_{t=1}^{T} \left\{\log\left[E_{\theta}Pr(\mathrm{y}_t|\theta)\right] - E_{\theta}\left[\log Pr(\mathrm{y}_t|\theta)\right]\right\}, \tag{5.45}$$

and the second by

$$p_{\text{WAIC}_2} = \sum_{t=1}^{T} V_{\theta}\left[\log Pr(\mathrm{y}_t|\theta)\right], \tag{5.46}$$

where V_{θ} is the variance of individual terms in the log-predictive density summed over the T data points. Gelman et al. (2014) preferred the second version p_{WAIC_2} as it was found to be more stable. We therefore adopt this version in our calculations. The quantity elppd$_{\tilde{\mathbf{y}}}$ in (5.43) can be then written as the difference between the log pointwise predictive density (lppd$_{\mathbf{y}}$) given in Equation (5.44) and one of the effective numbers of parameters defined in Equations (5.45) and (5.46), i.e.

$$\widehat{\text{elppd}}_{\text{WAIC}} = \widehat{\text{lppd}}_{\mathbf{y}} - p_{\text{WAIC}_j};\ j = 1,2. \tag{5.47}$$

The WAIC is then defined as

$$\begin{aligned}\text{WAIC} &= -2\widehat{\text{lppd}}_{\mathbf{y}} + 2p_{\text{WAIC}_j},\\ &= -2\sum_{t=1}^{T}\log E_{\theta}\left[Pr(\text{y}_t|\theta)\right] + 2p_{\text{WAIC}_j};\ j = 1,2,\end{aligned} \tag{5.48}$$

and is thus on the same scale as the information criteria. Next, we develop the WAIC for HMMs by first integrating out the hidden states to obtain the so-called integrated log pointwise predictive density.

5.6.2 The WAIC for HMMs

We now define the integrated pointwise predictive density (ilppd) and WAIC for a HMM. Given a data set, $\mathbf{y} = (\text{y}_1, \text{y}_2, ..., \text{y}_T)$, generated from a HMM with parameters $\Theta = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\theta})$ and a sequence of hidden states $\mathbf{z} = (\text{z}_1, \text{z}_2, ..., \text{z}_T)$, the ilppd can be given by

$$\begin{aligned}\widehat{\text{ilppd}}_{\mathbf{y}} = \log\prod_{t=1}^{T} p_{\text{post}}(\text{y}_t) &= \sum_{t=1}^{T}\log E_{\mathbf{z},\theta}\left[Pr(\text{y}_t|\mathbf{z},\theta)|\mathbf{y}\right],\\ &= \sum_{t=1}^{T}\log\int_{\mathbf{z}}\int_{\theta} Pr(\text{y}_t|\text{z}_t,\theta)Pr(\mathbf{z},\theta|\mathbf{y})d\mathbf{z}d\theta,\end{aligned} \tag{5.49}$$

which is obtained by integrating out the state-dependent parameter, θ, and hidden states, $\mathbf{z}$. In Equation (5.49), $Pr(\text{y}_t|\mathbf{z},\theta)$, represents the pointwise predictive density of point data, y_t, given the hidden states z_t and state-specific parameter, θ, weighted by the joint posterior distribution, $Pr(\mathbf{z},\theta|\mathbf{y})$, of the model parameters. Hence, by integrating individually over each hidden state z_t and the parameter θ, we can obtain the integrated pointwise predictive density of each data point, y_t. It can be noted that the focus here is on the state-specific parameter, θ, and the hidden states, $\mathbf{z}$. As explained earlier, through the graphical representation of HMM parameters (Figure 5.1), the observations, $\mathbf{y}$, do not depend directly on the parameters of the hidden part: $\boldsymbol{\pi}$ and $\mathbf{A}$, but, they depend instead on the hidden states, $\mathbf{z}$, that are essentially obtained from those parameters. The integrated log-pointwise predictive density in Equation (5.49) can be approximated by the posterior samples of the model parameters over an MCMC run. As mentioned earlier, we adopt the second version of the effective number of parameters, p_{WAIC_2}. We will refer to it here as $p_{\text{WAIC}_{var}}$, defined as

$$p_{\text{WAIC}_{var}} = \sum_{t=1}^{T} V_{\mathbf{z},\theta}\left[\log Pr(\text{y}_t|\mathbf{z},\theta)\right], \tag{5.50}$$

where $V_{\mathbf{z},\theta}$ is the variance of individual terms in the integrated log predictive pointwise density summed over the T data points. We can thus define the WAIC for a HMM as

$$\begin{aligned}\text{WAIC} &= -2\widehat{\text{ilppd}}_{\mathbf{y}} + 2p_{\text{WAIC}_{var}},\\ &= -2\sum_{t=1}^{T} \log E_{\mathbf{z},\theta}\left[Pr(\mathrm{y}_t|\mathbf{z},\theta)\right] + 2p_{\text{WAIC}_{var}}.\end{aligned} \tag{5.51}$$

We provide MC approximations of the ilppd, p_{WAIC} and WAIC at the appendix of this chapter.

5.7 Sampling variability in selection criteria

Since the criteria developed in this chapter are based on the output of an MCMC sampling, they will be subject to simulation variability. Therefore, we measure the variability of these criteria by following the idea proposed by Zhu and Carlin (2000). This is a "brute force" approach to check the stability of the DIC via computing the variance of DIC, var(DIC), which it is estimated by its sample variance

$$s^2(\text{DIC}) = \frac{1}{L-1}\sum_{l=1}^{L}(\text{DIC}_l - \overline{\text{DIC}})^2,$$

where L denotes the number of independent MCMC runs. Nevertheless, this approach is computationally expensive as it requires several runs. We also apply this approach for all criteria considered in this chapter.

5.8 Summary

In this chapter we have introduced well-known likelihood-based criteria, namely the AIC, BIC and DIC, for model selection in a HMM context, assuming an application requiring a fixed but unknown number of the states. We have used the data augmentation approach, where the parameter space is extended by adding hidden data for the unknown states. This provides several closed forms of the likelihood for the HMM, namely, the recursive (observed), complete data and conditional likelihood. Hence, it gave rise to define several versions of those criteria. More specifically, we have extended the original definitions of AIC and BIC, which use traditionally the classical approach, using the Bayesian principle, given two types of likelihood functions, namely the conditional and recursive likelihood. We introduced three cases for each criterion. In the first case, we introduced two versions that are called the AIC$_{rec_1}$ and BIC$_{rec_1}$, where the term of model fit for both criteria is the expected recursive deviance evaluated at the posterior samples of the model parameters. These versions are inspired by Brooks (2002), who applied such versions to autoregressive models. The second and third cases of each criterion are new applications in the HMMs context. In the second case, they are called as the AIC$_{rec_2}$ and BIC$_{rec_2}$, we proposed that the term of model fit of these two criteria is the recursive deviance evaluated at the posterior means of the model parameters. In the last case, they are called as the AIC$_{rec_3}$ and BIC$_{rec_3}$, we proposed that such versions are based on a minimum recursive deviance observed through an MCMC run. Given the conditional likelihood in the closed form, we also introduced three versions of the AIC and BIC. In the first case, we denoted these criteria as the AIC$_{con_1}$ and BIC$_{con_1}$, the term of model fit is the expected conditional deviance evaluated at the posterior draws of the model parameters. These two versions are based on the same as the idea proposed by Brooks (2002). The second and third cases of each criterion are new applications in the HMMs context. In the second case, the criteria are referred to as the AIC$_{con_2}$ and BIC$_{con_2}$, we proposed that the term of model fit of these two criteria is the expected conditional deviance, given a joint Maximum a posteriori (MAP) estimator of the state-dependent parameters θ and hidden states $\mathbf{z}$. In the third case, they are denoted as the AIC$_{con_3}$ and BIC$_{con_3}$, we proposed that such versions are based on a minimum conditional deviance observed through an MCMC run.

In addition, we have introduced several versions of the original DIC. We have constructed these versions based on the type of likelihood and the concept of focus as proposed by Celeux et al. (2006). Firstly, we introduced two versions of the DIC based on the recursive deviance, namely, the DIC$_{rec_1}$ and DIC$_{rec_2}$. The first version, DIC$_{rec_1}$ based on the posterior recursive deviance mean, is the same as the observed DIC$_1$ proposed by Celeux et al. (2006). The second version,

DIC_{rec_2}, is a new modified version of the observed DIC_3 introduced by Celeux et al. (2006). In this latter version, we proposed that the focus is based on a minimum recursive deviance observed through an MCMC run, as a function estimator that differs from what is introduced by Celeux et al. (2006) which was the expected density function. On the other hand, given a conditional deviance obtained through an MCMC run, we also introduced two versions of the DIC based on the conditional deviance, namely, the DIC_{con_1} and DIC_{con_2}. The first version, DIC_{con_1}, is the same as the conditional version DIC_7 proposed by Celeux et al. (2006), where the focus is the joint MAP estimator approximated using the best vector of state-specific parameters and hidden states of the model. The second version, DIC_{con_2}, is a new modified version of the DIC_7, where the focus is based on a functional estimator that is a minimum conditional deviance observed through an MCMC run.

Finally, we have considered the model selection issue from a predictive perspective. In this aspect, we contributed in applying a new criterion in the HMMs context, the so-called widely applicable information (WAIC) (Watanabe, 2009) which considers, for our knowledge, a new application for the HMMs till writing this thesis.

Appendix

5.9 Approximations of the model selection criteria

5.9.1 Approximations of the recursive likelihood-based criteria: AICs$_{rec}$, BICs$_{rec}$ and DICs$_{rec}$

Given posterior draws $\left\{\left(\pi_k^{(m)}, \mathbf{A}^{(m)} = \left\{a_{jk}^{(m)}\right\}, \mu_k^{(m)}, \sigma_k^{2(m)}\right);\ m = 1,2,...,M;\ j,k = 1,2,...,K\right\}$ simulated from the joint posterior distribution, $Pr(\mathbf{z}, \boldsymbol{\pi}, \mathbf{A}, \mu, \sigma^2|\mathbf{y})$, of K-state Normal HMM, the integrals for the criteria: AICs$_{rec}$, BICs$_{rec}$ and DICs$_{rec}$, can be approximated as follows

DIC$_{rec_1}$:

The first term is approximated as

$$E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right] \approx \frac{1}{M}\sum_{m=1}^{M}\log Pr(\mathbf{y}|\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2(m)}),$$
$$\approx \frac{1}{M}\sum_{m=1}^{M}\left\{\log\sum_{\forall\mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2(m)})\right]\right\} \tag{5.52}$$

where the quantity in bracket, $\{.\}$, represents the m^{th} recursive log-likelihood values computed using the forward algorithm. Given the marginal posterior means of the model parameters, $\bar{\boldsymbol{\pi}}, \bar{\mathbf{A}}, \bar{\mu}$ and $\bar{\sigma}^2$, summarized from their full conditional distributions

$$\bar{\boldsymbol{\pi}} = E_{\Theta}\left[\boldsymbol{\pi}|\mathbf{y},\mathbf{z}\right] \approx \frac{1}{M}\sum_{m=1}^{M}\pi_j^{(m)},$$
$$\bar{\mathbf{A}} = E_{\Theta}\left[\mathbf{A}|\mathbf{y},\mathbf{z}\right] = \bar{a}_{jk} \approx \frac{1}{M}\sum_{m=1}^{M}a_{jk}^{(m)},$$
$$\bar{\mu} = E_{\Theta}\left[\mu|\mathbf{y},\mathbf{z}\right] \approx \frac{1}{M}\sum_{m=1}^{M}\mu_j^{(m)},$$
$$\bar{\sigma}^2 = E_{\Theta}\left[\sigma|\mathbf{y},\mathbf{z}\right] \approx \frac{1}{M}\sum_{m=1}^{M}\sigma_j^{2(m)},\ \text{ for } j,k = 1,2,...,K,$$

the second term of DIC$_{rec_1}$ is given by

$$\log Pr(\mathbf{y}|E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z}\right]) = \log Pr(\mathbf{y}|\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\mu},\bar{\sigma}^2),$$
$$= \log\sum_{\forall\mathbf{z}}\left[Pr(\mathrm{z}_1|\bar{\boldsymbol{\pi}})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\bar{\mathbf{A}})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\bar{\mu},\bar{\sigma}^2)\right],$$

which requires only one recursive step using the forward algorithm. As a result, the DIC$_{rec_1}$ can be written as

$$\mathrm{DIC}_{rec_1} = \frac{-4}{M}\sum_{m=1}^{M}\left\{\log\sum_{\forall \mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}$$
$$+2\log\sum_{\forall \mathbf{z}}\left[Pr(\mathrm{z}_1|\bar{\boldsymbol{\pi}})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\bar{\mathbf{A}})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\bar{\mu},\bar{\sigma}^2)\right]. \tag{5.53}$$

<u>DIC$_{rec_2}$</u>:

The first term of the DIC$_{rec_2}$ is the same as in DIC$_{rec_1}$, whereas the second term can be approximated as the maximum recursive log-likelihood obtained across M iterations in an MCMC run as follows

$$E_{\widehat{\log p}(.)}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\boldsymbol{\theta})\right] = \underset{\ell(\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}})}{\mathrm{argmax}}\left\{\log\textstyle\sum_{\forall \mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\},$$

where

$$\ell(\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}}) = \log\sum_{\forall \mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right],$$

is the m^{th} log-likelihood evaluated recursively of the K-state Normal HMM. Hence, the DIC$_{rec_2}$ can be written as follows:

$$\mathrm{DIC}_{rec_2} = \frac{-4}{M}\sum_{m=1}^{M}\left\{\log\sum_{\forall \mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}$$
$$+2\underset{\ell(\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}})}{\mathrm{argmax}}\left\{\log\sum_{\forall \mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}. \tag{5.54}$$

<u>AICs$_{rec}$ and BICs$_{rec}$</u>:

The recursive likelihood based AIC and BIC, namely, AIC$_{rec_1}$, BIC$_{rec_1}$, AIC$_{rec_2}$, BIC$_{rec_2}$, AIC$_{rec_3}$, BIC$_{rec_3}$, can be easily approximated as their fit model terms are already approximated in the DIC$_{rec_1}$ and DIC$_{rec_2}$. Given the number of free parameters, h, which for a K-state Normal HMM is computed as: $h = K^2 + 2K - 1$, the versions of the AICs$_{rec}$ and BICs$_{rec}$ can be written as follows:

$$\begin{aligned}
\mathrm{AIC}_{rec_1} &= -2E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]+2h,\\
&= \frac{-2}{M}\sum_{m=1}^{M}\log Pr(\mathbf{y}|\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}})+2h,\\
&= \frac{-2}{M}\sum_{m=1}^{M}\left\{\log\sum_{\forall\mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}+2h,
\end{aligned} \tag{5.55}$$

and

$$\begin{aligned}
\mathrm{BIC}_{rec_1} &= -2E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]+h\log(T),\\
&= \frac{-2}{M}\sum_{m=1}^{M}\log Pr(\mathbf{y}|\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}})+h\log(T),\\
&= \frac{-2}{M}\sum_{m=1}^{M}\left\{\log\sum_{\forall\mathbf{z}}\left[Pr(\mathrm{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}+h\log(T),
\end{aligned} \tag{5.56}$$

where the fit model term in both versions above is the same as first term in the DIC_{rec_1}. On the other hand, the AIC_{rec_2} and BIC_{rec_2}, respectively, are based on a conditional deviance evaluated at posterior point estimates of the model parameters $\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\mu}$ and $\bar{\sigma}^2$, summarized from their full conditional distributions. Thus, these versions are then approximated as

$$\begin{aligned}
\mathrm{AIC}_{rec_2} &= -2\log Pr(\mathbf{y}|E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z}\right])+2h,\\
&= -2\log Pr(\mathbf{y}|\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\mu},\bar{\sigma}^2)+2h,\\
&\approx -2\log\sum_{\forall\mathbf{z}}\left[Pr(\mathrm{z}_1|\bar{\boldsymbol{\pi}})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\bar{\mathbf{A}})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\bar{\mu},\bar{\sigma}^2)\right]+2h,
\end{aligned} \tag{5.57}$$

and

$$\begin{aligned}
\mathrm{BIC}_{rec_2} &= -2\log Pr(\mathbf{y}|E_{\boldsymbol{\pi},\mathbf{A},\theta}\left[\boldsymbol{\pi},\mathbf{A},\theta|\mathbf{y},\mathbf{z}\right])+h\log(T),\\
&= -2\log Pr(\mathbf{y}|\bar{\boldsymbol{\pi}},\bar{\mathbf{A}},\bar{\mu},\bar{\sigma}^2)+h\log(T),\\
&\approx -2\log\sum_{\forall\mathbf{z}}\left[Pr(\mathrm{z}_1|\bar{\boldsymbol{\pi}})\prod_{t=2}^{T}Pr(\mathrm{z}_t|\mathrm{z}_{t-1};\bar{\mathbf{A}})\prod_{t=1}^{T}\phi(\mathrm{y}_t|\mathrm{z}_t,\bar{\mu},\bar{\sigma}^2)\right]+h\log(T).
\end{aligned} \tag{5.58}$$

This common fit model term of the version above is similar to the second term of the DIC_{rec_1}. The last two versions of AIC and BIC, namely, AIC_{rec_3} and BIC_{rec_3} have a fit model term approximated as maximum recursive log-likelihood obtained across M iteration of the Gibbs sampler. This approximation is the same as the second term of DIC$_{rec_2}$ defined earlier. As a result, the AIC$_{rec_3}$ and BIC$_{rec_3}$ can then be written, respectively, as follows:

$$\begin{aligned}\text{AIC}_{rec_3} &= -2E_{\widehat{\log p}(.)}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]+2h,\\ &\approx -2 \underset{\ell(\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}})}{\text{argmax}} \left\{\log\sum_{\forall\mathbf{z}}\left[Pr(\text{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\text{z}_t|\text{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\text{y}_t|\text{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}+2h.\end{aligned}$$

$$\begin{aligned}\text{BIC}_{rec_3} &= -2E_{\widehat{\log p}(.)}\left[\log Pr(\mathbf{y}|\boldsymbol{\pi},\mathbf{A},\theta)\right]+h\log(T),\\ &\approx -2 \underset{\ell(\boldsymbol{\pi}^{(m)},\mathbf{A}^{(m)},\mu^{(m)},\sigma^{2^{(m)}})}{\text{argmax}} \left\{\log\sum_{\forall\mathbf{z}}\left[Pr(\text{z}_1|\boldsymbol{\pi}^{(m)})\prod_{t=2}^{T}Pr(\text{z}_t|\text{z}_{t-1};\mathbf{A}^{(m)})\prod_{t=1}^{T}\phi(\text{y}_t|\text{z}_t,\mu^{(m)},\sigma^{2^{(m)}})\right]\right\}+h\log(T).\end{aligned}$$

5.9.2 Approximations of the conditional likelihood-based criteria: AICs$_{con}$, BICs$_{con}$ and DICs$_{con}$

Given posterior draws of hidden states, $\mathbf{z}^{(m)} = (\text{z}_1^{(m)},\text{z}_2^{(m)},...,\text{z}_T^{(m)})$ and state-specific model parameters $\left(\mu_k^{(m)},\sigma_k^{2^{(m)}}\right)$; $m=1,2,...,M$; $k=1,2,...,K$, simulated from the joint posterior distribution, $Pr(\mathbf{z},\mu,\sigma^2|\mathbf{y})$, of K-state Normal HMM, MC approximations of the integrals concerning the criteria: AICs$_{con}$, BICs$_{con}$ and DICs$_{con}$, are given as follows:

DIC$_{con_1}$:

The first term of this version can be approximated, where hidden states are expressed as integer parameters, $(\text{z}_1,\text{z}_2,...,\text{z}_T)$, sampled from a multinomial distribution, as

$$\overline{D_{con}(\mathbf{z},\mu,\sigma^2)} = -2E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\mu,\sigma^2)\right] \approx \frac{-2}{M}\sum_{m=1}^{M}\sum_{t=1}^{T}\log\phi(\text{y}_t|\mu^{(m)}_{\text{z}_t^{(m)}},\sigma^{2\;(m)}_{\text{z}_t^{(m)}}). \tag{5.59}$$

The second term of DIC$_{con_1}$ is based on a plugged-in estimator, $(\{\hat{\mathbf{z}}\},\{\hat{\mu}\},\{\hat{\sigma}^2\})$, that is the joint MAP of vector $(\{\mathbf{z}\},\{\mu\},\{\sigma^2\})$, which can be approximated as the vector corresponding to the maximum posterior density, i.e.,

$$(\{\hat{\mathbf{z}}\},\{\hat{\mu}\},\{\hat{\sigma}^2\}) = \underset{\mathbf{z},\mu,\sigma^2}{\text{argmax}} Pr(\mathbf{y},\mathbf{z}|\mu,\sigma^2)Pr(\mathbf{z}|\boldsymbol{\pi},\mathbf{A})Pr(\boldsymbol{\pi})Pr(\mathbf{A})Pr(\mu)Pr(\sigma^2),$$

where $Pr(\mu)$ and $Pr(\sigma^2)$ are priors on the mean and variance of the model, respectively, whereas the $Pr(\boldsymbol{\pi})$ and $Pr(\mathbf{A})$ are priors on the hidden part, $\mathbf{z}$. In post-processing, we compute the highest posterior density at each iteration, given the posterior sample of the model parameters. We then select the index of highest density among all highest posterior densities computed above. Hence, the joint MAP estimator is approximated using the best vector $(\hat{\mathbf{z}}, \hat{\mu}, \hat{\sigma}^{\mathbf{2}})$ corresponding to the index of highest posterior density.

The second term then is given by

$$E_{\mathbf{z},\theta}\left[\log\phi(\mathbf{y}|\hat{\mathbf{z}},\hat{\mu},\hat{\sigma}^2)\right] = \sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\hat{\mu}_{\hat{z}_t},\hat{\sigma}^2_{\hat{z}_t}),$$

where $\theta = (\mu, \sigma^2)$. The DIC$_{con_1}$ can then be written as

$$\mathrm{DIC}_{con_1} \approx \frac{-4}{M}\sum_{m=1}^{M}\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}}) + 2\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\hat{\mu}_{\hat{z}_t},\hat{\sigma}^2_{\hat{z}_t}). \tag{5.60}$$

DIC$_{con_2}$:

The first term to this criterion is the same as the first term approximated in the DIC$_{con_2}$. In contrast, the second term is approximated as the maximum conditional log-likelihood evaluated at the posterior draws of the hidden states, $\mathbf{z}$, and state-specific mean and variance, $\mu_{\mathrm{z}_t}, \sigma^2_{\mathrm{z}_t}$, as follows

$$E_{\log p(.)}\left[\log\phi(\mathbf{y}|\mathbf{z},\theta)\right] \approx \underset{\widehat{\log Pr(.)}}{\mathrm{argmax}}\left[\log\phi(\mathbf{y}|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}})\right].$$

Hence, the DIC$_{con_2}$ can be written as

$$\mathrm{DIC}_{con_2} \approx \frac{-4}{M}\sum_{m=1}^{M}\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}}) + 2\underset{\widehat{\log Pr(.)}}{\mathrm{argmax}}\left[\log Pr(\mathbf{y}|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}})\right] \tag{5.61}$$

AICs$_{con}$ and BICs$_{con}$:

The versions of conditional likelihood-based AIC and BIC, namely, AIC$_{con_1}$, BIC$_{con_1}$, AIC$_{con_2}$, BIC$_{con_2}$, AIC$_{con_3}$, BIC$_{con_3}$, can also be easily approximated as they include fit model terms which are similar to those approximated in the DIC$_{con_1}$ and DIC$_{con_2}$. Given the number of free parameters, $h = K^2 + 2K - 1$, of a K-state Normal HMM, the approximated versions of both criteria can be given, respectively, by

$$\begin{aligned}\mathrm{AIC}_{con_1} &= -2E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\mu,\sigma^2)\right] + 2h,\\ &\approx \frac{-2}{M}\sum_{m=1}^{M}\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}}) + 2h,\end{aligned} \tag{5.62}$$

and

$$\begin{aligned}\text{BIC}_{con_1} &= -2E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\mathbf{z},\mu,\sigma^2)\right] + h\log(T),\\ &\approx \frac{-2}{M}\sum_{m=1}^{M}\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}}) + h\log(T),\end{aligned} \tag{5.63}$$

where the fit model term in both criteria above is the same as the first term of the DIC$_{con_1}$. The AIC$_{con_2}$ and BIC$_{con_2}$ approximated below include a fit model term which is the same as the second term of the DIC$_{con_1}$. Thus, the AIC$_{con_2}$ and BIC$_{con_2}$ can be written as

$$\begin{aligned}\text{AIC}_{con_2} &= -2E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\mu},\hat{\sigma}^2)\right] + 2h,\\ &= -2\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\hat{\mu}_{\hat{\mathrm{z}}_t},\hat{\sigma}^2_{\hat{\mathrm{z}}_t}) + 2h,\end{aligned} \tag{5.64}$$

$$\begin{aligned}\text{BIC}_{con_2} &= -2E_{\mathbf{z},\theta}\left[\log Pr(\mathbf{y}|\hat{\mathbf{z}},\hat{\mu},\hat{\sigma}^2)\right] + h\log(T),\\ &= -2\sum_{t=1}^{T}\log\phi(\mathrm{y}_t|\hat{\mu}_{\hat{\mathrm{z}}_t},\hat{\sigma}^2_{\hat{\mathrm{z}}_t}) + h\log(T),\end{aligned} \tag{5.65}$$

where $(\{\hat{\mathbf{z}}\},\{\hat{\mu}\},\{\hat{\sigma}^2\})$, is the joint MAP estimator of vector $(\{\mathbf{z}\},\{\mu\},\{\sigma^2\})$ which has already been used in approximating the second term of the the DIC$_{con_1}$. The versions of AIC and BIC, namely, AIC$_{con_3}$ and BIC$_{con_3}$ are based on the fit model term which is the same as the second term in the DIC$_{con_2}$. Thus, these versions can be written, respectively, as follows:

$$\begin{aligned}\text{AIC}_{con_3} &= -2E_{\log p(.)}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right] + 2h,\\ &\approx -2\underset{\widehat{\log Pr(.)}}{\operatorname{argmax}}\left[\log Pr(\mathbf{y}|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}})\right] + 2h,\end{aligned} \tag{5.66}$$

and

$$\begin{aligned}\text{BIC}_{con_3} &= -2E_{\log p(.)}\left[\log Pr(\mathbf{y}|\mathbf{z},\theta)\right] + h\log(T),\\ &\approx -2\underset{\widehat{\log Pr(.)}}{\operatorname{argmax}}\left[\log\phi(\mathbf{y}|\mu^{(m)}_{\mathrm{z}_t^{(m)}},\sigma^{2\ (m)}_{\mathrm{z}_t^{(m)}})\right] + h\log(T).\end{aligned} \tag{5.67}$$

5.9.3 Approximations of the ilppd, p_{WAIC} and WAIC

Given the posterior draws of hidden states, $\mathbf{z}^{(m)} = (\mathrm{z}_1^{(m)}, \mathrm{z}_2^{(m)}, ..., \mathrm{z}_T^{(m)})$ and the state-specific model parameters $\left(\mu_k^{(m)}, \sigma_k^{2(m)}\right)$; $m = 1,2,...,M$; $k = 1,2,...,K$, simulated from the joint posterior distribution, $Pr(\mathbf{z}, \mu, \sigma^2|\mathbf{y})$, of K-state Normal HMM, the integrated log-pointwise predictive density, the effective number of parameters and WAIC can be approximated as follows:

$$\begin{aligned}\widehat{\text{ilppd}}_{\mathbf{y}} &= \sum_{t=1}^{T} \log E_{\mathbf{z},\mu,\sigma^2}\left[\phi(\mathrm{y}_t|\mathbf{z},\mu,\sigma^2)|\mathbf{y}\right], \\ &\approx \sum_{t=1}^{T} \log\left(\frac{1}{M}\sum_{m=1}^{M} Pr(\mathrm{y}_y|\mu_{\mathrm{z}_t^{(m)}}^{(m)}, \sigma_{\mathrm{z}_t^{(m)}}^{2\;\;(m)})\right).\end{aligned} \tag{5.68}$$

The effective numbers of parameters, $p_{\text{WAIC}_{var}}$, can be given by

$$\begin{aligned}p_{\text{WAIC}_{var}} &= \sum_{t=1}^{T} V_{\mathbf{z},\mu,\sigma^2}\left[\log\phi(\mathrm{y}_t|\mathbf{z},\mu,\sigma^2)\right], \\ &\approx \sum_{t=1}^{T} V_{m=1}^{M} \log\phi(\mathrm{y}_t|\mu_{\mathrm{z}_t^{(m)}}^{(m)}, \sigma_{\mathrm{z}_t^{(m)}}^{2\;\;(m)}),\end{aligned} \tag{5.69}$$

where $V_{m=1}^{M}$ denotes the sample variance, $V_{m=1}^{M} x_m = \frac{1}{M-1}\sum_{m=1}^{M}(x_m - \bar{x})^2$. The WAIC, therefore, can be given by

$$\begin{aligned}\text{WAIC} &= -2\sum_{t=1}^{T} \log E_{\mathbf{z},\theta}\left[\phi(\mathrm{y}_t|\mathbf{z},\mu,\sigma^2)|\mathbf{y}\right] + 2p_{\text{WAIC}_{var}}, \\ &\approx -2\sum_{t=1}^{T} \log\left(\frac{1}{M}\sum_{m=1}^{M} \phi(\mathbf{y}|\mu_{\mathrm{z}_t^{(m)}}^{(tm)}, \sigma_{\mathrm{z}_t^{(m)}}^{2\;\;(tm)})\right) + 2p_{\text{WAIC}_{var}}.\end{aligned} \tag{5.70}$$

5.10 Computational relationships between proposed criteria; AIC, BIC and DIC

As the proposed criteria in this chapter are based on some similarly parameterized deviances, some computational asymptotic relationships between those criteria can be explained.

5.10.1 Computational relations between the recursive likelihood-based criteria: AICs$_{rec}$, BICs$_{rec}$ and DICs$_{rec}$

By looking at the fit terms of all proposed versions of AICs$_{rec}$ and BICs$_{rec}$, we can clearly see that these terms form the main components of the DIC$_{rec_1}$ and DIC$_{rec_2}$. As the penalty terms,

$2h$, of the AICs$_{rec}$ and also, $h\log(T)$ of the BICs$_{rec}$, where $h = K^2 + sK + 1$, are fixed and known a priori, let

$$H_1 = 2h, \tag{5.71}$$

and

$$H_2 = h\log(T). \tag{5.72}$$

Thus, the fit terms of all versions of the AICs$_{rec}$ and BICs$_{rec}$ can be obtained as

$$\begin{aligned}
\text{AIC}_{rec_1} &= \overline{D_{rec}(\Theta)} + H_1 \longleftrightarrow \overline{D_{rec}(\Theta)} = \text{AIC}_{rec_1} - H_1. \\
\text{BIC}_{rec_1} &= \overline{D_{rec}(\Theta)} + H_2 \longleftrightarrow \overline{D_{rec}(\Theta)} = \text{BIC}_{rec_1} - H_2.
\end{aligned} \tag{5.73}$$

$$\begin{aligned}
\text{AIC}_{rec_2} &= D_{rec}(\bar{\Theta}) + H_1 \longleftrightarrow D_{rec}(\bar{\Theta}) = \text{AIC}_{rec_2} - H_1. \\
\text{BIC}_{rec_2} &= D_{rec}(\bar{\Theta}) + H_2 \longleftrightarrow D_{rec}(\bar{\Theta}) = \text{BIC}_{rec_2} - H_2.
\end{aligned} \tag{5.74}$$

$$\begin{aligned}
\text{AIC}_{rec_3} &= \hat{D}_{rec}(\Theta) + H_1 \longleftrightarrow \hat{D}_{rec}(\Theta) = \text{AIC}_{rec_2} - H_1. \\
\text{BIC}_{rec_3} &= \hat{D}_{rec}(\Theta) + H_2 \longleftrightarrow \hat{D}_{rec}(\Theta) = \text{BIC}_{rec_2} - H_2.
\end{aligned} \tag{5.75}$$

From Equations (5.73) and (5.74), and according to the definition of DIC$_{rec_1}$ and its corresponding effective number of parameters, $p_{\text{DIC}_{rec_1}}$, given in subsection (5.5.1), we conclude that the DIC$_{rec_1}$ and $p_{\text{DIC}_{rec_1}}$ can be computationally obtained by either

$$\begin{aligned}
\text{DIC}_{rec_1} &= 2\overline{D_{rec}(\Theta)} - D_{rec}(\bar{\Theta}), \\
&= 2(\text{AIC}_{rec_1} - H_1) - (\text{AIC}_{rec_2} - H_1), \\
&= 2\text{AIC}_{rec_1} - \text{AIC}_{rec_2} - H_1,
\end{aligned} \tag{5.76}$$

$$\begin{aligned}
p_{\text{DIC}_{rec_1}} &= \overline{D_{rec}(\Theta)} - D_{rec}(\bar{\Theta}), \\
&= (\text{AIC}_{rec_1} - H_1) - (\text{AIC}_{rec_2} - H_1), \\
&= \text{AIC}_{rec_1} - \text{AIC}_{rec_2},
\end{aligned} \tag{5.77}$$

or,

$$\begin{aligned}
\mathrm{DIC}_{rec_1} &= 2\overline{D_{rec}(\Theta)} - D_{rec}(\bar{\Theta}),\\
&= 2(\mathrm{BIC}_{rec_1} - H_2) - (\mathrm{BIC}_{rec_2} - H_2),\\
&= 2\mathrm{BIC}_{rec_1} - \mathrm{BIC}_{rec_2} - H_2, \qquad (5.78)\\
p_{\mathrm{DIC}_{rec_1}} &= \overline{D_{rec}(\Theta)} - D_{rec}(\bar{\Theta}),\\
&= (\mathrm{BIC}_{rec_1} - H_2) - (\mathrm{BIC}_{rec_2} - H_2),\\
&= \mathrm{BIC}_{rec_1} - \mathrm{BIC}_{rec_2}, \qquad (5.79)
\end{aligned}$$

On the other hand, the DIC_{rec_2} is also obtained, in particular, based on the AIC_{rec_1}, BIC_{rec_1}, AIC_{rec_2} and BIC_{rec_2}. Given the fit terms of the obtained in Equations (5.73) and (5.75), and according to the definitions of DIC_{rec_2} and $p_{\mathrm{DIC}_{rec_2}}$ given in section (5.5.1), we can write the computational relationship between these criteria as follows

$$\begin{aligned}
\mathrm{DIC}_{rec_2} &= 2\overline{D_{rec}(\Theta)} - \hat{D}_{rec}(\Theta),\\
&= 2(\mathrm{AIC}_{rec_1} - H_1) - (\mathrm{AIC}_{rec_3} - H_1),\\
&= 2\mathrm{AIC}_{rec_1} - \mathrm{AIC}_{rec_3} - H_1, \qquad (5.80)\\
p_{\mathrm{DIC}_{rec_2}} &= \overline{D_{rec}(\Theta)} - \hat{D}_{rec}(\Theta),\\
&= (\mathrm{AIC}_{rec_1} - H_1) - (\mathrm{AIC}_{rec_3} - H_1),\\
&= \mathrm{AIC}_{rec_1} - \mathrm{AIC}_{rec_3}, \qquad (5.81)
\end{aligned}$$

or based on BIC_{rec_1} and BIC_{rec_3}, as

$$\begin{aligned}
\mathrm{DIC}_{rec_2} &= 2\overline{D_{rec}(\Theta)} - \hat{D}_{rec}(\Theta),\\
&= 2(\mathrm{BIC}_{rec_1} - H_2) - (\mathrm{BIC}_{rec_3} - H_2),\\
&= 2\mathrm{BIC}_{rec_1} - \mathrm{BIC}_{rec_3} - H_2, \qquad (5.82)\\
p_{\mathrm{DIC}_{rec_2}} &= \overline{D_{rec}(\Theta)} - \hat{D}_{rec}(\Theta),\\
&= (\mathrm{BIC}_{rec_1} - H_2) - (\mathrm{BIC}_{rec_3} - H_2),\\
&= \mathrm{BIC}_{rec_1} - \mathrm{BIC}_{rec_3}, \qquad (5.83)
\end{aligned}$$

5.10.2 Computational relations between the conditional likelihood-based criteria: AICs$_{con}$, BICs$_{con}$ and DICs$_{con}$

Similarly, we can explore the computational relationship between the conditional log likelihood-based AIC, BIC and DIC. First, we rewrite the fit terms of all versions of the AICs$_{con}$ and BICs$_{con}$ as

$$\begin{aligned} \text{AIC}_{con_1} &\approx \overline{D_{con}(\mathbf{z},\theta)} + H_1 \longleftrightarrow \overline{D_{con}(\mathbf{z},\theta)} \approx \text{AIC}_{con_1} - H_1, \\ \text{BIC}_{con_1} &\approx \overline{D_{con}(\mathbf{z},\theta)} + H_2 \longleftrightarrow \overline{D_{con}(\mathbf{z},\theta)} \approx \text{BIC}_{con_1} - H_2, \end{aligned} \tag{5.84}$$

$$\begin{aligned} \text{AIC}_{con_2} &\approx D_{con}(\hat{\mathbf{z}},\hat{\theta}) + H_1 \longleftrightarrow D_{con}(\hat{\mathbf{z}},\hat{\theta}) \approx \text{AIC}_{con_2} - H_1, \\ \text{BIC}_{con_2} &\approx D_{con}(\hat{\mathbf{z}},\hat{\theta}) + H_2 \longleftrightarrow D_{con}(\hat{\mathbf{z}},\hat{\theta}) \approx \text{BIC}_{con_2} - H_2, \end{aligned} \tag{5.85}$$

$$\begin{aligned} \text{AIC}_{con_3} &\approx \hat{D}_{con}(\mathbf{z},\theta) + H_1 \longleftrightarrow \hat{D}_{con}(\mathbf{z},\theta) \approx \text{AIC}_{con_2} - H_1, \\ \text{BIC}_{con_3} &\approx \hat{D}_{con}(\mathbf{z},\theta) + H_2 \longleftrightarrow \hat{D}_{con}(\mathbf{z},\theta) \approx \text{BIC}_{con_2} - H_2, \end{aligned} \tag{5.86}$$

where H_1 and H_2 are the same as those defined with AICs$_{rec}$ and BICs$_{rec}$ in subsection (5.10.1). Now, using Equations (5.84 - 5.86), it can be noted that these fit model terms represent the components of DICs$_{con}$ and its corresponding effective number of parameters, $p_{\text{DIC}_{con}}$, defined in subsection (5.5.2). Hence, we can rewrite the DIC$_{con_1}$ and $p_{\text{DIC}_{con_1}}$ as, either

$$\begin{aligned} \text{DIC}_{con_1} &= 2\overline{D_{con}(\mathbf{z},\theta)} - D_{con}(\hat{\mathbf{z}},\hat{\theta}), \\ &= 2(\text{AIC}_{con_1} - H_1) - (\text{AIC}_{con_2} - H_1), \\ &= 2\text{AIC}_{con_1} - \text{AIC}_{con_2} - H_1, \end{aligned} \tag{5.87}$$

$$\begin{aligned} p_{\text{DIC}_{con_1}} &= \overline{D_{con}(\mathbf{z},\theta)} - D_{con}(\hat{\mathbf{z}},\hat{\theta}), \\ &= (\text{AIC}_{con_1} - H_1) - (\text{AIC}_{con_2} - H_1), \\ &= \text{AIC}_{con_1} - \text{AIC}_{con_2}, \end{aligned} \tag{5.88}$$

or,

$$\begin{aligned}
\mathrm{DIC}_{con_1} &= 2\overline{D_{con}(\mathbf{z},\theta)} - D_{con}(\hat{\mathbf{z}},\hat{\theta}),\\
&= 2(\mathrm{BIC}_{con_1} - H_2) - (\mathrm{BIC}_{con_2} - H_2),\\
&= 2\mathrm{BIC}_{con_1} - \mathrm{BIC}_{con_2} - H_2, \qquad (5.89)\\
p_{\mathrm{DIC}_{con_1}} &= \overline{D_{con}(\mathbf{z},\theta)} - D_{con}(\hat{\mathbf{z}},\hat{\theta}),\\
&= (\mathrm{BIC}_{con_1} - H_2) - (\mathrm{BIC}_{con_2} - H_2),\\
&= \mathrm{BIC}_{con_1} - \mathrm{BIC}_{con_2}, \qquad (5.90)
\end{aligned}$$

respectively. The DIC_{con_2} defined in subsection (5.5.2), can be rewritten based on the AIC_{con_1} and AIC_{con_3}, or the BIC_{con_1} and BIC_{con_3}. That is,

$$\begin{aligned}
\mathrm{DIC}_{con_2} &= 2\overline{D_{con}(\mathbf{z},\theta)} - \hat{D}_{con}(\mathbf{z},\theta),\\
&= 2(\mathrm{AIC}_{con_1} - H_1) - (\mathrm{AIC}_{con_3} - H_1),\\
&= 2\mathrm{AIC}_{con_1} - \mathrm{AIC}_{con_3} - H_1, \qquad (5.91)\\
p_{\mathrm{DIC}_{con_2}} &= \overline{D_{con}(\mathbf{z},\theta)} - \hat{D}_{con}(\mathbf{z},\theta),\\
&= (\mathrm{AIC}_{con_1} - H_1) - (\mathrm{AIC}_{con_3} - H_1),\\
&= \mathrm{AIC}_{con_1} - \mathrm{AIC}_{con_3}, \qquad (5.92)
\end{aligned}$$

or,

$$\begin{aligned}
\mathrm{DIC}_{con_2} &= 2\overline{D_{con}(\mathbf{z},\theta)} - \hat{D}_{con}(\mathbf{z},\theta),\\
&= 2(\mathrm{BIC}_{con_1} - H_2) - (\mathrm{BIC}_{con_3} - H_2),\\
&= 2\mathrm{BIC}_{con_1} - \mathrm{BIC}_{con_3} - H_2, \qquad (5.93)\\
p_{\mathrm{DIC}_{con_2}} &= \overline{D_{con}(\mathbf{z},\theta)} - \hat{D}_{con}(\mathbf{z},\theta),\\
&= (\mathrm{BIC}_{con_1} - H_2) - (\mathrm{BIC}_{con_3} - H_2),\\
&= \mathrm{BIC}_{con_1} - \mathrm{BIC}_{con_3}. \qquad (5.94)
\end{aligned}$$

Chapter 6

Evaluation of model selection criteria

6.1 Introduction

In this chapter, we implement simulation studies to assess the performance of the model selection criteria introduced in Chapter 5.

These criteria will be assessed on the basis of many synthetic data sets simulated from data generating mechanisms with different complexity, K. The aim is to check the ability of these criteria in selecting the correct model, given different data generating mechanisms. Furthermore, the same criteria will be evaluated on an application of real data involving the waiting time of Faithful Old geyser data.

6.2 Simulation study

In this section, we design a simulation study aimed at comparing the performance of the following model selection criteria introduced in Chapter 5:

- Recursive deviance-based criteria: AIC_{rec_1}, BIC_{rec_1}, AIC_{rec_2}, BIC_{rec_2}, AIC_{rec_3}, BIC_{rec_3}, DIC_{rec_1}, DIC_{rec_2};

- Conditional deviance-based criteria: AIC_{con_1}, BIC_{con_1}, AIC_{con_2}, BIC_{con_2}, AIC_{con_3}, BIC_{con_3}, DIC_{con_1}, DIC_{con_2};

- Predictive ability-based criterion: WAIC.

6.2.1 Generating simulated data

We evaluate the model selection criteria under four groups of normally distributed data with the assumption of equality of variance, with 500 observations each, simulated from four data-generating mechanisms, with different complexities as follows:

2-states model

$$\pi = \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix}, \; \mathbf{A} = \begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}, \; \mu = \begin{bmatrix} 10 \\ 20 \end{bmatrix},$$

with equal variances $\sigma_1^2 = \sigma_2^2 = 1$.

3-states model

$$\pi = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.3 \end{bmatrix}, \; \mathbf{A} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \; \mu = \begin{bmatrix} 2 \\ 12 \\ 19 \end{bmatrix},$$

with equal variances $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 1$.

5-states model

$$\pi = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.2 \\ 0.1 \\ 0.1 \end{bmatrix}, \; \mathbf{A} = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.5 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.2 & 0.5 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.2 & 0.5 \end{bmatrix}, \; \mu = \begin{bmatrix} 4 \\ 6 \\ 12 \\ 16 \\ 20 \end{bmatrix},$$

with equal variances $\sigma_1^2 = ... = \sigma_5^2 = 1$.

7-states model

$$\pi = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}, \; \mathbf{A} = \begin{bmatrix} 0.4 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.4 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.2 & 0.1 & 0.1 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.4 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.4 \end{bmatrix}, \; \mu = \begin{bmatrix} 4 \\ 8 \\ 12 \\ 16 \\ 20 \\ 22 \\ 24 \end{bmatrix},$$

with equal variances $\sigma_1^2 = ... = \sigma_7^2 = 1$.

From each model, we simulate 200 data sets. Figures (6.1–6.4) show 200 data sets simulated from each considered generating data model, of 500 observations each, plotted on the same picture. In what follows, for each case, we refer to the model from which the data are generated as K_0. The aim of this simulation study is to evaluate the model selection ability of each criterion, given increased model complexity.

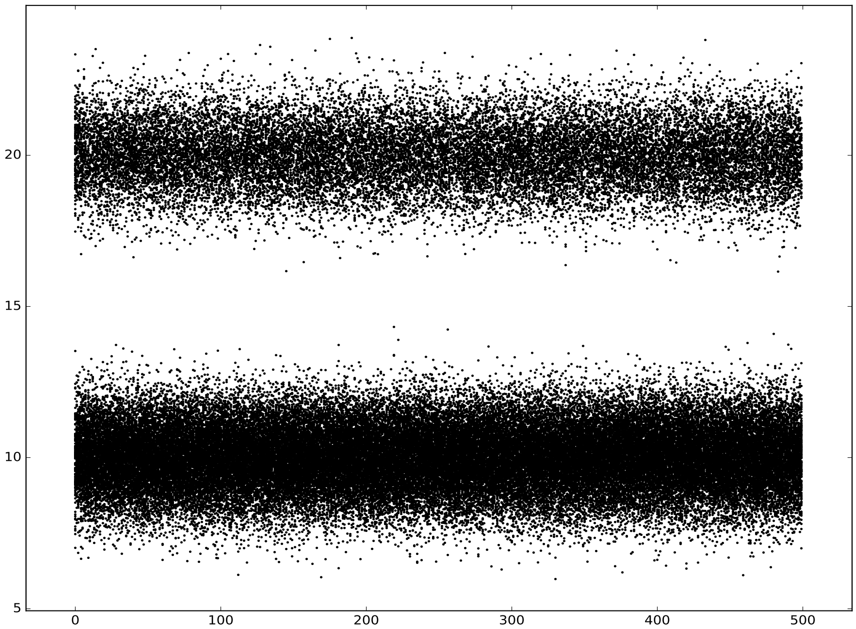


Figure 6.1: Simulated data from 200 Normal HMMs of length 500 with $K_0 = 2$.

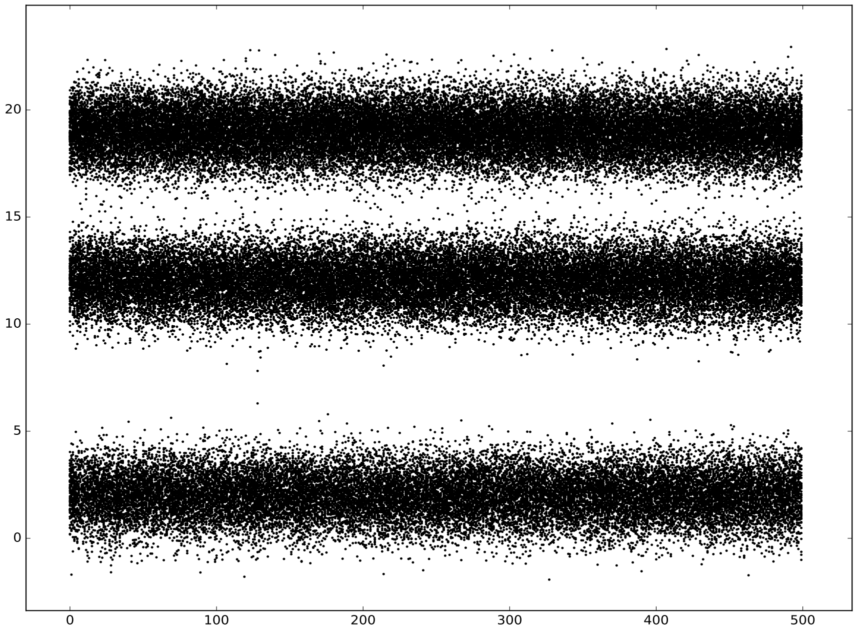


Figure 6.2: Simulated data from 200 Normal HMMs of length 500 with $K_0 = 3$.

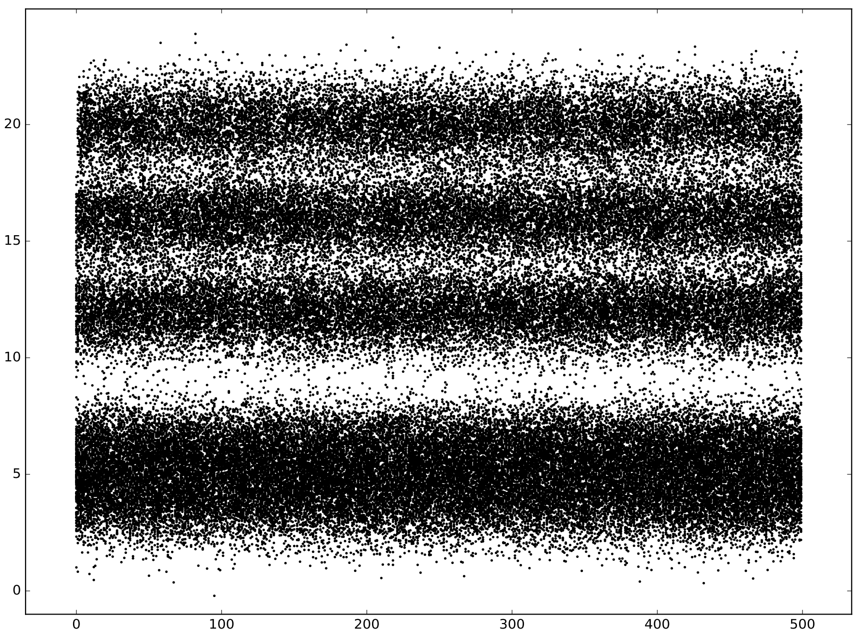


Figure 6.3: Simulated data from 200 Normal HMMs of length 500 with $K_0 = 5$.

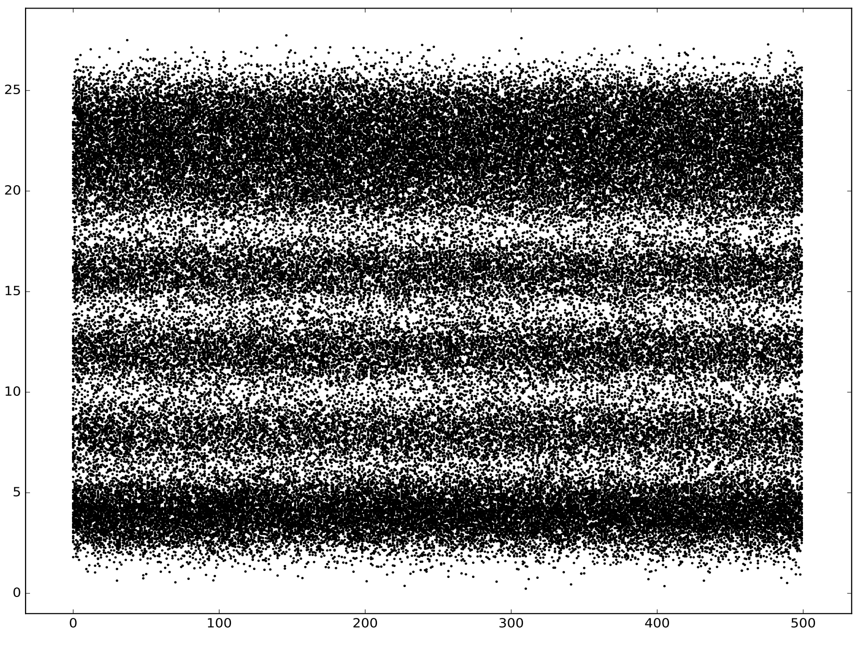


Figure 6.4: Simulated data from 200 Normal HMMs of length 500 with $K_0 = 7$.

6.2.2 Fitting competing models

We fit several competing models for each data set being simulated from each of the generating data models defined earlier. We assume a different number of competing models fitted for each data set. We fit competing models with the number of hidden states K varying between K=2 and K=5 for the first two datasets (i.e. for the generating data models with $K_0 = 2$ and $K_0 = 3$ states, respectively), whereas we fit competing models with the number of hidden states K allowed to vary between K=3 and K=7 for the generating model with $K_0 = 5$ states. For the data sets simulated from the model with $K_0 = 7$ states, we allow K to vary between K=5 and K=9. For each competing model being fitted to each data set, we run the Gibbs sampler for 20000 iterations and discarded the first 5000 iterations as burn-in period. We assume non-informative priors on all parameters of all competing models. For the state-specific parameters, μ and σ^2, we assume the Normal and Gamma distribution priors respectively, so that:

$$\mu_j \overset{ind.}{\sim} N(0, 1000),$$
$$\sigma_j^{-2} \overset{ind.}{\sim} \Gamma(0.001, 0.001), \; \forall\, j,$$

where j is an index that refers to each component in the model fitted to each data set. For the state initial distribution, π, and each row, $\{a_{j.}\}$, of the transition matrix, $\mathbf{A}$, we assume independently a Dirichlet prior with a hyper parameter $\delta_k = 1, \; \forall j, k$, for each fitted model K, so that:

$$\{a_{j.}\} \text{ and } \pi \overset{ind.}{\sim} Dir(1, 1, ..., 1_k).$$

The k is also the index of each model's size being fitted for each simulated data set. In what follows, we show the results of model selection of each considered criterion under various scenarios. Since all considered criteria here are based on the output of an MCMC sampling, they are most likely to be subjected to simulation variability. Therefore, we measure the variability of these criteria by computing the numerical standard error of each criterion, to report the criteria accuracy.

6.2.3 Simulation results

Among the 200 simulated datasets, Tables (6.1-6.4) report the percentage of the number of times each model is selected by one of the considered criteria, given four data-generating schemes, respectively. These tables include also the numerical standard errors computed over 200 replications.

In the case of the generating data model with K_0 =2 states, it can be seen that all versions of the criteria AIC, BIC and DIC, based on recursive deviances, except the DIC$_{rec_1}$, perform well in selecting the correct model as shown in Table (6.1). The DIC$_{rec_1}$ has a satisfactory selection percentage for the correct model (43.00% at $K = 2$ compared with other values of K), but it has a marked tendency to overestimate the number of states K. From the same table, it can be noted that the criteria AIC, BIC and DIC based on conditional deviances, as well as the WAIC, have a different behavior. Among these criteria, the versions: BIC$_{con_1}$, BIC$_{con_2}$ and DIC$_{con_2}$ seems to have the most satisfactory performance. However, they have a tendency to over-penalize the complexity of the model with a slight percentages. In contrast, the versions AIC$_{con_1}$, AIC$_{con_2}$, AIC$_{con_3}$, BIC$_{con_3}$ DIC$_{con_1}$ and also WAIC do not select the correct model and all have a high tendency to choose a more complicated model ($K = 5$).

With respect to the results of the generating model with $K_0 = 3$ states shown in Table (6.2), it can be seen that all versions of criteria AIC, BIC and DIC based on recursive deviances still select the right model, indicating their reliability. Except for the DIC$_{rec_1}$, they all have almost the same percentage of selecting the correct model. It is worth noting that the DIC$_{rec_1}$ has the same behaviour as that observed in the previous case concerning the generating model with $K_0 = 2$ states (Table (6.1)), where it has the smallest selection percentage (81.50%) compared with other criteria. In contrast, we can see from the same table that some criteria based on conditional deviances again have satisfactory behavior. For example, the BIC$_{con_1}$, BIC$_{con_2}$ and DIC$_{con_2}$, and this time the AIC$_{con_2}$, all select the correct generating model $(K_0 = 3)$. Note that the AIC$_{con_2}$, despite its satisfactory performance here, it behaves differently compared to the case of the generating model with $K_0 = 2$ states (Table (6.1)), indicating its invalidity, whereas the BIC$_{con_1}$, BIC$_{con_2}$ and DIC$_{con_2}$ have similar performance with respect to selecting the correct model, suggesting their validity. From the same table, it can be seen that the remaining versions: AIC$_{con_1}$, AIC$_{con_3}$, BIC$_{con_3}$, DIC$_{con_1}$ and WAIC have the same poor behaviour as the one observed in the case of the generating model with $K_0 = 2$ states (Table (6.1)), where they have a high tendency to select the most complicated model $(K = 5)$.

Regarding the generating model with $K_0 = 5$ states, there are different choices of the criteria as reported in Table (6.3). It can be seen that all versions of recursive deviance-based criteria: AIC, BIC and the second version of DIC, DIC$_{rec_2}$), behave differently and tend to pick an under-fit model with $K = 4$ states as a representation of the data, which seems a sensible choice as suggested by Figure (6.3). Specifically, the observed process produced by this generating HMM offers some overlapping for some modes of the model due to the existence of state-dependent means very closed each other, e.g. $\mu_1 = 2$ and $\mu_2 = 4$. It appears that the DIC$_{rec_1}$ has the most satisfactory performance with respect to selecting the correct model, but it has a tendency to overestimate the number of states K in about 41% of cases. In contrast, it can be noted that among the criteria based on conditional deviances that only the version AIC$_{con_2}$ selects the correct model and has a high tendency to over-penalise the model in about 43% of cases. The versions BIC$_{con_1}$ and BIC$_{con_1}$ behave in the same way as the majority of the recursive deviance-based criteria above, which favour the model with $K = 4$, a reasonable solution as indicated by Figure (6.3). Both versions have a somewhat similar tendency to underestimate the number of states K in almost 33% of cases. The AIC$_{con_1}$ and WAIC both have the same performance as in the two previous cases (i.e. $K_0 = 2$ and $K_0 = 3$), where they select the most complex model, $K = 7$. Regarding the versions: AIC$_{con_3}$, BIC$_{con_3}$ and DIC$_{con_1}$, they have the same behaviour as the AIC$_{con_1}$ and WAIC, where they also select an overfitting model, but with lower complexity ($K = 6$).

Finally, it appears from the results concerning the generating model with $K_0 = 7$, in Table (6.4), that all versions of the AIC and BIC have very high orientation to under-penalize the model complexity, a behaviour similar to the previous case (the case of $K_0 = 5$), where they select the smallest model, $K = 5$, for the data. This latter selection, made by the above criteria, seems also a reasonable solution according to the observed pattern of the data in Figure (6.4). It seems that they take into account the overlapping in the data, where it is difficult, for example, to diagnose the existence of 7 states. More clearly, the state-specific means: $\mu_5 = 20$, $\mu_6 = 22$ and $\mu_7 = 24$ are very close to each other and can form a single mode. This highlights that these criteria appear to be sensitive to the real representation of observed process produced by the HMM. The DIC$_{rec_1}$ has similar performance as that observed in the cases $K_0 = 2$, $K_0 = 3$ and $K_0 = 5$, and it tends to choose the real number of generating data model ($K = K_0 = 7$). On the other hand, the version DIC$_{rec_2}$ behaves in the same way as the recursive deviance-based versions of the AIC and BIC, where it tends to select the model with fewest parameters ($K = 5$). From the same table, it can be noted that the BIC$_{con_1}$ and BIC$_{con_2}$ also select the smaller model, $K = 5$, to the data. It can see that the AIC$_{con_1}$, AIC$_{con_2}$, BIC$_{con_3}$ and WAIC behave differently from their poor performance in the previous three cases, as this time they select the correct model. This can indicate that such criteria may be appropriate in evaluating large models. The DIC$_{con_2}$ has a tendency to underestimate the real number of hidden states. Note that the versions AIC$_{con_3}$ and DIC$_{con_1}$ still have a poor performance, where they select more complicated model for the data with $K = 8$ and $K = 9$, respectively.

It should be noted that all criteria based on the recursive likelihood: the AIC, BIC and DIC provide generally lower standard errors, as shown in brackets in all tables, compared with those based on the conditional likelihood and also the WAIC, indicating better accuracy. This high variability in the latter criteria may be because of the MCMC sampling includes high dimensional vectors of hidden state along with the model parameters, which would affect the variability of MC approximations of those criteria.

		Fitted models			
K_0	Criterion	**2**	3	4	5
		Percentage (%) of times selected			
2	AIC_{rec_1}	**99.5**%	0.50%	0.00	0.00
		(3.351)	(3.332)	(3.344)	(3.315)
	BIC_{rec_1}	**99.5**	0.50	0.00%	0.00
		(3.351)	(3.357)	(3.344)	(3.335)
	AIC_{rec_2}	**99.5**%	0.50%	0.00%	0.00%
		(3.350)	(3.339)	(3.312)	(3.329)
	BIC_{rec_2}	**99.5**%	0.50%	0.00%	0.00%
		(3.350)	(3.333)	(3.312)	(3.352)
	AIC_{rec_3}	**94.50**%	5.50%	0.00%	0.00%
		(3.351)	(3.385)	(3.373)	(3.347)
	BIC_{rec_3}	**100.00**%	0.00%	0.00%	0.00%
		(3.351)	(3.385)	(3.373)	(3.347)
	DIC_{rec_1}	**43.00**%	28.50%	15.00%	13.50%
		(3.352)	(3.337)	(3.425)	(3.354)
	DIC_{rec_2}	**94.00**%	6.00%	0.00%	0.00%
		(3.351)	(3.424)	(3.325)	(3.302)
2	AIC_{con_1}	0.50%	2.50%	18.00%	**79.00**%
		(3.213)	(8.898)	(8.607)	(14.217)
	BIC_{con_1}	**55.00**%	15.50%	14.00%	15.50%
		(3.213)	(8.923)	(8.607)	(14.305)
	AIC_{con_2}	13.00%	10.50%	21.50%	**55.00**%
		(4.212)	(6.703)	(8.756)	(13.732)
	BIC_{con_2}	**87.50**%	6.50%	3.50%	2.50%
		(4.212)	(6.714)	(8.359)	(13.735)
	AIC_{con_3}	0.50%	3.50%	21.50%	**74.50**%
		(3.212)	(5.802)	(9.198)	(8.088)
	BIC_{con_3}	0.50%	9.50%	32.00%	**58.00**%
		(3.212)	(5.802)	(9.198)	(8.002)
	DIC_{con_1}	0.50%	0.00%	7.50%	**92.00**%
		(3.213)	(5.185)	(8.513)	(14.918)
	DIC_{con_2}	**95.00**%	1.00%	1.50%	2.50%
		(3.213)	(5.450)	(9.885)	(15.168)
2	WAIC	0.50%	0.50%	6.50%	**92.50**%
		(3.212)	(7.377)	(6.560)	(9.808)

Table 6.1: Percentage of the number of times in which the models with 2–5 states are chosen by each criterion over 200 independent simulation data sets, each of which of length 500 observations, generated from a HMM with $K_0 = 2$ states. The numbers in brackets indicate numerical standard errors.

		Fitted models			
K_0	Criterion	2	**3**	4	5
		Percentage (%) of times selected			
3	AIC_{rec_1}	0.00%	**94.00**%	6.00%	0.00%
		(3.319)	(3.291)	(6.497)	(9.782)
	BIC_{rec_1}	0.00%	**94.00**%	6.00%	0.00%
		(3.319)	(3.291)	(6.497)	(9.782)
	AIC_{rec_2}	0.00	**94.00**	6.00	0.00%
		(3.322)	(3.251)	(6.497)	(9.848)
	BIC_{rec_2}	0.00%	**94.00**%	6.00%	0.00%
		(3.322)	(3.251)	(6.497)	(9.848)
	AIC_{rec_3}	0.00%	**94.00**%	6.00%	0.00%
		(3.342)	(3.352)	(6.497)	(9.779)
	BIC_{rec_3}	0.00%	**94.00**%	6.00%	0.00%
		(3.342)	(3.352)	(6.497)	(9.779)
	DIC_{rec_1}	0.00%	**81.50**%	17.00%	1.50%
		(3.361)	(3.376)	(6.498)	(9.715)
	DIC_{rec_2}	0.00%	**93.50**%	6.50%	0.00%
		(3.301)	(3.242)	(6.498)	(9.784)
3	AIC_{con_1}	0.00%	5.50%	28.50%	**66.00**%
		(3.674)	(4.799)	(6.211)	(13.623)
	BIC_{con_1}	0.00%	**83.00**%	16.00%	1.00%
		(3.674)	(4.799)	(6.211)	(13.623)
	AIC_{con_2}	0.00%	**48.00**%	33.00%	19.00%
		(4.058)	(3.618)	(6.219)	(13.908)
	BIC_{con_2}	0.00%	**90.50**%	9.50%	0.00%
		(4.058)	(3.618)	(6.219)	(13.908)
	AIC_{con_3}	0.00%	0.00%	9.50%	**90.50**%
		(3.320)	(3.586)	(5.446)	(11.821)
	BIC_{con_3}	0.00%	0.00%	41.00%	**59.00**%
		(3.320)	(3.586)	(5.446)	(11.821)
	DIC_{con_1}	0.00%	0.00%	4.50%	**95.50**%
		(4.208)	(3.913)	(6.202)	(13.338)
	DIC_{con_2}	0.00%	**86.00**%	9.00%	6.00%
		(3.108)	(3.497)	(7.153)	(15.453)
3	WAIC	0.00%	1.00%	10.00%	**89.00**%
		(4.491)	(4.561)	(8.101)	(14.382)

Table 6.2: Percentage of the number of times in which the models with 2–5 states are chosen by each criterion over 200 independent simulation data sets, each of which of length 500 observations, generated from a HMM with $K_0 = 3$ states. The numbers in brackets indicate numerical standard errors.

		Fitted models				
K_0	Criterion	3	4	**5**	6	7
		Percentage (%) of times selected				
5	AIC_{rec_1}	0.00%	**82.00**%	17.5%	0.50%	0.00%
		(2.981)	(3.319)	(2.998)	(2.915)	(2.759)
	BIC_{rec_1}	21.50%	**78.00**%	0.50%	0.00%	0.00%
		(2.981)	(3.319)	(2.998)	(2.915)	(2.759)
	AIC_{rec_2}	0.00%	**80.00**%	19.00%	1.00%	0.00%
		(2.988)	(3.554)	(3.303)	(3.653)	(3.622)
	BIC_{rec_2}	18.00%	**81.50**%	0.50%	0.00%	0.00%
		(2.988)	(3.554)	(3.303)	(3.653)	(3.622)
	AIC_{rec_3}	0.00%	**76.50**%	22.50%	1.00%	0.00%
		(2.982)	(3.302)	(3.003)	(2.963)	(2.840)
	BIC_{rec_3}	15.50%	**82.00**%	2.50%	0.00%	0.00%
		(2.982)	(3.302)	(3.003)	(2.963)	(2.840)
	DIC_{rec_1}	0.00%	19.00%	**39.50**%	21.50%	21.00%
		(2.981)	(3.293)	(3.112)	(3.097)	(3.561)
	DIC_{rec_2}	0.00%	**64.50**%	32.00%	3.50%	0.00%
		(2.982)	(3.347)	(3.003)	(2.888)	(2.703)
5	AIC_{con_1}	0.00%	3.50%	22.50%	25.50%	**48.50**%
		(4.970)	(5.648)	(5.422)	(5.385)	(4.907)
	BIC_{con_1}	0.00%	**39.00**%	34.50%	20.00%	6.50%
		(4.970)	(5.648)	(5.422)	(5.385)	(4.907)
	AIC_{con_2}	0.00%	24.50%	**33.00**%	22.50%	20.00%
		(5.488)	(8.046)	(9.488)	(13.966)	(15.557)
	BIC_{con_2}	0.00%	**55.00**%	33.00%	9.50%	2.50%
		(5.488)	(8.046)	(9.488)	(13.966)	(15.557)
	AIC_{con_3}	0.00%	0.00%	1.50%	18.00%	**80.50**%
		(3.805)	(3.773)	(4.720)	(5.461)	(4.938)
	BIC_{con_3}	0.00%	0.00%	25.00%	35.50%	**39.50**%
		(3.805)	(3.773)	(4.720)	(5.461)	(4.938)
	DIC_{con_1}	0.00%	0.00%	3.00%	15.50%	**81.50**%
		(5.062)	(6.435)	(7.558)	(12.113)	(15.113)
	DIC_{con_2}	0.00%	23.00%	28.00%	20.00%	**29.00**%
		(6.246)	(7.733)	(6.539)	(6.112)	(5.984)
5	WAIC	0.00%	6.00%	18.50	20.00%	**55.50**%
		(6.317)	(8.190)	(5.242)	(4.855)	(4.696)

Table 6.3: Percentage of the number of times in which the models with 3–7 states are chosen by each criterion over 200 independent simulation data sets, each of which of length 500 observations, generated from a HMM with $K_0 = 5$ states. The numbers in brackets indicate numerical standard errors.

K_0	Criterion	Fitted models				
		5	6	**7**	8	9
		Percentage (%) of times selected				
7	AIC_{rec_1}	**96.00**%	4.00%	0.00%	0.00%	0.00%
		(3.246)	(3.073)	(3.095)	(2.995)	(2.966)
	BIC_{rec_1}	**100.00**%	0.00%	0.00%	0.00%	0.00%
		(3.246)	(3.073)	(3.095)	(2.995)	(2.966)
	AIC_{rec_2}	**89.00**%	11.00%	0.00%	0.00%	0.00%
		(3.328)	(4.019)	(3.909)	(4.010)	(3.669)
	BIC_{rec_2}	**100.00**%	0.00%	0.00%	0.00%	0.00%
		(3.328)	(4.019)	(3.909)	(4.010)	(3.669)
	AIC_{rec_3}	**92.00**%	8.00%	0.00%	0.00%	0.00%
		(3.224)	(3.074)	(3.117)	(3.070)	(3.006)
	BIC_{rec_3}	**100.00**%	0.00%	0.00%	0.00%	0.00%
		(3.224)	(3.074)	(3.117)	(3.070)	(3.006)
	DIC_{rec_1}	8.50%	22.00%	**36.00**%	20.00%	13.50%
		(3.199)	(3.489)	(3.118)	(3.309)	(3.275)
	DIC_{rec_2}	**42.50**%	40.00%	16.00%	1.50%	0.00%
		(3.290)	(3.101)	(3.116)	(2.981)	(2.979)
7	AIC_{con_1}	4.50%	18.50%	**51.00**%	19.00%	7.00%
		(7.717)	(6.980)	(7.542)	(7.176)	(7.283)
	BIC_{con_1}	**47.00**%	38.50%	14.00%	0.50%	0.00%
		(7.717)	(6.980)	(7.542)	(7.176)	(7.283)
	AIC_{con_2}	16.50%	32.00%	**44.50**%	6.00%	1.00%
		(10.152)	(16.958)	(14.638)	(15.215)	(13.708)
	BIC_{con_2}	**52.50**%	37.50%	10.00%	0.00%	0.00%
		(10.152)	(16.958)	(14.638)	(15.215)	(13.708)
	AIC_{con_3}	0.00%	0.50%	15.00%	41.50%	**42.50**%
		(4.071)	(3.943)	(5.745)	(6.615)	(7.639)
	BIC_{con_3}	2.50%	25.00%	**53.50**%	17.00%	2.00%
		(4.071)	(3.943)	(5.745)	(6.615)	(7.639)
	DIC_{con_1}	0.00%	2.00%	2.50%	18.00%	**77.50**%
		(6.576)	(13.838)	(10.948)	(14.370)	(11.780)
	DIC_{con_2}	20.00%	**30.50**%	29.00%	12.00%	8.50%
		(12.154)	(10.927)	(10.238)	(9.690)	(8.735)
7	WAIC	7.00%	16.50%	**40.50**%	20.50%	15.50%
		(10.626)	(8.134)	(8.402)	(8.312)	(8.379)

Table 6.4: Percentage of the number of times in which the models with 5–9 states are chosen by each criterion over 200 independent simulation data sets, each of which of length 500 observations, generated from a HMM with $K_0 = 7$ states. The numbers in brackets indicate numerical standard errors.

6.3 Application to real data

In this section we assess the performance of our criteria using a real application involving the waiting times of the Old Faithful geyser data. These data were used in Chapter 4 to examine the MCMC sampler employed in model estimation. The waiting times of the Old Faithful geyser data consist of 299 observations. These data have been used by many authors to address the model selection issue in HMMs. For example, based on the frequentist framework using the EM approach, Zucchini and MacDonald (2009) used the AIC and BIC to select the best Normal HMM of those data. They concluded that the best model to adequately fit these data is the model with $K = 3$ according to the BIC, whereas the model with $K = 4$ is selected based on the AIC. Robert and Titterington (1998) graphically proposed, without using a named criterion, that $K = 3$ states are adequate for fitting those data. A study introduced by McGillivray and Khalili (2014), based on the frequentist framework, concluded that the Normal HMMs with $K = 2$ and $K = 3$ states are the best according to the BIC and AIC respectively. We display the histogram of these 299 observations in Figure (6.5) which also includes plots of the probability densities of several competing models fitted to these observations. We put an upper bound, $K_{max} = 7$ on the number of competing models, K, fitted to those data, where $K = 2,3,\dots,K_{max}$. We use the same procedure followed in the simulation study in section (6.2.2) with respect to the prior specification. For each test model, we run the Gibbs sampler for 20000 iterations and discard the first 5000 iterations as a burn-in period. To avoid the label switching issue, we impose identifiability constraints on the mean parameter, μ, of each test model so that: $\mu_1 < \mu_2 < \dots < \mu_k$.

6.3.1 Results

Tables (6.5- 6.7) show results for all proposed criteria: the AIC, BIC, DIC and the WAIC, respectively. We display the graphical estimation results of the models fitted in Figure (6.5).

Table (6.5) shows that all modified versions for the AIC and BIC, based on the recursive deviances, choose the model with $K = 3$ as the best model for the waiting times between two successive eruptions. Similarly, the second version of DIC, DIC$_{rec_2}$ behaves in the same pattern as the AICs$_{rec}$ and BICs$_{rec}$, where it selects the model with $K = 3$ states. Note also that the effective number of parameters of DIC$_{rec_2}$ gives values that increase as expected until $K = 4$. At the next states, $K = 5,6$ and 7, the effective number of parameters increases slightly, indicating that the additional states do not considerably contribute to the model's deviance. This can be also noted from the models fitting observed in Figure (6.5) where there is no marked improvement after $K = 4$. In contrast, the first version, DIC_{rec_1}, has a different behaviour and chooses a more complicated model for these data, $K = 7$. Note that the effective number of parameters of the DIC_{rec_1} has an unsatisfactory behavior after $K = 5$, as it decreases at $K = 6$ and then increases at $K = 7$.

On the other hand, Table (6.6) shows different choices are made by the conditional deviance-based criteria. For instance, the BIC_{con_1} selects the more parsimonious model which is $K = 4$. Similarly, both versions of DIC based on conditional deviance: DIC_{con_1} and DIC_{con_2} choose the same model, $K = 4$. It can be seen that the DIC_{con_1} provides non-increasing and highly fluctuating p_{DICs}. On the other hand, the DIC_{con_2} gives increased p_{DICs} with too large values. This may be the result of including the hidden cases as additional parameters in the model. Note that, despite the large values of $p_{\text{DIC}_{con_2}}$, the DIC_{con_2} has a similar behavior to the DIC_{rec_2}, where the former has a slight increase in $p_{\text{DIC}_{con_2}}$ after $K = 5$. This pattern may be attributed to both versions defined based on plugged-in functional estimators. For other versions of the conditional deviance-based AIC and BIC, it can be seen that both the AIC_{con_2} and the AIC_{con_3} select the more complicated model $K = 6$, whereas both versions of BIC: BIC_{con_2} and BIC_{con_3} tend to select a less complicated model, $K = 5$.

Table (6.7) shows that the WAIC provides effective dimensions that increase with increasing K. However, it tends to select the most complicated model for these data, $K = 7$.

It is interesting to note that the model choice with $K = 4$, made by BIC_{con_1} and also the DIC_{con_1} and DIC_{con_2} can be a plausible selection for these data, as a marked improvement in model fitting with $K = 3$ through the model with $K = 4$ is observed, whereas is only a slight improvement after the model with $K = 4$ as shown in Figure (6.5).

K	Deviance			AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	DIC$_{rec_1}$	$p_{\text{DIC}_{rec_1}}$	DIC$_{rec_2}$	$p_{\text{DIC}_{rec_2}}$
	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$										
2	2194.582	2189.569	2186.798	2208.582	2234.485	2203.569	2229.472	2200.798	2226.702	2199.596	5.013	2202.366	7.784
3	2122.987	2114.158	2108.885	**2150.987**	**2202.794**	**2142.158**	**2193.964**	**2136.885**	**2188.691**	2131.817	8.829	**2137.091**	14.103
4	2121.108	2109.455	2102.309	2167.108	2252.218	2155.455	2240.565	2148.309	2233.419	2132.761	11.653	2139.907	18.799
5	2120.616	2108.057	2099.334	2188.616	2314.431	2176.057	2301.872	2167.334	2293.149	2133.175	12.559	2141.898	21.282
6	2118.113	2107.141	2096.115	2212.133	2386.013	2201.141	2375.041	2190.195	2364.015	2129.085	10.972	2140.071	21.938
7	2117.039	2105.663	2094.184	2241.039	2470.439	2229.663	2459.063	2218.184	2447.584	**2128.415**	11.376	2139.894	22.855

Table 6.5: Results of the proposed criteria, based on recursive deviances, for the waiting time of Old Faithful geyser data.

K	Deviance			AIC$_{con_1}$	BIC$_{con_1}$	AIC$_{con_2}$	BIC$_{con_2}$	AIC$_{con_3}$	BIC$_{con_3}$	DIC$_{con_1}$	$p_{\text{DIC}_{con_1}}$	DIC$_{con_2}$	$p_{\text{DIC}_{con_2}}$
	$\overline{D_{con}(\mathbf{z},\theta)}$	$D_{con}(\hat{\mathbf{z}},\hat{\theta})$	$\hat{D}_{con}(\mathbf{z},\theta)$										
2	2051.746	2044.062	2011.382	2065.746	2091.649	2058.062	2083.965	2025.382	2051.285	2059.431	7.684	2090.618	39.618
3	1825.760	1822.704	1781.642	1853.760	1905.566	1850.704	1902.510	1809.642	1861.448	1828.816	3.056	1869.877	44.117
4	1752.659	1742.693	1661.090	1798.659	**1883.769**	1788.693	1873.804	1707.090	1792.201	**1762.624**	9.965	**1844.228**	91.568
5	1731.827	1669.454	1582.849	1799.827	1925.643	1737.454	**1863.269**	1650.849	**1776.665**	1794.201	62.373	1880.805	148.977
6	1688.982	1612.221	1535.904	**1782.982**	1956.903	**1706.221**	1880.142	**1625.904**	1799.825	1842.504	76.761	1955.138	153.078
7	1662.665	1594.349	1506.884	1784.665	2016.092	1716.349	1947.776	1628.884	1860.311	1799.297	68.316	1974.227	155.781

Table 6.6: Results of the proposed criteria, based on conditional deviances, for the waiting time of Old Faithful geyser data.

K	ilppd	WAIC$_{var}$	$p_{\text{WAIC}_{var}}$
2	-1012.608	2101.489	38.135
3	-897.214	1893.996	49.784
4	-845.430	1858.040	83.590
5	-821.5745	1861.909	109.380
6	-790.1925	1837.593	128.604
7	-757.084	**1790.206**	138.019

Table 6.7: Results of the WAIC and the effective number of parameters, based on the integrated log-pointwise predictive density, applied for the waiting time of Old Faithful geyser data.

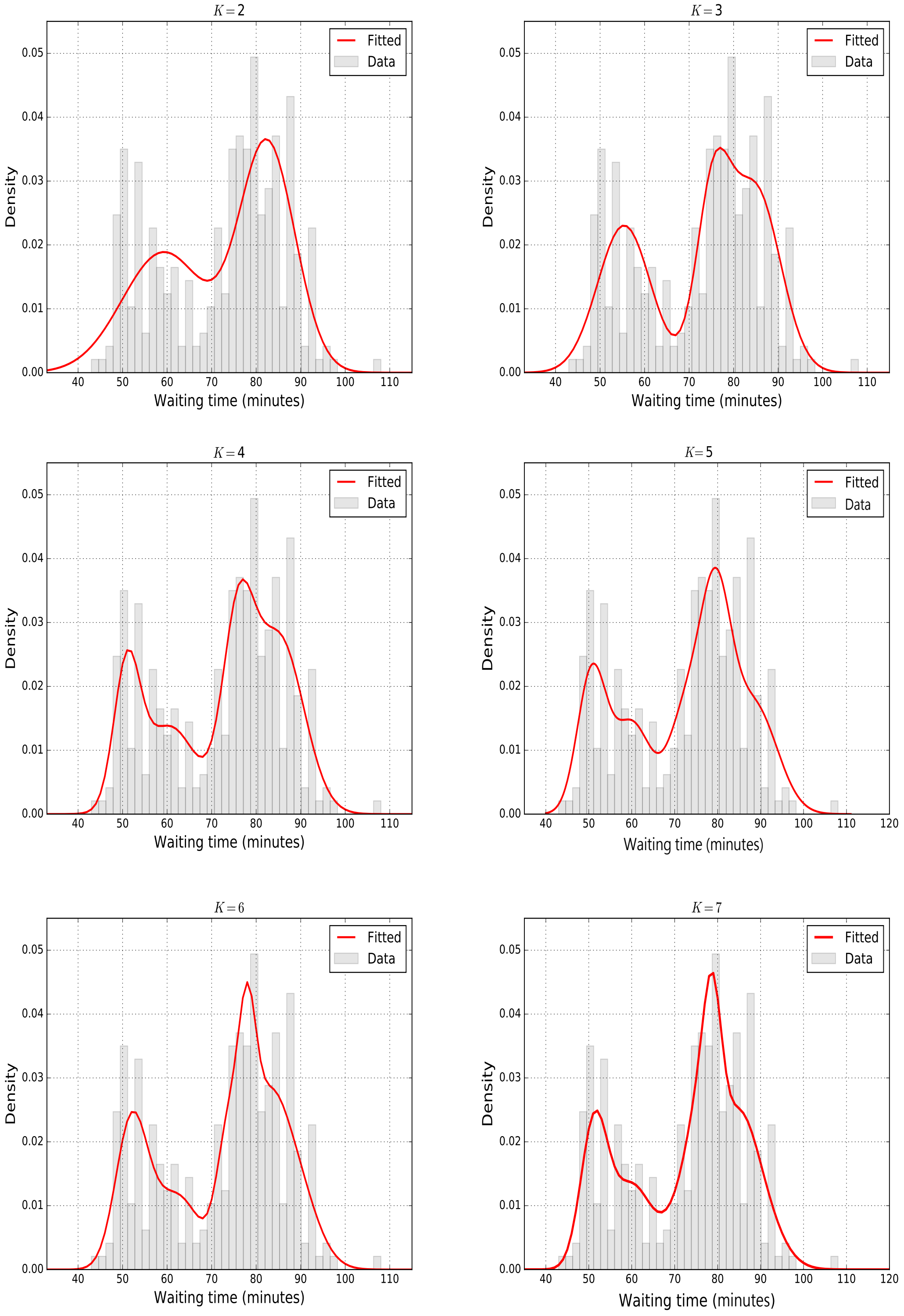


Figure 6.5: Histograms of the densities for several Normal HMMs, with different states, fitted to the waiting time of Old Faithful geyser data.

6.4 Summary and discussion

In this chapter, we have investigated the model selection issue in the HMMs context using several criteria proposed in Chapter 5 of this thesis. We have assessed those criteria using simulated data sets generated from models with different complexities. In addition, we have evaluated those criteria based on an application to real data. The conclusions of this study can be summarised the following.

1. In terms of the recursive deviance-based criteria, it was noted that all versions of the AIC and BIC, and also the second version of the DIC, DIC_{rec_2} that is based on a functional estimator (minimum recursive deviance evaluated at the posterior draws), seem to have a satisfactory performance towards selecting the correct model. In the simulation study designed on the basis of assuming different generating models with vary complexities K_0, all versions of those criteria selected the correct model in the case of the less complicated models ($K_0 = 2$ and $K_0 = 3$). In the case of the more complex models, $K_0 = 5$ and $K_0 = 7$, these criteria behave reasonably, as they tended to underestimate the number of hidden states, which match to the real representation of the observed process produced by the generating-data HMMs as displayed in Figures (6.3) and (6.4). The same above criteria also seem to perform well in the real data application, where they selected a reasonable solution to represent the data. It was noted that the effective number of parameters of DIC_{rec_2} gives values that increase up to $K = 4$. At the next states, $K = 5, 6$ and 7, there is only a slight increase in the effective number of parameters, indicating that the additional states do not considerably contribute to the model's deviance. This can explain the model fitting observed in Figure (6.5), where there is no marked improvement after $K = 4$. On the other hand, the first version of the DIC, DIC_{rec_1}, selected the correct model in all cases considered in this study, and it seems to prefer no overlapping solutions, but has a tendency to overestimate the number of hidden states in the model. In contrast, this criterion had a poor performance in the real data example, where it preferred the most overfit model. Also, it provides arbitrary values of $p_{\text{DIC}_{rec_1}}$ which do not agree with the principle of increasing the effective number of the parameters as the model complexity increases. This may be due to the unsatisfactory estimation of the plugged-in estimator $D_{rec}(\bar{\Theta})$ of this criterion.

2. For the conditional deviance-based criteria, some of them had a satisfactory performance. For instance, the two versions of the BIC: BIC_{con_1} and BIC_{con_2} had the

same behaviour as the AIC and BIC based on recursive deviance, where they also select the right model in the less complicated generating models, and under-fit model in the case of more complexity generating models. Is seems that these two criteria are also sensitive to the overlapping observed in data. In the real data example, the BIC$_{con_1}$ selects the more parsimonious model, whereas the BIC$_{con_2}$ favors a lager model. In the simulation example, the AIC$_{con_1}$ and BIC$_{con_3}$, except in the case $K_0 = 7$ which they selected the correct models, act symmetrically and both prefer the most overfitting models when $K_0 = 2$, $K_0 = 3$ and $K_0 = 5$ states, suggesting that they are not reliable. Moreover, both versions prefer the large models in the real data application. Conversely, the AIC$_{con_2}$, except the case $K_0 = 2$ which it chooses the most complicated model, picked up the correct number of hidden states in cases $K_0 = 3$, $K_0 = 4$ and $K_0 = 5$ states. Nevertheless, it also tends to select the more complicated model for the real data. The DIC$_{con_1}$ had the worse performance, as it selects the most overfitting model for all cases in the simulation study. However, it selects a reasonable solution for the real data and also introduces increased values of $p_{\text{DIC}_{con_1}}$, but large. The DIC$_{con_2}$ performed well for the first two simulated models, $K_0 = 2$ and $K_0 = 3$, and then tends to highly underestimate the number of states K in the main part of cases. This criteria also selected a sensible solution for the waiting time data example and provides $p_{\text{DIC}_{con_1}}$ with increased and large values.

3. The WAIC had an unsatisfactory behaviour, similarly to the versions AIC$_{con_1}$ and BIC$_{con_3}$, as it also tends to favour the most overfitting model for the first three cases (i.e. $K_0 = 2$, $K_0 = 3$ and $K_0 = 5$), and selects the correct model for the case $K_0 = 7$ with a high overestimate of K. Moreover, it tends to choose very complicated model for the the Old Faithful data. This can give an indicator that this criterion is not appropriated for HMMs.

4. It was noted that all criteria based on the recursive likelihood: the AIC, BIC and DIC provide generally lower MC standard errors than those based on the conditional likelihood and also the WAIC, indicating their accuracy. This high variability in the latter criteria may be because of the MCMC sampling includes high dimensional vectors of hidden state along with the model parameters, which would affect the variability of MC approximations of those criteria.

Chapter 7

Modeling and diagnosing of traffic crash rates using Poisson hidden Markov models

7.1 Introduction

This chapter presents an application of the HMM to traffic crash data. We use the HMM to model the spatial rather than the temporal dependency on a highway segment. We assume that crash counts can be considered as realizations of a Poisson random variable. The assumption of a standard Poisson distribution to describe such counts is often violated in applications due to the presence of over-dispersion (McCullagh and Nelder, 1989) in non-dependent data. This can be accommodated using alternatives such as the Negative Binomial distribution or by assuming components that are defined by independent (hidden) latent variables (McLachlan and Peel, 2000). However, these latter solutions assume independence between the latent variables and do not take into account the possibility of serial dependence in the data. We describe the PHMM as an alternative that allows both for over-dispersion and a specific form of serial correlation in the data. PHMMs have been used before for modelling over-dispersed series of count data. For example, Leroux and Puterman (1992) fitted a PHMM to a data set of movement counts by fetal lambs observed through ultrasound. Albert (1991) used such models to model the daily counts of epileptic patients seizures. Under a frequentist framework, Zucchini and MacDonald (2009) also used PHMMs to model the series of annual counts of earthquakes (i.e. magnitude of 7 or greater in the Richter scale) for the years 1900-2006. In epidemiological studies, the PHMM has also been used for modelling spatially structured heterogeneity (Green and Richardson, 2002) by means of a Pott's model. To illustrate the potential of our method we consider several data sets involving crashes counts which occurred in the UK over a 5-year period (2010-2014). We model the number of police reported crashes on individual segments relative to traffic volume. Our interest is to identify highway segments which have distinctly higher crash risk, and this classification problem is well suited to a HMM. In order to do this, we first need to select an optimal number of states.

7.2 Description and data preparation

We now describe the substantive application of this chapter, which involves traffic accidents occurring on the highway network in the UK. The crashes data considered in this study were obtained from the Department for Transport in Great Britain. Every road crash on the highway network is recorded on a STATS19 report form by police officers in the Department for Transport. This form provides detailed information about the time, location at the segment level, road condition, behaviour of the driver and the vehicles that were caused the accident. The accidents are recorded at a road-segment level that are labelled by references, measured as points, that refer to the locations of accidents (the easting and northing coordinates in a local British National Grid Coordinate system). The road network model is constructed from individual sections of road, that are segmented correctly in junction to junction link for management purposes, with different lengths. The accidents may be more likely near junctions. However, the available crash data are provided as a point process for each segment over a time period. Given the nature of the data available to us, we are compelled to work with network-constrained point data, which have been aggregated by road segments.

We consider reported road injury crashes over a 5-year period (2010-2014) in three motorways in the UK: the M5, M6 and M42 (Figure (7.1)). The M5 which links Exeter with Birmingham consists of 52 segments, the M6 which links Coventry with Carlisle city involves 90 segments and the M42 motorway which passes through the South East of Birmingham consists of 21 segments. The data needed in this study involve two formats. The first form involves data on the number of crashes, $\mathbf{y}$, at the road segment level for every year, obtained from the Department for Transport as an Open Government Archive (OGA, 2016) file. The second form provides information on the some traffic safety characteristics such as the length of segment, L, and the “Annual Average Daily Traffic flow” (AADT). AADT as defined by FHWA (2011) represents the officially estimated total annual average traffic flow on each segment. This second form of data is also obtained from the Department for Transport as an Open Data (OD, 2016) file. This latter allows us to derive a measure of exposure.

From this, we derive a measure of exposure which is akin to the population at risk used in epidemiology. This is the “Vehicle Miles Travelled” (VMT), which is simply the product of the segment length, L_t, and the segment AADT$_t$, so that (FHWA, 2011)

$$VMT_t = AADT_t \times L_t. \tag{7.1}$$

For computational convenience, we then compute the expected crash frequencies at segment level o_t by dividing the exposure of each segment (VMT$_t$) by the total exposure (VMT) and multiplying this by the total number of crashes, **y** (Li et al., 2007). Thus,

$$o_t = \frac{VMT_t}{\sum_{t=1}^{T} VMT_t} \times \sum_{t=1}^{T} \mathrm{y}_t. \tag{7.2}$$

In other words, we follow a conventional idea in epidemiology around modelling variations in the Standardized Morbidity Ratio (SMR), although in our case we model the “Observed Crash Rate ”(OCR) defined as:

$$OCR_t = \frac{\mathrm{y}_t}{o_t};\ t = 1, 2, ..., T_i, \tag{7.3}$$

where T_i refers to the number of road segments of the i^{th} motorway, and i= M5, M6 and M42. Figures (7.2-7.4) present the observed crash counts and the observed crash rates (OCR) at segment level of each motorway, respectively, which we wish to model using a Poisson HMM, as described in the next sections.

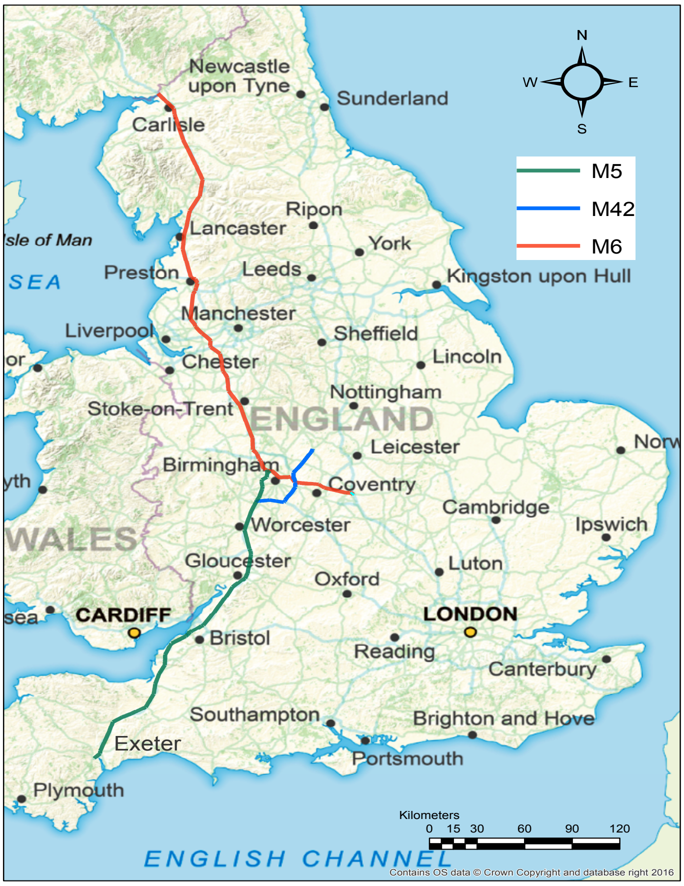


Figure 7.1: Plots of the M5, M6 and M42 motorways in the UK.

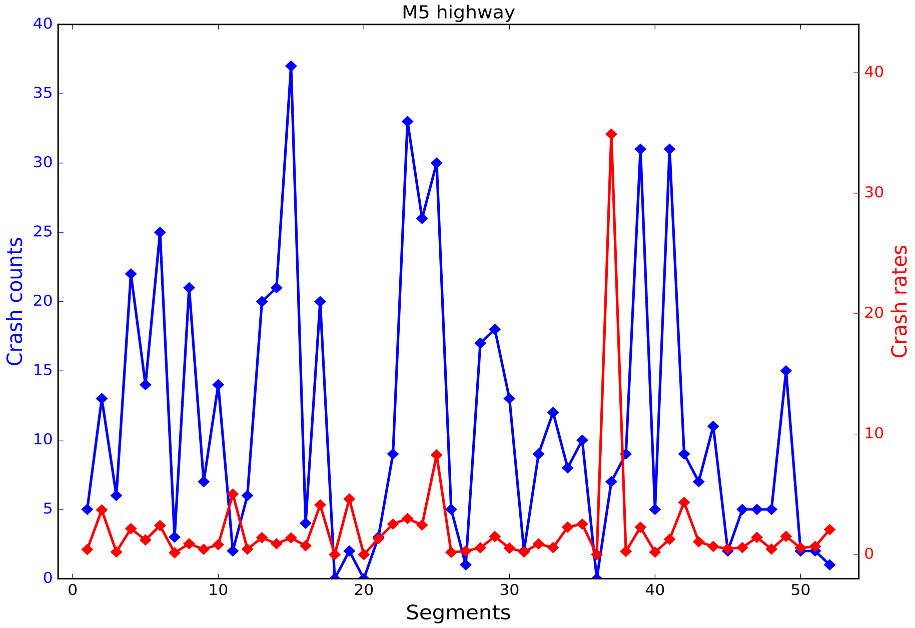


Figure 7.2: Plot of observed crash counts and rates at the segment level of the M5 highway.

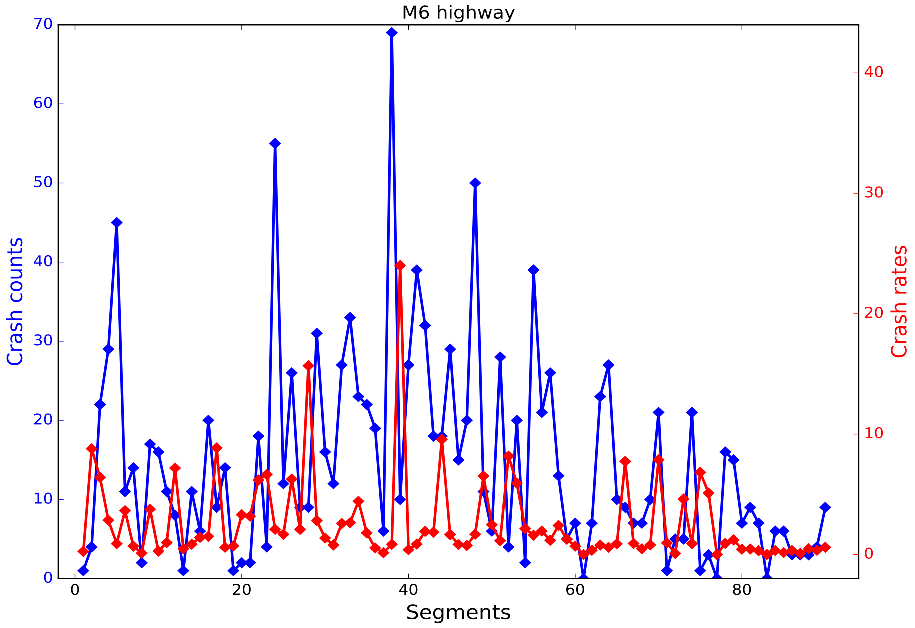


Figure 7.3: Plot of observed crash counts and rates at the segment level of the M6 highway.

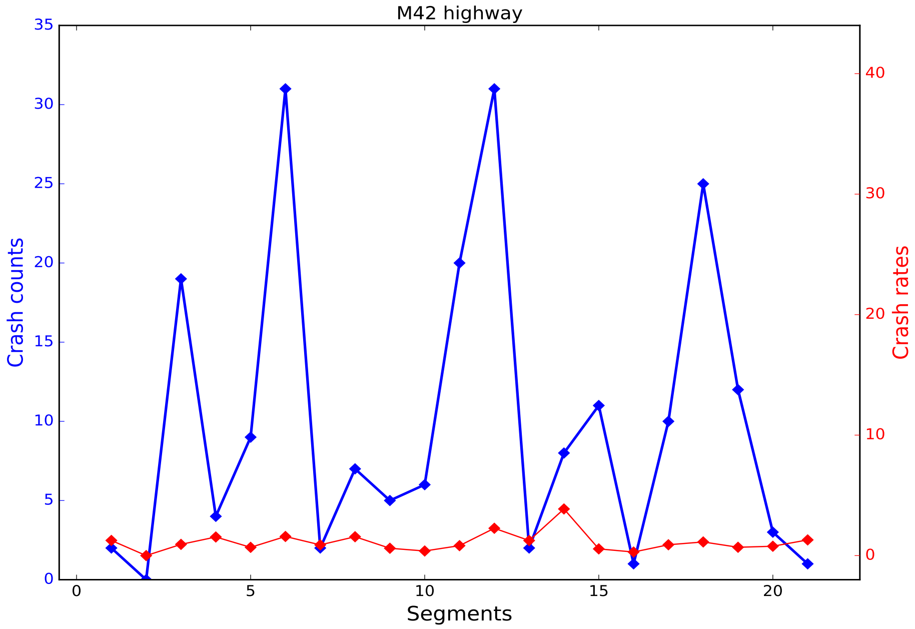


Figure 7.4: Plot of observed crash counts and rates at the segment level of the M42 highway.

7.3 Bayesian PHMM

We now need to construct a PHMM under the Bayesian framework. Obtaining the posterior distributions of a Bayesian PHMM is similar to the procedure for a Normal HMM presented in Chapter 4. The main difference is that the state-dependent distribution of the observed process Y follows a Poisson distribution $Y \in \mathbb{N}$, where $\mathbb{N}$ denotes non-negative integer values, with a rate parameter λ, depending only on a particular hidden state evolving over the state space. The motivation of using PHMMs is that they allow to model the unobserved heterogeneity and the serial dependency in crash rates among segments. We assume spatial dependency among segments through allowing each segment to be switched spatially according to a Markov state.

7.3.1 Model construction

Given a road segment t; $t = 1, 2, ..., T$, and reported crashes, Y_t, we seek to model

$$Y_t \sim Poi(o_t \lambda_t);\ \lambda_t > 0, \tag{7.4}$$

where o_t; $t = 1, 2, ..., T$, refers to the expected number of crashes given the length and traffic volume of segment t, explained earlier in Equation (7.2) and λ_t is an unknown crash rate parameter that can be viewed as the theoretical value of the OCR_t computed in Equation (7.3).

The probability mass function of the observed count of crashes is given by

$$Pr(Y_t = \mathrm{y}_t | o_t \lambda_t) \sim \frac{e^{-o_t\lambda_t}(o_t\lambda_t)^{\mathrm{y}_t}}{\mathrm{y}_t!}; \ \mathrm{y}_t = 0, 1, 2, ... \tag{7.5}$$

Now we model the unobserved heterogeneity in λ_t; $t = 1, 2, ..., T$, and the serial dependence using a PHMM. This assumes an underlying hidden state, z_t; $t = 1, 2, ..., T$, which follows a Markov process and takes values j in $\{1, 2, .., K\}$. Thus, the standard probability mass function in Equation (7.5) can be modified to take into account the state-specific heterogeneity, λ_{z_t}, between road segments, by defining the state-dependent probability mass as follows

$$Pr(Y_t = \mathrm{y}_t | o_t \lambda_{\mathrm{z}_t}) = \frac{e^{-o_t\lambda_{\mathrm{z}_t}}(o_t\lambda_{\mathrm{z}_t})^{\mathrm{y}_t}}{\mathrm{y}_t!}; \ \lambda_{\mathrm{z}_t} > 0. \tag{7.6}$$

In order to fit the model, we follow the data augmentation approach (Tanner and Wong, 1987) and write the complete likelihood function as

$$\begin{aligned} L_c(\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{o}\lambda; \mathbf{y}, \mathbf{z}) &= \pi_{\mathrm{z}_1} Pr(Y_1 = \mathrm{y}_1 | o_1 \lambda_{\mathrm{z}_1}) a_{\mathrm{z}_1\mathrm{z}_2} Pr(Y_2 = \mathrm{y}_2 | o_2 \lambda_{\mathrm{z}_2}) ... \ a_{\mathrm{z}_{T-1}\mathrm{z}_T} Pr(Y_T = \mathrm{y}_T | o_T \lambda_{\mathrm{z}_T}) \\ &= (\pi_{\mathrm{z}_1})(a_{\mathrm{z}_1\mathrm{z}_2} ... a_{\mathrm{z}_{T-1}\mathrm{z}_T})(Pr(Y_1 = \mathrm{y}_1 | o_1 \lambda_{\mathrm{z}_1}) Pr(Y_2 = \mathrm{y}_2 | o_2 \lambda_{\mathrm{z}_2}) ... \ Pr(Y_T = \mathrm{y}_T | o_T \lambda_{\mathrm{z}_T})) \\ &= \prod_{j=1}^{K} \pi_j^{N_j} \prod_{j=1}^{K} \prod_{k=1}^{K} (a_{jk})^{N_{jk}} \prod_{j=1}^{K} \prod_{t:\mathrm{z}_t=j}^{T} Pr(Y_t = \mathrm{y}_t | o_t \lambda_j) \\ &= \prod_{j=1}^{K} \pi_j^{N_j} \prod_{j=1}^{K} \prod_{k=1}^{K} (a_{jk})^{N_{jk}} \prod_{j=1}^{K} \prod_{t:\mathrm{z}_t=j}^{T} e^{-o_t\lambda_j}(o_t\lambda_j)^{\mathrm{y}_t}, \end{aligned} \tag{7.7}$$

where $N_j = \sum_{t=1}^{T} \mathbb{I}_{(\mathrm{z}_t=j)}$ denotes the number of transitions from state j at spatial segment t and $N_{jk} = \sum_{t=1}^{T-1} \mathbb{I}_{(\mathrm{z}_t=k, \mathrm{z}_{t-1}=j)}$ denotes the number of transitions from state j at spatial segment $t-1$ into the state k at spatial segment t. Hence, a Bayesian PHMM is given by

$$\begin{aligned} Pr(\boldsymbol{\pi}, \mathbf{A}, \lambda | \mathbf{y}, \mathbf{z}, \boldsymbol{o}) &= L_c(\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{o}\lambda; \mathbf{y}, \mathbf{z}) Pr(\boldsymbol{\pi}, \mathbf{A}, \lambda) \\ &= L_c(\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{o}\lambda; \mathbf{y}, \mathbf{z}) Pr(\boldsymbol{\pi}) Pr(\mathbf{A}) Pr(\lambda). \end{aligned} \tag{7.8}$$

By assuming independence of the priors of the model parameters, the joint prior distribution of all model parameters, $Pr(\boldsymbol{\pi}, \mathbf{A}, \lambda)$, in the first line of Equation (7.8) can be written as the product of the prior of the individual parameters as shown in the second line of the same equation. To complete our model specification, we need to assign priors for $\boldsymbol{\pi}$, $\mathbf{A}$ and λ. As with the Bayesian Normal HMM derived in Chapter (4), the priors of both $\boldsymbol{\pi}$ and each row of $\mathbf{A}$ are given independently; these are Dirichlet distributions with hyper-parameter δ. For the state-based rate parameter, λ, we assume independently a Gamma distribution, as a conjugate prior (Carlin and Louis, 2009), on each distinct rate, $\lambda_{z_t} \equiv \lambda_j$, such that

$$\begin{aligned} Pr(\lambda_j|\alpha,\beta) &\sim Gamma(\alpha,b), \\ &= \lambda_j^{a-1} e^{-\beta\lambda_j} \beta^{\alpha}/\Gamma(\alpha),\ \lambda_j > 0;\ \alpha > 0,\ \beta > 0, \end{aligned}$$

where α and β are hyper-parameters which represent the shape and rate or inverse scale parameters of the Gamma distribution respectively. The Gamma prior density above has mean α/β and variance α/β^2.

The Bayesian PHMM in Equation (7.8) can be rewritten as

$$\begin{aligned} Pr(\boldsymbol{\pi},\mathbf{A},\lambda|\mathbf{y},\mathbf{z},\mathbf{o}) = \prod_{j=1}^{K} \pi_j^{N_j} \prod_{j=1}^{K} \pi_j^{\delta_j - 1} \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{N_{jk}} \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{\delta_j - 1} \\ \times \prod_{j=1}^{K} \prod_{t:z_t=j}^{T} e^{-o_t\lambda_j} (o_t\lambda_j)^{y_t} \prod_{j=1}^{K} \lambda_j^{a-1} e^{-\beta\lambda_j}. \end{aligned} \tag{7.9}$$

A closed-form expression for the posterior distribution in Equation (7.9) is not available. We thus use the Gibbs sampler to simulate the posterior using the full conditional distribution of the model parameters.

7.3.2 Developing an MCMC algorithm

The joint distribution in Equation (7.9) above can be decomposed into full conditional distributions for the parameters. The full conditional distributions of the initial state vector, $\boldsymbol{\pi}$, and each row in the transition probability matrix, $a_{j.}$, are the same as those derived earlier with Normal HMM (Appendix A) which can be rewritten, respectively, as

$$Pr(\pi_j|\mathbf{y},\mathbf{z}) \propto \prod_{j=1}^{K} \pi^{N_j+\delta_j-1} = Dir(N_j+\delta_j), \tag{7.10}$$

$$Pr(a_{j.}|\mathbf{y},\mathbf{z}) \propto \prod_{j=1}^{K} a_{j.}{}^{N_{j.}+\delta_j-1} = Dir(N_{j.}+\delta_j), \tag{7.11}$$

whereas the full conditional posterior distribution for each λ_j is given by

$$\begin{aligned} Pr(\lambda_j|\mathbf{y},\boldsymbol{o},\mathbf{z}) &\propto \prod_{t=1}^{T} Pr(\mathrm{y}_t|o_t\lambda_j)Pr(\lambda_j|\alpha,\beta), \\ &\propto \prod_{t=1}^{T}\left[e^{(-o_t\lambda_j)}(o_t\lambda_j)^{\mathrm{y}_t}\right]\times\left[\lambda_j^{\alpha-1}e^{-\beta\lambda_j}\right], \\ &= \lambda_j^{\sum_{t:z_t=j}\mathrm{y}_t+\alpha-1}e^{-\lambda_j(\beta+\sum_{t:z_t=j}o_t)}, \end{aligned} \tag{7.12}$$

where the latter represents the kernel of a Gamma density, $Gamma(\alpha+\sum_{t:\mathrm{z}_t=j}\mathrm{y}_t,\beta+\sum_{t:\mathrm{z}_t=j}o_t)$. Sampling from the full conditional distribution of hidden states, $\mathbf{z}$, can be done by using the so-called Direct Gibbs (DG) sampler (Chib, 1996) as follows: For $t=1$,

$$Pr(\mathrm{z}_1=j|\mathrm{z}_2,...,\mathbf{y},\boldsymbol{o},\boldsymbol{\pi},\mathbf{A},\lambda)=\frac{\pi_j a_{j\mathrm{z}_2}\lambda_j^{\mathrm{y}_1}e^{-o_1\lambda_j}}{\sum_{l=1}^{K}\pi_l a_{l\mathrm{z}_2}\lambda_l^{\mathrm{y}_1}e^{-o_1\lambda_l}}=\mathbb{P}_{1j}, \tag{7.13}$$

for $2<t<T$ the full conditional distribution of z_t is

$$Pr(\mathrm{z}_t=j|...,\mathrm{z}_{t-1},\mathrm{z}_{t+1},...,\mathbf{y},\boldsymbol{o},\mathbf{A},\lambda)=\frac{a_{\mathrm{z}_{t-1}j}a_{j\mathrm{z}_{t+1}}\lambda_j^{\mathrm{y}_t}e^{-o_t\lambda_j}}{\sum_{l=1}^{K}a_{\mathrm{z}_{t-1}l}a_{l\mathrm{z}_{t+1}}\lambda_j^{\mathrm{y}_t}e^{-o_t\lambda_j}}=\mathbb{P}_{tj}, \tag{7.14}$$

and for $t=T$

$$Pr(\mathrm{z}_T=j|...\mathrm{z}_T,\mathbf{y},\boldsymbol{o},\mathbf{A},\lambda)=\frac{a_{\mathrm{z}_{T-1}j}\lambda_j^{\mathrm{y}_T}e^{-o_T\lambda_j}}{\sum_{l=1}^{K}a_{\mathrm{z}_{T-1}l}\lambda_j^{\mathrm{y}_T}e^{-o_T\lambda_j}}=\mathbb{P}_{Tj}, \tag{7.15}$$

where $\mathbb{P}$ represents the posterior allocation matrix, of dimension $(T\times K)$ which summarizes all posterior probabilities of hidden states. Given $\mathbb{P}$, a sequence of discrete values of the hidden states of length T can be locally drawn from a multinomial distribution as

$$\begin{aligned} \mathrm{z}_1 &= j\sim\text{Multinomial}(\mathbb{P}_1.), \\ \mathrm{z}_t &= j\sim\text{Multinomial}(\mathbb{P}_t.), \\ \mathrm{z}_T &= j\sim\text{Multinomial}(\mathbb{P}_T.),\ \text{for } j=1,2,...,K. \end{aligned} \tag{7.16}$$

Given a Poisson HMM to model the safety traffic crashes, we obtain the posterior classification probabilities for each segment t to belong to a hidden state j. This can be accomplished implicitly via calculating the posterior marginal distribution of each state j from Equation (7.16), i.e. the number of states j visited by the segment t. Given m; $m=1,2,...,M$ of MCMC iterations, this can be equivalently written as

$$\hat{Pr}(z_t = j) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}(z_t^{(m)} = j), \ \forall j = 1,2,...,K. \tag{7.17}$$

The hidden states estimated marginally in Equation (7.17) can allow interpretation of the state-distinct expected crash rate at segment level and identification of the segment(s) with potentially higher crash rate(s) than others.

7.3.3 Prior specification

As before, a natural prior distribution for the parameters of the hidden part of the model, initial state, $\boldsymbol{\pi}$, and transition probabilities matrix, $\mathbf{A}$, is the Dirichlet distribution. We assume that the $\boldsymbol{\pi}$ and each row in the matrix $\mathbf{A} = \{a_{j.}\}$ are each assigned independently a Dirichlet prior with a hyper-parameter δ equal to 1, i.e. $\delta_j = 1, \forall j \in \{1,2,...,K\}$ (Cappé et al., 2005; Rydén, 2008). We use a conjugate gamma prior for the rate parameter λ. We conduct a sensitivity analysis to check the effects of changing the values of α and β, the hyper-priors for the crash rate, so that we consider a range of priors shown in Table (7.1).

Priors	mean	Variance
$Gamma(0.1, 0.1)$	1	10
$Gamma(0.01, 0.01)$	1	100
$Gamma(0.001, 0.001)$	1	1000
$Gamma(0.0001, 0.0001)$	1	10000

Table 7.1: Four proposed gamma priors with different parameterizations.

7.4 Model selection and assessment

For our intended application, we need to select the "best" model in terms of the number of states in order to provide a readily understood estimate of the probability of a particular state belonging to the highest risk category. Particularly, we conduct the process of model selection by employing the criteria: AIC$_{rec_1}$, BIC$_{rec_1}$, AIC$_{rec_2}$, BIC$_{rec_2}$, AIC$_{rec_3}$, BIC$_{rec_3}$, DIC$_{rec_2}$, BIC$_{con_1}$ and BIC$_{con_2}$ which were found to work well as introduced in Chapter 5.

Besides using model selection criteria to select the best model, we assess the adequacy of the model using a variety of goodness-of-fit testing tools. One approach is to plot the posterior predictive distribution (PPD, Gelman et al. (1996)). Here we take a sequence of replicated or predictive observations, y_t^*; $t = 1,2,...,T$, so that the PPD for a PHMM can be written as

$$Pr(\mathrm{y}_t^*|\mathbf{y}) = \int\int Pr(\mathrm{y}_t^*|\boldsymbol{o\lambda},\mathbf{z})p_{\text{post}}(\boldsymbol{o\lambda},\mathbf{z}|\mathbf{y})d\mathbf{z}d\boldsymbol{o\lambda}, \tag{7.18}$$

where $p_{\text{post}}(\boldsymbol{o\lambda},\mathbf{z}|\mathbf{y})$ represents the joint posterior distribution of the observed process, $\boldsymbol{o\lambda}$, and the unobserved process, $\mathbf{z}$. Given samples of the crash rate parameter, $\lambda^{(m)}$, and hidden states, $\mathbf{z}^{(m)}$, drawn from an MCMC run, the predictive data of a PHMM can be approximated as

$$Y_t^{*(m)} \sim Poi(o_t\lambda^{(m)}_{\mathrm{z}_t^{(m)}});\ t = 1,2,...,T. \tag{7.19}$$

The model adequacy can be then checked based on the its predictive ability by the visual checking to the degree the closeness or discrepancy between the mean of posterior predictive distribution, $\bar{Y}_t^*$, accounted for by the center of predictive interval, and the observed crash count, y_t, at level each segment t.

We also adopt a method proposed by Zucchini and MacDonald (2009) and use pseudo-residuals. Zucchini and MacDonald (2009) introduced two kinds of pseudo-residuals. For discrete observations, the ordinary pseudo-residuals can be defined as line segments $\left[r_t^+; r_t^*\right]$, where

$$r_t^- = \Phi^{-1}(u_t^{-1}),$$

and

$$r_t^+ = \Phi^{-1}(u_t^{+1}),$$

where Φ^{-1} denotes the inverse distribution function of a standard Normal-distributed random variable, and by following the notation in Zucchini and MacDonald (2009),

$$u_t^- = Pr(\mathrm{X}_t < \mathrm{x}_t|\mathrm{X}_{(-t)} = \mathrm{x}_{(-t)}), \tag{7.20}$$

and

$$u_t^+ = Pr(\mathrm{X}_t \leq \mathrm{x}_t|\mathrm{X}_{(-t)} = \mathrm{x}_{(-t)}). \tag{7.21}$$

For continuous observations, the ordinary pseudo-residuals are defined as

$$u_t = Pr(\mathrm{X}_t \leq \mathrm{x}_t | \mathrm{X}_{(-t)} = \mathrm{x}_{(-t)}), \tag{7.22}$$

where $\mathrm{x}_{(-t)}$ denotes the vector of all data except x_t, i.e. $\mathrm{x}_{(-t)} = (\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_{t-1}, \mathrm{x}_{t+1}, ..., \mathrm{x}_T)$. We extend the idea of pseudo-residuals from a Bayesian viewpoint. Given M replicated data $Y_t^*; t = 1, 2, ..., T$, simulated from the PPD in Equation (7.19), we can rewrite the probabilities in Equation (7.22) as follows

$$u_t = Pr(\mathrm{Y}_t^* \leq \mathrm{y}_t | \mathrm{Y}_{(-t)}^* = \mathrm{y}_{(-t)}, \mathbf{z}, \boldsymbol{o}\boldsymbol{\lambda}), \tag{7.23}$$

which can be approximated over an MCMC run by

$$\bar{u}_t = \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}(\mathrm{Y}_t^{*(m)} \leq \mathrm{y}_t | \mathbf{z}^{(m)}, \boldsymbol{o}\boldsymbol{\lambda}^{(m)}). \tag{7.24}$$

We are interested in comparing the proportion of each segment's predictions with the observed counts. These predictive proportions, $\bar{u}$, are obtained by averaging predictions over M iterations as shown in Equation (7.24) which are no longer discrete. When ordinary pseudo-residuals follow a standard Normal distribution, this can be considered as an indicator of model adequacy (Zucchini and MacDonald, 2009). Therefore, the validation of the fitted model can be assessed graphically using tools such as a QQ-plot.

7.5 Model fitting

In this section, we first fit a certain number of competing models for each dataset of the three highways; M5, M6 and M42 adopted in this study. The aim is to select the best model for each data set. As before, we set an upper bound, K_{max}, on the number of competing models. Care is taken here given the small sample sizes (Gelman et al., 2014). Zucchini and MacDonald (2009) also pointed out that a reasonable upper bound on the number of states should be selected that is suitable for the number of observations. This may lead to unsatisfactory behaviour of the likelihood and possibly invalid model selection criteria.

We set $K_{max(42)} = 4$ states as an upper bound on the number of competing models that are being fitted to the dataset of the M42 motorway ($T = 21$). We set $K_{max(M5)} = 5$ on the number of competing models that are being fitted to the dataset reported from the M5 motorway ($T = 52$). Finally, we set $K_{max(M6)} = 6$ on those reported from the M6 motorway ($T = 90$).

At the same time, for each candidate model being fitted to each data set, using a certain number of MCMC iterations, we investigate the effect of the priors chosen on the resulting posterior distributions of the crash rate parameter, λ. We are interested in checking the sensitivity of the crash rate parameter, λ, to the prior choice, due to the small size of datasets considered in this study. The results are discussed based on convergence diagnostics.

7.5.1 MCMC sampling

For each competing model fitted to each dataset considered in this study, we use the Gibbs sampler to obtain the posterior distributions of all model parameters.

For each study, we ran the Gibbs sampler taking into account the following sampling information. We sampled 104000 iterations, as main samples, and discarded the first 4000 iterations as a burn-in period. To avoid the possibility of obtaining correlated samples, we thinned the remaining 100000 iterations by keeping every 100^{th} iteration to obtain 1000 thinned samples. We used the Gelman-Rubin statistics $\hat{R}$, (Gelman and Rubin, 1992) and the Geweke's diagnostic $\hat{G}$, (Geweke, 1992) to check convergence. The Gelman-Rubin statistic is based on running multiple sequences; we use three chains with highly dispersed starting points.

7.6 Results and discussion

7.6.1 The M5 motorway data

7.6.1.1 Convergence results of the M5 motorway data

In this section, we display the convergence results of each crash rate parameter, λ, from the models (K=2,3,4 and 5) fitted to the M5 motorway data ($T = 52$), given four proposed priors. Figures (7.5-7.8) display the ACF plots of each crash rate parameter of those models, given chosen priors, which suggest no indication of autocorrelation. Values of the Gelman-Rubin statistic, $\hat{R}$, were less than 1.1 for all crash rate parameters of considered models under all chosen priors (see Tables (7.2-7.5)). Conversely, the Geweke statistic, $\hat{G}$, provided some values that lie outside the interval $[-2,\ 2]$ for some models, given highly diffused priors such as $Gamma(0.001, 0.001)$ and $Gamma(0.0001, 0.0001)$, indicating that convergence is not achieved. The problem of convergence results from different diagnostics is noted in the literature (Cowles and Carlin, 1996).

In Figure (7.9), we display only the trace-plots of the full posterior distributions, produced from running three chains, each of which begins from different starting points, of each state-specific crash rate parameter, λ_j; $j = 1,2,3$, sampled from a 3-state PHMM, given the prior $Gamma(0.1,0.1)$.

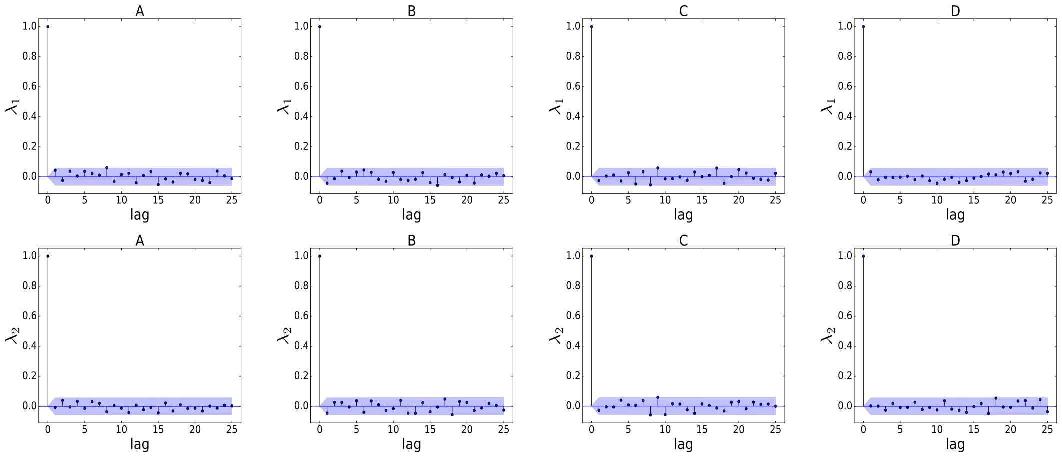


Figure 7.5: The plots of ACF functions of the crash rate parameter λ_j of a 2-state PHMM given priors: (A): $Gamma(0.1,0.1)$, (B): $Gamma(0.01,0.01)$, (C): $Gamma(0.001,0.001)$, (D): $Gamma(0.0001,0.0001)$.

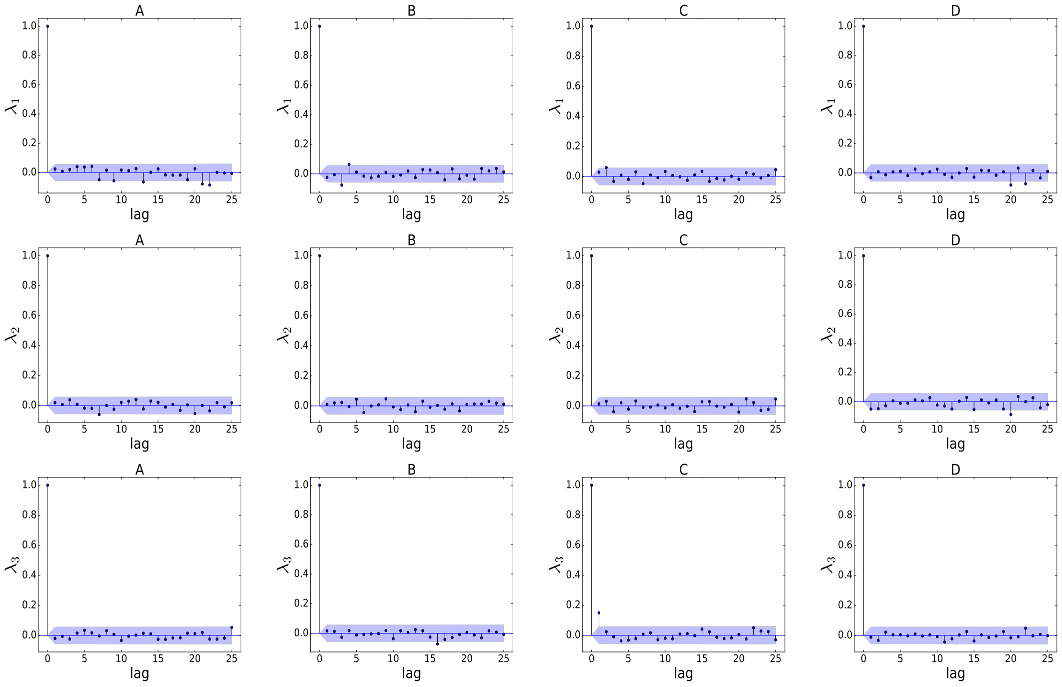


Figure 7.6: The plots of ACF functions of the crash rate parameter λ_j of a 3-state PHMM given priors: (A): $Gamma(0.1,0.1)$, (B): $Gamma(0.01,0.01)$, (C): $Gamma(0.001,0.001)$, (D): $Gamma(0.0001,0.0001)$.

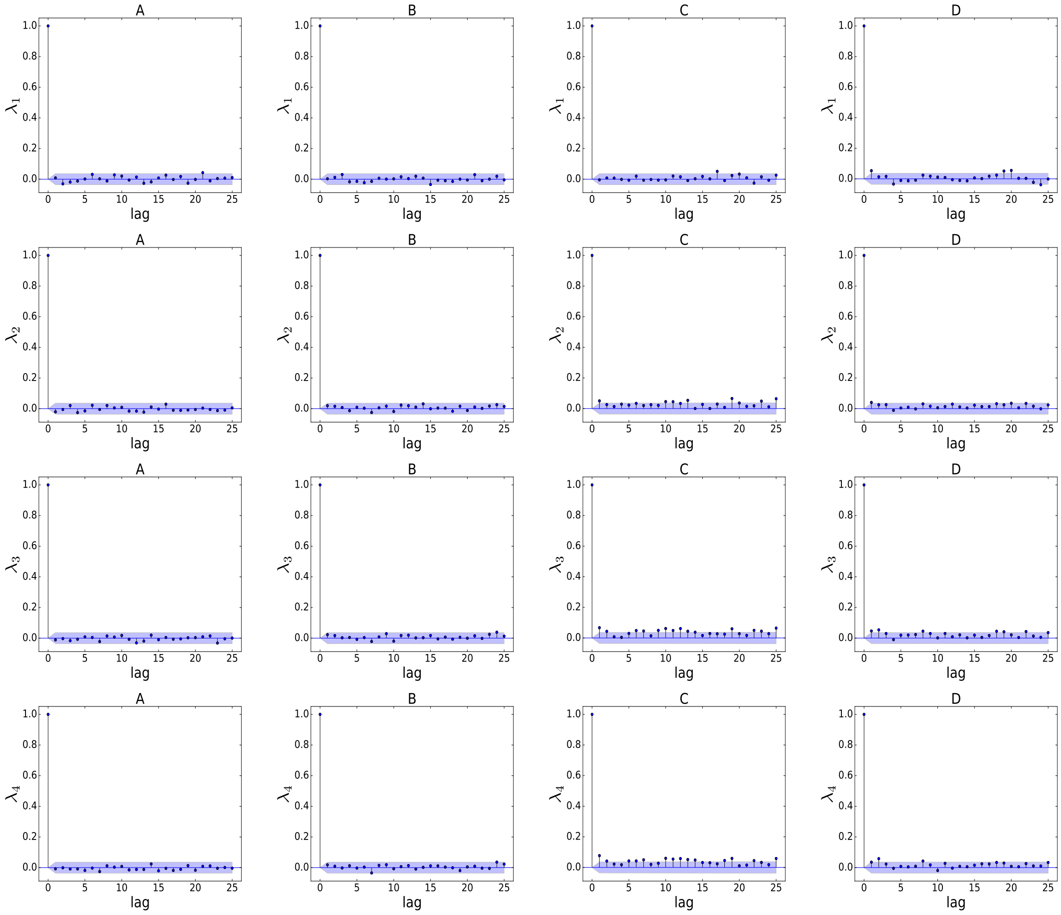


Figure 7.7: The plots of ACF functions of the crash rate parameter λ_j of a 4-state PHMM with priors: (A): *Gamma*$(0.1, 0.1)$, (B): *Gamma*$(0.01, 0.01)$, (C): *Gamma*$(0.001, 0.001)$, (D): *Gamma*$(0.0001, 0.0001)$.

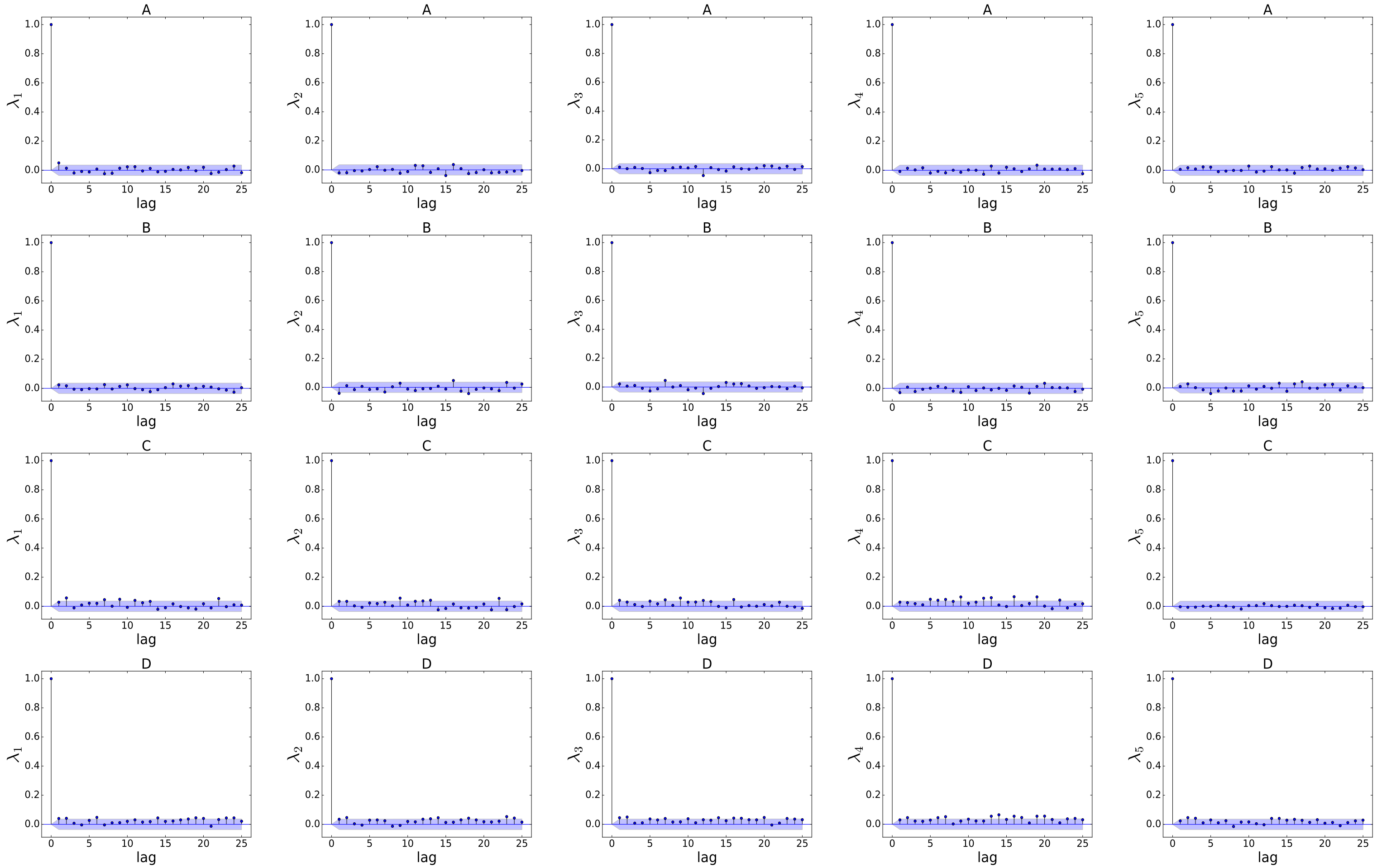


Figure 7.8: The plots of ACF functions of the crash rate parameter λ_j of a 5-state PHMM with priors: (A): $Gamma(0.1, 0.1)$, (B): $Gamma(0.01, 0.01)$, (C): $Gamma(0.001, 0.001)$, (D): $Gamma(0.0001, 0.0001)$.

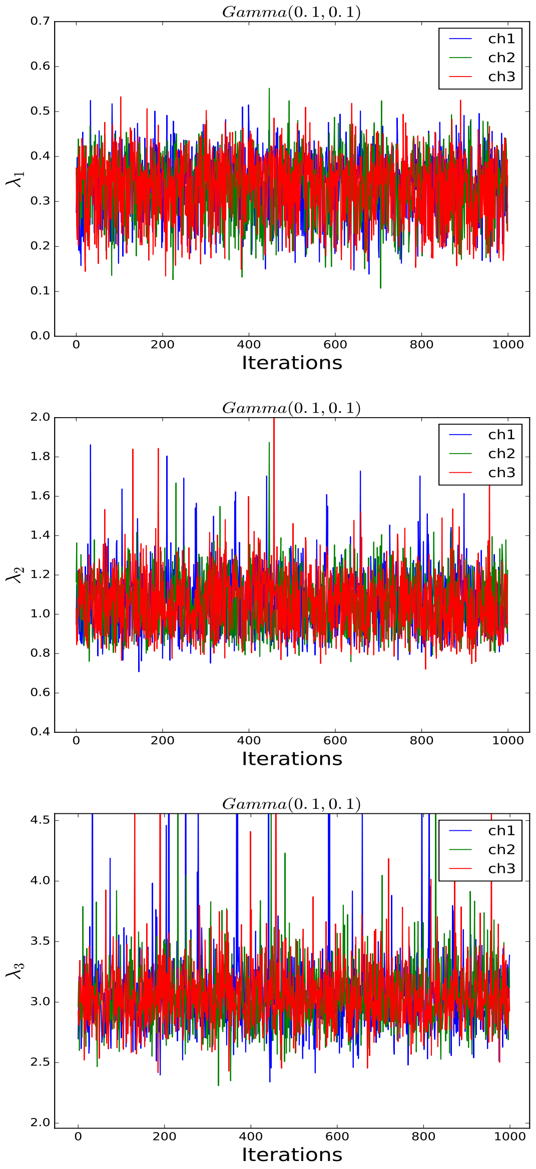


Figure 7.9: Trace plots for the thinned samples of state-dependent crash rate parameters, $\lambda_j; j = 1, 2, 3$, sampled from a 3-state PHMM under a $Gamma(0.1, 0.1)$ prior.

7.6.1.2 Results of the estimation of the crash rate parameter of the M5 motorway data

Tables (7.2-7.5) and Figures (7.10-7.13) present the estimation results of each crash rate parameter, λ_j, as well as 95% credible intervals from models fitted to the M5 motorway data, given four prior choices. We can see that the model with $K = 2$ provides very similar estimates. The same situation is noted in relation to the model with $K = 3$, where the posterior estimates of the state-specific crash rate parameters and their corresponding credible intervals were all somewhat similar as shown in Table (7.3) and Figure (7.11). This may suggest that the parameter estimates of these models are not sensitive to the choice of priors. However, as we increase the model's size, the models with states $K = 4$ and $K = 5$, some posterior estimates of λ were sensitive when using non-informative priors, as shown in Tables (7.4-7.5) and the box-plots in Figures (7.12-7.13), respectively.

Estimation and convergence findings have shown that when the fitted model's size was increased, its parameters were more sensitive to the prior choice and possibly fail to converge. This could be due to the small size of data.

Priors	$\hat{\lambda}_k$	Mean	95% CI			$\hat{R}$	$\hat{G}$
			2.5%	Median	97.5%		
$Gamma(0.1, 0.1)$	$\hat{\lambda}_1$	0.495	0.423	0.494	0.573	0.999	0.257
$Gamma(0.01, 0.01)$	$\hat{\lambda}_1$	0.496	0.422	0.495	0.576	0.999	-0.595
$Gamma(0.001, 0.001)$	$\hat{\lambda}_1$	0.495	0.421	0.495	0.573	1.000	-0.090
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_1$	0.496	0.422	0.495	0.575	1.001	1.373
$Gamma(0.1, 0.1)$	$\hat{\lambda}_2$	2.085	1.858	2.081	2.336	0.999	0.027
$Gamma(0.01, 0.01)$	$\hat{\lambda}_2$	2.086	1.856	2.084	2.336	1.000	-1.159
$Gamma(0.001, 0.001)$	$\hat{\lambda}_2$	2.088	1.857	2.085	2.347	0.999	0.253
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_2$	2.087	1.862	2.084	2.340	0.999	-0.033

Table 7.2: Results of the estimation and convergence of the rate parameter of a 2-state PHMM, given four gamma priors. The third column provides the ergodic posterior means of the rate parameter. The fourth column provides the median and 95% CI. The last two columns include the values of the Gelman-Rubin statistic, $\hat{R}$, and the Geweke statistic, $\hat{G}$, respectively.

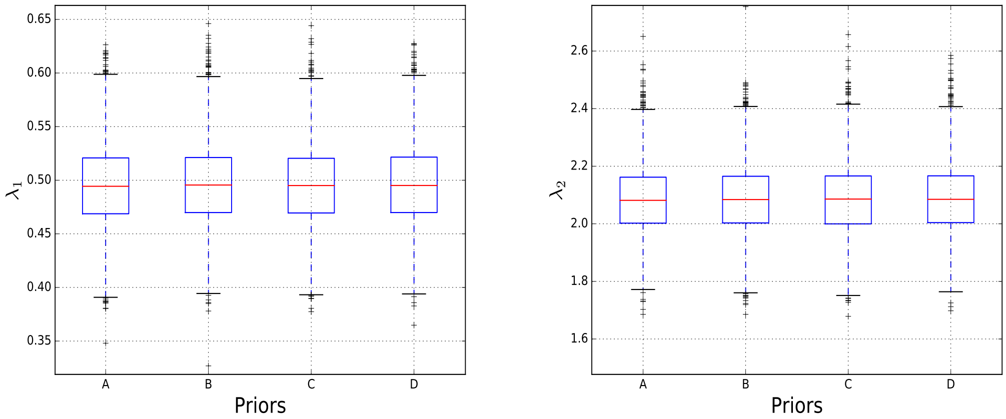


Figure 7.10: Box-plots of the estimates of the crash rate parameter λ_j for a 2-state PHMM given priors: (A): $Gamma(0.1, 0.1)$, (B): $Gamma(0.01, 0.01)$, (C): $Gamma(0.001, 0.001)$, (D): $Gamma(0.0001, 0.0001)$.

Priors	$\hat{\lambda}_k$	Mean	95% CI			$\hat{R}$	$\hat{G}$
			2.5%	Median	97.5%		
$Gamma(0.1, 0.1)$	$\hat{\lambda}_1$	0.336	0.186	0.348	0.459	1.001	1.426
$Gamma(0.01, 0.01)$	$\hat{\lambda}_1$	0.335	0.183	0.349	0.462	0.999	0.379
$Gamma(0.001, 0.001)$	$\hat{\lambda}_1$	0.334	0.171	0.348	0.465	1.000	0.775
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_1$	0.326	0.154	0.344	0.457	1.000	-3.072
$Gamma(0.1, 0.1)$	$\hat{\lambda}_2$	1.065	0.811	1.067	1.363	1.001	1.295
$Gamma(0.01, 0.01)$	$\hat{\lambda}_2$	1.068	0.817	1.066	1.369	0.999	0.821
$Gamma(0.001, 0.001)$	$\hat{\lambda}_2$	1.066	0.803	1.065	1.404	1.000	0.908
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_2$	1.050	0.783	1.055	1.358	0.999	-2.835
$Gamma(0.1, 0.1)$	$\hat{\lambda}_3$	3.057	2.628	3.019	3.649	0.999	0.478
$Gamma(0.01, 0.01)$	$\hat{\lambda}_3$	3.081	2.631	3.032	3.682	1.000	-1.623
$Gamma(0.001, 0.001)$	$\hat{\lambda}_3$	3.197	2.573	3.025	3.197	1.000	0.472
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_3$	3.037	2.489	3.016	3.037	0.999	-2.454

Table 7.3: Results of the estimation and convergence of the rate parameter of a 3-state PHMM, given four gamma priors. The third column provides the ergodic posterior means of the rate parameter. The fourth column provides the median and 95% CI. The last two columns include the values of the Gelman-Rubin statistic, $\hat{R}$, and the Geweke statistic, $\hat{G}$, respectively.

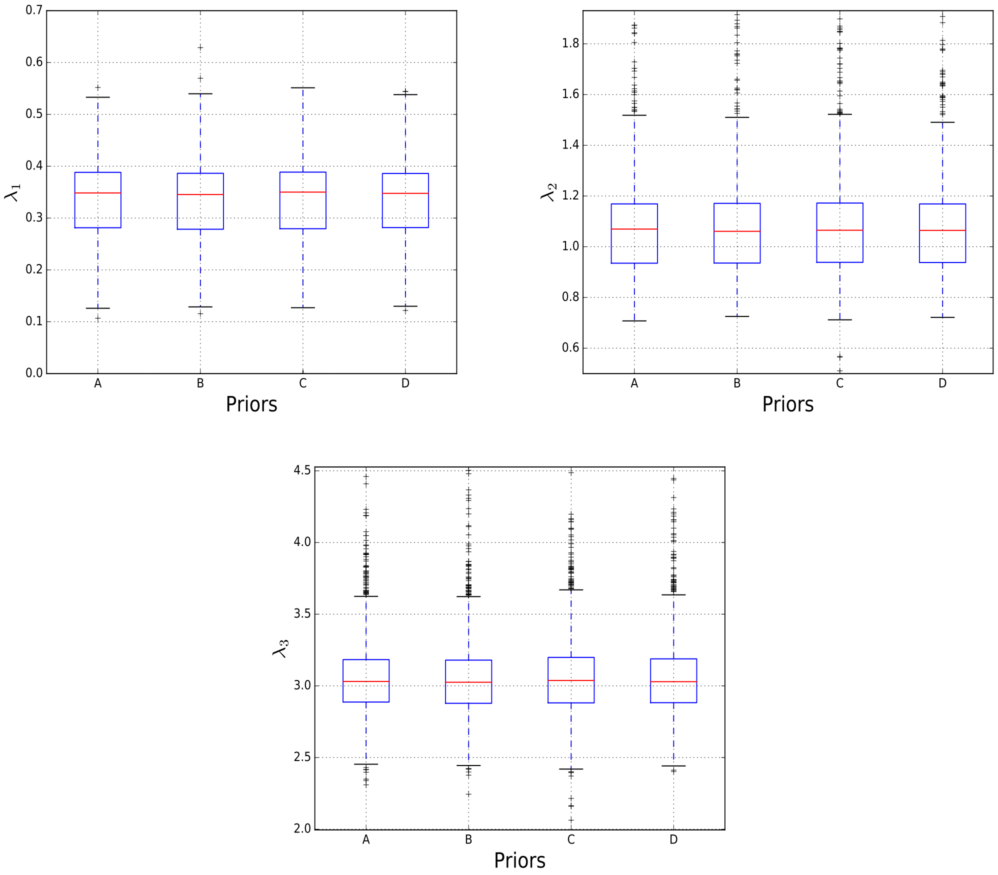


Figure 7.11: Box-plots of the estimates of the crash rate parameter λ_j for a 3-state PHMM given priors: (A): $Gamma(0.1, 0.1)$, (B): $Gamma(0.01, 0.01)$, (C): $Gamma(0.001, 0.001)$, (D): $Gamma(0.0001, 0.0001)$.

Priors	$\hat{\lambda}_k$	Mean	95% CI			$\hat{R}$	$\hat{G}$
			2.5%	Median	97.5%		
$Gamma(0.1, 0.1)$	$\hat{\lambda}_1$	0.251	0.154	0.242	0.396	1.001	-1.903
$Gamma(0.01, 0.01)$	$\hat{\lambda}_1$	0.249	0.153	0.238	0.403	0.999	0.279
$Gamma(0.001, 0.001)$	$\hat{\lambda}_1$	0.254	0.141	0.242	0.413	1.005	1.683
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_1$	0.248	0.147	0.240	0.404	1.001	0.408
$Gamma(0.1, 0.1)$	$\hat{\lambda}_2$	0.711	0.493	0.658	1.173	1.000	1.128
$Gamma(0.01, 0.01)$	$\hat{\lambda}_2$	0.721	0.498	0.665	1.164	1.003	-1.192
$Gamma(0.001, 0.001)$	$\hat{\lambda}_2$	0.758	0.478	0.691	1.212	1.029	6.242
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_2$	0.718	0.482	0.661	1.172	1.010	-0.814
$Gamma(0.1, 0.1)$	$\hat{\lambda}_3$	1.596	1.045	1.337	2.865	1.001	-1.825
$Gamma(0.01, 0.01)$	$\hat{\lambda}_3$	1.654	1.049	1.361	2.883	1.004	-2.494
$Gamma(0.001, 0.001)$	$\hat{\lambda}_3$	1.801	1.041	1.469	2.918	1.036	7.143
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_3$	1.649	1.054	1.352	2.856	1.014	-0.450
$Gamma(0.1, 0.1)$	$\hat{\lambda}_4$	4.505	2.672	3.177	10.953	1.000	-1.520
$Gamma(0.01, 0.01)$	$\hat{\lambda}_4$	4.839	2.686	3.216	11.401	1.002	-1.816
$Gamma(0.001, 0.001)$	$\hat{\lambda}_4$	5.577	2.707	3.419	11.725	1.038	7.465
$Gamma(0.0001, 0.0001)$	$\hat{\lambda}_4$	4.846	2.701	3.222	11.568	1.014	-0.188

Table 7.4: Results of the estimation and convergence of the rate parameter of a 4-state PHMM, given four gamma priors. The third column provides the ergodic posterior means of the rate parameter. The fourth column provides the median and 95% CI. The last two columns include the values of the Gelman-Rubin statistic, $\hat{R}$, and the Geweke statistic, $\hat{G}$, respectively.

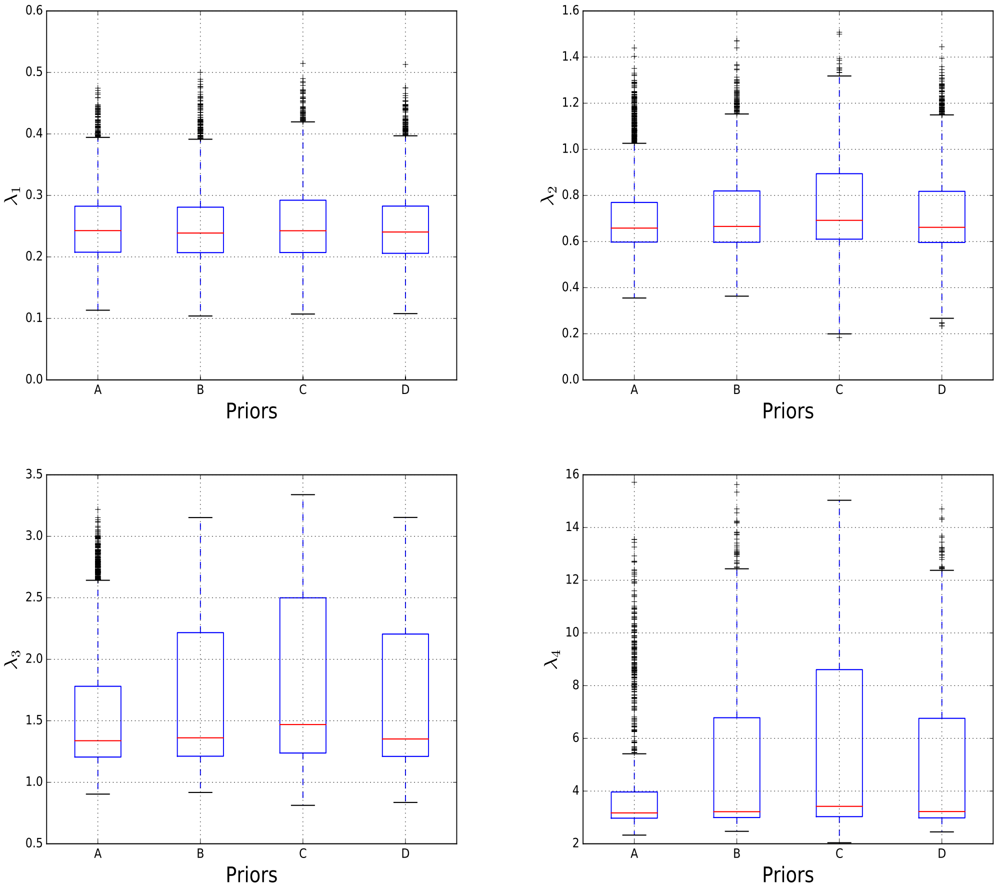


Figure 7.12: Box-plots of the estimates of the crash rate parameter λ_j for a 4-state PHMM given priors: (A): $\mathit{Gamma}(0.1, 0.1)$, (B): $\mathit{Gamma}(0.01, 0.01)$, (C): $\mathit{Gamma}(0.001, 0.001)$, (D): $\mathit{Gamma}(0.0001, 0.0001)$.

Priors	$\hat{\lambda}_k$	Mean	95% CI			$\hat{R}$	$\hat{G}$
			2.5%	Median	97.5%		
$Gamma(0.1,0.1)$	$\hat{\lambda}_1$	0.227	0.144	0.225	0.324	0.999	1.261
$Gamma(0.01,0.01)$	$\hat{\lambda}_1$	0.215	0.001	0.222	0.325	1.000	0.882
$Gamma(0.001,0.001)$	$\hat{\lambda}_1$	0.190	0.001	0.215	0.319	1.015	0.455
$Gamma(0.0001,0.0001)$	$\hat{\lambda}_1$	0.177	0.001	0.209	0.315	1.021	-1.526
$Gamma(0.1,0.1)$	$\hat{\lambda}_2$	0.602	0.398	0.607	0.758	0.999	-0.056
$Gamma(0.01,0.01)$	$\hat{\lambda}_2$	0.594	0.244	0.610	0.769	0.999	-0.115
$Gamma(0.001,0.001)$	$\hat{\lambda}_2$	0.551	0.192	0.594	0.753	1.014	1.331
$Gamma(0.0001,0.0001)$	$\hat{\lambda}_2$	0.533	0.181	0.588	0.751	1.019	-1.218
$Gamma(0.1,0.1)$	$\hat{\lambda}_3$	1.203	0.680	1.218	1.547	1.000	0.447
$Gamma(0.01,0.01)$	$\hat{\lambda}_3$	1.187	0.624	1.213	1.538	0.999	-0.062
$Gamma(0.001,0.001)$	$\hat{\lambda}_3$	1.140	0.568	1.195	1.512	1.015	1.335
$Gamma(0.0001,0.0001)$	$\hat{\lambda}_3$	1.105	0.544	1.176	1.508	1.024	-1.583
$Gamma(0.1,0.1)$	$\hat{\lambda}_4$	2.661	2.270	2.659	3.073	1.001	0.817
$Gamma(0.01,0.01)$	$\hat{\lambda}_4$	2.513	1.233	2.606	3.054	1.000	-0.927
$Gamma(0.001,0.001)$	$\hat{\lambda}_4$	2.438	1.163	2.592	3.090	1.021	1.687
$Gamma(0.0001,0.0001)$	$\hat{\lambda}_4$	2.371	1.131	2.569	3.059	1.030	-0.977
$Gamma(0.1,0.1)$	$\hat{\lambda}_5$	9.154	5.741	9.096	12.599	1.002	1.600
$Gamma(0.01,0.01)$	$\hat{\lambda}_5$	8.234	2.921	8.600	12.307	0.999	-0.997
$Gamma(0.001,0.001)$	$\hat{\lambda}_5$	8.437	2.856	8.689	12.785	1.000	1.956
$Gamma(0.0001,0.0001)$	$\hat{\lambda}_5$	7.874	2.854	8.472	12.361	1.015	-1.119

Table 7.5: Results of the estimation and convergence of the rate parameter of a 5-state PHMM, given four gamma priors. The third column provides the ergodic posterior means of the rate parameter. The fourth column provides the median and 95% CI. The last two columns include the values of the Gelman-Rubin statistic, $\hat{R}$, and the Geweke statistic, $\hat{G}$, respectively.

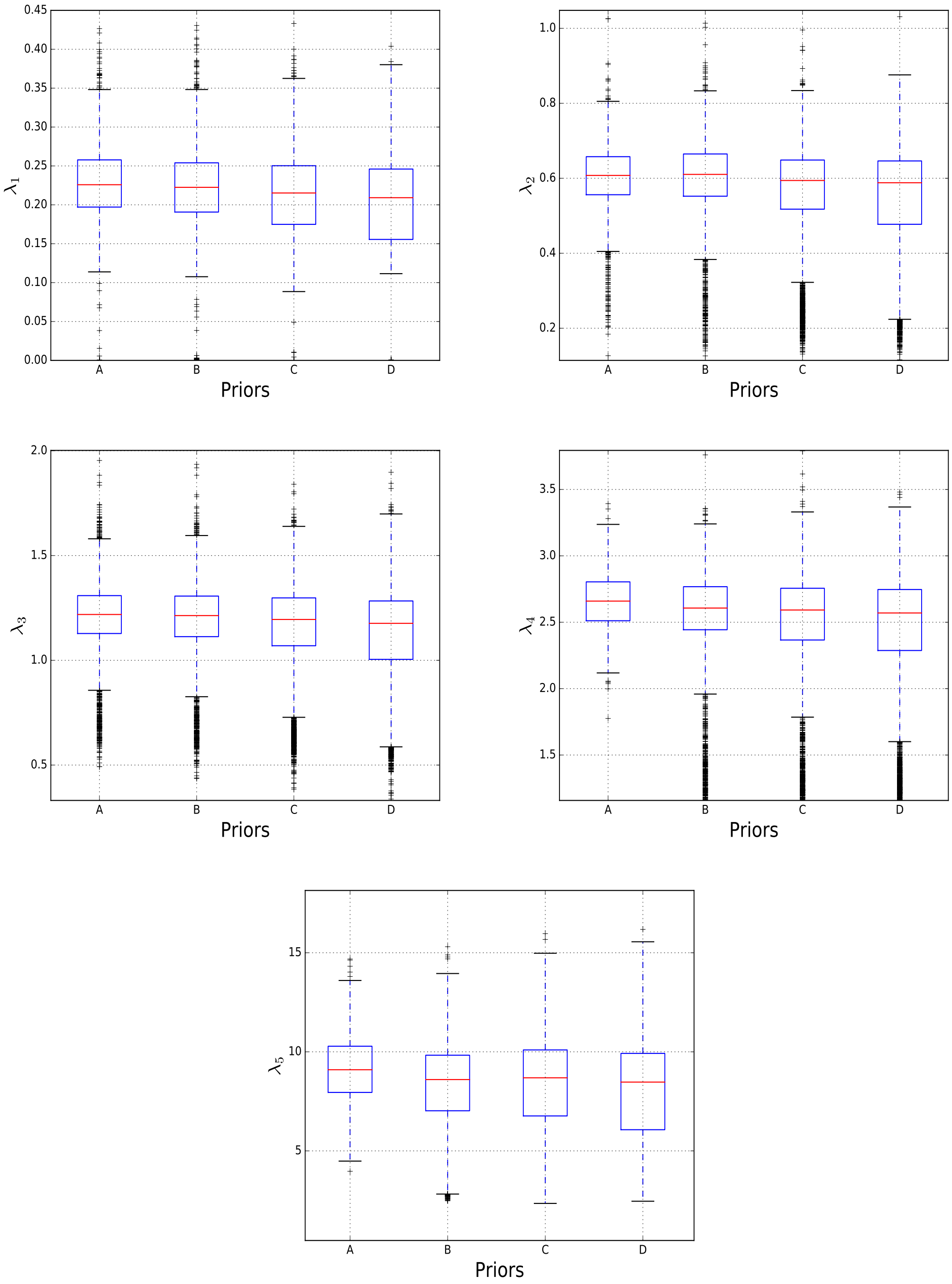


Figure 7.13: Box-plots of the estimates of the crash rate parameter λ_j for a 5-state PHMM given priors: (A): $Gamma(0.1, 0.1)$, (B): $Gamma(0.01, 0.01)$, (C): $Gamma(0.001, 0.001)$, (D): $Gamma(0.0001, 0.0001)$.

7.6.1.3 Results of model selection and assessment of the M5 motorway data

We show the results of model selection of the M5 motorway data using criteria: the AIC$_{rec_1}$, BIC$_{rec_1}$, AIC$_{rec_2}$, BIC$_{rec_2}$, AIC$_{rec_3}$, BIC$_{rec_3}$, DIC$_{rec_2}$, BIC$_{con_1}$ and BIC$_{con_2}$ which were found to work well as introduced in Chapter 5, given chosen priors. We also include the results of model assessment using the posterior predictive distribution plots (Gelman et al., 1996) and the QQ plot of normal pseudo-residuals (Zucchini and MacDonald, 2009).

7.6.1.3.1 Results of model selection criteria of the M5 motorway data

Table (7.6) displays the model-selection results, given four chosen priors.

It can be seen that the criteria: BIC$_{rec_1}$, BIC$_{rec_2}$, BIC$_{rec_3}$ and BIC$_{con_1}$ have a fixed behaviour in selecting the model, suggesting not affected by the assumed prior, and select the model with fewer parameters, $K = 3$, as the best model for the M5 highway. The same behaviour is observed for the AIC$_{rec_1}$ and AIC$_{rec_3}$, but with larger size models, where the former chooses a model with $K = 4$ states, and the latter favors the most complicated model, $K = 5$. In contrast, it can be seen that the AIC$_{rec_2}$, DIC$_{rec_2}$, BIC$_{con_2}$ have variable performance, indicating their effect by the prior. The AIC$_{rec_2}$ selects a large model with $K = 5$, given the more diffuse priors: $Gamma(0.001, 0.001)$ and $Gamma(0.0001, 0.0001)$, whereas it tends to choose a model with more fewer parameters, $K = 3$, given less diffuse priors: $Gamma(0.01, 0.01)$ and $Gamma(0.1, 0.1)$. The DIC$_{rec_2}$ tends to select the model with $K = 5$, given the priors: $Gamma(0.01, 0.01)$, $Gamma(0.001, 0.001)$ and $Gamma(0.0001, 0.0001)$, and then chooses the model $K = 4$, given a less diffuse prior, $Gamma(0.1, 0.1)$. The BIC$_{con_2}$ chooses a complicated model with $K = 4$ states given more diffuse priors: $Gamma(0.001, 0.001)$ and $Gamma(0.0001, 0.0001)$ and then tends to choose a model with more fewer parameters, $K = 3$, given less diffuse priors: $Gamma(0.01, 0.01)$ and $Gamma(0.1, 0.1)$. The tendency for these three criteria, to select the models with smaller size when less diffuse priors are assumed, is agreement with the results of model estimation and convergence diagnostics discussed earlier, which have suggested that the large size models are more sensitive when using more diffuse priors and they improve whenever less diffuse priors are used. Accordingly, the model selection results with a less diffuse prior can be more reliable.

In practice, more adequate fits to the data can be likely obtained from more complex models. However, this case is not often favoured in applications as the large number of parameters can result in high variance in parameter estimates and overfitting. As a result, we prefer the more parsimonious model, $K = 3$, selected by the most criteria: the BIC$_{rec_1}$, AIC$_{rec_2}$, BIC$_{rec_2}$, BIC$_{rec_3}$, DIC$_{con_2}$, BIC$_{con_1}$ and BIC$_{con_2}$.

Gamma$(\alpha = 0.1, \beta = 0.1)$															
K	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$	AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	$p_{\text{DIC}_{rec_2}}$	DIC$_{rec_2}$	$\overline{D_{con}(\mathbf{z},\lambda)}$	$D_{con}(\hat{\mathbf{z}},\hat{\lambda})$	BIC$_{con_1}$	BIC$_{con_2}$
2	439.096	435.093	434.823	449.096	458.852	445.093	454.849	444.823	454.580	4.272	443.369	380.355	381.377	400.111	401.133
3	389.202	383.138	381.324	411.202	**432.666**	**405.138**	**426.601**	403.324	**424.788**	7.878	397.081	314.674	317.109	**358.137**	**360.573**
4	370.643	372.995	354.888	**408.643**	445.717	410.995	448.068	392.888	429.962	15.754	**386.398**	288.876	285.525	363.949	360.599
5	358.749	356.136	331.027	416.749	473.335	414.136	470.722	**389.027**	445.613	27.721	386.471	267.425	280.099	382.011	394.685
Gamma$(\alpha = 0.01, \beta = 0.01)$															
K	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$	AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	$p_{\text{DIC}_{rec_2}}$	DIC$_{rec_2}$	$\overline{D_{con}(\mathbf{z},\lambda)}$	$D_{con}(\hat{\mathbf{z}},\hat{\lambda})$	BIC$_{con_1}$	BIC$_{con_2}$
2	438.997	435.111	434.794	448.997	458.753	445.111	454.867	444.794	454.550	4.203	443.200	380.608	382.366	400.364	402.122
3	389.520	383.122	381.715	411.520	**432.983**	**405.122**	**426.585**	403.715	**425.179**	7.804	397.324	314.542	321.848	**358.006**	**365.312**
4	370.902	373.199	353.970	**408.902**	445.976	411.199	448.273	391.970	429.044	16.932	387.834	287.955	290.669	363.029	365.743
5	358.058	352.730	331.785	416.058	472.644	410.730	467.316	**389.785**	446.371	26.273	**384.332**	261.282	288.246	375.868	402.832
Gamma$(\alpha = 0.001, \beta = 0.001)$															
K	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$	AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	$p_{\text{DIC}_{rec_2}}$	DIC$_{rec_2}$	$\overline{D_{con}(\mathbf{z},\lambda)}$	$D_{con}(\hat{\mathbf{z}},\hat{\lambda})$	BIC$_{con_1}$	BIC$_{con_2}$
2	439.061	435.083	435.003	449.061	458.818	445.083	454.839	445.003	454.759	4.058	443.120	380.329	377.977	400.086	397.733
3	389.459	383.382	381.211	411.459	**432.923**	405.382	**426.846**	403.211	**424.674**	8.248	397.708	314.928	328.532	**358.392**	371.995
4	370.176	373.252	353.804	**408.176**	445.249	411.252	448.325	391.804	428.878	16.371	386.548	289.526	295.034	364.600	**370.108**
5	353.873	341.950	328.621	411.873	468.459	**399.950**	456.536	**386.621**	443.207	25.252	**379.125**	266.586	293.184	381.172	407.770
Gamma$(\alpha = 0.0001, \beta = 0.0001)$															
K	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$	AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	$p_{\text{DIC}_{rec_2}}$	DIC$_{rec_2}$	$\overline{D_{con}(\mathbf{z},\lambda)}$	$D_{con}(\hat{\mathbf{z}},\hat{\lambda})$	BIC$_{con_1}$	BIC$_{con_2}$
2	439.219	435.097	434.985	449.219	458.976	445.097	454.854	444.985	454.742	4.234	443.453	380.480	373.374	400.236	393.130
3	389.483	383.046	382.237	411.483	**432.947**	405.046	**426.509**	404.237	**425.701**	7.245	396.729	314.875	317.155	**358.338**	360.618
4	370.139	372.250	353.287	**408.139**	445.213	410.250	447.323	391.287	428.361	16.852	386.992	287.960	281.026	363.033	**356.099**
5	351.230	339.850	330.711	409.230	465.816	**397.850**	454.436	**388.711**	445.298	20.518	**371.748**	258.854	283.843	373.440	398.429

Table 7.6: Results of the model selection criteria for several PHMMs with $K = 2, ..., 5$ fitted to the M5 highway data under four prior choices: *Gamma*$(0.1, 0.1)$, *Gamma*$(0.01, 0.01)$, *Gamma*$(0.001, 0.001)$ and *Gamma*$(0.0001, 0.0001)$.

7.6.1.3.2 Model assessment of the M5 motorway data

Figures (7.14 - 7.17) display 95% predictive intervals constructed from the posterior predictive distributions simulated from a PHMM with K= 2, 3, 4 and 5 respectively, against the observed crash counts, where the centers of these intervals account for the means of the posterior predictive distributions. Figure (7.14) shows the inadequacy of the model with $K = 2$ due to its poor predictive performance, since it provides predictive means that are far from the observed crash counts. Also, the QQ-plot of pseudo-residuals shows a clear deviation from normality in Figure (7.18), which supports that the model with $K = 2$ does not give a good fit to the data. In contrast, our preferred model, $K = 3$, and the models with, $K = 4$ and $K = 5$ show a better performance, and are thus more suitable as shown in Figures (7.15-7.17) and Figure (7.18), respectively.

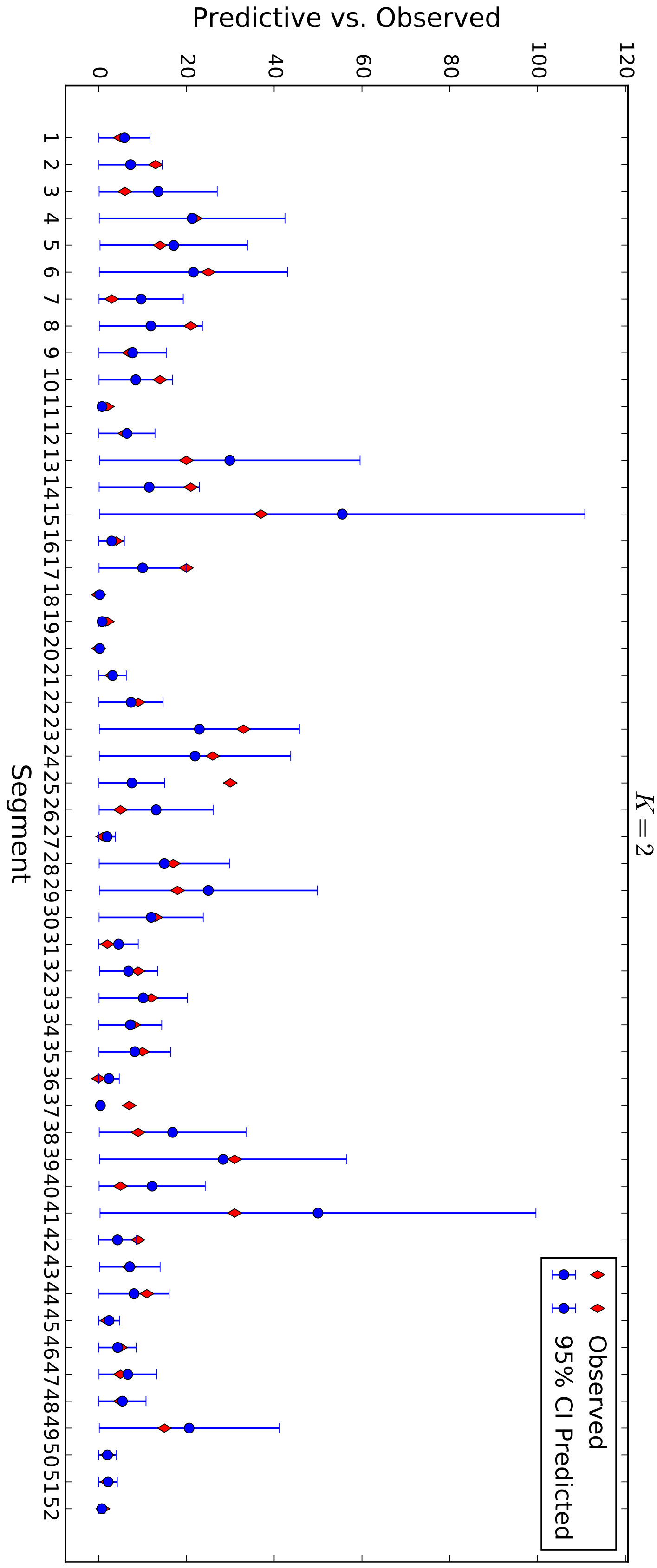


Figure 7.14: The posterior predictive distribution with 95% predictive intervals simulated from a 2-state PHMM vs observed crash counts (M5 data).

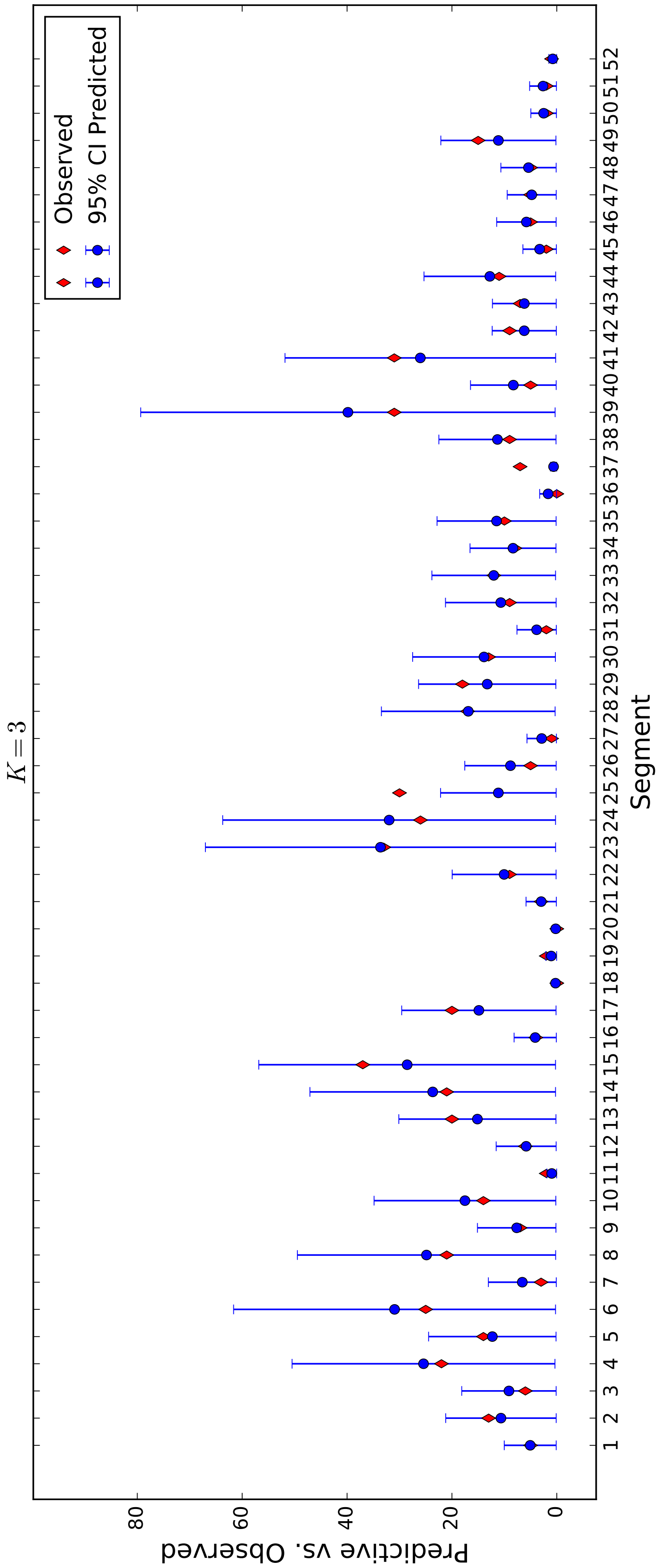


Figure 7.15: The posterior predictive distribution with 95% predictive intervals simulated from a 3-state PHMM vs observed crash counts (M5 data).

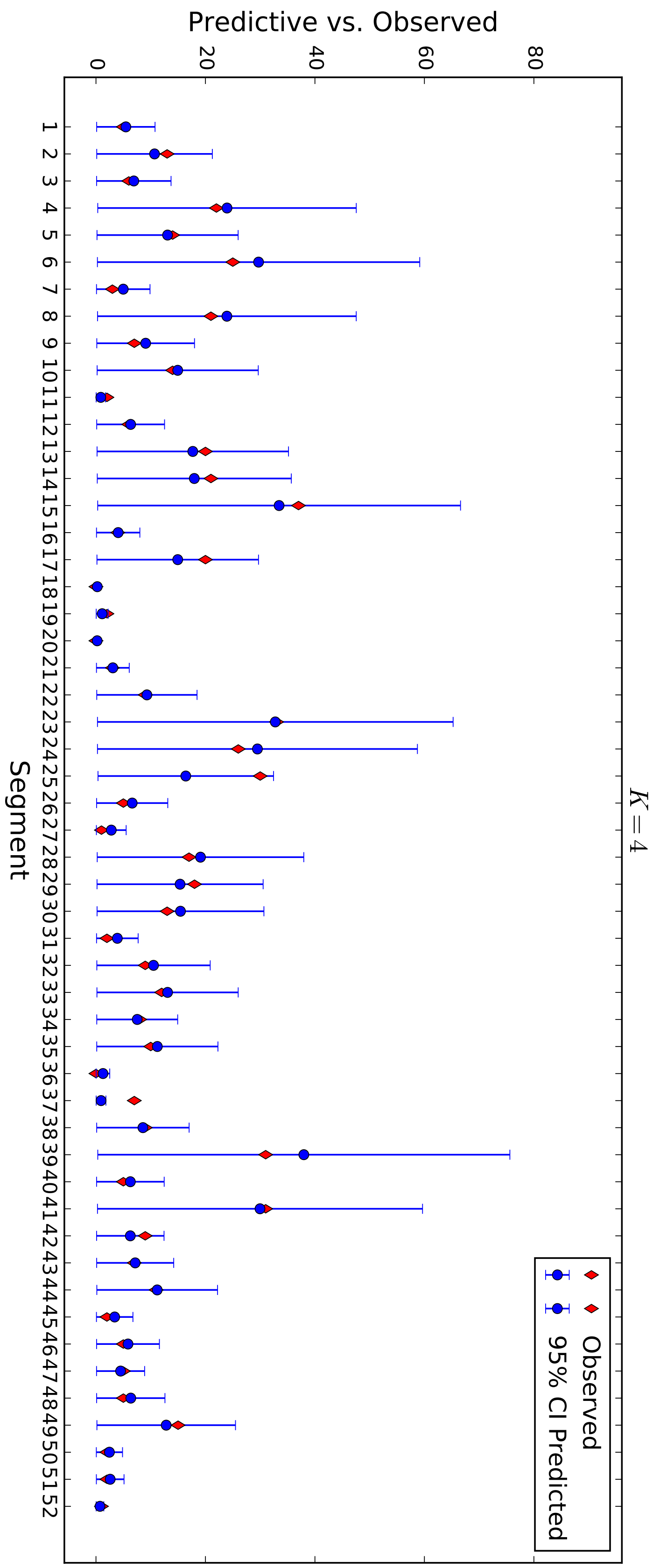


Figure 7.16: The posterior predictive distribution with 95% predictive intervals simulated from a 4-state PHMM vs observed crash counts (M5 data).

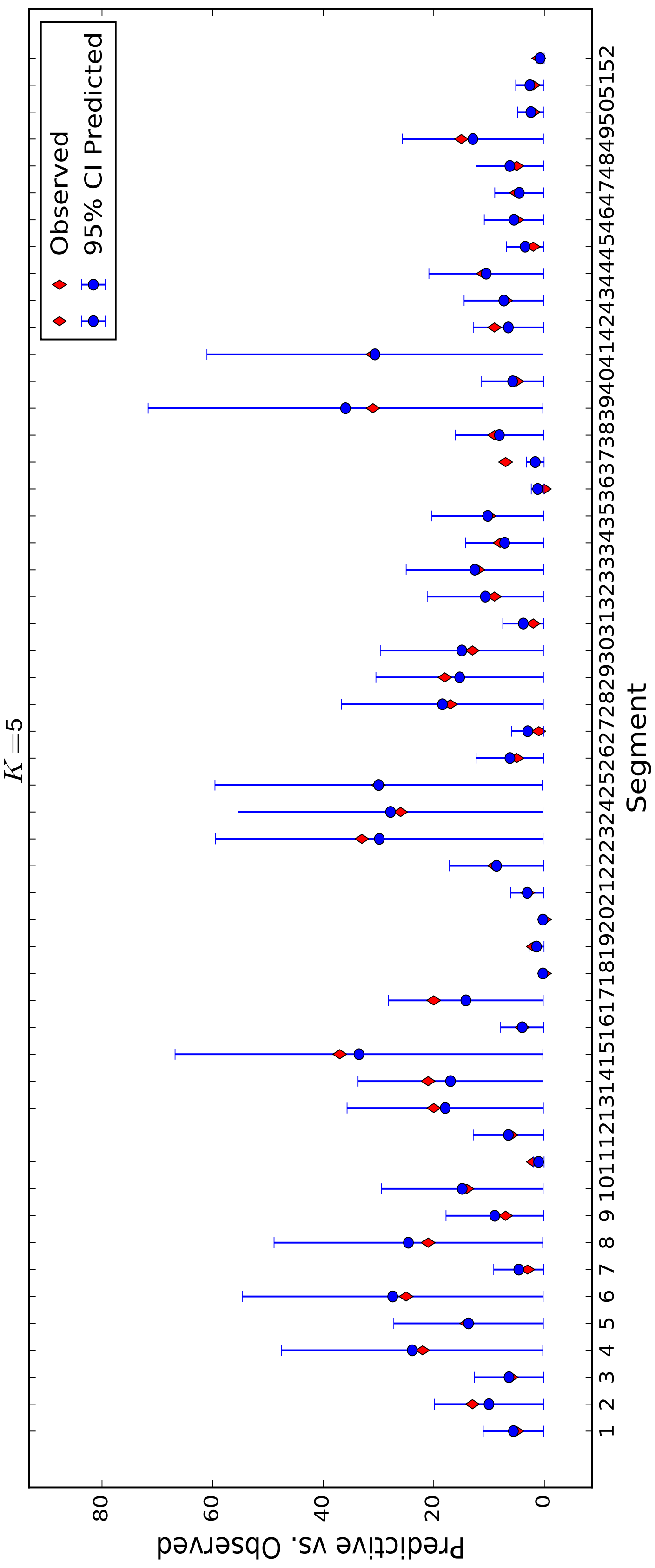


Figure 7.17: The posterior predictive distribution with 95% predictive intervals simulated from a 5-state PHMM vs observed crash counts (M5 data).

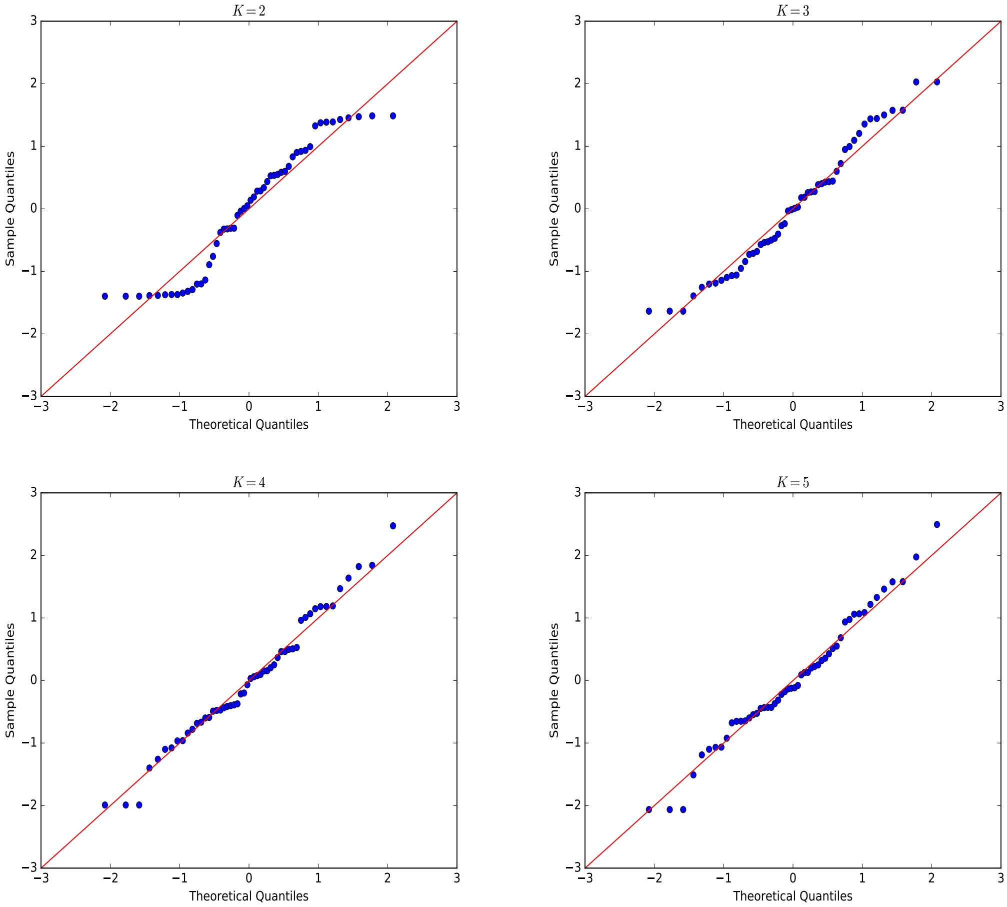


Figure 7.18: Normal QQ-plots of ordinary pseudo-residuals for crashes data under 2, 3, 4 and 5-state PHMM (M5 data).

7.6.2 The M6 motorway data

Work for the M6 motorway ($T = 90$) was carried out as for the M5. We display all results of the convergence, estimation, selection and assessment of each model is being fitted to these data, given less diffuse prior, $Gamma(0.1, 0.1)$, obtained from MCMC sampling with 104000 iterations (thin=100, burn-in=4000).

7.6.2.1 Convergence results of the M6 motorway data

Figure (7.19) shows the ACF plots of the posterior samples of the state-specific crash rate parameters, λ_j, of a PHMM with $K = 2,\ldots,6$ fitted to the M6 motorway data. All ACF plots suggest that there is no autocorrelation in the samples. Moreover, the Gelman-Rubin statistics, $\hat{R}$, and the Geweke statistics, $\hat{G}$, provided in the Table (7.7) suggest there is no lack in convergence.

7.6.2.2 Results of the estimation of the crash rate parameter of the M6 motorway data

Table (7.7) shows the estimation results of state-specific crash rate parameters, λ_j. Figure (7.20) shows the box-plots of the posterior distributions of state-specific crash rate parameters sampled from a PHMM with $K = 2,\ldots,6$ fitted to the data.

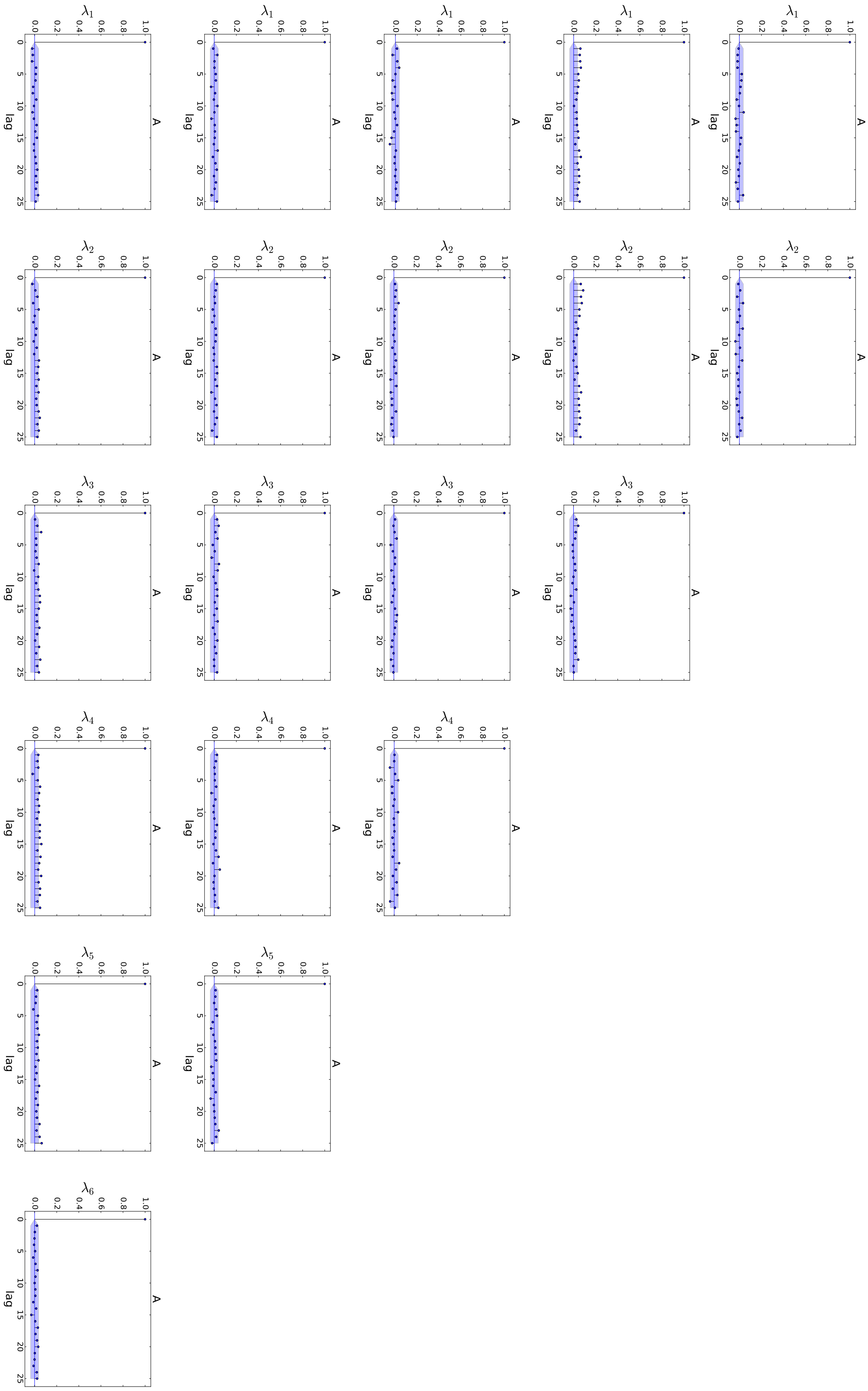


Figure 7.19: The ACF plots of the crash rate parameter, λ, of a PHMM with $K = 2, \ldots, 6$, given a less diffuse prior, *Gamma*$(0.1, 0.1)$, fitted to the M6 data.

Model	$\hat{\lambda}_k$	Mean	95% CI			$\hat{R}$	Geweke
			2.5%	Median	97.5%		
2	$\hat{\lambda}_1$	0.577	0.528	0.576	0.626	0.999	-1.258
	$\hat{\lambda}_2$	2.579	2.371	2.580	2.786	0.999	-0.860
3	$\hat{\lambda}_1$	0.354	0.261	0.342	0.549	1.037	-1.591
	$\hat{\lambda}_2$	1.252	1.038	1.218	1.848	1.041	-1.644
	$\hat{\lambda}_3$	5.275	3.902	5.172	6.932	1.006	-1.593
4	$\hat{\lambda}_1$	0.253	0.195	0.2529	0.314	1.000	-0.069
	$\hat{\lambda}_2$	0.809	0.723	0.809	0.900	1.000	0.961
	$\hat{\lambda}_3$	1.972	1.773	1.971	2.190	0.999	0.116
	$\hat{\lambda}_4$	6.471	5.571	6.450	7.522	1.001	1.304
5	$\hat{\lambda}_1$	0.150	0.078	0.142	0.270	1.004	0.527
	$\hat{\lambda}_2$	0.486	0.347	0.464	0.805	1.008	1.068
	$\hat{\lambda}_3$	0.940	0.772	0.881	1.788	1.010	1.710
	$\hat{\lambda}_4$	2.068	1.782	1.993	2.931	1.005	1.204
	$\hat{\lambda}_5$	6.536	5.604	6.501	7.697	1.000	-0.688
6	$\hat{\lambda}_1$	0.143	0.078	0.138	0.223	1.001	1.302
	$\hat{\lambda}_2$	0.443	0.272	0.444	0.603	1.009	1.757
	$\hat{\lambda}_4$	0.791	0.476	0.820	0.970	1.019	1.973
	$\hat{\lambda}_4$	1.350	0.809	1.406	2.032	1.030	-1.922
	$\hat{\lambda}_5$	2.473	1.842	2.158	5.582	1.016	1.507
	$\hat{\lambda}_6$	6.746	5.663	6.678	8.182	1.005	1.258

Table 7.7: Results of the estimation and convergence of the crash rate parameter, λ, of a PHMM with $K = 2,...,6$, given a less diffuse prior, $Gamma(0.1, 0.1)$, fitted to the M6 motorway data. The third column provides the ergodic posterior means of the rate parameter. The fourth column provides the corresponding 95% CI. The last two columns include the values of the Gelman-Rubin statistic, $\hat{R}$, and the Geweke statistic, $\hat{G}$, respectively.

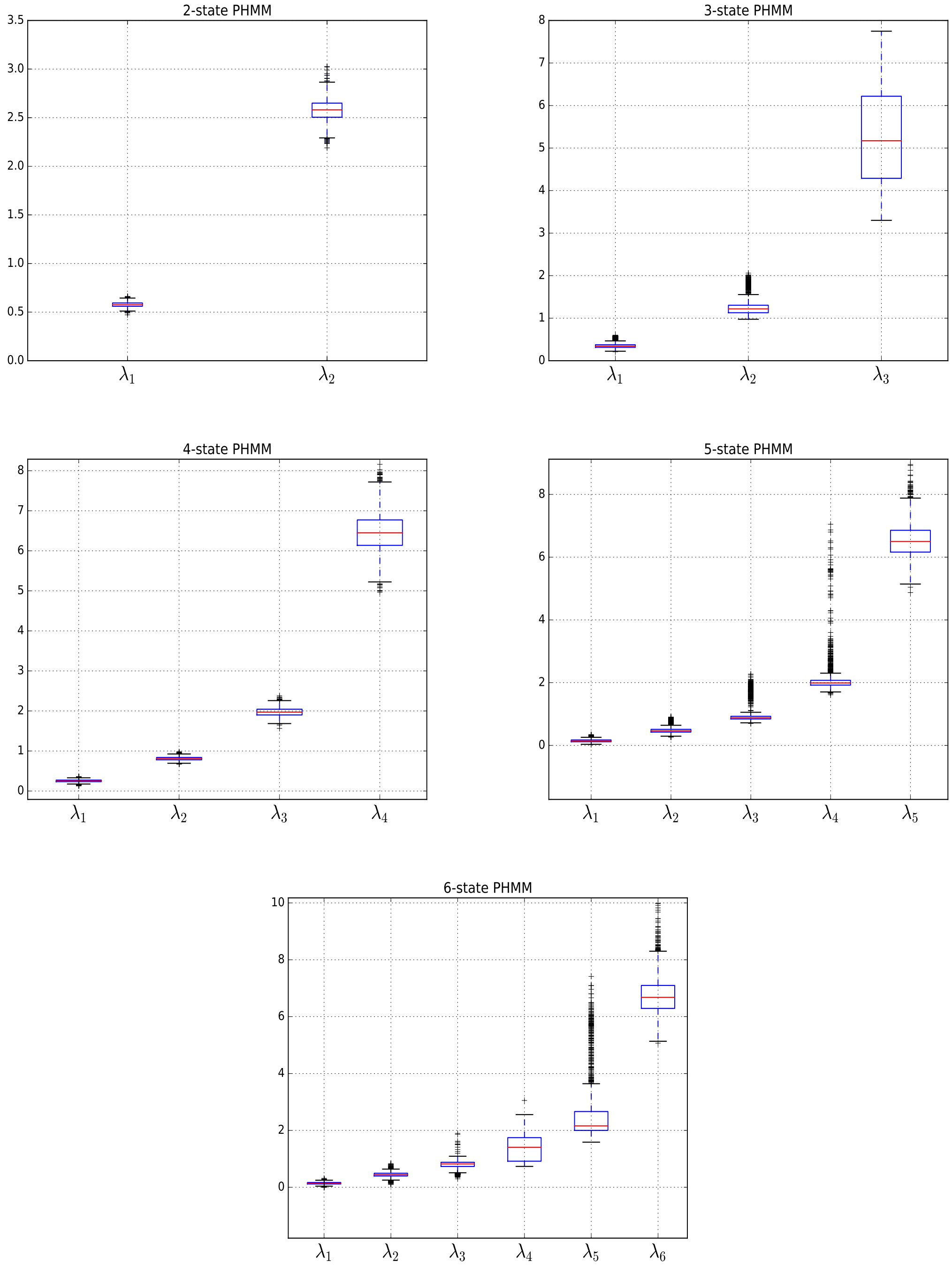


Figure 7.20: Box-plots of the posterior distributions of state-specific crash rate parameters of a PHMM with $K = 2, \ldots, 6$, fitted to the M6 motorway data.

7.6.2.3 Results of model selection and assessment of the M6 motorway data

This section displays the results of model selection criteria for PHMMs with states $K = 2, \ldots, 6$, fitted to the M6 motorway data as shown in Table (7.8), given a less diffuse prior. We can see that all criteria select the model with $K = 4$.

$Gamma(\alpha = 0.1, \beta = 0.1)$														
K	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$	AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	$p_{\text{DIC}_{rec_2}}$	DIC$_{rec_2}$	$\overline{D_{con}(\mathbf{z}, \lambda)}$	$D_{con}(\hat{\mathbf{z}}, \hat{\lambda})$	BIC$_{con_1}$	BIC$_{con_2}$
2	939.032	934.768	934.580	949.032	961.531	944.768	957.267	944.580	957.079	4.452	943.484	833.552	824.825	856.051	847.324
3	796.170	792.354	784.424	818.170	845.668	814.354	841.852	806.424	833.922	11.746	807.917	648.261	626.030	697.759	675.528
4	688.735	676.586	677.224	**726.735**	**774.232**	**714.586**	**762.083**	**715.224**	**762.720**	12.148	**700.884**	505.040	481.339	**590.537**	**566.835**
5	687.756	671.410	668.524	745.756	818.250	729.410	801.904	726.524	799.018	16.346	704.102	501.518	461.913	616.104	592.408
6	685.782	665.859	664.015	767.782	870.274	747.859	850.351	746.015	848.507	19.923	705.705	494.167	450.419	678.659	634.911

Table 7.8: Results of the model selection criteria for several PHMMs with $K = 2, \ldots, 6$ fitted to the M6 highway data under four prior choices: $Gamma(0.1, 0.1)$, $Gamma(0.01, 0.01)$, $Gamma(0.001, 0.001)$ and $Gamma(0.0001, 0.0001)$.

Figures (7.21 - 7.25) display 95% predictive intervals constructed from the posterior predictive distributions simulated from a PHMM with $K = 2, ..., 6$, respectively. Figure (7.21) provides posterior predictive means, centers of predictive intervals, that are far from the observed crash counts. This indicates the inadequacy of the model with $K = 2$ compared with other competing models. The QQ-plot of pseudo-residuals also appears a clear deviation from normality in Figure (7.26). On the other hand, it can be seen that the models with $K = 3$, $K = 4$, $K = 5$ and $K = 6$ appear to have adequate predictive performance in Figures (7.22-7.25), respectively. According to QQ-plots in Figure (7.26), it can be seen that the selected model, $K = 4$, has a good predictive performance.

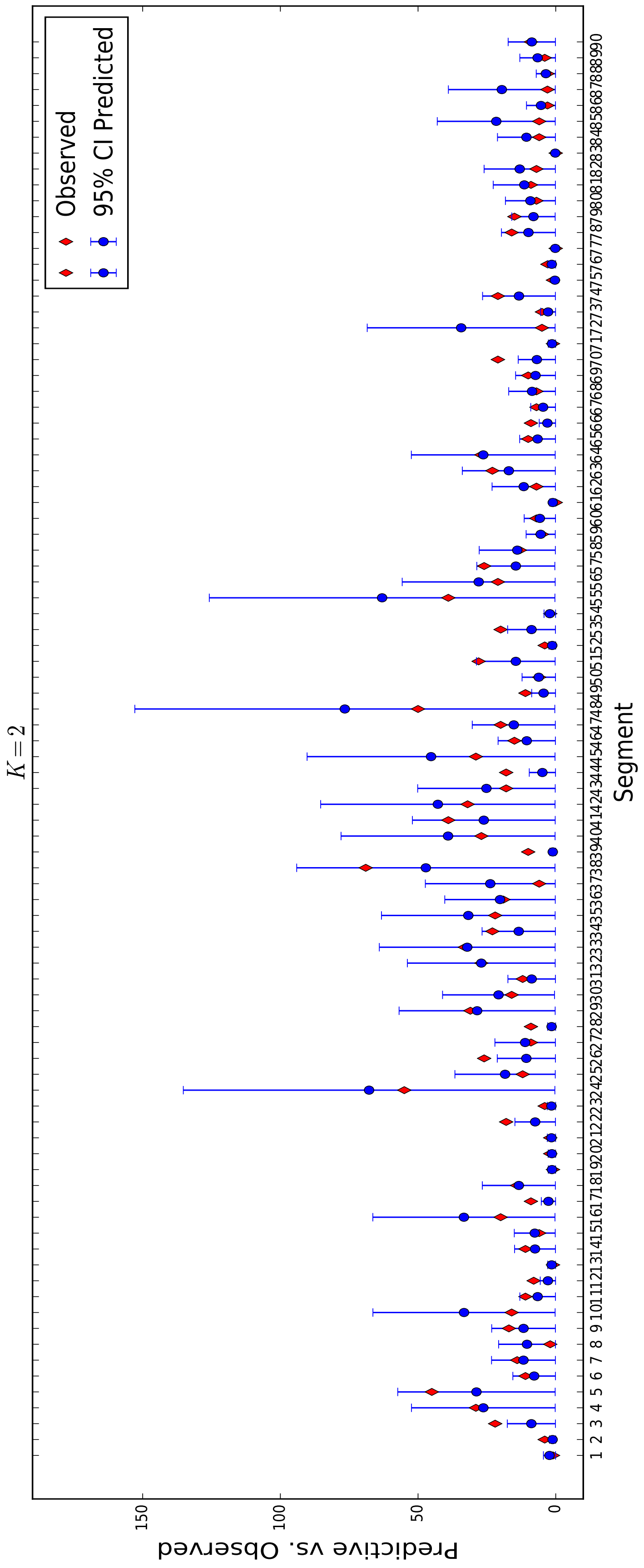


Figure 7.21: The posterior predictive distribution with 95% predictive intervals simulated from a 2-state PHMM vs observed crash counts (M6 data).

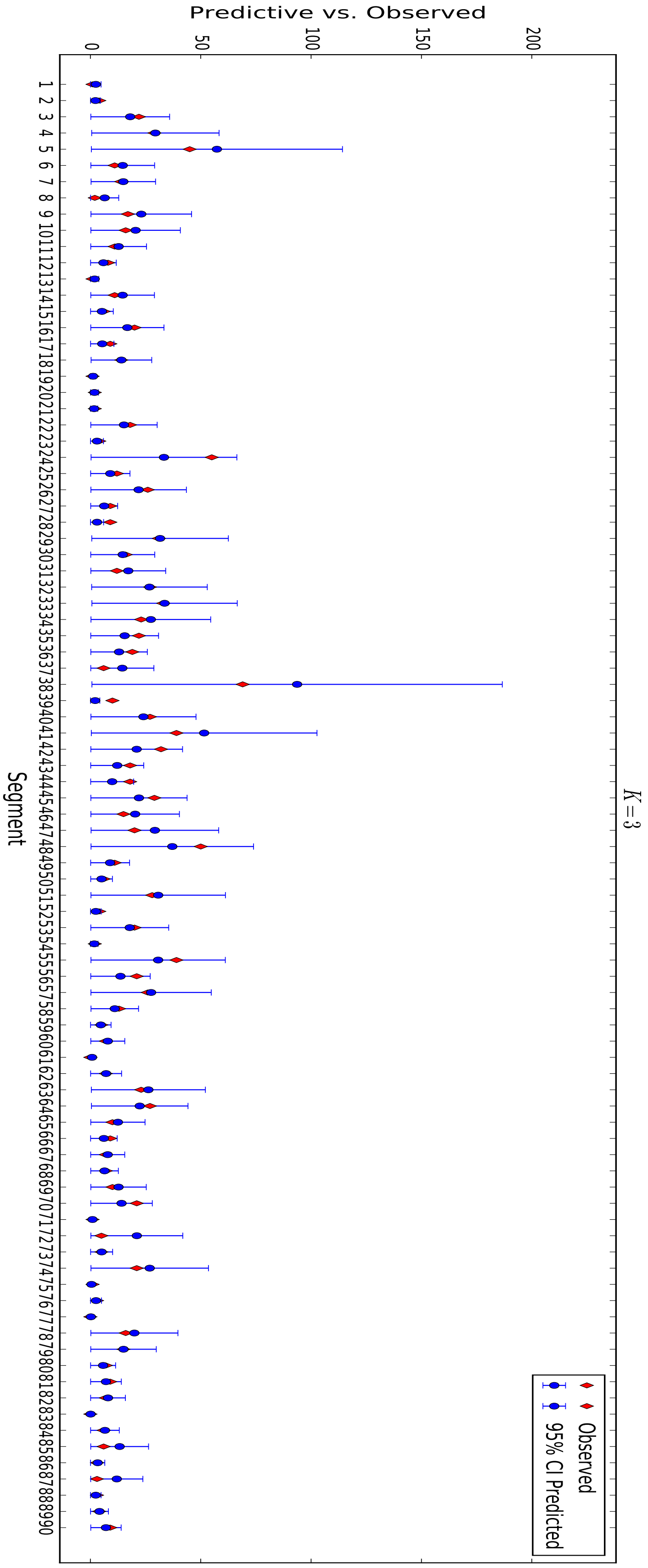


Figure 7.22: The posterior predictive distribution with 95% predictive intervals simulated from a 3-state PHMM vs observed crash counts (M6 data).

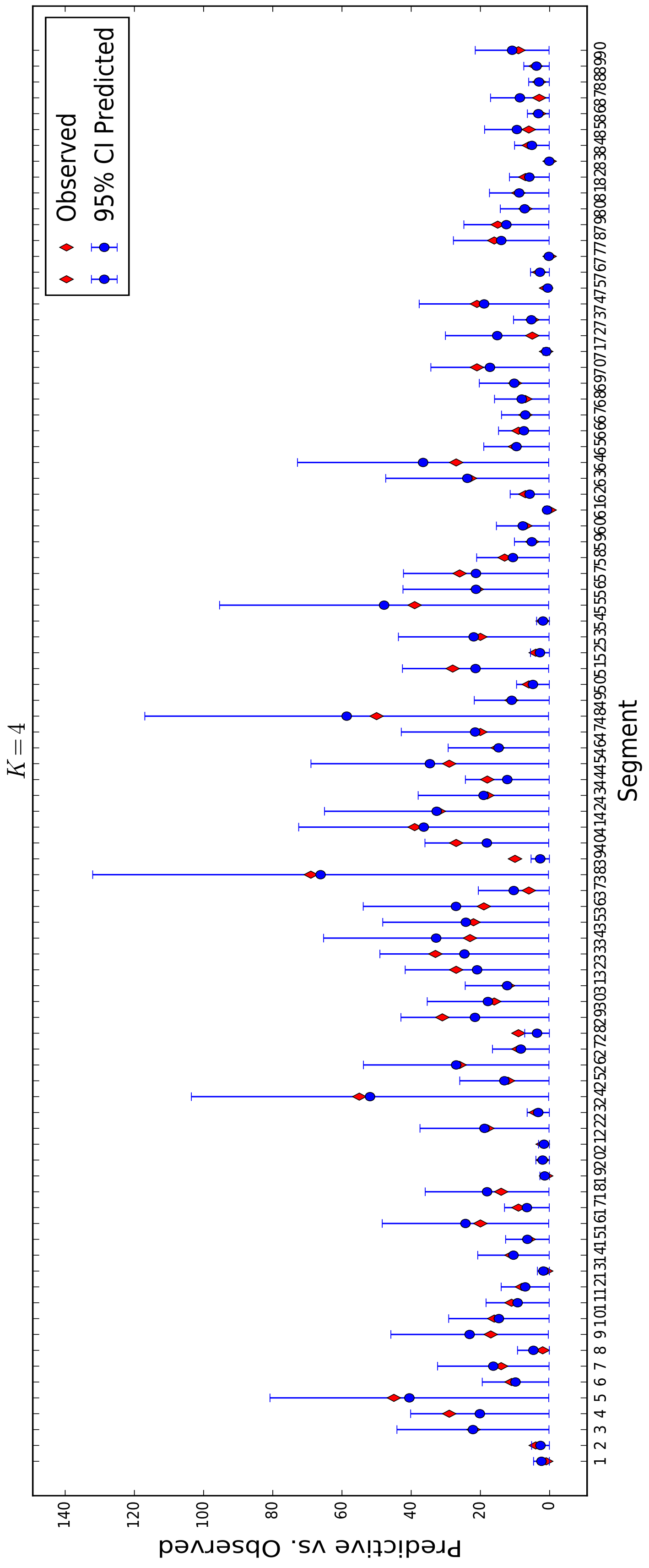


Figure 7.23: The posterior predictive distribution with 95% predictive intervals simulated from a 4-state PHMM vs observed crash counts (M6 data).

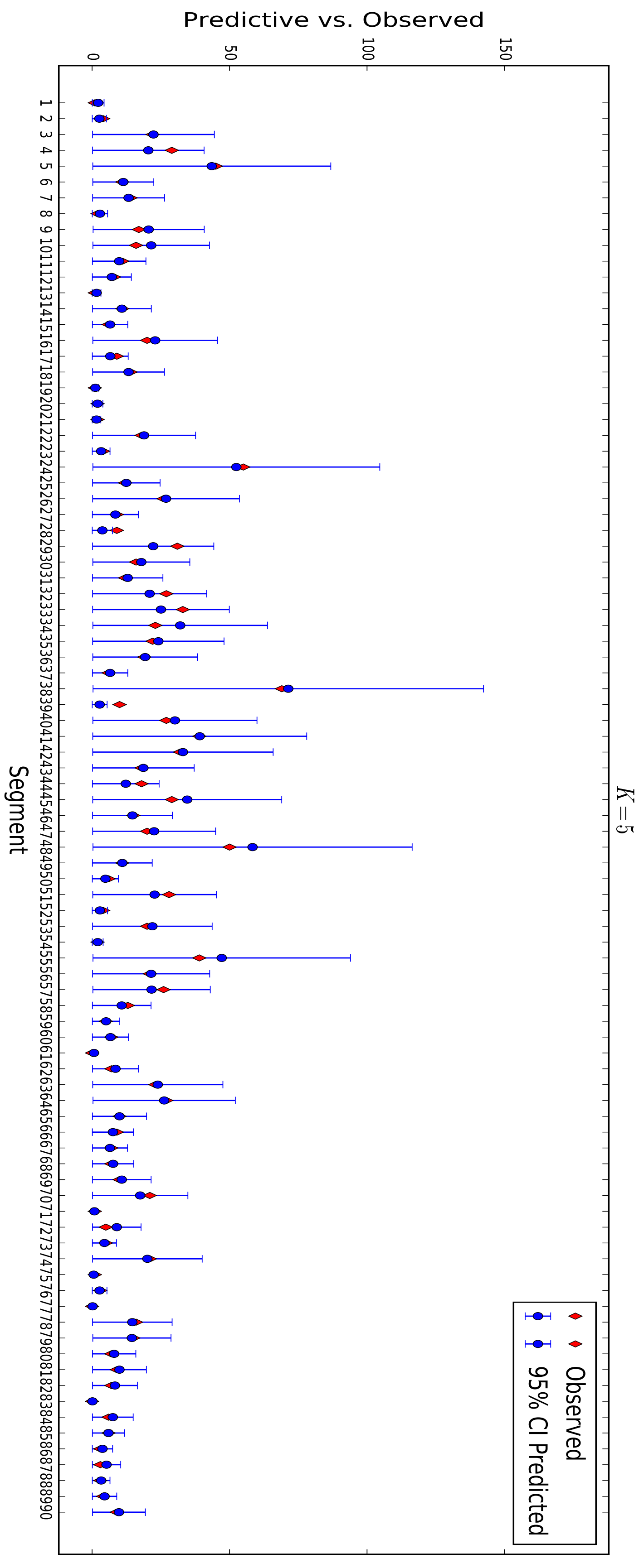


Figure 7.24: The posterior predictive distribution with 95% predictive intervals simulated from a 5-state PHMM vs observed crash counts (M6 data).

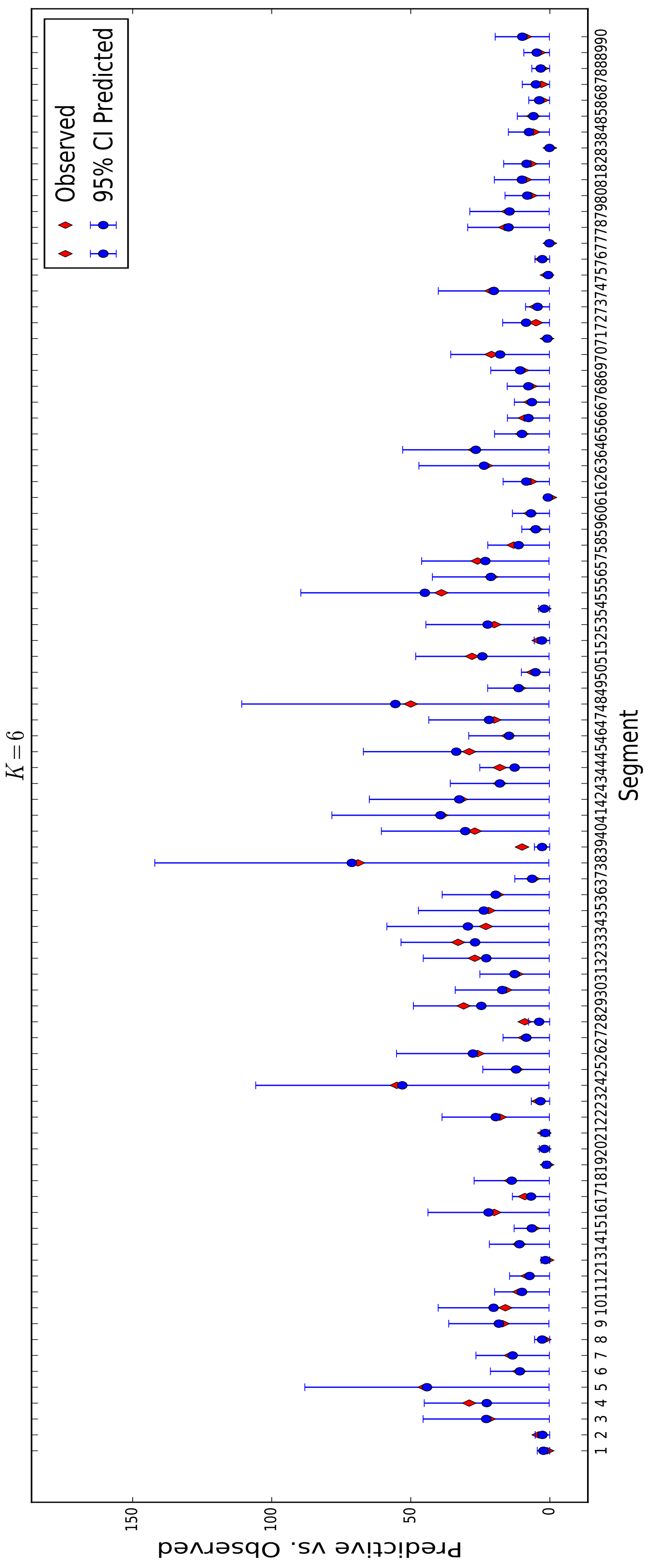


Figure 7.25: The posterior predictive distribution with 95% predictive intervals simulated from a 6-state PHMM vs observed crash counts (M6 data).

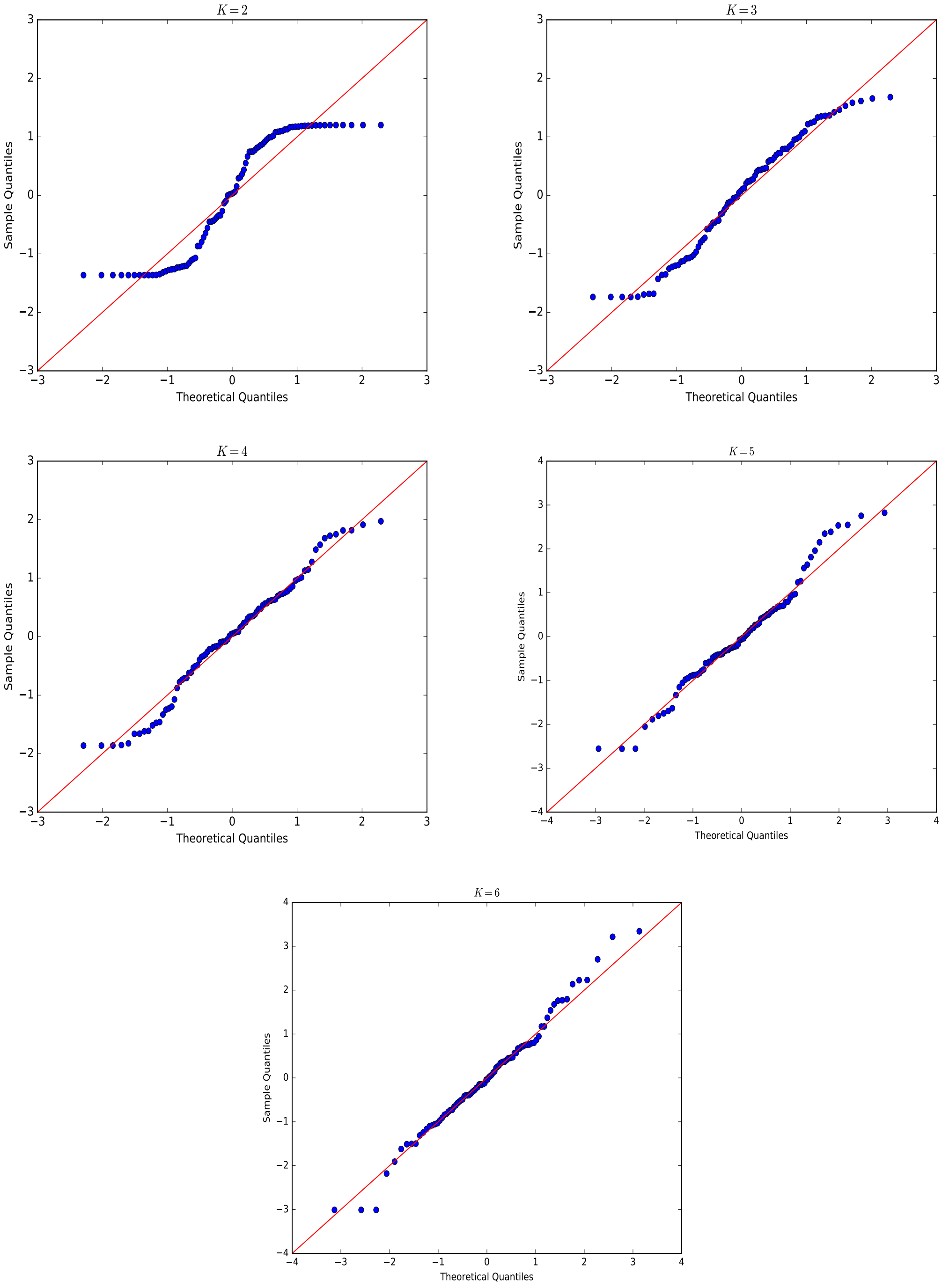


Figure 7.26: Normal QQ-plots of ordinary pseudo-residuals for crashes data under a PHMM with $K = 2, \ldots, 6$ (M6 data).

7.6.3 The M42 motorway data

As for the first two motorways, we display results of the convergence, estimation, selection and assessment of each model fitted to the M42 motorway data ($T = 21$).

7.6.3.1 Convergence results of the M42 motorway data

Figure (7.27) shows the ACF plots of the thinned posterior samples of the crash rate parameter, λ, which gives no concerns about autocorrelation. Also, the Gelman-Rubin and the Geweke statistics suggest convergence may have been achieved, as they provide values less than 1.1 and within the interval $[-2, 2]$, respectively, as shown in Table (7.9).

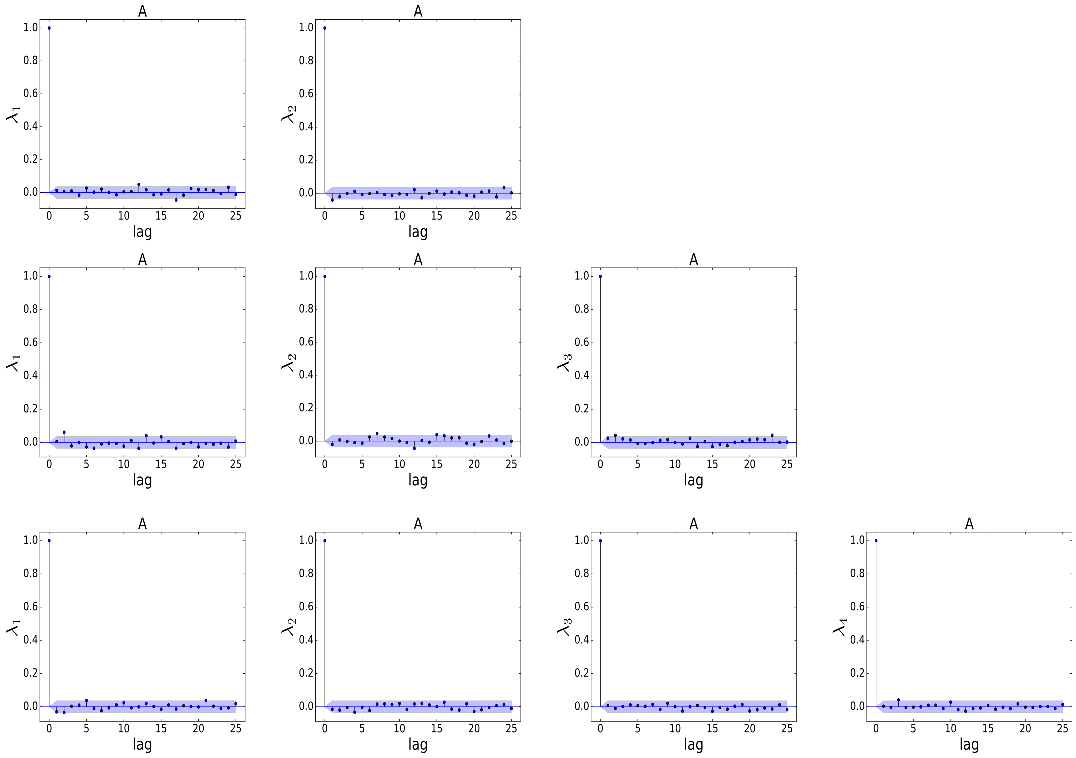


Figure 7.27: The ACF plots of the crash rate parameter, λ, of a PHMM with $K = 2, 3$ and 4, given a less diffuse prior, $Gamma(0.1, 0.1)$, fitted to the M42 motorway data.

7.6.3.2 Results of the estimation of the crash rate parameter of the M42 motorway data

Table (7.9) and Figure (7.28) show the estimation results of the crash rate parameters, λ_j, for each model fitted to the crash data of M42 motorway.

Model	$\hat{\lambda}_k$	Mean	95% CI			$\hat{R}$	$\hat{G}$
			2.5%	Median	97.5%		
2	$\hat{\lambda}_1$	0.762	0.603	0.759	0.937	0.999	1.580
	$\hat{\lambda}_2$	1.792	1.306	1.772	2.413	0.999	0.367
3	$\hat{\lambda}_1$	0.585	0.032	0.632	0.858	0.999	1.053
	$\hat{\lambda}_2$	0.934	0.670	0.880	1.550	0.999	-0.660
	$\hat{\lambda}_3$	1.924	1.351	1.849	2.829	0.999	-0.078
4	$\hat{\lambda}_1$	0.447	0.011	0.525	0.800	0.999	-0.865
	$\hat{\lambda}_2$	0.761	0.389	0.760	1.081	0.999	-1.706
	$\hat{\lambda}_3$	1.131	0.724	1.029	2.001	0.999	-1.800
	$\hat{\lambda}_4$	2.169	1.421	1.961	4.223	1.001	-1.221

Table 7.9: Results of the estimation and convergence of the crash rate parameter, λ, of a PHMM with $K = 2,3$ and 4, given a less diffuse prior, $Gamma(0.1, 0.1)$, fitted to the M42 motorway data. The third column provides the ergodic posterior means of the rate parameter. The fourth column provides the corresponding 95% CI. The last two columns include the values of the Gelman-Rubin statistic, $\hat{R}$, and the Geweke statistic, $\hat{G}$, respectively.

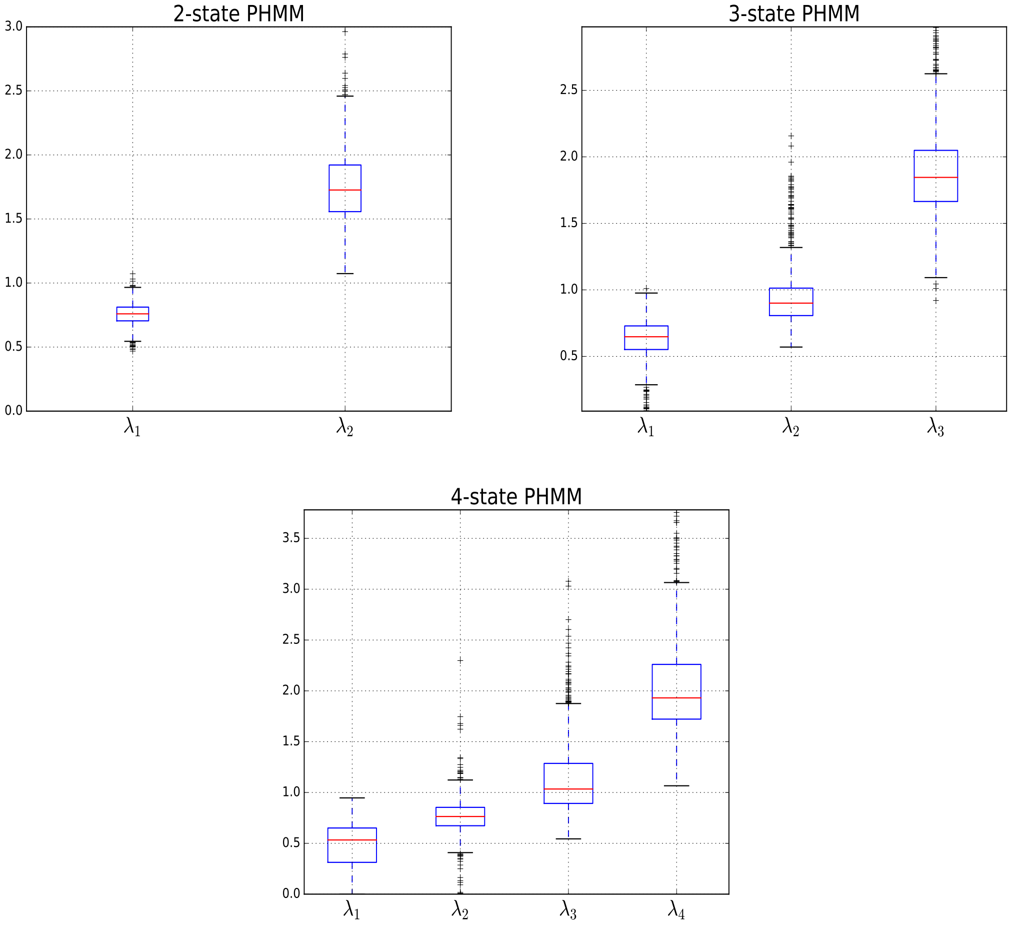


Figure 7.28: Box-plots of the posterior distributions of state-specific crash rate parameters of a PHMM with $K = 2, 3$ and 4, fitted to the M42 motorway data.

7.6.3.3 Results of model selection and assessment of the M42 motorway data

Table (7.10) shows the results of model selection for the M42 motorway data. All criteria suggest that these data can be sufficiently modelled by only two states. The results of posterior predictive distributions shown in Figure (7.29) also suggest that the model with $K = 2$ is adequate. We can see that the all models provide somewhat similar normal pseudo-residuals as shown in Figure (7.30). Given that, the M42 motorway data can be represented by only two states.

$Gamma(\alpha = 0.1, \beta = 0.1)$															
K	$\overline{D_{rec}(\Theta)}$	$D_{rec}(\bar{\Theta})$	$\hat{D}_{rec}(\Theta)$	AIC$_{rec_1}$	BIC$_{rec_1}$	AIC$_{rec_2}$	BIC$_{rec_2}$	AIC$_{rec_3}$	BIC$_{rec_3}$	$p_{\text{DIC}_{rec_2}}$	DIC$_{rec_2}$	$\overline{D_{con}(\mathbf{z}, \lambda)}$	$D_{con}(\hat{\mathbf{z}}, \hat{\lambda})$	BIC$_{con_1}$	BIC$_{con_2}$
2	115.406	112.677	110.442	**125.406**	**130.629**	**122.677**	**127.900**	**120.442**	**125.665**	4.964	**120.370**	101.973	94.737	**117.196**	**109.960**
3	116.168	112.132	107.739	138.168	149.657	134.132	145.621	129.739	141.228	8.429	124.597	99.965	88.003	133.455	121.493
4	116.113	111.871	107.013	154.113	173.959	149.871	169.719	145.013	164.858	9.101	126.214	98.798	86.007	156.644	143.853
$Gamma(\alpha = 0.01, \beta = 0.01)$															

Table 7.10: Results of the model selection criteria for several PHMMs with $K = 2, 3, 4$ fitted to the M42 highway data under four prior choices: $Gamma(0.1, 0.1)$, $Gamma(0.01, 0.01)$, $Gamma(0.001, 0.001)$ and $Gamma(0.0001, 0.0001)$.

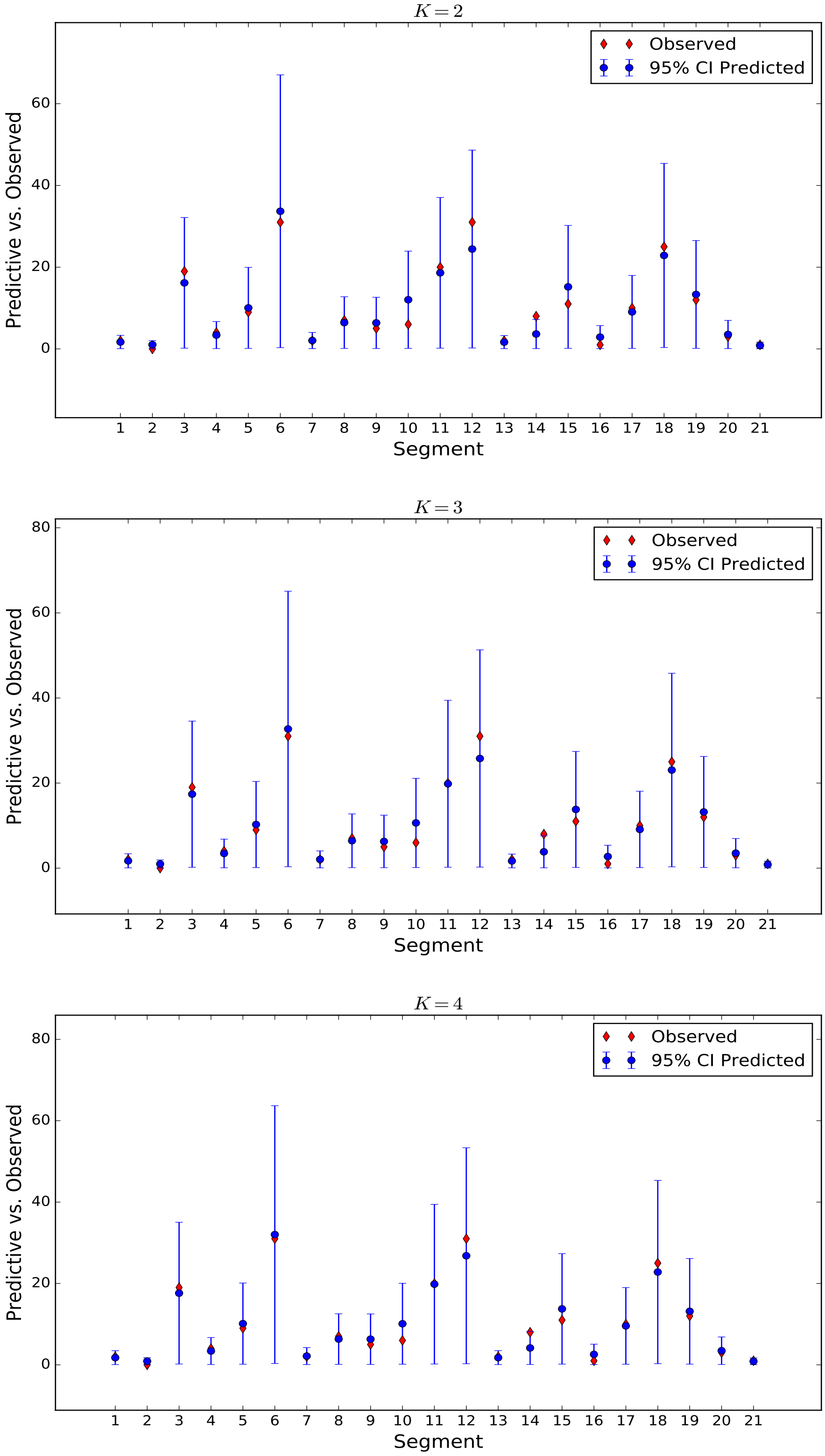


Figure 7.29: The posterior predictive distribution with 95% predictive intervals simulated from a PHMM with states *K*=2, 3 and 4 vs observed crash counts.

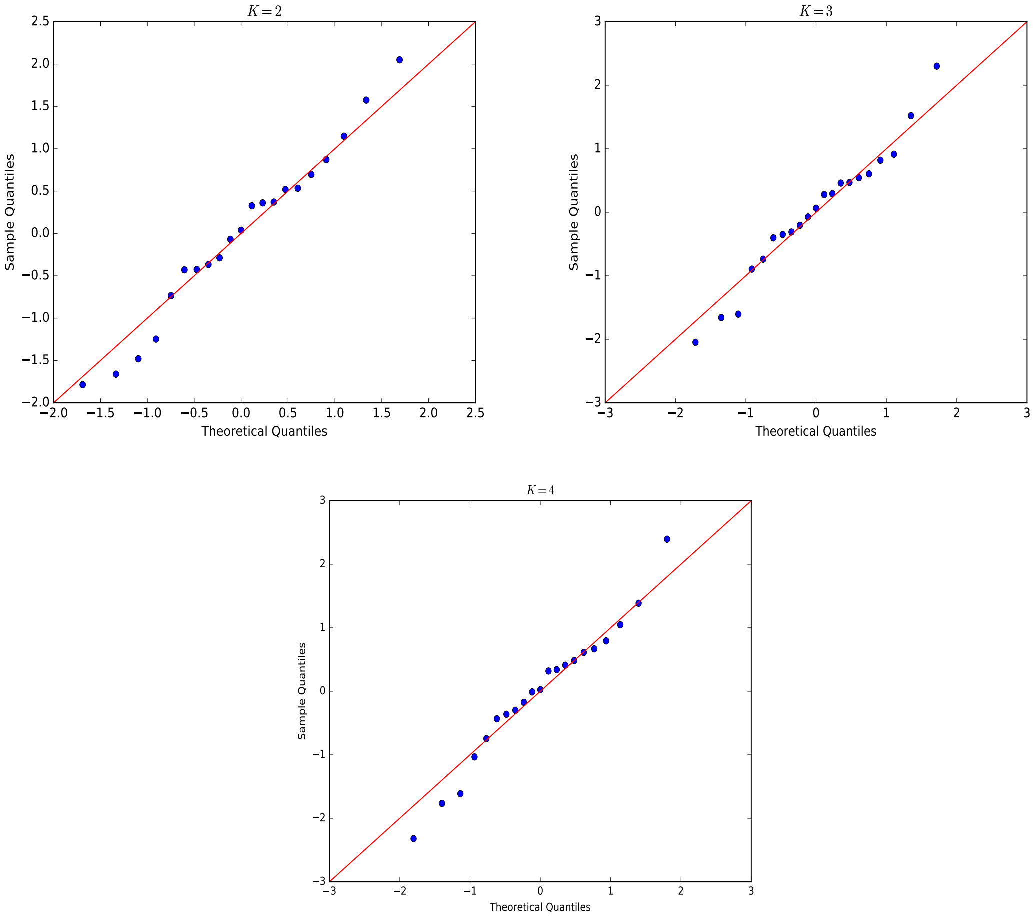


Figure 7.30: Normal QQ-plots of ordinary pseudo-residuals obtained from a PHMM with K=2, 3 and 4 of the M42 motorway data.

7.6.4 Estimation results of selected models

Table (7.11) summarizes the estimates: the initial state probability, $\hat{\pi}$, transition probabilities, $\hat{\mathbf{A}}$, and state-dependent crash parameter, $\hat{\lambda}$, of the selected models for all traffic crashes datasets presented in this Chapter. The results indicate that each highway demonstrates different spatial dependence. Figures (7.31–7.33) represent the posterior classification probabilities for each segment belonging to a given state. In each case, the bottom plot of each figure represents the highest risk state.

In addition, we display the state-specific means of crash rates for each motorway, depicted by the most likely state sequence, as show in Figures (7.34–7.36).

Given the context, it is informative to examine these results on a map. We plotted and mapped these probabilities using the Arc Geographic Information System (ArcGIS, 2014) as displayed in Figures (7.37-7.39).

Motorway	Model	Crash rate $\hat{\lambda}_j$	Initial state $\hat{\pi}_j$	Transition probabilities $\hat{\mathbf{A}}$
M5	3	$\begin{bmatrix}0.336\\1.065\\3.057\end{bmatrix}$	$\begin{bmatrix}0.319\\0.399\\0.282\end{bmatrix}$	$\begin{bmatrix}0.134 & 0.538 & 0.327\\0.322 & 0.406 & 0.271\\0.530 & 0.176 & 0.293\end{bmatrix}$
M6	4	$\begin{bmatrix}0.253\\0.809\\1.972\\6.471\end{bmatrix}$	$\begin{bmatrix}0.194\\0.324\\0.271\\0.211\end{bmatrix}$	$\begin{bmatrix}0.386 & 0.389 & 0.083 & 0.141\\0.216 & 0.291 & 0.252 & 0.239\\0.059 & 0.353 & 0.295 & 0.291\\0.160 & 0.226 & 0.436 & 0.176\end{bmatrix}$
M42	2	$\begin{bmatrix}0.762\\1.792\end{bmatrix}$	$\begin{bmatrix}0.655\\0.345\end{bmatrix}$	$\begin{bmatrix}0.688 & 0.312\\0.761 & 0.239\end{bmatrix}$

Table 7.11: Results of the selected models for the highways crash data: M5, M6, and M42.

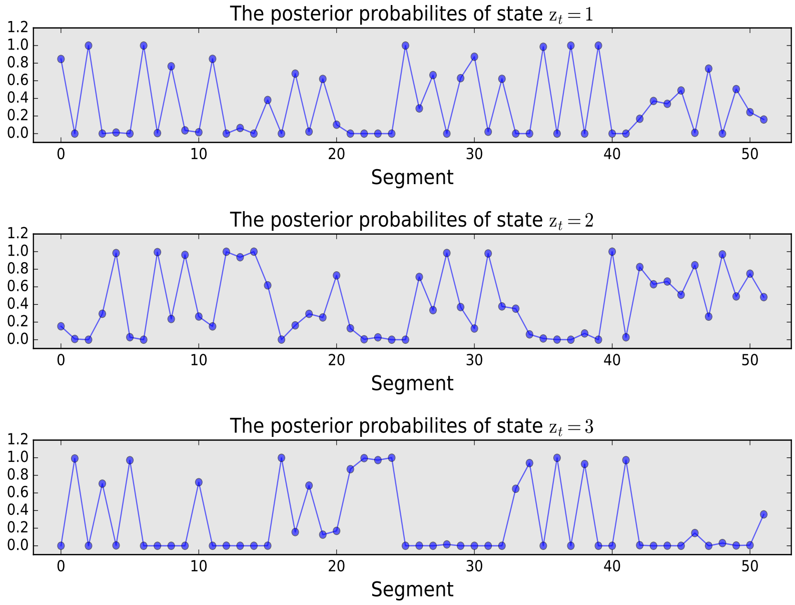


Figure 7.31: Trace-plots of the posterior probabilities of hidden states for the 3-state Poisson HMM for the M5 motorway.

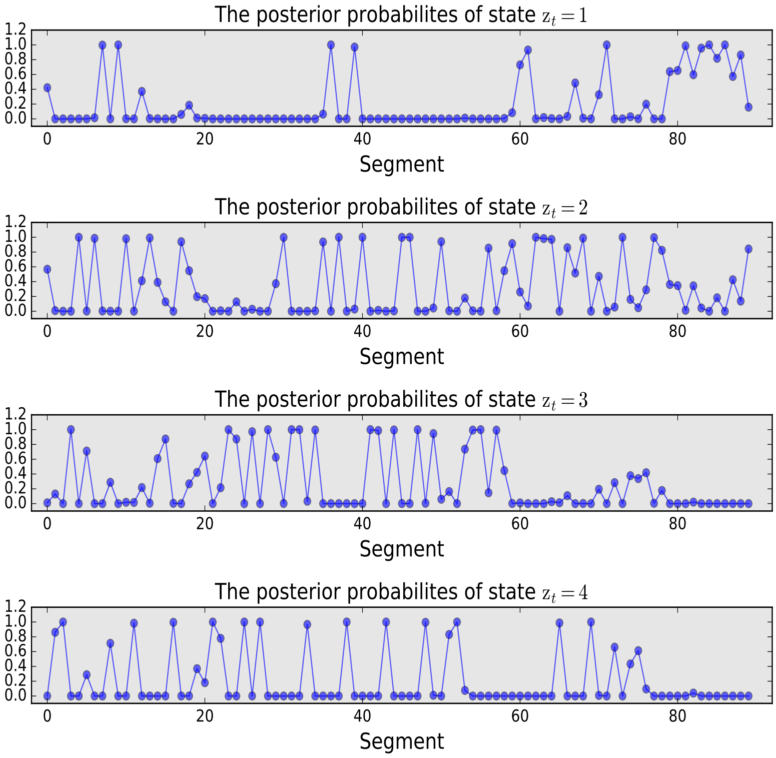


Figure 7.32: Trace-plots of the posterior probabilities of hidden states for the 4-state Poisson HMM for the M6 motorway.

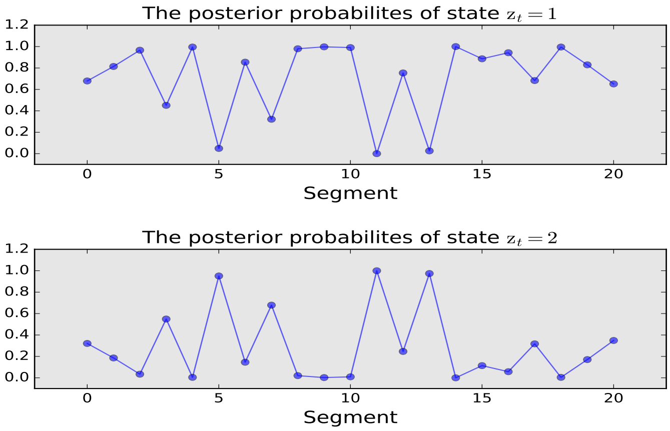
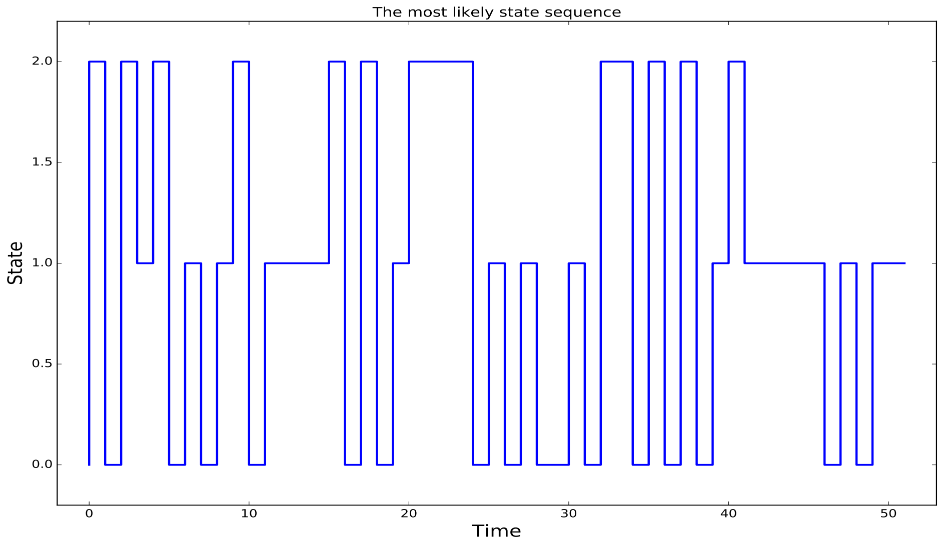


Figure 7.33: Trace-plots of the posterior probabilities of hidden states for the 2-state Poisson HMM for the M42 motorway.



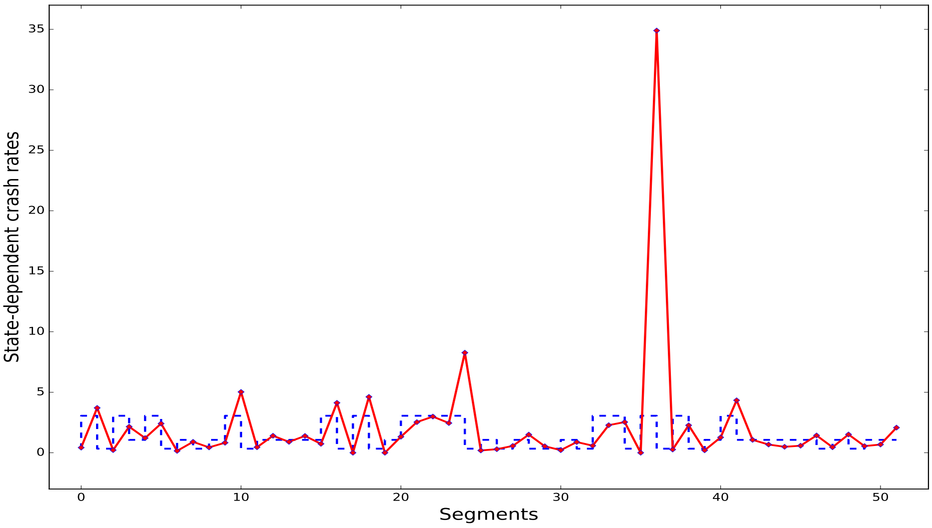
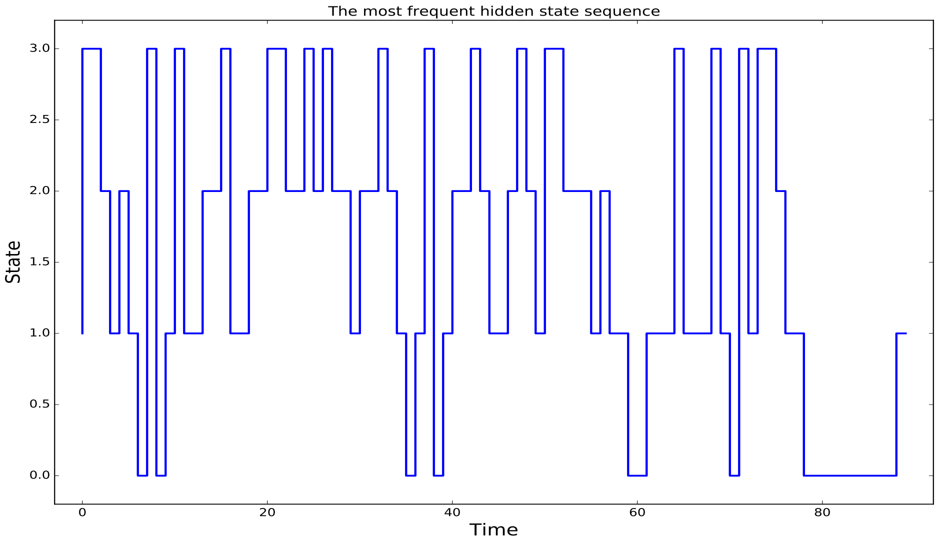


Figure 7.34: Trace-plot of the most frequent hidden state sequence (Top). State-dependent crash rates of each segment (dashed line) of the M5 motorway, depicted by the most likely state sequence (Bottom).



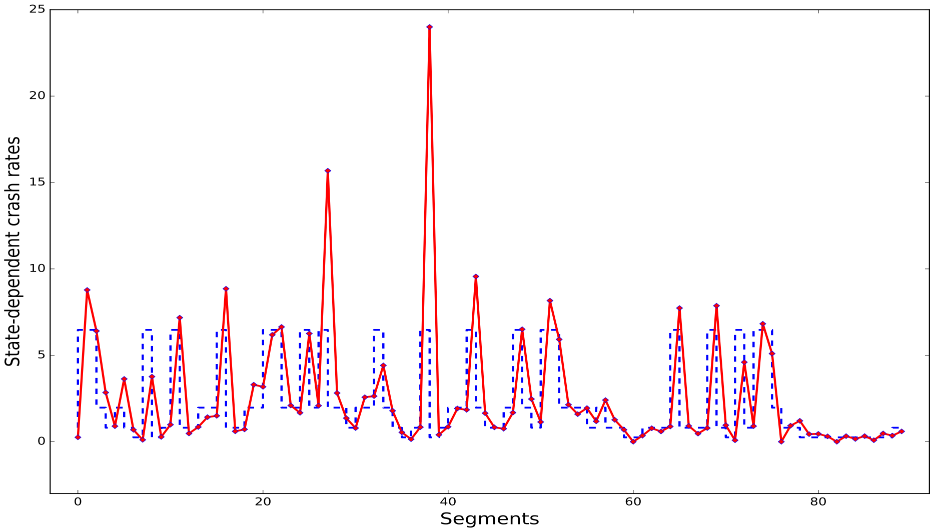


Figure 7.35: Trace-plot of the most frequent hidden state sequence (Top). State-dependent crash rates of each segment (dashed line) of the M6 motorway, depicted by the most likely state sequence (Bottom).

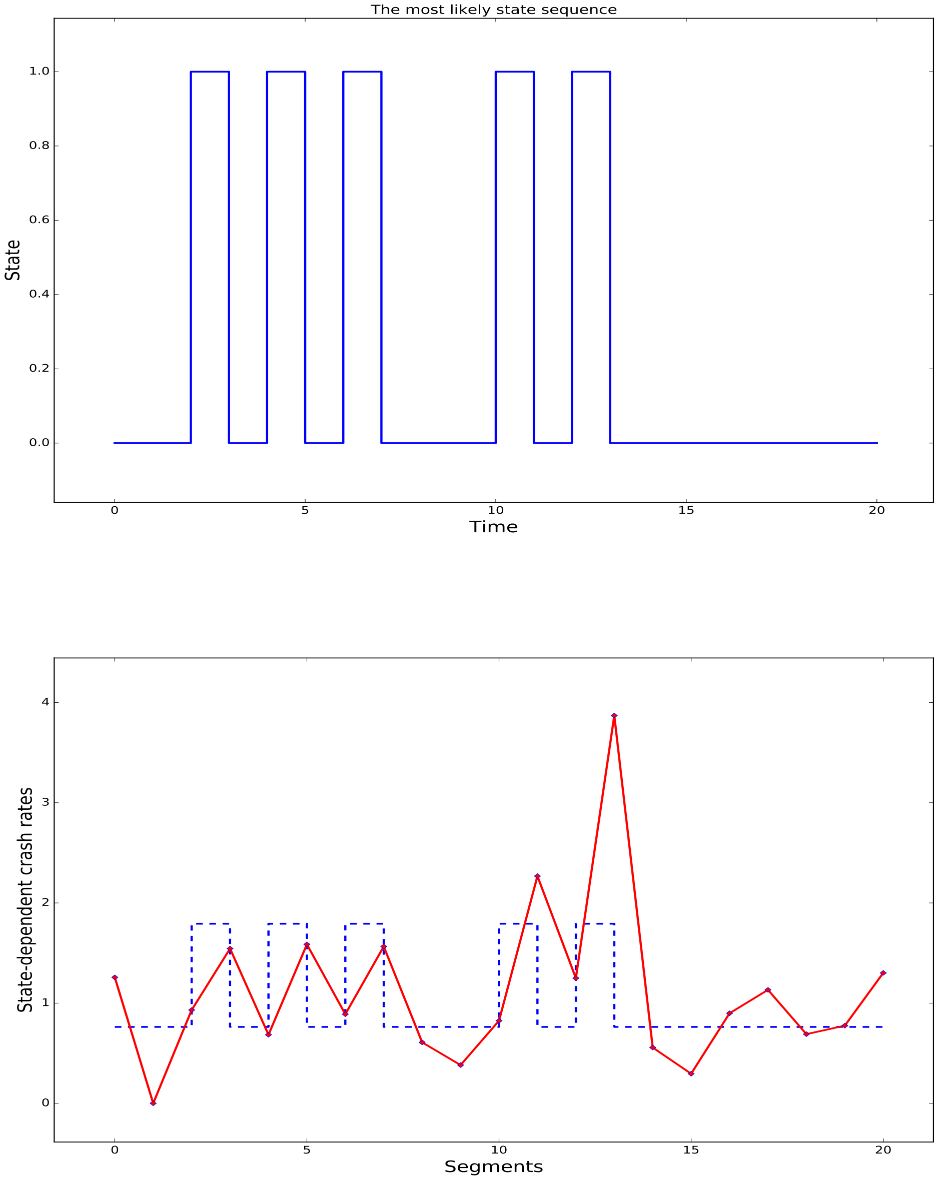


Figure 7.36: Trace-plot of the most frequent hidden state sequence (Top). State-dependent crash rates of each segment (dashed line) of the M42 motorway, depicted by the most likely state sequence (Bottom).

7.6.4.1 The selected model for M5 highway data

Based on the estimation results and the mapped classification probabilities of the model identified for M5 motorway data, we classify the estimated state-specific crash rate means, $\hat{\lambda}_{Z_t}$, of the selected model into 3 levels. We refer to state 1 ($\hat{\lambda}_1 = 0.336$) as a low-risk state, state 2 ($\hat{\lambda}_2 = 1.065$) as a moderate-risk state and state 3 ($\hat{\lambda}_3 = 3.057$) as a high- risk state.

The moderate-risk case ($a_{22} = 0.406$) represents the general traffic safety case of the M5 motorway. According to the results in Figures (7.31) and (7.37), the segments: 2, 6, 17, 22, 23, 24, 25, 35, 37, 39 and 42 which represent the intersections on the M5 motorway have the highest risk.

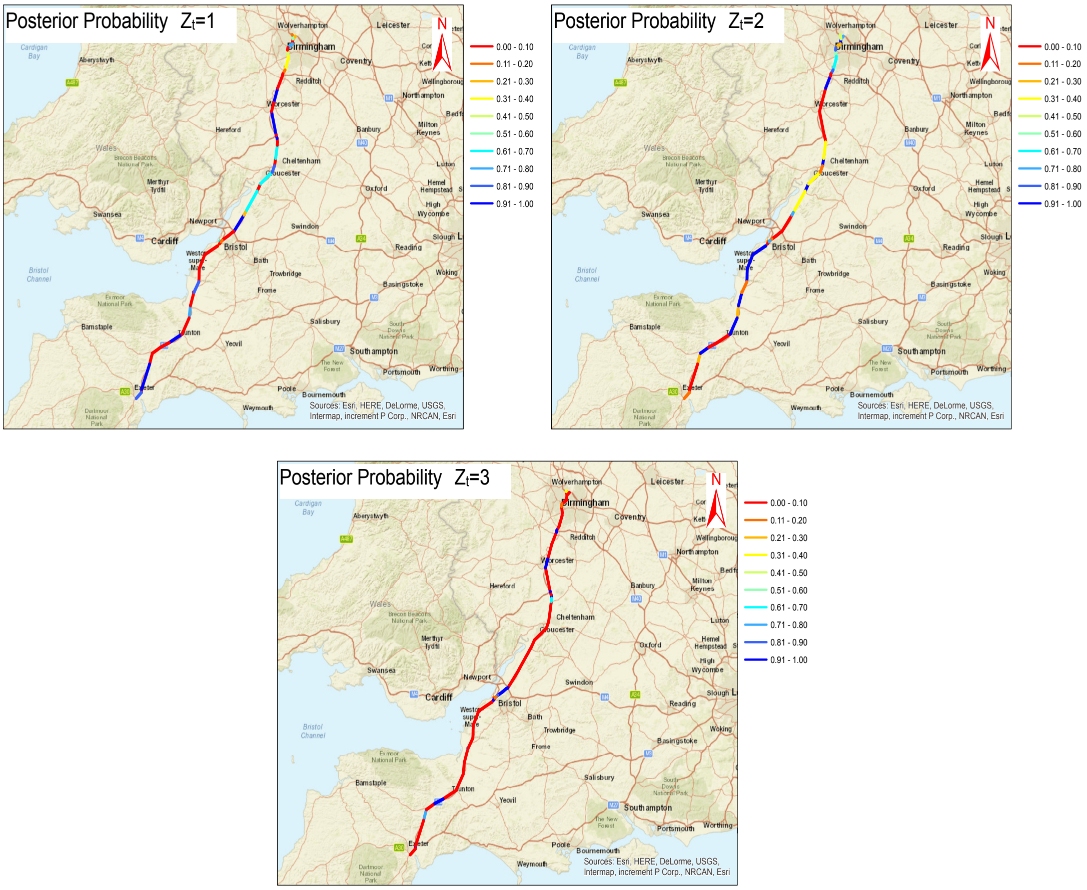


Figure 7.37: The spatial results mapped at segment level for the M5 motorway.

7.6.4.2 The selected model for M6 highway data

Based on the results of the selected model for the M6 highway data, we can characterize the traffic safety case of this highway as: state 1 ($\hat{\lambda}_1 = 0.253$) as a very low-risk case, state 2 as a low-risk case ($\hat{\lambda}_2 = 0.809$), state 3 as a high-risk ($\hat{\lambda}_3 = 1.972$) case and state 4 as a very high-risk case ($\hat{\lambda}_4 = 6.471$). As shown from the posterior transition probabilities in Table (7.11), the general behavior of the M6 highway seems similar to the M5 motorway in terms of the traffic safety, where the low-risk segments are more common. However, it reveals a different spatial dependency in the data as shown from its posterior transition probabilities.

According to the posterior allocation probabilities shown in Figure (7.32) and mapped in Figure (7.38), the segments: 2, 3, 12, 17, 22, 23, 26, 28, 34, 39, 44, 49, 52, 53, 66 and 70 form the highest risk segments which are mostly the intersections on this highway.

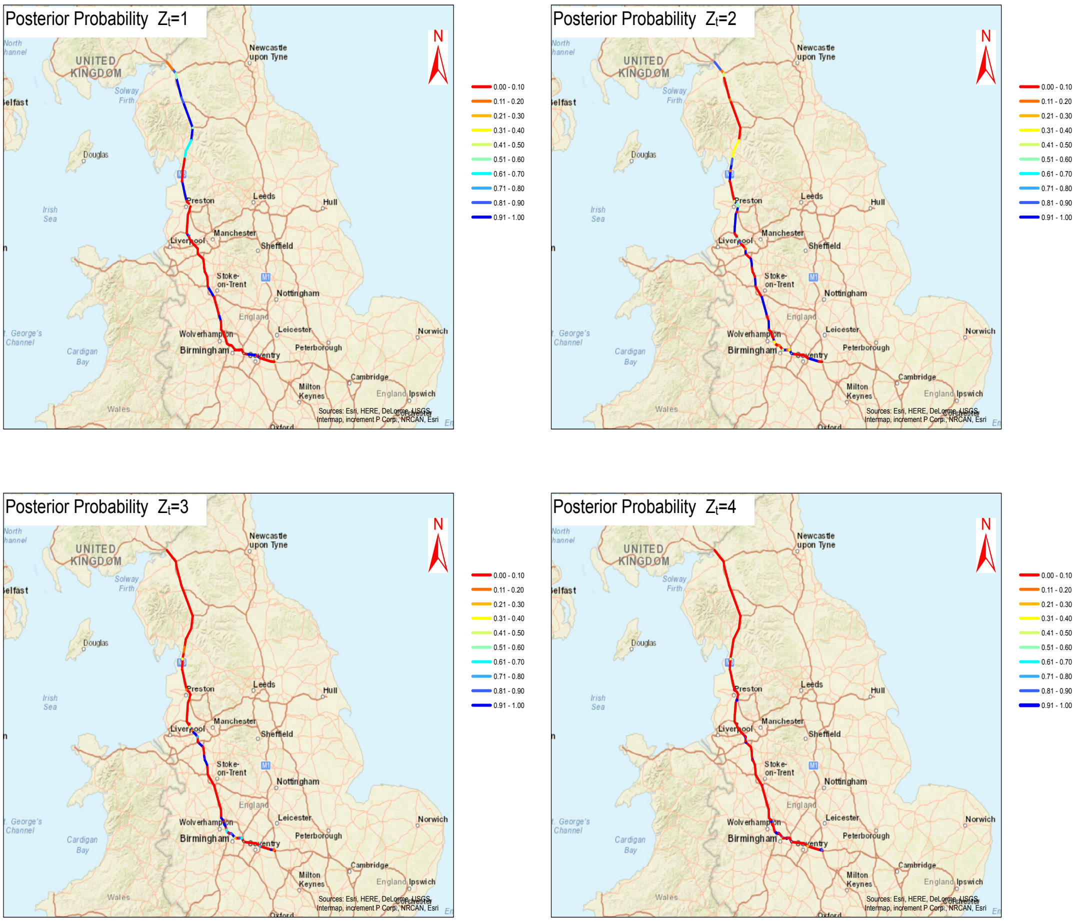


Figure 7.38: The spatial results mapped at segment level for the M6 motorway.

7.6.4.3 The selected model for M42 highway data

We selected a two-state model for the M42 highway. State 1 can be characterized as a low-risk ($\hat{\lambda}_1 = 0.762$), whereas state 2 as a high-risk ($\hat{\lambda}_2 = 1.792$) state. Based on the posterior allocation probabilities displayed in Figure (7.33), the segments of M42 highway that have high risk associated with them ($z_t = 2$) are only 6, 12 and 14, as also mapped in Figure (7.39).

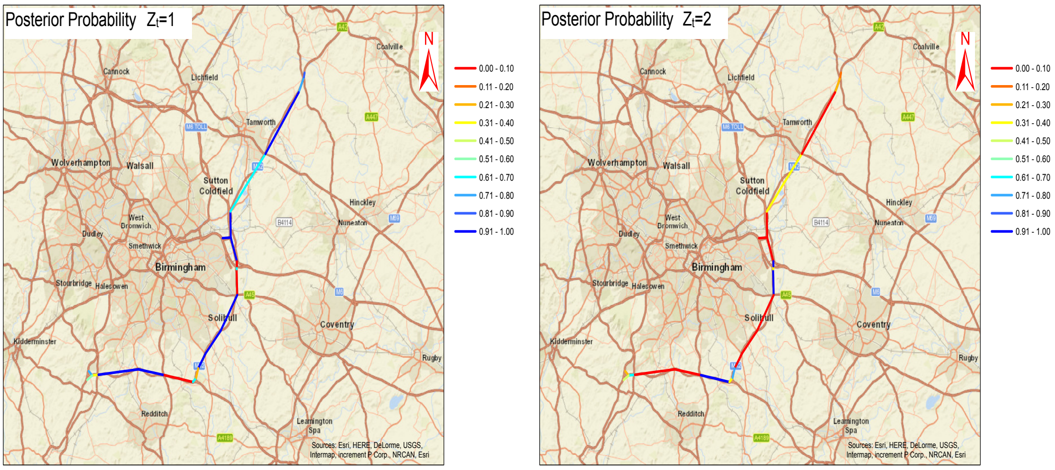


Figure 7.39: The spatial results mapped at segment level for the M42 motorway.

7.7 Summary and discussion

Under a Bayesian framework, we have introduced a spatial modelling approach for traffic crashes using a Poisson hidden Markov model (PHMM). Models of this class are able to accommodate the heterogeneity and serial dependence in count data simultaneously.

Our methodology was demonstrated using an application involving traffic crash data. We have introduced the idea of modelling the spatial dependence at segment level. The application included three different crash datasets from three highways in the UK, the M5, M6 and M42. We have considered a series of models with increasing complexity. Due to the small size of the three datasets considered in this study, we have carefully investigated each model. We have performed a sensitivity analysis by conducting numerical and visual inference on the state-specific crash rate parameter, λ_{z_t}, under chosen priors. These proposed prior choices varied from very highly non-informative towards less diffuse priors. We have used the Gelman-Rubin statistic, the Geweke statistic, the ACF and trace plots for convergence monitoring of the posterior distributions of the state-specific crash rates parameters of each model.

In general, the results revealed that a lack in convergence may reflect problems in model identifiability using highly non-informative priors with relatively large models. This could be due to the small size of the available data sets. The process of model choice was implemented by using the criteria that were found to work well in Chapter 6. Overall, the findings of model selection and assessment have suggested that some road safety datasets considered in this chapter were described by a small number of states. The PHMMs have provided a different way for analyzing spatial dependence on networks susceptible to road crashes.

The univariate PHMM employed in this chapter can be considered as an approach for a simple preliminary analysis of the traffic safety situation of a certain highway with different levels of road crash risk (e.g. very low, low, high, etc.). This can assist highway authorities to arrange their priorities in road treatment. As an extension, in future work, we could develop a multivariate Poisson HMM to consider the severity and type of crashes at the individual level at each segment.

Chapter 8

Conclusion

8.1 Summary

This thesis contributes in introducing new ideas for Bayesian model selection criteria to determine the number of states, K, in a hidden Markov model. Chapter 1 provided a brief review of Bayesian theory and MCMC methods used to estimate model parameters. Chapter 2 discussed finite mixture models in which the discrete latent variables are assumed to be independent, a special and simplified case of HMMs. Chapter 3 introduced the fundamental definitions of the HMM. We briefly reviewed the problems inherent in HMMs: the likelihood function and parameter estimation of a HMM. In Chapter 4 we introduced a Bayesian approach to HMMs. We concentrated on the sampling process of the hidden state sequence of a HMM. We adopted the Direct Gibbs (DG) sampler, also called the local updating approach, to sample the hidden states. In addition, this chapter discussed the problem of label switching. In Chapter 5, we introduced the most common used likelihood-based model selection criteria: AIC, BIC and DIC in a HMM context. All those criteria have been built using the Bayesian framework in the sense that they are assessed over the posterior distribution. The construction of such criteria requires the availability of the likelihood function in closed form which is a challenge here. This is solved by using the data augmentation approach (Tanner and Wong, 1987). Based on the work of Celeux et al. (2006), we have considered the concept of focus in formulating our criteria. Accordingly, we have introduced several conditional and observed likelihood-based versions for our information criteria. In addition, the model selection issue was also discussed from a predictive viewpoint using a more recent criterion called the widely applicable information criterion (WAIC; Watanabe, 2009) which is a asymptotically equivalent to the leave one-out cross validated predictive density.

Chapter 6 was dedicated to assess the model selection criteria introduced in Chapter 5 where we investigated the performance of those criteria using simulation studies and also using an application to real data, by assuming a fixed and unknown number of hidden states, K. From those studies, our aim was to understand and uncover the performance of those criteria, given different several scenarios involving a generating data procedure, with different complexities, differ from the assumed structure. The results of model selection criteria assessment explained the following:

Firstly, for the simulation studies implemented on synthetic datasets, it was noted that all the versions of the recursive deviance-based AIC and BIC, and also the second version of the DIC, DIC_{rec_2}, had a satisfactory performance. More specifically, those criteria behaved well in selecting the correct model when assuming data that are generated from models with less complexities. On the other hand, the same criteria had a tendency to underestimate the real number of hidden states when generating data models with large sizes. It is worth noting that this latter behaviour of those criteria was reasonable, where matches to the real representation of the observed process produced by the generating data HMMs. More specifically, those criteria were more sensitive to the overlapping that appeared in the data which could lead to underestimating some redundant states in the model. The same above criteria also seem to perform well in the real data application, where they selected reasonable solutions to represent the data. On the other hand, the first version of the DIC, DIC_{rec_1}, selected the correct model among all the generating models considered, and it prefers no overlapping solutions, but has a tendency to overestimate the number of hidden states in the model. In contrast, this criterion had a poor performance in the real data example, where it preferred the most overfit model. Also, it provides arbitrary values of $p_{\text{DIC}_{rec_1}}$ which conflict with the principle of increasing the effective number of the parameters as the model complexity increases. This may be due to the unsatisfactory estimation of the plugged-in estimator $D_{rec}(\bar{\Theta})$ of this criterion.

Secondly, for the conditional deviance-based criteria, two versions of the BIC: BIC_{con_1} and BIC_{con_2} had the same behaviour as the AIC and BIC based on recursive deviance, which also select the right model in the less complicated generating models, and under-penalise the model in the case of more complexity generating models, indicating their sensitivity to the overlapping observed in the data. In the real data example, the BIC_{con_1} selects the more parsimonious model, whereas the BIC_{con_2} favors a lager model. In the simulation example, the AIC_{con_1} and BIC_{con_3} preferred the most overfitting models, suggesting it is not a reliable criterion. Moreover, both versions tended to select the large models in the real data application. The AIC$_{con_2}$, picked up the correct number of hidden states in the most cases. Nevertheless, it also tends to select the more complicated model for the real data. The DIC$_{con_1}$ had the worse performance, and favored the most overfitting model for all cases in the simulation study. However, it selects a reasonable solution for the real data and also introduces increased but too large values of $p_{\mathrm{DIC}_{con_1}}$. The DIC$_{con_2}$ performed well in the case of less complicated models, but highly underestimated the number of states K in the main part of cases. This criterion also selected a sensible solution for the waiting time data example and provides $p_{\mathrm{DIC}_{con_1}}$ with increased and large values.

Thirdly, the WAIC, based on predictive pointwise calculations, had a worst performance, as it always tends to favour the most overfitting model in all examples considered in this thesis. This can indicate that this criterion is not appropriated for HMMs.

Finally, all versions of the AIC, BIC and DIC based on recursive deviances have provided a lower level of variability compared with the versions of AIC, BIC and DIC based on conditional deviances as well as the WAIC, indicating their accuracy. The reason might be that the conditional deviance-based versions involve high-dimensional vectors of hidden states which are treated as an additional focus along with the state-specific parameters. This would lead to large variations in Monte Carlo simulations. In contrast, the hidden states are dealt with as missing data in the recursive deviance-based versions, where we sum all possible hidden states when computing the observed likelihood. This may contribute in reducing the variability in Monte Carlo simulations.

In Chapter 7 we presented a Bayesian modelling framework to capture the spatial dependence in count data using the PHMM. Our methodology has been illustrated via an application to traffic safety crashes for three highways in the UK. Our interest was in identifying highway segments which have distinctly higher crash rates of the safety process. Selecting an optimal number of states is an important part of the interpretation required by highways managers. As a result, we have used model-selection criteria to determine the optimal number of states. We have also used several goodness-of-fit checks to assess each model fitted to the data.

We implemented an MCMC algorithm and checked convergence. We have used tools such as the Gelman-Rubin statistic, the Geweke statistic, the ACF plots and trace plots for convergence monitoring of the posterior distributions of the state-specific crash rates parameters of each model. We explored a range of priors to check for prior sensitivity, a potential problem given small sample sizes. We saw a lack of convergence when fitting models that may reflect problems in model identifiability due to over-fitting when using highly non-informative priors. The posterior distribution of the crash rate parameters of models with large sizes were usually more sensitive to highly non-informative priors, and may have provided unrealistic estimates. Some criteria for the high-dimensional models had unsatisfactory performance for all chosen priors. This may be due to the lack of identifiability of the parameter estimates.

Overall, we believe that model selection and assessment have suggested that the datasets considered in this chapter could be well described by a small number of states. The PHMMs have provided a different model analyzing spatial dependence on networks susceptible to road crashes. It is possible to identify segments with a higher posterior probability of classification in a high risk state, a finding that would prioritise management action.

8.2 Future work

There are several interesting extensions of the work done in this thesis, some of which are summarised below.

There may be more efficient samplers. We have used the DG sampler which required extensive thinning. It would be good to use more computationally efficient samplers such as FBG or Hamiltonian Monte Carlo (HMC; Neal (2011)) algorithm.

In general, the problem of the estimation of the penalty term, p_{DIC}, of the DIC or the bias correction term (Gelman et al., 2014) in the WAIC can be due to the mischoice of focus in the model that have to be carefully selected. This need further attention.

In Chapter 7, we introduced a univariate PHMM to model and diagnose the spatial heterogeneity represented by the crash rates on highways. As explained, such a model can be considered as an approach that introduces a simple preliminary analysis for the traffic safety situation to a certain highway. As an extension, in future work, we could develop a multivariate Poisson HMM to consider the severity and type of crashes of the considered datasets of highways: M5, M6 and M42. Such a model would potentially enable inside on other interesting aspects for the safety authorities, such as the injury types following a crash at the individual level at each segment, so that

$$\begin{aligned} Y_{itr} = \mathrm{y}_{itr} | \mathrm{z}_{tr}, \lambda_{itr} &\sim \text{Poisson}(o_{tr}\lambda_{i\mathrm{z}_{tr}r}), \\ \mathrm{z}_{tr} &\sim \text{Markov}(\pi, \mathbf{A}), \end{aligned}$$

where

y_{itr}: refers to a discrete outcome of the crash injury type i (for instance, no-injury, injury and fatality) observed at the t^{th} segment of the r^{th} highway, given a fixed time period.

o_{tr}: is the expected number of crashes of the t^{th} segment of the r^{th} highway which is derived based on some traffic safety variables such the length and traffic volume of a certain segment.

$\lambda_{iz_{tr}r}$: is the state-specific crash rate of the crash injured type i of the t^{th} segment of the r^{th} highway.

Appendix A

Derivations

A.1 The full conditional posteriors of Normal HMM and Poisson HMM

A.1.1 The Normal HMM

The posterior distribution of a Normal HMM can be derived as follows:

1- Likelihood

$$\begin{aligned} L(\mu,\tau,\mathbf{A};\mathbf{y},\mathbf{z}) &= \pi_{z_1}\phi_{z_1}(y_1|\mu_{z_1},\tau_{z_1})a_{z_1z_2}\phi_{z_2}(y_2|\mu_{z_2},\tau_{z_2})...a_{z_{T-1}z_T}\phi_{z_T}(y_T|\mu_{z_T},\tau_{z_T}) \\ &= (\pi_{z_1})(a_{z_1z_2}...a_{z_{T-1}z_T})(\phi_{z_1}(y_1|\mu_{z_1},\tau_{z_1})...\phi_{z_T}(y_T|\mu_{z_T},\tau_{z_T})) \\ &= \prod_{k=1}^{K}\pi_j^{N_k}\prod_{j=1}^{K}\prod_{k=1}^{K}(a_{jk})^{N_{jk}}\prod_{j=1}^{K}\prod_{t:z_t=j}^{T_j}\phi_j(y_t|\mu_j,\tau_j) \\ &= \prod_{k=1}^{K}\pi_k^{N_j}\prod_{j=1}^{K}\prod_{k=1}^{K}(a_{jk})^{N_{jk}}\prod_{j=1}^{K}\prod_{t:z_t=j}^{T_j}\sqrt{\frac{\tau_j}{2\pi}}\,\exp\left(-\frac{\tau_j}{2}(y_t-\mu_j)^2\right). \end{aligned} \tag{A.1}$$

2- Prior distributions of the π, $\mathbf{A}=\{a_{jk}\}$, μ and τ respectively

$$Pr(\pi|\delta) = \prod_{k=1}^{K}Pr(\pi_k) = \prod_{k=1}^{K}\pi_k^{\delta_k-1}. \tag{A.2}$$

$$Pr(a_{j.}|\delta) = \prod_{j=1}^{K}Pr(a_{j.}|\delta) = \prod_{j=1}^{K}\prod_{k=1}^{K}\{a_{j.}\}^{\delta_k-1}. \tag{A.3}$$

$$\begin{aligned} Pr(\mu_j|\xi,\eta) = \prod_{j=1}^{K}Pr(\mu_j) &\propto \prod_{j=1}^{K}\exp\left(-\frac{\eta}{2}(\mu_j-\xi)^2\right) \\ &= \exp\left(\frac{\eta}{2}\sum_{j=1}^{K}(\mu_j-\xi)^2\right) \end{aligned} \tag{A.4}$$

$$Pr(\tau_j|\kappa,\nu) = \prod_{j=1}^{K} Pr(\tau_j) \propto \prod_{j=1}^{K} \tau_j^{\kappa-1} \exp(-\nu\tau_j) \tag{A.5}$$

3- Joint Posterior distributions

$$\begin{aligned} Pr(\mu,\tau,\mathbf{A},\boldsymbol{\pi}|\mathbf{y},\mathbf{z}) = &\prod_{k=1}^{K} \pi_k^{N_k} \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{N_{jk}} \prod_{j=1}^{K}\prod_{t:\mathrm{z}_t=j}^{T_j} \sqrt{\tau_j}\, \exp\Big(-\frac{\tau_j}{2}(\mathrm{y}_t-\mu_j)^2\Big) \\ &\times \prod_{k=1}^{K} \pi_k^{\delta_k-1} \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{\delta_k-1} \exp\left(\frac{\eta}{2}\sum_{j=1}^{K}(\mu_j-\xi)^2\right) \times \prod_{j=1}^{K} \tau^{\kappa-1} \exp(-\nu\tau_j). \end{aligned} \tag{A.6}$$

4- Full Conditionals of the $\boldsymbol{\pi}$, $\mathbf{A} = \{a_{jk}\}$, μ and τ respectively

$$\begin{aligned} Pr(\boldsymbol{\pi} \mid \mathbf{y},\mathbf{z},\mu,\tau) &= \prod_{k=1}^{K} Pr(\pi_k)^{N_k} \prod_{k=1}^{K} \pi_k^{\delta_k-1}, \\ &\propto \prod_{k=1}^{K} Pr(\pi_k)^{N_k+\delta_k-1}, \\ &= Dir(N_k+\delta_k). \end{aligned} \tag{A.7}$$

$$\begin{aligned} Pr(a_{j.}|\mathbf{y},\mathbf{z},\mu,\tau) &\propto \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{N_{jk}} \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{\delta_k-1}, \\ &\propto \prod_{j=1}^{K}\prod_{k=1}^{K} (a_{jk})^{N_{jk}+\delta_k-1}, \\ &= Dir(N_{j.}+\delta_k). \end{aligned} \tag{A.8}$$

$$
\begin{aligned}
Pr(\mu_j \mid \mathbf{y}, \mathbf{z}, \tau, \mathbf{A}) &\propto \prod_{t:\mathrm{z}_t=j}^{T_j} \exp\left\{-0.5\tau_j(\mathrm{y}_t-\mu_j)^2\right\} \exp\left\{-0.5\eta(\mu_j-\zeta)^2\right\}, \\
&= \exp\left\{-0.5\tau_j \sum_{t:\mathrm{z}_t=j}^{T_j} (\mathrm{y}_t-\mu_j)^2\right\} \exp\left\{-0.5\eta(\mu_j-\zeta)^2\right\}, \\
&= \exp\left\{-0.5\tau_j \sum_{t:\mathrm{z}_t=j}^{T_j} (\mathrm{y}_t-\mu_j)^2\right\} \exp\left\{-0.5\eta(\mu_j-\zeta)^2\right\}, \\
&= \exp\left\{-0.5\tau_j \sum_{t:\mathrm{z}_t=j}^{T_j} (\mathrm{y}_t-\mu_j)^2 - 0.5\eta(\mu_j-\zeta)^2\right\}, \\
&= \exp\left\{-0.5\tau_j \sum_{t:\mathrm{z}_t=j}^{T_j} (\mathrm{y}_t^2-2\mathrm{y}_t\mu_j+\mu_j^2) - 0.5\eta(\mu_j^2-2\mu_j\zeta+\zeta^2)\right\}, \\
&= \exp\left\{-0.5\tau_j \left(\sum_{t:\mathrm{z}_t=j}^{T_j} \mathrm{y}_t^2 - 2\mu_j \sum_{t:\mathrm{z}_t=j}^{T_j} \mathrm{y}_t + T_j\mu_j^2\right) - 0.5\eta(\mu_j^2-2\mu_j\zeta+\zeta^2)\right\}, \\
&= \exp\left\{-0.5\left[(\tau_jT_j+\eta)\mu_j^2\left(\tau_j \sum_{t:\mathrm{z}_t=j}^{T_j} \mathrm{y}_t+\eta\zeta\right)\mu_j + \left(\tau_j \sum_{t:\mathrm{z}_t=j}^{T_j} \mathrm{y}_t^2+\eta\zeta\right)\right]\right\}, \\
&\propto \exp\left\{-\frac{\tau_jT_j+\eta}{2}\left[\mu_j^2 - 2\mu_j\frac{\tau_j\sum_{t:\mathrm{z}_t=j}^{T_j}\mathrm{y}_t+\eta\zeta}{\tau_jT_j+\eta} + \left(\frac{\tau_j\sum_{t:\mathrm{z}_t=j}^{T_j}\mathrm{y}_t+\eta\zeta}{\tau_jT_j+\eta}\right)^2\right]\right\}, \\
&= \exp\left\{-\frac{\tau_jT_j+\eta}{2}\left(\mu_j - \frac{\tau_j\sum_{t:\mathrm{z}_t=j}^{T_j}\mathrm{y}_t+\eta\zeta}{\tau_jT_j+\eta}\right)^2\right\}, \\
&= Normal\left(\frac{\tau_j\sum_{t:\mathrm{z}_t=j}^{T_j}\mathrm{y}_t+\eta\zeta}{\tau_jT_j+\eta}, \frac{1}{\tau_jT_j+\eta}\right).
\end{aligned}
\tag{A.9}
$$

If we use σ^2 instead of τ, the formula of the full conditional distribution of μ in Equation (A.9) becomes

$$
\begin{aligned}
Pr(\mu_j \mid \mathbf{y}, \mathbf{z}, \tau, \mathbf{A}) &= Normal\left(\frac{\frac{\sum_{t:\mathrm{z}_t=j}^{T_j}\mathrm{y}_t}{\sigma_j^2}+\eta\zeta}{\frac{T_j}{\sigma_j^2}+\eta}, \frac{1}{\frac{T_j}{\sigma_j^2}+\eta}\right), \\
&= Normal\left(\frac{\sum_{t:\mathrm{z}_t=j}^{T_j}\mathrm{y}_t+\eta\zeta\sigma_j^2}{T_j+\eta\sigma_j^2}, \frac{\sigma_j^2}{T_j+\eta\sigma_j^2}\right).
\end{aligned}
\tag{A.10}
$$

$$\begin{aligned}
Pr(\tau_j \mid \mathbf{y},\mathbf{z},\mu,\mathbf{A}) &\propto \tau_j^{\kappa-1}\exp\left(-\nu\tau_j\right)\prod_{t:\mathrm{z}_t=j}^{T_j}\sqrt{\tau_j}\,\exp(-0.5\tau_j(\mathrm{y}_t-\mu_j)^2),\\
&= \tau_j^{\kappa-1}\exp\left(-\nu\tau_j\right)\,\tau_j^{0.5T_j}\,\exp(-0.5\tau_j\sum_{t:\mathrm{z}_t=j}^{T_j}(\mathrm{y}_t-\mu_j)^2),\\
&= \tau_j^{\kappa+0.5T_j-1}\exp\left\{-\tau_j\left(\nu+\sum_{t:\mathrm{z}_t=j}^{T_j}0.5(\mathrm{y}_t-\mu_j)^2\right)\right\},\\
&= Gamma\left(\kappa+0.5T_j,\nu+0.5\sum_{t:\mathrm{z}_t=j}^{T_j}(\mathrm{y}_t-\mu_j)^2\right). \qquad \text{(A.11)}
\end{aligned}$$

The product of Equation (A.11) is Gamma distribution with mean κ/ν and variance κ/ν^2. Consequently, the full conditional distribution of variance parameter σ^2 can be obtained as

$$\begin{aligned}
Pr(\sigma_j^2|\mathbf{y},\mathbf{z},\mu,\mathbf{A}) &= \frac{1}{Pr(\tau_j \mid \mathbf{y},\mathbf{z},\mu,\mathbf{A})},\\
&\equiv InvGamma\left(\kappa+0.5T_j,\nu+0.5\sum_{t:\mathrm{z}_t=j}^{T_j}(\mathrm{y}_t-\mu_j)^2\right). \qquad \text{(A.12)}
\end{aligned}$$

A.1.2 The Poisson HMM

The prior and posterior distributions of the π and $\mathbf{A}$ are the same as those derived in the Normal HMM in subsection (A.1.1). We introduce here only the the prior and posterior of the state-dependent parameter of PHMM, λ, as well as the likelihood.

1- Likelihood:

$$\begin{aligned}
L(\lambda,\mathbf{A},\pi;\mathbf{y},\mathbf{z}) &= \prod_{j=1}^{K}\pi_j^{N_j}\prod_{j=1}^{K}\prod_{k=1}^{K}(a_{jk})^{N_{jk}}\prod_{j=1}^{K}\prod_{t:\mathrm{z}_t=j}^{T_j}f_j(\mathrm{y}_t|\lambda_j)\\
&= \prod_{j=1}^{K}\pi_j^{N_j}\prod_{j=1}^{K}\prod_{k=1}^{K}(a_{jk})^{N_{jk}}\prod_{j=1}^{K}\prod_{t:\mathrm{z}_t=j}^{T_j}\lambda^{\mathrm{y}_t}e^{-\lambda}. \qquad \text{(A.13)}
\end{aligned}$$

2- Priors for λ:

$$\begin{aligned}
Pr(\lambda) &= \prod_{j=1}^{K}(\lambda_j|\kappa,\nu)\\
&\propto \prod_{j=1}^{K}\lambda_j^{\kappa-1}\exp(-\nu\lambda_j). \qquad \text{(A.14)}
\end{aligned}$$

3- Joint Posterior

$$Pr(\lambda,\mathbf{A},\pi \mid \mathbf{y},\mathbf{z}) = \prod_{j=1}^{K}\pi_j^{N_j}\prod_{k=1}^{K}\pi_j^{\delta_j-1}\prod_{j=1}^{K}\prod_{k=1}^{K}(a_{jk})^{N_{jk}}\prod_{j=1}^{K}\prod_{k=1}^{K}(a_{jk})^{\delta_k-1} \tag{A.15}$$
$$\times\prod_{j=1}^{K}\prod_{t:\mathrm{z}_t=j}^{T_j}\lambda_j^{\mathrm{y}_t}e^{-\lambda_j}\prod_{j=1}^{K}\lambda^{\kappa-1}\exp(-\nu\lambda_j).$$

4- Full Conditional of λ

$$\begin{aligned}Pr(\lambda_j \mid \mathbf{y},\mathbf{z},\mathbf{A},\pi) &\propto \left(\prod_{t:\mathrm{z}_t=j}^{T_j}\lambda_j^{\mathrm{y}_t}e^{-\lambda_j}\right)\times\lambda_j^{\kappa-1}e^{-\nu\lambda_j},\\ &= \left(\lambda_j^{\sum_{t:\mathrm{z}_t=j}\mathrm{y}_t}e^{-T_j\lambda_j}\right)\times\lambda_j^{\kappa-1}e^{-\nu\lambda_j},\\ &= \lambda_j^{\sum_{t:\mathrm{z}_t=j}\mathrm{y}_t+\kappa-1}e^{-\lambda_j(T_j+\nu)},\\ &= Gamma(\kappa+\sum_{t:\mathrm{z}_t=j}\mathrm{y}_t,\nu+T_j). \end{aligned} \tag{A.16}$$

A.2 The derivation of variables $\gamma_j(t)$ and $\xi_{jk}(t)$

The $\xi_{jk}(t)$ can be defined as the posterior joint probability of being in state j at time t, and state k at time $t+1$, given the model, Θ, and observation sequence $\mathbf{y}$, i.e.

$$\xi_{jk}(t) = p(\mathrm{z}_t = j, \mathrm{z}_{t+1} = k|\mathbf{y},\Theta),$$

which can be calculated from the 'α' and 'β' variables as follows

$$\begin{aligned}\xi_{jk}(t) &= Pr(\mathrm{z}_t = j,\mathrm{z}_{t+1} = k|\mathbf{y},\Theta),\\ &= \frac{Pr(\mathrm{z}_t = j,\mathrm{z}_{t+1} = k,\mathbf{y}|\Theta)}{L(\Theta|\mathbf{y})},\\ &= \frac{Pr(\mathrm{y}_1,\mathrm{y}_2,...,\mathrm{y}_t,\mathrm{z}_t = j|\Theta)a_{jk}f_k(\mathrm{y}_{t+1})Pr(\mathrm{y}_{t+1},\mathrm{y}_{t+2},...,\mathrm{y}_T|\mathrm{z}_t = j,\Theta)}{L(\Theta|\mathbf{y})},\\ &= \frac{\alpha_t(j)a_{jk}f_k(\mathrm{y}_{t+1})\beta_{t+1}(k)}{L(\Theta|\mathbf{y})},\\ &= \frac{\alpha_t(j)a_{jk}f_k(\mathrm{y}_{t+1})\beta_{t+1}(k)}{\sum_{j=1}^{K}\sum_{k=1}^{K}\alpha_t(j)a_{jk}f_k(\mathrm{y}_{t+1})\beta_{t+1}(k)}.\end{aligned} \tag{A.17}$$

The variable $\gamma_j(t)$ is a posteriori probability variable and can be defined as

$$\gamma_j(t) = Pr(\mathrm{z}_t = j|\mathbf{y},\Theta), \tag{A.18}$$

i.e., the probability of being in state j at time t, given the observation sequence $\mathbf{y}$ and model Θ. We can compute $\gamma_j(t)$ from the forward and backward variables as follows

$$\begin{aligned}
\gamma_j(t) &= Pr(\mathrm{z}_t = j|\mathbf{y}, \Theta), \\
&= \frac{Pr(\mathbf{y}, \mathrm{z}_t = j|\Theta)}{L(\Theta|\mathbf{y})}, \\
&= \frac{Pr(\mathbf{y}, \mathrm{z}_t = j|\Theta)}{\sum_{j=1}^{K} Pr(\mathbf{y}, \mathrm{z}_t = j|\Theta)}, \\
&= \frac{Pr(\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_t, \mathrm{z}_t = j|\Theta) Pr(\mathrm{y}_{t+1}, \mathrm{y}_{t+2}, ..., \mathrm{y}_T|\mathrm{z}_t = j, \Theta)}{\sum_{j=1}^{K} Pr(\mathrm{y}_1, \mathrm{y}_2, ..., \mathrm{y}_t, \mathrm{z}_t = j|\Theta) Pr(\mathrm{y}_{t+1}, \mathrm{y}_{t+2}, ..., \mathrm{y}_T|\mathrm{z}_t = j, \Theta)}, \\
&= \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{K} \alpha_t(j)\beta_t(j)}.
\end{aligned} \tag{A.19}$$

Appendix B

Python Codes

B.1 Generating observations from Normal and Poisson HMMs

```python
""" A function to sample from the Multinomial distribution"""
def Multinomial(probability):
    probability[np.isnan(np.array(probability))]=0
    return np.where(np.random.multinomial(1,probability)==1)[0][0]

"""A function to simulate continuous data set of length T from Normal distribution"""
def norm_obs(mu,sig):
    return np.random.normal(mu,sig)

"""A function to simulate discrete data set of length T from Poisson distribution"""
def pois_obs(lambda_):
    np.random.poisson(lambda_)

"""Simulate dataset from K-state Normal HMM"""
def Normal_K_HMM(T,pi,A,Mu,Sig):
    Hidden=np.zeros([T],int)
    Obs=np.zeros([T])
    Hidden[0]=Multinomial(pi)
    Obs[0]=norm_obs(Mu[Hidden[0]],np.sqrt(Sig[Hidden[0]]))
    for t in range(1,T):
        Hidden[t]=Multinomial(A[Hidden[t-1]])
        Obs[t]=norm_obs(Mu[Hidden[t]],np.sqrt(Sig[Hidden[t]]))
    return Obs, Hidden

"""Simulate data from K-state Normal HMM"""
def Poisson_K_HMM(T,pi,A,lambda):
    Hidden=np.zeros([T],int)
    Obs=np.zeros([T])
    Hidden[0]=Multinomial(pi)
    Obs[0]=pois_obs(lambda[Hidden[0]])
    for t in range(1,T):
        Hidden[t]=Multinomial(A[Hidden[t-1]])
        Obs[t]=pois_obs(lambda[Hidden[t]])
    return Obs, Hidden
#### Inputs
T=T # length of the required sequence
pi=pi # initial distribution vector of dimension (1*K)
A=A #  traintion matrix of dimension (K*K)
Mu=Mu #  vector of means of dimension (1*K)
Sig=Sig #  vector of variances of dimension (1*K)
lambda=lambda #  vector of lambda's of dimension (1*K)
```

B.2 Estimation, convergence checking, model selection of Normal HMMs

```python
from __future__ import division
import numpy as np
import matplotlib.pyplot as pl
from numpy.random import normal,gamma,multinomial
from scipy.stats import norm,poisson
from scipy.stats.distributions import invgamma

"""A function to simulate continuous data set of length T from Normal distribution"""
def norm_obs(mu,sig):
    return np.random.normal(mu,sig)

"""A function to compute the probability density of Normal distribution"""
def norm_pdf(data, mu, sigma):
    return norm.pdf(data,mu,sigma)

"""A function to simulate discrete data set of length T from Poisson distribution"""
def pois_obs(lambda_):
    np.random.poisson(lambda_)

""" Define the Dirichlet function. This function return samples (probabilities) from Dirichlet
    distribution that use to define the full conditional posterior and prior distributions
    for the mixing weights. delta is a vector of hyperparameters, which is assumed =1 for
    k=1,2, ..., K. delte=np.ones((k)) return (1,1, ....,1_K)"""
def Dirichlet_(delta,x):
    return np.random.dirichlet(delta+(x),1)[0]

""" A function to sample from the Multinomial distribution"""
def Multinomial(probability):
    probability[np.isnan(np.array(probability))]=0
    return np.where(np.random.multinomial(1,probability)==1)[0][0]

""" A function to compute all sufficient statistics required which are:
    1- sum of observations that are generated from state j.
    2- number of observations that are generated from state j.
    3- average of observation that are allocated to the state j.
    4- sum of square of observations that are generated from state j."""
def Sufficient_Statistics(observation,hidden,k):
    sum_obs_j=np.zeros((k))
    num_obs_j=np.zeros((k))
    for j in range(k):
    sum_obs_j[j]=np.sum((hidden==j)*observation)
    num_obs_j[j]=np.sum(hidden==j)
    return sum_obs_j,num_obs_j

"""This function is specified for counting the the number of transitions from the state
   j, denoted by a vector N_j of dimension 1*K. N_j: sum up from t=1 until t=T."""
def Counting_Iinitial(hidden,k):
    Number_initial_state=np.zeros((k))
    for j in range(k):
Number_initial_state[j]=np.sum(hidden==j)
    return Number_initial_state

""" This function is specified for counting the number of transitions from j into k,
    denoted by a matrix N_jk of dimension K*K. N_jk: sum up from t=1 until t=T-1"""
def Counting_Transition(observation,hidden,k):
    Number_transition_state=np.zeros((k,k))
    for t in range(len(observation)-1):
        Number_transition_state[hidden[t],hidden[t+1]] = Number_transition_state[hidden[t], hidden[t+1]]+ 1
    return Number_transition_state

""" Update the transition matrix probabilities and initial state probability from
    corresponding counts. Transi: a matrix of dimension K*K, its rows and the
    vector Initial are independently updated from Dirichlet distribution."""
def Update_Transition(delta,Number_transition_state,k):
    Transi=np.zeros((k,k))
    for j in range(k):
        Transi[j,:]=Dirichlet_(delta,Number_transition_state[j,:])
    return Transi

""" Update the transition matrix probabilities and initial state probability from
    corresponding counts. Transi: a matrix of dimension K*K, its rows and the vector
    Initial are independently updated from Dirichlet distribution."""
def Update_Initial(delta,Number_initial_state):
    return  Dirichlet_(delta,Number_initial_state)

""" Code for computing the recursive log-likelihood"""
def Recursive_loglikelihood(k,observation,pi,AAA,mu,sigma):
    # we create a function to compute the recursive log-likelihood.
    # This function returns the log-likelihood based on - sum the
    # logarithm of the scaling factors C.
    T=len(observation)
    ll=[] #log-likelihood
    alpha_star=np.zeros((k,T))# forward unscaled
    alpha_hat=np.zeros((k,T)) # forward scaled
    C=np.zeros([T]) # scaling factors
    for s in range(k):
        alpha_hat[s,0] = pi[s]* norm_pdf(observation[0],mu[s],sigma[s])
    C[0]=1./(np.sum(alpha_hat[:,0]))
    alpha_star[:,0]=C[0]*alpha_hat[:,0]
    for t in range(1,T):
        for j in range(k):
            for i in range(k):
            alpha_hat[j,t] += alpha_star[i,t-1]*AAA[i,j] *norm_pdf(observation[t],mu[j],sigma[j])
        C[t]=1./(np.sum(alpha_hat[:,t]))
        alpha_star[:,t]=C[t]*alpha_hat[:,t]
    ll=-np.sum(np.log(C))
    return ll,alpha_hat,alpha_star

""" Code for updating the hidden state locally"""
def Local_updating(k,observation,pi,AAA,mu,sigma,hidden):
    # A function to update the hidden state locally.
    # Local updating method assumes there is a given hidden state sequence. We
    # denoted it as 'hidden' and the updating is done in the new sequence. We
    # denoted it as 'hidden_new'.
    T=len(observation)
    alloc_non_normlized=np.zeros((k,T))
    alloc_normlized=np.zeros((k,T))
    hidden_new=np.zeros([T], dtype=int)# new hidden state vector
    for i in range(k):
        alloc_non_normlized[i,0]=pi[i]*AAA[i,hidden[1]]*norm_pdf(observation[0],mu[i],sigma[i])
    alloc_normlized[:,0]=alloc_non_normlized[:,0]*1.0/(np.sum(alloc_non_normlized[:,0]))
    hidden_new[0]=Multinomial(alloc_normlized[:,0])
    for t in range(1,T-1):
        for i in range(k):
            alloc_non_normlized[i,t]=AAA[hidden_new[t-1],i]*(norm_pdf(observation[t],mu[i],sigma[i]))
            *AAA[i,hidden[t+1]]
        alloc_normlized[:,t]=alloc_non_normlized[:,t]*1.0/(np.sum(alloc_non_normlized[:,t]))
        hidden_new[t]=Multinomial(alloc_normlized[:,t])
    for i in range(k):
        alloc_non_normlized[i,T-1]=AAA[hidden_new[T-2],i]*norm_pdf(observation[T-1],mu[i],sigma[i])
    alloc_normlized[:,T-1]=alloc_non_normlized[:,T-1]*1.0/(np.sum(alloc_non_normlized[:,T-1]))
    hidden_new[T-1]=Multinomial(alloc_normlized[:,T-1])
    return alloc_normlized,hidden_new

""" Code for computing the conditional log-likelihood"""
def Conditional_loglikelihood(k,observation,pi,AAA,mu,sigma,hidden):
    T=len(observation)
    con_loglike= np.zeros((T))
    for t in range(T):
        con_loglike[t]=np.log(norm_pdf(observation[t],mu[hidden[t]],sigma[hidden[t]]))
```

```python
    return np.sum(con_loglike)

''' Code to compute the highest posterior density obtained over MCMC draws'''
def highestPostDensity(data,k, pi, mu,sig):
    ''' This function return the index of highest posterior density obtained
        over MCMC draws'''
    #Max_Density_AtEach_Iteration=np.zeros([iteration])
    Max_Density=[]
    Density=np.zeros([len(data)])
    for t in range(len(data)):
        for s in range(k):
            Density[t]+=(pi[s]* norm.pdf(data[t],mu[s],sig[s]))
    Max_Density.extend([np.max(Density)])
    return Density,Max_Density[0]

""" Code for estimating the Normal HMM parameters using Gibbs sampler"""
def Normal_HMM_Gibbs(obs,T,M,d,K,zeta,eta,a,b,delta):

    """Storage"""
    alloc_post=np.zeros([M,K,T],np.float)
    N_jk=np.zeros([M,K,K],int)
    N_j=np.zeros([M,K],int)
    A_post=np.zeros([M,K,K],np.float)
    p_post=np.zeros([M,K],np.float)
    z_post=np.zeros([M,T], dtype=int)
    n_y=np.zeros([M,K],int)
    sum_y=np.zeros([M,K],np.float)
    mus_post=np.zeros([M,K],np.float) # storage the samples of posterior of mean parameter
    s_2=np.zeros([M,K],np.float)
    sig_post=np.zeros([M,K],np.float) # storage the samples of posterior of variance parameter
    tau_post=np.zeros([M,K],np.float)
    recursive_loglikelihood=np.zeros([M],np.float)
    conditional_loglikelihood=np.zeros([M],np.float)
    ippd=np.zeros([M,T]) # array for the integrated pointwise predictive density
    ilppd=np.zeros([M,T])# array for the integrated log pointwise predictive density

    """Initialization """
    for r in range(K):
        tau_post[0,r]=1
        sig_post[0,r]=tau_post[0,r]
        mus_post[0,r]=0
        A_post[0,r,:]=Dirichlet_(delta,np.ones((K)))
    p_post[0,:]=Dirichlet_(delta,np.ones((K)))
    z_post[0,0]=Multinomial(p_post[0,:])
    for t in range(1,T):
        z_post[0,t]=Multinomial(A_post[0,z_post[0,t-1],:])#np.where(np.random.multinomial(1,)==1)[0][0]

    """compute the recursive and conditional likelihoods at iteration 0"""
    recursive_loglikelihood[0]=Recursive_loglikelihood(K,obs,p_post[0,:],A_post[0,:,:],mus_post[0,:],
     (sig_post[0,:]))[0]
    conditional_loglikelihood[0]=Conditional_loglikelihood(K,obs,p_post[0,:],A_post[0,:,:],mus_post[0,:],
     (sig_post[0,:]),z_post[0,:])

    """Run MCMC"""
    for m in range(1,M):
        sum_y[m,:]=Sufficient_Statistics(obs,z_post[m-1,:],K)[0]
        n_y[m,:]=Sufficient_Statistics(obs,z_post[m-1,:],K)[1]

        """Updating  mean parameter"""
        mus_post[m,:]=np.random.normal(((eta*zeta)+(tau_post[m-1,:]*sum_y[m,:]))*1./(zeta+(n_y[m,:]*tau_post[m-1,:])),
        np.sqrt(1./(zeta+(n_y[m,:]*tau_post[m-1,:]))))

        """ Identifiability constraints (IC) to handle the label switching"""
        IC=sorted(mus_post[m,:])
        mus_post[m,:]=IC

        """Sufficient statistics of the sum of squares of observations  generated from state k."""
        for s in range(K):
            s_2[m,s]=np.sum((z_post[m,:]==s)*((obs-mus_post[m,s])**2))

        """Updating  variance parameter"""
        tau_post[m,:]=np.random.gamma(shape=a+(0.5*(n_y[m,:])), scale=1./(b+(0.5*s_2[m,:])))
        sig_post[m,:]=1./np.sqrt(tau_post[m,:])

        """Updating the inital and transitions probabilities"""
        N_jk[m,:,:]=Counting_Transition(obs,z_post[m-1,:],K)
        N_j[m,:]=Counting_Iinitial(z_post[m-1,:],K)
        for r in range(K):
            A_post[m,r,:]=Dirichlet_(delta,N_jk[m,r,:])
        p_post[m,:]=Dirichlet_(delta,N_j[m,:])

        """ Local updating for hidden states"""
        alloc_post[m,:,:]=Local_updating(K,obs,p_post[m,:],A_post[m,:,:],mus_post[m,:],sig_post[m,:],
        z_post[m-1,:])[0]# Posterior allocation probabilities
        z_post[m,:]=Local_updating(K,obs,p_post[m,:],A_post[m,:,:],mus_post[m,:],sig_post[m,:],
        z_post[m-1,:])[1]# Posterior hidden states

        """Compute the recursive and conditional likelihoods at iteration m"""
        recursive_loglikelihood[m]=Recursive_loglikelihood(K,obs,p_post[m,:],A_post[m,:,:],mus_post[m,:],
        (sig_post[m,:]))[0]
        conditional_loglikelihood[m]=Conditional_loglikelihood(K,obs,p_post[m,:],A_post[m,:,:],mus_post[m,:],
        (sig_post[m,:]),z_post[m,:])

        """ Compute: integrated log predictive pointwise density and  the replicated data"""
        for t in range(T):
            ippd[m,t]=norm.pdf(obs[t],mus_post[m,z_post[m,t]], (sig_post[m,z_post[m,t]]))
            ilppd[m,t]=np.log(norm.pdf(obs[t],mus_post[m,z_post[m,t]], (sig_post[m,z_post[m,t]])))

        return mus_post,sig_post,A_post,p_post,alloc_post,z_post,recursive_loglikelihood,conditional_loglikelihood,
        ippd,ilppd

"""Checking convergence using Gelman-Rubin test"""
def Convergance_Normal_HMM_Gibbs(obs,T,M,d,K,dispersion_mus,dispersion_sig,dispersion_tran,chain):
    """ Call the paramter to be ckecked"""
    mus_post,sig_post,A_post,p_post,alloc_post,z_post,recursive_loglikelihood,conditional_loglikelihood,
    ippd,ilppd=Normal_HMM_Gibbs(obs,T,M,d,K,zeta,eta,a,b,delta)

    """create array for the parameters to be checked with adding a dimension to the number of chains"""
    mean_mus=np.zeros([K,chain])
    mean_sig=np.zeros([K,chain])
    mean_tran=np.zeros([K,K,chain])
    mean_init=np.zeros([K,chain])

    """ Variance within"""
    s_within_mus=np.zeros([K,chain])
    s_within_sig=np.zeros([K,chain])
    s_within_tran=np.zeros([K,K,chain])
    s_within_init=np.zeros([K,chain])

    """Variance between"""
    W_mus=np.zeros([K])
    W_sig=np.zeros([K])
    W_tran=np.zeros([K,K])
    W_init=np.zeros([K])
    B_mus=np.zeros([K])
    B_sig=np.zeros([K])
    B_tran=np.zeros([K,K])
    B_init=np.zeros([K])
    VAR_mus=np.zeros([K])
    VAR_sig=np.zeros([K])
    VAR_tran=np.zeros([K,K])
    VAR_init=np.zeros([K])

    """Compute the Gelubin-Rubin statistics, R_hat, for each parameter"""
    R_hat_mus=np.zeros([K])
    R_hat_sig=np.zeros([K])
    R_hat_tran=np.zeros([K,K])
    R_hat_init=np.zeros([K])

    for k in range(K):  # run for the  number of state
        for f in range(chain):    # run for the number of chains

        """compute mean of each parameter for each chain"""
        mean_mus[k,f]=np.mean(mus_post[f,d,k])
        mean_sig[k,f]=np.mean(sig_post[f,d,k])
        mean_init[k,f]=np.mean(p_post[f,d,k])
        for s in range(K):
            mean_tran[k,s,f]=np.mean(A_post[f,d,k,s])

        """compute with-in variance each paramter for each chain"""
        s_within_mus[k,f]=1./(len(d)-1)*np.sum(((mus_post[f,d,k]-mean_mus[k,f])**2))
        s_within_sig[k,f]=1./(len(d)-1)*np.sum(((sig_post[f,d,k]-mean_sig[k,f])**2))
        s_within_init[k,f]=1./(len(d)-1)*np.sum(((p_post[f,d,k ]-mean_init[k,f])**2))
        for s in range(K):
            s_within_tran[k,s,f]=1./(len(d)-1)*np.sum(((A_post[f,d,s,k]-mean_tran[k,s,f])**2))

        """compute between-variance of each chain for each parameter"""
        B_mus[k]=len(d)*1./(chain -1)*np.sum(mean_mus[k,f]-np.mean(mean_mus[k,:]))**2
        B_sig[k]=len(d)*1./(chain -1)*np.sum(mean_sig[k,f]-np.mean(mean_sig[k,:]))**2
        B_init[k]=len(d)*1./(chain -1)*np.sum(mean_init[k,f]-np.mean(mean_init[k,:]))**2
        for s in range(K):
            B_tran[k,s]=len(d)*1./(chain -1)*np.sum(mean_tran[k,s,f]-np.mean(mean_tran[k,s,:]))**2

        """compute the means of chains"""
        W_mus[k]=np.sum(s_within_mus[k,f])*1./(chain)
        W_sig[k]=np.sum(s_within_sig[k,f])*1./(chain)
        W_init[k]=np.sum(s_within_init[k,f])*1./(chain)
        for s in range(K):
            W_tran[k,s]=np.sum(s_within_tran[k,s,f])*1./(chain)

    """compute the variances of chains"""
    VAR_mus[k]=(1 -(1./(len(d))))*W_mus[k]  +(1./(len(d)))*B_mus[k]
    VAR_sig[k]=(1 -(1./(len(d))))*W_sig[k]  +(1./(len(d)))*B_sig[k]
    VAR_init[k]=(1 -(1./(len(d))))*W_init[k]  +(1./(len(d)))*B_init[k]
    for s in range(K):
        VAR_tran[k,s]=(1 -(1./(len(d))))*W_tran[k,s]  +(1./(len(d)))*B_tran[k,s]

    """compute R_hat of all paramters"""
    R_hat_mus[k]= np.sqrt(VAR_mus[k]*1./W_mus[k])
    R_hat_sig[k]= np.sqrt(VAR_sig[k]*1./W_sig[k])
    R_hat_init[k]= np.sqrt(VAR_init[k]*1./W_init[k])
    for s in range(K):
        R_hat_tran[k,s]= np.sqrt(VAR_tran[k,s]*1./W_tran[k,s])
    return R_hat_mus, R_hat_sig, R_hat_init,R_hat_tran

""" Model selection criteria"""
def model_selections(obs,T,M,d,K,zeta,eta,a,b,delta):
    collection_posteriors=[] # array for the posteriors distributions of the model parameters
    collection_posteriors_means=[] # array for the posteriors means of the model parameters
    collection_all_criteria_rec=[] # array for all criteria based on recursive deviance
    collection_all_criteria_con=[] # array for all criteria based on conditional  deviance
    collection_all_criteria_WAIC=[] # array for WAIC
    h=[] # number of actual parameters
    bar_deviance_rec=[] # array for posterior menas recursive deviance
    deviance_bar_rec=[] # array for recursive deviance evaluated at posterior means
    hat_deviance_rec=[] # array for minimum recursive deviance
    AIC_1_rec=[]
    BIC_1_rec=[]
    AIC_2_rec=[]
    BIC_2_rec=[]
    AIC_3_rec=[]
    BIC_3_rec=[]
    p_DIC_1_rec=[]
    DIC_1_rec=[]
    p_DIC_2_rec=[]
    DIC_2_rec=[]
    bar_deviance_con=[]
    deviance_MAP_con=[]
    hat_deviance_con=[]
    AIC_1_con=[]
    BIC_1_con=[]
    AIC_2_con=[]
    BIC_2_con=[]
    AIC_3_con=[]
    BIC_3_con=[]
    p_DIC_1_con=[]
    DIC_1_con=[]
    p_DIC_2_con=[]
    DIC_2_con=[]
    Integrated_lppd=[]
    p_iWAIC_var=[]
    iWAIC_var=[]

    """ Call the model estimation code"""
    mus_post,sig_post,A_post,p_post,alloc_post,z_post,recursive_loglikelihood,conditional_loglikelihood,
    ippd,ilppd=Normal_HMM_Gibbs(obs,T,M,d,K,zeta,eta,a,b,delta)

    """ Posterior means of parameters"""
    mean_mus_post=np.zeros([K])
    mean_sig_post=np.zeros([K])
    mean_transition=np.zeros([K,K])
    mean_initial=np.zeros([K])
        for s in range(K):
            mean_mus_post[s]=np.mean(mus_post[d,s])
            mean_sig_post[s]=np.mean((sig_post[d,s]))
            mean_initial[s]=np.mean(p_post[d,s])
            for r in range(K):
                mean_transition[r,s]=np.mean(A_post[d,r,s])

    """AIC, BIC, DIC"""
    h=(K**2)+(2*K)-1 #number of free parameters with respect of AIC and BIC

    """Recursive deviance-based AIC, BIC and DIC"""
    bar_deviance_rec=-2*(np.mean(recursive_loglikelihood[d]))
    deviance_bar_rec=-2*(Recursive_loglikelihood(K,obs,mean_initial,mean_transition,mean_mus_post,mean_sig_post)[0])
    hat_deviance_rec=-2*(max(recursive_loglikelihood[d]))
    AIC_1_rec= bar_deviance_rec+(2*h)
    BIC_1_rec= bar_deviance_rec+(h*np.log(T))
    AIC_2_rec= deviance_bar_rec+(2*h)
    BIC_2_rec=deviance_bar_rec+(h*np.log(T))
    AIC_3_rec=hat_deviance_rec+(2*h)
    BIC_3_rec=hat_deviance_rec+(h*np.log(T))
    p_DIC_1_rec= bar_deviance_rec-deviance_bar_rec
    DIC_1_rec= deviance_bar_rec+(2*p_DIC_1_rec)
    p_DIC_2_rec= bar_deviance_rec-hat_deviance_rec
    DIC_2_rec= hat_deviance_rec+(2*p_DIC_2_rec)

    '''Compute the highest posterior density at each MCMC draw'''
    HighestPostDensity=np.zeros([M])
    for m in xrange(M):
        HighestPostDensity[m]=highestPostDensity(obs, K1,p_post[m,:], mus_post[m,:],sig_post[m,:])[1]

    # Find the index of the highest posterior density among MCMC draws
    index=np.argmax(HighestPostDensity[d])
    # compupte log-conditional likelihood at the best pair (theta, z) corresponing the index
    # of the highest posterior density
    log_con_MAP=np.zeros([T])
    for t in range(T):
        log_con_MAP[t]=np.log(norm.pdf(obs[t],mus_post[d[index],z_post[d[index],t]],sig_post[d[index],
        z_post[d[index],t]]))
    """ Conditional deviance evaluated at best pair"""
    deviance_MAP_con=-2*(np.sum(log_con_MAP,axis=0))

    """ mean conditional deviance"""
    bar_deviance_con=-2*(np.mean(conditional_loglikelihood[d]))

    """ Minimum conditional deviance"""
    hat_deviance_con=-2*(max(conditional_loglikelihood[d]))
    AIC_1_con= bar_deviance_con+(2*h)
    BIC_1_con= bar_deviance_con+(h*np.log(T))
    AIC_2_con= deviance_MAP_con+(2*h)
    BIC_2_con=deviance_MAP_con+(h*np.log(T))
    AIC_3_con= hat_deviance_con+(2*h)
    BIC_3_con=hat_deviance_con+(h*np.log(T))
    p_DIC_1_con= bar_deviance_con-deviance_MAP_con
    DIC_1_con= deviance_MAP_con+(2*p_DIC_1_con)
    p_DIC_2_con= bar_deviance_con-hat_deviance_con
    DIC_2_con= hat_deviance_con+(2*p_DIC_2_con)

    """Compute iWAIC"""
    Integrated_lppd=np.sum(np.log(np.sum(ippd[d,:],axis=0)*1./len(d)))
    p_iWAIC_var=np.sum(np.var(ilppd[d,:],axis=0))
    iWAIC_var=-2*(Integrated_lppd)+(2*p_iWAIC_var)

    """ Appending all criteria"""
    collection_posteriors_means.append([mean_initial,mean_transition,mean_mus_post, mean_sig_post])
    collection_posteriors.append([alloc_post,mus_post,sig_post,p_post,A_post,z_post,recursive_loglikelihood,
    conditional_loglikelihood,ippd,ilppd])
    collection_all_criteria_rec.append([bar_deviance_rec,deviance_bar_rec,hat_deviance_rec,AIC_1_rec,BIC_1_rec,
    AIC_2_rec,BIC_2_rec,AIC_3_rec,BIC_3_rec,p_DIC_1_rec,DIC_1_rec,p_DIC_2_rec,DIC_2_rec])
    collection_all_criteria_con.append([bar_deviance_con,deviance_MAP_con,hat_deviance_con,AIC_1_con,BI C_1_con,
    AIC_2_con, BIC_2_con,AIC_3_con,BIC_3_con,p_DIC_1_con,DIC_1_con,p_DIC_2_con,DIC_2_con])
    collection_all_criteria_WAIC.append([Integrated_lppd,p_iWAIC_var,iWAIC_var])
    return h,collection_posteriors,collection_posteriors_means,collection_all_criteria_rec,
          collection_all_criteria_con, collection_all_criteria_WAIC

""" Sampling information"""
obs=obs # obs is observation sequence
T=len(obs) # compute the length of data
M=M    # put a number of iterations
burnin=burnin    # put burn-in period
d=range(burnin,M)
K=K # Number of states, we have to put a number, e.g,K=3
zeta=0.001;eta=0;a=0.001;b=0.001;delta=np.ones((K),int) # hyper-parameters
h,collection_posteriors,collection_posteriors_means,collection_all_criteria_rec,
collection_all_criteria_con,collection_all_criteria_WAIC=model_selections(obs,T,M,d,K,zeta,eta,a,b,delta)
```

B.3 Code of Chapter 6

```
""" A code for modeling and diagnosing of traffic crash rates using Poisson hidden Markov models"""
import numpy as np
import matplotlib.pylab as pl
from numpy.random import gamma as Gamma
import scipy.stats as sc
from scipy.stats import poisson
import pandas as pd
from statsmodels.graphics import tsaplots
import itertools
import statsmodels.api as sm
from scipy import stats
from scipy.stats import norm

def Dirichlet(delta,x):
    return np.random.dirichlet(delta+(x),1)[0]

def Multinomial(prob):
    return np.where(np.random.multinomial(1,prob) == 1)[0][0]

def Sufficient_Statistics(data,OO,hid,k):
    # OO: expected crashes.
    # data: observed crashes.
    # hid: vector of hidden states.
    sum_data=np.zeros([k])# sum observed crashes at the state k.
    sum_OO=np.zeros((k)) # sum expected crashes at the state k.
    for j in range(k):
        sum_data[j]=np.sum((hid==j)*data)
        sum_OO[j]=np.sum((hid==j)*OO)
    return sum_data,sum_OO

""" This function is specified for counting the the number of transitions from the state j,  denoted by a
    vector  N_j of dimension 1*K. N_j: sum up from t=1 until t=T. """
def Initial_Number(hid,k):
    N_j=np.zeros((k))
    for j in range(k):
        N_j[j]=np.sum(hid==j)
    return N_j

"""This function is specified for counting the number of transitions from j into k, denoted by a matrix N_jk
    of dimention K*K."""
def Transition_Number(data,hid,k):
    N_jk=np.zeros((k,k)) # N_jk: sum up from t=1 until t=T-1.
    for t in range(len(data)-1):
        N_jk[hid[t],hid[t+1]]=N_jk[hid[t],hid[t+1]]+ 1
    return N_jk
""" A function to update the hidden state locally. Local updating method assumes there is a given hidden
    state sequence. We denoted it as 'hidden' and the updating is done in the new sequence. We denoted
    it as 'hidden_new'."""
def Hidden_updating(k,data,OO,A,pi,lam,hid):
    T=len(data)
    hid_new=np.zeros([T], dtype=int)
    alloc=np.zeros([T,k])
    for i in range(k):
        alloc[0,i]=pi[i]*A[i,hid[1]]*((np.e**(-OO[0]*lam[i]))*((lam[i])**data[0]))
    alloc[0,:]=alloc[0,:]*1./(np.sum(alloc[0,:]))
    hid_new[0]=Multinomial(alloc[0,:])
    for t in range(1,T-1):
        for i in range(k):
            alloc[t,i]=A[hid_new[t-1],i]*((np.e**(-OO[t]*lam[i]))*((lam[i])**data[t]))*A[i,hid[t+1]]
        alloc[t,:]=alloc[t,:]*1./(np.sum(alloc[t,:]))
        hid_new[t]=Multinomial(alloc[t,:])
    for i in range(k):
        alloc[T-1,i]=A[hid_new[T-2],i]*((np.e**(-OO[T-1]*lam[i]))*((lam[i])**data[T-1]))
    alloc[T-1,:]=alloc[T-1,:]*1./(np.sum(alloc[T-1,:]))
```

```python
    hid_new[T-1]=Multinomial(alloc[T-1,:])
    return alloc,hid_new

""" Poisson density function """
def Pois_pmf(data, lamd):
    return poisson.pmf(data,lamd)

""" Compute the recursive likelihood fucntion """
def Recursive_loglikelihood(k,data,OO,A,pi,lam):
    T=len(data)
    alpha_hat = np.zeros((k,T))
    alpha_star = np.zeros((k,T))
    C = np.zeros([T])
    for s in range(k):
        alpha_hat[s,0] = pi[s]* poisson.pmf(data[0], (OO[0]*lam[s]))
    C[0]=1.0/(np.sum(alpha_hat[:,0]))
    alpha_star[:,0]=C[0]*alpha_hat[:,0]
    for t in range(1,T):
        for j in range(k):
            for i in range(k):
                alpha_hat[j,t] += np.dot(alpha_star[i,t-1],A[i,j])* poisson.pmf(data[t], (OO[t]*lam[j]))
        C[t]=1.0/(np.sum(alpha_hat[:,t]))
        alpha_star[:,t]=C[t]*alpha_hat[:,t]
    return -np.sum(np.log(C))

""" Compute the conditional likelihood fucntion """
def Conditional_loglikelihood(k,data,OO,A,pi,lam,hidden):
    # define a function to compute the conditional log likelihood over  integrating # out the hidden states.
    alloc,hid_new=Hidden_updating(k,data,OO,A,pi,lam,hid)
    T=len(data)
    con_loglike= np.zeros((T))
    for t in range(T):
        for s in range(k):
            con_loglike[t]+=alloc[t,s]*np.log(poisson.pmf(data[t],(OO[t]*lam[s])))
    return np.sum(con_loglike)

""" Run the Direct Gibbs sampler for samplimg from K-state Poisson HMM """
def PoissonHMM_DirectGibbs(obs,O,T,M,d,K,ch,a,b,delta):
    # obs: observed crash.
    # O: expected crash.
    # T: lenght observed crash.
    # a,b: the shape and scale parameters of Gamma.
    # delta: the parameter of Dirichlet.

    """Storage posteriors"""
    N_jk=np.zeros([ch,M,K,K], dtype=int) # the number of transitions from state j into k.
    N_j=np.zeros([ch,M,K], dtype=int) # the number of transitions in state j.
    sum_y=np.zeros([ch,M,K],np.float) # sum of observed crashes
    sum_O=np.zeros([ch,M,K],np.float)# sum of expected crashes
    alloc_post=np.zeros([ch,M,T,K],np.float) # allocation probabilities
    z_post=np.zeros([ch,M,T], dtype=int) # hidden states
    tran_post=np.zeros([ch,M,K,K],np.float) # tansition matrix
    init_post=np.zeros([ch,M,K],np.float) # initial state vector
    lamd_post=np.zeros([ch,M,K],np.float) # crash rate parameter
    recursive_loglikelihood_post=np.zeros([ch,M],np.float) # recursive log-likelihood
    conditional_loglikelihood_post=np.zeros([ch,M],np.float)# conditional log_likelihood
    for f in range(ch): # running over L chains
        for r in range(K):
           #Initialization
           lamd_post[f,0,r]=Gamma(shape=(1.0+(50*f)),scale=1.0,size=1) [0]
           tran_post[f,0,r,:]=Dirichlet(delta,np.ones((K)))
        init_post[f,0,:]=Dirichlet(delta,np.ones((K)))

    """Compute the hidden states at iteration 0"""
    z_post[f,0,0]=Multinomial(init_post[f,0,:])
    for t in range(1,T):
        z_post[f,0,t]=Multinomial(tran_post[f,0,z_post[f,0,t-1],:])

    """ Compute the recursive and conditional likelihoods at iteration 0"""
    recursive_loglikelihood_post[f,0]=Recursive_loglikelihood(K,obs,O,tran_post[f,0,:,:],
                                                              init_post[f,0,:],lamd_post[f,0,:])
    conditional_loglikelihood_post[f,0]=Conditional_loglikelihood(K,obs,O,tran_post[f,0,:,:],
                                                              init_post[f,0,:],lamd_post[f,0,:],z_post[f,0,:])

    """MCMC Running"""
    for m in range(1,M):
        """Sufficient statistics"""
        sum_y[f,m,:]= Sufficient_Statistics(obs,O,z_post[f,m-1,:],K)[0]
        sum_O[f,m,:]= Sufficient_Statistics(obs,O,z_post[f,m-1,:],K)[1]

        """ Updating rate parameter, lmabda"""
        lamd_post[f,m,:]=Gamma(shape=a+sum_y[f,m,:],scale=1./(b+sum_O[f,m,:]))

        """Apply Artificial Constraints (IC) to handle the label switching"""
        IC=sorted(lamd_post[f,m,:])
        lamd_post[f,m,:]=IC

        """ Updating the initial and transition parameters"""
        N_jk[f,m,:,:]= Transition_Number(obs,z_post[f,m-1,:],K)
        N_j[f,m,:]= Initial_Number(z_post[f,m-1,:],K)
        for r in range(K):
            tran_post[f,m,r,:]= Dirichlet(delta,N_jk[f,m,r,:])
        init_post[f,m,:]= Dirichlet(delta,N_j[f,m,:])

        """ Compute the allocation and hidden states"""
       alloc_post[f,m,:,:]= Hidden_updating(K,obs,O,tran_post
       [f,m,:,:],init_post[f,m,:],lamd_post[f,m,:],z_post[f,m-1,:])[0]
       z_post[f,m,:]= Hidden_updating(K,obs,O,tran_post[f,m,:,:],init_post[f,m,:],lamd_post[f,m,:],z_post[f,m-1,:])[1]

       recursive_loglikelihood_post[f,m]=Recursive_loglikelihood(K,obs,O,tran_post[f,m,:,:],init_post[f,m,:],
                                                                   lamd_post[f,m,:])
       conditional_loglikelihood_post[f,m]=Conditional_loglikelihood(K,obs,O,tran_post[f,m,:,:],
                                                                       init_post[f,m,:],lamd_post[f,m,:],z_post[f,m,:])

    return alloc_post,z_post,tran_post,init_post,lamd_post,
    recursive_loglikelihood_post,conditional_loglikelihood_post

""" Sub-code to implement the thinning """
def tinning(obs,O,T,M,d,K,ch,a,b,delta):
    colloction_posteriors=list()
    alloc_post,z_post,tran_post,init_post,lamd_post,
    recursive_loglikelihood_post,conditional_loglikelihood_post
   =PoissonHMM_DirectGibbs(obs,O,T,M,d,K,ch,a,b,delta)
    colloction_posteriors.append((alloc_post,z_post,tran_post,init_post,
    lamd_post,recursive_loglikelihood_post,conditional_loglikelihood_post))
    lamda_thin=np.zeros([ch,thin,K])
    init_thin=np.zeros([ch,thin,K])
    trans_thin=np.zeros([ch,thin,K,K])
    alloc_thin=np.zeros([ch,thin,T,K])
    Z_thin=np.zeros([ch,thin,T])
    Recursive_loglikelihood_thin=np.zeros([ch,thin])
    Conditional_loglikelihood_thin=np.zeros([ch,thin])
    for f in range(ch):
        for m in range(thin):
            lamda_thin[f,m,:]=lamd_post[f,d[lag*m],:]
            init_thin[f,m,:]=init_post[f,d[lag*m],:]
            trans_thin[f,m,:,:]=tran_post[f,d[lag*m],:,:]
            alloc_thin[f,m,:,:]=alloc_post[f,d[lag*m],:,:]
            Z_thin[f,m,:]=z_post[f,d[lag*m],:]
            Recursive_loglikelihood_thin[f,m]=recursive_loglikelihood_post
            [f,d[lag*m]]
            Conditional_loglikelihood_thin[f,m]=
            conditional_loglikelihood_post
            [f,d[lag*m]]

    return lamda_thin,init_thin,trans_thin,alloc_thin,Z_thin,Recursive_loglikelihood_thin,
            Conditional_loglikelihood_thin, colloction_posteriors
""" Gelman_Rubin statistic """
def Convergance_DG(obs,O,T,M,d,K,ch,thin,lag,a,b,delta):
    lamda_thin,init_thin,trans_thin,alloc_thin,Z_thin,
    Recursive_loglikelihood_thin,
    Conditional_loglikelihood_thin,colloction_posteriors=
    tinning(obs,O,T,M,d,K,ch,a,b,delta)
    mean_lamd_thin=np.zeros([K,ch])
    mean_mean_lamd_thin=np.zeros([K])
    mean_init_thin=np.zeros([K,ch])
    mean_mean_init_thin=np.zeros([K])
    mean_tran_thin=np.zeros([K,K,ch])
    mean_mean_tran_thin=np.zeros([K,K])
    Within_lamd_thin=np.zeros([K,ch])
    Within_tran_thin=np.zeros([K,K,ch])
    Within_init_thin=np.zeros([K,ch])
    W_lamd_thin=np.zeros([K])
    W_tran_thin=np.zeros([K,K])
    W_init_thin=np.zeros([K])
    Between_lamd_thin=np.zeros([K])
    Between_tran_thin=np.zeros([K,K])
    Between_init_thin=np.zeros([K])
    VAR_lamd_thin=np.zeros([K])
    VAR_tran_thin=np.zeros([K,K])
    VAR_init_thin=np.zeros([K])
    R_hat_lamd_thin=np.zeros([K]) #R_hat for mean parameter
    R_hat_tran_thin=np.zeros([K,K])# R_hat for transition parameters
    R_hat_init_thin=np.zeros([K])# R_hat for initial parameter

    for k in range(K):
        for f in range(ch):
        # compute mean each parameter for each chain
            mean_lamd_thin[k,f]=np.mean(lamda_thin[f,:,k])
            mean_init_thin[k,f]=np.mean(init_thin[f,:,k])
            for s in range(K):
                mean_tran_thin[k,s,f]=np.mean(trans_thin[f,:,k,s])
            mean_mean_lamd_thin[k]=np.mean(mean_lamd_thin[k,:])
            mean_mean_init_thin[k]=np.mean(mean_init_thin[k,:])
            for s in range(K):
                mean_mean_tran_thin[k,s]=np.mean(mean_tran_thin[k,s,:])
            # compute with-in variance each paramter for each chain
            for f in range(ch):
                Within_lamd_thin[k,f]=(1./(thin-1))*(np.sum(((lamda_thin[f,:,k ]-mean_lamd_thin[k,f])**2)))
                Within_init_thin[k,f]=(1./(thin-1))*(np.sum(((init_thin[f,:,k ]-mean_init_thin[k,f])**2)))
                for s in range(K):
                    Within_tran_thin[k,s,f]=(1./(thin-1))*(np.sum(((trans_thin[f,:,s,k]-mean_tran_thin[k,s,f])**2)))
         W_lamd_thin[k]=np.sum(Within_lamd_thin[k,:],axis=0)*1./(ch)
         W_init_thin[k]=np.sum(Within_init_thin[k,:],axis=0)*1./(ch)
         for s in range(K):
             W_tran_thin[k,s]=np.sum(Within_tran_thin[k,s,:],axis=0)*1./(ch)
         # compute between-variance each of each chain for each parameter
         Between_lamd_thin[k]=(thin*1./(ch-1))*(np.sum((mean_lamd_thin[k,:] - mean_mean_lamd_thin[k])**2,axis=0))
         Between_init_thin[k]=(thin*1./(ch-1))*(np.sum((mean_init_thin[k,:] - mean_mean_init_thin[k])**2,axis=0))
         for s in range(K):
             Between_tran_thin[k,s]=(thin*1./(ch-1))*(np.sum((mean_tran_thin[k,s,:] - mean_mean_tran_thin[k,s])
                                    **2,axis=0))
```

```python
        # compute the variances of chains
        VAR_lamd_thin[k]=((1-(1./(thin)))*W_lamd_thin[k]) +((1./(thin))*Between_lamd_thin[k])
        VAR_init_thin[k]=((1-(1./(thin)))*W_init_thin[k]) +((1./(thin))*Between_init_thin[k])
        for s in range(K):
             VAR_tran_thin[k,s]=((1-(1./(thin)))*W_tran_thin[k,s])+((1./(thin))*Between_tran_thin[k,s])

        # compute R_hat of all paramters
        R_hat_lamd_thin[k]= np.sqrt(VAR_lamd_thin[k]*1./(W_lamd_thin[k]))
        R_hat_init_thin[k]= np.sqrt(VAR_init_thin[k]*1./(W_init_thin[k]))
        for s in range(K):
            R_hat_tran_thin[k,s]= np.sqrt(VAR_tran_thin[k,s]*1./(W_tran_thin[k,s]))

    return lamda_thin,init_thin,trans_thin,alloc_thin,Z_thin,Recursive_loglikelihood_thin,
    Conditional_loglikelihood_thin, R_hat_lamd_thin,R_hat_init_thin,R_hat_tran_thin,colloction_posteriors

""" Thinning the chains for each parameter"""
def thinning_chains(obs,O,T,M,d,K,ch,Total_thin,a,b,delta):
    lamda_thin,init_thin,trans_thin,alloc_thin,Z_thin,
    Recursive_loglikelihood_thin,Conditional_loglikelihood_thin, R_hat_lamd_thin,
    R_hat_init_thin,R_hat_tran_thin,colloction_posteriors
    =Convergance_DG(obs,O,T,M,d,K,ch,thin,lag,a,b,delta)
    # keep all thinned samples
    thin_lamds=np.zeros([Total_thin,K])
    thin_inits=np.zeros([Total_thin,K])
    thin_trans=np.zeros([Total_thin,K,K])
    thin_Z=np.zeros([Total_thin,T])
    thin_allocs=np.zeros([Total_thin,T,K])
    thin_recursive_loglikelihoods=np.zeros([Total_thin])
    thin_conditional_loglikelihoods=np.zeros([Total_thin])
    for s in range(K):
        thin_lamds[:,s]=list(itertools.chain(lamda_thin[0,:,s],lamda_thin[1,:,s],lamda_thin[2,:,s]))
        thin_inits[:,s]=list(itertools.chain(init_thin[0,:,s],init_thin[1,:,s],init_thin[2,:,s]))
        for r in range(K):
            thin_trans[:,s,r]=list(itertools.chain(trans_thin[0,:,s,r],trans_thin[1,:,s,r],trans_thin[2,:,s,r]))
    for t in range(T):
        thin_Z[:,t]=list(itertools.chain(Z_thin[0,:,t],Z_thin[1,:,t], Z_thin[2,:,t]))
        for s in range(K):
             thin_allocs[:,t,s]=list(itertools.chain(alloc_thin[0,:,t,s],alloc_thin[1,:,t,s],alloc_thin[2,:,t,s]))
     thin_recursive_loglikelihoods=list(itertools.chain(Recursive_loglikelihood_thin[0,:],
     Recursive_loglikelihood_thin[1,:], Recursive_loglikelihood_thin[2,:]))
     thin_conditional_loglikelihoods=list(itertools.chain(Conditional_loglikelihood_thin[0,:],
     Conditional_loglikelihood_thin[1,:], Conditional_loglikelihood_thin[2,:]))

    return thin_lamds,thin_inits,thin_trans,thin_Z,thin_allocs,thin_recursive_loglikelihoods,
    thin_conditional_loglikelihoods, R_hat_lamd_thin,R_hat_init_thin, R_hat_tran_thin, colloction_posteriors

""" model selection: AIC, BIC and DIC"""
def Model_selection(obs,K,O,T,M,d,Total_thin,a,b,delta):
    Collection_colloction_posteriors=[]
    # storage for criteria
    Collection_AIC_BIC_3=[]
    Collection_DIC_rec_2=[]
    Collection_DIC_con_2=[]
    replicated_results=[]

    """call the thinned chains"""
    thin_lamds,thin_inits,thin_trans,thin_Z,thin_allocs,
    thin_recursive_loglikelihoods,thin_conditional_loglikelihoods,
    R_hat_lamd_thin,R_hat_init_thin,R_hat_tran_thin,colloction_posteriors=
    thinning_chains(obs,O,T,M,d,K,ch,Total_thin,a,b,delta)
    colloction_thinned_posteriors=[]
    colloction_thinned_posteriors.append([thin_lamds,thin_inits,thin_trans,thin_Z,
    thin_allocs, thin_recursive_loglikelihoods,thin_conditional_loglikelihoods,
     R_hat_lamd_thin, R_hat_init_thin,R_hat_tran_thin,colloction_posteriors])

    """ compute the means of the posteriors """
    mean_lamd=np.zeros([K])
    mean_transition=np.zeros([K,K])
    mean_initial=np.zeros([K])
    for s in range(K):
        mean_lamd[s]=np.mean(thin_lamds[:,s])
        mean_initial[s]=np.mean(thin_inits[:,s])
        # the posterior probabilies of each segemnt given different states.
        for r in range(K):
            mean_transition[r,s]=np.mean(thin_trans[:,r,s])

    """AIC and BIC"""
    Free_para=(K**2)+(K)-1
    AIC_rec_3=-2*np.max(thin_recursive_loglikelihoods)+(2*Free_para)
    BIC_rec_3=-2*np.max(thin_recursive_loglikelihoods) +(np.log(T)*(Free_para))
    """recursive DIC"""
    bar_D_rec=-2*np.mean(thin_recursive_loglikelihoods)
    D_hat_rec=-2*np.max(thin_recursive_loglikelihoods)
    p_DIC_rec2=bar_D_rec-D_hat_rec
    DIC_rec2=bar_D_rec+p_DIC_rec2
    """conditional DIC"""
    bar_D_con=-2*np.mean(thin_conditional_loglikelihoods)
    D_hat_con=-2*np.max(thin_conditional_loglikelihoods)
    p_DIC_con2=bar_D_con-D_hat_con
    DIC_con2=bar_D_con+p_DIC_con2

    Collection_AIC_BIC_3.append([AIC_rec_3,BIC_rec_3])
    Collection_DIC_rec_2.append([bar_D_rec,D_hat_rec,p_DIC_rec2,DIC_rec2])
    Collection_DIC_con_2.append([bar_D_con,D_hat_con,p_DIC_con2,DIC_con2])
    Collection_colloction_posteriors.append([colloction_posteriors])
    return Collection_AIC_BIC_3,Collection_DIC_rec_2,Collection_DIC_con_2,replicated_results,
    Collection_colloction_posteriors, colloction_thinned_posteriors

"""data"""
obs=obs # where obs is crash counts
O= O # wheer O is expected crash counts
CR=CR# where CR is the observed crash rates
"""MCMC information"""
K=K # K is the number of states
M=M # iterations
burnin=burnin  # burn-in period
d=range(burnin,M) # the kept samples after discarding the burn-in period

""" Four different priors"""
a=np.array([0.1,0.01,0.001,0.0001]) # values of the shape paramter of Gamma
b=np.array([0.1,0.01,0.001,0.0001]) # values of the scale paramter of Gamma
delta=np.ones((K),int) # delta value of Dirichlet distribution
T=len(obs) # length data
ch=3  # the number of chains
lag=100  # thinning lag
thin=int(len(d)*1./lag)
Total_thin=(thin*ch) # length of thinned chain
Results=[]
for i  in range(len(a)):
    """ selection model over different priors"""
    Collection_AIC_BIC_3,Collection_DIC_rec_2,Collection_DIC_con_2,
    replicated_results, Collection_colloction_posteriors,
    colloction_thinned_posteriors=Model_selection(obs,K,O,T,M,d,Total_thin,a[i],b[i],delta)
    Results.append([Collection_AIC_BIC_3,Collection_DIC_rec_2,Collection_DIC_con_2,replicated_results, colloction_thinned_posteriors])

"""Results of model selection criteria: AIC, BIC, DIC and WAIC"""
print 'AIC_rec_3=',[Results[i][0][0][0] for i in range(len(a))]
print 'BIC_rec_3=',[Results[i][0][0][1] for i in range(len(a))]
print 'bar_D_rec',  [Results[i][1][0][0] for i in range(len(a))]
print 'D_hat_rec',  [Results[i][1][0][1] for i in range(len(a))]
print 'p_DIC_rec2=',[Results[i][1][0][2] for i in range(len(a))]
print 'DIC_rec2=',  [Results[i][1][0][3] for i in range(len(a))]
print 'bar_D_con',[Results[i][2][0][0] for i in range(len(a))]
print 'D_hat_con',[Results[i][2][0][1] for i in range(len(a))]
print 'p_DIC_con2',[Results[i][2][0][2] for i in range(len(a))]
print 'DIC_con2',  [Results[i][2][0][3] for i in range(len(a))]

"""Convereganece Diagnostic results"""
#1-Gelman statistic R for convergence
for i in range(len(a)):
    for j in range(K):
        R_hat=Results[i][5][0][7][j]
        print 'Gelman lamd',j+1,'=',R_hat

#2- the Geweke Diagnostic for convergence
def Geweke(trace):
    L1=np.round(0.1*len(trace))
    L2=np.round(0.5*len(trace))
    s1=trace[0:L1]
    s2=trace[L2:]
    var_s1 = np.var(s1)*1./(L1)
    var_s2 = np.var(s2)*1./(L2)
    z=(np.mean(s1)-np.mean(s2))*1./(np.sqrt( var_s1 + var_s2))
    return z
    for i in range(len(a)):
        for j in range(K):
             Z_score=Geweke(Results[i][5][0][0][:,j])
             print 'Geweke lamd',j+1,'=',Z_score

"""Compute and plot the iPPD, given  Gamma(0.1,0.1) prior"""
y_replication=np.zeros([Total_thin,T],int)# replications
for m in range(Total_thin):
    for t in range(T):
        y_replication[m,t]=(np.random.poisson(Results[0][5][0][0][m,Results[0][5][0][3][m,t]]*O[t]))

""" Plot centers and 95% CI of iPPD vs the observed crash count"""
fig, ax = pl.subplots(1, 1)
pl.xlim(-2, 92)
pl.ylim(-7, 140)
x = np.arange(1,91,1)
ax.set_xticks(x)
pl.plot(x,obs, 'rd',linewidth=5.0, label='Observed')
y= np.mean(y_replication, axis=0, dtype=np.float64)
ci95 = np.abs(y - 1.96 * sc.sem(y_replication, axis=0))
pl.errorbar(x, y, yerr=ci95, fmt='o', label='95% CI Predicted')
ax.set_xticks(x)
pl.title('$K=3$',fontsize=16)
pl.xlabel('Segment',fontsize=16)
pl.ylabel('Predictive vs. Observed',fontsize=16)
pl.legend(loc='upper right')
pl.show()

""" QQ_plots for the pseudo-residuals""""
y_residual=np.zeros([Total_thin,T])
for m in range(Total_thin):
    y_residual[m,:]=(y_replication[m,:]<obs)
    ave_residual_3=np.sum(y_residual,axis=0)*1./(Total_thin)
    sm.qqplot(ave_residual_3, stats.t, fit=True, line='45')
pl.show()
```

Bibliography

(2016). Gb road traffic counts. https://data.gov.uk/dataset/gb-road-traffic-counts. Accessed: 2016-09-30.

(2016). Road safety data. https://data.gov.uk/dataset/road-accidents-safety-data. Accessed: 2016-09-30.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. Petrov and F. Csaki (Eds.), *Proceedings of the Second International Symposium on Information*, Budapest, pp. 267–281.

Albert, J. H. and S. Chib (1993). Bayes inference via Gibbs sampling of autoregressive time series subject to Markov mean and variance shifts. *Journal of Business & Economic Statistics* ***11***(1), 1–15.

Albert, P. S. (1991). A two-state Markov mixture model for a time series of epileptic seizure counts. *Biometrics* ***47***, 1371–1381.

Ando, T. (2010). *Bayesian Model Selection and Statistical Modeling*. Chapman and Hall/CRC, Boca Raton.

ArcGIS (2014). *Version 10.2*. Esri, New York.

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In O. E. In Shisha (Ed.), *Inequalities III: Proceedings of the 3rd Symposium on Inequalities*, University of California, Los Angeles, pp. 1–8.

Baum, L. E. and J. A. Eagon (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Amer. Math. Soc.* ***73***(3), 360–363.

Baum, L. E. and T. Petrie (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics* ***37***(6), 1554–1563.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *In The Annals of Mathematical Statistics **41***(1), 164–171.

Berger, J. O. and J. M. Bernardo (1989). Estimating a product of means: Bayesian analysis with reference priors. *Journal of the American Statistical Association **84***(405), 200–207.

Bhar, R. and S. Hamori (2006). Hidden Markov models: Applications to financial economics.

Biernacki, C., G. Celeux, and G. Govaert (2001). Mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence **22***(7).

Billio, M., A. Monfort, and C. P. Robert (1999). Bayesian estimation of switching arma models. *Journal of Econometrics **93***(2), 229–255.

Bilmes, J. A. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models. *Technical Report TR-97-021.*

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Brooks, S. (2002). Discussion on the paper by Spiegelhalter, Best, Carlin and van der Linde. *Journal of the Royal Statistical Society, B **64***, 616–618.

Burnham, K. and D. Anderson (2002). *Model Selection and Inference: A Practical Information-Theoretical Approach.* 2d ed. New York: Springer-Verlag.

Burnham, K. and D. Anderson (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods and Research **33***, 261–304.

Cappé, O., E. Moulines, and T. Rydén (2005). *Inference in Hidden Markov Models.* New York.

Carlin, B. P. (2006). Comment on article by celeux, et al. *Bayesian analysis **4***(1), 675–676.

Carlin, B. P. and T. A. Louis (2009). *Bayesian Methods for Data Analysis* (3 ed.). Boca Raton, FL: Chapman and Hall/CRC Press.

Casella, G. and C. P. Robert (2004). *Monte Carlo Statistical Methods.* Springer.

Celeux, G. and J. B. Durand (2008). Selecting hidden Markov model state number with cross-validated likelihood. *Computational Statistics **23***(4), 541–564.

Celeux, G., F. Forbes, C. P. Robert, and D. M. Titterington. (2006). Deviance information criteria for missing data models. *Bayesian Analysis* ***1***(4), 651–673.

Celeux, G., M. Hurn, and C. P. Robert (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association* ***451***(95), 957–970.

Chib, S. (1996). Calculating posterior distributions and modal estimates in Markov mixture models. *Journal of Econometrics* ***75***(1), 79–97.

Congdon, P. (2014). *Applied Bayesian Modelling* (2 ed.). Mew York: John Wiley & Sons.

Cooper, B. and M. Lipsitch (2004). The analysis of hospital infection data using hidden Markov models. *Biostatistics 5*(2), 223–237.

Cowles, M. K. and B. P. Carlin (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association* ***91***(434), 883–904.

Crawford, S. L. (1994). An application of the laplace method to finite mixture distributions. *Journal of the American Statistical Association* ***89***(425), 259–267.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of Royal Statistical Society, Serial B.* ***39***(1), 1–38.

Derrode, S., L. Benyoussef, and W. Pieczynski (2006). Contextual estimation of hidden Markov chains with application to image segmentation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, Volume **2**, pp. II–II. IEEE.

Diebolt, J. and C. Robert (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society, B* ***56***(2), 363–375.

Dymarski, P. (2011). *Hidden Markov Models, Theory and Applications*. Janeza Trdine 9, 51000 Rijeka, Croatia.

Fan, Y. and S. A. Sisson (2011). Reversible jump MCMC. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*, Chapter 3, pp. 67–91. CRC press.

FHWA (Accessed January, 2011). Roadway safety information analysis: A manual for local rural road owners.

Frühwirth-Schnatter, S. (2001). Markov chain Monte Carlo estimation of classical and dynamic switching and mixture models. *Journal of the American Statistical Association* ***96***(453), 194–209.

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer Series in Statistics. Springer, New York.

Gassiat, E. and C. Keribin (2000). The likelihood ratio test for the number of components in a mixture with Markov regime. *ESAIM: Probability and Statistics* ***4***, 25–52.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association* ***70***(350), 320–328.

Geisser, S. and W. F. Eddy (1979). A predictive approach to model selection. *Journal of the American Statistical Association* ***74***(365), 153–160.

Gelfand, A. E. and D. K. Dey (1994). Bayesian model choice: Asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B (Methodological)* ***56***, 501–514.

Gelfand, A. E. and S. K. Ghosh (1998). Model choice: A minimum posterior predictive loss approach. *Biometrika* ***85***(1), 1–11.

Gelman, A., J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin (2014). *Bayesian Data Analysis* (3 ed.). Chapman and Hall/CRC.

Gelman, A. and J. Hill (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press, Cambridge, UK.

Gelman, A., J. Hwang, and A. Vehtari (2014). Understanding predictive information criterion for Bayesian models. *Statistics and Computing* ***24***(6), 997–1016.

Gelman, A., X.-L. Meng, and H. S. Stern (1996). Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica sinica*, 733–760.

Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequence. *Statistical science* ***7***, 457–511.

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

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In J. Bernardo, J. Berger, J. Dawid, and A. Smith (Eds.), *Bayesian Statistics 4*, pp. 169–193. Oxford, UK: Oxford University Press.

Geyer, C. J. (2011). Introduction to Markov chain Monte Carlo. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*, Chapter 1, pp. 3–48. CRC press.

Green, P. J. and S. Richardson (2002). Hidden Markov models and disease mapping. *Journal of the American statistical association* ***97***(460), 1055–1070.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society* ***57***, 357–384.

Han, C. and B. P. Carlin (2011). Markov chain Monte Carlo methods for computing Bayes factors. *Journal of the American Statistical Association*.

Hardle, W. (1991). *Smoothing techniques*. Springer Series in Statistics, New York.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* ***57***(1), 97–109.

Hurn, M., A. Justel, and C. P. Robert (2003). Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics* ***12***(1), 55–79.

Jackson, C. H., L. D. Sharples, S. G. Thompson, S. W. Duffy, and E. Couto (2003). Multistate Markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)* ***52***(2), 193–209.

Jasra, A., C. Holmes, and D. Stephens (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science* ***1***(20), 50–67.

Jeffreys, H. (1961). *Theory of Probability*. Oxford: Oxford University Press.

Johnson, M. (2007, June). Why doesn't EM find good HMM POS-taggers?. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, pp. 296–305. Association for Computational Linguistics.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association* ***90***(430), 773–795.

Konishi, S. and G. Kitagawa (2008). *Information Criteria and Statistical modeling*. Springer-Verlag, New York.

Kroese, D. P. and J. C. Chan (2014). *Statistical modeling and computation*. Springer.

Kullback, S. and R. A. Leibler (1951). On information and sufficiency. *The annals of mathematical statistics* ***22***(1), 79–86.

Laud, P. and J. Ibrahim (1995). Predictive model selection. *Journal of the Royal Statistical Society. Series B (Methodological)* ***57***, 247–262.

Leroux, B. G. and M. L. Puterman (1992). Maximum-penalized likelihood estimation for independent and Markov-dependent mixture models. *Biometrics* ***48***(2), 545–558.

Li, L., L. Zhu, and D. Z. Sui (2007). A gis-based Bayesian approach for analyzing spatial-temporal patterns of intra-city motor vehicle crashes. *Journal of Transport Geography* ***15***(4), 274–285.

Li, Y., T. Zeng, and J. Yu (2015). Robust deviance information criterion for latent variable models. *SMU Economics and Statistics Working Paper Series*. http://www.mysmu.edu/faculty/yujun/Research/BCE36.pdf.

Marin, J.-M., K. Mengersen, and C. P. Robert (2005). Bayesian modelling and inference on mixtures of distributions. *Handbook of statistics* ***25***, 459–507.

Marin, J.-M. and C. Robert (2014). *Bayesian Essentials with R* (2 ed.). Springer New York Heidelberg Dordrecht London.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models*. Springer US.

McGillivray, A. and A. Khalili (2014). A new penalized quasi-likelihood approach for estimating the number of states in a hidden Markov model. *Perspectives on Big Data Analysis: Methodologies and Applications* ***622***, 37–48.

McLachlan, G. and D. Peel (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics, New York, USA.

Meng, X.-L. (1994). Posterior predictive p-values. *The Annals of statistics* ***22***(3), 1142–1160.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, H. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *The journal of chemical physics* ***21***(6), 1087–1092.

Neal, R. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo*, Chapter 5, pp. 113–162. CRC press.

Ntzoufras, I. (2009). *Bayesian Modeling Using WinBUGS*. Wiley Series in Computational Statistics, Hoboken, USA.

Papastamoulis, P. and G. Iliopoulos (2010). An artificial allocations based solution to the label switching problem in Bayesian analysis of mixtures of distributions. *Journal of Computational and Graphical Statistics* ***19***(2), 313–331.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceeding of the IEEE* ***77***(2), 257–286.

Rafei, A., E. Pasha, and R. J. Orak (2012). Tuberculosis surveillance using a hidden Markov model. *Iranian journal of public health* ***41***(10), 87.

Raftery, A. (1995). Bayesian model selection in social research. *Sociological Methodology* ***25***, 111–163.

Raftery, A. and S. Lewis (1992). How many iterations in the Gibbs sampler? In J. Bernardo, J. Berger, J. Dawid, and A. Smith (Eds.), *Bayesian Statistics 4*, pp. 763–773. Oxford, UK: Oxford University Press.

Rahimi, A. (2000). An erratum for "A tutorial on hidden Markov models and selected applications in speech recognition". `http://xenia.media.mit.edu/rahimi/rabiner/rabinererrata/rabiner-errata.html`.

Raijmakers, M. E. and P. Molenaar (2004). Modeling developmental transitions in adaptive resonance theory. *Developmental Science* ***7***(2), 149–157.

Richardson, S. (2002). Discussion on the paper by Spiegelhalter, Best, Carlin and van der Linde. *Journal of the Royal Statistical Society, B* ***64***, 626–627.

Richardson, S. and P. Green (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)* ***59***(4), 731–792.

Rizzo, M. L. (2008). *Statistical computing with R.* Chapman and Hall.

Robert, C., T. Ryden, and D. Titterington (1999). Convergence controls for MCMC algorithms, with applications to hidden Markov chains. *Journal of Statistical Computing and Simulation* ***64***(4), 327–355.

Robert, C. and D. Titterington (1998). Reparameterization strategies for hidden Markov models and Bayesian approaches to maximum-likelihood estimation. *Statistics and Computing* ***8***(2), 145–158.

Robert, C. P., G. Celeux, and J. Diebolt (1993). Bayesian estimation of hidden Markov models: A stochastic implementation. *Statistics & Probability Letters* ***16***(1), 77–83.

Robert, C. P., T. Ryden, and D. M. Titterington (2000). Bayesian inference in hidden Markov models through the reversible jump Markov chain Monte Carlo method. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* ***62***(1), 57–75.

Rydén, T. (2008). EM versus Markov chain Monte Carlo for estimation of hidden Markov models: A computational perspective. *Bayesian Analysis* ***3***(4), 659–688.

Schlattmann, P. (2009). *Medical Applications of Finite Mixture Models*. Berlin: Springer.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics* ***6***(2), 461–464.

Scott, S., G. James, and C. Sugar (2005). Hidden Markov models for longitudinal comparisons. *Journal of the American Statistical Association 100*, 359–369.

Scott, S. L. (2002). Bayesian methods for Hidden Markov models: Recursive computing in the 21th century. *Journal of the American Statistical Association* ***97***(457), 337–351.

Serfling, R. E. (1963). Methods for current statistical analysis of excess pneumonia-influenza deaths. *Public health reports 78*(6), 494–506.

Spiegelhalter, D., N. Best, B. Carlin, and A. van der Linde (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* ***64***(4), 583–639.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology* ***4***(62), 795–809.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)* ***36***, 111–147.

Tanner, M. Y. and W. H. Wong (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association* ***82***(398), 528–540.

Vehtari, A. and J. Ojanen (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys* ***6***, 142–228.

Visser, I. (2011). Seven things to remember about hidden Markov models: A tutorial on Markovian models for time series. *Journal of Mathematical Psychology* ***55***(6), 403–415.

Visser, I., M. E. J. Raijmakers, and P. C. M. Molenaar (2002). Fitting hidden Markov models to psychological data. *Scientific Programming* ***10***(3), 185–199.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory 13*(2), 260–269.

Wall, M. M. and R. Li (2009). Multiple indicator hidden Markov model with an application to medical utilization data. *Statistics in medicine* ***28***(2), 293–310.

Watanabe, S. (2009). *Algebraic geometry and statistical learning theory*. Cambridge University Press.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research* ***11***(Dec), 3571–3594.

Zhu, L. and B. P. Carlin (2000). Comparing hierarchical models for spatio-temporally misaligned data using the deviance information criterion. *Statistics in Medicine* ***19***(17-18), 2265–2278.

Zucchini, W. and I. L. MacDonald (2009). *Hidden Markov Models for Time Series: An Introduction Using R*. Chapman and Hall, London.